



Entry strategy of Southwest Airlines

Jonathan Cobb, Nico Metzger, Steve Lawford

► To cite this version:

Jonathan Cobb, Nico Metzger, Steve Lawford. Entry strategy of Southwest Airlines. 2009. hal-01021532

HAL Id: hal-01021532

<https://enac.hal.science/hal-01021532>

Submitted on 11 Jul 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Entry strategy of Southwest Airlines*

JONATHAN COBB and NICO METZGER

(supervisor: STEVE LAWFORD)[†]

Department of Economics and Econometrics, ENAC, France.

January 23, 2009

*Journal of Economic Literature (JEL) classification: C01 Econometrics, C13 Estimation, C25 Discrete regression and qualitative choice models; discrete regressors, C51 Model construction and estimation, C80 Data collection and data estimation methodology; computer programs - general, L10 Market structure, firm strategy, and market performance, L93 Air transportation (see http://www.aeaweb.org/journal/jel_class_system.html for classification system).

Keywords: airline strategy, discrete choice models, entry models, Southwest Airlines.

[†]Supervisor contact details: Steve Lawford, LH/ECO, ENAC, 7 avenue Edouard Belin, BP 54005, 31055, Toulouse, Cedex 4, France. Tel: +33 5 62 17 43 37. Fax: +33 5 62 17 40 17. Email: steve_lawford@yahoo.co.uk.

Executive Summary

This report presents the results of a third year project “*Projet de Synthèse*” of the “*IENAC06T*” course, and was completed between Oct 2008 and Jan 2009.

The aim of this project is to understand the main determinants of Southwest Airline’s presence on U.S. domestic routes, over the period 2002 to 2007. A new dataset has been constructed, using quarterly and annual data from the U.S. Department of Transportation (DB1B and T100 Origin and Destination databases), and a number of socio-economic and geographic (regional) variables. A range of discrete choice (probit and logit) models have been built and estimated, both for individual quarters, and for the full sample period, and under various assumptions on the construction of key variables. The resulting model is seen to outperform one of Boguslaski et al.’s (2004, Review of Industrial Organization) recently published Southwest Airlines entry models, in terms of model fit.

The model is used both to examine Southwest’s current route presence, and to form predictions of its likely future behaviour. In particular, it is suggested that Southwest’s announced Mar 2009 entry into Minneapolis-Saint Paul International Airport (MSP) is likely to be more successful than its planned expansion into New York’s LaGuardia Airport (LGA). Further, the model is able to identify likely expansion of routes out of airports that are currently served by Southwest, such as Denver International Airport (DEN) and Bob Hope Airport (BUR), and can be used to explore the potential implications of a repeal of the 1979 Wright Amendment restricting Southwest’s traffic out of Dallas Love Field (DAL).

Contents

1 Southwest Airlines	1
1.1 Company development	1
1.2 Network development	3
2 Core data	12
3 Modelling market and route presence	12
3.1 Motivation and collection of additional data	12
3.2 Socio-economic variables	14
3.3 Geographic (spatial) variables	16
3.3.1 Regional division	16
3.3.2 Distance	17
3.4 Airport-specific variables	18
3.4.1 Political and legal / regulatory issues	18
3.4.2 Hubs and focus airports	19
3.4.3 Secondary airports	21
3.4.4 Airport congestion	23
3.4.5 Low-cost and other competition	24
4 Discrete choice models	25
4.1 Model motivation	25
4.2 A first attempt: the linear probability model	26
4.3 Particular models: the probit and the logit	27
4.4 Further model interpretation (the logit)	29
4.5 Model estimation	29

4.6	Model inference	31
5	Model development	32
5.1	<i>Model 1: Distance and population</i>	32
5.2	<i>Model 2: Carriers and competitors</i>	33
5.3	<i>Model 3: Competitors and quadratic terms</i>	35
5.4	<i>Model 4: Commercial sectors</i>	38
5.5	<i>Model 5: Traffic corridors</i>	38
5.6	<i>Model 6: Significance of airports</i>	40
5.7	<i>Model 7: Secondary airports</i>	41
5.8	<i>Model 8: Congestion effects</i>	43
5.9	<i>Model 9: Low-cost carrier competition</i>	46
6	Model interpretation	49
6.1	Adapting the model to the dataset	49
6.2	Interpretation of the results	50
6.2.1	Initial analysis	50
6.2.2	Marginal effects	55
6.3	Filtered data	59
6.3.1	Entire dataset	64
6.4	Southwest performance	66
6.5	Verification of the model	68
7	Conclusion	77
A	U.S. air traffic hubs	79
B	Competitors of Southwest	80

C	Definition of GDP data	87
D	Metropolitan and micropolitan Statistical Areas	89
E	EViews source code	98
F	Python source code	113

1 Southwest Airlines

1.1 Company development

Southwest Airlines (IATA code WN), is an American airline with its headquarters in Dallas, Texas. It was founded in 1967 as the U.S. domestic aviation market started to deregulate. Following the overruling of several objections by American Airlines (AA), Delta Air Lines (DL), Eastern Airlines (EA) and United Airlines (UA), that were primarily made for strategic reasons, Southwest established its first flights in 1971, offering connections between Dallas Love Field (DAL), Houston, and San Antonio International Airport (SAT).¹

Southwest is often recognized as the first company to have been successful in introducing a low-cost airline business model. Even though no generally accepted definition of a low-cost-carrier exists, there are some key features which identify a low-cost business model, and which are often referred to as the *Southwest Airlines Paradigm*: homogenous fleet (see Table 1), traditionally a dense point-to-point-network (mostly out of less congested secondary airports), internet-based electronic ticketing, and largely reduced on-board-service [14]. Southwest has operated profitably since 1973, and has attracted many other companies around the world to imitate elements of their strategy to some extent.²

¹The first of the Houston airports to be served by Southwest was George Bush Intercontinental Airport (IAH) in 1971, but all operations were moved to William P. Hobby Airport (HOU) in 1972 [2].

²Southwest's concentration on short-haul and medium-haul routes allows it to operate aircraft with similar performance characteristics (see Figure 2). Southwest's major competitors, such as United and Delta, tend to have more diverse networks, with routes reaching from regional to intercontinental long-haul connections, and more heterogeneous fleets (see Table 33). Secondary airports can essentially be characterized as smaller airports (in terms of passenger numbers), that are close to important hub airports. They are often less congested than the hubs, which allows for lower operating costs and greater availability of slots. One such example can be found in Chicago, where Chicago Midway International Airport (MDW), which is used by Southwest, is a secondary airport to O'Hare International Airport (ORD). ORD is an important hub for the major carriers American and United, and an important international gateway.

Type	Fleet size	Range (km)	Capacity (PAX)
Boeing 737-300	188	3360	137
Boeing 737-500	25	2815	122
Boeing 737-700	322	3980	137

Table 1: Southwest fleet composition at June 2008. Range is measured in kilometers for an aircraft with maximum take off weight (MTOW) and the highest possible payload and PAX is the number of passengers

Recently, a shift can be observed in Southwest’s business model. Mainly due to a less successful fuel-hedging strategy, Southwest posted a loss of \$120 million for the third quarter of 2008, its first quarterly loss for 17 years.³ The airline is actively looking for new sources of revenue, and these include an increase in its share of business travellers, which fell from 40% to 25% over the last decade. The Business Select fare category, which provides advantages such as priority check-in and a free on-board drink, is expected to contribute to an annual revenue surplus of about \$100 million. Southwest also plans to extend its market presence through codeshare agreements with the Mexican low-cost carrier Volaris and the Canadian low-cost carrier WestJet. These are due to come into effect in 2009 and 2010 respectively, enabling Southwest to offer international flights [20] [21].

Southwest is currently the world’s largest airline, with more than 100 million passengers carried in 2007 [1]. It operates 3400 daily flights, to 64 cities and 32 states across the U.S., with a fleet of more than 500 Boeing 737 (of the -300, -500 and -700-series as shown in Table 1 [2]). Table 2 illustrates the ranking of Southwest in comparison to its main domestic competitors. It can be seen that in

³Fuel hedging is one means by which an airline can stabilize its fuel costs. A given amount of fuel is purchased for a specified date in the future, based on a pre-determined ‘forward’ price. The airline benefits if the forward price is lower than the market price at the time of delivery. Regarding the 2008:3 quarter, Southwest negotiated prices for these fuel deliveries that were generally higher than the market price [29] [30].

Rank	Airline	Passengers ('000s)
1	Southwest Airlines (WN)	101,911
2	American Airlines (AA)	76,552
3	Delta Air Lines (DL)	61,494
4	United Airlines (UA)	56,399
5	Northwest Airlines (NW)	43,776
6	US Airways (US)	37,194
7	Continental Airlines (CO)	37,094
8	AirTran (FL)	23,705
9	SkyWest (OO)	20,964
10	JetBlue (B6)	20,528

Table 2: Top 10 U.S. airlines, ranked by 2007 Domestic Scheduled Enplanements [4].

addition to the ‘major’ carriers, several other U.S. low-cost carriers like AirTran (FL) and JetBlue (B6) account for a reasonably large number of passengers in the U.S. domestic market.⁴ Hence, Southwest cannot only be considered to compete with the majors, but also with other low-cost carriers (LCCs). For instance, SkyWest operates regional networks across the entire U.S. It employs its fleet for UA as United Express, for DL as Delta Connection and for Midwest Connect. Hence, SkyWest competes with Southwest through its link with these three carriers [9].

1.2 Network development

The area within the circles in Figure 1 around representative airports served by Southwest, on the U.S. west coast (San Diego International Airport, SAN),

⁴The official definition of a U.S. ‘major’ carrier is given by the U.S. Department of Transportation (DOT) [8] as an airline with an annual operations revenue exceeding one billion US dollars. Following this definition, the U.S. major passenger carriers, in addition to AA, CO, DL, NW, UA and US, are AirTran Airways (FL), Alaska Airlines (AS), Frontier Airlines (F9), JetBlue Airways (B6) and Southwest itself. It is also common to deviate from this official definition, and to use ‘major’ in order to identify only AA, CO, DL, NW, UA and US.

the central U.S. (Dallas Love Field, DAL) and the U.S. east coast (Baltimore / Washington International Thurgood Marshall Airport, BWI) illustrates the destinations which can be reached nonstop with Southwest's longest range aircraft, the Boeing 737-700 (which has a maximum range of 2150 nautical miles).



Figure 1: Representative range chart [10] [11].

Flights to Hawaii, even though they would be within the range circle around San Diego, are not offered by Southwest. Flights with a two-engine aircraft such as the Boeing 737, across a region with no nearby alternate airport (like an ocean), require special performance by both aircraft and crew, which are not

satisfied by Southwest.⁵

Southwest already offered its passengers access to Hawaii through a codeshare agreement with the Indianapolis based carrier ATA Airlines between 2005 and 2008. The agreement with ATA also allowed Southwest to have (codeshared) access to New York's LaGuardia (LGA) and Ronald Reagan Washington National Airport (DCA). These airports are, with John F. Kennedy International Airport (JFK) and ORD, the only airports in the U.S. which are slot restricted, which complicates new services to and from these airports. Usually, the slots are assigned by the Federal Aviation Administration during an auction process [36]. These Southwest services ended with ATA's bankruptcy in April 2008 [22] [23].

Table 3 gives the airports at which Southwest has operations [2]. While the airline claims to be a point-to-point-carrier [2], some centers of operations can clearly be identified. These, in terms of weekly departures, are the *focus cities* Las Vegas McCarran International Airport (LAS), Chicago Midway International Airport (MDW), Phoenix Sky Harbor International Airport (PHX), Baltimore (BWI), Oakland International Airport (OAK), Houston (HOU), Dallas (DAL), Los Angeles International Airport (LAX), Orlando International Airport (MCO), San Diego (SAN), Nashville International Airport (BNA) and, lastly, Denver International Airport (DEN).

⁵These performance requirements are placed under the umbrella of ETOPS (Extended-range Twin-engine Operation Performance Standards). Under normal conditions, a two-engine aircraft and its engines are designed to be able to fly 60 minutes with only one engine running to reach an airport for an emergency landing. An ETOPS-Certificate can extend this security margin, allowing an airline to widen its range of operation. To operate under ETOPS-conditions, an airline has to obtain an Operational Certificate from the responsible aviation authority. It has to account for special maintenance of the respective aircraft and special training for the crew, both on board and on the ground. For example, a two-engine aircraft operating between mainland U.S. and Hawaii needs an ETOPS-certificate for a one-engine-flight of 180 minutes. [19]



Figure 2: Cities served by Southwest at July 2008 [2]. Service to Minneapolis - St. Paul will begin in March 2009.

Southwest's airports							
City	Code	State	Dep.	Gates	Con.	Since	Reg.
Las Vegas	(LAS)	NV	225	21	53	1982	SW
Chicago-Midway	(MDW)	IL	218	29	47	1985	MW
Phoenix	(PHX)	AZ	207	24	42	1982	SW
Baltimore/Washington	(BWI)	MD	173	26	38	1993	NE
Oakland	(OAK)	CA	142	11	20	1989	SW
Houston-Hobby	(HOU)	TX	141	17	28	1971	SW
Dallas-Love Field	(DAL)	TX	127	14	14	1971	SW
Los Angeles	(LAX)	CA	118	11	19	1982	SW
Orlando	(MCO)	FL	100	12	32	1996	SE
San Diego	(SAN)	CA	92	10	15	1982	SW
Nashville	(BNA)	TN	85	10	28	1986	MW
Sacramento	(SMF)	CA	81	8	11	1991	SW
Denver	(DEN)	CO	78	8	23	2006	SW
Tampa	(TPA)	FL	78	10	29	1996	SE
San Jose	(SJC)	CA	77	7	11	1993	SW
Kansas City	(MCI)	MO	71	8	20	1982	MW
St Louis	(STL)	MO	70	9	22	1985	MW
Philadelphia	(PHL)	PA	65	8	19	2004	NE
Albuquerque	(ABU)	NM	58	6	21	1980	SW
Burbank	(BUR)	CA	58	6	5	1990	SW
Ontario	(ONT)	CA	56	8	6	1985	SW
San Antonio	(SAT)	TX	50	5	13	1971	SW
Austin	(AUS)	TX	48	6	15	1977	SW
Salt Lake City	(SLC)	UT	43	7	14	1994	SW
Fort Lauderdale	(FLL)	FL	43	6	12	1996	SE
Reno	(RNO)	NV	42	4	10	1990	SW

continued on next page

<i>continued from previous page</i>							
City	Code	State	Dep.	Gates	Con.	Since	Reg.
Seattle	(SEA)	WA	40	5	13	1994	NW
Orange County	(SNA)	CA	40	2	6	1994	SW
San Francisco	(SFO)	CA	40	4	6	2007	SW
Portland	(PDX)	OR	39	4	12	1994	NW
New Orleans	(MSY)	LA	38	6	12	1979	SE
El Paso	(ELP)	TX	35	4	11	1977	SW
Providence	(PVD)	RI	34	4	9	1996	NE
Islip	(ISP)	NY	33	8	8	1999	NE
Raleigh-Durham	(RDU)	NC	30	4	8	1999	SE
Manchester	(MHT)	NH	28	4	6	1998	NE
Birmingham	(BHM)	AL	27	4	13	1987	SE
Columbus	(CMH)	OH	25	3	9	1992	MW
Jacksonville	(JAX)	FL	23	3	9	1997	SE
Pittsburgh	(PIT)	PA	23	3	7	2005	NE
Hartford	(BDL)	CT	22	3	7	1999	NE
Detroit	(DTW)	MI	22	4	6	1987	MW
Cleveland	(CLE)	OH	21	4	6	1992	MW
Tulsa	(TUL)	OK	21	4	7	1980	SW
Boise	(BOI)	ID	19	3	7	1994	NW
Louisville	(SDF)	KY	19	3	8	1993	MW
Oklahoma City	(OKC)	OK	19	3	6	1980	SW
Tucson	(TUC)	AZ	21	3	6	1994	SW
Indianapolis	(IND)	IN	16	2	8	1989	MW
Spokane	(GEG)	WA	15	4	6	1994	NW
Buffalo	(BUF)	NY	15	2	6	2000	NE
Omaha	(OMA)	NE	15	2	4	1995	MW
Little Rock	(LIT)	AR	14	3	7	1984	SE
Albany	(ALB)	NY	14	2	5	2000	NE
Midland/Odessa	(MAF)	TX	13	3	6	1977	SW
Lubbock	(LBB)	TX	13	3	5	1977	SW
Washington Dulles	(IAD)	VA	12	2	4	2006	NE
West Palm Beach	(PBI)	FL	12	2	4	2001	SE
Harlingen	(HRL)	TX	12	2	3	1975	SW
Norfolk	(ORF)	VA	11	2	6	2001	NE
Amarillo	(AMA)	TX	11	2	3	1978	SW
Fort Myers	(RSW)	FL	10	2	5	2005	SE
Jackson	(JAN)	MS	9	1	4	1997	SE
Corpus Christi	(CRP)	TX	6	1	1	1977	SW
Southwest's airports at September 2008. 'Dep.' refers to the number of weekly departures, 'Con.' is the number of nonstop connections from each airport, 'Gates' is the number of gates used by Southwest, and 'Reg.' is the airport's geographic region [2].							

Table 3: Southwest's airports at September 2008.

Figure 3 shows the current allocation of the airports served by Southwest according to their geographic location.⁶ The southwest region (SW) is the core area of Southwest's operations with 28 airports being served. The southeast (SE), northeast (NE) and midwest (MW) account for a total of 11 airports respectively, while only 4 airports are served by Southwest in the U.S. northwest (NW). Figure 4 shows how Southwest has expanded in the different regions to reach its current total of 65 destinations, the geographic situation of which is displayed in Figure 3.⁷ During the first ten years of operations (until 1980), Southwest's network was concentrated in the southwest region of the U.S. From the late 1970s through to 1987, airports in Texas' neighbouring southeast states were also added (Louis Armstrong New Orleans International Airport (MSY), Birmingham-Shuttlesworth International Airport (BHM) and also to Little Rock National Airport (LIT)).

Between 1980 and 1995, Southwest expanded into the midwest. Also significant is the extension of its network to the northwest: all of these destinations (Seattle-Tacoma International Airport (SEA), Portland International Airport (PDX), Boise Airport (BOI) and Spokane International Airport (GEG)) were added in 1994. This was essentially due to the merger with Morris Air in 1993. Morris Air was a Salt-Lake-City-based low-fare carrier, and its acquisition added a total of 14 new cities to the Southwest network, and the first of its airports in the northwest [12]. Operations to the northeast began in 1993, with flights

⁶The U.S. states are allocated to regions as follows: **southwest** (SW): CA, NV, AZ, NM, UT, CO, TX, OK, AK, HI, **southeast** (SE): AR, LA, MS, AL, GA, SC, FL, NC, Puerto Rico, U.S. Virgin Islands, **midwest** (MW): ND, SD, NE, KS, MO, IA, MN, WI, MI, IN, IL, OH, KY, TN, **northwest** (NW): OR, WA, ID, MT, WY, and **northeast** (NE): VA, WV, PA, DE, MD, NJ, NY, CT, VT, RI, MA, NH, ME. For a complete list of U.S. airports and their corresponding regions, see Appendix D.

⁷In both Figures 3 and 4 Minneapolis-Saint Paul International Airport (MSP) has already been counted as a Southwest destination, although it will only come into service in March 2009.

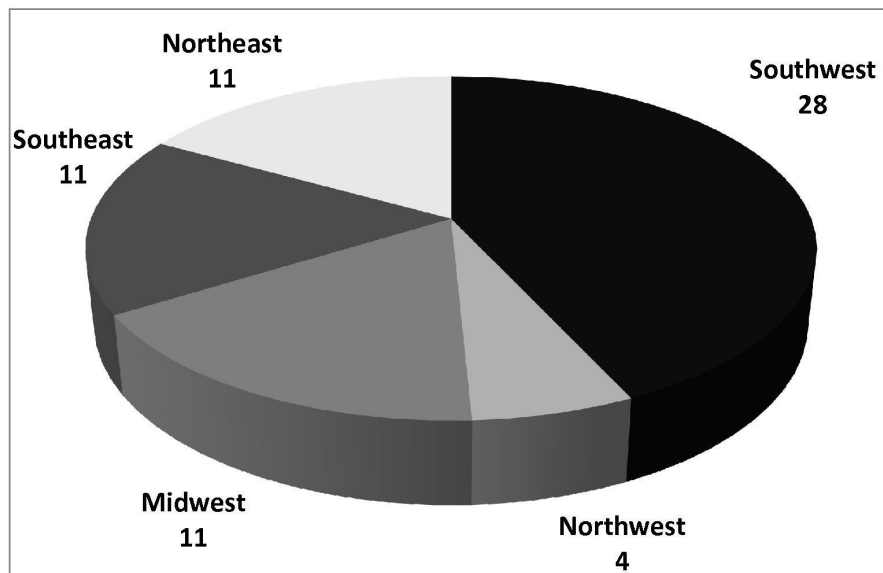


Figure 3: Network diagram. The different coloured areas illustrate the five identified regions of the U.S.: the southwest (SW), northwest (NW), midwest (MW), southeast (SE) and northeast (NE). An allocation of airports to these regions is given in Table 3. The numbers denote the total number of Southwest's airports in the respective area [2].

to Baltimore (BWI), and saw a steady extension, most notably between 1996 and 2000. A second sudden expansion, like that which occurred in the northwest in 1994, took place in the southeast in 1996, with the inauguration of Florida services to Orlando (MCO), Tampa International Airport (TPA) and also to Fort Lauderdale-Hollywood International Airport (FLL). This was followed in 1997, 2001 and 2005, by Jacksonville International Airport (JAX), Palm Beach International Airport (PBI) and Southwest Florida International Airport (RSW).

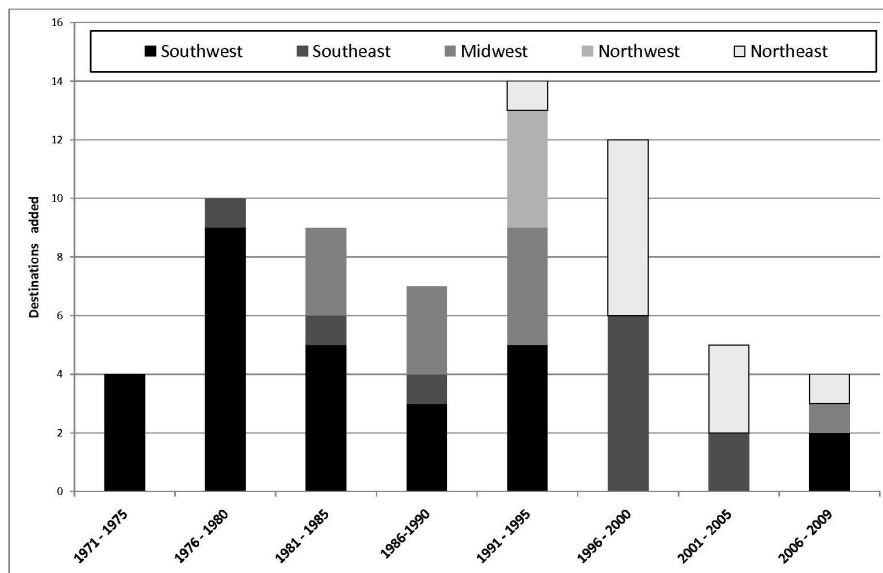


Figure 4: Network development. The columns refer to the given time periods in steps of five years, except for the last period which goes as far as 2009. As in Figure 3, the different colours illustrate the five regions of the U.S. that have been identified [2].

Table 4 gives the number of domestic passengers carried at Southwest’s ten most important airports (defined in terms of the number of connections offered, and weekly departures), with Southwest’s ranking in terms of market share. In eight out of of ten of its most important airports, Southwest has the biggest share in terms of passenger numbers. Regarding the airports served by Southwest (see the Appendix A), there is no clear evidence that Southwest serves only secondary airports. Also (with regards to Table 4), only five of these airports, Midway (MDW), Baltimore (BWI), Oakland (OAK), Dallas Love Field (DAL) and Houston (HOU), can be identified as airports that are close to

Airport		Total PAX (‘000s)	WN PAX (‘000s)	WN share (%) (rank)
Las Vegas	(LAS)	40,000	15,881	39.40 (1)
Chicago	(MDW)	16,909	14,055	83.12 (1)
Phoenix	(PHX)	38,000	11,955	31.17 (2)
Baltimore	(BWI)	20,478	11,133	54.37 (1)
Oakland	(OAK)	12,584	8,374	66.55 (1)
Houston	(HOU)	8,721	7,805	89.49 (1)
Dallas	(DAL)	8,159	7,717	94.58 (1)
Los Angeles	(LAX)	42,000	6,926	16.67 (3)
Orlando	(MCO)	34,000	7,803	23.38 (1)
San Diego	(SAN)	17,997	6,601	36.68 (1)

Table 4: Southwest market share July 2007 - August 2008. [3]

bigger hubs. While Oakland (OAK) is Southwest’s most important airport in the San Francisco area, San Francisco International Airport (SFO) and Norman Y. Mineta San Jos International Airport (SJC) are also served. In the Washington-Baltimore-Area, Southwest offers flights from Baltimore (BWI) and Washington Dulles International Airport (IAD). Las Vegas McCarran International Airport (LAS) and Phoenix Sky Harbor International Airport (PHX) are the only airports in their respective areas where no other airports are offered (Phoenix (PHX) is also a hub of US Airways). Regarding Orlando (MCO) and San Diego (SAN), smaller airports can be found, in close proximity, which are not served by Southwest. Los Angeles (LAX), for instance, is a focus city for United, American and Delta. Moreover, the Los Angeles area offers many other, less congested airports, such as Burbank (BUR), Los Angeles Ontario (ONT) and John Wayne Airport (SNA), which are also served by Southwest.

2 Core data

The main dataset used as the starting point for this project was provided by the U.S. Department of Transportation (DOT). The raw data comprises details on the individual tickets (itineraries) for 10% of all flights on U.S. domestic routes, by quarter, over the 6-year period 2002:1 to 2007:4. The raw data was parsed by Roseline Bilina and Steve Lawford, to give (a) operating carrier, taken from the six ‘major’ carriers listed in the Introduction (American, Continental, Delta, Northwest, United and US), and Southwest, (b) nondirectional nonstop or one-stop route, defined by the endpoint airports, and (c) total number of passengers carried by each carrier on the route. From a separate datafile containing the geographical coordinates (latitude and longitude) of the endpoints, (d) the length of each route (km) was calculated. The database used in the econometric modelling of route presence below consists of 27,966 airport-pairs (one-stop as well as nonstop routes), between a total of 237 airports (a list of the airports is given in Appendix D).

3 Modelling market and route presence

3.1 Motivation and collection of additional data

The analysis of Southwest’s presence on any given route will depend upon multiple factors, and this project reports the attempt to build an explicit econometric model in order to capture some of the more important effects. As a first step, it is necessary to consider route-specific variables that could determine the intrinsic nature of a given route, such as route demand, the composition of the customer base, and geographical characteristics of the route. Secondly, the degree of

competition faced by Southwest on each given route must be assessed, and this will include hub analysis, and the presence of particular types of competitor (low-cost or major, for instance). The econometric modelling step will use these variables to ‘explain’ the likelihood that Southwest will be present on a route that has particular characteristics. Below, and in the following sections, the additional variables that will be used in this project are presented.

In particular, data (in addition to the Core Data detailed above) was collected or constructed on (e) Gross Domestic Product by region (annual data for 2007), (f) population by region (annual data for 2007), (g) U.S. airport by region (time-invariant), (h) U.S. airports that are affected by the 1979 Wright Amendment restricting Southwest’s traffic out of Dallas Love Field (DAL) (generally time-invariant across our sample period, although a minor change led to Southwest serving Dallas DAL to Kansas City International Airport (MCI) and Lambert - St. Louis International Airport (STL) from 2005 onwards), (i) U.S. airports that are slot-restricted (annual data for 2006), (j) the number of years for which Southwest has served each given airport, (k) the average delays at each U.S. airport (annual data covering the period 2002 to 2007, collected from the DOT), and (l) the number of scheduled departures of JetBlue Airways, AirTran Airways, Frontier Airlines and Alaska Airlines, by airport (quarterly data for 2002 to 2007, collected from the T100 database).

From this data, proxies are built for e.g. potential route demand (based on population), for airports in terms of use as hubs by given carriers, and for use as secondary airports, using location data to examine *nearby* airports and/or routes. Conditioning on sectors is used to examine regional behaviour, while average airport delays are used to explore Southwest’s ‘reaction’ to congestion.

3.2 Socio-economic variables

The decision of an airline to serve a given airport (or route) are likely to be influenced by local socio-economic characteristics. Variables based upon regional Gross Domestic Product (GDP) and regional population (Pop.) are introduced into the models, in order to capture the attractiveness of an airport (or route) in terms of major sources of regional revenue (for instance, tourism or business activities, which may be used to determine whether a route can be classified as primarily leisure or business), and the size of the potential customer base.

Data on GDP was obtained from the U.S. Bureau of Economic Analysis (BEA) [42]. The U.S. is divided by region, into *Metropolitan* and *micropolitan Statistical Areas* (MSAs and mSAs), and GDP data is available for the MSAs. Total GDP is used, as are the parts that are made up by the *leisure-sector* and the *business-sector*, which is available for most of the MSAs. An exact definition of the fields of activity within the leisure and business subcategories (by the BEA) is given in Appendix C.⁸ Population data was taken from the U.S. Census Bureau, and is available for both MSAs and mSAs [37]. This division allows for population and GDP figures to be assigned to distinct regional areas, and consequently to the airports that are found within those regions (see Figure 33).

Based on these regionally determined (and hence airport-specific) datasets on GDP and population, the following variables are constructed:

⁸The U.S. Bureau of Economic Analysis (BEA) gives the following definition of the professional and leisure sectors: *business-sector*: The “Professional and Business Supersector” is part of the “Service-providing Industries” and consists of the following sectors: “Professional, Scientific and Technical Services”, “Management of Companies and Enterprises” and “Administrative and Support and Waste Management and Remediation Services; and the *leisure-sector*: The “Leisure and Hospitality Supersector” sector consists of the following sectors: “Arts, entertainment and recreation”, “Accommodation and Food Services”. An exact definition, given by the North American Industry Classification System, containing the activities that are included within these sectors, can be found in Appendix C.

The geometric mean of the population at the departure and arrival airports gives the ‘population’ variable for each route (and is intended to approximate the potential customer demand for the route, since the demand itself is very difficult to define and measure exactly); the square of the population variable is also used, for reasons that will be discussed below:

$$Pop. \text{ and } Pop.^2$$

The region-specific ratio of that component of GDP that is contributed by the *leisure-sector* or the *business-sector*, to the total GDP, is computed for each airport. This creates two dummy (binary) variables for each airport, which take value 1 when the business ratio is greater than 10 percent, or when the leisure ratio is greater than 5 percent, and value 0 otherwise:

$$\frac{GDP_{Leisure}}{GDP_{Total}} > 5\% : Leisure = 1,$$

and

$$\frac{GDP_{Business}}{GDP_{Total}} > 10\% : Business = 1.$$

Presumably, population will have a positive effect on Southwest’s likelihood of serving a route, since a larger population should represent a bigger (potential) local market. The influence of the GDP indicators, that determine whether a route is predominantly professional or leisure, cannot easily be determined in advance of the modelling process. Southwest serves not only typical business destinations such as Midway (MDW) or Philadelphia International Airport (PHL), but also leisure destinations such as Fort Lauderdale (FLL) and Southwest Florida International Airport (RSW) (see Southwest’s route map in Figure 2).

3.3 Geographic (spatial) variables

3.3.1 Regional division

The models also take the geographic location of the airports into account. The surface of the U.S. is divided into five categories, with values ranging from 1 to 5: southwest (1), southeast (2), midwest (3), northwest (4) and northeast (5) (see Table 2 for a classification of Southwest’s airports by region). The use of this variable allows for a more precise description of the traffic flows between the respective regions. The 5 region values are used to create another 15 dummy (binary) variables for the model, corresponding to the 15 possible combinations of the regions, and including intra-regional traffic (see Table 5). For example, *Reg.1* denotes routes within the southwest region, such as Houston (HOU) - Dallas Love Field (DAL), while *Reg.14* stands for routes between the northwest and the northeast, such as Seattle (SEA) - Philadelphia (PHL).

Pairs SW		Pairs SE		Pairs MW		Pairs NW		Pairs NE	
SW-SW	1	SE-SE	6	MW-MW	10	NW-NW	13	NE-NE	15
SW-SE	2	SE-MW	7	MW-NW	11	NW-NE	14		
SW-MW	3	SE-NW	8	MW-NE	12				
SW-NW	4	SE-NE	9						
SW-NE	5								

Table 5: Regional division variable. The 5 geographic regions give rise to 15 combinations for regional routes, including intra-zonal traffic, which are denoted by the variables *Reg.1* to *Reg.15*.

In the models, *Reg.1* to *Reg.15* correspond to traffic ‘corridors’ between and within the five geographical regions. The idea is to see whether Southwest’s expansion strategy can be identified in terms of preferred regions. As shown in

Figures 3 and 4, Southwest's network now covers all five geographic regions of the U.S., and so it is not immediately obvious which traffic corridor(s) will be most important, before the modelling step has been completed.

3.3.2 Distance

The distance between two airports is likely to have an important influence on the decision to serve a route. Southwest's fleet permits operations across the entire U.S. (see Figure 1). However, the aim here is to examine whether there is any tendency for Southwest to operate routes within a certain range. For example, Southwest may only enter routes which are sufficiently long, so that they do not directly compete with car travel. Moreover, there is perhaps also a certain upper limit on route length, above which Southwest will not wish to enter a route. The following route-distance-related variables are introduced:

$$Dist. \text{ and } Dist.^2$$

and

$$Dist. \leq 200.$$

The quadratic term $Dist^2$ allows for measurement of a 'saturation limit' for route length. That is, it is expected that the influence of distance on the probability of route presence is positive, up to a certain level, after which an increase in distance may start to reduce the probability. The dummy variable, taking value 1 (0) when distance is no more than (above) 200km is expected to be negative due to the avoidance of competition with car travel that was mentioned above. However, this variable has an additional purpose, which is to 'force' the model to 'recognize' routes that are technically possible, such as San Francisco (SFO) - Oakland (OAK), but that do not have any practical meaning (in this

case, the airports are separated by only a few kilometers). The data on distance between airports is computed directly as the great circle distance based upon the (difference between the) latitude and longitudes at the airport endpoints.

3.4 Airport-specific variables

3.4.1 Political and legal / regulatory issues

So far, a number of geographic, technical and economic constraints on Southwest's route choices have been considered: network expansion may follow a certain geographic structure; technical issues such as requirement of ETOPS-certification may preclude air transport service in certain regions (see Section 1.1); also, the nature of the local economy may affect a carrier's decision to serve a route. However, political and legal constraints may also impose strong constraints on network development.

When considering the Dallas airports, the *Wright Amendment* must be taken into account. In 1974, a new international airport was built in the Dallas - Fort Worth region: DFW. To protect this investment, the Wright Amendment restricted the competition the new airport could face from Dallas Love Field (DAL). Flights from DAL were restricted to those states that directly neighbour Texas (namely, Louisiana, Arkansas, Oklahoma and New Mexico). Flights to other states were only allowed on planes with a maximum capacity of 56 seats. In 1997, the law was weakened to allow services to Alabama, Kansas and Mississippi, and Missouri followed in 2005. A further change was made in 2006, enabling one-stop flights to airports outside of the Wright-zone. Only Southwest and American currently operate out of DAL. The Wright Amendment will be repealed in 2014, from which time domestic nonstop flights will be possible throughout the U.S.

At the same time, DAL’s capacity will be reduced from 32 to 20 gates [24] [25].

Additionally, *slot restrictions* must be taken into account. High levels of congestion results in slots being auctioned at some airports, as mentioned in Section 1.1. The airports in our dataset that are affected by slot restrictions are Washington National (DCA), John F Kennedy (JFK), Chicago O’Hare (ORD) and LaGuardia (LGA).

Introducing these variables into the econometric models is not straightforward, due to a particular technical relationship between the Wright Amendment and slot restriction variables with the variable that captures Southwest’s presence on a route. This will be discussed further in Section 5. Even if these variables cannot be formally introduced into the models though, they can nevertheless be studied. For example, the model could predict a high likelihood of Southwest’s presence on a route such as Dallas Love Field (DAL) - San Francisco (SFO), or on routes out of LaGuardia (LGA). Since these routes *cannot* currently be served, either due to the Wright Amendment or because of slot restrictions, while the model does not have this information in the form of an explanatory variable, the results can then be interpreted in terms of the probable effect of a *removal* of the restriction(s).

3.4.2 Hubs and focus airports

Some key airports can be identified in the Southwest network, whether in terms of weekly departures or local market share (see Tables 3 and 4). To accommodate this, a Southwest ‘hub’ variable is constructed. For each airport and for each quarter in the core data set, the ratio of Southwest passengers to the total number of passengers at the airport is computed. A ranking of these ratios enables

construction of a variable measuring the degree of importance of each airport within the Southwest network. Four groups of ratios are retained, which result in four mutually exclusive Southwest hub dummy variables, each taking value 1 when the airport is a hub of that level:

$$\frac{PAX_{\text{Southwest}}}{PAX_{\text{Total}}} < 30\% : \quad SW_Hub_1 = 1,$$

$$30\% \leq \frac{PAX_{\text{Southwest}}}{PAX_{\text{Total}}} < 50\% : \quad SW_Hub_2 = 1,$$

$$50\% \leq \frac{PAX_{\text{Southwest}}}{PAX_{\text{Total}}} < 70\% : \quad SW_Hub_3 = 1,$$

$$\frac{PAX_{\text{Southwest}}}{PAX_{\text{Total}}} \geq 70\% : \quad SW_Hub_4 = 1.$$

For each airport pair in the sample, the two endpoint hub dummy variables are compared, and the higher value is chosen to determine the ‘hub nature’ of the route itself. For example, the route Houston (HOU) - San Francisco (SFO) would be a *SW_Hub_4* route, due to the fact that HOU is a *SW_Hub_4* airport and SFO is a *SW_Hub_1* airport for Southwest. It could be expected that, even if Southwest is not a typical hub-and-spoke airline, the influence of the *SW_Hub_x* variable would have a bigger influence on Southwest’s route presence as *x* increases.

The same ranking of the significance of each airport is performed for each of the six majors: AA, CO, DL, NW, UA and US, which gives another 24 (6 carriers \times 4 variables) hub dummy variables. For instance, for US, this is based

upon the ratio $\frac{PAX_{US\ Airways}}{PAX_{Total}}$, and gives the hub variables *US_Hub_1* to *_4*.

In an econometric model with a constant term (as is usually the case), a ‘full set’ of mutually exclusive dummies cannot be included, since a linear combination of the variables will be perfectly correlated with the constant (the problem of perfect multicollinearity), which means that the estimation cannot be performed. For this reason, the variables corresponding to percentages below 30%, such as *SW_Hub_1*, cannot be included in the model, if the other hub variables i.e. *SW_Hub_2*, *SW_Hub_3* and *SW_Hub_4* are included. Further, the variables for the six majors (but not for Southwest) are aggregated respectively: for example, the US hub variables *US_Hub_2*, *US_Hub_3* and *US_Hub_4* are used to build *US_Hub*, giving one variable which designates an airport to be a hub of US when the passenger percentage is above 30%. More explanation of the mathematical impact of the dummy variables is given in Section 5.

The influence of these variables on Southwest’s behaviour cannot easily be predicted. The model results will permit examination of whether Southwest avoids direct competition with a major carrier in one of its hub airports (or vice-versa), or not; and the impact on Southwest’s entry strategy of its own hubs and focus airports. For example, Southwest is not present in Delta’s Atlanta (ATL) hub, but entered US’s hub in Philadelphia (PHL) in 2004 (see Tables 3 and 33.)

3.4.3 Secondary airports

As described in Section 1.1, an important feature of many low-cost carriers has been operations out of secondary airports. These airports allow access to the markets of large metropolitan areas (such as Chicago for Midway (MDW)), without some of the negative impacts like congestion (affecting Chicago O’Hare

(ORD), for instance). Southwest currently serves both types of airports: it only serves the secondary airport Fort Lauderdale (FLL) in the Miami Area (and not American Airline's Miami International (MIA) hub). In contrast, Southwest's services in the San Francisco Bay Area go to San Francisco International (SFO) (a hub airport for UA: see Table 33), as well as to Oakland (OAK) and San Jose International (SJC), which are considered to be secondary airports for SFO.

An airport is determined to be secondary if it is in close proximity to a major's hub airport (and not for Southwest), as defined in Section 3.4.2. For example, MDW is a secondary airport for ORD, and is located close to ORD, which is a hub airport for AA (and also for UA). Thus, the variable *AA_Hub* will take the value 1 for each ORD city-pair. Midway (MDW) is not even served by AA (or by any other major), and so *AA_Hub* will take the value 0. This is a simple example of a hub airport which has a secondary airport in a neighbourhood of less than 50km.

The proximity is defined in accordance with two distances, creating dummy variables for two cases, depending upon whether a hub airport is within a range circle of 50km or 100km around an airport (calculated using coordinates):

$$\text{Distance to hub airport} < 50\text{km}: \quad \text{Hub_0_50} = 1,$$

$$\text{Distance to hub airport} \geq 50\text{km and } \leq 100\text{km}: \quad \text{Hub_50_100} = 1.$$

Again, Southwest's current route network does not permit clear conclusions to be drawn in advance of modelling. Southwest is a low-cost carrier, but its airport map (see Figure 2) shows that it does not operate only out of secondary airports.

3.4.4 Airport congestion

The choice of secondary airports is, among other things, strongly affected by the problem of airport congestion. One of the key aims of low-cost companies is to maintain short turn-around times for their airplanes, to maximize the utilization of their fleet. The effects of congestion on airports, such as delayed arrivals and departures, compromise efficient fleet usage. It is checked whether the maximum average departure or arrival delay (across the endpoints) of a route is important factor in determining Southwest's route presence. Data on average airline on-time-performance was taken from the U.S. Bureau of Transport Statistics (BTS). The data reports the ratio of the number of departures and arrivals which were delayed by more than 15 minutes at a given airport, to the respective total of departures and arrivals at that airport, for each of the years 2002 to 2007. Two variables are derived:

$$Max_DEP_Delay = \frac{DEP_{Delay > 15}}{DEP_{Total}},$$

$$Max_ARR_Delay = \frac{ARR_{Delay > 15}}{ARR_{Total}}.$$

Presumably, these variables will have a strong negative effect on Southwest's route presence. Nevertheless, a definite answer cannot yet be given. For example, Southwest is currently looking to enter New York's LaGuardia (LGA) airport, which is heavily affected by congestion [36].

3.4.5 Low-cost and other competition

The model already contains two variables that aim to measure the competition that Southwest faces from the six major carriers: (1) the core data includes the number n of carriers (i.e. the number of majors and Southwest) present on a route. To use this variable, Southwest has to be removed from the number of carriers n , creating the number of *competitors*. Then, the effect of the correlation between the number of competitors and population must be investigated (since the number of airlines serving a certain route might be strongly influenced by population, and so by the size of the markets of the two endpoint cities). These considerations are discussed further in Section 5, but for a first analysis, the number of competitors gives a measure of the degree of competition on each route; (2) the passenger-hub dummy variables defined in Section 3.4.2 measure the importance of an airport for each of the major carriers (or Southwest) in terms of passenger percentages, and give an additional measure of the level of competition that Southwest will face at each airport.

Two additional competition factors that affect Southwest are identified: (3) the influence of competition from low-cost carriers in the U.S. domestic market has to be taken into account: AirTran Airways (FL), Frontier Airlines (F9) and JetBlue Airways (B6) are considered (and are not assumed here to be major carriers) and as seen in Table 2, each of these three low-cost companies accounts for a reasonable number of passengers; (4) lastly, Alaska Airlines (AS) is incorporated in the construction of variables to measure competition.⁹ Alaska Airlines does not operate intercontinental services, which distinguishes it from the six majors, but neither does it follow a low-cost carrier business model,

⁹Following Ito and Lee (2003), “Low cost carrier growth in the U.S. airline industry: past, present and future”, Alaska Airlines is not considered to be a low-cost carrier [13].

which marks a difference between Alaska and the other low-cost carriers. Alaska is included in the models as a further competition factor for Southwest.

The influence of the four carriers Jetblue, Frontier, Airtran and Alaska is included through dummy variables, which take value 1 at each of the sample airports at which one of the respective carriers is present.¹⁰

4 Discrete choice models

4.1 Model motivation

The model of Southwest's route presence has to satisfy one major requirement: the outcome, while potentially being driven by discrete factors (such as the number of competitors on the route, or the airport category), and continuous factors (such as the population and GDP associated with the route's endpoints), is a scalar binary variable, i.e. it will take the value $y = 1$ if Southwest is present on any given route, and $y = 0$ if Southwest is absent from the route. A probability model is designed that will capture the likelihood that Southwest will be present on any particular route, as a function of relevant explanatory variables:

$$\text{Prob}(\text{Southwest present on a route}) := \text{Prob}(y = 1) = F(\text{relevant parameters}), \quad (1)$$

where $F()$ is some function. It is assumed that x is a $k \times 1$ vector of explanatory variables (e.g. number of competitors on a route, GDP, etc.), and β is a $k \times 1$ vector of unknown weights, that are to be estimated. Then, (1) can be rewritten

¹⁰An overview of the routes served by these carriers is given in the figures of Appendix B.

as:

$$\text{Prob}(y = 1) = F(x, \beta), \quad \text{Prob}(y = 0) = 1 - F(x, \beta), \quad (2)$$

using the fact that $y = 0$ or $y = 1$.

4.2 A first attempt: the linear probability model

The simplest approach to specifying the right-hand-side of equation (2) is to use a linear function

$$F(x, \beta) = x' \beta, \quad (3)$$

where it can be seen that the linear probability model has an underlying linear regression, since the conditional expectation $E(y|x) = F(x, \beta)$, whereupon

$$y = E(y|x) + (y - E(y|x)) := x' \beta + u, \quad (4)$$

where u is a residual *error* term. Several difficulties arise from the linear regression model when $y = 0$ or $y = 1$ are the only possible realizations of the variable of interest.

The **first problem** comes from the residuals. Due to the fact that y is a binary variable, the residuals can only take two values, as follows:

$$u \in \{-x' \beta, 1 - x' \beta\}, \quad (5)$$

where $u = 1 - x' \beta$ as $y = 1$, with probability $\text{Prob}(y = 1) = x' \beta$, and $u = -x' \beta$ as $y = 0$, with probability $\text{Prob}(y = 0) = 1 - x' \beta$. The conditional expectation

of u is calculated in (6) as:

$$\begin{aligned} E(u|x) &= (-x'\beta)\text{Prob}(y=0) + (1-x'\beta)\text{Prob}(y=1) \\ &= (-x'\beta)(1-x'\beta) + (1-x'\beta)(x'\beta) = 0, \end{aligned} \quad (6)$$

from which the conditional variance of u is:

$$\begin{aligned} \text{Var}(u|x) &:= E(u^2) = (-x'\beta)^2\text{Prob}(y=0) + (1-x'\beta)^2\text{Prob}(y=1) \\ &= (-x'\beta)^2(1-x'\beta) + (1-x'\beta)^2(x'\beta) = (x'\beta)(1-x'\beta). \end{aligned} \quad (7)$$

Equation (7) clearly shows that the variance is not constant, but depends upon x , which implies heteroscedasticity. The **second problem** is rather more serious: essentially, there is no guarantee that the estimated probabilities $\widehat{\text{Prob}}(y=1)$ and $\widehat{\text{Prob}}(y=0)$ will be between 0 and 1, and furthermore, the estimated variances from equation (7) will not necessarily be positive.

4.3 Particular models: the probit and the logit

Two probability models are now introduced which are more appropriate, since they impose the requirement that the predicted probabilities lie between 0 and 1. The function $F()$ is chosen to be a statistical cumulative distribution function (cdf). These functions are well-understood, and map any real number to the

interval $[0, 1]$, and so:

$$\lim_{x'\beta \rightarrow \infty} F(x'\beta) = \lim_{x'\beta \rightarrow \infty} \text{Prob}(y = 1) = 1,$$

$$\lim_{x'\beta \rightarrow -\infty} F(x'\beta) = \lim_{x'\beta \rightarrow -\infty} \text{Prob}(y = 1) = 0.$$

In theory, any valid cdf could be used, but for practical reasons either the normal distribution or the logistic distribution is chosen. The normal distribution leads to the *probit model*, where ϕ is the probability density function (pdf) of the standard normal $N(0, 1)$, and Φ is the cdf:

$$\text{Prob}(y = 1) = \int_{-\infty}^{x'\beta} \phi(t)dt = F(x'\beta) = \Phi(x'\beta). \quad (8)$$

Due to its mathematical convenience, the logistic distribution Λ is also used, which leads to the *logit model*:

$$\text{Prob}(y = 1) = \frac{e^{x'\beta}}{1 + e^{x'\beta}} = F(x'\beta) = \Lambda(x'\beta). \quad (9)$$

The shape of these two distributions (both the cdfs and the pdfs) are relatively similar, although the logistic distribution has heavier tails. The two models often give similar fits to the data [43]. Regardless of which distribution is used, it is important to note that, as well as the coefficients in the β vector, the marginal effects must also be taken into account. The general form of the derivative of the cdf $F()$ gives the following for the pdf $f()$:

$$\frac{\partial \text{Prob}(y = 1)}{\partial x} := \frac{\partial F(x'\beta)}{\partial x} = \frac{\partial F(x'\beta)}{\partial x'\beta} \frac{\partial x'\beta}{\partial x} = f(x'\beta)\beta. \quad (10)$$

For the probit and logit models, (10) gives the following:

$$\frac{\partial \text{Prob}(y = 1)}{\partial x} = \Phi(x'\beta)\beta, \quad \frac{\partial \text{Prob}(y = 1)}{\partial x} = \Lambda(x'\beta)[1 - \Lambda(x'\beta)]\beta, \quad (11)$$

which depends upon both β and x (unlike standard linear regression).

4.4 Further model interpretation (the logit)

The following numerical example illustrates a way to interpret the results from a logit model. The odds-ratio gives the ratio of the probabilities of events $y = 1$ and $y = 0$. For the logit model (9), this ratio may be expressed as follows:

$$\frac{\text{Prob}(y = 1)}{\text{Prob}(y = 0)} = \frac{\frac{e^{x'\beta}}{1 + e^{x'\beta}}}{1 - \frac{e^{x'\beta}}{1 + e^{x'\beta}}} = e^{x'\beta}, \quad (12)$$

where the derivative with respect to the j th explanatory variable is:

$$\frac{\delta \left(\frac{\text{Prob}(y=1)}{\text{Prob}(y=0)} \right)}{\delta x_j} = e^{x'\beta} e^{\beta_j}. \quad (13)$$

Now assume that x_j increases by one unit, and that the estimated value $\hat{\beta}_j = 0.1$. Then, the result of a changed x_j can be seen using (13). The multiplicative factor is $e^{0.1} \approx 1.105$. Thus, an increase in the explanatory variable x_j by one unit raises the probability of the event $y = 1$, relative to the $y = 0$, by about 10.5%.

4.5 Model estimation

The unknown vector of weights β is estimated by maximum likelihood. This chooses β to maximize the probability of observing $\{y_i\}_{i=1}^N$ (*this variable indicates Southwest's presence or absence from any given route, from a total of N routes*),

given a set of data in $\{x_i\}_{i=1}^N$ (*these are the corresponding vectors of explanatory variables*). The *likelihood function* is the product of the pdfs of the individual observations y_i , which each follow a Bernoulli distribution:

$$f(y_i) = p^{y_i} (1 - p)^{1-y_i} \quad y_i \in \{0, 1\} \quad p \in [0, 1], \quad (14)$$

where $y = 1$ with $\text{Prob}(y = 1) = F(x'\beta) := p$ and $y = 0$ with $\text{Prob}(y = 0) = 1 - F(x'\beta) = 1 - p$. The log-likelihood (taken for mathematical convenience) is then given by:

$$\ln L(\beta) = \ln \prod_{i=1}^N f(y_i) = \sum_{i=1}^N \ln f(y_i) = \sum_{i=1}^N \{y_i \ln F(x'\beta) + (1 - y_i) \ln [1 - F(x'\beta)]\}, \quad (15)$$

with $f(y_i) \sim y_i, p_i$ and $p_i \sim x_i'\beta$. The *maximum likelihood estimator* solves:

$$\hat{\beta}_{\text{ML}} = \arg \max_{\beta} \ln L(\beta),$$

which has first-order conditions

$$\begin{aligned} \frac{\partial \ln L(\beta)}{\partial \beta} &= \sum_{i=1}^N \frac{y - i}{F_i(x'\beta)} f_i(x'\beta) x_i + (1 - y_i) \frac{1}{1 - F(x'\beta)} (-f(x_i'\beta) x_i) \\ &= \sum_{i=1}^N \frac{f_i(y_i - F_i)}{F_i(1 - F_i)} x_i = 0. \end{aligned} \quad (16)$$

For the probit and logit models, (16) reduces to:

$$\frac{\partial \ln L(\beta)}{\partial \beta} = \sum_{i=1}^N (y_i - \Lambda_i) x_i \quad (\text{logit}), \quad \frac{\partial \ln L(\beta)}{\partial \beta} = \sum_{i=1}^N \frac{\phi_i(y_i - \Phi_i)}{\Phi_i(1 - \Phi_i)} x_i \quad (\text{probit}). \quad (17)$$

Further, the *Hessian matrix* H of second derivatives of (16) with respect to β has

to be negative definite (the second-order condition for the maximization). The estimated Hessian \hat{H} is used to build an estimated variance-covariance matrix \widehat{AVar} , whose elements are used for hypothesis testing:

$$\widehat{AVar}(\hat{\beta}_{ML}) = \hat{H}^{-1} B \hat{H}^{-1},$$

with

$$\hat{H} := \frac{\partial \ln(L)}{\partial \beta \partial \beta'} \Big|_{\beta = \hat{\beta}_{ML}}, \quad B =: \sum_{i=1}^N g_i^2 x_i x_i', \quad (18)$$

where the g_i 's in (18) are the terms multiplying x_i in the foc of the probit and the logit models respectively, specified in (17).

4.6 Model inference

The *t tests* with null hypothesis $H_0 : \beta_j = 0$ are assessed for the *significance* of the estimated coefficients $\hat{\beta}$ (these are used to check whether the *j*th explanatory variable x_j has an influence on the probability of Southwest being present on a route ($y = 1$)). To go one step further regarding the chosen model, the *likelihood ratio test* can also be performed, and examines whether the explanatory variables jointly drive the probability of Southwest being present on a route (in other words, whether the model is ‘useful’ at all). The null hypothesis is $H_0 : \beta_1 = \beta_2 = \dots = \beta_k = 0$, and the test statistic follows a (large sample) chi-squared distribution. The ‘information value’ (or goodness-of-fit) of the models is examined by computing *McFadden’s* R^2 . Similar to conventional R^2 , this should be as far as possible away from zero. However, qualitative interpretation is not easy, and multiple models must be compared, in order to deliver a reliable interpretation of model quality.

5 Model development

In this section, the construction of the model is described step-by-step. The objective is to see how well each variable, or change in model specification, changes the quality of the regression results, and to explain the mathematical impacts of introducing each of the variables in the model (as introduced in Sections 3.1 to 3.4.5). The discussion of Models 1 to 10 below is based on the data from the final quarter of the sample, 2007:4. Once the best model has been specified, it is applied to the full sample period of 24 quarters (i.e. 2002:1 through to 2007:4). Furthermore, regression results following slight modifications of the best model are presented, as a check on the robustness of some of the assumptions that were made when constructing the variables themselves.

5.1 *Model 1: Distance and population*

The first model takes only two variables into account: the Distance *Dist.*, and the population *Pop.*. For each airport-pair, the geometric mean of the population is computed, giving one population value for each route. The scaling of the variables is intended to give estimated coefficients of a similar magnitude, which aids the numerical stability of the estimation procedure. Specifically, population is divided by 1,000,000, and distance by 1,000. Throughout the paper, the scaling numbers are omitted when denoting the variables, and it must be kept in mind that they have been added to the regressions.

$$\text{Prob}(y = 1) = F \left(\beta_0 + \beta_1 \frac{Dist.}{1000} + \beta_2 \frac{Pop.}{1000000} \right).$$

	Coeff.	SE	Prob.	MEΦ	MEΛ	OR	END
Dist.	-0.0280	0.0066	0.0000	-0.0153	-0.0069	1.1656	0.0002
Pop.	0.1906	0.0078	0.0000	0.1042	0.0472	1.7134	0.0641
R ²	0.080						
OR	1.2259						

Table 6: Results from Model 1.

The result of this first regression gives an R² of only 8%. This shows that modelling Southwest's route presence only by accounting for distance and population is not sufficient, and falls far short in covering all of the relevant effects.

5.2 *Model 2: Carriers and competitors*

In this second step, data on competition is introduced, i.e. whether there are any other airlines operating on the same route. The variable *Carrier* is used, which gives the total number of airlines operating on the route, including Southwest. In the modelling itself, attention has to be given to the fact that to study the number of competitors, one has to subtract 1 from *Carrier* when Southwest operates. For example, if *Carrier* is equal to 6, and Southwest has operations on the route, then there are only 5 competitors. This has some effect on the way the data is treated.

$$\text{Prob}(y = 1) = F(\beta_0 + \beta_1 \text{Dist.} + \beta_2 \text{Pop.} + \beta_3 \text{Comp.})$$

The mathematical impact of considering one kind of competition (presence on a route), imposed on Southwest by the major carriers, is given in Table 7.

	Coeff.	SE	Prob.	ME Φ	ME Λ	ME OR	RCI
Dist.	-0.1335	0.0098	0.0000	-0.0667	-0.0334	0.8736	0%
Pop.	0.0832	0.0078	0.0000	0.0416	0.0208	1.0850	6%
Carrier	0,4470	0,0106	0,0000	0,2232	0,1118	1,5610	17%
R ²	25%						
OR	0.9983						

Table 7: Results from Model 2 without distinguishing between carrier and competitor.

It can be seen that for the model results presented in Table 7, the R^2 increases considerably from 8% in Model 1 to 25%. This is caused by endogeneity: when a model of Southwest's presence is estimated by accounting for number of carriers, we implicitly model Southwest's presence (on the left hand side of the equation as a dependent variable), as a function of Southwest's presence (as being part of the number of carriers serving a route). This 'reverse causality' can be investigated by a regression of the explanatory variable, here *Carrier*, on the dependent variable, here Southwest's presence on a route $Prob(y = 1)$. The R^2 of this regression is named the *Reverse Causality Index* (RCI). If endogeneity occurs, then RCI is relatively high, meaning that *Carrier* as an independent variable has a reasonable influence on the dependent variable $Prob(y = 1)$. Table 7 gives an RCI of 17% for the *Carrier* variable. When Southwest's presence is modelled by a variable which is strongly linked to Southwest's presence, the R^2 for the model increases considerably, but has low explanatory power.

Therefore, a variable capturing competition on a route has to be built, which is independent of Southwest: the number of competitors (for Southwest) is used, rather than the number of carriers on route. The results are given in Table 8. The influence of competitors on a route has an influence on Southwest's route

choice, as expected. However, the increase of R^2 of about 3% compared to Model 1 is more reasonable and the RCI only gives 4%.

	Coeff.	SE	Prob.	ME Φ	MEA	ME OR	RCI
Dist.	-0.0586	0.0071	0.0000	-0.0314	-0.0146	1.0344	0%
Pop.	0.1552	0.0078	0.0000	0.0833	0.0387	1.2809	6%
Comp.	0.1792	0.0110	0.0000	0.0962	0.0447	1.3120	4%
R^2	11%						
OR	1.0968						

Table 8: Results from Model 2 with the number of carriers replaced by the number of competitors.

5.3 Model 3: Competitors and quadratic terms

In this part, the quadratic terms are added to allow for some nonlinearity within the model. Additionally, it must be investigated whether there is correlation between number of competitors, derived in Model 2, and the population.

$$\text{Prob}(y = 1) = F(\beta_0 + \beta_1 \text{Dist.} + \beta_2 \text{Dist.}^2 + \beta_3 \text{Pop.} + \beta_4 \text{Pop.}^2 + \beta_5 \text{Comp.}^* + \beta_6 \text{Comp.}^{*2})$$

The covariance data matrix in Table 9 gives a value of 38% for the correlation between population and the number of competitors. This correlation is created due to the fact that a route pair combining two cities with a high population should create a big market for airlines. The large size of the market may attract many airlines, which means that there is correlation: two of the variables which influence Southwest's route presence, population and number of competitors, are linked and therefore contain to a certain extent the same information.

An auxiliary regression of number of competitors on population eliminates the effect of correlation, resulting in a reduced correlation coefficient of less than

	Dist.	Pop.	Comp.
Dist.	1.0000		
Pop.	0.0010	1.0000	
Comp.	0.1284	0.3866	1.0000

Table 9: The covariance data matrix for Model 3 before the first auxiliary regression.

	Dist.	Pop.	Comp.
Dist.	1.0000		
Pop.	0.1388	1.0000	
Comp.*	0.0000	0.0010	1.0000

Table 10: The covariance data matrix for Model 3 after the first auxiliary regression.

1%, as shown in Table 10.

Due to the introduction of quadratic terms for population and number of competitors, correlation must also be controlled for in this case. As given in Table 12, correlation occurs a second time for the quadratic number of competitors and both population and the squared value of population. With the results of the second auxiliary regression given in Table 13, regression for Model 4 are given in Table 14.

	Coeff.	SE	Prob.	ME Φ	MEA	ME OR	RCI
Dist.	-0.0586	0.0071	0.0000	-0.0308	-0.0146	0.9925	0%
Pop.	0.1998	0.0071	0.0000	0.1052	0.0498	1.5552	6%
Comp.* ²	0.1792	0.0110	0.0000	0.0943	0.0447	1.5243	1%
R ²	11%						
OR	1.1048						

Table 11: Results for Model 3 without quadratic terms.

	Dist.	Dist. ²	Pop.	Pop. ²	Comp. [*]	Comp. ^{*2}
Dist.	1.0000					
Dist. ²	0.9364	1.000000				
Pop.	0.001008	-0.0207	1.0000			
Pop. ²	0.0096	0.0008	0.8610	1.0000		
Comp. [*]	0.1388	0.1093	5.86E-16	-0.1314	1.0000	
Comp. ^{*2}	0.1321	0.1100	0.4354	0.3544	0.6985	1.0000

Table 12: Covariance data matrix for Model 3 with quadratic terms, before second auxiliary regression.

	Dist.	Dist. ²	Pop.	Pop. ²	Comp. [*]	Comp. ^{*2}
Dist.	1.0000					
Dist. ²	0.9364	1.0000				
Pop.	0.0011	-0.0207	1.0000			
Pop. ²	0.0096	0.0008	0.8610	1.0000		
Comp. [*]	0.1388	0.1093	5.86E-16	-0.1314	1.0000	
Comp. ^{*2}	0.1472	0.1340	1.96E-16	4.42E-16	0.7651	1.0000

Table 13: Covariance data matrix for Model 3 with quadratic terms, after second auxiliary regression.

	Coeff.	SE	Prob.	ME Φ	MEA	ME OR	RCI
Dist.	0.4707	0.0397	0.0000	0.3591	0.0715	11.4395	0%
Dist ²	-0.1057	0.0086	0.0000	-0.0806	-0.0161	3.5165	0%
Pop.	0.6338	0.0401	0.0000	0.4836	0.0963	16.5271	6%
Pop. ²	-0.0463	0.0058	0.0000	-0.0353	-0.0070	3.8834	1%
Comp. [*]	0.1144	0.0215	0.0000	0.0873	0.0174	5.4711	1%
Comp. ^{*2}	-0.0170	0.0078	0.0293	-0.0130	-0.0026	4.1287	0%
R ²	20%						
OR	4.3509						

Table 14: Results for Model 3 without quadratic terms, after second auxiliary regression.

5.4 *Model 4: Commercial sectors*

In the next step, the influence of regional economic factors is taken into account, introducing the the dummy variables of *Leisure* and *Business* as the ratios of sector-specific GDP over total GDP.

$$\begin{aligned}
 \text{Prob}(y = 1) = & F(\beta_0 + \beta_1 \text{Dist.} + \beta_2 \text{Dist.}^2 \\
 & + \beta_3 \text{Pop.} + \beta_4 \text{Pop}^2 + \beta_5 \text{Comp}^* + \beta_6 \text{Comp.}^{*2} \\
 & + \beta_7 \text{Leisure} + \beta_8 \text{Business})
 \end{aligned}$$

5.5 *Model 5: Traffic corridors*

The introduction of traffic corridors should give additional explanatory power to the model by allocating airports to certain regions. This assignment creates five traffic areas, and their connections create 15 traffic corridors. These new

[illegible]

Table 15: Results of Model 4.

variables account for an increase in R^2 of 6%.

$$\begin{aligned} \text{Prob}(y = 1) = & F(\beta_0 + \beta_1 Dist. + \beta_2 Dist^2 \\ & + \beta_3 Pop. + \beta_4 Pop^2 + \beta_5 Comp.^* + \beta_6 Comp.^2 \\ & + \beta_7 Leisure + \beta_8 Business + \sum_{i=1}^{14} \beta_{i+8} Region_i) \end{aligned}$$

	Coeff.	SE	Prob.	MEΦ	MEA	ME OR	RCI
Dist.	0.3036	0.0507	0.0000	0.2648	0.0244	19.5819	0%
Dist. ²	-0.0931	0.0102	0.0000	-0.0812	-0.0075	8.5710	0%
Pop.	0.5282	0.0464	0.0000	0.4607	0.0425	32.4778	6%
Pop. ²	-0.0411	0.0062	0.0000	-0.0358	-0.0033	9.3095	1%
Comp. [*]	0.0992	0.0232	0.0000	0.0865	0.0080	12.7065	1%
Comp. ^{*2}	-0.0261	0.0083	0.0016	-0.0228	-0.0021	9.6182	0%
Leisure	0.0785	0.0380	0.0388	0.0684	0.0063	12.1469	0%
Business	0.5962	0.0395	0.0000	0.5200	0.0480	30.8189	9%
Region1	1.6448	0.1270	0.0000	1.4346	0.1324	257.4133	3%
Region2	1.1629	0.1302	0.0000	1.0143	0.0936	101.0177	0%
Region3	0.9376	0.1304	0.0000	0.8178	0.0755	67.4843	0%
Region4	1.2650	0.1346	0.0000	1.1033	0.1018	137.2653	0%
Region5	1.1135	0.1328	0.0000	0.9711	0.0896	91.6474	0%
Region6	0.5509	0.1354	0.0000	0.4805	0.0443	29.1803	0%
Region7	0.5440	0.1314	0.0000	0.4744	0.0438	30.2760	0%

continued on next page

<i>continued from previous page</i>							
	Coeff.	SE	Prob.	MEΦ	MEΛ	ME OR	RCI
Region8	1.0736	0.1491	0.0000	0.9363	0.0864	90.5901	0%
Region9	0.4641	0.1297	0.0003	0.4048	0.0373	25.2268	0%
Region10	0.3733	0.1424	0.0087	0.3255	0.0300	21.8380	0%
Region11	0.8197	0.1470	0.0000	0.7150	0.0660	56.9782	0%
Region12	0.3253	0.1323	0.0139	0.2837	0.0262	19.5147	0%
Region13	0.9529	0.2276	0.0000	0.8311	0.0767	74.6156	0%
Region14	0.9720	0.1527	0.0000	0.8478	0.0782	75.2348	0%
R ²	28%						
OR	10.3288						

Table 16: Results of Model 5.

5.6 Model 6: Significance of airports

$$\begin{aligned}
\text{Prob}(y = 1) = & F(\beta_0 + \beta_1 \text{Dist.} + \beta_2 \text{Dist.}^2 \\
& + \beta_3 \text{Pop.} + \beta_4 \text{Pop.}^2 + \beta_5 \text{Comp.}^* + \beta_6 \text{Comp.}^{*2} \\
& + \beta_7 \text{Leisure} + \beta_8 \text{Business} + \sum_{i=1}^{14} \beta_{i+8} \text{Region}_i \\
& + \sum_{i=2}^4 \beta_{21+i} \text{SW_Hub}_i + \beta_{26} \text{AA_Hub} + \beta_{27} \text{US_Hub} \\
& + \beta_{28} \text{UA_Hub} + \beta_{29} \text{NW_Hub} + \beta_{30} \text{DL_Hub} + \beta_{31} \text{CO_Hub})
\end{aligned}$$

	Coeff.	SE	Prob.	MEΦ	MEΛ	ME OR	RCI
Dist.	0.3946	0.0698	0.0000	0.3506	0.0323	22.0833	0%
Dist. ²	-0.1179	0.0146	0.0000	-0.1048	-0.0096	8.0338	0%
Pop.	0.5251	0.0526	0.0000	0.4665	0.0429	31.2407	6%
Pop. ²	-0.0358	0.0067	0.0000	-0.0318	-0.0029	9.2593	1%
Comp. [*]	0.3584	0.0314	0.0000	0.3184	0.0293	21.6828	1%
Comp. ^{*2}	-0.0754	0.0101	0.0000	-0.0670	-0.0062	8.5097	0%
Leisure	0.0481	0.0470	0.3061	0.0427	0.0039	10.6595	0%
Business	0.6885	0.0476	0.0000	0.6117	0.0563	34.2038	9%
Region1	1.5516	0.1439	0.0000	1.3784	0.1268	177.4820	3%

continued on next page

<i>continued from previous page</i>							
	Coeff.	SE	Prob.	ME Φ	ME Λ	ME OR	RCI
Region2	1.2000	0.1464	0.0000	1.0661	0.0981	96.1694	0%
Region3	1.0499	0.1469	0.0000	0.9327	0.0858	71.4237	0%
Region4	1.1313	0.1539	0.0000	1.0051	0.0925	82.4568	0%
Region5	0.9769	0.1504	0.0000	0.8679	0.0798	62.0772	0%
Region6	0.8760	0.1522	0.0000	0.7783	0.0716	54.9966	0%
Region7	0.9590	0.1475	0.0000	0.8520	0.0784	61.8032	0%
Region8	1.2208	0.1708	0.0000	1.0846	0.0998	108.4085	0%
Region9	0.6482	0.1479	0.0000	0.5758	0.0530	34.2658	0%
Region10	0.8595	0.1700	0.0000	0.7636	0.0703	47.5634	0%
Region11	1.0622	0.1733	0.0000	0.9437	0.0868	76.3539	0%
Region12	0.6024	0.1530	0.0001	0.5352	0.0492	29.6936	0%
Region13	1.0532	0.2700	0.0001	0.9357	0.0861	78.5214	0%
Region14	0.9384	0.1768	0.0000	0.8337	0.0767	63.0863	0%
SW_Hub_2	1.2124	0.0991	0.0000	1.0771	0.0991	136.9271	0%
SW_Hub_3	1.5746	0.0934	0.0000	1.3989	0.1287	271.9172	1%
SW_Hub_4	1.9697	0.0909	0.0000	1.7499	0.1610	574.8552	8%
AA_Hub	-0.9380	0.0657	0.0000	-0.8334	-0.0767	1.4479	0%
US_Hub	-0.2764	0.0832	0.0009	-0.2455	-0.0226	5.3198	0%
UA_Hub	-0.3811	0.0609	0.0000	-0.3386	-0.0312	4.7756	0%
NW_hub	-0.7821	0.0611	0.0000	-0.6949	-0.0639	1.9663	1%
DL_Hub	-1.2280	0.0805	0.0000	-1.0910	-0.1004	0.8607	1%
CO_Hub	-0.9518	0.1143	0.0000	-0.8456	-0.0778	1.3481	0%
R ²	48%						
OR	10.137						

Table 17: Results of Model 6.

5.7 Model 7: Secondary airports

This model includes the estimated impact of secondary airports, $Dist.\leq 200km$, Hub_{0_50} and Hub_{50_100} . As described in Section 3.4.2, accounting for secondary airports may explain an important part of the behaviour of a low-cost carrier like Southwest Airlines.

$$\begin{aligned}
Prob(y = 1) = & F(\beta_0 + \beta_1 Dist. + \beta_2 Dist^2 + \beta_3 Pop. + \beta_4 Pop^2 \\
& + \beta_5 Comp.^* + \beta_6 Comp.^{*2} + \beta_7 Leisure + \beta_8 Business \\
& + \sum_{i=1}^{14} \beta_{i+8} Region_i + \sum_{i=2}^4 \beta_{21+i} SW_Hub_i + \beta_{26} AA_Hub \\
& + \beta_{27} US_Hub + \beta_{28} UA_Hub + \beta_{29} NW_Hub + \beta_{30} DL_Hub \\
& + \beta_{31} CO_Hub + \beta_{32} Dist. \leq 200 + \beta_{33} Hub_0_50 + \beta_{34} Hub_50_100)
\end{aligned}$$

	Coeff.	SE	Prob.	ME Φ	ME Λ	ME OR	RCI
Dist.	0.2464	0.0686	0.0003	0.2194	0.0186	17.7365	0%
Dist.	-0.0954	0.0139	0.0000	-0.0850	-0.0072	9.2726	0%
Pop.	0.6737	0.0579	0.0000	0.6001	0.0508	46.9860	6%
Pop. ²	-0.0439	0.0073	0.0000	-0.0391	-0.0033	10.0092	1%
Comp.*	0.3843	0.0333	0.0000	0.3423	0.0290	25.4824	1%
Comp. ^{*2}	-0.0765	0.0105	0.0000	-0.0681	-0.0058	9.3245	0%
Leisure	0.0434	0.0491	0.3763	0.0387	0.0033	11.3962	0%
Business	0.6228	0.0479	0.0000	0.5548	0.0470	34.3304	9%
Region1	1.7457	0.1530	0.0000	1.5549	0.1317	309.9745	3%
Region2	1.3423	0.1540	0.0000	1.1956	0.1013	150.5438	0%
Region3	1.1885	0.1546	0.0000	1.0587	0.0897	114.3534	0%
Region4	1.2497	0.1621	0.0000	1.1132	0.0943	127.3877	0%
Region5	1.1154	0.1581	0.0000	0.9935	0.0841	96.6219	0%
Region6	0.8930	0.1611	0.0000	0.7954	0.0674	64.6034	0%
Region7	0.9814	0.1545	0.0000	0.8742	0.0740	73.3247	0%
Region8	1.3561	0.1808	0.0000	1.2079	0.1023	172.2953	0%
Region9	0.6794	0.1551	0.0000	0.6052	0.0513	40.6086	0%
Region10	0.8382	0.1755	0.0000	0.7466	0.0632	54.7984	0%
Region11	1.1418	0.1801	0.0000	1.0171	0.0861	109.4376	0%
Region12	0.5660	0.1590	0.0004	0.5041	0.0427	32.2130	0%
Region13	1.1266	0.2761	0.0000	1.0035	0.0850	109.1417	0%
Region14	1.0691	0.1837	0.0000	0.9523	0.0807	95.4465	0%
SW_Hub_2	1.1675	0.1066	0.0000	1.0399	0.0881	138.0710	0%
SW_Hub_3	1.5641	0.1016	0.0000	1.3932	0.1180	299.6611	1%
SW_Hub_4	2.0134	0.0998	0.0000	1.7934	0.1519	715.3739	8%
AA_Hub	-0.7657	0.0677	0.0000	-0.6820	-0.0578	2.4099	0%

continued on next page

<i>continued from previous page</i>							
	Coeff.	SE	Prob.	ME Φ	ME Λ	ME OR	RCI
US_Hub	-0.1639	0.0844	0.0520	-0.1460	-0.0124	7.2754	0%
UA_Hub	-0.0844	0.0612	0.1683	-0.0751	-0.0064	9.1378	0%
NW_Hub	-0.6807	0.0691	0.0000	-0.6063	-0.0514	2.5708	1%
DL_Hub	-1.1106	0.0914	0.0000	-0.9893	-0.0838	1.1497	1%
CO_Hub	-0.7150	0.1107	0.0000	-0.6368	-0.0539	2.6243	0%
Dist. \leq 200	-2.6026	0.6062	0.0000	-2.3182	-0.1963	0.0522	0%
Hub_0_50	-0.9726	0.0780	0.0000	-0.8663	-0.0734	1.4990	0%
Hub_50_100	0.1398	0.0582	0.0163	0.1245	0.0105	16.3324	2%
R ²	51%						
OR	11.1665						

Table 18: Results of Model 7.

5.8 Model 8: Congestion effects

In this model the effects of congestion in form of delayed departures and arrivals, *Max_DEP_Delay* and *Max_ARR_Delay* are introduced. Once more, the effect of correlation must be controlled for, because a delayed arrival at an airport can cause a delay for the following departure.

$$\begin{aligned}
Prob(y = 1) = & F(\beta_0 + \beta_1 Dist. + \beta_2 Dist^2 + \beta_3 Pop. + \beta_4 Pop^2 \\
& + \beta_5 Comp.^* + \beta_6 Comp.^2 + \beta_7 Leisure + \beta_8 Business \\
& + \sum_{i=1}^{14} \beta_{i+8} Region_i + \sum_{i=2}^4 \beta_{21+i} SW_Hub_i + \beta_{26} AA_Hub \\
& + \beta_{27} US_Hub + \beta_{28} UA_Hub + \beta_{29} NW_Hub + \beta_{30} DL_Hub \\
& + \beta_{31} CO_Hub + \beta_{32} Dist. \leq 200 + \beta_{33} Hub_0_50 + \\
& \beta_{34} Hub_50_100 + \beta_{35} MAX_Delay)
\end{aligned}$$

For the further modelling the impact of delays is captured by only taking

	DEP_Delay	ARR_Delay
DEP_Delay	1.0000	
ARR_Delay	0.6892	1.0000

Table 19: Covariance data matrix for Model 8 before auxiliary regression.

one congestion variable, *MAX_Delay*, which is the maximum of either delay or departure. The regression results for the new regression with a single delay parameter are given in Table 21.

Table 19 gives a correlation between the maximum of delayed departures and delayed arrivals of about 69%. Table 20 gives the regression result with the two variables *Max_DEP_Delay* and *Max_ARR_Delay*.

	Coeff.	SE	Prob.	ME Φ	ME Λ	ME OR	RCI
Dist.	0.2753	0.0722	0.0001	0.0052	0.0084	0.0546	0%
Dist. ²	-0.1071	0.0146	0.0000	-0.0020	-0.0033	0.0266	0%
Pop.	0.6627	0.0544	0.0000	0.0124	0.0203	0.1308	6%
Pop. ²	-0.0424	0.0068	0.0000	-0.0008	-0.0013	0.0295	1%
Comp.*	0.4212	0.0333	0.0000	0.0079	0.0129	0.0781	1%
Comp.* ²	-0.0824	0.0104	0.0000	-0.0015	-0.0025	0.0272	0%
Leisure	-0.0580	0.0515	0.2599	-0.0011	-0.0018	0.0278	0%
Business	0.5888	0.0482	0.0000	0.0111	0.0181	0.0945	9%
Region1	1.3360	0.1570	0.0000	0.0251	0.0410	0.4283	3%
Region2	1.2498	0.1565	0.0000	0.0235	0.0383	0.3625	0%
Region3	0.9548	0.1570	0.0000	0.0179	0.0293	0.2139	0%
Region4	0.7598	0.1663	0.0000	0.0143	0.0233	0.1482	0%
Region5	1.0636	0.1604	0.0000	0.0200	0.0326	0.2507	0%
Region6	0.8534	0.1653	0.0000	0.0160	0.0262	0.1690	0%
Region7	0.8740	0.1566	0.0000	0.0164	0.0268	0.1755	0%
Region8	1.1850	0.1857	0.0000	0.0223	0.0363	0.3543	0%
Region9	0.6801	0.1574	0.0000	0.0128	0.0209	0.1165	0%
Region10	0.6920	0.1755	0.0001	0.0130	0.0212	0.1227	0%
Region11	0.8833	0.1817	0.0000	0.0166	0.0271	0.1973	0%
Region12	0.4942	0.1607	0.0021	0.0093	0.0152	0.0816	0%
Region13	0.5516	0.2922	0.0590	0.0104	0.0169	0.1079	0%
Region14	0.9557	0.1855	0.0000	0.0179	0.0293	0.2232	0%
SW_Hub_2	1.1312	0.1120	0.0000	0.0212	0.0347	0.3768	0%

continued on next page

	Coeff.	SE	Prob.	MEΦ	MEA	ME OR	RCI
Dist.	0.2815	0.0718	0.0001	0.0132	0.0168	0.1157	0%
Dist. ²	-0.1031	0.0144	0.0000	-0.0048	-0.0062	0.0557	0%
Pop.	0.6328	0.0544	0.0000	0.0297	0.0378	0.2576	6%
Pop. ²	-0.0408	0.0068	0.0000	-0.0019	-0.0024	0.0616	1%
Comp. [*]	0.4184	0.0331	0.0000	0.0196	0.0250	0.1620	1%
Comp. ^{*2}	-0.0848	0.0103	0.0000	-0.0040	-0.0051	0.0563	0%
Leisure	-0.0731	0.0518	0.1586	-0.0034	-0.0044	0.0564	0%
Business	0.5834	0.0484	0.0000	0.0274	0.0348	0.1954	9%
Region1	1.3425	0.1545	0.0000	0.0630	0.0801	0.8926	3%
Region2	1.2327	0.1546	0.0000	0.0579	0.0735	0.7218	0%
Region3	0.9288	0.1549	0.0000	0.0436	0.0554	0.4188	0%
Region4	0.7981	0.1634	0.0000	0.0375	0.0476	0.3249	0%
Region5	1.0131	0.1580	0.0000	0.0476	0.0604	0.4706	0%
Region6	0.8576	0.1641	0.0000	0.0403	0.0512	0.3504	0%
Region7	0.8559	0.1548	0.0000	0.0402	0.0511	0.3475	0%
Region8	1.1516	0.1836	0.0000	0.0541	0.0687	0.6862	0%
Region9	0.6764	0.1552	0.0000	0.0318	0.0404	0.2390	0%
Region10	0.6410	0.1742	0.0002	0.0301	0.0382	0.2294	0%
Region11	0.8670	0.1799	0.0000	0.0407	0.0517	0.3927	0%
Region12	0.4690	0.1583	0.0031	0.0220	0.0280	0.1614	0%
Region13	0.7155	0.2810	0.0109	0.0336	0.0427	0.3059	0%
Region14	0.9172	0.1826	0.0000	0.0431	0.0547	0.4291	0%
SW_Hub_2	1.1457	0.1109	0.0000	0.0538	0.0684	0.8006	0%

continued on next page

<i>continued from previous page</i>							
	Coeff.	SE	Prob.	ME Φ	ME Λ	ME OR	RCI
SW_Hub_3	1.5821	0.1060	0.0000	0.0743	0.0944	1.8638	1%
SW_Hub_4	2.0139	0.1045	0.0000	0.0946	0.1202	4.2393	8%
AA_Hub	-0.7655	0.0681	0.0000	-0.0359	-0.0457	0.0149	0%
US_Hub	-0.0361	0.0912	0.6925	-0.0017	-0.0022	0.0525	0%
UA_Hub	-0.0312	0.0615	0.6117	-0.0015	-0.0019	0.0609	0%
NW_Hub	-0.6742	0.0723	0.0000	-0.0317	-0.0402	0.0157	1%
DL_Hub	-1.2333	0.0965	0.0000	-0.0579	-0.0736	0.0054	1%
CO_Hub	-0.7074	0.1129	0.0000	-0.0332	-0.0422	0.0166	0%
Dist. \leq 200	-2.7367	0.6502	0.0000	-0.1285	-0.1633	0.0002	0%
Hub_0_50	-0.9010	0.0767	0.0000	-0.0423	-0.0538	0.0107	0%
Hub_50_100	0.1559	0.0594	0.0087	0.0073	0.0093	0.1047	2%
MAX_Delay	-10.290	0.5484	0.0000	-0.4832	-0.6139	0.0000	3%
R ²	53%						
OR	0.0681						

Table 21: Results of Model 8 after the auxiliary regression.

5.9 *Model 9: Low-cost carrier competition*

As a final step, the last variable describing the potential influence of competition is introduced: the presence of one of the defined low-cost carriers, namely *Jetblue*, *Airtran*, *Frontier* and *Alaska*.

$$\begin{aligned}
Prob(y = 1) = & F(\beta_0 + \beta_1 Dist. + \beta_2 Dist.^2 + \beta_3 Pop. + \beta_4 Pop.^2 \\
& + \beta_5 Comp.^* + \beta_6 Comp.^{*2} + \beta_7 Leisure + \beta_8 Business \\
& + \sum_{i=1}^{14} \beta_{i+8} Region_i + \sum_{i=2}^4 \beta_{21+i} SW_Hub_i + \beta_{26} AA_Hub \\
& + \beta_{27} US_Hub + \beta_{28} UA_Hub \beta_{29} NW_Hub + \beta_{30} DL_Hub \\
& + \beta_{31} CO_Hub + \beta_{32} Dist. \leq 200 + \beta_{33} Hub_0_50 \\
& + \beta_{34} Hub_50_100 + \beta_{35} MAX_Delay + \beta_{36} Jetblue \\
& + \beta_{37} Airtran + \beta_{38} Frontier + \beta_{39} Alaska)
\end{aligned}$$

[illegible]

<i>continued from previous page</i>							
	Coeff.	SE	Prob.	ME Φ	ME Λ	ME OR	RCI
SW_Hub_2	1.1839	0.1529	0.0000	0.0345	0.0494	0.7337	0%
SW_Hub_3	1.7721	0.1475	0.0000	0.0516	0.0739	2.2328	1%
SW_Hub_4	2.2447	0.1482	0.0000	0.0653	0.0936	5.3952	8%
AA_Hub	-0.7534	0.0690	0.0000	-0.0219	-0.0314	0.0104	0%
US_Hub	-0.0559	0.0938	0.5510	-0.0016	-0.0023	0.0332	0%
UA_Hub	-0.2757	0.0756	0.0003	-0.0080	-0.0115	0.0244	0%
NW_Hub	-0.6687	0.0747	0.0000	-0.0195	-0.0279	0.0104	1%
DL_Hub	-1.2884	0.1025	0.0000	-0.0375	-0.0537	0.0032	1%
CO_Hub	-0.6573	0.1157	0.0000	-0.0191	-0.0274	0.0120	0%
Dist. \leq 200	-2.7550	0.6696	0.0000	-0.0802	-0.1149	0.0001	0%
Hub_0_50	-0.8027	0.0760	0.0000	-0.0234	-0.0335	0.0090	0%
Hub_50_100	0.1809	0.0613	0.0032	0.0053	0.0075	0.0747	2%
MAX_Delay	-10.915	0.5820	0.0000	-0.3177	-0.4552	0.0000	3%
Jetblue	0.3387	0.0414	0.0000	0.0099	0.0141	0.0784	4%
Airtran	0.0012	0.0006	0.0528	0.0000	0.0000	0.0457	0%
Frontier	0.0051	0.0007	0.0000	0.0001	0.0002	0.0461	1%
Alaska	0.0025	0.0004	0.0000	0.0001	0.0001	0.0458	2%
R ²	54%						
OR	0.0261						

Table 22: Results of Model 9.

Unsuccessful Variables: Other variables were added to the model without success. The first effect that could not be measured is the use of a substitute airport. The idea was to create a dummy variable that gave information on the presence of a secondary airport close to a hub. The intuition is that Southwest would have preferred a secondary, cheaper and less congested airport for its operations. These variables turned out to be insignificant in our model. It is probably due a poor specification of the variables, as it is quite difficult to create a variable that measures exactly the effect of a substitute airport.

The second type of variables that were not successfully added are network specific variables. For example, a dummy variable accounting for the Wright

Amendment was added to exclude some routes that cannot be served out of Dallas Love Field (DAL). The problem is that this variable had too many zeroes, introducing some collinearity with the constant in the regression. More thought can be given on how to introduce these network specificities into the model.

6 Model interpretation

In the previous part we created the model, adding variables step-by-step to get an R^2 as high as possible. In this section, the objective is to study the results when applying this model to the entire dataset, quarter-by-quarter.

6.1 Adapting the model to the dataset

When applying the code to the entire dataset, some errors occur. The first issue concerns the two variables UA_Hub and US_Hub. The model cannot distinguish between the values 0 or 1. These two variables are dropped for the analysis. Also, the regression fails for some quarters because the Dist.<=200 (km) is too restrictive. It is therefore increased to 300km. The regression works once these changes are made. It then appears that the estimated parameter $\hat{\beta}$ for the variable MAX_Delay has a much larger value than the others. MAX_Delay is therefore scaled in the model to guarantee coefficients of similar magnitudes. This concludes the changes that have to be made to the model.

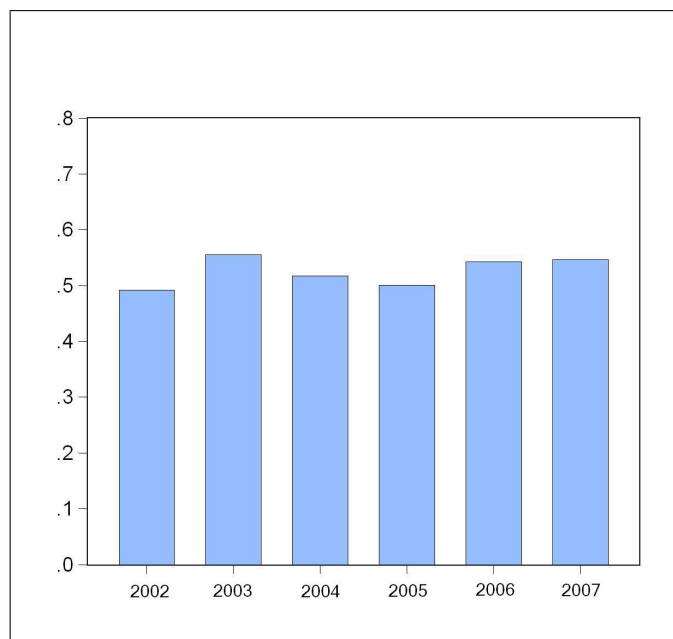


Figure 5: R^2 values for 2002 to 2007 (unfiltered data)

6.2 Interpretation of the results

6.2.1 Initial analysis

The model we estimated for 2007:4 remains quite good for 2002 to 2006. The model is relatively robust as R^2 remains quite high, around 55% for all quarters.

The estimated parameters $\hat{\beta}$ are plotted below on Figure 6. The numbers 5, 10, 15 and 20 on the X-axis represent the quarters. For example, quarter 15 refers to 2005:3. Several variables seem to switch between the fourth and the fifth quarters. These variables include Region1, Region4, Region5, Region13, NW_Hub and MAX_Delay. On first reflection, this shift suggests the model is capturing a change in Southwest's strategy. Estimated parameters $\hat{\beta}$ are

plotted a second time with a scaled y-axis in Figure 7 to measure the change in magnitude. MAX_Delay has a negative shift of 600% from -0.3 to -2.2. When trying to get the best fit, the model shows some variables follow MAX_Delay's shift which would mean correlation between variables. This idea is rejected after inspecting the data correlation matrix: none of the variables that shift are correlated with MAX_Delay.

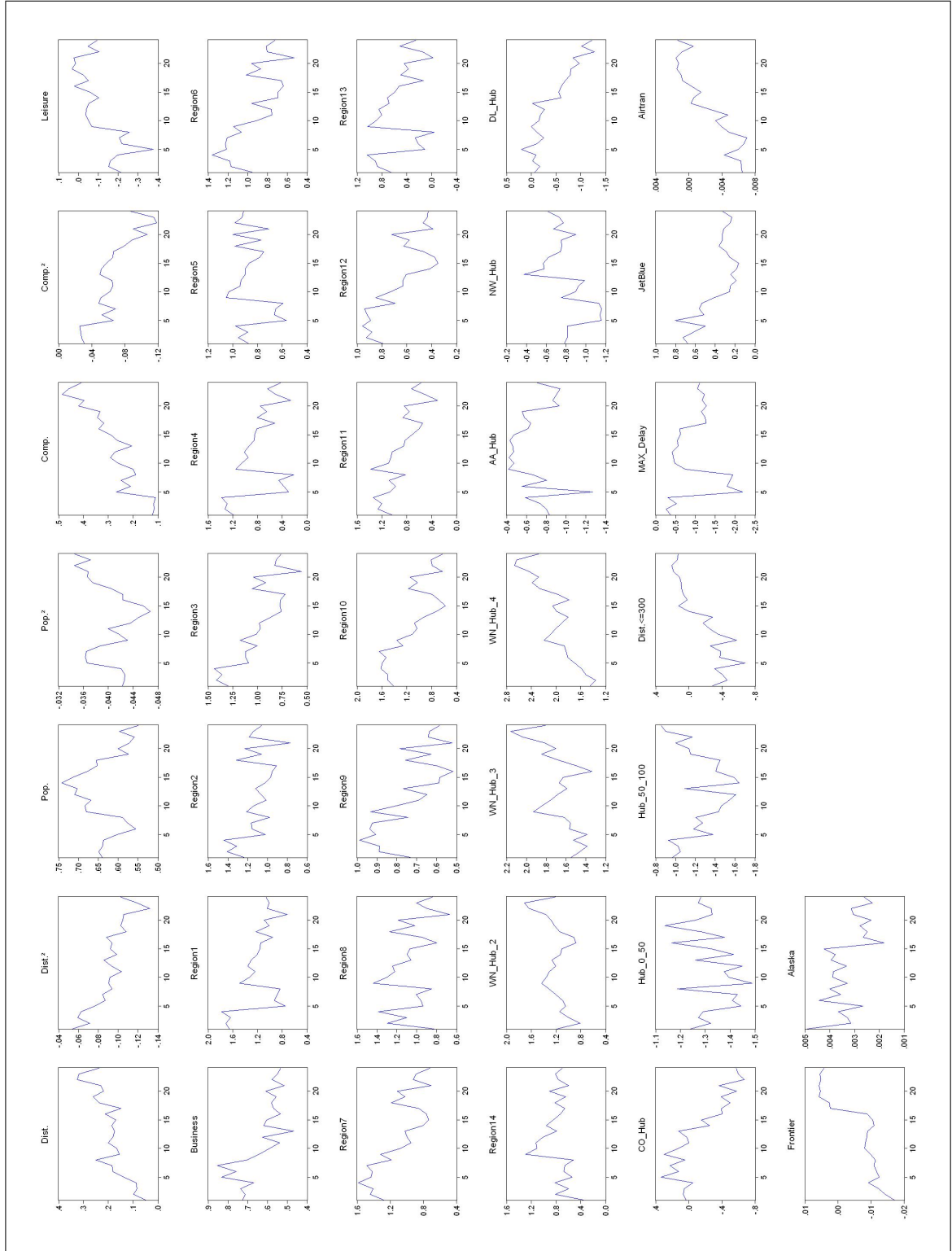


Figure 6: $\hat{\beta}$ value for all variables (unfiltered data). The X-axis counts the quarters : quarter 15 is 2005:3.

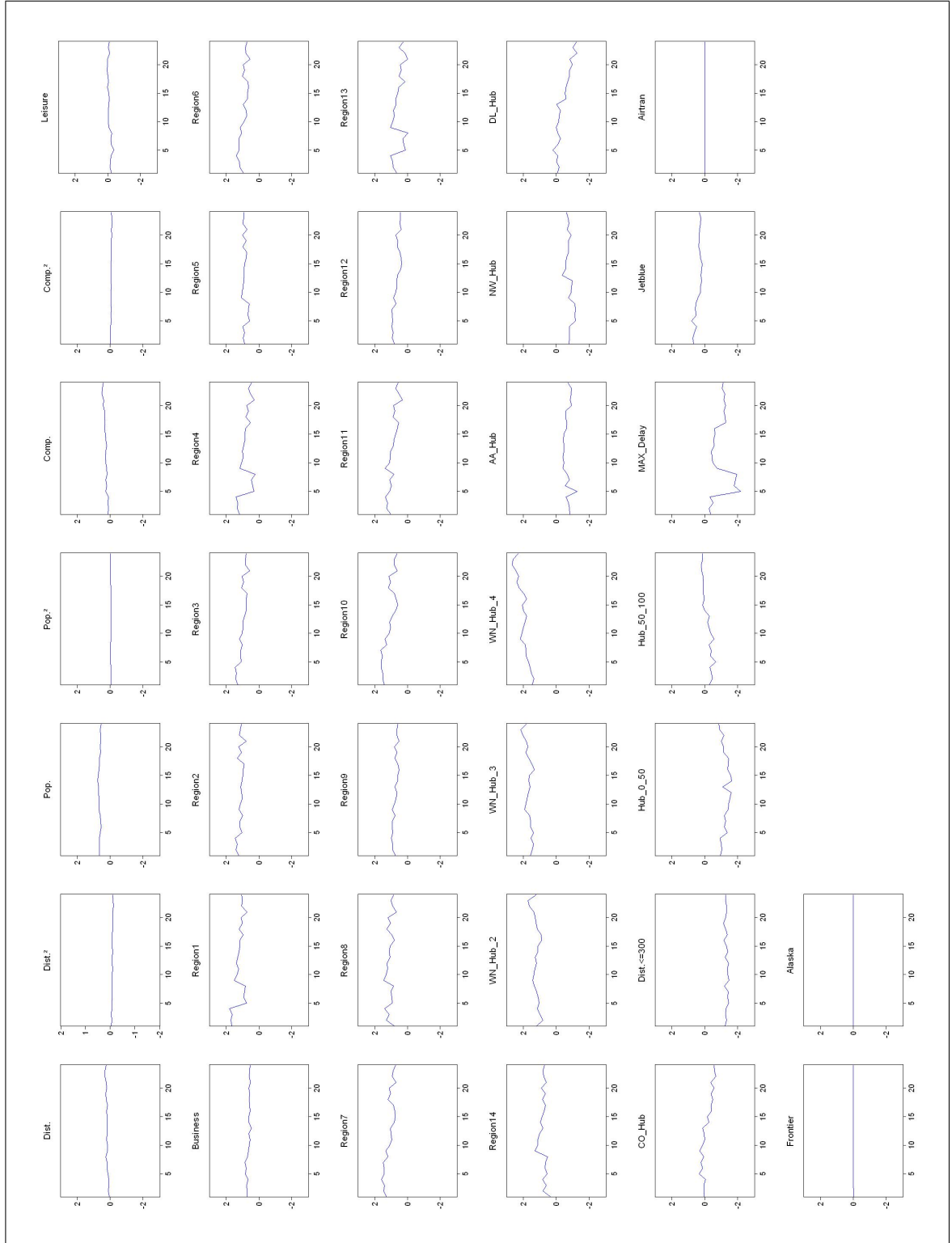


Figure 7: $\hat{\beta}$ value for all variables on a normalized y-axis (unfiltered data). Allows comparison of relative magnitudes.

This coefficient shift means routes that have high delay are not being served as much as before: Southwest became very concerned by congestion issues between 2002:4 and 2003:1. Does this mean a change in strategy? The answer should come from the data. If the answer is yes then either a complete re-organization of Southwest's network to avoided crowded routes took place, or airports served by Southwest suddenly get less congested. The former seems more plausible but 67 changes of routes, both new routes and routes that no longer exist, took place between 2002:4 and 2003:1, which accounts for a 5% change in the 1461 total routes. This is not really a significant network change, so not a valid explanation for MAX_Delay's break. Moreover, the average MAX_Delay for routes served (not served) is 19% (19%) in 2002:4 and 16% (21%) in 2003:3, which shows that Southwest is clearly serving fewer routes that are congested. This is illustrated in Figure 8.

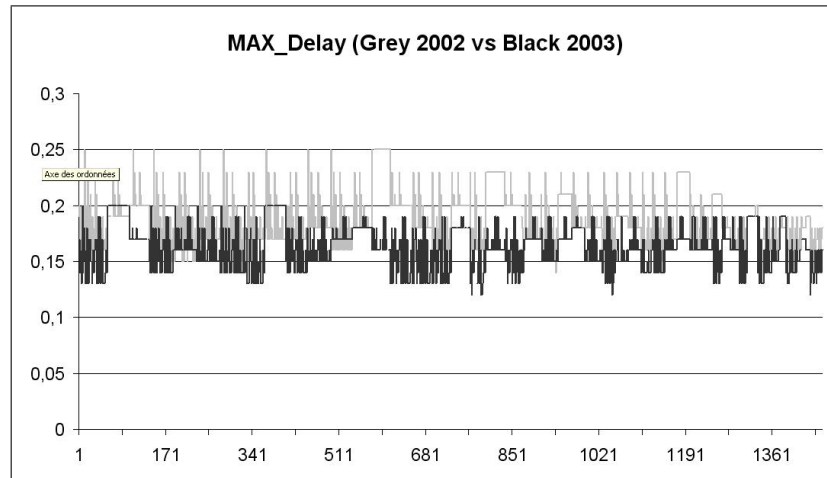


Figure 8: MAX_Delay plotted across the 1461 routes served in 2002 (grey) and 2003 (black). Notice delay is shorter for routes served in 2003.

To specifically evaluate how this affects Southwest's presence, we would normally look at the marginal effects of each variables. In this case, marginal effects are difficult to analyze because they are all equal to zero. This comes as no surprise: marginal effects are worth $\phi(x'\beta)\beta$ for the probit model. The argument in the probability density function is the product of the mean-values by the estimated parameters. For MAX_Delay, this product decreases from -0.5 to -4.5. The value of the normal probability distribution function is extremely low: $\phi(-4.5) = 0.0000001$. Thus marginal effects are strange to interpret, any change in the variables would have absolutely no influence on Southwest's presence. To sum up, MAX_Delay is changing greatly from 2002 to 2003. It is hard to understand what is really going and saying none of the variables has an effect on Southwest's presence seems absurd, especially as R^2 remains high.

6.2.2 Marginal effects

Regional variables: For 2004 and 2005, we can interpret the marginal effects of each variable on Southwest's presence as seen in Figure 6.2.2. In 2005:2, an increase of 0.1 points of Region5 increases the probability of Southwest's presence by 2.7 points, bearing in mind region5 counts the number of NE-SW routes. Mathematically, an increase of 0.1 point of Region5 represents about 25 new routes because the mean of Region5 is 2484. In real life though, the only way to open new routes is by creating new airports. So the number of NE-SW routes is constant across the dataset, no airports are created. To make things clear, the number of NE-SW routes remains constant and would be costly to change (new airport). What this marginal effect means in real life is that Southwest is very attracted by connecting the NE and SW regions. Indeed, the SW region

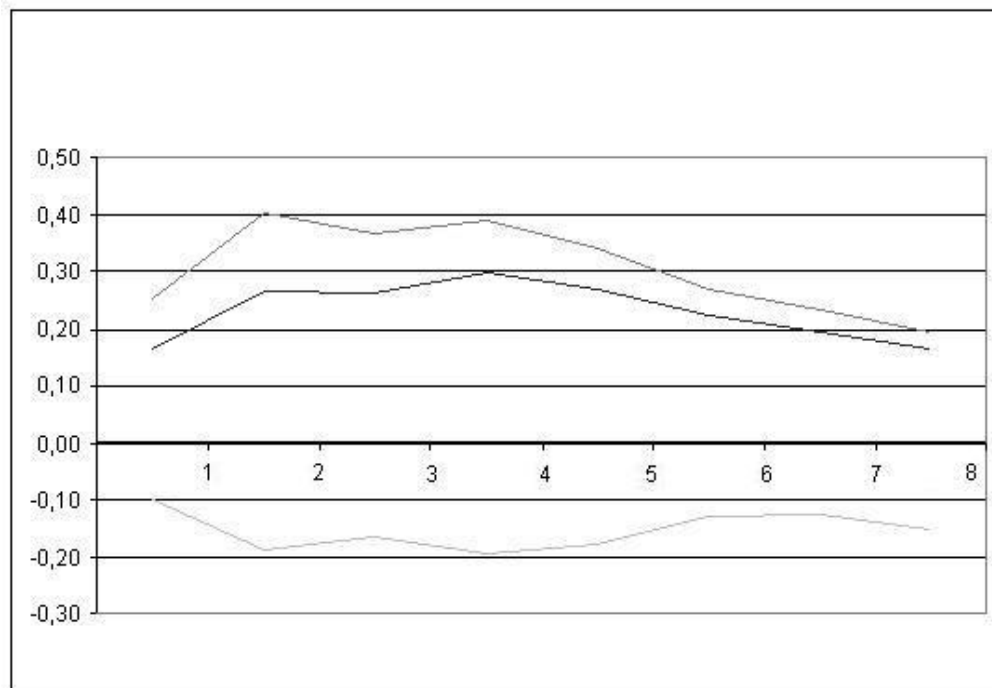


Figure 9: Marginal effects for years 2004 and 2005. Top to bottom: Region5, Population, AA_Hub. On the x-axis, number 6 represents 2005:2.

includes densely populated MSAs (Los Angeles, Dallas, Houston) and so does the NE region (New York, Boston, Philadelphia).

	Region code	Regions	Marginal effect
1.	Region1	SW_SW	0.36
2.	Region8	SE_NW	0.33
3.	Region2	SW_SE	0.31

Table 23: Top three regions Southwest wants to connect based on 2005Q2.

	Region code	Regions	Marginal effect
13.	Region6	SE_SE	0.21
14.	Region9	SE_NE	0.18
15.	Region12	MW_NE	0.12

Table 24: Last three regions Southwest wants to connect based on 2005Q2.

Other variables for 2005:2: An increase of 0.1 points of population increases the probability of Southwest's presence by 1.9 points. This means densely populated areas have a positive marginal effect on Southwest's presence. To give a rough idea, an increase of population of 0.1 point represents 9,500 inhabitants.

An increase of 0.1 points of AA_Hub decreases the probability of Southwest's presence by 1.3 points. The negative marginal effect means that Southwest avoids having routes which have one of American Airlines' hubs at an endpoint. It could also mean American Airlines is avoiding having hubs where Southwest operates but given the investment it represents to set up or move a hub, it is very unlikely American Airlines is placing its hubs to avoid Southwest. In any case, the model is suggesting something is happening between American Airlines' hubs and Southwest's strategy.

AirTran, Frontier and Alaska have very little marginal effect on Southwest's presence on the route across the dataset. JetBlue however has a strong positive marginal effect which suggests either Southwest is attacking JetBlue, or JetBlue is attacking Southwest. Again, the model is saying something is going on between Southwest and JetBlue; a dynamic approach with details on pricing policies would definitely give more details on the nature of the competition between these two carriers.

Interpretation of the marginal effects for 2003, 2006 and 2007 is more difficult

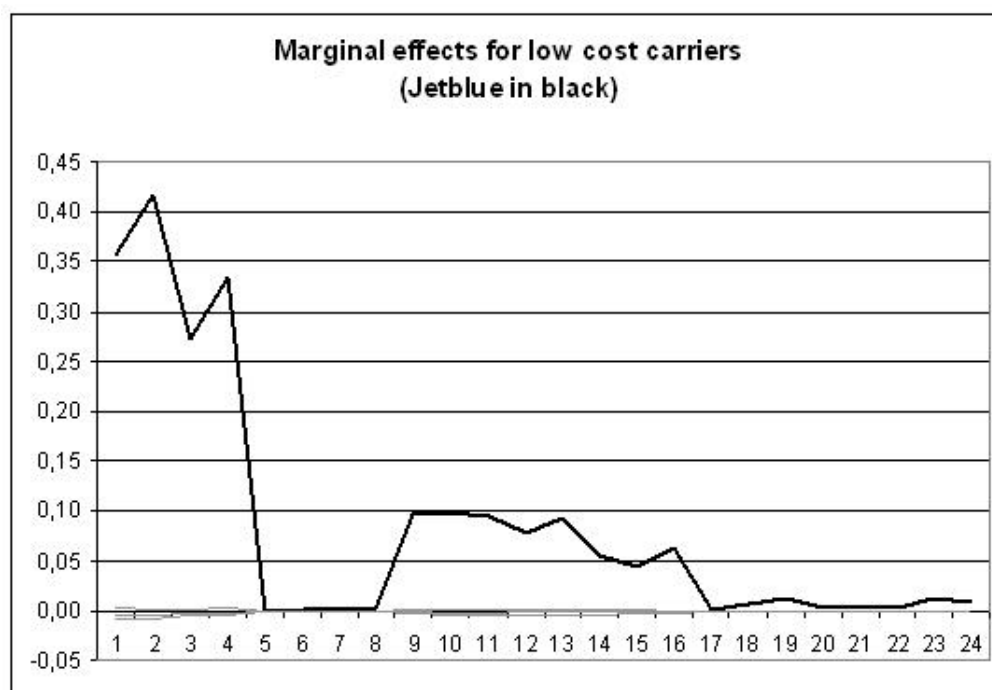


Figure 10: Marginal effects of other lowcost carriers. JetBlue's presence seems to have a significant effect on Southwest's presence. Possible battle between these two carriers.

for the reasons we mentioned in 6.2.1. MAX_Delay is changing greatly from 2002 to 2003 hence pushing the marginal effect of all variables to almost zero.

6.3 Filtered data

Filtered data allows to exclude most of the routes that have one stop. Without going into details, we simply increased the passenger threshold for each route, in order to account for a certain distinction between nonstop and one-stop flights. A reasonable attempt to so is the introduction a threshold for passenger numbers. City-pairs with passenger numbers below that threshold are supposed to be one-stop connections, otherwise nonstop connections. The DOT database reports flights on the base of O-and D-markets and does not take eventual stopovers (one-stop flights) into account. To identify a certain number of one-stop flights, it is supposed that only flights above a certain number of passengers are served as nonstop flights. This threshold is set to be 90 (minimum number of passengers per quarter in the dataset, corresponding to 900 passengers per quarter in reality). Southwest now has fewer routes in our model, but most of these routes are served by not non-stop flights. The code works for all quarters except 2007:3. The model is unable to find an optimum when running the maximum likelihood algorithm. This is due to a poorly conditioned objective surface for 2007:3. It is hard to understand why this is happening for just one quarter. We will therefore decide to exclude this quarter from our results.

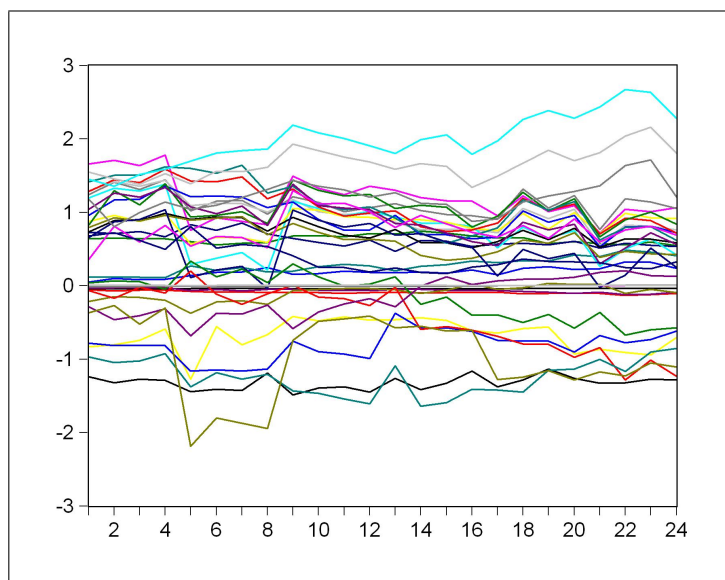


Figure 11: Graph showing the value of the estimated parameters for the unfiltered data.

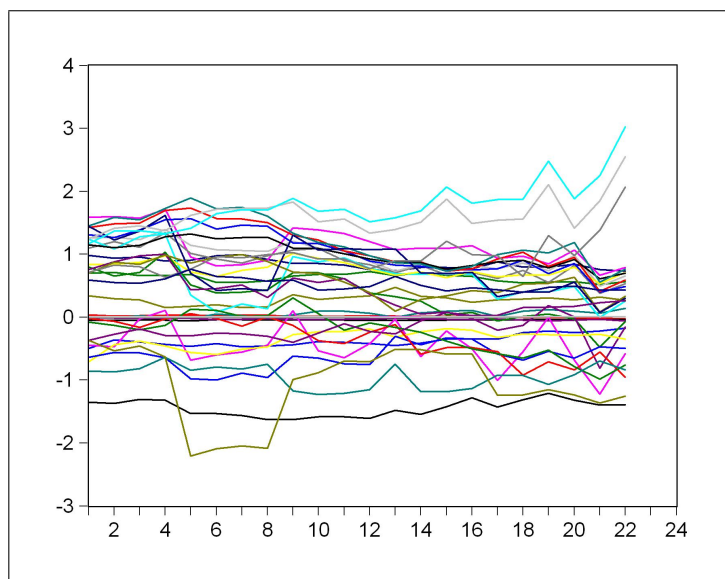


Figure 12: Graph showing the value of the estimated parameters for the filtered data. 2007:3 and 2007:4 are missing because of an error in the regression. Results are similar to the unfiltered data.

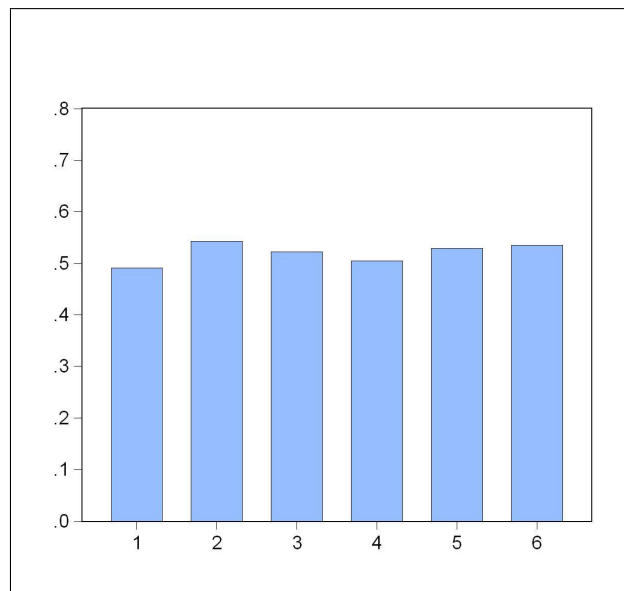


Figure 13: R^2 remains quite high for the filtered data which shows some robustness in the model.

This data represents a test of robustness for our code, most of the results seen on Figures 14 and 16 show similar results for both filtered and non-filtered data. The R^2 plot graph is corrected for 2007 by averaging the first, second and fourth quarter.

Notice the relative magnitude of the marginal effects on the unfiltered data is greater. This suggests a change in the variables has more impact on Southwest's presence. MAX_Delay is shifting again at the end of 2002 for the same reasons as we saw before.

To sum up, filtered data is used to take a closer look at only the non-stop routes. The model manages to fit the data quite well and results are pretty similar. The results are generally the same with some interesting differences, Region14 is insignificant which proves Southwest is not interested in connecting

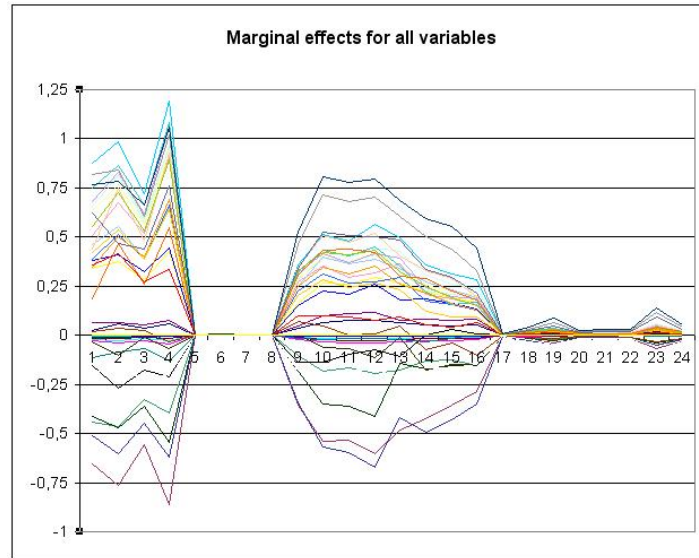


Figure 14: Marginal effects for the unfiltered data.

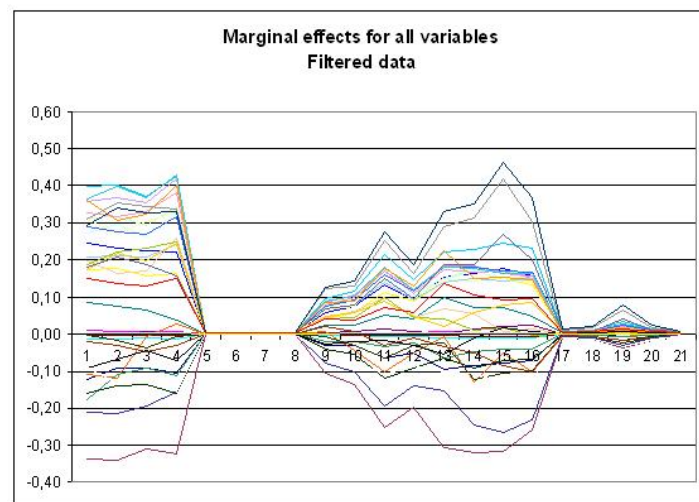


Figure 15: Marginal effects for the unfiltered data. Decrease in magnitude of the coefficients suggests the variables have more impact on Southwest's presence.

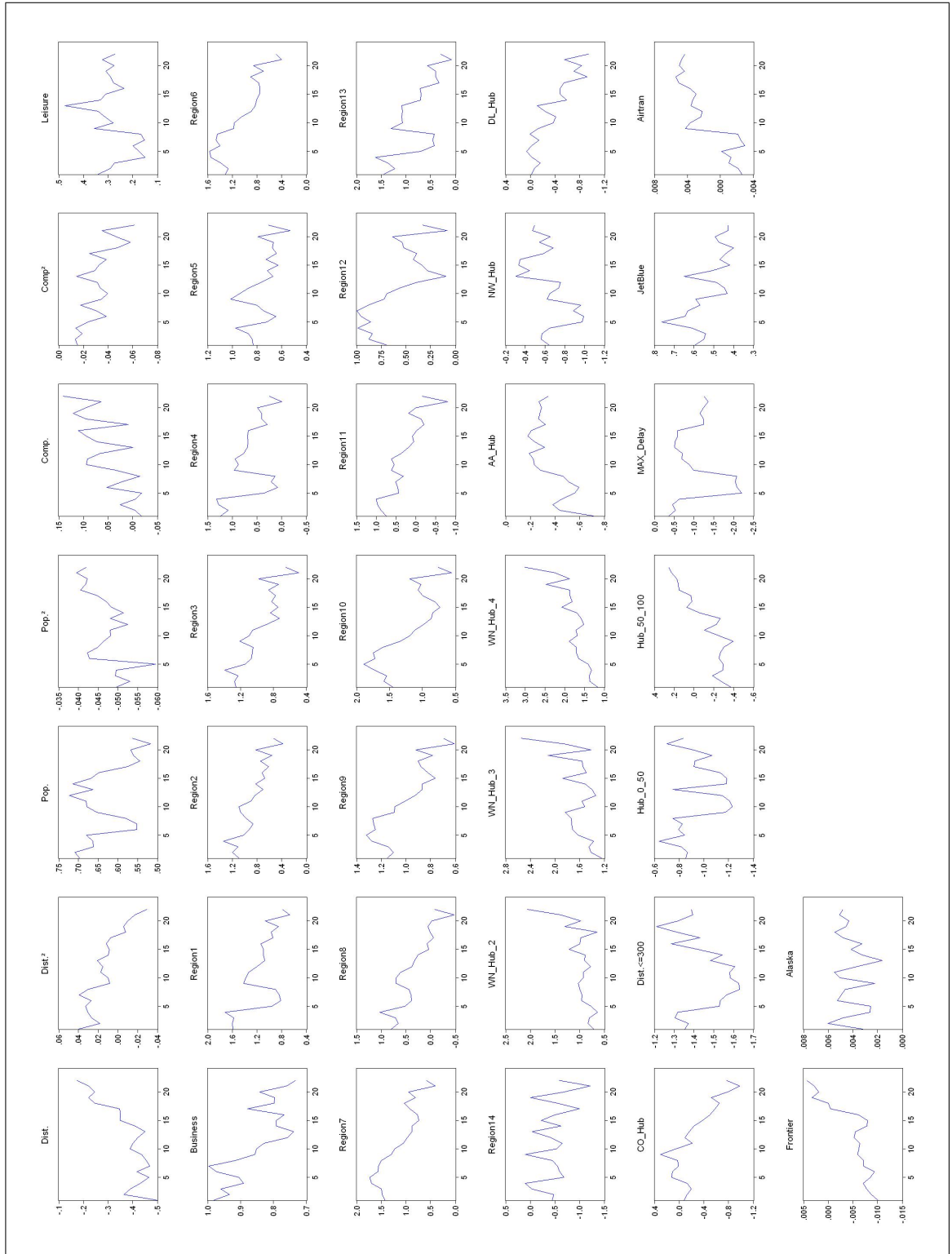


Figure 16: Output of the regression for the filtered data. Magnitude similar to the non-filtered model.

the North East and the center-region with non stop flights. Presence of competitor or competitor's hub has virtually no impact on Southwest. Maybe because they are the most competitive on non-stop flights.

6.3.1 Entire dataset

In this section, the data used is made of the twenty-four aggregated data files. What is the difference? The model is now making one big regression on all the data instead of twenty-four smaller ones. This is done for two reasons. First, this is a robustness test for the model. The data is not identical per say, so the model will have to find the best fit for the variables from 2002:1 to 2007:4. This allows to see how the model copes with various types of data. Second, this allows to interpret the data as a whole, looking at Southwest' presence over six years. The results remain good as R^2 is worth 52%. General results are similar to those we saw before which is what one would expect. Two variables become insignificant, Region9 and Region12 which suggests Southwest didn't really care about connecting the associated regions. Concerning marginal effects, two variables stand out, SW_Hub_4 has the greatest positive effect whereas Dist.<=200 has the greatest negative effect. This means Southwest is more interested in connecting its focus cities avoiding routes that are too short.

	Coeff.	SE	Prob.	MEΦ	MEΛ	ME OR	RCI	Mean
Dist.	0,1662	0,0154	0,0000	0,0184	0,0126	0,1251	0%	2,0739
Dist. ²	-0,0911	0,0032	0,0000	-0,0101	-0,0069	0,0750	0%	6,7355
Pop.	0,6029	0,0112	0,0000	0,0666	0,0457	0,3162	5%	0,9532
Pop ²	-0,0380	0,0014	0,0000	-0,0042	-0,0029	0,0822	1%	3,1948
Comp. ²	0,3052	0,0066	0,0000	0,0337	0,0231	0,1782	0%	0,0000
Comp. ^{*2}	-0,0669	0,0022	0,0000	-0,0074	-0,0051	0,0766	0%	0,0000
Leisure	-0,0802	0,0103	0,0000	-0,0089	-0,0061	0,0791	0%	0,1620
Business	0,6349	0,0105	0,0000	0,0701	0,0481	0,2913	8%	0,1338
Region1	0,5868	0,0583	0,0000	0,0648	0,0445	0,2410	2%	0,0512
Region2	0,4558	0,0586	0,0000	0,0504	0,0346	0,1992	0%	0,1178
Region3	0,2985	0,0583	0,0000	0,0330	0,0226	0,1461	0%	0,1101
Region4	0,3011	0,0583	0,0000	0,0333	0,0228	0,1488	0%	0,0367
Region5	0,2398	0,0591	0,0000	0,0265	0,0182	0,1319	0%	0,0888
Region6	0,1880	0,0604	0,0019	0,0208	0,0143	0,1162	0%	0,0654
Region7	0,3885	0,0591	0,0000	0,0429	0,0295	0,1654	0%	0,1243
Region8	0,5719	0,0606	0,0000	0,0632	0,0434	0,2766	0%	0,0414
Region9	0,0658	0,0595	0,2687	0,0073	0,0050	0,0904	0%	0,1003
Region10	0,3052	0,0616	0,0000	0,0337	0,0231	0,1390	0%	0,0571
Region11	0,3264	0,0602	0,0000	0,0361	0,0247	0,1669	0%	0,0387
Region12	-0,0238	0,0600	0,6910	-0,0026	-0,0018	0,0737	0%	0,0938
Region13	0,2939	0,0622	0,0000	0,0325	0,0223	0,1583	0%	0,0061
Region14	-0,6528	0,0674	0,0000	-0,0721	-0,0495	0,0210	0%	0,0313
SW_Hub_2	0,9608	0,0173	0,0000	0,1061	0,0728	0,7088	0%	0,1037
SW_Hub_3	1,3685	0,0167	0,0000	0,1512	0,1037	1,5265	2%	0,1274
SW_Hub_4	1,7549	0,0168	0,0000	0,1939	0,1330	3,1892	6%	0,1476
AA_Hub	-0,7731	0,0154	0,0000	-0,0854	-0,0586	0,0191	0%	0,1958
US_Hub	-0,7352	0,0199	0,0000	-0,0812	-0,0557	0,0187	0%	0,1286
UA_Hub	-1,1828	0,0301	0,0000	-0,1307	-0,0897	0,0075	1%	0,1401
NW_Hub	-0,9033	0,0159	0,0000	-0,0998	-0,0685	0,0133	2%	0,2730
DL_Hub	-0,6685	0,0151	0,0000	-0,0739	-0,0507	0,0213	1%	0,2444
CO_Hub	-0,5130	0,0229	0,0000	-0,0567	-0,0389	0,0321	0%	0,0662
Dist.≤200	-3,4498	0,1962	0,0000	-0,3811	-0,2615	0,0001	0%	0,0181
Hub_0_50	-1,0430	0,0179	0,0000	-0,1152	-0,0791	0,0107	0%	0,2132
Hub_50_100	0,1177	0,0131	0,0000	0,0130	0,0089	0,1239	2%	0,5088
MAX_Delay	-0,5271	0,0088	0,0000	-0,0582	-0,0400	0,0345	0%	2,3394
Jetblue	0,3919	0,0087	0,0000	0,0433	0,0297	0,1841	0%	0,2367
Airtran	-0,0018	0,0002	0,0000	-0,0002	-0,0001	0,0897	2%	5,6122
Frontier	0,0021	0,0002	0,0000	0,0002	0,0002	0,0905	3%	3,6983
Alaska	0,0020	0,0001	0,0000	0,0002	0,0002	0,0904	0%	7,9099
R ² OR	52% 0,0901							

Table 25: Results of the entire dataset.

6.4 Southwest performance

In this section, we are looking for possible relations between Southwest's financial performance and their network changes. Does Southwest open more routes when they have better results?

Quarterly data on Southwest's performance can be found on their website. Some key financial variables are used as indicators of Southwest's performance in the short-term, mid-term and long-term.

- Three variables are used to account for Southwest's short term performance. Net Income gives an idea of the profit made on a quarterly basis. Available Seat Mile (ASM), Fuel and oil expenses are two variables used to measure short-term costs. Fuel and Oil expenses depends on the number of flights, so it is divided by ASM to get a raw indicator. This new variable represents short-term costs.
- Total current liabilities and total current assets are used as mid-term/long-term indicators. The six years covered by our dataset is just starting to be long-term decisions. Total liabilities divided by total assets is used as a mid-term/long-term indicator.

These indicators are scaled then plotted with the number of changes in the network. A change in the network is either a route being served or a route that Southwest stops serving. A positive change is a network expansion. A negative change is a network contraction. The results are shown on Figure 17.

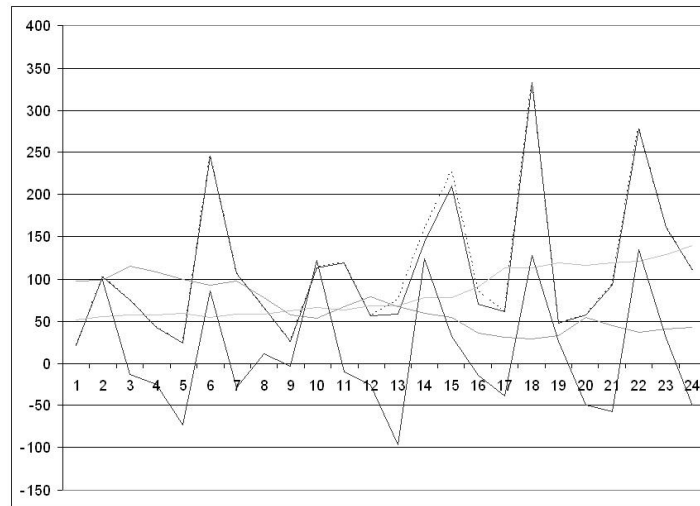


Figure 17: Performance indicators are plotted, the darker curves show net income and route changes seem to be correlated.

The graph shows that the number of changes in the network is linked to the net income. A look at the data correlation matrix shows high correlation between the change in network and Net income. This would suggest Southwest's strategy for the route network is based on short term decisions, whether the route is profitable. Longer-term strategies may be revealed either with a longer time period or maybe using other variables that represent long-term performance like long term debt less current maturities.

	Change	Net Income	Fuel/ASM	Liabilities/Assets
Change	1,0000			
Net Income	0,7001	1,0000		
Fuel/ASM	-0,0056	0,2466	1,0000	
Liabilities/Assets	-0,1795	-0,3114	-0,8082	1,0000

Table 26: Correlation between performance variables. Coefficient very high for change and Net Income.

6.5 Verification of the model

The regression results of 2007:4 are compared to the current Southwest network in terms of nonstop connections. The objective is to test whether the model has some predictive capacity. Figures 19 and 18 demonstrate the general estimation results for the route presence of Southwest for 2007:4. During this period Southwest was present on 1837 nonstop and one-stop routes; during the same period 27,966 routes have been identified that Southwest was not on. As seen in Figure 18, the model estimates for about half of the routes Southwest was actually serving, a reasonably high probability of presence, of more than 50%.

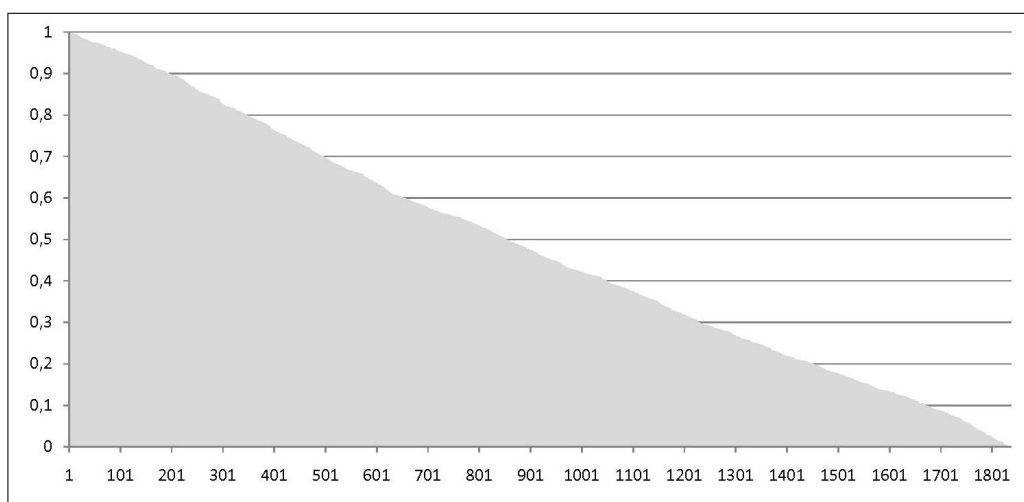


Figure 18: Fitted probabilities for Southwest's presence.

For the routes not being served, shown in Figure 19, the model tells quite well that the vast majority of these routes have indeed a low estimated probability. The increased proportion of probability, in the range between 0 and 2000, simply says that once Southwest is in an airport, even the probability of routes not served out of this airport rise.

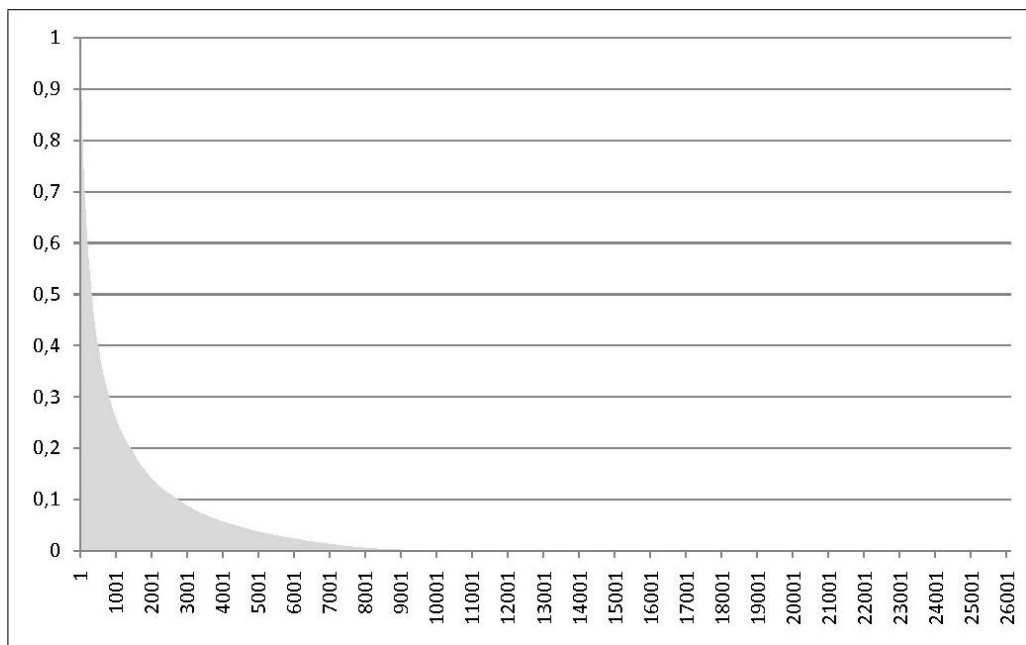


Figure 19: Fitted probabilities for Southwest not being present.

For the practical demonstration of the model, applied on Southwest's actual network, a difference is made between three types of airports:

- Airports where Southwest announced entry and did not have any operations before: LGA and MSP, Figures 20 and 21 [2].
- Airports that Southwest serves and has recently announced an increase in service. The assumption of an increase of service at these airports is based on an announcement made by Southwest in December 2008 [30]. The increase is not specified, but can be realized in terms of new added destinations and/or increase in frequencies on existing routes. Namely, these airports are MDW, DEN, BNA and LAX, Figures 22, 23, 24 and 25.
- Airports that Southwest serves but *has not* recently announced an increase in service. In contrast to the above, some airports not mentioned in this announcement are assumed not to experience a significant expansion of Southwest operations. The following airports are taken as an example: SFO and DTW, Figures 26 and 27.

The base for the verification of the model and the comparison of different outputs, the regression results of 2007:Q4 are used. The selected probabilities shown in Figures 20 to 27 refer to the highest calculated probabilities for Southwest's presence for nonstop routes where Southwest was not present, at least during 2007:Q4.

Contrary to an LCC's aim of efficiently saving times on ground, Southwest will enter LGA, being one of U.S.'s most congested airports [36]. Within the range of the 15 nonstop connections of the highest probability, the best value only gives 30%. These low probabilities can be caused by the strong influence of the congestion variable in the model. So, even if Southwest enters LGA, it is generally concerned about airport congestion. Additionally, the presence of both *Major carriers* and other LCCs, notably JetBlue Airways (See Table 33), at LGA, may result in a strong competition at this airport.

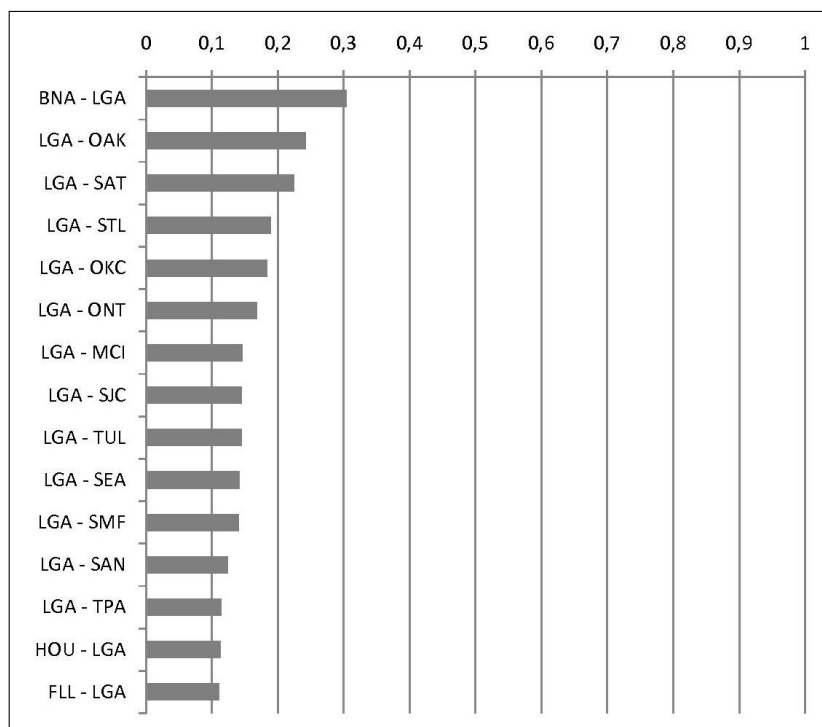


Figure 20: Fitted probabilities for routes from/to LGA.

For the routes out of the new destination MSP, the model estimates a route to BUR being the most likely, though MSP will be served only from MDW [2]. Nevertheless, the model clearly predicts that the 15 most likely services out of MSP see a reasonable higher probability than those out of LGA. Whereas the probability of the possible routes at LGA shows a fast decrease to below 20% after a short time, MSP's estimated routes give an average higher likelihood. MSP is neither a congested nor a slot restricted airport which may contribute to a better on-time performance for an airline.

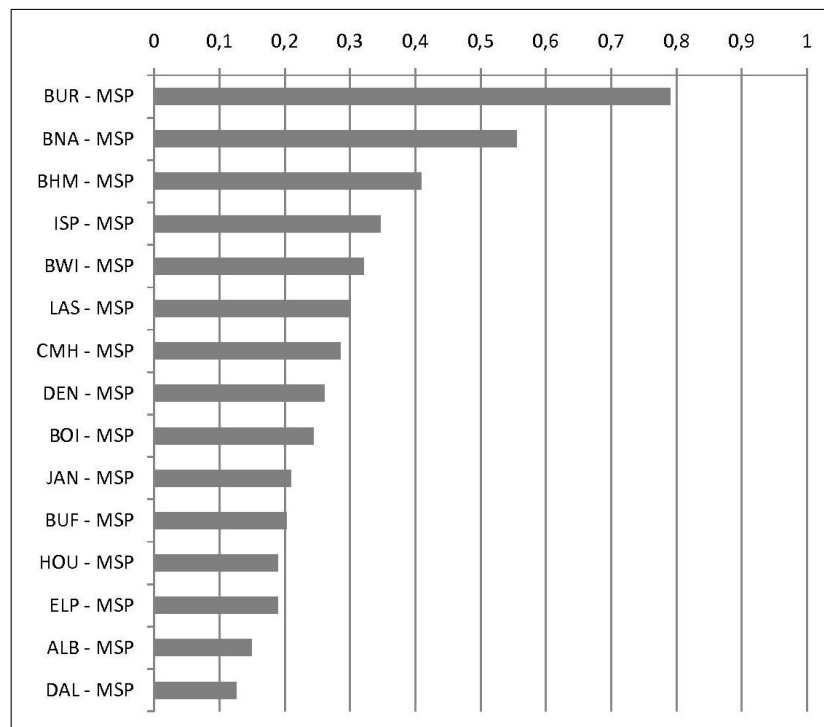


Figure 21: Fitted probabilities for routes from/to MSP.

MDW as one of Southwest's focus airports does already provide a large number of nonstop connections (see Table 2). The model estimates that missing nonstop connections to Florida (PBI and JAX) are likely to be served in the near future. Regarding the route between MDW and DAL, one effect becomes obvious: the political issue of the *Wright Amendment* cannot be reflected by the model. Nevertheless, the model estimates that there might be huge demand for nonstop connections between these two Southwest focus airports, even though they cannot be offered due to political reasons. In accordance to the announcement of Southwest airlines to increase services out of DEN, BNA and LAX, where the airline currently does not offer as many nonstop connections as out of MDW, the model predicts a set of possible nonstop connections with a high probability of presence. DEN offers nine connections with a probability of over 90% where Southwest should be present. As seen for the possible routes out of MDW in Figure 22, the model gives a high probability for services to DAL from DEN, BNA and LAX.

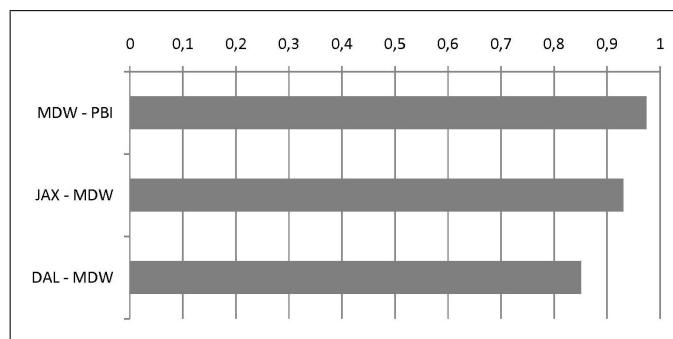


Figure 22: Fitted probabilities for routes from/to MDW.

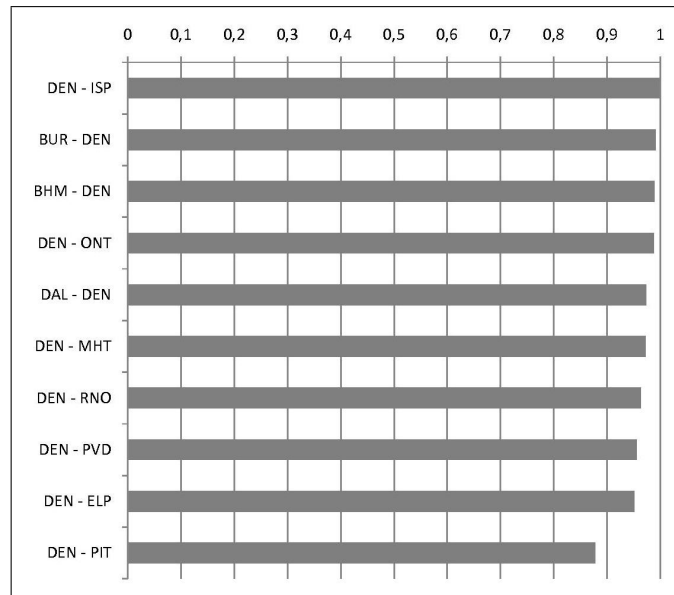


Figure 23: Fitted probabilities for routes from/to DEN.

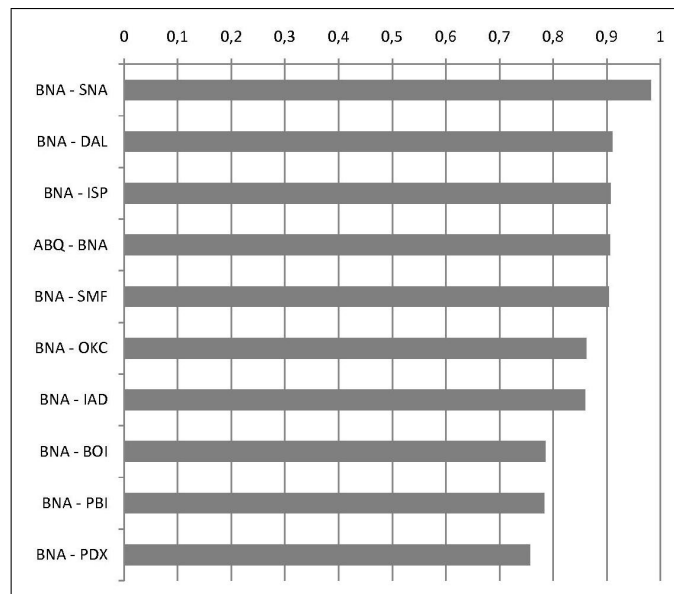


Figure 24: Fitted probabilities for routes from/to BNA.

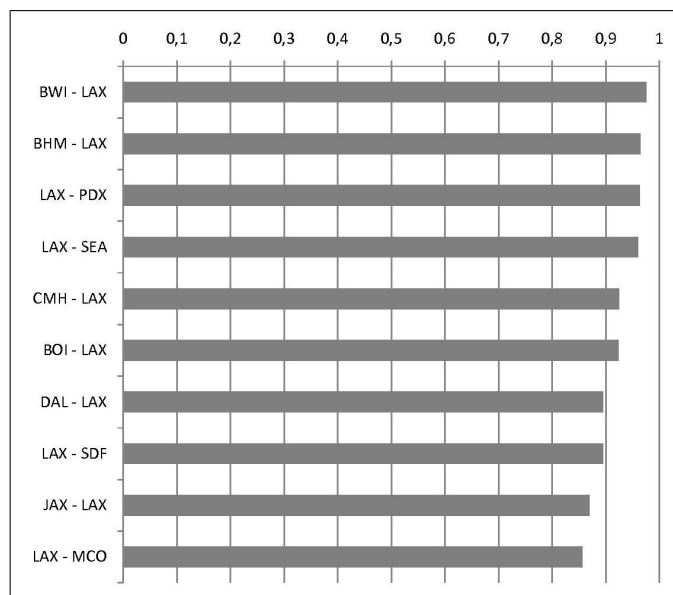


Figure 25: Fitted probabilities for routes from/to LAX.

Figures 26 and 27 serve as an example to demonstrate estimated probabilities for a route entry at common airports of the Southwest and at airports, where expansion is explicitly announced. The estimation results show for both DTW and SFO very high percentages for the first connections, but then decays very quickly. The maximum variation in probability for the announced airports within the first ten airport pairs is about 20% (for BNA), whereas this variation at the airports without significant expansion is between 30 (for DTW) and 40% (for SFO). Furthermore, by comparing the three groups of estimation results for Southwest's presence on a certain route, it stands out that the probabilities may depend on Southwest's presence at an airport. Generally speaking, once Southwest is present at an airport, either an airport like SFO or an airport like DEN, the probabilities of entering new nonstop services are significantly higher than for airports where Southwest is initially entering.

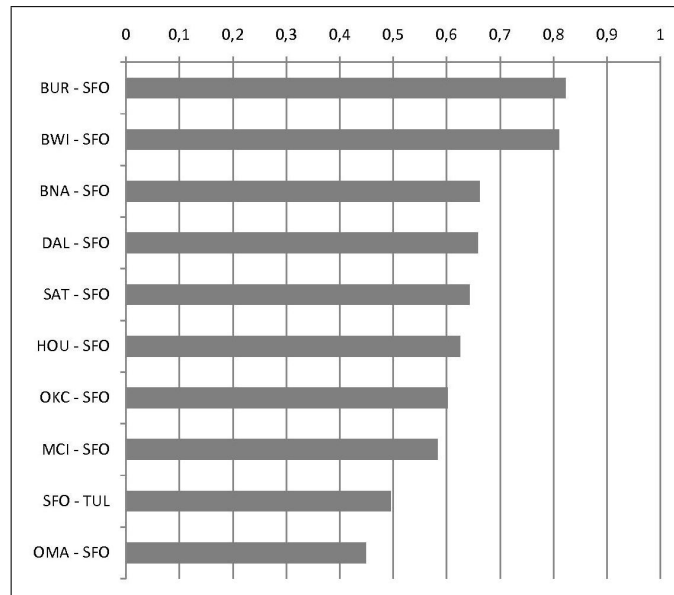


Figure 26: Fitted probabilities for routes from/to SFO.

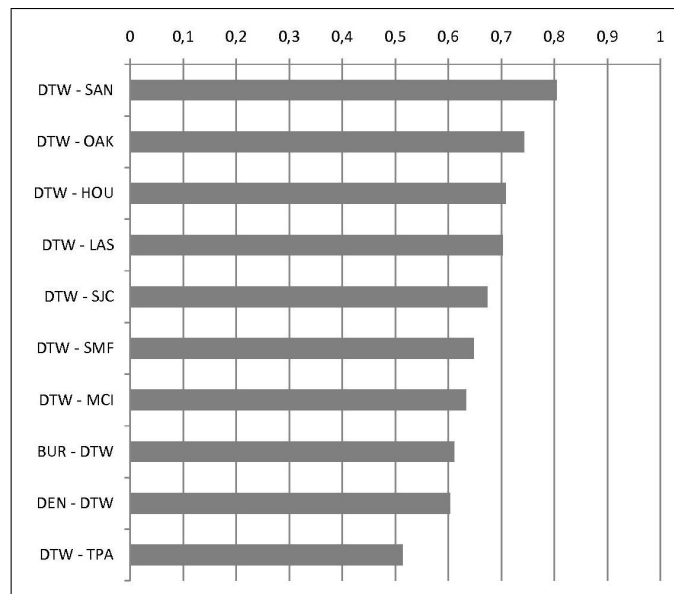


Figure 27: Fitted probabilities for routes from/to DTW.

7 Conclusion

The objective of this paper was to research which parameters are driving Southwest Airlines' presence on a given route. The first step consisted in getting some intuition, by gathering information from different sources. For example, it was likely that the presence of competitors on the route would have an impact. To get a quantified idea of the impact of some parameters, a thorough search for relevant databases is carried out in step two. The final step allows to create a model based on regressions to explain Southwest's presence.

The model is working well at explaining why Southwest is present. It is generally looking at how demand is contributing via population and GDP. It manages to capture 22% of why Southwest is present. It then includes data on the structure of the network through region and hub-specific dummy variables. This increases the model's quality to 48%. Finally, the input of new data and tweaking of the model allowed to capture about 55% of what is happening. This figure is relatively high for a presence model.

The model cannot be used to perform a dynamic forecast but it can be used for a static forecast. This gives an idea of *where too look* to study Southwest's strategy. For example, it appears Southwest became concerned by delays in the year 2003. Strong competition seems to have taken place with JetBlue in 2002. American Airlines is the major airline Southwest is avoiding the most. These are some of the results the model can give.

However, the model is not perfect and several areas remain to be explored. The first path to explore are the unexpected results the model sometimes gives. Changes could be made to improve the model's quality or some important strategy decisions may turn up. The second path would be introducing a dynamic

variable to model the entry decision, based on what competitors are doing and pricing policies. This would be both very revealing and challenging.

The final word would come to the quality of the model. Regardless of the direction this projects turns to, improving the quality of the static description or starting a new dynamic model, it is necessary to guarantee the robustness of the model before making any forecasts or explaining serious strategy decisions.

A U.S. air traffic hubs

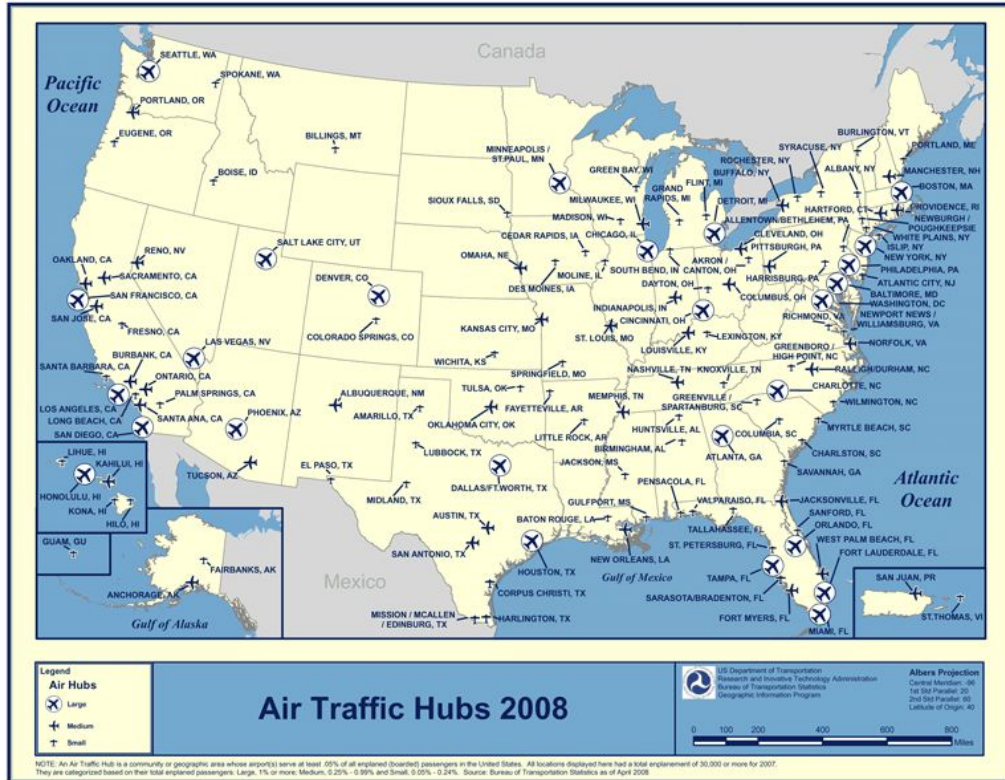


Figure 28: U.S. air traffic hubs in 2008 [16].

The classification of a hub airport is made according to its percentage of the total annual domestic passenger boardings, and enables the separation of U.S. hub airports into three categories: Large Hub: 1% or more; Medium Hub: between 0.25% and 1%; Small Hub: between 0.05% and 0.25%. [17]

B Competitors of Southwest

This section provides information about competitors of Southwest. The Figures 29, 30, 31 and 32 show the route network of Airtran Airways, Frontier Airlines, JetBlue Airways and Alaska Airlines. The tables summarized as Table 33 give an overview of the fleet and the hub structure of the *Major carriers*. The distinction between a hub city and a focus city can be made the following way: A focus city does not have the importance (e.g. regarding the offered destinations) of a hub. But from a focus city an airline has flights to destinations other than only to its hubs. For example, Northwest Airlines offers from its focus city IND flights to its hubs in MEM, DTW and MSP, but additional flights go, inter alia, to SFO and LAX, which are neither hubs nor focus cities for Northwest Airlines [28]. Regarding Southwest, this definition of hub and focus city does not hold. Southwest does not operate hubs [2], and therefore the definition of a focus city is slightly different. In the case of Southwest, a focus city means a concentration of operations, as described in the Introduction.



Figure 29: Route map from AirTran. The routes given in red show the nonstop connections out of their hub in ATL, the routes in blue show connections from other hubs. AirTran Airways operates further hubs in MCo and BWI [38].

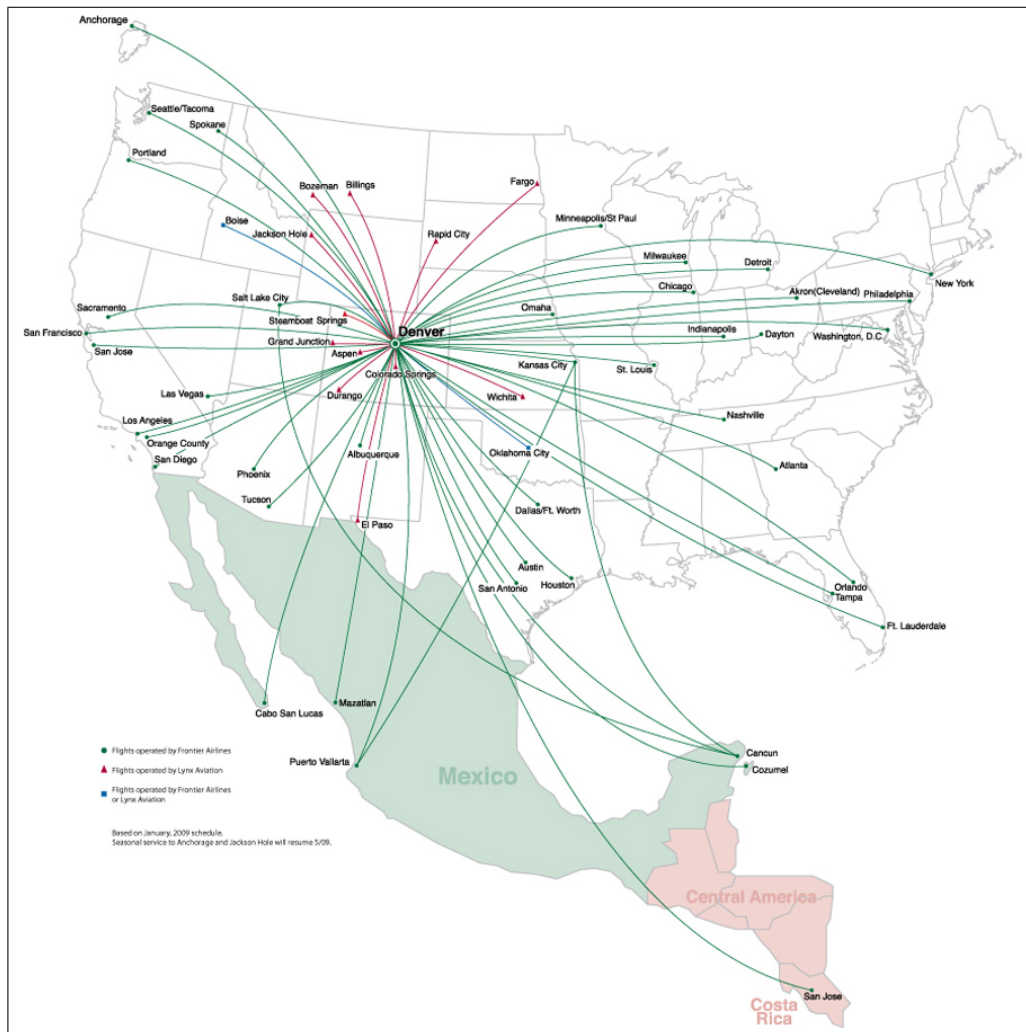


Figure 30: Route map of Frontier. The routes shown in green are routes operated by Frontier Airlines; routes shown in red are operated by its partner airline Lynx aviation; routes in blue are operated by either Frontier or Lynx Aviation [39].

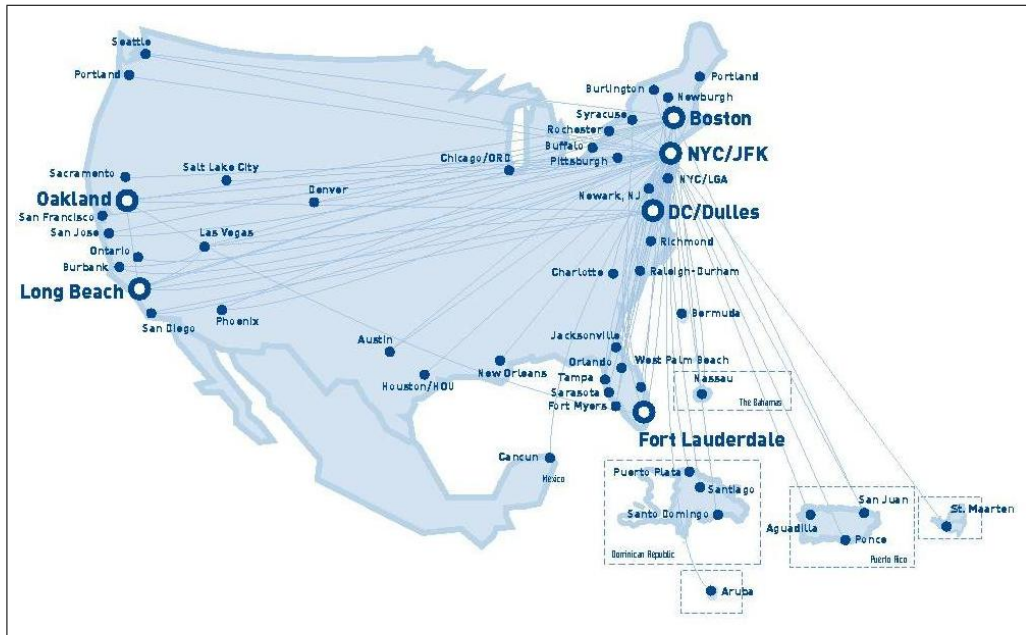


Figure 31: Route map of JetBlue Airways [40].



Figure 32: Route map of Alaska Airlines. Routes shown in blue are operated by Alaska Airlines, routes shown in brown by its partner airline Horizon Air [41].

American Airlines (AA)				
Fleet				Hubs
47	Boeing 777-200ER	Long Haul		ORD
57	Boeing 767-300ER	Long Haul		DFW
15	Boeing 767-200ER	Medium- / Long Haul		MIA
124	Boeing 757-200	Medium- / Long Haul		SJU
28	Airbus A300-600R	Short- / Medium Haul		STL
77	Boeing 737-800	Short- / Medium Haul		LAX*
283	MD-82/83	Short- / Medium Haul		JFK*
				LGA*
				BOS*

Continental Airlines (CO)				
Fleet				Hubs
20	Boeing 777-200ER	Long Haul		IAH
16	Boeing 767-400ER	Long Haul		CLE
10	Boeing 767-200ER	Medium- / Long Haul		EWB
58	Boeing 757-2/300	Medium- / Long Haul		GUM
27	Boeing 737-900ER	Medium- / Long Haul		
237	Boeing 737-3/5/7/8/900	Short- / Medium Haul		

Delta Air Lines (DL)				
Fleet				Hubs
10	Boeing 777-200ER/LR	Long Haul		ATL
21	Boeing 767-400ER	Medium- / Long Haul		CVG
76	Boeing 767-3/300ER	Medium- / Long-Haul		SLT
132	Boeing 757-200	Medium- / Long Haul		JFK
75	Boeing 737-7/800	Short- / Medium Haul		
130	MD-88/90	Short- / Medium Haul		

Northwest Airlines (NW)			
Fleet			Hubs
16	Boeing 747-400	Long Haul	MSP
33	Airbus A330-2/300	Medium-/ Long Haul	DTW
71	Boeing 757-2/300	Medium-/ Long Haul	MEM
73	Airbus A320-200	Short-/ Medium Haul	AMS
57	Airbus A319-100	Short-/ Medium Haul	NRT
79	DC 9-30/40/50	Short Haul	SEA*
			IND*

United Airlines (UA)			
Fleet			Hubs
27	Boeing 747-400	Long Haul	ORD
52	Boeing 777-2/200ER	Long Haul	IAD
35	Boeing 767-3/300ER	Medium-/ Long Haul	DEN
37	Boeing 757-200	Medium Haul	SFO
59	Boeing 737-3/500	Short-/ Medium Haul	LAX
97	Airbus A320-200	Short-/ Medium Haul	
55	Airbus A319-100	Short-/ Medium Haul	

USAirways (US)			
Fleet			Hubs
9	Airbus A330-300	Long Haul	CLT
9	Boeing 767-200ER	Long Haul	PHL
41	Boeing 757-200	Medium-/ Long Haul	PHX
78	Boeing 737-3/400	Short-/ Medium Haul	LGA*
31	Airbus A321-200	Short-/ Medium Haul	LAS*
75	Airbus A320-200	Short-/ Medium Haul	BOS*
93	Airbus A319-100	Short-/ Medium Haul	DCA*
24	Embraer 190	Short Haul	

Table 33: These six tables give information about fleet, fleet assignment and hubs of the U.S. *Major carriers*. The first two columns give the number of aircraft types in the fleet, the third column the assignment of the aircraft type and the last column the hubs and focus cities (*). For the model hubs and focus cities are not taken from these information. It is done mathematically by computing passenger ratios for each airport [28]

C Definition of GDP data

The following excerpts are definitions of the economic sectors included in the supersectors “Professional and Business Service” and “Leisure and hospitality”, whose GDP-data were used in the model. The excerpts are taken from *U.S. Department of Labour* [44] and refer to definitions given by the North American Industry classification system. They shall give an explanation about the working sectors they include.

Professional, Scientific, and Technical Services sector: “The Professional, Scientific, and Technical Services sector comprises establishments that specialize in performing professional, scientific, and technical activities for others. These activities require a high degree of expertise and training. The establishments in this sector specialize according to expertise and provide these services to clients in a variety of industries and, in some cases, to households. Activities performed include: legal advice and representation; accounting, bookkeeping, and payroll services; architectural, engineering, and specialized design services; computer services; consulting services; research services; advertising services; photographic services; translation and interpretation services; veterinary services; and other professional, scientific, and technical services.”

Management of Companies and Enterprises sector: “The Management of Companies and Enterprises sector comprises (1) establishments that hold the securities of (or other equity interests in) companies and enterprises for the purpose of owning a controlling interest or influencing management decisions or (2) establishments (except government establishments) that administer, oversee,

and manage establishments of the company or enterprise and that normally undertake the strategic or organizational planning and decision making role of the company or enterprise. Establishments that administer, oversee, and manage may hold the securities of the company or enterprise. Establishments in this sector perform essential activities that are often undertaken, in-house, by establishments in many sectors of the economy. By consolidating the performance of these activities of the enterprise at one establishment, economies of scale are achieved.”

Administrative and Support and Waste Management and Remediation

Services sector: “The Administrative and Support and Waste Management and Remediation Services sector comprises establishments performing routine support activities for the day-to-day operations of other organizations. These essential activities are often undertaken in-house by establishments in many sectors of the economy. The establishments in this sector specialize in one or more of these support activities and provide these services to clients in a variety of industries and, in some cases, to households. Activities performed include: office administration, hiring and placing of personnel, document preparation and similar clerical services, solicitation, collection, security and surveillance services, cleaning, and waste disposal services.”

Arts, Entertainment, and Recreation sector: “The Arts, Entertainment, and Recreation sector includes a wide range of establishments that operate facilities or provide services to meet varied cultural, entertainment, and recreational interests of their patrons. This sector comprises (1) establishments that are

involved in producing, promoting, or participating in live performances, events, or exhibits intended for public viewing; (2) establishments that preserve and exhibit objects and sites of historical, cultural, or educational interest; and (3) establishments that operate facilities or provide services that enable patrons to participate in recreational activities or pursue amusement, hobby, and leisure-time interests. Some establishments that provide cultural, entertainment, or recreational facilities and services are classified in other sectors.”

Accommodation and Food Services sector: “The Accommodation and Food Services sector comprises establishments providing customers with lodging and/or preparing meals, snacks, and beverages for immediate consumption. The sector includes both accommodation and food services establishments because the two activities are often combined at the same establishment.”

D Metropolitan and micropolitan Statistical Areas

This section gives an overview about assumptions made on treatment of the MSA-/mSA- and population-data given by the *U.S. Census Bureau* [37] and the *U.S. Bureau of Economic Analysis (BEA)* [42]. Assumptions regarding the treatment of the MSAs in terms of averaging and aggregating certain data have been made as follows: According to the timetable information of Southwest [2], some airports, even being located in a separate MSA, are counted as part of a bigger MSA that is in close proximity. For example, PVD is located in the Providence-New Bedford-Fall River (MSA). On Southwest’s website [2] though, this airport is referred to as being part of the “Boston Area”. Therefore, the

social and economic influence of the Boston-Cambridge-Quincy (MSA) must also be taken into account, and the data on population and GDP of both MSAs is aggregated.

- SJC with San Jose-Sunnyvale-Santa Clara (MSA) to San Francisco-Oakland-Fremont (MSA)
- PVD with Providence-New Bedford-Fall River (MSA) and MHT with Manchester-Nashua (MSA) to Boston-Cambridge-Quincy (MSA)
- BWI with Baltimore-Townsend (MSA) to Washington-Arlington-Alexandria (MSA)
- ONT with Riverside-San Bernadino-Ontario (MSA) to Los Angeles-Long Beach-Santa Ana (MSA)

Independent from the allocation in Southwest's booking system, these airports serve the following MSAs according to their assignment of the *U.S. Office of Management and Business (OMB)* [37].:

- BUR to Los Angeles-Long Beach-Santa Ana (MSA)
- PBI to Miami-Ft. Lauderdale-Pompano Beach (MSA)

The data on population and GDP of the MSAs Odessa, TX and Midland, TX have been aggregated, because the airport in this region, Odessa/Midland International Airport (MAF) serves both cities.

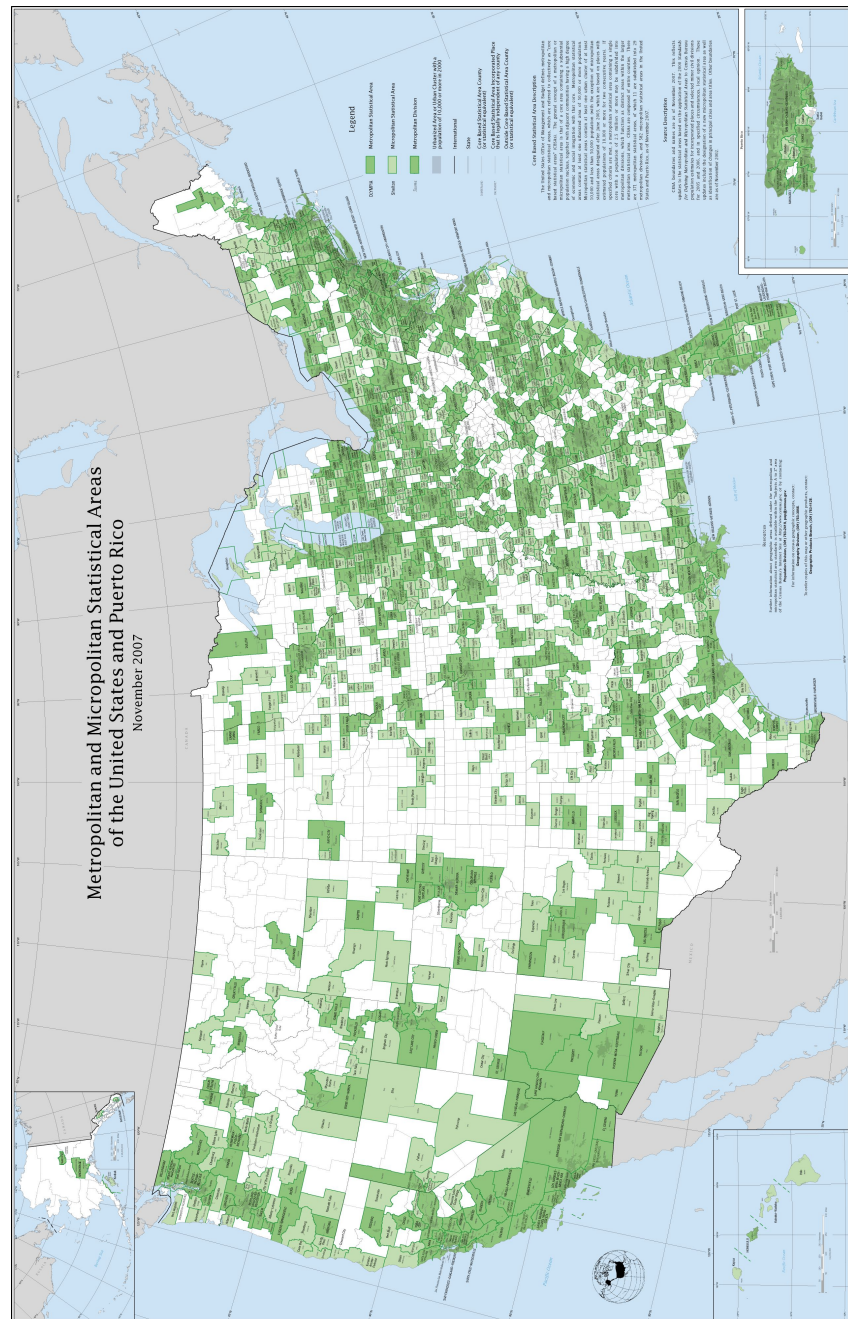


Figure 33: MSAs and mSAs 2007 [37]. This map shows the Metro (MSAs)- and micropolitan areas (mSAs) of the U.S. for the year 2007. The dark green shadowed areas mark the MSAs, the light-colored areas the mSAs.

Metro-(M)/ Micropolitan (m) Statistical Areas and airports						
Code	City / City Region	MSA/ mSA	Pop.	GDP Total	GDP Leisure	GDP Prof.
CAK	Akron, OH	MSA	700510	23561	690	3162
ALB	Albany-Schenectady-Troy, NY	MSA	840871	31930	894	3995
ABQ	Albuquerque, NM	MSA	775597	28967	932	4114
ABE	Allentown-Bethlehem-Easton, PA-NJ	MSA	774232	24556	727	2669
AOO	Altoona, PA	MSA	127264	3436	106	246
AMA	Amarillo, TX	MSA	234814	7160	243	505
ANC	Anchorage, AK	MSA	342380	18977	622	1451
ATW	Appleton, WI	MSA	211704	7944	204	590
AVL	Asheville, NC	MSA	385128	10946	596	672
AHN	Athens-Clarke County, GA	MSA	175345	5024	175	284
ATL	Atlanta-Sandy Springs-Marietta, GA	MSA	4770111	225766	7384	31010
AGS	Augusta-Richmond County, GA-SC	MSA	512434	15127	445	
AUS	Austin-Round Rock, TX	MSA	1403837	59588	2108	8263
BWI	Baltimore-Towson, MD	MSA (*)	7756242	421819	13186	77054
BTR	Baton Rouge, LA	MSA	727868	28200	869	2468
RDM	Bend, OR	MSA	133386	4798	238	370
BIL	Billings, MT	MSA	143975	5043	221	
BHM	Birmingham-Hoover, AL	MSA	1077737	45246	1129	4448
BIS	Bismarck, ND	MSA	97769	3377	116	300
BOI	Boise City-Nampa, ID	MSA	522164	19691	535	2502
BOS	Boston-Cambridge-Quincy, MA-NH	MSA (*)	6460483	320347	10302	48384
SRQ	Bradenton-Sarasota-Venice, FL	MSA	644178	20352	995	3450
BRO	Brownsville-Harlingen, TX	MSA	366148	5736	271	344
HRL	Brownsville-Harlingen, TX	MSA	366148	5736	271	344
BUF	Buffalo-Niagara Falls, NY	MSA	1152577	36779	1316	4075
BUR	Burbank	MSA (*)	16521541	672563	29741	83283
BTW	Burlington, NC	MSA	137822	3903	128	444
RSW	Cape Coral-Fort Myers, FL	MSA	509431	17477	900	1543
CPR	Casper, WY	MSA	68608	4145	86	175
CID	Cedar Rapids, IA	MSA	244319	10066	254	700
CRW	Charleston, WV	MSA	306591	11694	310	1031
CHS	Charleston-North Summerville, SC	Charleston- MSA	577620	20335	1026	1795
CLT	Charlotte-Gastonia-Concord, NC-SC	MSA	1465976	95443	2383	9556
CHO	Charlottesville, VA	MSA	184244	6950	297	628
CHA	Chattanooga, TN-GA	MSA	487539	17159	575	1528
MDW	Chicago-Naperville-Joliet, IL-IN-WI	MSA	9360590	434482	14389	64956
ORD	Chicago-Naperville-Joliet, IL-IN-WI	MSA	9360590	434482	14389	64956
CVG	Cincinnati-Middletown, OH-KY-IN	MSA	2067743	84727	3114	11994
CLE	Cleveland-Elyria-Mentor, OH	MSA	2133455	92503	2768	10498
COS	Colorado Springs, CO	MSA	576256	19493	849	2704
CAE	Columbia, SC	MSA	678791	24579	659	2176
CMH	Columbus, OH	MSA	1681647	77567	2345	10506
CRP	Corpus Christi, TX	MSA	408311	11901	489	1008
DFW	Dallas-Fort Worth-Arlington, TX	MSA	5656538	289752	9691	34150
DAL	Dallas-Fort Worth-Arlington, TX	MSA	5656538	289752	9691	34150
MLI	Davenport-Moline-Rock Island, IA-IL	MSA	375524	13528	531	1579
DAY	Dayton, OH	MSA	842976	30899	795	3620
DAB	Deltona-Daytona Beach-Ormond Beach, FL	MSA	473435	10215	658	812
DEN	Denver-Aurora, CO	MSA	2319693	121958	4413	17342
DSM	Des Moines-West Des Moines, IA	MSA	510292	27506	790	1976

continued on next page

<i>continued from previous page</i>						
Code	City / City Region	MSA/ mSA	Pop.	GDP Total	GDP Leisure	GDP Prof.
DTW	Detroit-Warren-Livonia, MI	MSA	4477072	193254	6055	33469
DLH	Duluth, MN-WI	MSA	275540	8178	296	399
ELP	El Paso, TX	MSA	709041	20074	558	1128
ERI	Erie, PA	MSA	281324	7830	231	473
EUG	Eugene-Springfield, OR	MSA	330846	9391	311	807
ACV	Eureka-Arcata-Fortuna, CA	mSA	127790			
EVV	Evansville, IN-KY	MSA	346825	13735	399	953
FAI	Fairbanks, AK	MSA	85795	3683	129	139
FAR	Fargo, ND-MN	MSA	180880	7610	284	569
FAY	Fayetteville, NC	MSA	341139	11381	253	491
XNA	Fayetteville-Springdale-Rogers, AR-MO	MSA	387370	13340	382	1976
FNT	Flint, MI	MSA	441382	12017	341	1065
FSM	Fort Smith, AR-OK	MSA	281410	8143	180	723
VPS	Fort Walton Beach-Crestview-Destin, FL	MSA	177781	7676	305	667
FWA	Fort Wayne, IN	MSA	400645	14812	420	1113
FAT	Fresno, CA	MSA	855500	22897	651	1681
GFK	Grand Forks, ND-MN	MSA	96237	2947	102	144
GJT	Grand Junction, CO	MSA	126181	3366	143	252
GRR	Grand Rapids-Wyoming, MI	MSA	763134	29658	777	3322
GTF	Great Falls, MT	MSA	79680	2244	98	148
GRB	Green Bay, WI	MSA	292898	12235	402	895
GSO	Greensboro-High Point, NC	MSA	665748	28470	740	2557
PGV	Greenville, NC	MSA	159504	4554	141	281
GSP	Greenville-Mauldin-Easley, SC	MSA	582400	21359	677	2475
GPT	Gulfport-Biloxi, MS	MSA	246638	8183	1194	561
HGR	Hagerstown-Martinsburg, MD-WV	MSA	241401	6434	236	430
MDT	Harrisburg-Carlisle, PA	MSA	517540	22984	710	2412
BDL	Hartford-West Hartford-East Hartford, CT	MSA	1176733	61345	1550	6711
SSI	Hinesville-Fort Stewart, GA	MSA	71511	2146	28	84
HNL	Honolulu, HI	MSA	894814	37263	2474	3723
HOU	Houston-Sugar Land-Baytown, TX	MSA	5184526	275892	7870	29809
IAH	Houston-Sugar Land-Baytown, TX	MSA	5184526	275892	7870	29809
HTS	Huntington-Ashland, WV-KY-OH	MSA	286179	7488	237	504
HSV	Huntsville, AL	MSA	361180	14512	363	2822
IND	Indianapolis-Carmel, IN	MSA	1608921	80293	2887	7688
ITH	Ithaca, NY	MSA	99322	3057	97	194
JAN	Jackson, MS	MSA	513275	18840	524	1752
JAX	Jacksonville, FL	MSA	1210647	47765	1887	4820
OAJ	Jacksonville, NC	MSA	150701	4578	109	150
AZO	Kalamazoo-Portage, MI	MSA	318263	10392	331	741
MCI	Kansas City, MO-KS	MSA	1916287	84679	3053	10621
PSC	Kennewick-Pasco-Richland, WA	MSA	212117	6732	212	2218
TRI	Kingsport-Bristol-Bristol, TN-VA	MSA	299981	7968	270	620
TYS	Knoxville, TN	MSA	644519	24099	944	2966
LSE	La Crosse, WI-MN	MSA	128321	4384	144	326
LFT	Lafayette, LA	MSA	245236	11454	348	990
LAN	Lansing-East Lansing, MI	MSA	453314	16196	422	
LAS	Las Vegas-Paradise, NV	MSA	1613897	70686	14062	7203
LEX	Lexington-Fayette, KY	MSA	423278	18815	686	2068
LNK	Lincoln, NE	MSA	277259	11115	345	928

continued on next page

<i>continued from previous page</i>						
Code	City / City Region	MSA/ mSA	Pop.	GDP Total	GDP Leisure	GDP Prof.
LIT	Little Rock-North Little Rock-Conway, AR	MSA	632835	24815	660	2174
LAX	Los Angeles-Long Beach-Santa Ana, CA	MSA(*)	16521541	672563	29741	83283
LGB	Los Angeles-Long Beach-Santa Ana, CA	MSA(*)	16521541	672563	29741	83283
SDF	Louisville-Jefferson County, KY-IN	MSA	1196150	47566	1838	4417
LBB	Lubbock, TX	MSA	256936	7347	346	456
LYH	Lynchburg, VA	MSA	233289	6756	162	419
MSN	Madison, WI	MSA	527442	26439	701	
MHT	Manchester-Nashua, NH	MSA(*)	6460483	320347	10302	48384
MFE	McAllen-Edinburg-Mission, TX	MSA	646199	9832	363	552
MFR	Medford, OR	MSA	191132	5277	222	426
MEM	Memphis, TN-MS-AR	MSA	1241918	53704	2835	5831
MIA	Miami-Fort Lauderdale-Pompano Beach, FL	MSA(*)	5307403	208807	9639	26129
FLL	Miami-Fort Lauderdale-Pompano Beach, FL	MSA(*)	5307403	208807	9639	26129
PBI	Miami-Fort Lauderdale-Pompano Beach, FL	MSA(*)	5307403	208807	9639	26129
MKE	Milwaukee-Waukesha-West Allis, WI	MSA	1509105	69590	1884	8100
MSP	Minneapolis-St. Paul-Bloomington, MN-WI	MSA	3098239	160343	4948	
MSO	Missoula, MT	MSA	98887	3606	152	296
MOB	Mobile, AL	MSA	400209	11791	344	1068
MLU	Monroe, LA	MSA	170425	5281	153	503
MGM	Montgomery, AL	MSA	353595	12048	332	1078
MGW	Morgantown, WV	MSA	113489	3879	114	246
MKG	Muskegon-Norton Shores, MI	MSA	173432	4282	152	206
MYR	Myrtle Beach-North Myrtle Beach-Conway, SC	MSA	216940	7676	1038	472
APF	Naples-Marco Island, FL	MSA	291001	11782	1000	1128
BNA	Nashville-Davidson-Murfreesboro-Franklin, TN	MSA	1388158	62756	3063	7310
MSY	New Orleans-Metairie-Kenner, LA	MSA	1264545	57216	3367	5356
EWR	New York-Northern New Jersey-Long Island, NY-NJ-PA	MSA	18702387	985633	30837	134533
ISP	New York-Northern New Jersey-Long Island, NY-NJ-PA	MSA	18702387	985633	30837	134533
JFK	New York-Northern New Jersey-Long Island, NY-NJ-PA	MSA	18702387	985633	30837	134533
LGA	New York-Northern New Jersey-Long Island, NY-NJ-PA	MSA	18702387	985633	30837	134533
HPN	New York-Northern New Jersey-Long Island, NY-NJ-PA	MSA	18702387	985633	30837	134533
GON	Norwich-New London, CT	MSA	263563	10623	460	928
MAF	Midland-Odessa, TX	MSA	243487	9809	265	444
OKC	Oklahoma City, OK	MSA	1137612	42198	1577	4615
OMA	Omaha-Council Bluffs, NE-IA	MSA	797898	36004	1143	4223
MCO	Orlando-Kissimmee, FL	MSA	1840022	79033	8118	10374
MLB	Palm Bay-Melbourne-Titusville, FL	MSA	511293	14153	541	2616
PSP	Palm Springs	MSA(*)	16521541	672563	29741	83283
PFN	Panama City-Lynn Haven, FL	MSA	156535	4924	309	417
PKB	Parkersburg-Marietta-Vienna, WV-OH	MSA	162874	4713	145	268

continued on next page

<i>continued from previous page</i>						
Code	City / City Region	MSA/ mSA	Pop.	GDP Total	GDP Leisure	GDP Prof.
PNS	Pensacola-Ferry Pass-Brent, FL	MSA	430820	10972	403	1104
PIA	Peoria, IL	MSA	367146	12993	425	1283
PHL	Philadelphia-Camden-Wilmington, PA- NJ-DE-MD	MSA	5772326	275015	7083	40189
PHX	Phoenix-Mesa-Scottsdale, AZ	MSA	3686339	147160	6051	18221
PIT	Pittsburgh, PA	MSA	2397683	95125	2916	11554
LBE	Pittsburgh, PA	MSA	2397683	95125	2916	11554
PWM	Portland-South Portland-Biddeford, ME	MSA	506928	20512	841	1854
PDX	Portland-Vancouver-Beaverton, OR-WA	MSA	2054637	87458	2590	9892
SWF	Poughkeepsie-Newburgh-Middletown, NY	MSA	655800	17315	559	1421
PVD	Providence-New Bedford-Fall River, RI- MA	MSA(*)	6460483	320347	10302	48384
RDU	Raleigh-Cary, NC	MSA	907658	40367	1153	5586
RAP	Rapid City, SD	MSA	116571	3928	206	220
RDG	Reading, PA	MSA	388984	11977	329	1305
RNO	Reno-Sparks, NV	MSA	378951	16449	1853	1743
RIC	Richmond, VA	MSA	1149500	51124	1243	6968
ONT	Riverside-San Bernardino-Ontario, CA	MSA(*)	16521541	672563	29741	83283
ROA	Roanoke, VA	MSA	290999	10668	317	1098
RST	Rochester, MN	MSA	173350	7262	232	383
ROC	Rochester, NY	MSA	1038437	38904	1032	4334
SMF	Sacramento-Arden-Arcade-Roseville, CA	MSA	1981842	76524	2425	7737
MBS	Saginaw-Saginaw Township North, MI	MSA	208562	6483	209	503
SGU	St. George, UT	MSA	109117	2454	143	145
MRY	Salinas, CA	MSA	412074	15309	950	1045
SBY	Salisbury, MD	MSA	113733	3091	95	281
SLC	Salt Lake City, UT	MSA	1021508	47168	1678	5965
SAT	San Antonio, TX	MSA	1836963	61024	2821	6597
SAN	San Diego-Carlsbad-San Marcos, CA	MSA	2916686	133995	5484	17817
SFO	San Francisco-Oakland-Fremont, CA	MSA(*)	5916954	371893	11905	65560
OAK	San Francisco-Oakland-Fremont, CA	MSA(*)	5916954	371893	11905	65560
SJC	San Jose-Sunnyvale-Santa Clara, CA	MSA(*)	5916954	371893	11905	65560
SBP	San Luis Obispo-Paso Robles, CA	MSA	253785	8434	425	653
SBA	Santa Barbara-Santa Maria-Goleta, CA	MSA	401441	15831	805	1869
SNA	Los Angeles-Long Beach-Santa Ana, CA	MSA(*)	16521541	672563	29741	83283
SAV	Savannah, GA	MSA	307579	10053	492	840
AVP	Scranton-Wilkes-Barre, PA	MSA	552244	16094	481	1114
SEA	Seattle-Tacoma-Bellevue, WA	MSA	3167553	171680	5651	19286
SHV	Shreveport-Bossier City, LA	MSA	379792	15498	950	822
SUX	Sioux City, IA-NE-SD	MSA	142951	5001	198	470
FSD	Sioux Falls, SD	MSA	201537	11492	311	527
SBN	South Bend-Mishawaka, IN-MI	MSA	317373	10327	276	868
GEG	Spokane, WA	MSA	433734	13862	549	1212
SGF	Springfield, MO	MSA	388648	11739	387	1036
TPA	Tampa-St. Petersburg-Clearwater, FL	MSA	2564448	91443	4157	13753
TOL	Toledo, OH	MSA	657180	23725	699	2130
TUS	Tucson, AZ	MSA	901018	25103	1133	2508
TUL	Tulsa, OK	MSA	881016	34615	929	3942
PHF	Virginia Beach-Norfolk-Newport News, VA-NC	MSA	1623238	61328	2042	5579

continued on next page

continued from previous page						
Code	City / City Region	MSA/ mSA	Pop.	GDP Total	GDP Leisure	GDP Prof.
ORF	Virginia Beach-Norfolk-Newport News, VA-NC	MSA	1623238	61328	2042	5579
IAD	Washington-Arlington-Alexandria, DC-VA-MD-WV	MSA (*)	7756242	421819	13186	77054
DCA	Washington-Arlington-Alexandria, DC-VA-MD-WV	MSA (*)	7756242	421819	13186	77054
ART	Watertown-Fort Drum, NY	mSA	113928			
CWA	Wausau, WI	MSA	127894	4733	105	259
ICT	Wichita, KS	MSA	583082	21663	690	1492
ILM	Wilmington, NC	MSA	300928	10622	441	1018
ASE	Edwards, CO	mSA	53780			
HDN	Edwards, CO	mSA	53780			
EGE	Edwards, CO	mSA	53780			
bfd	Bradford, PA	mSA	44783			
PQI	Bangor, ME	MSA	146597	4535	138	323
SCE	State College, PA	MSA	139313	4400	157	349
STL	St. Louis, MO-IL	MSA	2758750	108636	4164	15653
SYR	Syracuse	MSA	651038	22187	657	2172
TLH	Tallahassee, FL	MSA	328911	10568	296	1117
SJU	Puerto Rico	mSA	3994259			
STT	US Virgin Islands	mSA	108448			
STX	US Virgin Islands	mSA	108448			
BQN	Puerto Rico	mSA	3994259			
PSE	Puerto Rico	mSA	3994259			
PLN	Alpena, MI	mSA	30673			
LWB	Beckley, WV	mSA	79079			
BZN	Bozeman, MT	mSA	74826			
CKB	Clarksburg, WV	mSA	91871			
DDC	Dodge City, KS	mSA	33164			
DUJ	DuBois, PA	mSA	82903			
DRO	Durango, CO	mSA	46425			
GNV	Gainesville, TX	mSA	38180			
GCK	Garden City, KS	mSA	39359			
HLN	Helena, MT	mSA	68305			
KOA	Hilo, HI	mSA	160523			
ITO	Hilo, HI	mSA	160523			
HHH	Hilton Head Island-Beaufort, SC	mSA	153983			
JAC	Jackson, WY-ID	mSA	25983			
COD	Jackson, WY-ID	mSA	25983			
JHW	Jamestown, ND	mSA	21079			
OGG	Kahului-Wailuku, HI	mSA	136516			
FCA	Kalispell, MT	mSA	80395			
LIH	Kapaa, HI	mSA	61079			
EYW	Key West-Marathon, FL	mSA	77639			
MHK	Manhattan, KS	mSA	106568			
MOT	Minot, ND	mSA	64477			
MTJ	Montrose, CO	mSA	36377			
EWN	New Bern, NC	mSA	116461			
MSS	Ogdensburg-Massena, NY	mSA	111273			
SOP	Southern Pines-Pinehurst, NC	mSA	79551			
TVC	Traverse City, MI	mSA	138599			
GUC	Canon City, CO	mSA	47507			

<i>continued from previous page</i>						
Code	City / City Region	MSA/ mSA	Pop.	GDP Total	GDP Leisure	GDP Prof.
ISN	Williston, ND	mSA	19375			
The first column gives the code of an airport, the second column the respective MSA or mSA. Entries marked with (*) mean that Statistical Areas are aggregated, as described in the introduction of this Appendix. The fourth column gives the population of the areas, the last three columns give information about the GDP in millions of US Dollars. The values of Population and GDP are the geometric means of the data for 2001 to 2006 [37] [42].						

Table 34: MSAs and mSAs and airports [37] [42].

E EViews source code

```

' Updated on 19th January by Jon
' Code rewritten to keep track of possible dummy variables included
' in the data but excluded from the modelling. The number of
' variables imported has to be explicited, same goes for the
' number of variables excluded. Changed the plotted coefficients.
' Now not the betas but the marginal effects for probit.
*****

' The next line creates a workfile, necessary to start working.
wfcreate u 27966

' Declaration of variables
scalar nbimp      'number of variables imported.
'Has to be updated with new variables.
'Currently, there are fifty-nine
'variables
'in the model.
scalar nbvar 'number of variables in the model.
'INCLUDING constant.
scalar nb_case  'number of cells in the matrix for one
'quarter
scalar length 'length of the matrix = number of rows

```



```

scalar excluded 'number of variables excluded in the
'regression.
nbimp = 77
nb_case = 7
length = nb_case*6*4
excluded = 39
'Variables excluded are : pax_nb, reg15, paxhub1,
'wright slot aa_paxhub1 aa_paxhub2 aa_paxhub3 aa_paxhub4 us_paxhub1
'us_paxhub2 us_paxhub3 us_paxhub4 ua_paxhub1 ua_paxhub2 ua_paxhub3
'ua_paxhub4 nw_paxhub1 nw_paxhub2 nw_paxhub3 nw_paxhub4 dl_paxhub1
'dl_paxhub2 dl_paxhub3 dl_paxhub4 co_paxhub1 co_paxhub2 co_paxhub3
'co_paxhub4 hub150 route50 route100 endpoint50 endpoint100 n_
'dep_delay arr_delay dist_ratio ops

'the order beneath is to compute number of variables in the regression:
'nbvar = nbimp + quadratic terms - year&quarter + dist<=200 - excluded
'variables
'nbvar = nbimp + 3 - 2 + 1 - excluded

'We create a matrix to store all the regression results
matrix(length,nbvar) results

'This next step opens the data files, then runs the regression
'and finally stores the result in the matrix.

```

```

*****

for !i = 7 to 7

for !j = 4 to 4

'Opens the data files. Has to be updated with new
'variables. Dropstrings excludes all cells that
'include letters.

read(t=txt,dropstrings) G:\Southwest\data_7\
WN_actual_routes_200!i_!j_.txt year quarter pax_nb
dist carrier_nb population leisure_ind business_ind
reg1 reg2 reg3 reg4 reg5 reg6 reg7 reg8 reg9 reg10
reg11 reg12 reg13 reg14 reg15 paxhub1 paxhub2 paxhub3
paxhub4 wright slot aa_paxhub1 aa_paxhub2 aa_paxhub3
aa_paxhub4 us_paxhub1 us_paxhub2 us_paxhub3 us_paxhub4
ua_paxhub1 ua_paxhub2 ua_paxhub3 ua_paxhub4 nw_paxhub1
nw_paxhub2 nw_paxhub3 nw_paxhub4 dl_paxhub1 dl_paxhub2
dl_paxhub3 dl_paxhub4 co_paxhub1 co_paxhub2 co_paxhub3
co_paxhub4 hub50 hub100 hub150 route50 route100 endpoint50
endpoint100 n_ aa_hub us_hub ua_hub nw_hub dl_hub co_hub
dep_delay arr_delay max_delay dist_ratio jetblue airtran
frontier alaska ops wn_presence

'77 variables imported

```

'Before scaling of dist, is test if distance is shorter
'than 200km

series dist_t = dist<=200

'In this part, we create the series containing the
'variables and scale them.

dist = dist/1000

population = population/1000000

'Trick to compute number of competitors, nb of competitors =
'total carriers - southwest's presence

series competitor_nb

competitor_nb = (carrier_nb - wn_presence)

'Series with the number 2 on the end are used for
'quadratic terms.

series dist2 = dist^2

series population2 = population^2

series competitor_nb2 = competitor_nb^2

'Series that corrects the endpoints by excluding hubs

endpoint50 = endpoint50 - hub50

endpoint100 = endpoint100 - hub100

'Auxiliary regressions

equation eq_aux1.ls competitor_nb c population

series comp_star = resid

series comp_star2_temp = comp_star^2

equation eq_aux2.ls comp_star2_temp c population population2

series comp_star2 = resid

'Models have to be updated with new variables

equation eq_probit.binary(d=n,q,h) wn_presence c dist dist2

population population2 comp_star comp_star2 leisure_ind

business_ind reg1 reg2 reg3 reg4 reg5 reg6 reg7 reg8 reg9

reg10 reg11 reg12 reg13 reg14 paxhub2 paxhub3 paxhub4 aa_hub

us_hub ua_hub nw_hub dl_hub co_hub dist_t hub50 hub100

max_delay jetblue airtran frontier alaska

'40 variables in the model,

'including constant, wn_presence is omitted as it is not a variable

```
equation eq_logit.binary(d=l,q,h)  wn_presence c dist dist2
population population2 comp_star comp_star2 leisure_ind
business_ind reg1 reg2 reg3 reg4 reg5 reg6 reg7 reg8 reg9
reg10 reg11 reg12 reg13 reg14 paxhub2 paxhub3 paxhub4 aa_hub
us_hub ua_hub nw_hub dl_hub co_hub dist_t hub50 hub100 max_delay
jetblue airtran frontier alaska
```

```
'Test for endogeneity. Has to be updated with variables in the model.
vector(nbvar-1) r2_results
```

```
equation eq01.ls dist c wn_presence
equation eq02.ls dist2 c wn_presence
equation eq03.ls population c wn_presence
equation eq04.ls population2 c wn_presence
equation eq05.ls comp_star c wn_presence
equation eq06.ls comp_star2 c wn_presence
equation eq07.ls leisure_ind c wn_presence
equation eq08.ls business_ind c wn_presence
```

```
equation eq09.ls reg1 c wn_presence
equation eq10.ls reg2 c wn_presence
equation eq11.ls reg3 c wn_presence
equation eq12.ls reg4 c wn_presence
```

equation eq13.ls reg5 c wn_presence

equation eq14.ls reg6 c wn_presence

equation eq15.ls reg7 c wn_presence

equation eq16.ls reg8 c wn_presence

equation eq17.ls reg9 c wn_presence

equation eq18.ls reg10 c wn_presence

equation eq19.ls reg11 c wn_presence

equation eq20.ls reg12 c wn_presence

equation eq21.ls reg13 c wn_presence

equation eq22.ls reg14 c wn_presence

equation eq23.ls paxhub2 c wn_presence

equation eq24.ls paxhub3 c wn_presence

equation eq25.ls paxhub4 c wn_presence

equation eq26.ls aa_hub c wn_presence

equation eq27.ls us_hub c wn_presence

equation eq28.ls ua_hub c wn_presence

equation eq29.ls nw_hub c wn_presence

equation eq30.ls dl_hub c wn_presence

equation eq31.ls co_hub c wn_presence

equation eq32.ls dist_t c wn_presence

```
equation eq33.ls hub50 c wn_presence
```

```
equation eq34.ls hub100 c wn_presence
```

```
equation eq35.ls max_delay c wn_presence
```

```
equation eq36.ls jetblue c wn_presence
```

```
equation eq37.ls airtran c wn_presence
```

```
equation eq38.ls frontier c wn_presence
```

```
equation eq39.ls alaska c wn_presence
```

```
'We store the r2 of each equation in the vector r2_results
```

```
for !k = 1 to nbvar-1
```

```
if !k<10 then r2_results(!k) = eq0!k.@r2
```

```
else r2_results(!k) = eq!k.@r2
```

```
endif
```

```
next
```

```
'Declaration of the regression results. They are stored in a
```

```
'list of coefficients.
```

```
coef beta_probit = eq_probit.@coefs
```

```
coef beta_logit = eq_logit.@coefs
```

```
coef se_probit = eq_probit.@stderrs
```

```
coef signif_probit = eq_probit.@tstats

'Getting probabilities instead of tstats.
coef(nbvar-1) probs
for !k = 2 to nbvar
probs(!k-1) = 2*(1 - @cnorm(@abs(signif_probit(!k))))
next

'Compute the mean of each series. Needed to compute derivatives
'on logit and probit. Has to be updated with new variables with
'same order as regression.
vector(nbvar) mean_values
mean_values(1) = @mean(c)
mean_values(2) = @mean(dist)
mean_values(3) = @mean(dist2)
mean_values(4) = @mean(population)
mean_values(5) = @mean(population2)
mean_values(6) = @mean(comp_star)
mean_values(7) = @mean(comp_star2)
mean_values(8) = @mean(leisure_ind)
mean_values(9) = @mean(business_ind)

mean_values(10) = @mean(reg1)
mean_values(11) = @mean(reg2)
```



```
mean_values(12) = @mean(reg3)
mean_values(13) = @mean(reg4)
mean_values(14) = @mean(reg5)
mean_values(15) = @mean(reg6)
mean_values(16) = @mean(reg7)
mean_values(17) = @mean(reg8)
mean_values(18) = @mean(reg9)
mean_values(19) = @mean(reg10)
mean_values(20) = @mean(reg11)
mean_values(21) = @mean(reg12)
mean_values(22) = @mean(reg13)
mean_values(23) = @mean(reg14)

mean_values(24) = @mean(paxhub2)
mean_values(25) = @mean(paxhub3)
mean_values(26) = @mean(paxhub4)
mean_values(27) = @mean(aa_hub)
mean_values(28) = @mean(us_hub)
mean_values(29) = @mean(ua_hub)
mean_values(30) = @mean(nw_hub)
mean_values(31) = @mean(dl_hub)
mean_values(32) = @mean(co_hub)

mean_values(33) = @mean(dist_t)
mean_values(34) = @mean(hub50)
```

```
mean_values(35) = @mean(hub100)
```

```
mean_values(36) = @mean(max_delay)
```

```
mean_values(37) = @mean(jetblue)
```

```
mean_values(38) = @mean(airtran)
```

```
mean_values(39) = @mean(frontier)
```

```
mean_values(40) = @mean(alaska)
```

```
'This next loop is used to place the regression results in the matrix.
```

```
for !k = 2 to nbvar
```

```
results(4*nb_case*(!i-2) + nb_case*(!j-1) + 1,!k-1)  
    = beta_probit(!k)
```

```
results(4*nb_case*(!i-2) + nb_case*(!j-1) + 2,!k-1)  
    = se_probit(!k)
```

```
results(4*nb_case*(!i-2) + nb_case*(!j-1) + 3,!k-1)  
    = probs(!k-1)
```

```
results(4*nb_case*(!i-2) + nb_case*(!j-1) + 4,!k-1)  
    = @cnorm((@transpose(mean_values)*beta_probit)(1))
```

```

*beta_probit(!k)

results(4*nb_case*(!i-2) + nb_case*(!j-1) + 5,!k-1)
  = @clogistic((@transpose(mean_values)*beta_logit)(1))
*(1-@clogistic((@transpose(mean_values)*beta_logit)(1)))
*beta_probit(!k)

results(4*nb_case*(!i-2) + nb_case*(!j-1) + 6,!k-1)
  = @exp((@transpose(mean_values)*beta_logit)(1))
*@exp(beta_logit(!k))

results(4*nb_case*(!i-2) + nb_case*(!j-1) + 7,!k-1)
  = r2_results(!k-1)
next

results(4*nb_case*(!i-2) + nb_case*(!j-1) + 1,nbvar)
  = eq_probit.@r2

results(4*nb_case*(!i-2) + nb_case*(!j-1) + 2,nbvar)
  = @exp((@transpose(mean_values)*beta_logit)(1))

'Plotting a desired variable
'The following loop allows to create a matrix that

```

```
'contains all the regression coefficients in easy-to-use
'form for plotting.
```

```
matrix(24,nbvar-1) coeff_plot
```

```
for !k = 2 to nbvar
coeff_plot(4*(!i-2) + !j,!k-1)
    = @cnorm((@transpose(mean_values)
*beta_probit)(1))*beta_probit(!k)
next
```

```
next
```

```
next
```

```
*****
```

```
'At this stage, the matrix with all the regression results
'is created. The nextstep is to create the excel file which
'will contain all these numbers.To do this, you first have to convert
'the matrix into a group.This is done with the mtos command. Adding
'the sample size is a trick
'to make the size of the matrix and the group match.
```

```
sample s_export 1 168 '168 = nb_case*6*4
mtos(results, export_results, s_export)
```

'Declaration of a series containing letters. EvIEWS calls it an
'alpha series. This code adds a column containing 200X:QY.

```
alpha(length) year_quarter
```

```
for !i = 2 to 7  
for !j = 1 to 4  
year_quarter(4*nb_case*(!i-2) +  
nb_case*(!j-1) + 1)  
= "200"+@str(!i)+":Q"+@str(!j)  
next  
next
```

'We add 200X:QY in the group just before writing the excel file.
export_results.add year_quarter

```
write G:\temp\results_model9.xls export_results
```

'In this next step, we clear some of the objects in the workfile that
'are not really needed.

```
for !k = 1 to nbvar-1
```

```
if !k<10 then delete ser0!k eq0!k
else delete ser!k eq!k
endif
next
```

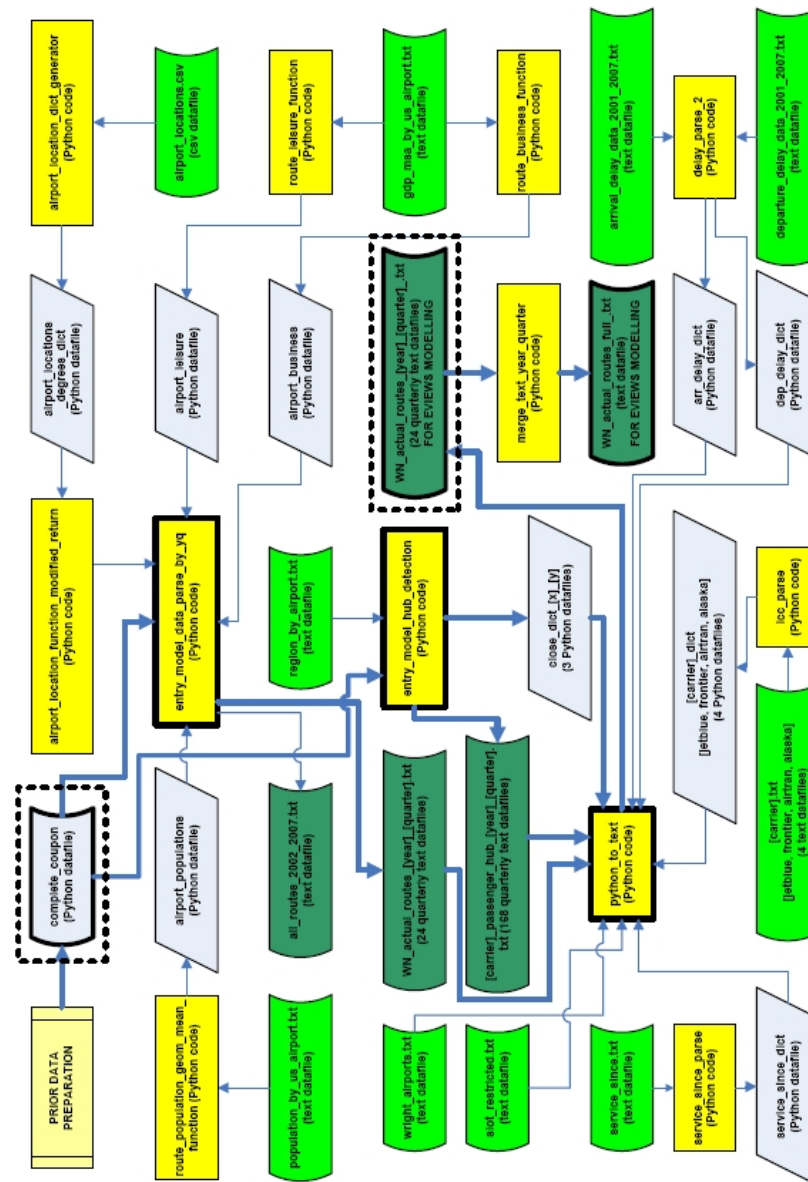


Figure 34: Python Flowchart

```
# entry_model_data_parse_by_yq
```

```
"""
```

Parse complete_coupon dataset for WN entry model project.

Steve Lawford: 10 November 2008.

Multiple modifications 14 November 2008. Hardcoding remains.

Modified 19 November 2008 for WN and 6 main carriers.

Modified 28 November 2008 with addition of nautical mile distance.

Modified 29 November 2008 with 0-1 routes, c.f. 'all large carrier static network'.

Modified 30 November 2008 with number of carriers on all actual routes.

Modified 3 December 2008 with .txt output of full airport list.

Modified 13 December 2008 with route population (geometric mean of endpoints).

Modified 19 December 2008 with leisure indicator function call.

Modified 19 December 2008 with business indicator function call.

Modified 19 January 2009 with manual CHI, NYC IATA code error traps (to change).

Modified 20 January 2009 with automatic passenger filter.

```
"""
```

```
import cPickle,time,os,sys
```



```
module_dir=os.path.join('Project','code')
sys.path.insert(0,module_dir)

from airport_location_function_modified_return import *
from safe_cPickle import *
from scipy import *
from route_population_geom_mean_function import *
from route_leisure_function import leisure_function
from route_business_function import business_function

data_path='C:/Project/data/'; filename='complete_coupon'

passenger_filter_flag=True

year_list=[2002,2003,2004,2005,2006,2007]; quarter_list=[1,2,3,4]
#year_list=[2007]; quarter_list=[4]

leisure_dict=leisure_function(0.05)
business_dict=business_function(0.10)

t01=time.time(); print; print 'Loading dictionary: '+filename
c_x_x=open(data_path+filename,'r'); dt=cPickle.load(c_x_x)
print str(time.time()-t01)+' seconds to load dictionary.'
c_x_x.close()
```

```

# the following CHI-removal (from complete_coupon) error trap is not generic
aa=dt['Origin'].index('CHI')
for jj in dt.keys(): dt[jj].pop(aa)

# the following NYC-removal (from complete_coupon) error trap is not generic
aa=dt['Origin'].index('NYC')
for jj in dt.keys(): dt[jj].pop(aa)

r_sphere=6372.7974775959065 # km converted to miles and nautical miles below
carrier='WN'; print; print 'Carrier '+carrier

def distance_calculator(dest_coords_degrees,origin_coords_degrees):

    dest_coords_radians=[(pi/180)*dest_coords_degrees[0],\
                          (pi/180)*dest_coords_degrees[1]]
    origin_coords_radians=[(pi/180)*origin_coords_degrees[0],\
                          (pi/180)*origin_coords_degrees[1]]
    p1,l1=dest_coords_radians[0],dest_coords_radians[1]
    p2,l2=origin_coords_radians[0],origin_coords_radians[1]
    l=abs(l1-l2)
    num=sqrt(((cos(p2)*sin(l))**2)+(((cos(p1)*sin(p2))-(sin(p1)*cos(p2)*cos(l)))**2))
    den=sin(p1)*sin(p2)+cos(p1)*cos(p2)*cos(l)
    theta=arctan(num/den)
    distance=r_sphere*theta

```

```
        return distance

t02=time.time(); print; print 'Parsing data: '+filename

full_data_dict={}
for i in year_list:
    for j in quarter_list:
        key=str(i)+'_'+str(j)
        full_data_dict[key]=[]

airport_list_all=[]; competitor_dict_by_route={}
for i in range(381526):

    if (float(i)/3815)==i/3815: print str(i/3815)+'%'

    try:

        key_i=str(dt['Year'][i])+'_'+str(dt['Quarter'][i])
        dest_name=airport_location(dt['Dest'][i])[0]
        origin_name=airport_location(dt['Origin'][i])[0]

        if dt['Dest'][i] not in airport_list_all:
            airport_list_all.append(dt['Dest'][i])
        if dt['Origin'][i] not in airport_list_all:
            airport_list_all.append(dt['Origin'][i])
```

```

dest_coords_degrees=airport_location(dt['Dest'][i])[1]
origin_coords_degrees=airport_location(dt['Origin'][i])[1]

distance=distance_calculator(dest_coords_degrees,origin_coords_degrees)
distance=abs(distance)

# see distance error trap comment below

route_list=[dt['Dest'][i],dt['Origin'][i]]
route_list.sort()
route_=route_list[0]+'_'+route_list[1]

try: competitor_dict_by_route[route_][key_i].append(dt['OpCarrier'][i])
except KeyError:
    try: competitor_dict_by_route[route_][key_i]=[dt['OpCarrier'][i]]
    except KeyError:
        competitor_dict_by_route[route_]={}
        competitor_dict_by_route[route_][key_i]=[dt['OpCarrier'][i]]

if dt['OpCarrier'][i]==carrier:

    """

    data=[dt['OpCarrier'][i],dt['Origin'][i],origin_name,\
dt['Dest'][i],dest_name,dt['Origin'][i]+'_'+dt['Dest'][i],\

```

```

        dt['Year'][i],dt['Quarter'][i],\
        int(dt['TotalPass'][i]),int(round(distance)),\
        int(round(distance/1.609344)),\
        int(round(distance/1.852)),1]

"""

try: leisure_route=max(leisure_dict[dt['Origin'][i]],\
                        leisure_dict[dt['Dest'][i]])
except KeyError:
    try: leisure_route=leisure_dict[dt['Origin'][i]]
    except KeyError:
        try: leisure_route=leisure_dict[dt['Dest'][i]]
        except KeyError: leisure_route=0

try:
    if business_dict[dt['Origin'][i]]==1 and\
        business_dict[dt['Dest'][i]]==1:
        business_route=1
    else: business_route=0
except KeyError: business_route=0

data=[dt['OpCarrier'][i],dt['Origin'][i]+'_'+dt['Dest'][i],\
      dt['Year'][i],dt['Quarter'][i],int(dt['TotalPass'][i]),\
      int(round(distance)),len(competitor_dict_by_route[dt['Origin'][i]+'_\
      '_+dt['Dest'][i]][key_i]),\

```

```

        route_population(dt['Origin'][i],dt['Dest'][i]),\
        leisure_route,business_route,1]

    if passenger_filter_flag:
        if (eval(dt['Year'][i]) in year_list) and\
            (eval(dt['Quarter'][i]) in quarter_list) and\
            int(dt['TotalPass'][i])>=90:
            full_data_dict[key_i].append(data)
    elif not passenger_filter_flag:
        if (eval(dt['Year'][i]) in year_list) and\
            (eval(dt['Quarter'][i]) in quarter_list):
            full_data_dict[key_i].append(data)

except TypeError: continue

print str(time.time()-t02)+' seconds to parse data.'

airport_list_all.sort(); airport_list_out=[]
for ii in airport_list_all: airport_list_out.append([ii,airport_location(ii)[0],\
                                                    airport_location(ii)[1]])

path=''

if passenger_filter_flag: f=open('all_routes_2002_2007_passfilt.txt','w')
elif not passenger_filter_flag: f=open('all_routes_2002_2007.txt','w')
for jj in airport_list_out:

```

```
        output=str(jj[0])+'\t'+str(jj[1])+'\t'+str(jj[2])+'\n'
        f.write(output)
f.close()

if passenger_filter_flag: f=open(data_path+'all_routes_2002_2007_passfilt.txt','w')
elif not passenger_filter_flag: f=open(data_path+'all_routes_2002_2007.txt','w')
for jj in airport_list_out:
    output=str(jj[0])+'\t'+str(jj[1])+'\t'+str(jj[2])+'\n'
    f.write(output)
f.close()

t03=time.time(); print; print 'Checking all possible routes.'

actual_route_dict={}
for i in full_data_dict.keys(): actual_route_dict[i]=[]

for i in full_data_dict.keys():
    for j in full_data_dict[i]:
        key=j[1]; period_key=j[2]+'_'+j[3]
        if key not in actual_route_dict[period_key]:
            actual_route_dict[period_key].append(key)

route_list_all_possible=[]
for x in airport_list_all:
    for y in airport_list_all:
```

```

        if x!=y:
            pair_list=[x,y]
            pair_list.sort()
            pair_string=pair_list[0]+'_'+pair_list[1]
            if pair_string not in route_list_all_possible:
                route_list_all_possible.append(pair_string)

print; print 'Airports in full static network '+str(len(airport_list_all))
print 'Total number of possible routes '+str(len(route_list_all_possible)); print

for i in year_list:
    for j in quarter_list:
        key=str(i)+'_'+str(j)
        path=''
        if passenger_filter_flag:
            f=open(data_path+carrier+'_actual_routes_'+key+'_passfilt.txt','w')
        elif not passenger_filter_flag:
            f=open(data_path+carrier+'_actual_routes_'+key+'.txt','w')
        for k in full_data_dict[key]:
            if k[0]==carrier:
                output=''
                for l in k: output=output+str(l)+'\t'
                output=output+'\n'
                f.write(output)
            else: pass

```



```

output=''

for mm in route_list_all_possible:
    if mm not in actual_route_dict[key]:
        distance_=distance_calculator(airport_location(mm.split('_')[1])[1],\
                                         airport_location(mm.split('_')[0])[1])
        distance_=abs(distance_)

    # need trap: some routes return <0 distances, e.g. OGG_SSI

    """
    output=output+carrier+'\t'+mm.split('_')[0]+\t'+\
    airport_location(mm.split('_')[0])[0]+\t'+\
        mm.split('_')[1]+\t'+airport_location(mm.split('_')[1])[0]+\
        '\t'+mm+'\t'+str(i)+'\t'+str(j)+'\t'+\
        '0'+'\t'+str(int(round(distance_)))+'\t'+\
        str(int(round(distance_/1.609344)))+'\t'+\
        str(int(round(distance_/1.852)))+'\t'+0'
    """

    try: leisure_route=max(leisure_dict[mm.split('_')[0]],\
                           leisure_dict[mm.split('_')[1]])
    except KeyError:
        try: leisure_route=leisure_dict[mm.split('_')[0]]
        except KeyError:
            try: leisure_route=leisure_dict[mm.split('_')[1]]

```

```

        except KeyError: leisure_route=0

    try:
        if business_dict[mm.split('_')[0]]==1 and\
            business_dict[mm.split('_')[1]]==1: business_route=1
        else: business_route=0
    except KeyError: business_route=0

    try:
        output=output+carrier+'\t'+mm+'\t'+str(i)+'\t'+str(j)+'\t'+\
            '0'+'\t'+str(int(round(distance_)))+'\t'+\
            str(len(competitor_dict_by_route[mm][key]))+'\t'+\
            route_population(mm.split('_')[0],mm.split('_')[1])+'\
            '\t'+str(leisure_route)+'\t'+str(business_route)+'\
            '\t'+0'

    except KeyError:
        output=output+carrier+'\t'+mm+'\t'+str(i)+'\t'+str(j)+'\t'+\
            '0'+'\t'+str(int(round(distance_)))+'\t'+0'+'\t'+\
            route_population(mm.split('_')[0],mm.split('_')[1])+'\
            '\t'+str(leisure_route)+'\t'+str(business_route)+'\
            '\t'+0'

    output=output+'\n'

f.write(output)

f.close()

```

```

        print 'Year:quarter, total number of actual routes '+str(i)+' ':'+str(j)+'\
              ', '+str(len(actual_route_dict[key]))

print str(time.time()-t03)+' seconds to complete.'

*****

# airport_location_function_modified_return

"""

Airport location degrees / function (2).

Steve Lawford : 22 July 2008.
Modified 10 November 2008 to return airport name.

"""

import os,sys,pickle

module_dir=os.path.join('Project','code')
sys.path.insert(0,module_dir)

from airport_location_dict_generator import *
```

```
data_path='C:/Project/data/'
try: f=open(data_path+'airport_locations_degrees_dict','r')
except IOError:
    create_location_dict()
    f=open(data_path+'airport_locations_degrees_dict','r')

airport_locations_degrees_dict=pickle.load(f)
f.close()

def airport_location(iata_code):

    try: return [airport_locations_degrees_dict[iata_code][0],\
                airport_locations_degrees_dict[iata_code][1]]
    except KeyError:
        print iata_code+' is not a valid code.'
        return

*****

# airport_location_dict_generator

"""

Airport location (degrees) / dictionary generator.
```

Parses .csv raw data file into Python dictionary.

Key: IATA code. Value: ['Airport (Country)',[latitude (N/S),longitude (W/E)]].

Note: coordinates in degrees, not radians.

Steve Lawford : 22 July 2008

Modified 19 January 2009 with removal of CHI and NYC IATA city codes.

```
"""
```

```
import os, csv, pickle
```

```
def create_location_dict():
```

```
    data_path='C:/Project/data/'
```

```
    f=open(data_path+'airport_locations.csv','r')
```

```
    reader=csv.reader(f)
```

```
    airport_location_dict={}
```

```
    print; print 'Parsing .csv file'; print
```

```
    for row in reader:
```

```
        if row[0]!='CHI' and row[0]!='NYC':
```

```
            airport_location_dict[row[0]]=row[1]+' ('+row[3]+'),\n
```

```

        [float(row[5]),float(row[6])]]

    f.close()

    g=open(data_path+'airport_locations_degrees_dict','w')
    pickle.dump(airport_location_dict,g)
    g.close()

    return

*****

# safe_cPickle

"""

Safe cPickle (dump) function.

Steve Lawford : 26 August 2008.

Call with:

from safe_cPickle import *
safe_cPickle_dump(data_path,filename,object)

"""

```

```

def safe_cPickle_dump(data_path,filename,object):

    import sys,cPickle

    try:

        f=open(data_path+filename,'r'); f.close(); rsp=''
        while rsp!='n' and rsp!='y':
            rsp=raw_input('(safe_cPickle has detected object "'+filename+\'
                           \'") Overwrite? (y/n) ')

        if rsp=='y':
            g=open(data_path+filename,'w'); cPickle.dump(object,g); g.close()
        else: sys.exit(0)
    except IOError:

        print '(safe_cPickle has not detected object "'+\'
              filename+\'") Saving object.'
        g=open(data_path+filename,'w')
        cPickle.dump(object,g)
        g.close()

    return

*****

# route_population_geom_mean_function

```

```
"""

Socio-demographic data parse (from .xls)
- for WN entry model project.

Steve Lawford: 12 December 2008.
Modified 13 December 2008 for population calculation and function return.

"""

import cPickle,time
from safe_cPickle import *
from scipy import *

def route_population(end1,end2):

    data_path='C:/Project/data/'
    try:
        c_x_x=open(data_path+'airport_populations','r')
        airport_dict=cPickle.load(c_x_x)
        c_x_x.close()
    except IOError:
        f=open(data_path+'population_by_us_airport.txt','r')
        airport_dict={}

```



```

        for i in f:
            j=i.split('\t')
            if j[0] not in airport_dict.keys():
                airport_dict[j[0]]=j[3].split('\n')[0]
            print j
        f.close()

        safe_cPickle_dump(data_path,'airport_populations',airport_dict)

ends=[end1,end2]
ends.sort()

try: return str(int(sqrt(float(airport_dict[ends[0]])*\
                                float(airport_dict[ends[1]]))))
except KeyError:
    print 'Error: endpoint not in dictionary.'
    sys.exit(0)

*****

# route_leisure_function

"""

Socio-demographic data parse (from .xls) / leisure route indicator.
- for WN entry model project.

```

Steve Lawford: 18 December 2008.

```
"""
```

```
import cPickle
```

```
from safe_cPickle import *
```

```
from scipy import *
```

```
def leisure_function(leisure_threshold):
```

```
    data_path='C:/Project/data/'
```

```
    try:
```

```
        c_x_x=open(data_path+'airport_leisure','r')
```

```
        airport_leisure_dict=cPickle.load(c_x_x)
```

```
        c_x_x.close()
```

```
    except IOError:
```

```
        f=open(data_path+'gdp_msa_by_us_airport.txt','r')
```

```
        airport_leisure_dict={}
```

```
        for i in f:
```

```
            j=i.split('\t')
```

```
            if j[0] not in airport_leisure_dict.keys() and j[2]!='' and\
```

```
                j[3]!='':
```

```
                airport_leisure_dict[j[0]]=[j[1],float(j[3])/float(j[2])]
```

```
airport_leisure_sort_list=[]
ak=airport_leisure_dict.keys()[:]; aksort=[]
for i in range(len(ak)):
    airport_leisure_sort_list.append(str(airport_leisure_dict\
        [ak[i]][1])+'_'+ak[i])
    aksort.append(airport_leisure_dict[ak[i]][1])
aksort.sort()

akrsort=[]
for i in range(len(aksort)-1,0,-1):
    akrsort.append(aksort[i])

leisure_list=[]
for j in akrsort:
    for k in range(len(airport_leisure_sort_list)):
        if str(j)==airport_leisure_sort_list[k].split('_')[0]:
            leisure_list.append([airport_leisure_sort_list[k].split\
                ('_')[1],airport_leisure_dict[airport_leisure_sort_list\
                [k].split('_')[1]][0],airport_leisure_dict\
                [airport_leisure_sort_list[k].split('_')[1]][1]])
            airport_leisure_sort_list.pop(k)
            break

print
for i in leisure_list: print i
```

```
print

indicator_dict={}

for i in leisure_list:
    if i[2]>=leisure_threshold: indicator_dict[i[0]]=1
    else: indicator_dict[i[0]]=0

return indicator_dict

*****

# route_business_function

"""

Socio-demographic data parse (from .xls) / business route indicator.
- for WN entry model project.

Steve Lawford: 18 December 2008.

"""

import cPickle
from safe_cPickle import *
from scipy import *
```

```

def business_function(business_threshold):

    data_path='C:/Project/data/'

    try:

        c_x_x=open(data_path+'airport_business','r')

        airport_business_dict=cPickle.load(c_x_x)

        c_x_x.close()

    except IOError:

        f=open(data_path+'gdp_msa_by_us_airport.txt','r')

        airport_business_dict={}

        for i in f:

            j=i.split('\t')

            if j[0] not in airport_business_dict.keys() and j[2]!='' and\
                j[4]!='' and j[4]!='\n':

                airport_business_dict[j[0]]=[j[1],\
                    float(j[4].split('\n')[0])/float(j[2])]

        airport_business_sort_list=[]

        ak=airport_business_dict.keys(); aksort=[]

        for i in range(len(ak)):

            airport_business_sort_list.append(str(airport_business_dict\
                [ak[i]][1])+ '_'+ak[i])

            aksort.append(airport_business_dict[ak[i]][1])

```

```
aksort.sort()

akrsort=[]

for i in range(len(aksort)-1,0,-1):
    akrsort.append(aksort[i])

business_list=[]

for j in akrsort:
    for k in range(len(airport_business_sort_list)):
        if str(j)==airport_business_sort_list[k].split('_')[0]:
            business_list.append([airport_business_sort_list[k].split\
                                   ('_')[1],airport_business_dict[airport_business_sort_list\
                                   [k].split('_')[1]][0],airport_business_dict\
                                   [airport_business_sort_list[k].split('_')[1]][1]])
            airport_business_sort_list.pop(k)
            break

print

for i in business_list: print i

print

indicator_dict={}

for i in business_list:
    if i[2]>=business_threshold: indicator_dict[i[0]]=1
    else: indicator_dict[i[0]]=0
```

```
        return indicator_dict

*****

# entry_model_hub_detection

"""

Hub detection based on complete_coupon for WN entry model project.

Steve Lawford: 3 January 2009.
Modified 12 January 2009 with text file output.
Modified 19 January 2009 with removal of CHI manual error trap.
Modified 20 January 2009 with automatic passenger filter.
Modified 21 January 2009 with carrier loop.

"""

import cPickle,time,os,sys
from airport_location_function_ import *

module_dir=os.path.join('Project','code')
sys.path.insert(0,module_dir)
```

```
from safe_cPickle import *
from scipy import *

data_path='C:/Project/data/'; filename='complete_coupon'

passenger_filter_flag=True

t01=time.time(); print; print 'Loading dictionary: '+filename
c_x_x=open(data_path+filename,'r'); dt=cPickle.load(c_x_x)
print str(time.time()-t01)+' seconds to load dictionary.'
c_x_x.close()

year=['2002','2003','2004','2005','2006','2007']; quarter=['1','2','3','4']
#year=['2007']; quarter=['4']

full_airports=[]
for i in range(len(dt['OpCarrier'])):
    if dt['Dest'][i] not in full_airports:
        full_airports.append(dt['Dest'][i])
    if dt['Origin'][i] not in full_airports:
        full_airports.append(dt['Origin'][i])

full_airports.pop(full_airports.index('NYC'))
full_airports.pop(full_airports.index('CHI'))
```



```

close_dict_0_50={}
for i in full_airports: close_dict_0_50[i]=[]
close_dict_50_100={}
for i in full_airports: close_dict_50_100[i]=[]
close_dict_100_150={}
for i in full_airports: close_dict_100_150[i]=[]

r=6372.7974775959065

print 'Creating nearby airport dictionaries.'
for i in full_airports:
    for j in full_airports:
        if i!=j:
            p1,l1=(pi/180)*airport_location(i)[0],(pi/180)*\
                airport_location(i)[1]
            p2,l2=(pi/180)*airport_location(j)[0],(pi/180)*\
                airport_location(j)[1]
            l=abs(l1-l2)
            num=sqrt(((cos(p2)*sin(l))**2)+(((cos(p1)*sin(p2))-
                (sin(p1)*cos(p2)*cos(l))))**2))
            den=sin(p1)*sin(p2)+cos(p1)*cos(p2)*cos(l)
            theta=arctan(num/den)
            distance=abs(int(round(r*theta)))
            if distance<=50: close_dict_0_50[i].append(j)
            if distance>50 and distance<=100: close_dict_50_100[i].append(j)
            if distance>100 and distance<=150: close_dict_100_150[i].append(j)

```

```
print 'Nearby airport dictionaries complete.'

safe_cPickle_dump(data_path,'close_dict_0_50',close_dict_0_50)
safe_cPickle_dump(data_path,'close_dict_50_100',close_dict_50_100)
safe_cPickle_dump(data_path,'close_dict_100_150',close_dict_100_150)

for carrier in ['WN','AA','CO','DL','NW','UA','US']:
    for year_ in year:
        for quarter_ in quarter:
            print year_; print quarter_; print
            carrier_airports=[]; print
            print 'Compiling carrier airport list: '+carrier; print
            for i in range(len(dt['OpCarrier'])):

                if passenger_filter_flag:
                    if dt['OpCarrier'][i]==carrier and dt['Year'][i]==year_ and\
                        dt['Quarter'][i]==quarter_ and int(dt['TotalPass'][i])>=90:
                        if dt['Dest'][i] not in carrier_airports:
                            carrier_airports.append(dt['Dest'][i])
                        if dt['Origin'][i] not in carrier_airports:
                            carrier_airports.append(dt['Origin'][i])
                elif not passenger_filter_flag:
                    if dt['OpCarrier'][i]==carrier and dt['Year'][i]==year_ and\
                        dt['Quarter'][i]==quarter_:
                        if dt['Dest'][i] not in carrier_airports:
```

```
        carrier_airports.append(dt['Dest'][i])
    if dt['Origin'][i] not in carrier_airports:
        carrier_airports.append(dt['Origin'][i])

print carrier_airports; print

carrier_dict={}
for i in carrier_airports: carrier_dict[i]=[0,0,0,0,0,0]

for i in range(len(dt['OpCarrier'])):
    if passenger_filter_flag:
        if dt['Year'][i]==year_ and dt['Quarter'][i]==quarter_ and\
            int(dt['TotalPass'][i])>=90:
            if dt['Origin'][i] in carrier_dict.keys():
                carrier_dict[dt['Origin'][i]][1]+=dt['TotalPass'][i]
                carrier_dict[dt['Origin'][i]][3]+=1
                if dt['OpCarrier'][i]==carrier:
                    carrier_dict[dt['Origin'][i]][0]+=dt['TotalPass'][i]
                    carrier_dict[dt['Origin'][i]][2]+=1
            if dt['Dest'][i] in carrier_dict.keys():
                carrier_dict[dt['Dest'][i]][1]+=dt['TotalPass'][i]
                carrier_dict[dt['Dest'][i]][3]+=1
                if dt['OpCarrier'][i]==carrier:
                    carrier_dict[dt['Dest'][i]][0]+=dt['TotalPass'][i]
                    carrier_dict[dt['Dest'][i]][2]+=1
```

```

elif not passenger_filter_flag:
    if dt['Year'][i]==year_ and dt['Quarter'][i]==quarter_:
        if dt['Origin'][i] in carrier_dict.keys():
            carrier_dict[dt['Origin'][i]][1]+=dt['TotalPass'][i]
            carrier_dict[dt['Origin'][i]][3]+=1
            if dt['OpCarrier'][i]==carrier:
                carrier_dict[dt['Origin'][i]][0]+=dt['TotalPass'][i]
                carrier_dict[dt['Origin'][i]][2]+=1
        if dt['Dest'][i] in carrier_dict.keys():
            carrier_dict[dt['Dest'][i]][1]+=dt['TotalPass'][i]
            carrier_dict[dt['Dest'][i]][3]+=1
            if dt['OpCarrier'][i]==carrier:
                carrier_dict[dt['Dest'][i]][0]+=dt['TotalPass'][i]
                carrier_dict[dt['Dest'][i]][2]+=1

for i in carrier_dict.keys():
    carrier_dict[i][4]=float(carrier_dict[i][0])/float(carrier_dict[i][1])
    carrier_dict[i][5]=float(carrier_dict[i][2])/float(carrier_dict[i][3])

print year_; print quarter_; print carrier_dict; print

carrier_hub_passenger=[]; carrier_hub_route=[]
for i in carrier_dict.keys():
    carrier_hub_passenger.append([carrier_dict[i][4],i])
    carrier_hub_route.append([carrier_dict[i][5],i])

```

```
carrier_hub_passenger.sort()
carrier_hub_route.sort()

print; print 'Passenger'; print carrier_hub_passenger; print
print 'Route'; print carrier_hub_route

output_list=[]
for i in carrier_hub_passenger:
    if i[0]<0.3: output_list.append([i[1],'1','0','0','0'])
    elif i[0]>=0.3 and i[0]<0.5: output_list.append\
        ([i[1],'0','1','0','0'])
    elif i[0]>=0.5 and i[0]<0.7: output_list.append\
        ([i[1],'0','0','1','0'])
    elif i[0]>=0.7: output_list.append([i[1],'0','0','0','1'])

print; print output_list; print

full_airport_list=[]; filename3='region_by_airport.txt'
g=open(data_path+filename3,'r')
while 1==1:
    a=g.readline()
    b=a.split('\t')
    if len(b[0])==3: full_airport_list.append(b[0])
    if len(b[0])==0: break
```

```

g.close()

for k in full_airport_list:
    if k in carrier_airports: pass
    else: output_list.append([k,'1','0','0','0'])

if passenger_filter_flag:
    filename2=carrier+'_passenger_hub_'+year+'_'+\
        quarter+'_passfilt.txt'
elif not passenger_filter_flag:
    filename2=carrier+'_passenger_hub_'+year+'_'+quarter+'.txt'

f=open(data_path+filename2,'w')
for i in output_list:
    output=''
    for j in i: output=output+j+'\t'
    output=output+'\n'
    f.write(output)
f.close()

*****

# airport_location_function_

"""

```

Airport location (degrees) / function.

Steve Lawford : 22 July 2008

Modified 14 January 2009 with removal of airport name print.

```
"""
```

```
import os,sys,pickle
```

```
module_dir=os.path.join('Project','code')
```

```
sys.path.insert(0,module_dir)
```

```
from airport_location_dict_generator import *
```

```
data_path='C:/Project/data/'
```

```
try: f=open(data_path+'airport_locations_degrees_dict','r')
```

```
except IOError:
```

```
    create_location_dict()
```

```
    f=open(data_path+'airport_locations_degrees_dict','r')
```

```
airport_locations_degrees_dict=pickle.load(f)
```

```
f.close()
```

```
def airport_location(iata_code):
```

```
    try:
```

```
        #print airport_locations_degrees_dict[iata_code][0]
```

```
        return airport_locations_degrees_dict[iata_code][1]
```

```
    except KeyError:
```

```
        print iata_code+' is not a valid code.'
```

```
    return
```

```
*****
```

```
# python_to_text
```

```
"""
```

WN_actual_routes_year_quarter, for WN entry model project.

Steve Lawford: 12 January 2009.

Modified 13 January 2009 with airports excluded by the Wright Amend. (from DAL).

Modified 13 January 2009 with slot dummy.

Modified 14 January 2009 with major carrier hubs.

Modified 16 January 2009 with minor correction in list initializations.

Modified 17 January 2009 with passenger filter (manual).

Modified 17 January 2009 with arrival delay data (original data artificial CHI data).

Modified 17 January 2009 with distance ratio.

Modified 17 January 2009 with LCC data (original data have artificial CHI data)

Modified 18 January 2009 with WN airport service length data.

Modified 20 January 2009 with automatic passenger filter.

```
"""
```

```
import cPickle,time,os,sys
```

```
module_dir=os.path.join('Project','code')
```

```
sys.path.insert(0,module_dir)
```

```
from safe_cPickle import *
```

```
from scipy import *
```

```
from airport_location_function_ import *
```

```
data_path='C:/Project/data/'
```

```
year_=['2002','2003','2004','2005','2006','2007']; quarter_=['1','2','3','4']
```

```
#year_=['2007']; quarter_=['4']
```

```
passenger_filter_flag=True
```

```
fixed_year={'2002':0,'2003':1,'2004':2,'2005':3,'2006':4,'2007':5}
```

```
fixed_year_quarter={'2002_1':0,'2002_2':1,'2002_3':2,'2002_4':3,\n                    '2003_1':4,'2003_2':5,'2003_3':6,'2003_4':7,\
```

```
'2004_1':8,'2004_2':9,'2004_3':10,'2004_4':11,\n'2005_1':12,'2005_2':13,'2005_3':14,'2005_4':15,\n'2006_1':16,'2006_2':17,'2006_3':18,'2006_4':19,\n'2007_1':20,'2007_2':21,'2007_3':22,'2007_4':23}
```

```
f=open(data_path+'close_dict_0_50','r'); close_dict_0_50=pickle.load(f); f.close()\nf=open(data_path+'close_dict_50_100','r'); close_dict_50_100=pickle.load(f); f.close()\nf=open(data_path+'close_dict_100_150','r'); close_dict_100_150=pickle.load(f)\nf.close()\nf=open(data_path+'service_since_dict','r'); service_since_dict=pickle.load(f)\nf.close()\nf=open(data_path+'jetblue_dict','r'); jetblue_dict=pickle.load(f); f.close()\nf=open(data_path+'frontier_dict','r'); frontier_dict=pickle.load(f); f.close()\nf=open(data_path+'airtran_dict','r'); airtran_dict=pickle.load(f); f.close()\nf=open(data_path+'alaska_dict','r'); alaska_dict=pickle.load(f); f.close()\n\nfilename='dep_delay_dict'\nt01=time.time(); print; print 'Loading dictionary: '+filename\nc_x_x=open(data_path+filename,'r'); dep_delay_dict=cPickle.load(c_x_x)\nprint str(time.time()-t01)+' seconds to load dictionary.'\nc_x_x.close()\n\nfilename='arr_delay_dict'\nt01=time.time(); print; print 'Loading dictionary: '+filename\nc_x_x=open(data_path+filename,'r'); arr_delay_dict=cPickle.load(c_x_x)
```

```

print str(time.time()-t01)+' seconds to load dictionary.'
c_x_x.close()

"""

filename='carrier_airport_dict'
t01=time.time(); print; print 'Loading dictionary: '+filename
c_x_x=open(data_path+filename,'r'); carrier_airport_dict=cPickle.load(c_x_x)
print str(time.time()-t01)+' seconds to load dictionary.'
c_x_x.close()
"""

for year in year_:
    for quarter in quarter_:

        if passenger_filter_flag:
            filename='WN_actual_routes_'+year+'_'+quarter+'_passfilt.txt'
        elif not passenger_filter_flag:
            filename='WN_actual_routes_'+year+'_'+quarter+'.txt'

        dt={}; t01=time.time(); print; print 'Parsing text file: '+filename

        keys=['carrier','route','year','quarter','passengers','distance',\
            'number','population','leisure','business','presence']

        for j in keys: dt[j]=[]

```

```
f=open(data_path+filename,'r')
while 1==1:
    try:
        a=f.readline()
        b=a.split('\t')
        dt['carrier'].append(b[0])
        dt['route'].append(b[1])
        dt['year'].append(b[2])
        dt['quarter'].append(b[3])
        dt['passengers'].append(b[4])
        dt['distance'].append(b[5])
        dt['number'].append(b[6])
        dt['population'].append(b[7])
        dt['leisure'].append(b[8])
        dt['business'].append(b[9])
        dt['presence'].append(b[10])
    except IndexError: break
print str(time.time()-t01)+' seconds to parse text file.'
f.close()

dt['region']=[]; region_dict={}; filename2='region_by_airport.txt'
t02=time.time(); print; print 'Loading airport region text file: '+filename2
g=open(data_path+filename2,'r')
while 1==1:
    try:
```

```

        a=g.readline()

        b=a.split('\t')

        region_dict[b[0]]=b[1].split('\n')[0]

    except IndexError: break

print str(time.time()-t02)+' seconds to parse text file.'

g.close()

region_pair_dict={'1_1':1,'1_2':2,'1_3':3,'1_4':4,'1_5':5,'2_2':6,\
                  '2_3':7,'2_4':8,'2_5':9,'3_3':10,'3_4':11,\
                  '3_5':12,'4_4':13,'4_5':14,'5_5':15}

for j in dt['route']:

    c=[int(region_dict[j.split('_')[0]]),int(region_dict[j.split('_')[1]])]

    c.sort()

    d=str(c[0])+ '_' +str(c[1])

    catch_list=['0']*15

    catch_list[region_pair_dict[d]-1]='1'

    dt['region'].append(catch_list)

keys_=['carrier','route','year','quarter','passengers','distance',\
       'number','population','leisure','business','region1','region2',\
       'region3','region4','region5','region6','region7','region8',\
       'region9','region10','region11','region12','region13','region14',\
       'region15','presence']

```

```

dt_={}

for k in keys_: dt_[k]=[]

for j in range(len(dt['carrier'])-1):
    for i in keys_:
        if i[:6]!='region': dt_[i].append(dt[i][j])
        else: dt_[i].append(dt['region'][j][int(i[6:])-1])

hub_dict={}

if passenger_filter_flag:
    filename4='WN_passenger_hub_'+year+'_'+quarter+'_passfilt.txt'
elif not passenger_filter_flag:
    filename4='WN_passenger_hub_'+year+'_'+quarter+'.txt'

f=open(data_path+filename4,'r')
while 1==1:
    a=f.readline()
    b=a.split('\t')
    if len(b[0])==3: hub_dict[b[0]]=[b[1],b[2],b[3],b[4]]
    else: break

wright_dict={}
filename4='wright_airports.txt'
t04=time.time(); print; print 'Parsing Wright Amend. text file: '+filename4

```

```
g=open(data_path+filename4,'r')
while 1==1:
    a=g.readline()
    b=a.split('\t')
    if len(b[0])==3: wright_dict[b[0]]=b[1].split('\n')[0]
    else: break
print str(time.time()-t04)+' seconds to parse text file.'
g.close()

slot_dict={}
filename5='slot_restricted.txt'
t05=time.time(); print; print 'Parsing slot retriected text file: '+filename5
g=open(data_path+filename5,'r')
while 1==1:
    a=g.readline()
    b=a.split('\t')
    if len(b[0])==3: slot_dict[b[0]]=b[1].split('\n')[0]
    else: break
print str(time.time()-t05)+' seconds to parse text file.'
g.close()

aa_hub_dict={}

if passenger_filter_flag:
    filename6='AA_passenger_hub_'+year+'_'+quarter+'_passfilt.txt'
```

```
elif not passenger_filter_flag:

    filename6='AA_passenger_hub_'+year+'_'+quarter+'.txt'

t06=time.time(); print; print 'Parsing major hub text file: '+filename6
g=open(data_path+filename6,'r')
while 1==1:

    a=g.readline()

    b=a.split('\t')

    if len(b[0])==3: aa_hub_dict[b[0]]=[b[1],b[2],b[3],b[4]]

    else: break

print str(time.time()-t06)+' seconds to parse text file.'
g.close()

us_hub_dict={}

if passenger_filter_flag:

    filename6='US_passenger_hub_'+year+'_'+quarter+'_passfilt.txt'
elif not passenger_filter_flag:

    filename6='US_passenger_hub_'+year+'_'+quarter+'.txt'

t06=time.time(); print; print 'Parsing major hub text file: '+filename6
g=open(data_path+filename6,'r')
while 1==1:

    a=g.readline()

    b=a.split('\t')
```



```
        if len(b[0])==3: us_hub_dict[b[0]]=[b[1],b[2],b[3],b[4]]
        else: break
print str(time.time()-t06)+' seconds to parse text file.'
g.close()

ua_hub_dict={}

if passenger_filter_flag:
    filename6='UA_passenger_hub_'+year+'_'+quarter+'_passfilt.txt'
elif not passenger_filter_flag:
    filename6='UA_passenger_hub_'+year+'_'+quarter+'.txt'

t06=time.time(); print; print 'Parsing major hub text file: '+filename6
g=open(data_path+filename6,'r')
while 1==1:
    a=g.readline()
    b=a.split('\t')
    if len(b[0])==3: ua_hub_dict[b[0]]=[b[1],b[2],b[3],b[4]]
    else: break
print str(time.time()-t06)+' seconds to parse text file.'
g.close()

nw_hub_dict={}

if passenger_filter_flag:
```

```
filename6='NW_passenger_hub_'+year+'_'+quarter+'_passfilt.txt'
elif not passenger_filter_flag:
    filename6='NW_passenger_hub_'+year+'_'+quarter+'.txt'

t06=time.time(); print; print 'Parsing major hub text file: '+filename6
g=open(data_path+filename6,'r')
while 1==1:
    a=g.readline()
    b=a.split('\t')
    if len(b[0])==3: nw_hub_dict[b[0]]=[b[1],b[2],b[3],b[4]]
    else: break
print str(time.time()-t06)+' seconds to parse text file.'
g.close()

dl_hub_dict={}

if passenger_filter_flag:
    filename6='DL_passenger_hub_'+year+'_'+quarter+'_passfilt.txt'
elif not passenger_filter_flag:
    filename6='DL_passenger_hub_'+year+'_'+quarter+'.txt'

t06=time.time(); print; print 'Parsing major hub text file: '+filename6
g=open(data_path+filename6,'r')
while 1==1:
    a=g.readline()
```

```

        b=a.split('\t')
        if len(b[0])==3: dl_hub_dict[b[0]]=[b[1],b[2],b[3],b[4]]
        else: break
print str(time.time()-t06)+' seconds to parse text file.'
g.close()

co_hub_dict={}

if passenger_filter_flag:
    filename6='CO_passenger_hub_'+year+'_'+quarter+'_passfilt.txt'
elif not passenger_filter_flag:
    filename6='CO_passenger_hub_'+year+'_'+quarter+'.txt'

t06=time.time(); print; print 'Parsing major hub text file: '+filename6
g=open(data_path+filename6,'r')
while 1==1:
    a=g.readline()
    b=a.split('\t')
    if len(b[0])==3: co_hub_dict[b[0]]=[b[1],b[2],b[3],b[4]]
    else: break
print str(time.time()-t06)+' seconds to parse text file.'
g.close()

keys__=['carrier','route','year','quarter','passengers','distance',\
        'number','population','leisure','business','region1','region2',\

```

```

'region3','region4','region5','region6','region7','region8',\
'region9','region10','region11','region12','region13','region14',\
'region15','pass_hub_1','pass_hub_2','pass_hub_3','pass_hub_4',\
'wright','slot','aa_pass_hub_1','aa_pass_hub_2','aa_pass_hub_3',\
'aa_pass_hub_4','us_pass_hub_1','us_pass_hub_2','us_pass_hub_3',\
'us_pass_hub_4','ua_pass_hub_1','ua_pass_hub_2','ua_pass_hub_3',\
'ua_pass_hub_4','nw_pass_hub_1','nw_pass_hub_2','nw_pass_hub_3',\
'nw_pass_hub_4','dl_pass_hub_1','dl_pass_hub_2','dl_pass_hub_3',\
'dl_pass_hub_4','co_pass_hub_1','co_pass_hub_2','co_pass_hub_3',\
'co_pass_hub_4','hub_0_50','hub_50_100','hub_100_150',\
'nearby_route_0_50','nearby_route_50_100',\
'nearby_endpoint_0_50','nearby_endpoint_50_100',\
'n_','aa_pass_hub','us_pass_hub','ua_pass_hub',\
'nw_pass_hub','dl_pass_hub','co_pass_hub','max_dep_delay',\
'max_arr_delay','max_delay','distance_ratio',\
'jetblue','airtran','frontier','alaska','service_since',\
'presence']

```

```
dt_['pass_hub_4']=[]; dt_['pass_hub_3']=[]
```

```
dt_['pass_hub_2']=[]; dt_['pass_hub_1']=[]
```

```
dt_['wright']=[]; dt_['slot']=[]; dt_['n_']=[]
```

```
dt_['aa_pass_hub_4']=[]; dt_['aa_pass_hub_3']=[]
```

```
dt_['aa_pass_hub_2']=[]; dt_['aa_pass_hub_1']=[]
```

```

dt_['us_pass_hub_4']=[]; dt_['us_pass_hub_3']=[]
dt_['us_pass_hub_2']=[]; dt_['us_pass_hub_1']=[]
dt_['ua_pass_hub_4']=[]; dt_['ua_pass_hub_3']=[]
dt_['ua_pass_hub_2']=[]; dt_['ua_pass_hub_1']=[]
dt_['nw_pass_hub_4']=[]; dt_['nw_pass_hub_3']=[]
dt_['nw_pass_hub_2']=[]; dt_['nw_pass_hub_1']=[]
dt_['dl_pass_hub_4']=[]; dt_['dl_pass_hub_3']=[]
dt_['dl_pass_hub_2']=[]; dt_['dl_pass_hub_1']=[]
dt_['co_pass_hub_4']=[]; dt_['co_pass_hub_3']=[]
dt_['co_pass_hub_2']=[]; dt_['co_pass_hub_1']=[]

dt_['hub_0_50']=[]; dt_['hub_50_100']=[]; dt_['hub_100_150']=[]

dt_['nearby_route_0_50']=[]; dt_['nearby_route_50_100']=[]
dt_['nearby_route_100_150']=[]; dt_['nearby_endpoint_0_50']=[]
dt_['nearby_endpoint_50_100']=[]; dt_['nearby_endpoint_100_150']=[]

dt_['max_dep_delay']=[]; dt_['max_arr_delay']=[]; dt_['max_delay']=[]

dt_['aa_pass_hub']=[]; dt_['us_pass_hub']=[]; dt_['ua_pass_hub']=[]
dt_['nw_pass_hub']=[]; dt_['dl_pass_hub']=[]; dt_['co_pass_hub']=[]

dt_['distance_ratio']=[]; dt_['service_since']=[]

dt_['jetblue']=[]; dt_['airtran']=[]

```

```

dt_['frontier']=[]; dt_['alaska']=[]

route_dist_dict={}; route_pass_dict={}

for i in range(len(dt_['carrier'])):

    """
    dt_['service_since'].append(str(service_since_dict\
        [dt_['route'][i].split('_')[0]][fixed_year[year]]+\
        service_since_dict[dt_['route'][i].split('_')[1]]\
        [fixed_year[year]]))
    """

    dt_['service_since'].append(str(max(service_since_dict\
        [dt_['route'][i].split('_')[0]][fixed_year[year]],\
        service_since_dict[dt_['route'][i].split('_')[1]]\
        [fixed_year[year]]))))

if dt_['presence'][i].split('\n')[0]=='1':
    if dt_['route'][i].split('_')[0] not in route_dist_dict.keys():
        route_dist_dict[dt_['route'][i].split('_')[0]]=\
            [[eval(dt_['distance'][i]),eval(dt_['passengers'][i])]]
        route_pass_dict[dt_['route'][i].split('_')[0]]=\
            eval(dt_['passengers'][i])

```

```

elif dt_['route'][i].split('_')[0] in route_dist_dict.keys():
    route_dist_dict[dt_['route'][i].split('_')[0]].append\
        ([eval(dt_['distance'][i]),eval(dt_['passengers'][i])])
    route_pass_dict[dt_['route'][i].split('_')[0]]=\
        route_pass_dict[dt_['route'][i].split('_')[0]]+\
        eval(dt_['passengers'][i])
else: pass

if dt_['presence'][i].split('\n')[0]=='1':
    if dt_['route'][i].split('_')[1] not in route_dist_dict.keys():
        route_dist_dict[dt_['route'][i].split('_')[1]]=\
            [[eval(dt_['distance'][i]),eval(dt_['passengers'][i])]]
        route_pass_dict[dt_['route'][i].split('_')[1]]=\
            eval(dt_['passengers'][i])
    elif dt_['route'][i].split('_')[1] in route_dist_dict.keys():
        route_dist_dict[dt_['route'][i].split('_')[1]].append\
            ([eval(dt_['distance'][i]),eval(dt_['passengers'][i])])
        route_pass_dict[dt_['route'][i].split('_')[1]]=\
            route_pass_dict[dt_['route'][i].split('_')[1]]+\
            eval(dt_['passengers'][i])
else: pass

"""
dt_['jetblue'].append(str(max(jetblue_dict\
    [dt_['route'][i].split('_')[0]][fixed_year[year]],\

```

```

        jetblue_dict[dt_['route'][i].split('_')[1]][fixed_year[year]])
dt_['frontier'].append(str(max(frontier_dict\
    [dt_['route'][i].split('_')[0]][fixed_year[year]],\
    frontier_dict[dt_['route'][i].split('_')[1]][fixed_year[year]])))
dt_['airtran'].append(str(max(airtran_dict\
    [dt_['route'][i].split('_')[0]][fixed_year[year]],\
    airtran_dict[dt_['route'][i].split('_')[1]][fixed_year[year]])))
dt_['alaska'].append(str(max(alaska_dict\
    [dt_['route'][i].split('_')[0]][fixed_year[year]],\
    alaska_dict[dt_['route'][i].split('_')[1]][fixed_year[year]])))
"""

dt_['jetblue'].append(str(max(jetblue_dict\
    [dt_['route'][i].split('_')[0]][fixed_year_quarter\
    [year+'_'+quarter]],jetblue_dict[dt_['route'][i].split\
    ('_')[1]][fixed_year_quarter[year+'_'+quarter]])))
dt_['frontier'].append(str(max(frontier_dict\
    [dt_['route'][i].split('_')[0]][fixed_year_quarter\
    [year+'_'+quarter]],frontier_dict[dt_['route'][i].split\
    ('_')[1]][fixed_year_quarter[year+'_'+quarter]])))
dt_['airtran'].append(str(max(airtran_dict\
    [dt_['route'][i].split('_')[0]][fixed_year_quarter\
    [year+'_'+quarter]],airtran_dict[dt_['route'][i].split\
    ('_')[1]][fixed_year_quarter[year+'_'+quarter]])))
dt_['alaska'].append(str(max(alaska_dict\

```



```

[dt_['route'][i].split('_')[0]][fixed_year_quarter\
[year+'_'+quarter]],alaska_dict[dt_['route'][i].split\
('_')[1]][fixed_year_quarter[year+'_'+quarter]])))

"""

if dt_['route'][i].split('_')[0]=='DAL':
    print '0 DAL',dt_['route'][i].split('_')
    raw_input()

if dt_['route'][i].split('_')[1]=='DAL':
    print '1 DAL',dt_['route'][i].split('_')
    raw_input()

"""

if hub_dict[dt_['route'][i].split('_')[0]].index('1')>=\
hub_dict[dt_['route'][i].split('_')[1]].index('1'):
    cc=hub_dict[dt_['route'][i].split('_')[0]]
else: cc=hub_dict[dt_['route'][i].split('_')[1]]

dt_['pass_hub_4'].append(cc[3]); dt_['pass_hub_3'].append(cc[2])
dt_['pass_hub_2'].append(cc[1]); dt_['pass_hub_1'].append(cc[0])

if dt_['route'][i].split('_')[0]=='DAL':
    dt_['wright'].append(wright_dict[dt_['route'][i].split('_')[1]])
elif dt_['route'][i].split('_')[1]=='DAL':

```

```

        dt_['wright'].append(wright_dict[dt_['route'][i].split('_')[0]])
    else: dt_['wright'].append('0')

    if (slot_dict[dt_['route'][i].split('_')[0]]=='1') or\
        (slot_dict[dt_['route'][i].split('_')[1]]=='1'):
        dt_['slot'].append('1')
    else: dt_['slot'].append('0')

    if aa_hub_dict[dt_['route'][i].split('_')[0]].index('1')>=\
        aa_hub_dict[dt_['route'][i].split('_')[1]].index('1'):
        cc=aa_hub_dict[dt_['route'][i].split('_')[0]]
    else: cc=aa_hub_dict[dt_['route'][i].split('_')[1]]

    dt_['aa_pass_hub_4'].append(cc[3]); dt_['aa_pass_hub_3'].append(cc[2])
    dt_['aa_pass_hub_2'].append(cc[1]); dt_['aa_pass_hub_1'].append(cc[0])

    if us_hub_dict[dt_['route'][i].split('_')[0]].index('1')>=\
        us_hub_dict[dt_['route'][i].split('_')[1]].index('1'):
        cc=us_hub_dict[dt_['route'][i].split('_')[0]]
    else: cc=us_hub_dict[dt_['route'][i].split('_')[1]]

    dt_['us_pass_hub_4'].append(cc[3])
    dt_['us_pass_hub_3'].append(cc[2])
    dt_['us_pass_hub_2'].append(cc[1])
    dt_['us_pass_hub_1'].append(cc[0])

```

```

if ua_hub_dict[dt_['route']][i].split('_')[0].index('1')>=\
    ua_hub_dict[dt_['route']][i].split('_')[1].index('1'):
    cc=ua_hub_dict[dt_['route']][i].split('_')[0]
else: cc=ua_hub_dict[dt_['route']][i].split('_')[1]
dt_['ua_pass_hub_4'].append(cc[3]); dt_['ua_pass_hub_3'].append(cc[2])
dt_['ua_pass_hub_2'].append(cc[1]); dt_['ua_pass_hub_1'].append(cc[0])

if nw_hub_dict[dt_['route']][i].split('_')[0].index('1')>=\
    nw_hub_dict[dt_['route']][i].split('_')[1].index('1'):
    cc=nw_hub_dict[dt_['route']][i].split('_')[0]
else: cc=nw_hub_dict[dt_['route']][i].split('_')[1]

dt_['nw_pass_hub_4'].append(cc[3]); dt_['nw_pass_hub_3'].append(cc[2])
dt_['nw_pass_hub_2'].append(cc[1]); dt_['nw_pass_hub_1'].append(cc[0])

if dl_hub_dict[dt_['route']][i].split('_')[0].index('1')>=\
    dl_hub_dict[dt_['route']][i].split('_')[1].index('1'):
    cc=dl_hub_dict[dt_['route']][i].split('_')[0]
else: cc=dl_hub_dict[dt_['route']][i].split('_')[1]

dt_['dl_pass_hub_4'].append(cc[3]); dt_['dl_pass_hub_3'].append(cc[2])
dt_['dl_pass_hub_2'].append(cc[1]); dt_['dl_pass_hub_1'].append(cc[0])

if co_hub_dict[dt_['route']][i].split('_')[0].index('1')>=\

```

```

co_hub_dict[dt_['route'][i].split('_')[1]].index('1'):
    cc=co_hub_dict[dt_['route'][i].split('_')[0]]
else: cc=co_hub_dict[dt_['route'][i].split('_')[1]]

dt_['co_pass_hub_4'].append(cc[3]); dt_['co_pass_hub_3'].append(cc[2])
dt_['co_pass_hub_2'].append(cc[1]); dt_['co_pass_hub_1'].append(cc[0])

flag=0

hub_major_0=['0']*18; hub_major_1=['0']*18

if close_dict_0_50[dt_['route'][i].split('_')[0]]!=[]:
    hub_major_0=[dt_['co_pass_hub_4'][i],dt_['co_pass_hub_3'][i],\
        dt_['co_pass_hub_2'][i],dt_['dl_pass_hub_4'][i],\
        dt_['dl_pass_hub_3'][i],dt_['dl_pass_hub_2'][i],\
        dt_['nw_pass_hub_4'][i],dt_['nw_pass_hub_3'][i],\
        dt_['nw_pass_hub_2'][i],dt_['ua_pass_hub_4'][i],\
        dt_['ua_pass_hub_3'][i],dt_['ua_pass_hub_2'][i],\
        dt_['us_pass_hub_4'][i],dt_['us_pass_hub_3'][i],\
        dt_['us_pass_hub_2'][i],dt_['aa_pass_hub_4'][i],\
        dt_['aa_pass_hub_3'][i],dt_['aa_pass_hub_2'][i]]

if close_dict_0_50[dt_['route'][i].split('_')[1]]!=[]:
    hub_major_1=[dt_['co_pass_hub_4'][i],dt_['co_pass_hub_3'][i],\
        dt_['co_pass_hub_2'][i],dt_['dl_pass_hub_4'][i],\

```

```

dt_['dl_pass_hub_3'][i],dt_['dl_pass_hub_2'][i],\
dt_['nw_pass_hub_4'][i],dt_['nw_pass_hub_3'][i],\
dt_['nw_pass_hub_2'][i],dt_['ua_pass_hub_4'][i],\
dt_['ua_pass_hub_3'][i],dt_['ua_pass_hub_2'][i],\
dt_['us_pass_hub_4'][i],dt_['us_pass_hub_3'][i],\
dt_['us_pass_hub_2'][i],dt_['aa_pass_hub_4'][i],\
dt_['aa_pass_hub_3'][i],dt_['aa_pass_hub_2'][i]]

if close_dict_0_50[dt_['route'][i].split('_')[0]]!=[] or\
close_dict_0_50[dt_['route'][i].split('_')[1]]!=[]:
    hub_major=hub_major_0+hub_major_1
    if hub_major.count('1')!=0: flag=1

if flag==1: dt_['hub_0_50'].append('1')
else: dt_['hub_0_50'].append('0')

flag=0

hub_major_0=['0']*18; hub_major_1=['0']*18

if close_dict_50_100[dt_['route'][i].split('_')[0]]!=[]:
    hub_major_0=[dt_['co_pass_hub_4'][i],dt_['co_pass_hub_3'][i],\
dt_['co_pass_hub_2'][i],dt_['dl_pass_hub_4'][i],\
dt_['dl_pass_hub_3'][i],dt_['dl_pass_hub_2'][i],\
dt_['nw_pass_hub_4'][i],dt_['nw_pass_hub_3'][i],\

```

```

dt_['nw_pass_hub_2'][i],dt_['ua_pass_hub_4'][i],\
dt_['ua_pass_hub_3'][i],dt_['ua_pass_hub_2'][i],\
dt_['us_pass_hub_4'][i],dt_['us_pass_hub_3'][i],\
dt_['us_pass_hub_2'][i],dt_['aa_pass_hub_4'][i],\
dt_['aa_pass_hub_3'][i],dt_['aa_pass_hub_2'][i]]

if close_dict_50_100[dt_['route'][i].split('_')[1]]!=[]:
    hub_major_1=[dt_['co_pass_hub_4'][i],dt_['co_pass_hub_3'][i],\
dt_['co_pass_hub_2'][i],dt_['dl_pass_hub_4'][i],\
dt_['dl_pass_hub_3'][i],dt_['dl_pass_hub_2'][i],\
dt_['nw_pass_hub_4'][i],dt_['nw_pass_hub_3'][i],\
dt_['nw_pass_hub_2'][i],dt_['ua_pass_hub_4'][i],\
dt_['ua_pass_hub_3'][i],dt_['ua_pass_hub_2'][i],\
dt_['us_pass_hub_4'][i],dt_['us_pass_hub_3'][i],\
dt_['us_pass_hub_2'][i],dt_['aa_pass_hub_4'][i],\
dt_['aa_pass_hub_3'][i],dt_['aa_pass_hub_2'][i]]

if close_dict_50_100[dt_['route'][i].split('_')[0]]!=[] or\
close_dict_50_100[dt_['route'][i].split('_')[1]]!=[]:
    hub_major=hub_major_0+hub_major_1
    if hub_major.count('1')!=0: flag=1

if flag==1: dt_['hub_50_100'].append('1')
else: dt_['hub_50_100'].append('0')

```

```
flag=0
```

```
hub_major_0=['0']*18; hub_major_1=['0']*18
```

```
if close_dict_100_150[dt_['route']][i].split('_')[0]]!=[]:
```

```
    hub_major_0=[dt_['co_pass_hub_4']][i],dt_['co_pass_hub_3']][i],\
                  dt_['co_pass_hub_2']][i],dt_['dl_pass_hub_4']][i],\
                  dt_['dl_pass_hub_3']][i],dt_['dl_pass_hub_2']][i],\
                  dt_['nw_pass_hub_4']][i],dt_['nw_pass_hub_3']][i],\
                  dt_['nw_pass_hub_2']][i],dt_['ua_pass_hub_4']][i],\
                  dt_['ua_pass_hub_3']][i],dt_['ua_pass_hub_2']][i],\
                  dt_['us_pass_hub_4']][i],dt_['us_pass_hub_3']][i],\
                  dt_['us_pass_hub_2']][i],dt_['aa_pass_hub_4']][i],\
                  dt_['aa_pass_hub_3']][i],dt_['aa_pass_hub_2']][i]]
```

```
if close_dict_100_150[dt_['route']][i].split('_')[1]]!=[]:
```

```
    hub_major_1=[dt_['co_pass_hub_4']][i],dt_['co_pass_hub_3']][i],\
                  dt_['co_pass_hub_2']][i],dt_['dl_pass_hub_4']][i],\
                  dt_['dl_pass_hub_3']][i],dt_['dl_pass_hub_2']][i],\
                  dt_['nw_pass_hub_4']][i],dt_['nw_pass_hub_3']][i],\
                  dt_['nw_pass_hub_2']][i],dt_['ua_pass_hub_4']][i],\
                  dt_['ua_pass_hub_3']][i],dt_['ua_pass_hub_2']][i],\
                  dt_['us_pass_hub_4']][i],dt_['us_pass_hub_3']][i],\
                  dt_['us_pass_hub_2']][i],dt_['aa_pass_hub_4']][i],\
                  dt_['aa_pass_hub_3']][i],dt_['aa_pass_hub_2']][i]]
```

```

if close_dict_100_150[dt_['route'][i].split('_')[0]]!=[] or\
    close_dict_100_150[dt_['route'][i].split('_')[1]]!=[]:
    hub_major=hub_major_0+hub_major_1
    if hub_major.count('1')!=0: flag=1

if flag==1: dt_['hub_100_150'].append('1')
else: dt_['hub_100_150'].append('0')

flagg=0
if close_dict_0_50[dt_['route'][i].split('_')[0]]!=[] and\
    close_dict_0_50[dt_['route'][i].split('_')[1]]!=[]:
    for aa in close_dict_0_50[dt_['route'][i].split('_')[0]]:
        for bb in close_dict_0_50[dt_['route'][i].split('_')[1]]:
            if aa!=bb:
                close_list=[aa,bb]
                close_list.sort()
                if dt_['presence'][dt_['route'].index(close_list[0]+'_'+\
                    close_list[1])].split('\n')[0]=='1':
                    flagg=1
if flagg: dt_['nearby_route_0_50'].append('1')
else: dt_['nearby_route_0_50'].append('0')

flagg=0
if close_dict_50_100[dt_['route'][i].split('_')[0]]!=[] and\

```



```

close_dict_50_100[dt_['route'][i].split('_')[1]]!=[]:
    for aa in close_dict_50_100[dt_['route'][i].split('_')[0]]:
        for bb in close_dict_50_100[dt_['route'][i].split('_')[1]]:
            if aa!=bb:
                close_list=[aa,bb]
                close_list.sort()
                if dt_['presence'][dt_['route'].index(close_list[0]+\
                    '_'+close_list[1])].split('\n')[0]=='1':
                    flagg=1
if flagg: dt_['nearby_route_50_100'].append('1')
else: dt_['nearby_route_50_100'].append('0')

"""
flagg=0
if close_dict_100_150[dt_['route'][i].split('_')[0]]!=[] and\
    close_dict_100_150[dt_['route'][i].split('_')[1]]!=[]:
    for aa in close_dict_100_150[dt_['route'][i].split('_')[0]]:
        for bb in close_dict_100_150[dt_['route'][i].split('_')[1]]:
            if aa!=bb:
                close_list=[aa,bb]
                close_list.sort()
                if dt_['presence'][dt_['route'].index(close_list[0]+\
                    '_'+close_list[1])].split('\n')[0]=='1':
                    flagg=1
if flagg: dt_['nearby_route_100_150'].append('1')

```

```

else: dt_['nearby_route_100_150'].append('0')
"""

flagg=0
if close_dict_0_50[dt_['route'][i].split('_')[0]]!=[] and not\
    (close_dict_0_50[dt_['route'][i].split('_')[0]]!=[] and\
     close_dict_0_50[dt_['route'][i].split('_')[1]]!=[]):
    for aa in close_dict_0_50[dt_['route'][i].split('_')[0]]:
        if aa!=dt_['route'][i].split('_')[1]:
            close_list=[aa,dt_['route'][i].split('_')[1]]
            close_list.sort()
            if dt_['presence'][dt_['route'].index(close_list[0]+'_' +\
                close_list[1])].split('\n')[0]=='1':
                flagg=1
if close_dict_0_50[dt_['route'][i].split('_')[1]]!=[] and not\
    (close_dict_0_50[dt_['route'][i].split('_')[0]]!=[] and\
     close_dict_0_50[dt_['route'][i].split('_')[1]]!=[]):
    for aa in close_dict_0_50[dt_['route'][i].split('_')[1]]:
        if aa!=dt_['route'][i].split('_')[0]:
            close_list=[aa,dt_['route'][i].split('_')[0]]
            close_list.sort()
            if dt_['presence'][dt_['route'].index(close_list[0]+'_' +\
                close_list[1])].split('\n')[0]=='1':
                flagg=1
if flagg: dt_['nearby_endpoint_0_50'].append('1')
```

```

else: dt_['nearby_endpoint_0_50'].append('0')

flagg=0
if close_dict_50_100[dt_['route'][i].split('_')[0]]!=[] and not\
    (close_dict_50_100[dt_['route'][i].split('_')[0]]!=[] and\
     close_dict_50_100[dt_['route'][i].split('_')[1]]!=[]):
    for aa in close_dict_50_100[dt_['route'][i].split('_')[0]]:
        if aa!=dt_['route'][i].split('_')[1]:
            close_list=[aa,dt_['route'][i].split('_')[1]]
            close_list.sort()
            if dt_['presence'][dt_['route'].index(close_list[0]+'_'+\
                close_list[1])].split('\n')[0]=='1':
                flagg=1
if close_dict_50_100[dt_['route'][i].split('_')[1]]!=[] and not\
    (close_dict_50_100[dt_['route'][i].split('_')[1]]!=[] and\
     close_dict_50_100[dt_['route'][i].split('_')[0]]!=[]):
    for aa in close_dict_50_100[dt_['route'][i].split('_')[1]]:
        if aa!=dt_['route'][i].split('_')[0]:
            close_list=[aa,dt_['route'][i].split('_')[0]]
            close_list.sort()
            if dt_['presence'][dt_['route'].index(close_list[0]+'_'+\
                close_list[1])].split('\n')[0]=='1':
                flagg=1
if flagg: dt_['nearby_endpoint_50_100'].append('1')
else: dt_['nearby_endpoint_50_100'].append('0')

```

```

"""
flagg=0
if close_dict_100_150[dt_['route'][i].split('_')[0]]!=[] and not\
    (close_dict_100_150[dt_['route'][i].split('_')[0]]!=[] and\
    close_dict_100_150[dt_['route'][i].split('_')[1]]!=[]):
    for aa in close_dict_100_150[dt_['route'][i].split('_')[0]]:
        if aa!=dt_['route'][i].split('_')[1]:
            close_list=[aa,dt_['route'][i].split('_')[1]]
            close_list.sort()
            if dt_['presence'][dt_['route'].index(close_list[0]+'_'+\
                close_list[1])].split('\n')[0]=='1':
                flagg=1
if close_dict_100_150[dt_['route'][i].split('_')[1]]!=[] and not\
    (close_dict_100_150[dt_['route'][i].split('_')[0]]!=[] and\
    close_dict_100_150[dt_['route'][i].split('_')[1]]!=[]):
    for aa in close_dict_100_150[dt_['route'][i].split('_')[1]]:
        if aa!=dt_['route'][i].split('_')[0]:
            close_list=[aa,dt_['route'][i].split('_')[0]]
            close_list.sort()
            if dt_['presence'][dt_['route'].index(close_list[0]+'_'+\
                close_list[1])].split('\n')[0]=='1':
                flagg=1
if flagg: dt_['nearby_endpoint_100_150'].append('1')
else: dt_['nearby_endpoint_100_150'].append('0')

```

```
"""
```

```
dt_['max_dep_delay'].append(str(max(dep_delay_dict\
    [dt_['route'][i].split('_')[0]][fixed_year[year]],\
    dep_delay_dict[dt_['route'][i].split('_')[1]]\
    [fixed_year[year]])))
dt_['max_arr_delay'].append(str(max(arr_delay_dict\
    [dt_['route'][i].split('_')[0]][fixed_year[year]],\
    arr_delay_dict[dt_['route'][i].split('_')[1]]\
    [fixed_year[year]])))
dt_['max_delay'].append(str(max(max(dep_delay_dict\
    [dt_['route'][i].split('_')[0]][fixed_year[year]],\
    dep_delay_dict[dt_['route'][i].split('_')[1]]\
    [fixed_year[year]]),\
    max(arr_delay_dict[dt_['route'][i].split('_')[0]]\
    [fixed_year[year]],\
    arr_delay_dict[dt_['route'][i].split('_')[1]]\
    [fixed_year[year]]))))

dt_['n_'].append(str(float(dt_['number'][i])-float(dt_['presence'][i])))
dt_['aa_pass_hub'].append(str(float(dt_['aa_pass_hub_2'][i])+
    float(dt_['aa_pass_hub_3'][i])+float(dt_['aa_pass_hub_4'][i])))
dt_['us_pass_hub'].append(str(float(dt_['us_pass_hub_2'][i])+
    float(dt_['us_pass_hub_3'][i])+float(dt_['us_pass_hub_4'][i])))
dt_['ua_pass_hub'].append(str(float(dt_['ua_pass_hub_2'][i])+
```

```

        float(dt_['ua_pass_hub_3'][i])+float(dt_['ua_pass_hub_4'][i]))
dt_['nw_pass_hub'].append(str(float(dt_['nw_pass_hub_2'][i])+
        float(dt_['nw_pass_hub_3'][i])+float(dt_['nw_pass_hub_4'][i]))
dt_['dl_pass_hub'].append(str(float(dt_['dl_pass_hub_2'][i])+
        float(dt_['dl_pass_hub_3'][i])+float(dt_['dl_pass_hub_4'][i]))
dt_['co_pass_hub'].append(str(float(dt_['co_pass_hub_2'][i])+
        float(dt_['co_pass_hub_3'][i])+float(dt_['co_pass_hub_4'][i]))

av_dict={}
for kk in route_dist_dict.keys():
    count_=0
    for ll in route_dist_dict[kk]:
        count_+=(float(ll[1])/float(route_pass_dict[kk]))*float(ll[0])
    av_dict[kk]=count_

r_=6372.7974775959065
for i in range(len(dt_['carrier'])):
    c_r=[dt_['route'][i].split('_')[0],dt_['route'][i].split('_')[1]]
    c_r.sort()

    if c_r[0] in av_dict.keys(): ccr0=float(av_dict[c_r[0]])
    else: ccr0=0.0

    if c_r[1] in av_dict.keys(): ccr1=float(av_dict[c_r[1]])
    else: ccr1=0.0

```

```

p1,l1=(pi/180)*airport_location(c_r[0])[0],\
      (pi/180)*airport_location(c_r[0])[1]
p2,l2=(pi/180)*airport_location(c_r[1])[0],\
      (pi/180)*airport_location(c_r[1])[1]
l=abs(l1-l2)
num=sqrt(((cos(p2)*sin(l))**2)+(((cos(p1)*sin(p2))-\\
      (sin(p1)*cos(p2)*cos(l))))**2))
den=sin(p1)*sin(p2)+cos(p1)*cos(p2)*cos(l)
theta=arctan(num/den)
distance__=abs(int(round(r_*theta)))
dt_['distance_ratio'].append(str(mean([ccr0,ccr1])/float(distance__)))

if passenger_filter_flag:
    filename3='WN_actual_routes_'+year+'_'+quarter+'_passfilt_.txt'
elif not passenger_filter_flag:
    filename3='WN_actual_routes_'+year+'_'+quarter+'_.txt'

t03=time.time(); print; print 'Writing output text file: '+filename3
g=open(data_path+filename3,'w')
for j in range(len(dt_['carrier'])):
    output=''
    for i in keys__:
        output=output+dt_[i][j]
        if output[-1]!='\\n': output=output+'\\t'

```

```

        if output[-1]!='\n': output=output+'\n'
        g.write(output)
    print str(time.time()-t03)+' seconds to write text file.'
    g.close()

print 'Ordered variable list: '; print
for i in keys__: print i

*****

# delay_parse_2

"""

Parse U.S. airport delay data, for WN entry model project.
(version 2)

Steve Lawford: 16 January 2009.
Updated 17 January 2009 with arrival delay data.

"""

import cPickle,time,os,sys

module_dir=os.path.join('Project','code')

```



```
sys.path.insert(0,module_dir)

from safe_cPickle import *
from scipy import *

data_path='C:/Project/data/'
filename='departure_delay_data_2001_2007.txt'

t0=time.time(); print; print 'Opening file: '+filename
f=open(data_path+filename,'r')
print 'Done. '+str(time.time()-t0)+' seconds to open file'

agg_delay_dict={}
for i in f:
    a=i.split('\t')
    if len(a[0])==3:
        if a[0] not in agg_delay_dict.keys():
            b=a[2:7]+[a[7].split('\n')[0]]; c=[]
            for l in b: c.append(eval(l))
            agg_delay_dict[a[0]]=c

safe_cPickle_dump(data_path,'dep_delay_dict',agg_delay_dict)

filename='arrival_delay_data_2001_2007.txt'
```

```
t0=time.time(); print; print 'Opening file: '+filename
f=open(data_path+filename,'r')
print 'Done. '+str(time.time()-t0)+' seconds to open file'
```

```
agg_delay_dict={}
for i in f:
    a=i.split('\t')
    if len(a[0])==3:
        if a[0] not in agg_delay_dict.keys():
            b=a[2:7]+[a[7].split('\n')[0]]; c=[]
            for l in b: c.append(eval(l))
            agg_delay_dict[a[0]]=c
```

```
safe_cPickle_dump(data_path,'arr_delay_dict',agg_delay_dict)
```

```
*****
```

```
# lcc_parse
```

```
"""
```

```
Parse U.S. airport data, for WN entry model project.
```

```
- JetBlue, Frontier, AirTran, Alaska.
```

```
Steve Lawford: 17 January 2009.
```

```
"""

import cPickle,time,os,sys

module_dir=os.path.join('Project','code')
sys.path.insert(0,module_dir)

from safe_cPickle import *
from scipy import *

data_path='C:/Project/data/'

filename='jetblue.txt'
t0=time.time(); print; print 'Opening file: '+filename
f=open(data_path+filename,'r')
print 'Done. '+str(time.time()-t0)+' seconds to open file'

lcc_dict={}
for i in f:
    a=i.split('\t'); b=[]
    if len(a[0])==3:
        c=[]
        #for j in a[1:]: b.append(float(j))
        for j in a[1:]: b.append((float(j)>0)*1)
```

```

(float(j)>0)*1
"""

c=[mean(b[1:5]),mean(b[5:9]),mean(b[9:13]),\
   mean(b[13:17]),mean(b[17:21]),mean(b[21:25])]
"""

if a[0] not in lcc_dict.keys():
    #lcc_dict[a[0]]=c
    lcc_dict[a[0]]=b
print lcc_dict

safe_cPickle_dump(data_path,'jetblue_dict',lcc_dict)

filename='frontier.txt'
t0=time.time(); print; print 'Opening file: '+filename
f=open(data_path+filename,'r')
print 'Done. '+str(time.time()-t0)+' seconds to open file'

lcc_dict={}
for i in f:
    a=i.split('\t'); b=[]
    if len(a[0])==3:
        c=[]
        for j in a[1:]: b.append(float(j))
        """

        c=[mean(b[1:5]),mean(b[5:9]),mean(b[9:13]),\

```

```

        mean(b[13:17]),mean(b[17:21]),mean(b[21:25]))
    """
    if a[0] not in lcc_dict.keys():
        #lcc_dict[a[0]]=c
        lcc_dict[a[0]]=b
print lcc_dict

safe_cPickle_dump(data_path,'frontier_dict',lcc_dict)

filename='alaska.txt'
t0=time.time(); print; print 'Opening file: '+filename
f=open(data_path+filename,'r')
print 'Done. '+str(time.time()-t0)+' seconds to open file'

lcc_dict={}
for i in f:
    a=i.split('\t'); b=[]
    if len(a[0])==3:
        c=[]
        for j in a[1:]: b.append(float(j))
        """
        c=[mean(b[1:5]),mean(b[5:9]),mean(b[9:13]),\
            mean(b[13:17]),mean(b[17:21]),mean(b[21:25])]
        """
    if a[0] not in lcc_dict.keys():

```

```
#lcc_dict[a[0]]=c
lcc_dict[a[0]]=b
print lcc_dict

safe_cPickle_dump(data_path,'alaska_dict',lcc_dict)

filename='airtran.txt'
t0=time.time(); print; print 'Opening file: '+filename
f=open(data_path+filename,'r')
print 'Done. '+str(time.time()-t0)+' seconds to open file'

lcc_dict={}
for i in f:
    a=i.split('\t'); b=[]
    if len(a[0])==3:
        c=[]
        for j in a[1:]: b.append(float(j))
        """
        c=[mean(b[1:5]),mean(b[5:9]),mean(b[9:13]),\
            mean(b[13:17]),mean(b[17:21]),mean(b[21:25])]
        """
        if a[0] not in lcc_dict.keys():
            #lcc_dict[a[0]]=c
            lcc_dict[a[0]]=b
print lcc_dict
```

```
safe_cPickle_dump(data_path,'airtran_dict',lcc_dict)
```

```
*****
```

```
# service_since_parse
```

```
"""
```

```
Parse WN airport service duration data, for WN entry model project.
```

```
Steve Lawford: 18 January 2009.
```

```
"""
```

```
import cPickle,time,os,sys
```

```
module_dir=os.path.join('Project','code')
```

```
sys.path.insert(0,module_dir)
```

```
from safe_cPickle import *
```

```
from scipy import *
```

```
data_path='C:/Project/data/'
```

```
filename='service_since.txt'

t0=time.time(); print; print 'Opening file: '+filename
f=open(data_path+filename,'r')

print 'Done. '+str(time.time()-t0)+' seconds to open file'

since_dict={}

for i in f:
    a=i.split('\t'); b=[]
    if len(a[0])==3:
        c=[]
        for j in a[1:]: b.append(float(j))
        if a[0] not in since_dict.keys():
            since_dict[a[0]]=b
print since_dict

safe_cPickle_dump(data_path,'service_since_dict',since_dict)
```

For helpful feedback and discussions, we are grateful to Roseline Bilina, Marie-Claude Beaudes and Nathalie Lenoir. This paper was typed in MiKTeX and econometric results were derived using Eviews 6. The original .csv-datasets from the *DOT*, *BTS*, *OMB* and the *U.S. Census Bureau* were parsed into Eviews-acceptable formats using Python.

References

- [1] **International Air Transport Association,**
World Air Transport Statistics, 52nd edition, 2007
<http://www.iata.org/ps/publications/wats-passenger-carried.htm>; 2008

- [2] **Southwest Airlines**
www.southwest.com; 2008

- [3] **Research And Innovative Technology Administration,**
Bureau of Transportation Statistics
<http://www.transtats.bts.gov/airports.asp>; 2008

- [4] **Research And Innovative Technology Administration,**
Bureau of Transportation Statistics
<http://www.rita.dot.gov/>; 2008

- [5] **Goolsbee, A. and Syverson, C.**
“How Do Incumbents Respond to the Threat of Entry? Evidence from Major Airlines”, 2006
Quarterly Journal of Economics, November 2008

- [6] **Pitfield, D.E.**
“The Southwest effect: A time-series analysis on passengers carried by selected routes and a market share comparison”
Journal of Air Transport Management (14), 2008, 113 - 122

- [7] **Vowles, T.M.**
“The Southwest effect in multi airport regions”
Journal of Air Transport Management (7), 2001, 251 - 258

- [8] **U.S. Department of Transportation**
www.dot.gov; 2008
- [9] **Sky West Airlines, 2008**
http://www.skywest.com; 2009
- [10] **The Boeing Company**
“**Boeing 737 Aircraft Performance for airport planning**”, 2007
http://www.boeing.com/commercial/airports/acaps/737sec3.pdf; 2008
- [11] **Great Circle Mapper**
http://gc.kls2.com/; 2008
- [12] “**Southwest Airlines to buy Morris Air**”, 1993
http://query.nytimes.com; 2008
- [13] **Ito, H. and Lee, D.**
“**Low Cost Carrier Growth in the U.S. Airline Industry: Past, Present, and Future**”, 2003
Brown University; 2003
- [14] **de Oliveira, A.V.M.**
“**Estimation of a model for low cost carrier entry**”, 2004
University of Warwick
- [15] **Hsin Lin, M.**
“**Airline alliances and entry deterrence**”, 2007
Osaka University of Economics
- [16] **Bureau of Transport Statistics, 2008**
http://www.bts.gov/; 11/2008

- [17] **Federal Aviation Administration, 2008**
<http://www.faa.gov/airports/categories/; 10/2008>

- [18] **Lee, D. and Ito, H.**
“Incumbent response to lower cost entry: Evidence from the U.S. airline industry”, 2003
Brown University; 2003

- [19] **Bachtel, B.**
“ETOPS, Extended Operations, and En Route Alternate Airports”, 2003
<http://www.boeing.com/commercial/airports/faqs/etopseropsenroutealt.pdf; 11/2008>

- [20] **“Southwest moves away from its low-cost-model”, 2008**
<http://www.airliners.de/news/artikelseite.php?articleid=16710; 11/2008>
(original article in German)

- [21] **Field, D.**
“Southwest’s model shift”, 2007
Flightglobal, December 2007, 14

- [22] **“ATA’s code-share deal with Southwest aids both”, 2006**
<http://www.usatoday.com; 11/2008>

- [23] **ATA Codeshare Information, 2008**
<http://www.southwest.com; 11/2008>

- [24] **Dallas Love Field (DAL) Service, 2005**
<http://www.aa.com; 11/2008>

- [25] **Gloabl Legal Information Network, Wright Amendment Reform Act, 2006**
www.glin.gov; 11/2008
- [26] **Bogulaski, C., Ito, H. and Lee, D.**
“Entry patterns in the Southwest Airlines Route System”, 2004
Review of Industrial Organization, Volume 25
- [27] **Lederman, M. and Januszewski, S.**
“Entry Patterns of Low Cost Airlines”, 2003
Massachusetts Institute of Technology, 2003
- [28] **Information about fleet and hubs of American Airlines, Continental Airlines, Delta Air Lines, Northwest Airlines, United Airlines and USAirways, 2008**
www.aa.com; www.continental.com; www.delta.com; www.nwa.com; www.united.com; www.usairways.com; 11/2008
- [29] **“Southwest’s first losses for years”, 2008**
www.airliners.de/news/artikelseite.php?articleid=16517; 11/2008
(original article in German)
- [30] **“Can fuel hedges keep Southwest in the money?”, 2008**
www.usatoday.com/money/industries/travel/2008-07-23; 11/2008
- [31] **“Southwest returns to SFO after six-year absence.”, 2007**
www.sfgate.com/cgi-bin/article.cgi?f=/c/a/2007/08/28/BUA1RQ7B5.DTL; 11/2008

- [32] **“Southwest Airlines 2006 Annual Report”, 2007**
www.southwest.com/investorrelations/annualreports; 12/2008
- [33] **Nonstop routes of American Airlines from its MIA hub, 2007**
http://www.airlineroutemaps.com; 11/2008
- [34] **Sample travel information for a flight FLL to SFO and back on Southwest Airlines for 23th/ 25th January 2009**
www.southwest.com/cgi-bin/buildItinerary; 1/2009
- [35] **“Southwest coming to La Guardia”, 2008**
www.nj.com/newark/index.ssf; 11/2008
- [36] **“Legal Analysis of the FAA’s Slot Auction Rule for JFK and Newark Part 1”, 2008**
www.aviationairportdevelopmentlaw.com; 11/2008
- [37] **Data on population by Metropolitan Statistical Areas, 2008**
www.census.gov; 11/2008
- [38] **AirTran Airways route map from ATL hub, 2007**
www.airlineroutemaps.com; 1/2009
- [39] **Frontier Airlines route map, 2009**
www.frontierairlines.com; 1/2009
- [40] **JetBlue Airways route map, 2008**
www.jetblue.com; 1/2009
- [41] **Alaska Airlines route map, 2008**
www.alaskaair.com; 1/2009

- [42] **Data on GDP by Metropolitan Statistical Areas, 2008**

www.bea.gov; 11/2008

- [43] **Greene, William H.**

**“Econometric Analysis”, Fourth Edition, Prentice Hall
International Inc.**

New York University

- [44] **U.S. Department of Labor, Bureau of Labor statistics**

www.bls.gov/iag/tgs/iag54.htm; 12/2008

List of Figures

1	Representative range chart	4
2	Cities served by Southwest	6
3	Network diagram	9
4	Network development	10
5	R^2 values for 2002 to 2007 (unfiltered data)	50
6	$\hat{\beta}$ value for all variables (unfiltered data)	52
7	$\hat{\beta}$ value for all variables on a normalized y-axis (unfiltered data)	53
8	MAX_Delay plotted across the 1461 routes	54
9	Marginal effects	56
10	Marginal effects of low cost carriers	58
11	all coefficients unfiltered data	60
12	all coefficients filtered data	60
13	Marginal effects filtered	61
14	Marginal effects unfiltered	62
15	Marginal effects filtered	62
16	R^2 for the filtered data	63
17	Performance	67
18	Fitted probabilities for Southwest's presence	68
19	Fitted probabilities for Southwest not being present	69
20	Fitted probabilities for routes from/to LGA	71
21	Fitted probabilities for routes from/to MSP	72
22	Fitted probabilities for routes from/to MDW	73
23	Fitted probabilities for routes from/to DEN	74
24	Fitted probabilities for routes from/to BNA	74

25	Fitted probabilities for routes from/to LAX	75
26	Fitted probabilities for routes from/to SFO	76
27	Fitted probabilities for routes from/to DTW	76
28	U.S. air traffic hubs in 2008	79
29	AirTran	81
30	Frontier	82
31	JetBlue	83
32	Alasla	84
33	MSAs and mSAs 2007	91
34	Python Flowchart	113

List of Tables

1	Southwest fleet composition	2
2	Top 10 U.S. airlines	3
3	Southwest's airports	7
4	Southwest market share July 2007 - August 2008	11
5	Regional division variable	16
6	Results from Model 1	33
7	Results from Model 2	34
8	Results from Model 2	35
9	Covariance data matrix for Model 3 before first auxiliary regression	36
10	Covariance data matrix for Model 3 after first auxiliary regression	36
11	Results for Model 3 without quadratic terms	36
12	Covariance data matrix for Model 3 with quadratic terms, before second auxiliary regression	37
13	Covariance data matrix for Model 3 with quadratic terms, after second auxiliary regression	37
14	Results for Model 3 with quadratic terms, after second auxiliary regression	38
15	Results for Model 4	39
16	Results of Model 5	40
17	Results of Model 6	41
18	Results of Model 7	43
19	Covariance data matrix for Model 8 before auxiliary regression .	44
20	Results of Model 8	45
21	Results of Model 8 after the auxiliary regression	46

22	Results of Model 9	48
23	Top three regions Southwest wants to connect (2005Q2)	56
24	Last three regions Southwest wants to connect (2005Q2)	57
25	Results of the entire dataset	65
26	Correlation between performance variables.	67
33	Fleet information and hubs of the U.S. <i>Major carriers</i>	86
34	MSAs, mSAs and airports	97