DK9901201

PETRA

- Data

PETRA Working Paper no. 8

September 1998

COWI

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

COWI

PETRA

- Data

PETRA Working Paper no. 8

September 1998

Report no.: 2
Issue no.: 2
Date of issue: 1998-09-30

Prepared: JJD
Checked: NBU
Approved: MOF

Table of Contents

1 Introduction

This paper describes the data that have been employed within the PETRA model. The basic data set is the TU, comprising both trip diaries, and individual and household information, collected during 1995. The TU data have been supplemented both by generalised travel cost (GTC) and level of service (LoS) data, and by population and employment data to measure the relative attractiveness of alternative destinations.

The TU data only consist of data (for instance distance and time) on the trips actually undertaken by the interviewed persons. The establishment of the choice model requires the same information on the alternative trips that were possible for the IP. Thus, register based data (GIS data) are utilised to calculate distance and travel time between different geographical locations.

Additional register based data are also employed to characterise both the chosen, and the alternative locations. The service level of public transport and the number of employees in 12 different branches are used as attraction variables.

Finally, a minor ad hoc study has been carried out with a view to describing, in more detail, the distance, time and costs of available public transport.

2 Data Sources

2.1 TU Data

The data are gathered by telephone interview with at least 1000 people on a monthly basis. The participants are drawn at random, to provide a representative sample of the total population between the ages of 16 to 74 years. The actual number interviewed varies according to the month: 1,800 persons were sampled in February, March, May, June, August, September, November and December, and 2,300 persons were sampled in January, April and October. There are no interviews conducted in the month of July.

The response rate varies between 61% and 66%, resulting in approximately 1,100 interviews in those months with a smaller sample, and approximately 1,500 interviews in those months with a sample size of 2,300 persons.

There is little information available on non-respondents. However, weights have been calculated to correct for the representativity of the sample. If non-response has lead to systematic skewness in the sample, these weights will partially correct for the resulting (PETRA Working Paper no. 3, 1997) .

The TU data employed, cover the period from January 1995 to December 1995, and are the result of the 13,793 telephone interviews undertaken during this period.

The interviewees (referred to as IP for interview person) were asked questions about the trips made the day before the interview, and longer trips (more than 100 km) in the month before the interview. A number of additional questions are asked about the IP and the family. Figure 2-1 shows the data structure.

Figure 2-1. Data structure

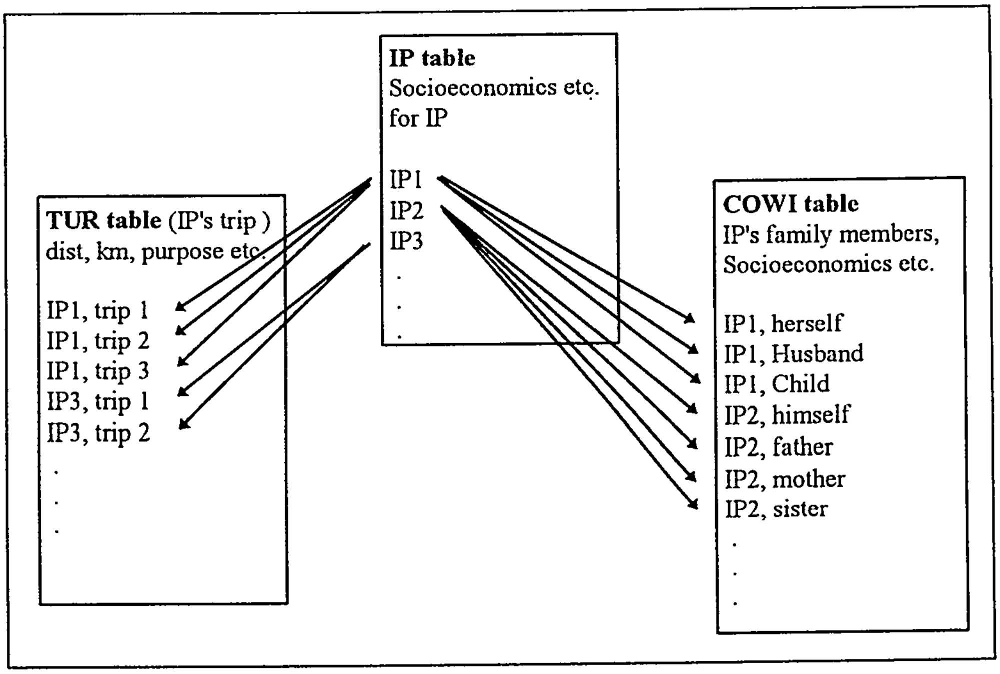


Table 2-1 shows descriptives of a few of the variables of the 13.793 IPs interviewed in 1995.

Table 2-1. Descriptives of selected IP variables

	Number of valid obs	Minimum	Maximum	Mean	Std. Deviation
ALDER (age)	13793	16	74	42.18335	15.63446
ARBTID (worktime)	13636	0	96	24.64711	19.83019
DAG (weekday)	13793	1	7	3.98898	2.000477
KNIP (gender)	13793	1	2	1.495614	0.499999
URBAN (urbanization)	13793	11	70	43.44066	18.83319
GAABUS (walk time, bus)	13759	1	8	1.701141	0.956044
GAASTA (walk time, train)	13773	1	8	3.80302	1.362398

Note: Only IP interviewed in 1995 are included.

Age ranges from 16 to 74, since only people at this age was interwieved. Worktime ranges from zero to 96 hours per week, but only few people has beond 40 hours per week. The reason why some people reports many hours of work every week could be due to people including travel time to and from work in the reported working time.

The gender of the IP is coded 1 for male and 2 for female. As can be seen by the average of KNIP the proportion of women and men interviewed are practically equal.

The urbanization level where IP lives is coded in URBAN. Table 2-2 shows the definition of the URBAN variable.

Table 2-2. Definition of urbanisation

Code	Definition	Frequency	Percent
11	Copenhagen, except	1238	8.975567
12	Frederiksberg	175	1.26876
20	Other Cph suburbs	1817	13.17335
30	>100,000 inhabitants	1229	8.910317
41	>70,000 inhabitants	196	1.421011
42	22-60,000 inhabitants	799	5.792793
43	22-60,000 inhabitants	951	6.894802
44	10-22,000 inhabitants	903	6.546799
50	2-10,000 inhabitants	2301	16.68238
60	200-2000 inhabitants	1990	14.42761
70	Rural areas	2194	15.90662
	Total	13793	100

DAG is the weekday of the interview. Table 2-3 shows the distribution of interviews over the week.

Table 2-3. Weekday of interview

	Frequency	Percent
Monday	2003	14.52186
Tuesday	1952	14.15211
Wednesday	1969	14.27536
Thursday	2001	14.50736
Friday	1975	14.31886
Saturday	1924	13.9491
Sunday	1969	14.27536
Total	13793	100

As can be seen from Table 2-3 the interviews are distributed with allmost excactly the same number of the different weekdays.

Table 2-4 shows the coding of the variable GAASTA, minutes to walk to nearest station.

Table 2-4. Minutes to walk to nearest train station

	Frequency	Percent
less than 5 minutes	862	6.249547
5-10 minutes	2597	18.82839
11-15 minutes	1418	10.28058
16-30 minutes	2423	17.56688
more than 30 minutes	6469	46.9006
Not imformed	4	0.029

Many people (47%) lives in places where it takes more than half an hour to walk to the train station. Only 25% lives within a working distance of less than 10 minutes to the nearest train station.

The family level

The TU interview defines the family as a unit comprising any family relations. Thus, grandparents and adult children living with their parents are considered to belong to the same family.

A narrower definition of the family is utilised in PETRA, consisting of one or two adults and possible children. Additional adults (grown up children or older parents) are considered to constitute a separate family.

Definition of household

The basis of the definition of the family in PETRA is the head of the family; if the family consists of a couple, then the head of the family is defined as the male partner. This definition follows the standard practice of Danmarks Statistik (DS). However, this definition is only used to construct the families, and in the later analyses (after the families are formed), the head of the family is defined as the individual with the highest income.

The cases of adult children living with their parents, are treated as separate, or auxiliary, families. The rationale is that if they have a job, and hence an income, their status regarding the demand for transport and decisions about buying a car are likely to be independent of their parents. This assumption also applies to single older parents living with middle aged children.

Thus, the data material have been divided into three types of families:

Single: families with one adult and with or without children. In this group we also place older parents living with their adult children, and younger adults who are the third (or higher) adult at this address with no family relations to the rest of the families, and therefore are most likely tenants.

Auxiliary: families consisting of adult children living with their parents.

Couples: families with two adults and with or without children, the traditional family.

2.1.1 Zone System

The interviewer requests the IP to describe any trips undertaken the previous day. The description contains both mode, time, distance and the actual location for the destination. These destinations are registered as zones - here after called TU zones.

The TU zone definition is based on a city number, 1,411 cities, and major cities are split up in small parts as separate zones. In addition, the rural area surrounding each city is considered a separate zone.

The destination zone for each trip is determined during the interview; the IP is first asked if the destination was the centre of a city, another part of a city, or if the destination was a rural area. In this way, the specified geographic destination is determined by the IPs perception of the destination, allowing for different perception of destinations between respondents, e.g. their perception of what constitutes "rural" areas.

Four levels of zones have been used in the TDM.

1. Main areas of Denmark, 3 zones
2. Commuting regions, 38 zones
3. Municipality, 275 zones
4. TU zones, 3010 zones

Denmark is split into three parts - Bornholm, Sjælland and Jylland/Fyn. The majority of the trips in the TU data material take place within these zones.

Table 2-5. Travel between parts of Denmark (TU 1995)

To From	Sjælland	Bornholm	Jylland	Sum
Sjælland	16352	1	103	16456
Bornholm		320		320
Jylland	90		22530	22620

Main areas of Denmark

Only 1 of 16,456 trips from Sjælland goes to Bornholm from elsewhere and 103 trips from Sjælland go to Jylland. There is a similar pattern for trips from Jylland.

In the following, only intra- main area trips are included, and since these trips constitute 99% of all trips, this limitation was considered unlikely to affect the results significantly.

Commuting Zones

At the second level, Denmark has been split up in 38 zones. This level is used when setting up strata for the sampling of destination alternatives.

Municipalities

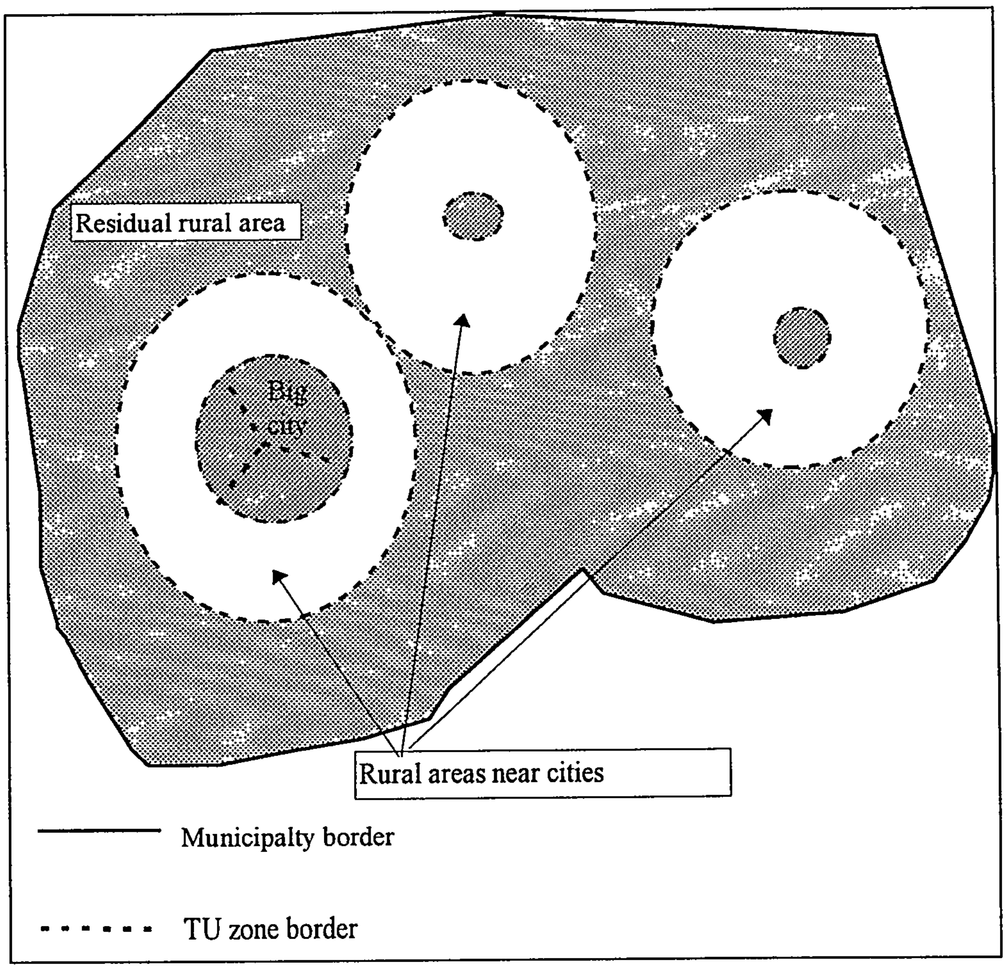
At the municipal level Denmark has been divided into the 275 municipalities.

TU zones

Each municipality has been divided into a number of areas centred around the cities in the municipality. Each city gets its own zone, and the area around the city is also considered a separate (rural) zone. Bigger cities are divided into a number of zones.

The example in Figure 2-2 shows how a typical municipality is divided into TU zones.

Figure 2-2 Example: Map of Municipality with zone boundaries indicated.



In the example, the municipality is divided into 9 TU zones. The biggest city in the municipality is divided into 3 TU zones, while the two smaller cities each form one zone. A rural zone is specified close to each of the cities, while the residual area in the municipality forms an additional rural TU zone.

Table 2-6 illustrates that each municipality is divided into approximately 10 TU zones. In addition, there are more TU zones per municipality east of Storebælt.

Table 2-6 TU zones and municipalities in Denmark

Main area	No of municipalities	No. of cities	No of TU zones	TU zones per municipality
Bornholm	5	21	42	8.3
Jylland/Fyn	173	1045	2160	8.4
Sjælland	97	345	808	12.5
Total	275	1411	3010	10.9

The area of the TU zones averages 14 square kilometres. However, city zones are much smaller, and rural zones bigger, relative to the average.

The most disaggregated level is the TU zones, which form the basis for the calculation of generalised travel costs and the level of service variables (GTC and LoS).

2.1.2 Travel Purposes

The IP is asked about the purpose for each of the specified trips, and can choose between 18 possible answers. These 18 different purposes are grouped into 3 main purposes (and home) for use in the TDM. Table 2-7 shows the purposes.

Table 2-7 Contents of purposes

Main purpose	Detailed purpose	Trips	Share of total
Work	work	5845	24%
	education	858	4%
	business visit	231	1%
	business meeting	289	1%
	business service	270	1%
	business	339	1%
Errand	bringing children	2371	10%
	shopping	4786	20%
	bank	807	3%
	doctor	577	2%
Leisure	visiting family/friends	3374	14%
	amusement	1564	7%
	weekend cottage	123	1%
	travel/ excursion	439	2%
	sport	893	4%
	meeting	385	2%
	walk	778	3%
Total		23929	100%

Work

The aggregate work purpose consists of work, education and four business purposes, with work forming the largest part of this main purpose.

Errand

20% of all trips have shopping as purpose, 10% of all trips fall within the category of bringing children. Other errands, such as seeing the doctor or going to the bank only take up 5% of the trips, or one out of several errand trips.

Leisure

The category visiting friends and family takes up 20% of all trips, amusement trips take up 7% and trips to sport activities takes up 4% of all purposes.

2.2 GIS (Graphic Information System)

The calculation of car distances and times from GIS is carried out by DMU. The distance between two locations is measured as the sum of the direct line distance between the centroids that is placed on the roads[1] between these two locations. This way of calculating the distance blurs some of the turns in the road, with the result that DMU estimate that the calculation underestimates the real distance by car, by 4 percent[2].

The travel time is calculated as the distance for each road segment, divided by the travel speed, which is dependent upon the quality of the road.

Denmark has been divided into approximately 1557 zones, most of which have a surrounding zone. The calculation of the distance between any combination of these locations requires 1557*1557 = 2,424,249 combinations. In order to limit the number of combinations, and because not all parts of Denmark are connected by roads, the GIS database has been split up in three parts.

- Jylland (1117 centroids)
- Sjælland (434 centroids)
- Bornholm (22 centroids)

Since there are only few tours between the three parts of Denmark, it is not considered worthwhile to allow travel between the three regions of Denmark, and no distance is calculated between these parts.

Matrixes on car distance and travel time by car are collected in three tables in the TDM database. A single centroid on Sjælland (Kalundborg) has dropped out of the database and the nearest centroid has been substituted.

In the car distance data, the island of Samsø is connected with Sjælland. However, the majority of travellers visit Samsø from Jylland, but it is not immediately possible to calculate distances from Jylland to Samsø. Therefore tours to and from Samsø are excluded from the analysis.

There is no centroid in the rural zones. It is assumed, that the centroid of the rural zones is the same as the centroid of the corresponding city zone. Hence, the dis-

[1] The centroids are placed at every crossroad.

[2] E-mail from DMU of 26th of march 1997.

tance between a rural zone and the corresponding city zone is automatically set to zero in the first place. This has been corrected by means of average travel distance calculations according to the following formula:

$$D = 0.7 * SQRT(A/\pi)$$

where D is the calculated distance and A is the area of the rural zone plus the area of the corresponding city zone.

Within the three parts of Denmark, the distances between each zone and all other zones have been calculated in GIS. The calculations are based on a combination of roads and characteristics of other roads and zones.

The trip distance is also sought in the TU survey. The IP is asked how many kilometres were travelled by the 18 different modes - and the travel time for each of the 18 modes, for each trip.

The reported travel distances from TU are in general a little higher than the calculated distances from GIS. This is expected, partly because the GIS distance is by definition the shortest distance between the starting point to the destination, but also partly due to the systematic underestimation mentioned above. Table 2-8 illustrates the average reported distance by car and the average GIS distance by car[3].

Table 2-8 Reported travel distance by car and GIS distance by car.

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	TOTKM	19.2784	12035	23.0936	.2105
	GISKM	17.0776	12035	20.8872	.1904

The people in TU report approximately 2 kilometre (13%) longer distances than the actual (shortest) distance calculated from GIS. The difference is highly significant.

This over reporting could also be due to some people making a detour which they do not report in the questionnaire, or by simply not driving the shortest way. Such cases would look like relatively large over reporting from these trips.

Table 2-9 shows selected distances calculated by GIS and alternative methods.

[3] Only trips where car is the only transport measure is included in this analysis.

Table 2-9 GIS distance and actually measured distances.

Origin	Destination	actual distance*	GIS distance	Difference
Sjælland				
Helsingør	Rødby Havn	203	190	6%
Arnakke	Hillested	155**	145	6%
Arnakke	Holbæk	9**	8	11%
Arnakke	Lyngby	65**	57	12%
Jylland				
Tønder	Skagen	387	361	7%
Ringkøbing	Skagen	277	262	5%
Ringkøbing	Grenå	188	179	5%
Bornholm				
Rønne	Allinge	24	22	8%
Rønne	Neksø	29	24	17%
Gudhjem	Hasle	20	20	0%
Gudhjem	Svaneke	15	14	7%
Total		1372	1282	7%

* Actual distance from distance matrices in "Topografisk Atlas", Kort- og Martikelstyrelsen, 1989.

** Distances from "Arnakke" measured by actual car trip from Arnakke.

Table 2-9 shows a general under-estimation of distance calculated from GIS. The average under estimation in the selected tours is 7%. Since DMU estimates that the chosen method in GIS will result in 4% under estimation, all GIS based car distances are increased by 4% in the analysis.

2.2.1 Car speeds

Table 2-10 shows the average car speed calculated from the GIS tables. Note that the car speed is higher in sjælland and Jylland relative to Bornholm. This difference is due to the fact that there is no highways at Bornholm.

Table 2-10. Average car speed in main parts of Denmark according to GIS

landsdel	Average car speed Km/h
Sjælland	77
Bornholm	70
Jylland	78

Table 2-11 Shows the average car speed grouped by start county and destination county. The counties are grouped according to the three main areas of Denmark, Sjælland, Jylland and Bornholm.

Table 2-11. Average car speed grouped by start point county and destination county, Km/h

County	Kb	Fr	Ro	Vs	St	Fy	Nj	Ri	Rn	Sj	Vj	Vi	År	Bo
København	68	76	82	85	91									
Frederiksborg	76	66	74	70	86									
Roskilde	82	74	66	74	82									
Vestsjælland	85	70	74	66	74									
Storstrøm	91	86	82	74	74									
Fyn						65	92	80	78	75	85	81	90	
Nordjylland						92	70	77	74	91	88	74	80	
Ribe						80	77	66	71	71	70	72	76	
Ringkjøbing						78	74	71	67	76	70	70	69	
Sønderjylland						75	91	71	76	71	83	80	91	
Vejle						85	88	70	70	83	70	73	81	
Viborg						81	74	72	70	80	73	69	72	
Århus						90	80	76	69	91	81	72	69	
Bornholm														70

The average car speed is lowest, 66 to 74 km/h, in the diagonal, where the trips start and end in the same county. When the trips go to another county, the speed depends upon if there is a highway between these two counties. For instance the average car speed is 91 km/h on average for trips which start in Copenhagen and end somewhere in Storstrøms county. This seems to be a resonable speed since there is a higway connecting the two counties and since the speed limit here is 110 km/h.

Table 2-12 shows the average car speed grouped by distance.

Table 2-12. Average car speed grouped by distance, km/h

Distance	Speed, km/h
0 - 9 km	62
10-19 km	65
20-49 km	70
50-99 km	75
100 km or longer	82

Table 2-12 shows that the average car speed is higher the longer the distance is. For trips shorter than 10 km the average car speed is 62 km/h, while the average car speed is 82 km/h for trips longer than 100 km.

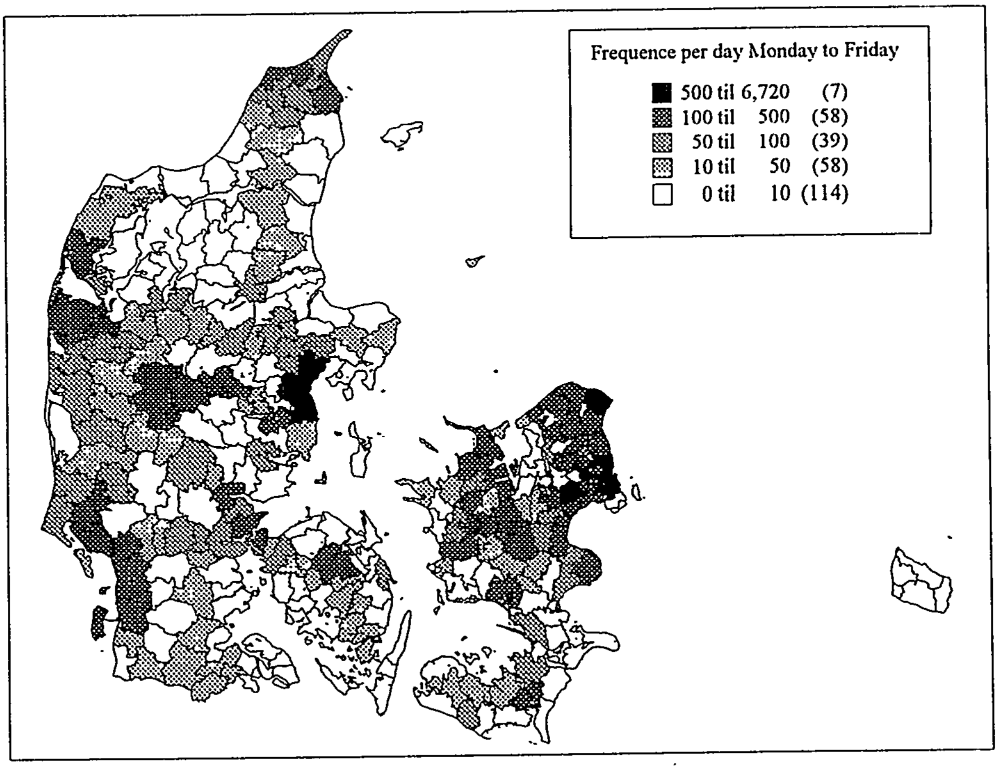
2.3 LoS Variables

Data relating to the level of service (LoS) of buses and trains in the different zones is delivered by The Danish Environmental Agency (DMU). Level of service is measured as the total number of departures on the specific weekday.

LoS, train

The count of train departures was made for workdays, Saturdays and non-working days, and the LoS in a zone is then given by the sum of departures from the relevant zone on the given day of travelling. The service level for trains is aggregated to municipality level. Figure 2-3 shows the service level of train at the municipality level.

Figure 2-3 Level of service, frequency of trains at municipality level



Distance to station

Time to walk to the nearest station (reported in TU questionnaire) is an indicator of the density of the network of train transport. In the TU-questionnaire, the distance is measured in minutes to walk to the nearest station.

Table 2-13 Minutes to walk to nearest station (GAASTA)

Walk time home-station	Number
Less than 5 minutes	862
5 - 10 minutes	2597
11 - 15 minutes	1418
16 - 30 minutes	2423
Over 30 minutes	6469
Not informed	4
Unknown	20

This averages[4] to the following figures for the counties in Denmark.

Table 2-14 Minutes to walk to the nearest station in different counties.

County	Minutes
København	17
Frederiksborg	22
Roskilde	24
Vestsjælland	27
Storstrøm	32
Bornholm	No trains
Fyn	34
Sønderjylland	37
Ribe	29
Vejle	33
Ringkjøbing	27
Århus	30
Viborg	35
Nordjylland	35

LoS, bus

The number of departures by bus from each zone has been calculated by combining bus schedules, and translating tables from bus stop location to TU zone.

The number of departures were only available in the material from DMU, for the six counties listed below:

- Vestsjælland
- Bornholm
- Ribe
- Ringkjøbing
- Viborg
- Nordjylland

The remaining of the 14 counties and Copenhagen:

- København
- Frederiksborg
- Roskilde
- Storstrøm,
- Fyn,
- Sønderjylland,
- Vejle and
- Århus

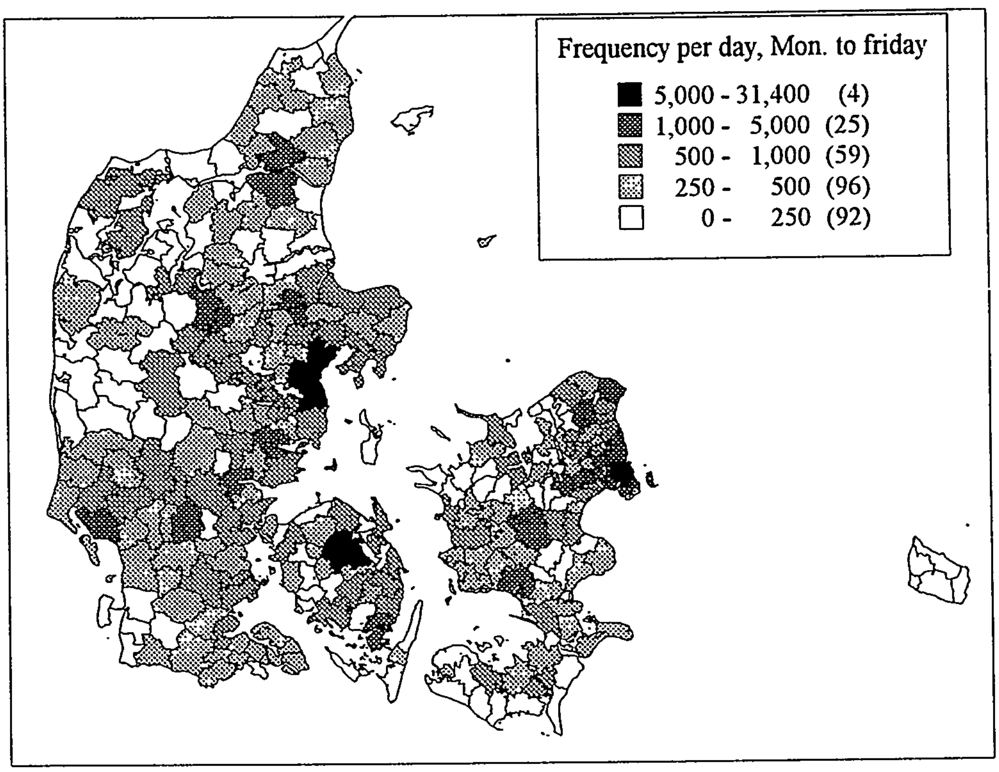
[4] Assuming 2.5, 7.5, 13, 23 and 45 minutes as representing the respective category.

relied on a survey undertaken by the Department of Traffic Planning, COWI for The Ministry of Transport.

A number of departures are specified for school days, non-school days, Saturdays and non-working days. The number of departures on school days is used to describe the LoS on working days.

LoS is linked to the database through the unique value of the centre-id of the zone. Rural zones have no centre-id, and the LoS is assigned the values of the city to which it belongs. This implies that rural zones are ascribed the same bus LoS as the city zone to which it belongs. This seems reasonable, since most of the buses travel through the surrounding rural zone to get into the city.

Figure 2-4 The Average Level of service, frequency of bus departures at municipality level



Distance to bus

A further measure of the level of service is the reported time to walk to nearest bus in the TU data. This measure is coded like the corresponding measure for train stations.

2.4 Attraction Variables

The number of employees in different professions, together with the number of inhabitants, have been used as attraction variables. DMU provided the number of employees in different branches linked to TU zones. On the basis of this data material, each TU zone is assigned the following attraction variables.

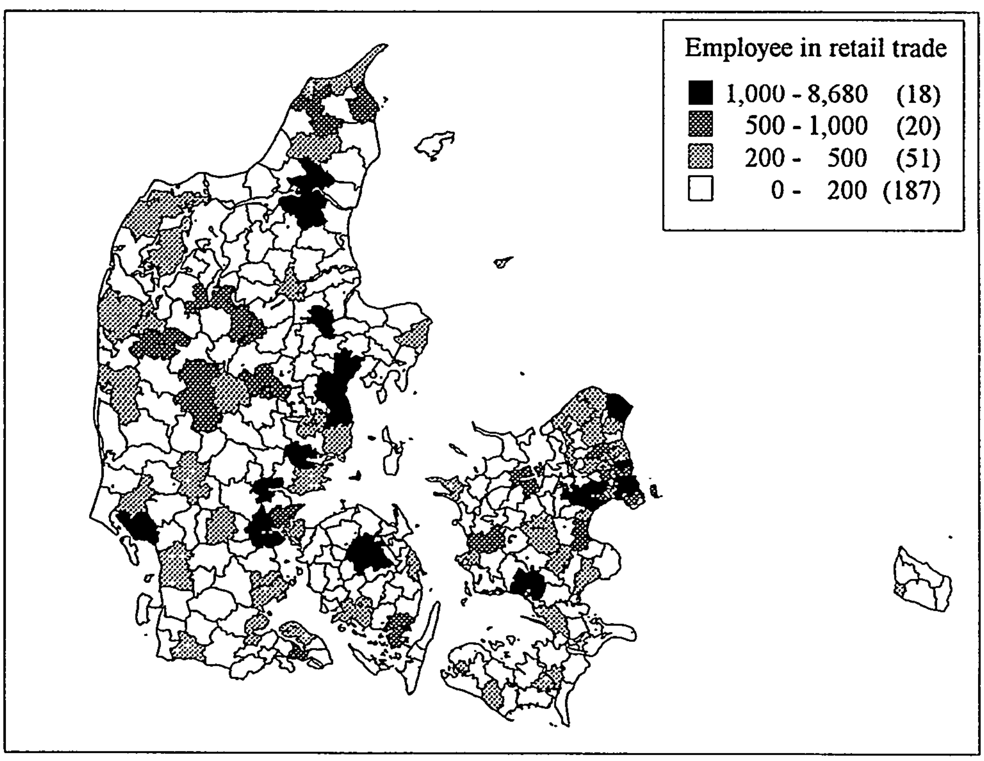
- Number of **employees** in 12 professions

  - agriculture
  - manufacturing
  - construction
  - private service
  - wholesale trade
  - retail trade
  - restaurants and hotels
  - transport and postal services
  - financing, insurance, etc.
  - public administration
  - primary and lower-secondary school
  - hospitals
  - grammar school
  - higher education

- Number of **inhabitants** in 5 years age groups and sex groups

The original data measure the number of persons in zones consisting of cities with more than 500 inhabitants. The rest - small towns and rural areas - is reported as a residual, but distributed at the 275 municipalities in Denmark. Figure 2-5 shows the distribution in Denmark.

Figure 2-5 Number of employees in retail trade

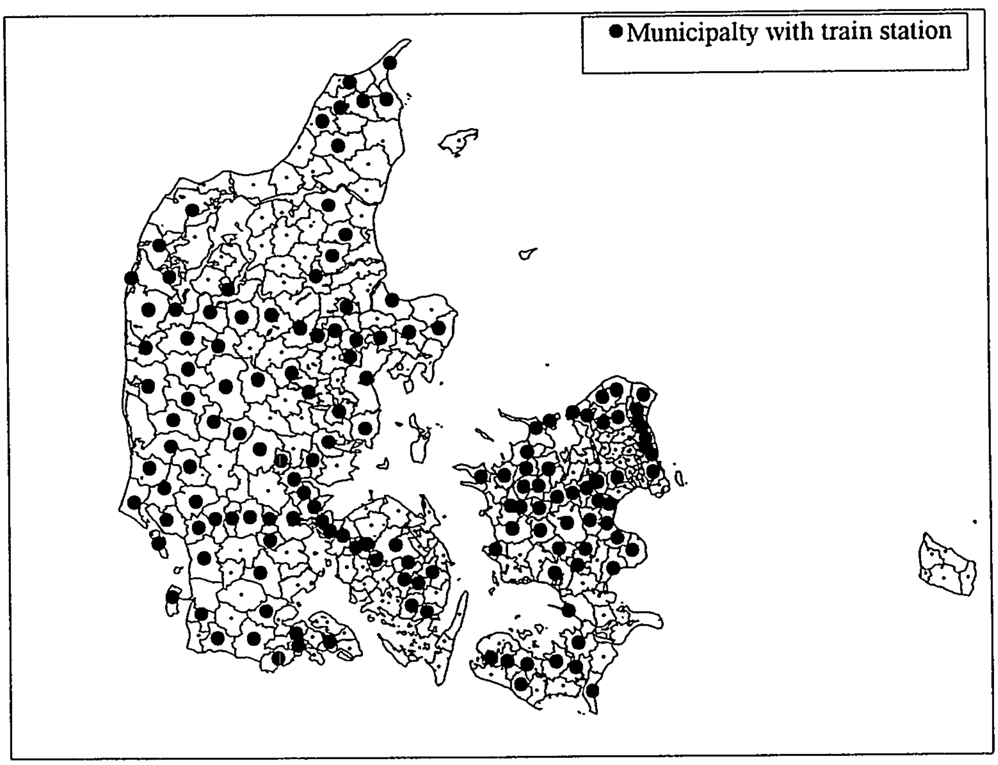


2.5 Public Transport

Travelling by train is assumed to imply a bus journey to the nearest train station, train travel from the station to that train station nearest to the destination, and then bus travel from this train station to the destination. The distance by train is assumed to be equal to the car distance between the two train stations. The bus distance to and from the train station is assumed to be equal to the car distance from home to the train station and from the train station to the destination. All train stations in Denmark have been marked by the TU zone in which they are located. From this data set, the nearest station to all TU zones in Denmark has been found.

Figure 2-6 shows the train stations in Denmark. Here the train stations are marked by their municipality for graphical convenience. The TU zones are used in the actual calculations.

Figure 2-6 Train stations in Denmark



2.5.1 Travel cost

A minor sample from the TU survey, consisting of 120 journeys from 10 randomly chosen destinations in each of the 12 counties, was conducted in order to specify functions for travel distance by means of public transport. Travel distances were measured by car (the GIS matrices), by train and by bus on the basis of detailed maps. For public transport, the route passes through a number of tariff zones. The cost of the tour is calculated on the basis of the number of zones that the tour passes through and the fare structure in the specific county.

The 120 trips represent 10,453 km, and travel through 1,850 tariff zones. This results in an average of 5.65 km per zone. However, the size of zones vary greatly among the counties. Table 2-15 shows the average number of kilometres that a tour passes through.

Table 2-15 Average travelled km[5] per tariff zones

County	Company	Average number of zones per journey	Average kilometre per journey	Average km. per zone
København	HT	3.90	18.71	4.28
Frederiksborg	HT	3.90	18.71	4.28
Roskilde	HT	3.90	18.71	4.28
Vestsjælland	VT	3.47	20.38	5.24
Storstrøm	STS	21.36	49.23	2.31
Fyn	FynB	9.35	37.48	3.85
Sønderjylland	SydB	4.21	34.21	7.42
Ribe	RIBUS	3.81	35.52	9.44
Vejle	VAT	5.03	38.41	6.63
Ringkjøbing	RINGBUS	4.45	35.05	7.07
Århus	ÅBUS	7.43	41.93	5.23
Viborg	VAFT	6.58	50.37	7.67
Nordjylland	NT	8.11	50.61	6.99

Since the price system and the size of the zones varies between these counties, it is hoped that calculating the costs of public transport will improve the variation in the explanatory variables.

The calculation of travel cost is based on the actual price structure in 1995. The price structure varies a lot between the counties of Denmark. Typically, there are three different prices for a bus trip.

- Single ticket
- Ten trip tickets
- Monthly Passes

[5] Kilometre is measured as car kilometre from the GIS database.

Table 2-16 Price structure for bus travels cost of travelling in three tariff zones

Company	Single ticket	Multiple Journey Ticket	Monthly Season Ticket
HT	15.00	10.00	7.25
VT	30.00	·17.00	10.00
STS	18.00	14.00	12.75
BAT	24.00	18.00	12.00
FynB	12.00	9.00	7.00
SydB	30.00	19.50	11.75
RIBUS	13.50	9.00	8.50
VAT	19.00	9.75	8.25
RINGBUS	24.00	15.75	12.50
ÅBUS	18.00	13.00	8.00
VAFT	16.00	10.50	8.50
NT	16.50	11.00	8.00

Table 2-16 shows travel fare for travels within three tariff zones. The data base, TakstTabel, consists of information on any number of zones.

2.5.2 Travel time

Travel time is calculated from average travel speed. The travel speed for regional trains is 57 km/h on average, based on a sample of 359 train services in Denmark. This estimate of travel time includes in train time and waiting time, but does not include any time for delays.

Travel speed by bus varies in different parts of Denmark. Table 2-17 shows average travel speed in counties of Denmark, and these figures have been used to calculate travel time by bus.

Table 2-17 Travel speed by bus

County	Average travel speed km/h
København	23.7
Frederiksborg	23.7
Roskilde	23.7
Vestsjælland	36.6
Storstrøm	35.9
Bornholm	38.7
Fyn	33.3
Sønderjylland	37.8
Ribe	33.8
Vejle	38.5
Ringkjøbing	39.5
Århus	30.8
Viborg	36.6
Nordjylland	35.1

In general, the variation in travel speed is due to different average travel speed in different counties. No systematic variation with regard to tour length is included.

2.5.3 Bus to Train

The location of all train stations has been registered, and from the data, the distance to the nearest[6] train station from any zone in Denmark has been calculated. No public network data has been included in these calculations, i.e. there can be situations, where it is more reasonable to go to a train station a little farther away, and there can be situations where the train line from the nearest train station conveys you in the wrong direction. Further, not all trains stop at all stations. These problems are ignored.

[6] As the distance is measured from the GIS, distance is measured as car distance.

3 Data Processing

3.1 Destination Sampling

3.1.1 Sampling Strategy

The mode destination choice sub-models predict the choice between a number of destination zones. Denmark is split up into about 3000 zones, and the inclusion of all of these zones within the models, as alternative destinations for each IP, is clearly impossible. Therefore, a sample of zones needs to be drawn for each IP.

This sampling of zones is based on stratified sampling, with the strategy dependent on the type of tour:

Simple tours

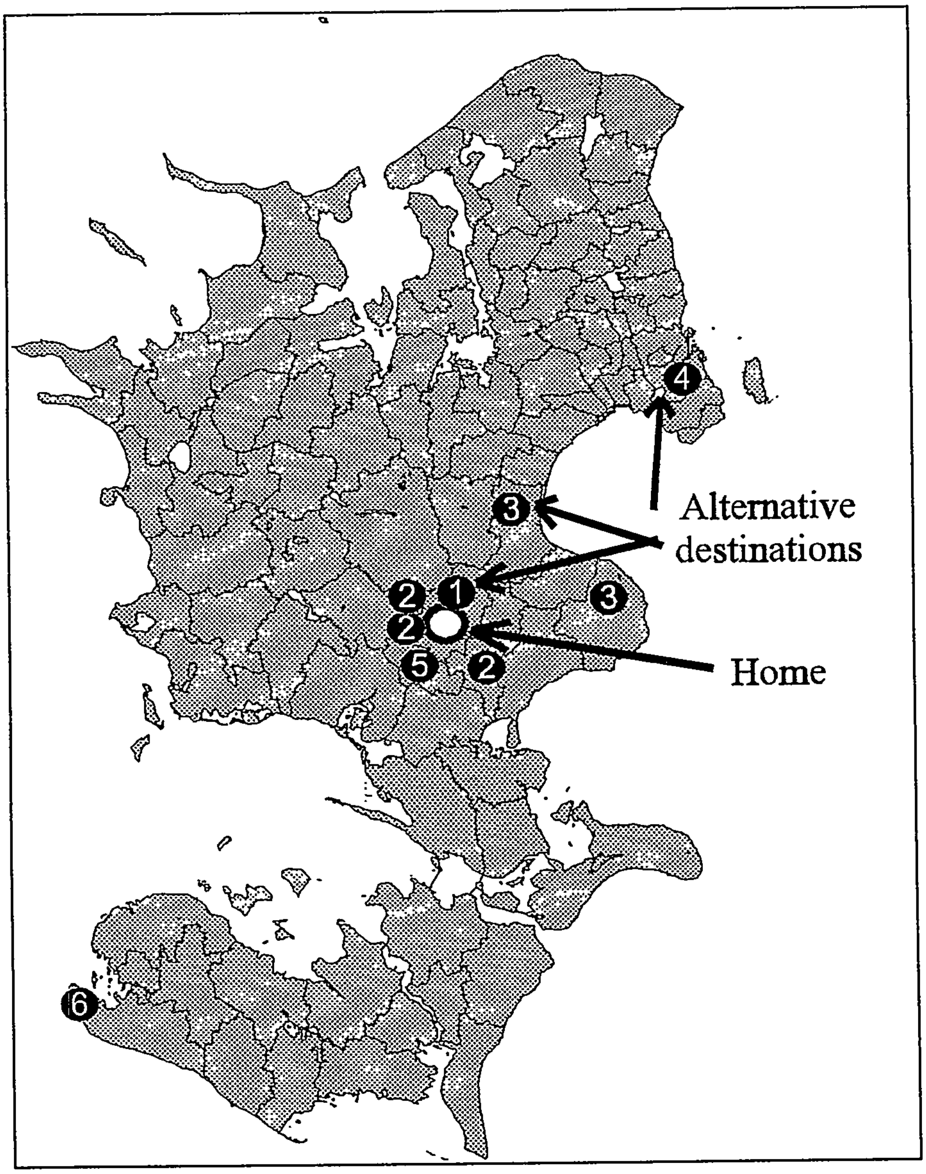
Six strata are defined for each individual. From which, 9 alternative destinations are sampled. The number of zones drawn from each strata is provided in brackets.

1. The TU zone where IP lives (1 zone)
2. TU city zones in the municipality where IP lives (3 zones)
3. TU city zones in the commuting region where IP lives (2 zones)
4. TU city zones in the residual main area of Denmark where IP lives (1 zone)
5. TU rural zones in the municipality where IP lives (1 zone)
6. TU rural zones in the residual main area of Denmark (1 zone)

The number of zones drawn from each sample is chosen in accordance with the observed frequency of choice between city- and rural zones and observed travel distance.

Figure 3-1 illustrate the sampling procedure for an IP living in Haslev on Sjælland.

Figure 3-1 Example: Stratum and alternatives one IP



The circle filled with white indicates the location where the IP lives. The filled circles indicate the sampled potential destinations of the IP. The numbers in the circles indicate which strata the possible destinations are drawn from, and the figure clearly illustrates that the alternative destinations are concentrated in the area where IP lives.

Table 3-1 provides the names of both the alternative destinations, and their respective strata. The location where IP lives (Haslev) is included in the alternative destinations. In this case, the IP actually chose Haslev as the destination for the trip.

Table 3-1 Alternative destinations in example

Strata	Alternative destinations
1	Haslev city (chosen destination)
2	Teestrup city
2	Førslev city
2	Rønnede city
3	Lellinge city
3	Klippinge city
4	Islands Brygge, Cph. city
5	Haslev (rural area)
6	Langø (rural area)

Triangle tours

The first nine destinations are drawn in the same way as for single tours: For each of these destinations, four strata are defined for each individual.

1. Living zone and work zone
2. TU city zones in the municipality where IP lives or works
3. TU city zones not in the municipality where IP lives or works
4. TU land zones different from living or working zone of IP

The access module used to sample alternative destinations is shown in Annex 1.

From each of these four strata, one alternative destination is drawn yielding 36 different combinations of first and secondary destinations.

Replace with chosen destination

For the estimation, the actual chosen destination replaces one of the alternative destinations in the relevant strata.

3.1.2 Correction for sampling

A correction term is included in the mode and destination choice models to account for the different sampling probabilities. The sampling probability is calculated as the number of sampled zones in the relevant stratum (S), divided by the number of zones in the sample chosen from this stratum:

$P_s(n)$ = (Number of sampled zones from stratum S) / (Number of zones in S)

The correction term is calculated as:

Correction = - Log($P_s(n)$)

3.2 Travel Cost

Travel costs for the chosen destination and for alternative destinations have been calculated for the following modes

- Walk

- Bicycle
- Car
- Car Passenger (CP)
- Bus, and
- Train

The distance is calculated from the GIS matrices, where intra-zone trips are seen as starting and ending at the same point. With no correction , this would imply a distance of zero.

In the TDM analysis intra-zone travel distance, D, is calculated from the area, A, of the zone by the formula:

$D = 0.7 * SQRT(A/\pi)$

This formula calculates the mathematically expected travel distance as a function of the area, assuming that trip ends are randomly distributed over the zone.

3.2.1 Walk and bicycle

Walking and cycling distance is taken from the GIS based car network. The walking speed is set to 5 km/h and the cycling speed is set to 15 km/h, the observed average speed from the TU survey. There are no other costs for these modes.

3.2.2 Car

Distance equal to car distance from GIS matrices, but multiplied with a factor 1.04, since the method employed to calculate car distance was found to systematically underestimate the real distance, (see chapter 2). Time by car is calculated from GIS by means of distance and the speed limit at the specific road.

Marginal costs of car driving are taken to only include petrol costs. Therefore marginal costs are calculated as the distance multiplied by the average fuel consumption and by the average petrol price, which was 5.92 DKK/l in 1995. Parking fees are not included in car costs.

Table 3-2 Petrol prices in 1995 (according to Dansk Benzinhandlerforening)

Date	Petrol prices DKK/l	days
January 1, 1995	5.80	45
February 15, 1995	5.89	27
March 14, 1995	5.76	18
April 1, 1995	5.76	5
April 6, 1995	5.95	13
April 19, 1995	6.03	30
May 19, 1995	6.12	19
June 7, 1995	6.06	14
June 21, 1995	5.95	14
July 5, 1995	5.84	20
July 25, 1995	5.77	10
August 4, 1995	5.85	13
August 17, 1995	5.93	21
September 7, 1995	5.98	24
October 1, 1995	6.04	3
October 4, 1995	5.89	30
November 3, 1995	6.01	32
December 5, 1995	5.91	14
December 19, 1995	5.86	13
Weighted average	5.92	365

3.2.3 Car Passenger (CP)

The same times and distances are assumed for CP as for cars. The costs are assumed to be half the petrol costs, i.e. 2.96 DKK/l.

3.2.4 Bus

Time for the bus travel is calculated from the average travel speed by bus reported in chapter 2.

Bus distance has not been directly available, instead bus distance is calculated as a function of the car distance. This function is based on the analysis of 120 combinations of origin and destination zones in Denmark drawn at random. For each of these 120 trips, the travel route by bus and train, actual distance travelled, number of zones crossed and county, have been recorded.

Based on these data, public transport distances by bus are found by a regression on distance by car (GIS). The results from this regression are provided in Table 3-3. Tests have been performed to see if transformation of the variables improves the model, but this was not found to be the case. In zones with high level of bus service, it is likely that it is easier to find a bus that goes more directly to the wanted destination. This leads to the hypothesis that zones with high level of service have a lower constant term and parameter estimate of car kilometre closer to 1. It has not been possible to support such hypothesis with the present data material.

Table 3-3 Regression of bus km with car km as independent variable

Coefficients[a]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	3.610	2.850		1.267	.208
	DISTBIL	1.128	.035	.943	32.022	.000

a. Dependent Variable: SUMOFKM

On the basis of this regression, bus kilometre is calculated as 3.6+1.13*carkm. The constant term is not significant at 5% level. However, is seems reasonable to assume some positive constant term since the bus never stops right at your front door or at the desired destination[7]. Therefore, we have chosen to include the constant term in the formula for the bus km.

The cost of public transport has been calculated from both the public transport distances, and knowledge about the cost structure of public transport in the counties of Denmark. Bus distances are transformed to tariff zones according to the average zone size in the counties of Denmark. Bus zones are then transformed to bus and train cost according to the tariff systems in the counties.

3.2.5 Train

A trip by train most often involves a trip by bus to the station, then a trip by train to the station nearest to the destination and finally a trip by bus from the train station to the destination.

For each trip, the train stations nearest to the living zone and the destination zone are identified, where the former has been taken as the one that requires the shortest car distance, measured from the GIS database.

The bus distance to the station is assumed to be equal to the car distance. Here the function from the bus km (section 3.2.4) is not used, as bus routes to and from train stations presumably are more direct than routes travelling from one random zone to another.

[7] Because of the definition of main mode for the whole trip all transport including walk or bicycle to the bus stop is calculated as buskm.

The train distance is measured as the car distance between the two stations i.e. the GIS figure, multiplied with the factor 1.04 to correct for the systematic underestimation of distance (se chapter 2).

Travel time by bus is calculated according to the average travel speed by bus in the relevant county, with the addition of 5 minutes per shift. Bus from home to station includes two shifts.

However, when the distance to the station is shorter than 500 metres, it is assumed that the IP walks, i.e. no costs, 5 km/h and only one supplement of 5 minutes is added to the journey.

Bus and train distances are transformed to tariff zones, according to the average zone size in the counties in Denmark.

Bus and train zones are transformed to bus and train cost according to the tariff systems in the counties.

Train time is calculated as total distance travelled by train divided by the average speed of 57 km/h.

Figure 3-2 shows an example of starting point and destination to illustrate distance and time calculations.

Figure 3-2. Example: Travel from Arnakke to COWI, Lyngby

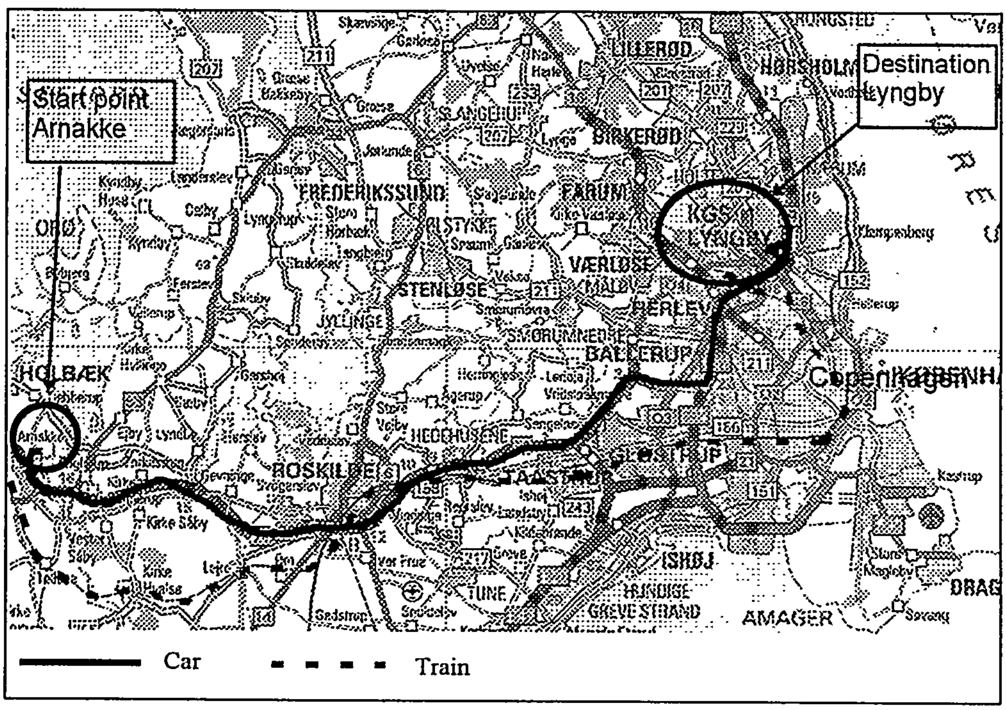


Table 3-4 shows distance and time for the example above.

Table 3-4 Example: Travel from Arnakke to COWI, Lyngby.

	Distance	Time (minutes)
Car		
Actual travel	65	40 - 45
GIS calculated	57	37
Train		
Actual travel	65	90
Calculated		
Train	57	62
bus to train	1.7	1.7 km / 36.6 km/h * 60 min = 2.8
bus from train to COWI	0.56	0.56 km / 23.7 km/h * 60 min = 1.4
Shifts		4*5=20
Total		86.2

The actual travel takes one hour and 30 minutes by train. However, the actual trip consists of 15 minutes by bicycle from home at Arnakke to the nearest train station and 6 minutes of walking from the train station at Lyngby to COWI.

The TDM measure includes bus travel from home (Arnakke) to the train station of 1.7 kilometre - there is actually 3 kilometres and the bus frequency is low. This bus travel would take at least 20 minutes.

Adding 5 minutes to the travel time for every shift (in this case 4 shifts) brings the calculated travel time up to 86 minutes - very close to the actual travel time.

3.3 Definition of Variables

The following section describes the variables used in the model estimation. The number (No) is a unique number referring to the order in the data file used in the analysis.

3.3.1 Family Variables

Table 3-5 lists the variables concerning the family. Variables refer to two definitions of family; the broad definition including adult children and grand parents and the narrow definition only including parents and children less than 18 years old.

Table 3-5 Family variables

No	Name	Description
3	famtyp	Family type: 1=single, 2=couple, 3=auxiliary, narrow definition
4	persons	Number of persons in narrow family
5	childs1	Number of children age 0 to 4 in narrow family
6	childs2	Number of children age 5 to 9 in narrow family
7	childs3	Number of children age 10 to 14 in narrow family
8	childs4	Number of children age 15 to 17 in narrow family
9	HAFTBIL	Number of cars in broad family, 1 = one car, 2 = two cars, 3 = three or more cars, 4 = no car
11	kk	Number of drivers' licences in family (broad)
18	Partner	Dummy, 1 if IP has a partner, narrow
19	ALDERP	Age of partner
20	STILP	Working status of partner (categories)
21	ARBTIDP	Weekly hours of work for partner
22	KOREP	Partners drivers licence status, 1 = licence, 0 and 2 = no licence

The different categories are listed in the appendix.

3.3.2 IP variables

This section details those variables that concern the IP.

Table 3-6 IP variables

No	Name	description
1	ID	unique ID number
2	ALDER	Age in years
10	KOREKORT	Drivers licence status, 1 = licence, 0 and 2 = no licence
14	KON	Sex, 1 = man, 2 = woman
15	SKOLUD	General education, 1 = 1-6 years, 2 = 7-10 years, 3 = high school, 4 = other, 5 = child at school
16	STIL	Working status (categories)
17	ARBTID	Weekly hours of work
37	TarxDayWgti	Weight to representativity
38	inknti	Net income (DKK per year) of IP

Income statistics

Income statistics are available in 25,000 DKK intervals. In this analysis it was decided to use net income for IP per year.

Income questions traditionally yield low response rates. In the present study approximately one third could not, or would not, answer the income questions. The response rates were, not surprisingly, lowest on family income, reflecting the lack of knowledge of many members of the family as to total family income.

Since partial drop out on income questions is not equally distributed over socio-economic and demographic groups, the problem cannot be solved simply by excluding observations with missing income information. This would lead to a more skewed sample.

The issue of partial drop-out on income questions is generally handled by assuming that the individual with missing income has the same income as the average in their respective socio-economic group. However, this solution is still somewhat problematic, since it is known that people who do not answer income questions diverge from those who are able/willing to reveal their income. However, it would be even more erroneous to assume that their income is the same as the total average, but deleting these individuals from the sample would also be unacceptable, as it would result in large loss of information, and would provide a biased picture of the total population.

3.3.3 Trip diary Variables

A number of variables come from the TU interview.

Table 3-7 Trip diary variables

No	Name	description
39	chain	Chain type code
12	GAABUS	Minutes to walk to bus (categories)
13	GAASTA	Minutes to walk to train station (categories)
26	Mode	Main mode, 1 = Slow, 2 = Car, 3 = Bus, 4 = Subway, 5 = Train
40	DAG	Weekday of travel

Mode has been recorded to 5 modes from a more detailed mode variable as follows.

1. Walk
2. Bike, moped
3. MC, car(driver), taxi, truck driver, delivery van (driver)
4. Car passenger, truck passenger, delivery van passenger
5. Bus and subway
6. Train

Aeroplane and ferry are excluded from the analysis. Tours with aeroplane and ferry as main mode are deleted from the estimation database.

3.3.4 GTC and LoS Variables

Table 3-8 GTC and level of service variables

No	Name	description
31	BusHvdg	Number of bus departures on weekdays from home zone
32	BusLør	Number of bus departures on Saturdays from home zone
33	BusSøn	Number of bus departures on Sundays from home zone
34	Htoglevel	Number of train departures from home municipalities
35	Hurban	Urbanisation level of home zone (categories)
41	strata	Strata for the chosen destination
42	Aurban	Urbanisation level of destination zone (categories)
43	chosen	
44	dist	Distance by car in km. (GIS)
45	tid	Travel time by car in minutes (GIS)
46	Buskm1	Distance by bus
47	Togkm1	Distance by train
48	BusTilTogkm	Distance, bus to train, home zone
49	BusFraTogkm	Distance, bus to train, destination zone
50	Buspris1	Bus travel cost (DKK)
51	Togpris1	Train travel cost (DKK)
52	BusTPris1	Bus to train travel cost (DKK)
53	BusLevel	Number of bus departures from destination TU zone
54	TogLevel	Number of train departures from destination municipality

Variables 31, 32 and 33 contain LoS for bus on different week days at the home TU zone of the respondent. LoS of the week day the travel which actually took place is used in the estimations. Variable 34 contains the LoS of train for the same weekday, but LoS for train is calculated at municipality level.

Urban contains the urbanisation level of the destination TU zone. Chosen tells which of the alternative TU zones the respondent has chosen to go to.

Three GIS matrices covering Jylland, Sjælland and Bornholm give the distance between the origin and destination (TU) zones. Rural zones do not exist in the GIS matrices. It is assumed that the distance from one rural zone to another is equal to the distance between the corresponding city zones.

3.3.5 Attraction Variables

Available attraction variables are the number of inhabitants in each TU zone and the number of employees in 12 branches in each TU zone.

Table 3-9 Attraction variables

No	Name	description
28	antal0t17	Population age 0 to 17
29	antal18t64	Population age 18 to 64
30	antalo64	Population age 65 and above
55	befolkning	Total population
56	Landbrug	Number of employed in agriculture
57	Fremstilling	Number of employed in manufacturing
58	ByggeOgAnl	Number of employed in construction
59	Handel1	Number of employed in private service
60	Handel2	Number of employed in wholesale trade
61	Handel3	Number of employed in retail trade
62	HotelOgRest	Number of employed in restaurants and hotels
63	TrnspPostOgTele	Number of employed in transport and postal services
64	PengeOgFin	Number of employed in financing, insurance, etc.
65	OffAdm	Number of employed in public administration
66	Folkeskoler	Number of employed in primary and lower-secondary school
67	Hospitaler	Number of employed in hospitals
68	Gymnasier	Number of employed in grammar school
69	Uddannelse	Number of employed in higher education

Register data at this level is not available for the TU - rural zones and for the small cities. In these cases, only a residual sum at municipality level is available. This residual has been distributed on the rural zones and small city zones. 90% of agriculture and 10% of the other branches are placed in rural zones. The opposite distribution with 10% of agriculture and 90% of the other branches is placed in the small city zones.

3.4 Variables for Triangular Tours

For triangular tours, some additional variables are used to describe the secondary destination.

The attraction variables follow the definitions for the primary tour. But the variable number has been increased by 30.

The distance, cost and time variables are calculated as the marginal increase in the relevant variable, when comparing the round trip in the triangle with the simple tour going to the main destination and back.

Thus, the distance is measured as the total tour distance (inclusive the detour), minus a tour from home to work and home again. The same assumption is made for time and cost variables.

3.4.1 Chain choice Model

The chain choice model employs the same variables describing the IP and the household, as the mode/destination choice. Variables 41 through 50 are the log-sums calculated from the mode/destination choice sub-models.

Table 3-10. Estimation data file

Variable no	Name	Variable no	Name	Variable no	Name
1	ID	18	Partner	35	Hurban
2	ALDER	19	ALDERP	36	t
3	famtyp	20	STILP	37	TarxDayWgti
4	persons	21	ARBTIDP	38	inknti
5	childs1	22	KOREP	39	chain
6	childs2	23	Forml2	40	DAG
7	childs3	24	Forml3	41	LogsCH12
8	childs4	25	Forml4	42	LogsC12x
9	HAFTBIL	26	Mode	43	LogsCH13
10	KOREKORT	27	H38zon	44	LogsCH14
11	kk	28	antal0t17	45	LogsCX13
12	GAABUS	29	antal18t64	46	LogsC214
13	GAASTA	30	antalo64	47	LogsC314
14	KON	31	BusHvdg	48	LogsC123
15	SKOLUD	32	BusLør	49	LogsC23x
16	STIL	33	BusSøn	50	LogsC24x
17	ARBTID	34	Htoglevel	51	Strata

Descriptive statistics are given in annex 2, table 1.

4 Simplifying Tours

The basic models in PETRA are the mode destination choice models. The estimation of which, are based upon the trip diaries from the TU survey, and the interviewee's actual travel pattern and mode choice.

Since the trip diary describes the real world it is often very complex, too complex for the models to handle. Therefore, it is necessary to simplify the travel pattern and mode choice information, prior to their use in the model. The following chapter describes the simplifications used to make data tractable for the estimation of the mode and destination choice models.

4.1 Tours and Trips

We are concerned with a person's travel over a day, characterised by the sequence of destinations reached during the day and the activities engaged in at these destinations.

Examples of observed sequences could be

- H W S H,
- H W H L H,
- H W S H S L H,

where H, W, S and L denote home, work, errand and leisure destinations, respectively. Such an observed sequence is hereafter called a *chain*.

The observed chains vary from the very simple (H W H, say) to the extremely complex with up to 18 destinations during a day. This complexity necessitates a reduced formal description of the chains to make it possible to build a model, requiring a few definitions.

A *destination* is either a location where a person engages in some activity that was the rationale of travelling, or is the home of the person. The activity is called a *travel purpose* or *purpose* for short. A *trip* is defined as the travel from one destination to the next; *intermediate destinations* where no activity is engaged in other than a change of travel mode are ignored. A *tour* is the sequence of trips that take a person from home to one or more destinations until he returns home.

Thus a chain consists of one or more tours, that comprise two or more trips between home and destinations. Each trip can make use of one, two or more models.

On short trips (less than 300 meter), information on mode is not available. For the short trips it is assumed that the IP walks.

4.2 Restrictions on Chains

4.2.1 Number of Purposes

The model is restricted to three possible travel purposes: Work, Errand and Leisure. Home is a destination, thus it is not considered as a travel purpose.

Each purpose may occur more than once in a chain. If we only consider whether each purpose occurs in a chain, and not how many times it occurs, then eight possible combinations arise:

W, only work purpose	0, no activities
WS, work and errand only	S, errand only
WL, work and leisure only	L, leisure only
WLS, work, errand and leisure	LS, both errand and leisure

The observed frequencies for each combination occurring in chains are given below.

Table 4-1. Observed frequencies of purpose combinations

Purpose combination	Percentage in sample
W	17%
WS	11%
WL	9%
WSL	6%
O	19%
S	13%
L	14%
SL	11%

All combinations of purposes are represented in enough cases to allow a model that distinguishes between them. Table 4-2 tabulates the frequencies of multiple occurrences of the same purpose in a chain.

Table 4-2. Occurrences of purposes

Purpose	0 occurrences	1 occurrence	2 occurrences	3+ occurrences
Work	57%	36%	4%	3%
Errand	59%	28%	8%	5%
Leisure	60%	29%	7%	3%

Work occurs in the observed chains at most once in 93% of all cases. Similarly, errand and leisure occur at most twice in 95% and 97% of all cases. This justifies a model restriction that each person travels to work at most once a day, and that errand and leisure purposes occur at most twice.

4.2.2 Number of Tours

A chain may contain any number of tours, and the maximum number of tours observed in the sample is 7. The table below shows the frequencies of chains with a different number of tours.

Table 4-3. Number of tours in chains

Number of tours	Frequency
0	23%
1	43%
2	26%
3	6%
3+	2%

The imposition of a restriction of a maximum of two tours, allows the use of all the tours undertaken for 92% of the individuals observed in the TU data.

Of all chains, 23% contain no tours. Of these 23%, about 15% actually contain very short tours (below 300 m).

4.2.3 Number of Purposes on a Tour

Table 4-4 illustrates that 91% of all tours include at most two destinations away from home. In the model, the restriction is adopted that tours can have only two such destinations.

Table 4-4. Number of purposes on tours

Number of purposes	1	2	3	4+
Frequency	77%	14%	5%	4%

4.2.4 Definition of Travel Patterns

The above provides justification for adopting the following restrictions for the model.

- A chain contains at most 2 tours
- Work occurs at most once in a chain
- Errand and leisure each occur at most twice in a chain
- A tour has one or two destinations away from home

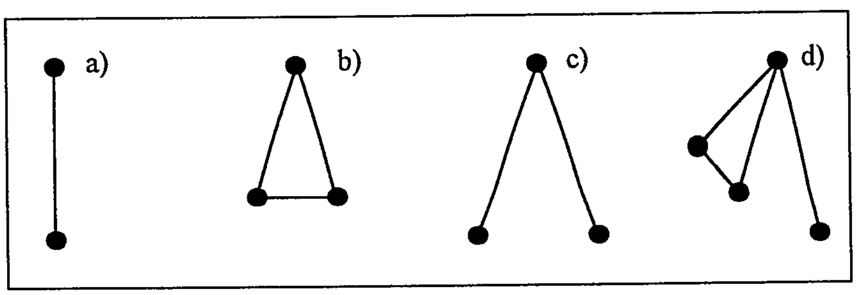
Furthermore, the following restrictions are adopted.

- A purpose can occur only once on a tour
- A chain contains at most three purposes

Tours are rearranged freely within chains, and destinations are likewise rearranged freely within tours.

With the adopted conventions and rearranging as required, four different travel patterns are possible, shown in Figure 4-1 below.

Figure 4-1. Travel patterns



For each tour, six different purpose combinations are possible, namely

- W, S, L, WS, WL, SL

For the chains the following 18 combinations are possible in addition to the stay home alternative.

a) **W, S, L**

b) **WS, WL, SL**

c) **W-S, W-L,** S-S, S-L, L-L

d) **WS-S, WS-L, WL-S, WL-L,** SL-S, SL-L, W-SL

Chains in bold are the ones represented in the estimations. The others are not excluded, just simplified i.e. one of the purposes are deleted.

4.3 Transformation of Chains

A number of the observed chains do not conform to the conventions adopted in the previous sections, and thus, require further transformation. The transformations imply that some precision is lost in representing the chains: Some intermediate destinations will be disregarded and trips will be added or subtracted in order to make observations conform to the conventions. The loss of precision is necessary for the ability to work with the observed travel behaviour from the point of view of choice between chains in an integrated way, rather than looking at tours or trips separately.

4.3.1 Transformation Rules

Incomplete diaries

By definition, a tour starts and ends at home. In the TU data, this is not always the case. The travel diary covers the period from 3 o'clock in the morning to 3 o'clock the following morning, so with some respondents the diary starts in the middle of a tour as it can also end before return to home on the last tour. In order to make these observations conform to the general format for chains, extra trips are added such that the chains always start and end at home. Thus, all chains consist of complete tours.

With only complete tours in the data, the next tours are transformed. We observe that the sequence of destinations on a tour is of no interest and hence the sequence can be rearranged for convenience.

Work - business

There is a distinction between work and business destinations in the data. This distinction is ignored, so business destinations are treated as work destinations.

Doublets

If a travel purpose occurs more than once on a tour, then all but one occurrences are deleted. The remaining destination is chosen to be the one farthest away from home.

More work tours

We have adopted the restriction that a work destination may occur at most once. If a work destination occurs on more than one tour in the observed travel diary, then all, but one work destination, are deleted. The tour is retained where the distance between the work destination and home is the highest. All other tours including work destinations are deleted from the chains

At most two tours

When errand and leisure happens twice in the chain, the purpose to the remotest destination is the selected one.

The abbreviation of the tours requires that the travel diaries made up by the tours (purposes) are also abbreviated, such that the travel diaries conform to the restriction that they contain at most two tours.

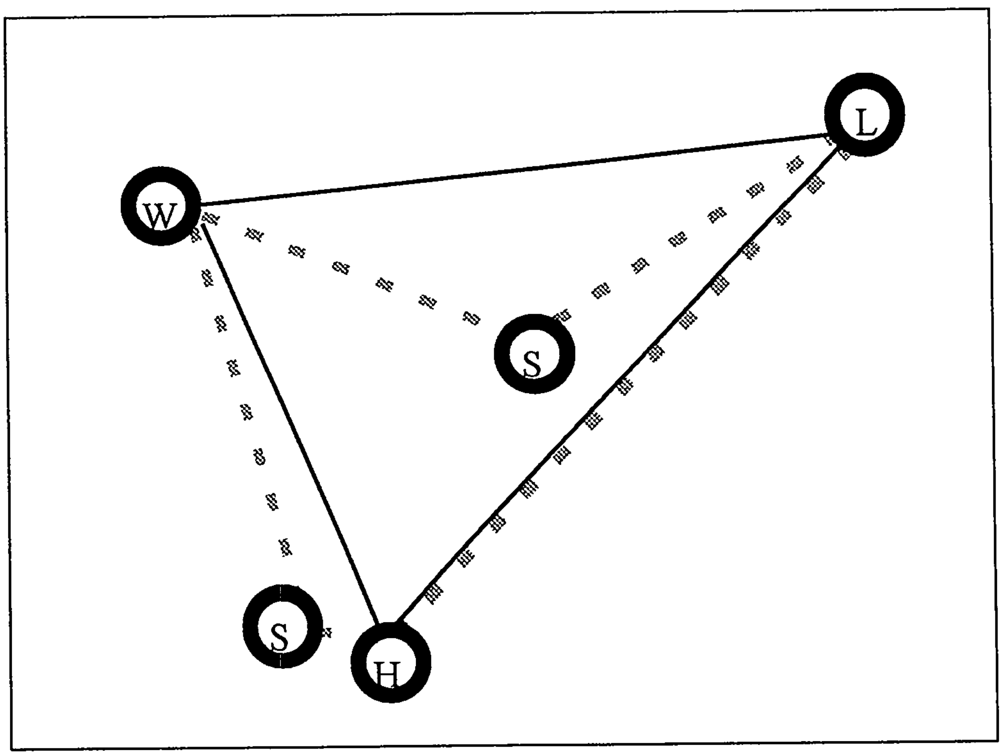
4.3.2 Transformed Observations

The simplifying of chains (elimination of purposes) results in a loss of information. One way to measure this loss of information is to calculate how much observed transport kilometre is lost when the chains are transformed.

The distances between actual destinations are reported in the TU questionnaire, whilst in the transformed data, it is assumed that the tour starts from home goes to the chosen destination and then back home, in the case of a simple tour. Figure 4-2 provides an example:

The IP starts from home and travels a short trip to the kindergarten with the child. From the kindergarten the IP goes to work. After work the IP visits some family far away from home and on the way to the family he/she shops.

Figure 4-2 Simplifying chains, example



The broken line represents the actual tour as it is reported by the IP. The other line represents the simplified tour used for the TDM analysis.

The simplification results in loss of some purposes, in this case two errand purposes drop out. In addition, the simplification introduces new trips which were not actually undertaken.

Table 4-5 provides the average number of purposes in the original diary, grouped on the simplified chains. A comparison of the number of purposes in the original diary, with the number of purposes in the simplified diary, reveals a loss from 2.13 purposes per trip to 1.4 purposes per trip, corresponding to a loss of 34%

due to the simplification of purposes. The loss is spread relatively evenly between the different chain types, with a weak tendency towards a smaller loss in work related chains.

Table 4-5. Loss of purposes

chain	Number of purposes in simplified diary	Average number of purposes in diary
W	1	1.38
W-S	2	2.72
W-L	2	2.69
WS	2	2.92
WS-S	3	4.52
WS-L	3	4.27
WL	2	2.87
WL-S	3	4.15
WL-L	3	4.00
S	1	1.67
SL	2	3.34
L	1	1.78
Total	1.40	2.13

The loss in purposes is also reflected in a loss of travelled kilometre as shown in Table 4-6.

Table 4-6 Loss of kilometre when simplifying chains

	slow	bicycle	car	car pass.	Bus	Train
	Total sample km					
Reported distance	6,803	22,351	289,833	85,060	40,054	26,477
Calculated distance	7,818	16,412	201,005	57,436	36,302	14,915
Share	115%	73%	69%	68%	91%	56%

Note: Reported distance is the total distance reported in the TU survey. Further the figures are weighted to ensure that they are representative.

The loss of kilometres is approx. 30 percent for bicycle, car and car passenger modes, whilst it is 44 percent for the train mode.

We add 15 per cent to the actually reported kilometres for the walk mode due to the assumption that the preferred mode for short trips is walking, and that short trips are not included in the actually reported figure.

The loss is only 9 percent for the bus mode, which reflects the fact that we add some bus kilometres to train modes. The rationale was explained earlier, but involves adding the distance to and from the train station, when the mode is train.

4.3.3 Chain Alternatives

The transformed observations now provide a more manageable description of chains, whilst retaining the most important aspects of travel behaviour. They also highlight the relevant alternative possibilities that individuals can choose, as reactions to policy measures or other changed conditions.

This can be employed to define the chain alternatives for the TDM. After transforming the observations, we end up with the different combinations shown in the table below.

Table 4-7. Chain alternatives

Chain	Frequency	Frequency weighted
0	2,617	2,630
W	2,318	2,238
W-S	771	750
W-L	1,113	1,103
WS	671	636
WS-S	207	191
WS-L	278	265
WL	305	322
WL-S	33	32
WL-L	67	73
S	2,102	2,149
SL	421	446
L	2,641	2,713
Other	1	1
Sum	13,545	13,549

The module "ReduceTours()" has been used to construct the simplified tours. This module is shown in annex 1.

4.4 Mode and Destination Choice Sub-models

A mode and destination choice sub-model is estimated for each possible tour. Table 4-8 illustrates the relationship between the observed (transformed) chains and the sub-models, as well as the number of observations corresponding to each model.

Chains containing only one work tour, W, are modelled in ch12. Chains with a work tour and a errand tour, W-S, occur in two sub-models; the work tour is modelled in c12x and the errand tour is modelled in cx13.

The sub-model cx13 includes all errand tours made as a second tour, i.e. from chains W-S, WS-S and WL-S.

Table 4-8 Tour types and mode/destination sub-models

	ch12	c12x	ch13	ch14	c123	c23x	c24x	c134	cx13	c214	c314
W	2,318										
W-S		771							771		
W-L		1,113								1,113	
WS					671						
WS-S						207			207		
WS-L						278					278
WL							305				
WL-S							33		33		
WL-L							67				67
S			2,102								
SL								421			
L				2,641							

The sub-models for simple tours are specified with 6 alternative modes and 9 alternative destinations, yielding 54 combined mode and destination alternatives. These are set up as logit models, and nested structures with a nest for each mode are estimated.

In the case of triangular tours, where a primary work destination occurs in conjunction with a secondary errand or leisure destination, 4 alternative destinations are available for each primary destination. This yields 36 alternative combinations of destinations and 216 combined mode and destination alternatives. These sub-models are also set up as logit models and nested structures with a nest for each mode are estimated, where possible.

4.5 Chain choice Model

The mode/destination choice sub-models for the different tour types, provide probabilities of mode and destination choice, dependent on choice of tour type or chain.

The choice of chain is determined by many factors, but is expected to be influenced, for the different types of tours, by accessibility to destinations and modes. The accessibility of different tour types is measured by the logsums from the mode/destination choice sub-models.

The model for choice between different chains, is specified with the alternatives illustrated in Table 4-9. The relationship between chains and mode/destination sub-models corresponds to Table 4-8.

Table 4-9. Accessibility measures in chain choice model

Alternative	First tour	Second tour
W	c12	
W-S	c12x	x13
W-L	c12x	214
WS	c123	
WS-S	c23x	x13
WS-L	c23x	314
WL	c24x	
WL-S	c24x	x13
WL-L	c24x	314
SL	c134	
S	13	
L	14	
0		

In addition to the accessibility measures, the model is explained by alternative specific constants for the day of the week plus a number of other socio-economic variables, i.a. sex, age and income.

5 Bibliography

Alogit Users' Guide, Version 3.8 August 1995, Hague Consulting Group, the Hague, Netherlands.

Ben-Akiva, Moshe and Steven R. Lerman: Discrete Choice Analysis - Theory and Application to Travel Demand. MIT-Press, London 1985.

COWIconsult 1994: PETRA - modelling concept, COWIconsult October 1994.

Institute of Transport Economics (TØI) 1992: The National Model System for Private Transport, TØI-rapport 150/1992.

PETRA Working Paper no. 3, Weights, COWI June 1997.

PETRA Working Paper no. 4, The Licence Holding Model, COWI June 1997.

PETRA Working Paper No. 5, The Cohort Model, COWI, June 1997

6 Annex

6.1 Modules used in Access

The following table lists the programs employed to establish the data for the estimations. Databases are placed on the network at COWI:

P:\25127\DATA\TDM\DATABASE\TDM.MDB

Table 6-1 Listing modules used to set up data

	Database	Purpose
ReduceTour()	TDM	Simplifying chains
Mode ()	TDM	Calculating main mode for each tour. Codes: 1=walk, 2=bicycle, 3=cardriver, 4=carpassenger, 5=bus, 6=subway, 7=train, 8=airplaine or ferry, 9=other
Sampledest1()	TDM	Sampling destinations for primary trip
Sampledest2()	TDM	Sampling destinations for detour
RunTDM ()	F_95_INI	Running forecast
Export Logsumdata	TDM1	Macro used to create data files for forecast
Export Logsumdata	TDM2	Macro used to create data files for forecast
Export Logsumdata	TDM3	Macro used to create data files for forecast

6.1.1 ReduceTour

```
Function reducetour()

On Error GoTo errorredtur
sq = ""
'Call cleanup
'Call deltable("turtype")
'Call deltable("W")
'Call deltable("WSL")
'Call deltable("SL")
'Call deltable("SL2")

sq = ""
DoCmd.Echo yes, "tur -> turtype  *** Turtype=1 <=> alle ture korte!"
sq = sq & "SELECT DISTINCTROW AARMD, avg(fraforml) as aff, avg(tilzon) as afz, count(iplobe) as trips, IPLOBE, Avg(U300M) AS TurType "
sq = sq & "INTO Turtype "
sq = sq & "FROM TUR "
sq = sq & "GROUP BY TUR.AARMD, TUR.IPLOBE "
sq = sq & "HAVING ((TUR.AARMD>9500));"
Call runsql(sq)

sq = ""
DoCmd.Echo yes, "turtype -> turtype  ***  der er mindst een tur, der ikke er kort"
sq = sq & "UPDATE DISTINCTROW Turtype "
sq = sq & "SET Turtype.TurType = 3 "
sq = sq & "WHERE ((Turtype.TurType>1));"
Call runsql(sq)

sq = ""
DoCmd.Echo yes, "turtype -> turtype  ***  Vi kender ikke zone på eneste formål > 1"
sq = sq & "UPDATE DISTINCTROW Turtype "
sq = sq & "SET Turtype.TurType = 2 "
sq = sq & "WHERE ((Turtype.TurType>1) AND (Turtype.trips=1) AND (Turtype.aff>1) AND (Turtype.afz=0));"
Call runsql(sq)

sq = ""
DoCmd.Echo yes, "TUR -> TMP   *** Danner midlertidig arbejdsfil"
sq = sq & "SELECT DISTINCTROW tur.totkm, tur.trafmin, 0 as arbtur, bozon, CLng(0) as bozonk, 0 AS TRIPNR, 0 AS DELTURNR, TUR.FRAFORML AS FF, TUR.TILFORML AS TF, TUR.frazon AS FZ, CLng(0) as fzk, TUR.tilzon AS TZ, CLng(0) as tzk, TUR.TURNR, TUR.AARMD, TUR.IPLOBE, TUR.TRKM1, TUR.TRKM2, TUR.TRKM3, TUR.TRKM4, TUR.TRKM5, TUR.TRKM6, TUR.TRKM7, TUR.TRKM8, TUR.TRKM9, TUR.TRKM10, TUR.TRKM11, TUR.TRKM12, TUR.TRKM13, TUR.TRKM15, TUR.TRKM16, TUR.TRKM17, TUR.TRKM18, TUR.TRKM19, Turtype.TurType "
sq = sq & "INTO Tmp "
sq = sq & "FROM TUR LEFT JOIN Turtype ON (TUR.IPLOBE = Turtype.IPLOBE) AND (TUR.AARMD = Turtype.AARMD) "
```

```
sq = sq & "WHERE (TUR.AARMD>9500) "
sq = sq & "ORDER BY Tur.AARMD, Tur.iPLOBE, Tur.TURNR;"
Call runsql(sq)

sq = ""
'sq = sq & "WHERE ((TUR.AARMD>9500) AND (tur.bokom<>741) AND
(Turtype.TurType=3)) "

DoCmd.Echo yes, "Sætter værdi på delture og tripnr i de enkelte trip i hver deltur"
Set MYDB = DBEngine.Workspaces(0).Databases(0)
Set m = MYDB.OpenRecordset("tmp", DB_OPEN_DYNASET)

Do
  a = 1: id = (m.AARMD * 100000 + m.IPLOBE)
   Do While id = (m.AARMD * 100000 + m.IPLOBE)
     b = 1: tf1 = 2
       Do While tf1 > 1 And id = (m.AARMD * 100000 + m.IPLOBE)
         m.Edit
         m.delturnr = a: m.tripnr = b
         m.Update
         tf1 = m.TF
         m.MoveNext
       If m.EOF Then GoTo jumpReducedTour1
         b = b + 1
       Loop
     a = a + 1
   Loop
Loop
jumpReducedTour1:
m.Close

DoCmd.Echo yes, "Beregner hovedtransportmiddel i underprogram"
Call MODE

DoCmd.Echo yes, "Tmp -> Tmp  ***  tilforml 5 -> 2"
Call runsql("UPDATE DISTINCTROW tmp SET tmp.TF = 2 WHERE ((tmp.TF=5));")

DoCmd.Echo yes, "Tmp+oversæt -> Tmp  ***  BO -zoner tilpasset afstandsmatricer"
sq = sq & "UPDATE DISTINCTROW Tmp "
sq = sq & "LEFT JOIN overLczonTilLczonk AS b ON Tmp.bozon = b.LCzon "
sq = sq & "SET Tmp.bozonk = [b].[lczonk];"
Call runsql(sq)

DoCmd.Echo yes, "Tmp+oversæt -> Tmp  ***  FRA-zoner tilpasset afstandsmatricer"
sq = sq & "UPDATE DISTINCTROW Tmp "
sq = sq & "LEFT JOIN overLczonTilLczonk AS b ON Tmp.fz = b.LCzon "
sq = sq & "SET Tmp.fzk = [b].[lczonk];"
Call runsql(sq)

DoCmd.Echo yes, "Tmp+oversæt -> Tmp  ***  TIL-zoner tilpasset afstandsmatricer"
```

```
sq = sq & "UPDATE DISTINCTROW Tmp "
sq = sq & "LEFT JOIN overLczonTilLczonk AS b ON Tmp.tz = b.LCzon "
sq = sq & "SET Tmp.tzk = [b].[lczonk];"
Call runsql(sq)

'DoCmd Echo yes, "Tmp -> Tmp  ***  recode zoner, missing=0"
'Call runsql("UPDATE DISTINCTROW tmp SET BOZON = 0 WHERE ((BOZON Is Null));")
'Call runsql("UPDATE DISTINCTROW tmp SET FZ = 0 WHERE ((FZ Is Null));")
'Call runsql("UPDATE DISTINCTROW tmp SET TZ = 0 WHERE ((TZ Is Null));")

DoCmd.Echo yes, "Tmp -> Tmp  *** Definerer arbejdstrip"
Call runsql("UPDATE DISTINCTROW tmp SET tmp.arbtur = 1 WHERE ((tmp.TF=2));")

DoCmd.Echo yes, "Tmp -> Tmp01  *** Definerer arb.deltur"
sq = sq & "SELECT DISTINCTROW AARMD, IPLOBE, DELTURNR, Sum(arbtur) AS
SumOfarbtur "
sq = sq & "INTO tmp01 "
sq = sq & "FROM tmp "
sq = sq & "GROUP BY AARMD, IPLOBE, DELTURNR;"
Call runsql(sq)

DoCmd.Echo yes, "Tmp + Tmp01 -> tmp  *** hægter oplysning på"
sq = sq & "UPDATE DISTINCTROW tmp "
sq = sq & "LEFT JOIN tmp01 ON (tmp.DELTURNR = tmp01.DELTURNR) AND
(tmp.IPLOBE = tmp01.IPLOBE) AND (tmp.AARMD = tmp01.AARMD) "
sq = sq & "SET tmp.arbtur = 1 "
sq = sq & "WHERE ((tmp01.SumOfarbtur>0));"
Call runsql(sq)

DoCmd.Echo yes, "Tmp -> tmpejarb  *** samler delture uden arbejde"
sq = sq & "SELECT DISTINCTROW tmp.* "
sq = sq & "INTO tmpejarb "
sq = sq & "FROM Tmp "
sq = sq & "WHERE ((tmp.arbtur=0 and tf>2));"
Call runsql(sq)

DoCmd.Echo yes, "Tmp -> tmparbW  *** samler arbejdsdelture med arb som formål"
sq = sq & "SELECT DISTINCTROW tmp.* "
sq = sq & "INTO tmparbW "
sq = sq & "FROM tmp "
sq = sq & "WHERE ((arbtur=1 and tf=2));"
Call runsql(sq)

DoCmd.Echo yes, "Tmp -> tmparbLS  *** samler arbejdsdelture uden arb som formål"
sq = sq & "SELECT DISTINCTROW tmp.* "
sq = sq & "INTO tmparbLS "
sq = sq & "FROM tmp "
sq = sq & "WHERE ((arbtur=1 and tf>2));"
Call runsql(sq)
```

```
DoCmd.Echo yes, "tmparbW -> W *** hægter afstande på"
sq = sq & "SELECT DISTINCTROW AARMD, IPLOBE, arbtur, DELTURNR, TRIPNR,
turnr, bozon, tz as wzon, bozonk, TZk AS Wzonk, -.1 AS wdist, -.1 AS wtid "
sq = sq & "INTO W "
sq = sq & "FROM tmparbW;"
Call runsql(sq)

DoCmd.Echo yes, "tmparbW -> W *** Jylland"
sq = sq & "UPDATE DISTINCTROW "
sq = sq & "W LEFT JOIN jylland as a ON (W.WZonK = a.Til) AND (W.bozonk = a.Fra) "
sq = sq & "SET W.wdist = [distance]/10 + (([delturnr]*10+[tripnr])/10000), W.wtid = [tid] "
sq = sq & "WHERE ((a.Til Is Not Null) AND (a.Fra Is Not Null));"
Call runsql(sq)

DoCmd.Echo yes, "tmparbW -> W *** Sjælland"
sq = sq & "UPDATE DISTINCTROW "
sq = sq & "W LEFT JOIN sjælland as a ON (W.WZonK = a.Til) AND (W.bozonk = a.Fra) "
sq = sq & "SET W.wdist = [distance]/10 + (([delturnr]*10+[tripnr])/10000), W.wtid = [tid] "
sq = sq & "WHERE ((a.Til Is Not Null) AND (a.Fra Is Not Null));"
Call runsql(sq)

DoCmd.Echo yes, "tmparbW -> W *** Bornholm"
sq = sq & "UPDATE DISTINCTROW "
sq = sq & "W LEFT JOIN bornholm as a ON (W.WZonK = a.Til) AND (W.bozonk = a.Fra) "
sq = sq & "SET W.wdist = [distance]/10 + (([delturnr]*10+[tripnr])/10000), W.wtid = [tid] "
sq = sq & "WHERE ((a.Til Is Not Null) AND (a.Fra Is Not Null));"
Call runsql(sq)

DoCmd.Echo yes, "W -> W  *** vælger arbture længst væk fra bolig"
sq = sq & "SELECT DISTINCTROW W.AARMD, W.IPLOBE, Max(W.wdist) AS
MaxOfwdist "
sq = sq & "INTO tmp02 "
sq = sq & "FROM W "
sq = sq & "WHERE ((W.wdist>=0)) "
sq = sq & "GROUP BY W.AARMD, W.IPLOBE;"
Call runsql(sq)

sq = sq & "DELETE DISTINCTROW W.*, W.wdist "
sq = sq & "FROM W LEFT JOIN tmp02 ON (W.IPLOBE = tmp02.IPLOBE) AND
(W.AARMD = tmp02.AARMD) "
sq = sq & "WHERE ((W.wdist<[MaxOfWdist] or W.wdist = -.1));"
Call runsql(sq)

Call deltable("WSL")
DoCmd.Echo yes, "tmparbW + tmparbLS -> WSL  ***  Hægter Arbzoneoplysninger på
øvrige formål"
sq = sq & "SELECT DISTINCTROW tmparbLS.AARMD, tmparbLS.IPLOBE,
tmparbLS.arbtur, tmparbLS.DELTURNR, tmparbLS.TRIPNR, tmparbLS.TuRNR,
tmparbLS.TF, tmparbLS.TZ, tmparbls.tzk, -0.1 AS WFdist, -0.1 AS FHdist, W.bozon,
W.bozonk, W.Wzon, W.WZonk, W.wdist, W.wtid "
```

```
sq = sq & "INTO WSL "
sq = sq & "FROM tmparbLS LEFT JOIN W ON (tmparbLS.AARMD = W.AARMD) AND (tmparbLS.IPLOBE = W.IPLOBE) AND (tmparbLS.DELTURNR = W.DELTURNR);"
Call runsql(sq)

DoCmd.Echo yes, "WSL -> WSL *** hægter jyllands afstande på (fra arbzone)"
sq = sq & "UPDATE DISTINCTROW WSL LEFT JOIN jylland as a ON (WSL.tzk = a.Til) AND (WSL.Wzonk = a.Fra) "
sq = sq & "SET WSL.wfdist = [distance]/10 + (([delturnr]*10+[tripnr])/10000) "
sq = sq & "WHERE ((a.Til Is Not Null) AND (a.Fra Is Not Null));"
Call runsql(sq)

DoCmd.Echo yes, "WSL -> WSL *** hægter sjællands afstande på (fra arbzone)"
sq = sq & "UPDATE DISTINCTROW WSL LEFT JOIN sjælland as a ON (WSL.tzk = a.Til) AND (WSL.Wzonk = a.Fra) "
sq = sq & "SET WSL.wfdist = [distance]/10 + (([delturnr]*10+[tripnr])/10000) "
sq = sq & "WHERE ((a.Til Is Not Null) AND (a.Fra Is Not Null));"
Call runsql(sq)

DoCmd.Echo yes, "WSL -> WSL *** hægter bornholms afstande på (fra arbzone)"
sq = sq & "UPDATE DISTINCTROW WSL LEFT JOIN bornholm as a ON (WSL.tzk = a.Til) AND (WSL.Wzonk = a.Fra) "
sq = sq & "SET WSL.wfdist = [distance]/10 + (([delturnr]*10+[tripnr])/10000) "
sq = sq & "WHERE ((a.Til Is Not Null) AND (a.Fra Is Not Null));"
Call runsql(sq)

DoCmd.Echo yes, "WSL -> WSL *** hægter jyllands afstande på (fra bozone)"
sq = sq & "UPDATE DISTINCTROW WSL LEFT JOIN jylland as a ON (WSL.tzk = a.Til) AND (WSL.Bozonk = a.Fra) "
sq = sq & "SET WSL.fhdist = [distance]/10 + (([delturnr]*10+[tripnr])/10000) "
sq = sq & "WHERE ((a.Til Is Not Null) AND (a.Fra Is Not Null));"
Call runsql(sq)

DoCmd.Echo yes, "WSL -> WSL *** hægter sjællands afstande på (fra bozone)"
sq = sq & "UPDATE DISTINCTROW WSL LEFT JOIN sjælland as a ON (WSL.tzk = a.Til) AND (WSL.Bozonk = a.Fra) "
sq = sq & "SET WSL.fhdist = [distance]/10 + (([delturnr]*10+[tripnr])/10000) "
sq = sq & "WHERE ((a.Til Is Not Null) AND (a.Fra Is Not Null));"
Call runsql(sq)

DoCmd.Echo yes, "WSL -> WSL *** hægter bornholms afstande på (fra bozone)"
sq = sq & "UPDATE DISTINCTROW WSL LEFT JOIN bornholm as a ON (WSL.tzk = a.Til) AND (WSL.Bozonk = a.Fra) "
sq = sq & "SET WSL.fhdist = [distance]/10 + (([delturnr]*10+[tripnr])/10000) "
sq = sq & "WHERE ((a.Til Is Not Null) AND (a.Fra Is Not Null));"
Call runsql(sq)

DoCmd.Echo yes, "WSL -> WSL *** fjerner få med manglende afstandsoplysninger"
sq = sq & "DELETE DISTINCTROW WSL.*, WSL.WFdist, WSL.FHdist "
sq = sq & "FROM WSL "
```

```
sq = sq & "WHERE ((WSL.WFdist=-0.1)) OR ((WSL.FHdist=-0.1));"
Call runsql(sq)

DoCmd.Echo yes, "WSL -> WSL  *** vælger det formål der giver størst omvej på vej hjem"
sq = sq & "SELECT DISTINCTROW WSL.AARMD, WSL.IPLOBE, WSL.DELTURNR,
Max([wfdist]+[fhdist]-[wdist]) AS MaxMargDist "
sq = sq & "INTO tmp03 "
sq = sq & "FROM WSL "
sq = sq & "GROUP BY WSL.AARMD, WSL.IPLOBE, WSL.DELTURNR;"
Call runsql(sq)

sq = sq & "DELETE DISTINCTROW WSL.*, [wfdist]+[fhdist]-[wdist] AS margdist "
sq = sq & "FROM WSL LEFT JOIN tmp03 ON (WSL.DELTURNR = tmp03.DELTURNR)
AND (WSL.IPLOBE = tmp03.IPLOBE) AND (WSL.AARMD = tmp03.AARMD) "
sq = sq & "WHERE ((([wfdist]+[fhdist]-[wdist])<[MaxMargDist]));"
Call runsql(sq)

sq = sq & "SELECT DISTINCTROW AARMD, IPLOBE, DELTURNR, TRIPNR, turnr,
bozon, bozonk, arbtur, TF, TZ, TZK, -.1 as FHdist "
sq = sq & "INTO SL "
sq = sq & "FROM tmpejarb;"
Call runsql(sq)

DoCmd.Echo yes, "SL -> SL *** hægter jyllands afstande på (fra bozone)"
sq = sq & "UPDATE DISTINCTROW SL LEFT JOIN jylland as a ON (SL.tzk = a.Til) AND
(SL.Bozonk = a.Fra) "
sq = sq & "SET SL.fhdist = [distance]/10 + (([delturnr]*10+[tripnr])/10000) "
sq = sq & "WHERE ((a.Til Is Not Null) AND (a.Fra Is Not Null));"
Call runsql(sq)

DoCmd.Echo yes, "SL -> SL *** hægter sjællands afstande på (fra bozone)"
sq = sq & "UPDATE DISTINCTROW SL LEFT JOIN sjælland as a ON (SL.tzk = a.Til) AND
(SL.Bozonk = a.Fra) "
sq = sq & "SET SL.fhdist = [distance]/10 + (([delturnr]*10+[tripnr])/10000) "
sq = sq & "WHERE ((a.Til Is Not Null) AND (a.Fra Is Not Null));"
Call runsql(sq)

DoCmd.Echo yes, "SL -> SL *** hægter bornholms afstande på (fra bozone)"
sq = sq & "UPDATE DISTINCTROW SL LEFT JOIN bornholm as a ON (SL.tzk = a.Til)
AND (SL.Bozonk = a.Fra) "
sq = sq & "SET SL.fhdist = [distance]/10 + (([delturnr]*10+[tripnr])/10000) "
sq = sq & "WHERE ((a.Til Is Not Null) AND (a.Fra Is Not Null));"
Call runsql(sq)

DoCmd.Echo yes, "SL -> SLt2  *** gemmer data til formål 2"
Call runsql("SELECT DISTINCTROW SL.* INTO SLt2 FROM SL;")

DoCmd.Echo yes, "SL -> tmp04 *** Finder max distance"
sq = sq & "SELECT DISTINCTROW SL.AARMD, SL.IPLOBE, Max(SL.FHdist) AS
MaxOfFHdist "
```

```
sq = sq & "INTO tmp04 "
sq = sq & "FROM SL "
sq = sq & "GROUP BY SL.AARMD, SL.IPLOBE;"
Call runsql(sq)

DoCmd.Echo yes, "tmp04 -> tmpejarbmissdist *** finder IP'er uden korrekte afstande"
sq = sq & "SELECT DISTINCTROW tmp04.AARMD, tmp04.IPLOBE, tmp04.MaxOfFHdist
"
sq = sq & "INTO tmpejarbmissdist "
sq = sq & "FROM tmp04 "
sq = sq & "WHERE ((tmp04.MaxOfFHdist=-0.0001));"
Call runsql(sq)

DoCmd.Echo yes, "SL -> SL *** beholder trip med størst korrekt afstand"
sq = sq & "DELETE DISTINCTROW SL.*, tmp04.MaxOfFHdist, tmp04.MaxOfFHdist "
sq = sq & "FROM SL LEFT JOIN tmp04 ON (SL.AARMD = tmp04.AARMD) AND
(SL.IPLOBE = tmp04.IPLOBE) "
sq = sq & "WHERE ((tmp04.MaxOfFHdist=-0.1)) OR ((tmp04.MaxOfFHdist>[fhdist]));"
Call runsql(sq)

DoCmd.Echo yes, "SLt2+SL -> SL2  *** gemmer data til formål 2"
sq = sq & "SELECT DISTINCTROW SLt2.*, SL.TF "
sq = sq & "INTO SL2 "
sq = sq & "FROM SL LEFT JOIN SLt2 ON (SL.AARMD = SLt2.AARMD) AND (SL.IPLOBE
= SLt2.IPLOBE) AND (SL.DELTURNR = SLt2.DELTURNR) "
sq = sq & "ORDER BY SL.AARMD, SL.IPLOBE;"
Call runsql(sq)

DoCmd.Echo yes, "SL2+SLt2-> SL2  ***  beholder formål forskellige fra dem vi allerede
har"
sq = sq & "DELETE DISTINCTROW SL2.*, SL2.SL_TF "
sq = sq & "FROM SL2 "
sq = sq & "WHERE ((SL2.SL_TF=[SLt2_TF]));"
Call runsql(sq)

DoCmd.Echo yes, "SL -> tmp05 *** Finder max distance"
sq = sq & "SELECT DISTINCTROW SL2.AARMD, SL2.IPLOBE, Max(SL2.FHdist) AS
MaxOfFHdist "
sq = sq & "INTO tmp05 "
sq = sq & "FROM SL2 "
sq = sq & "GROUP BY SL2.AARMD, SL2.IPLOBE;"
Call runsql(sq)

DoCmd.Echo yes, "tmp05 -> tmpejarbmissdist1 *** finder IP'er uden korrekte afstande"
sq = sq & "SELECT DISTINCTROW tmp05.AARMD, tmp05.IPLOBE, tmp05.MaxOfFHdist
"
sq = sq & "INTO tmpejarbmissdist1 "
sq = sq & "FROM tmp05 "
sq = sq & "WHERE ((tmp05.MaxOfFHdist=-0.0001));"
Call runsql(sq)
```

```
DoCmd.Echo yes, "SL2 -> SL2 *** beholder trip med størst korrekt afstand"
sq = sq & "DELETE DISTINCTROW SL2.*, tmp05.MaxOfFHdist, tmp05.MaxOfFHdist "
sq = sq & "FROM SL2 LEFT JOIN tmp05 ON (SL2.AARMD = tmp05.AARMD) AND
(SL2.IPLOBE = tmp05.IPLOBE) "
sq = sq & "WHERE ((tmp05.MaxOfFHdist=-0.1)) OR ((tmp05.MaxOfFHdist>[fhdist]));"
Call runsql(sq)

Call cleanup

Exit Function

errorredtur:
n = Err
If n = 3011 Or n = 3078 Then
 DoCmd.Echo yes, "Kunne ikke finde objekt"
 Resume Next
End If
response = MsgBox(Error$, 2)
If response = 3 Then 'abort
   Reset
   Exit Function
   End If
If response = 4 Then Resume Else Resume Next

End Function
```

6.1.2 Mode

```
Sub MODE()

DoCmd.Echo yes, "Aggregerer km på 9 forskellige modes"
Call runsql("SELECT DISTINCTROW tmp.AARMD, tmp.IPLOBE, tmp.DELTURNR,
Sum(tmp.totkm) AS dist, Sum(tmp.trafmin) AS tid, Avg(0) AS totkm, Sum(1) AS
countdeltrips, Avg(0) AS MainMode, Avg(0) AS strkm0, Sum(tmp.TRKM1) AS Strkm1,
Sum(trkm2)+Sum(trkm3) AS Strkm2,
Sum(trkm4)+Sum(trkm5)+Sum(trkm7)+Sum(trkm8)+Sum(trkm16) AS Strkm3,
Sum(trkm6)+Sum(trkm15)+Sum(trkm17) AS Strkm4, Sum(tmp.TRKM9) AS Strkm5,
Sum(tmp.TRKM10) AS Strkm6, Sum(tmp.TRKM11) AS Strkm7,
Sum(trkm12)+Sum(trkm13) AS Strkm8, Sum(trkm18)+Sum(trkm19) AS Strkm9 INTO
tmpmodeslet FROM tmp GROUP BY tmp.AARMD, tmp.IPLOBE, tmp.DELTURNR;")

DoCmd.Echo yes, "Finder hovedtransportmiddel"
Set MYDB = DBEngine.Workspaces(0).Databases(0)
Set m = MYDB.OpenRecordset("tmpmodeslet", DB_OPEN_DYNASET) ' Create dynaset.


 MainMode = 0
 TOTKM = 0

Do Until m.EOF
 MainMode = -1
 TOTKM = 0
 max = 0
    For I = 1 To 9
      If max < m("strkm" & CStr(I)) Then MainMode = I
      If max < m("strkm" & CStr(I)) Then max = m("strkm" & CStr(I))
      TOTKM = TOTKM + m("strkm" & CStr(I))
      Next
   m.Edit
   m.MainMode = MainMode
   m.TOTKM = TOTKM
   m.Update
   m.Move 1
 Loop
 m.Close

 End Sub
```

6.1.3 Sampledest1

```
Function sampledest1()
On Error GoTo errordest1


'Call deltable("sampleZoner")
UPPER = 175

Set MYDB = DBEngine.Workspaces(0).Databases(0)

DoCmd.Echo yes, "TDMDATA + strata1 -> sampleZoner  *** opretter tabel"
sq = ""
sq = sq & "SELECT DISTINCTROW T.AARMD, T.IPLOBE, T.ID, T.Hzon AS Bozon,
T.hrural, T.LANDZONES, T.HKOM, T.HKID, T.Hkomid, T.HagrKOM, T.HKomAgrID,
T.H38zon, T.Dest1, 0 as strata3, 1000000 AS altstr11, 1000000 AS altstr12, 1000000 AS
altstr13, 1000000 AS altstr14, 1000000 AS altstr41, 1000000 AS altstr42, 1000000 AS
altstr51, 1000000 AS altstr61, 1000000 AS altstr71 "
sq = sq & "INTO sampleZoner "
sq = sq & "FROM TDMDATA AS T "
sq = sq & "WHERE ((T.AARMD > 9500) And (T.BadRecord = 0)) "
sq = sq & "ORDER BY T.HagrKOM, T.H38zon;"
Call runsql(sq)


'1***** Strata1a (boKommuner uden landzoner)**********************************************

DoCmd.Echo yes, "Tæller antal zoner i stratak1  (bozoner)"
For T = 1 To UPPER
sq2 = "SELECT DISTINCTROW sT.zon FROM strata1 AS ST WHERE ST.GRP=" &
Str$(T) & ";"

Set MYDB = DBEngine.Workspaces(0).Databases(0)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)

nrec = 0
 source.MoveFirst
Do
 nrec = nrec + 1
 source.MoveNext
 If source.EOF Then GoTo Jumpk2
Loop
Jumpk2:
source.Close

numrec(T) = nrec

txtant = "***  grp = " & Str$(T) & " har " & Str$(nrec) & " zoner  ***"
DoCmd.Echo yes, txtant
```

```
Next

'***************************************************************

DoCmd.Echo yes, "definerer datasæt"
For T = 1 To UPPER
kriterie = "(sampleZoner.LANDZONES = 0 or sampleZoner.hrural = 1) AND
sampleZoner.hKOMAGRID = " & Str$(T)
nx = DCount("[ID]", "sampleZoner", kriterie)
If nx = 0 Then GoTo Jumpknx11
sq1 = "SELECT DISTINCTROW sampleZoner.*, sampleZoner.HKOMAGRID AS GRP
FROM sampleZoner WHERE " & kriterie & ";"
sq2 = "SELECT DISTINCTROW strata1.zon FROM strata1 WHERE strata1.grp=" &
Str$(T) & ";"


Set sample = MYDB.OpenRecordset(sq1, DB_OPEN_DYNASET)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)

For k = 1 To 4
txtant = "1***  sampler fra strata 1, grp = " & Str$(T) & ", opdaterer Altstr1" & CStr(k)
DoCmd.Echo yes, txtant
sample.MoveFirst
Do
TRY = 0
GoTo nextline1
nyzonk1:
TRY = TRY + 1
nextline1:
If TRY > 50 Then GoTo errordest1
Randomize
source.MoveFirst
source.Move Int(numrec(T) * Rnd)
zone = source.zon
  If zone = sample.Bozon Then GoTo nyzonk1
  If zone = sample.altstr11 Then GoTo nyzonk1
  If zone = sample.altstr12 Then GoTo nyzonk1
  If zone = sample.altstr13 Then GoTo nyzonk1
sample.Edit
  sample("altstr1" & CStr(k)) = zone
  sample.Strata3 = 1
  sample.Update
  sample.MoveNext
If sample.EOF Then GoTo Jumpk1
Loop
Jumpk1:
Next
sample.Close
source.Close
Jumpknx11:
```

```
Next

'2***** Strata2 (boKommuner med landzoner)************************************************

UPPER = 175

DoCmd.Echo yes, "Tæller antal zoner i stratak1  (bozoner)"
For T = 1 To UPPER
sq2 = "SELECT DISTINCTROW sT.zon FROM strata1 AS ST WHERE ST.GRP=" &
Str$(T) & ";"

Set MYDB = DBEngine.Workspaces(0).Databases(0)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)

nrec = 0
  source.MoveFirst
Do
  nrec = nrec + 1
  source.MoveNext
  If source.EOF Then GoTo Jumpk1b2
Loop
Jumpk1b2:
source.Close

numrec(T) = nrec

txtant = "***  grp = " & Str$(T) & " har " & Str$(nrec) & " zoner  ***"
DoCmd.Echo yes, txtant

Next

'*************************************************************

DoCmd.Echo yes, "definerer datasæt"
For T = 1 To UPPER
kriterie = "((sampleZoner.hrural = 1 and sampleZoner.LANDZONES > 1) or
(sampleZoner.hrural = 0 and sampleZoner.LANDZONES > 0))  AND
sampleZoner.hKOMAGRID = " & Str$(T)
nx = DCount("[ID]", "sampleZoner", kriterie)
If nx = 0 Then GoTo Jumpknx1b1
sq1 = "SELECT DISTINCTROW sampleZoner.*, sampleZoner.HKOMAGRID AS GRP
FROM sampleZoner WHERE " & kriterie & ";"
sq2 = "SELECT DISTINCTROW strata1.zon FROM strata1 WHERE strata1.grp=" &
Str$(T) & ";"


Set sample = MYDB.OpenRecordset(sq1, DB_OPEN_DYNASET)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)

For k = 1 To 3
```

```
txtant = "2*** sampler fra strata 1, grp = " & Str$(T) & ", opdaterer Altstr1" & CStr(k)
DoCmd.Echo yes, txtant
sample.MoveFirst
Do
TRY = 0
GoTo nextline1b1
nyzonk1b1:
TRY = TRY + 1
nextline1b1:
If TRY > 50 Then GoTo errordest1
source.MoveFirst
source.Move Int(numrec(T) * Rnd)
zone = source.zon
 If zone = sample.Bozon Then GoTo nyzonk1b1
 If zone = sample.altstr11 Then GoTo nyzonk1b1
 If zone = sample.altstr12 Then GoTo nyzonk1b1
 If zone = sample.altstr13 Then GoTo nyzonk1b1
sample.Edit
 sample("altstr1" & CStr(k)) = zone
 sample.Strata3 = 2
 sample.Update
 sample.MoveNext
If sample.EOF Then GoTo Jumpk1b1
Loop
Jumpk1b1:
Next
sample.Close
source.Close
Jumpknx1b1:
Next



'3***** Strata3 (landzoner i bokommune)**************************************************

UPPER = 246

DoCmd.Echo yes, "Tæller antal zoner i stratak1 (bozoner)"
For T = 1 To UPPER
sq2 = "SELECT DISTINCTROW sT.zon FROM strata3 AS ST WHERE ST.GRP=" &
Str$(T) & ";"

Set MYDB = DBEngine.Workspaces(0).Databases(0)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)

nrec = 0
 source.MoveFirst
Do
 nrec = nrec + 1
 source.MoveNext
```

```
 If source.EOF Then GoTo Jumpk22
Loop
Jumpk22:
source.Close

numrec(T) = nrec

txtant = "*** grp = " & Str$(T) & " har " & Str$(nrec) & " zoner ***"
DoCmd.Echo yes, txtant

Next

'************************************************************


'Destzone skulle ud inden sampling, men med eet alternativ har det ingen betydning
'Sampler landzone hvor der er en (ud over evt. bozone)

DoCmd.Echo yes, "definerer datasæt"
For T = 1 To UPPER
kriterie = "((sampleZoner.hrural = 1 and sampleZoner.LANDZONES > 1) or
(sampleZoner.hrural = 0 and sampleZoner.LANDZONES > 0)) AND sampleZoner.hKOMID
= " & Str$(T)
nx = DCount("[ID]", "sampleZoner", kriterie)
If nx = 0 Then GoTo Jumpknx21
sq1 = "SELECT DISTINCTROW sampleZoner.*, sampleZoner.HKOMID AS GRP FROM
sampleZoner WHERE " & kriterie & ";"
sq2 = "SELECT DISTINCTROW strata3.zon FROM strata3 WHERE strata3.grp=" &
Str$(T) & ";"


Set sample = MYDB.OpenRecordset(sq1, DB_OPEN_DYNASET)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)


txtant = "3*** sampler fra strata 3, grp = " & Str$(T) & ", opdaterer Altstr14" & CStr(k)
DoCmd.Echo yes, txtant
sample.MoveFirst
Do
TRY = 0
GoTo nextline21
nyzonk21:
TRY = TRY + 1
nextline21:
If TRY > 50 Then GoTo errordest1
Randomize
source.MoveFirst
source.Move Int(numrec(T) * Rnd)
zone = source.zon
 If zone = sample.Bozon Then GoTo nyzonk21
sample.Edit
 sample.altstr14 = zone
```

```
 sample.Strata3 = 3
 sample.Update
 sample.MoveNext
If sample.EOF Then GoTo Jumpk21
Loop
Jumpk21:
sample.Close
source.Close
Jumpknx21:

Next


'***** Strata4 (Byzoner i pendlzone)************************************************
'***************************************************************************************


UPPER = 175

DoCmd.Echo yes, "Tæller antal zoner i strata4  (byzoner i pendlzone)"
For T = 1 To UPPER
sq2 = "SELECT DISTINCTROW sT.zon FROM strata4 AS ST WHERE ST.GRP=" &
Str$(T) & ";"

Set MYDB = DBEngine.Workspaces(0).Databases(0)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)

nrec = 0
 source.MoveFirst
Do
 nrec = nrec + 1
 source.MoveNext
 If source.EOF Then GoTo Jumpk42
Loop
Jumpk42:
source.Close

numrec(T) = nrec

txtant = "***  grp = " & Str$(T) & " har " & Str$(nrec) & " zoner  ***"
DoCmd.Echo yes, txtant

Next

'*****************************************************************


DoCmd.Echo yes, "definerer datasæt"
For T = 1 To UPPER
kriterie = "sampleZoner.hKOMAGRID = " & Str$(T)
nx = DCount("[ID]", "sampleZoner", kriterie)
If nx = 0 Then GoTo Jumpknx41
```

```
sq1 = "SELECT DISTINCTROW sampleZoner.*, sampleZoner.HKOMAGRID AS GRP
FROM sampleZoner WHERE " & kriterie & ";"
sq2 = "SELECT DISTINCTROW sT.zon FROM strata4 AS ST WHERE ST.GRP=" &
Str$(T) & ";"


Set sample = MYDB.OpenRecordset(sq1, DB_OPEN_DYNASET)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)

For k = 1 To 2
txtant = "4*** sampler fra strata 4, grp = " & Str$(T) & ", opdaterer Altstr4" & CStr(k)
DoCmd.Echo yes, txtant
sample.MoveFirst
Do
TRY = 0
GoTo nextline41
nyzonk41:
TRY = TRY + 1
nextline41:
If TRY > 50 Then GoTo errordest1
Randomize
source.MoveFirst
source.Move Int(numrec(T) * Rnd)
zone = source.zon
  If zone = sample.Bozon Then GoTo nyzonk41
  If zone = sample.altstr41 Or zone = sample.Dest1 Then GoTo nyzonk41
  If zone = sample.altstr11 Then GoTo nyzonk41
  If zone = sample.altstr12 Then GoTo nyzonk41
  If zone = sample.altstr13 Then GoTo nyzonk41
  If zone = sample.altstr14 Then GoTo nyzonk41
sample.Edit
  sample("altstr4" & CStr(k)) = zone
  sample.Update
  sample.MoveNext
If sample.EOF Then GoTo Jumpk41
Loop
Jumpk41:
Next
sample.Close
source.Close
Jumpknx41:
Next

'***** Strata5 (Byzoner i restzone)*********************************************
'*******************************************************************************

UPPER = 38

DoCmd.Echo yes, "Tæller antal zoner i strata5  (byzoner i restzone)"
For T = 1 To UPPER
```

```
sq2 = "SELECT DISTINCTROW sT.zon FROM strata5 AS ST WHERE ST.GRP=" &
Str$(T) & ";"

Set MYDB = DBEngine.Workspaces(0).Databases(0)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)

nrec = 0
 source.MoveFirst
Do
 nrec = nrec + 1
 source.MoveNext
 If source.EOF Then GoTo Jumpk52
Loop
Jumpk52:
source.Close

numrec(T) = nrec

txtant = "***  grp = " & Str$(T) & " har " & Str$(nrec) & " zoner  ***"
DoCmd.Echo yes, txtant

Next

'****************************************************************

DoCmd.Echo yes, "definerer datasæt"
For T = 1 To UPPER
kriterie = "sampleZoner.h38zon = " & Str$(T)
nx = DCount("[ID]", "sampleZoner", kriterie)
If nx = 0 Then GoTo Jumpknx51
sq1 = "SELECT DISTINCTROW sampleZoner.*, sampleZoner.H38zon AS GRP FROM
sampleZoner WHERE " & kriterie & ";"
sq2 = "SELECT DISTINCTROW sT.zon FROM strata5 AS ST WHERE ST.GRP=" &
Str$(T) & ";"


Set sample = MYDB.OpenRecordset(sq1, DB_OPEN_DYNASET)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)

txtant = "1***  sampler fra strata 5, grp = " & Str$(T) & ", opdaterer Altstr5" & CStr(k)
DoCmd.Echo yes, txtant
sample.MoveFirst
Do
TRY = 0
GoTo nextline51
nyzonk51:
TRY = TRY + 1
nextline51:
If TRY > 50 Then GoTo errordest1
Randomize
```

```
source.MoveFirst
source.Move Int(numrec(T) * Rnd)
zone = source.zon
  If zone = sample.Bozon Then GoTo nyzonk51
  If zone = sample.Dest1 Then GoTo nyzonk51
  If zone = sample.altstr11 Then GoTo nyzonk51
  If zone = sample.altstr12 Then GoTo nyzonk51
  If zone = sample.altstr13 Then GoTo nyzonk51
  If zone = sample.altstr14 Then GoTo nyzonk51
  If zone = sample.altstr41 Then GoTo nyzonk51
  If zone = sample.altstr42 Then GoTo nyzonk51
sample.Edit
  sample.altstr51 = zone
  sample.Update
  sample.MoveNext
If sample.EOF Then GoTo Jumpk51
Loop
Jumpk51:
sample.Close
source.Close
Jumpknx51:
Next


nextsample:
'*****************************************************************************************
'***** Strata6 (Landzoner i restzone)**********************************************
'*****************************************************************************************

UPPER = 274

DoCmd.Echo yes, "Tæller antal zoner i strata6  (landzoner i restzone)"
For T = 1 To UPPER
sq2 = "SELECT DISTINCTROW sT.zon FROM strata6 AS ST WHERE ST.GRP=" &
Str$(T) & ";"

Set MYDB = DBEngine.Workspaces(0).Databases(0)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)

nrec = 0
  source.MoveFirst
Do
  nrec = nrec + 1
  source.MoveNext
  If source.EOF Then GoTo Jumpk62
Loop
Jumpk62:
source.Close

numrec(T) = nrec
```

```
txtant = "*** grp = " & Str$(T) & " har " & Str$(nrec) & " zoner ***"
DoCmd.Echo yes, txtant

Next

'*************************************************************

DoCmd.Echo yes, "definerer datasæt"
For T = 1 To UPPER
kriterie = "sampleZoner.HKID = " & Str$(T)
nx = DCount("[ID]", "sampleZoner", kriterie)
If nx = 0 Then GoTo Jumpknx61
sq1 = "SELECT DISTINCTROW sampleZoner.*, sampleZoner.HKID AS GRP FROM
sampleZoner WHERE " & kriterie & ";"
sq2 = "SELECT DISTINCTROW sT.zon FROM strata6 AS ST WHERE ST.GRP=" &
Str$(T) & ";"


Set sample = MYDB.OpenRecordset(sq1, DB_OPEN_DYNASET)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)

txtant = "6*** sampler fra strata 6, grp = " & Str$(T) & ", opdaterer Altstr6" & CStr(k)
DoCmd.Echo yes, txtant
sample.MoveFirst
Do
TRY = 0
GoTo nextline61
nyzonk61:
TRY = TRY + 1
nextline61:
If TRY > 50 Then GoTo errordest1
Randomize
source.MoveFirst
source.Move Int(numrec(T) * Rnd)
zone = source.zon
  If zone = sample.Bozon Then GoTo nyzonk61
  If zone = sample.Dest1 Then GoTo nyzonk61
  If zone = sample.altstr11 Then GoTo nyzonk61
  If zone = sample.altstr12 Then GoTo nyzonk61
  If zone = sample.altstr13 Then GoTo nyzonk61
  If zone = sample.altstr14 Then GoTo nyzonk61
  If zone = sample.altstr41 Then GoTo nyzonk61
  If zone = sample.altstr42 Then GoTo nyzonk61
  If zone = sample.altstr51 Then GoTo nyzonk61
 sample.Edit
  sample.altstr61 = zone
  sample.Update
  sample.MoveNext
 If sample.EOF Then GoTo Jumpk61
```

```
Loop
Jumpk61:
sample.Close
source.Close
Jumpknx61:
Next



Exit Function

errordest1:
n = Err
If n = 3011 Or n = 3078 Then
 DoCmd.Echo yes, "Kunne ikke finde objekt"
 Resume Next
End If
message = Error$
If TRY > 5 Then message = "Der er ikke 4 forskellige zoner"
response = MsgBox(message, 2)
If response = 3 Then 'abort
   Reset
   Exit Function
   End If
If response = 4 Then Resume Else Resume Next


End Function
```

6.1.4 Sampledest2

```
Function Sampledest2()

On Error GoTo errorDest2

lower = 1
UPPER = 175

Set MYDB = DBEngine.Workspaces(0).Databases(0)

GoTo notmakezones
'Call deltable("sampleZoner")


sq = ""
DoCmd.Echo yes, "TDMDATA + strata1 -> sampleZoner *** opretter tabel"
sq = sq & "SELECT DISTINCTROW s.ID, s.strata AS strata1, s.chosen, s.nr AS nr1, T.Hzon, T.HKOM, T.Landsdel, T.hkomagrid, s.Altzon, A.KOM AS AKOM, A.komagrid AS AKomAgrID, T.Dest1, T.Dest2, T.KOM1, T.KOM2, t.Hzon AS altstr11, s.Altzon AS altstr12, 1000000 AS altstr21, 1000000 AS altstr22, 1000000 AS altstr31, 1000000 AS altstr41, csng(0) as rand1, csng(0) as rand2 "
sq = sq & "INTO sampleZoner "
sq = sq & "FROM ((ch2firsttrip AS s LEFT JOIN TDMDATA AS T ON s.ID = T.ID) LEFT JOIN overLczonTilKom AS O ON s.Altzon = O.LCzon) LEFT JOIN AggrKom AS A ON O.KOM = A.KOM "
sq = sq & "WHERE ((s.chosen < 2) and (len(s.chain)=3 or len(s.chain)=5)) "
sq = sq & "ORDER BY T.HKOM;"
Call runsql(sq)

'*******************************************************************************
'1***** Strata2a (boKommune)****************************************************

notmakezones:

DoCmd.Echo yes, "Tæller antal zoner i strata2 (bokommune)"
For T = 1 To UPPER
sq2 = "SELECT DISTINCTROW sT.zon FROM Bstrata2 AS ST WHERE ST.KomagrID=" & Str$(T) & ";"

Set MYDB = DBEngine.Workspaces(0).Databases(0)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)

nrec = 0
  source.MoveFirst
Do
  nrec = nrec + 1
  source.MoveNext
  If source.EOF Then GoTo Jumpb2
```

```
Loop
Jumpb2:
source.Close

numrec(T) = nrec

txtant = "*** grp = " & Str$(T) & " har " & Str$(nrec) & " zoner ***"
DoCmd.Echo yes, txtant

Next

'************************************************************

DoCmd.Echo yes, "definerer datasæt"
For T = 1 To UPPER
kriterie = "sampleZoner.HKomagrID = " & Str$(T)
nx = DCount("[ID]", "sampleZoner", kriterie)
If nx = 0 Then GoTo JumpBnx11
sq1 = "SELECT DISTINCTROW sampleZoner.*, sampleZoner.HKomagrID AS GRP FROM
sampleZoner WHERE " & kriterie & ";"
sq2 = "SELECT DISTINCTROW sT.zon FROM Bstrata2 AS ST WHERE ST.KomagrID=" &
Str$(T) & ";"

Set sample = MYDB.OpenRecordset(sq1, DB_OPEN_DYNASET)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)

txtant = "1*** sampler fra strata 2 (byzoner i bokommuner), grp = " & Str$(T) & ",
opdaterer Altstr21"
DoCmd.Echo yes, txtant
sample.MoveFirst
Do
nyzonb1:
Randomize
source.MoveFirst
source.Move Int(numrec(T) * Rnd)
zone = source.zon
 If zone = sample.Hzon Then GoTo nyzonb1
sample.Edit
 sample.altstr21 = zone
 sample.Update
 sample.MoveNext
If sample.EOF Then GoTo Jumpb1
Loop
Jumpb1:
sample.Close
source.Close
JumpBnx11:
Next

'*******************************************************************************
```

```
'*****************************************************************************
'2***** Strata2b (ArbKommune)*****************************************************

UPPER = 175

DoCmd.Echo yes, "Tæller antal zoner i strata2  (Arbkommune)"
For T = 1 To UPPER
sq2 = "SELECT DISTINCTROW sT.zon FROM Bstrata2 AS ST WHERE ST.KomagrID=" &
Str$(T) & ";"

Set MYDB = DBEngine.Workspaces(0).Databases(0)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)

nrec = 0
 source.MoveFirst
Do
 nrec = nrec + 1
 source.MoveNext
 If source.EOF Then GoTo Jumpb22
Loop
Jumpb22:
source.Close

numrec(T) = nrec

txtant = "***  grp = " & Str$(T) & " har " & Str$(nrec) & " zoner  ***"
DoCmd.Echo yes, txtant

Next

'**************************************************************

DoCmd.Echo yes, "definerer datasæt"
For T = 1 To UPPER
kriterie = "sampleZoner.AKomagrID = " & Str$(T)
nx = DCount("[ID]", "sampleZoner", kriterie)
If nx = 0 Then GoTo JumpB2nx11
sq1 = "SELECT DISTINCTROW sampleZoner.*, sampleZoner.AKomagrid AS GRP FROM
sampleZoner WHERE " & kriterie & ";"
sq2 = "SELECT DISTINCTROW sT.zon FROM Bstrata2 AS ST WHERE ST.KomagrID=" &
Str$(T) & ";"

Set sample = MYDB.OpenRecordset(sq1, DB_OPEN_DYNASET)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)

txtant = "1***  sampler fra strata 2 (byzoner i Arbkommuner), grp = " & Str$(T) & ",
opdaterer Altstr22"
DoCmd.Echo yes, txtant
sample.MoveFirst
Do
```

```
nyzonb21:
Randomize
source.MoveFirst
source.Move Int(numrec(T) * Rnd)
zone = source.zon
  If zone = sample.Hzon Then GoTo nyzonb21
sample.Edit
  sample.altstr22 = zone
  sample.Update
  sample.MoveNext
If sample.EOF Then GoTo Jumpb21
Loop
Jumpb21:
sample.Close
source.Close
JumpB2nx11:
Next

'*********************************************************************************
'******Byzoner ej i arbkom (agr) ej i bokom (agr)*******************************
'*********************************************************************************


UPPER = 3
DoCmd.Echo yes, "Tæller antal zoner i strata2  (bokommune)"
For T = 1 To UPPER
sq2 = "SELECT DISTINCTROW sT.zon FROM Bstrata2 AS ST WHERE ST.Landsdel=" &
Str$(T) & ";"

Set MYDB = DBEngine.Workspaces(0).Databases(0)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)

nrec = 0
  source.MoveFirst
Do
  nrec = nrec + 1
  source.MoveNext
  If source.EOF Then GoTo Jumpb32
Loop
Jumpb32:
source.Close

numrec(T) = nrec

txtant = "***  grp = " & Str$(T) & " har " & Str$(nrec) & " zoner  ***"
DoCmd.Echo yes, txtant

Next

'**************************************************************
```

```
DoCmd.Echo yes, "definerer datasæt"
For T = 1 To UPPER
kriterie = "sampleZoner.landsdel = " & Str$(T)
nx = DCount("[ID]", "sampleZoner", kriterie)
If nx = 0 Then GoTo JumpB3nx11
sq1 = "SELECT DISTINCTROW sampleZoner.* FROM sampleZoner WHERE " & kriterie &
";"
sq2 = "SELECT DISTINCTROW sT.* FROM Bstrata2 AS ST WHERE ST.Landsdel=" &
Str$(T) & ";"

Set sample = MYDB.OpenRecordset(sq1, DB_OPEN_DYNASET)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)

txtant = "1***  sampler fra strata 2 (byzoner i bokommuner), grp = " & Str$(T) & ",
opdaterer Altstr31"
DoCmd.Echo yes, txtant
sample.MoveFirst
Do
nyzonb31:
Randomize
source.MoveFirst
source.Move Int(numrec(T) * Rnd)
zone = source.zon
  If zone = sample.Hzon Then GoTo nyzonb31
  If zone = sample.altzon Then GoTo nyzonb31
  If source.komagrid = sample.hkomagrid Then GoTo nyzonb31
  If source.komagrid = sample.AKomAgrID Then GoTo nyzonb31
sample.Edit
  sample.altstr31 = zone
  sample.Update
  sample.MoveNext
If sample.EOF Then GoTo Jumpb31
Loop
Jumpb31:
sample.Close
source.Close
JumpB3nx11:
Next

'*****************************************************************************************
'*******Landzoner ej i arbkom (agr) ej i bokom (agr)*******************************
'*****************************************************************************************


UPPER = 3
DoCmd.Echo yes, "Tæller antal zoner i strata4  (bokommune)"
For T = 1 To UPPER
sq2 = "SELECT DISTINCTROW sT.zon FROM Bstrata4 AS ST WHERE ST.Landsdel=" &
Str$(T) & ";"
```

```
Set MYDB = DBEngine.Workspaces(0).Databases(0)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)

nrec = 0
  source.MoveFirst
Do
  nrec = nrec + 1
  source.MoveNext
  If source.EOF Then GoTo Jumpb42
Loop
Jumpb42:
source.Close

numrec(T) = nrec

txtant = "***  grp = " & Str$(T) & " har " & Str$(nrec) & " zoner  ***"
DoCmd.Echo yes, txtant

Next

'**************************************************************

DoCmd.Echo yes, "definerer datasæt"
For T = 1 To UPPER
kriterie = "sampleZoner.landsdel = " & Str$(T)
nx = DCount("[ID]", "sampleZoner", kriterie)
If nx = 0 Then GoTo JumpB4nx11
sq1 = "SELECT DISTINCTROW sampleZoner.* FROM sampleZoner WHERE " & kriterie &
";"
sq2 = "SELECT DISTINCTROW sT.* FROM Bstrata4 AS ST WHERE ST.Landsdel=" &
Str$(T) & ";"

Set sample = MYDB.OpenRecordset(sq1, DB_OPEN_DYNASET)
Set source = MYDB.OpenRecordset(sq2, DB_OPEN_DYNASET)

txtant = "1***  sampler fra strata 2 (byzoner i bokommuner), grp = " & Str$(T) & ",
opdaterer Altstr31"
DoCmd.Echo yes, txtant
sample.MoveFirst
Do
nyzonb41:
Randomize
source.MoveFirst
source.Move Int(numrec(T) * Rnd)
zone = source.zon
  If zone = sample.Hzon Then GoTo nyzonb41
  If zone = sample.altzon Then GoTo nyzonb41
sample.Edit
  sample.altstr41 = zone
```

```
 sample.rand1 = Rnd(1)
 sample.rand2 = Rnd(2)
 sample.Update
 sample.MoveNext
If sample.EOF Then GoTo Jumpb41
Loop
Jumpb41:
sample.Close
source.Close
JumpB4nx11:
Next


'*********************************************************************************
'*********************************************************************************
'*********************************************************************************



DoCmd.Echo yes, "random arb eller bozone"
sq = sq & "UPDATE DISTINCTROW sampleZoner SET sampleZoner.altstr11 = [altstr12] "
sq = sq & "WHERE ((sampleZoner.rand1>0.5));"
Call runsql(sq)

DoCmd.Echo yes, "random arb eller bokommune"
sq = sq & "UPDATE DISTINCTROW sampleZoner SET sampleZoner.altstr21 = [altstr22] "
sq = sq & "WHERE ((sampleZoner.rand2>0.5));"
Call runsql(sq)

'*********************************************************************************

Exit Function

errorDest2:
n = Err
If n = 3011 Or n = 3078 Then
 DoCmd.Echo yes, "Kunne ikke finde objekt"
 Resume Next
End If
response = MsgBox(Error$, 2)
If response = 3 Then 'abort
   Reset
   Exit Function
   End If
If response = 4 Then Resume Else Resume Next

End Function
```

6.1.5 Forecast

```
Call AlogitRunning

If ARUN = 1 Then GoTo slut
If READY = 0 Then GoTo slut

Select Case STEP
  Case 1
    STEP = 2
    x1 = Shell("C:\25127\FORECAST\LsTopHP.BAT", 1)
    GoTo slut
  Case 2
    STEP = 3
    READY = 0
    DoCmd.RunMacro "F_DoTopYesnoHP"
    Call AccessFinished
    GoTo slut
  Case 3
    STEP = 4
    x1 = Shell("C:\25127\FORECAST\KmCarHP.bat", 1)
    GoTo slut
  Case 4
    STEP = 5
    x1 = Shell("C:\25127\FORECAST\PrTopHP.BAT", 1)
    GoTo slut
  Case 5
    STEP = 6
    READY = 0
    DoCmd.RunMacro "F_ImportKmTranspMiddel"
    Call AccessFinished
    GoTo slut
  Case 6
    STEP = 7
    READY = 0
    DoCmd.RunMacro "F_ImportTurtypeProbsCar012"
    Call AccessFinished
    GoTo slut
  Case 7
    STEP = 8
    READY = 0
    DoCmd.RunMacro "F_ForecastDelModelKmToYesNoDat"
    Call AccessFinished
    GoTo slut
  Case 8
    STEP = 9
    x1 = Shell("C:\25127\FORECAST\PROBCAR.BAT", 1)
    GoTo slut
  Case 9
    STEP = 10
```

```
        READY = 0
        DoCmd.RunMacro "F_Importcarprob"
        Call AccessFinished
        GoTo slut
    Case 10
        STEP = 11
        x1 = Shell("C:\25127\FORECAST\LsTopIP.BAT", 1)
        GoTo slut
    Case 11
        STEP = 12
        READY = 0
        DoCmd.RunMacro "F_DoTopYesnoIP"
        Call AccessFinished
        GoTo slut
    Case 12
        STEP = 13
        x1 = Shell("C:\25127\FORECAST\PrTopIP.BAT", 1)
        GoTo slut
    Case 13
        STEP = 14
        READY = 0
        DoCmd.RunMacro "F_ImportTurtypeProbsCar012"
        Call AccessFinished
        GoTo slut
    Case 14
        STEP = 15
        x1 = Shell("C:\25127\FORECAST\KmCarIP.bat", 1)
        GoTo slut
    Case 15
        STEP = 16
        READY = 0
        DoCmd.RunMacro "F_ImportKmTranspMiddel"
        Call AccessFinished
        GoTo slut
    Case 16
        STEP = 17
        READY = 0
        DoCmd.RunMacro "F_Forecasttable"
        Call AccessFinished
        GoTo slut
    Case Else
        Msg = "End"
End Select

slut:

Exit Function
```

6.2 Descriptive statistics

Table 6-2. Describtive statistics for chain choice model

Variable name	Number of valid obs.	Minimum	Maximum	Mean	Std. Deviation
FAMTYP	13545	1	3	1.85825	0.508481
PERSONS	13545	1	8	2.310816	1.138305
CHILDS1	13545	0	3	0.157918	0.43367
CHILDS2	13545	0	3	0.154965	0.426504
CHILDS3	13545	0	3	0.154522	0.424235
CHILDS4	13545	0	4	0.122259	0.353818
HAFTBIL	13545	1	4	1.816464	1.218572
KOREKORT	13544	0	1	0.811282	0.391299
KK	13545	0	7	1.685493	0.788006
GAABUS	13545	1	9	1.718642	1.01868
GAASTA	13545	1	9	3.805685	1.375428
KON	13545	1	2	1.496493	0.500006
SKOLUD	13545	1	9	2.239646	0.466392
STIL	13545	1	98	17.73562	12.01008
ARBTID	13545	0	99	25.45205	21.15968
PARTNER	13545	0	1	0.699668	0.458419
ALDERP	13545	0	99	31.77121	23.79809
STILP	13545	0	99	14.40605	13.50209
ARBTIDP	13545	0	99	20.12661	21.94278
KOREP	13545	0	9	0.639498	0.512158
FORML2	0				
FORML3	0				
FORML4	0				
MODE	10933	1	8	2.08799	1.064313
H38ZON	13545	1	38	20.2601	11.52474
ANTAL0T17	0				
ANTAL18T64	0				
ANTALO64	0				
BUSHVDG	13545	0	3131	378.6882	558.5556
BUSL_R	0				
BUSS_N	0				
HTOGLEVEL	13545	0	6712	625.9011	1635.576
HURBAN	13545	11	70	40.45028	18.87642
T	0				
TARXDAYWGT	13545	0.16004	7.51302	1.0003	0.488469
INKNTI	13545	1	21	5.021779	2.575479
CHAIN	13545	0	12414	721.2076	2500.748
DAG	13545	1	7	3.983389	1.999248

continued..

Table 6-3. Describtive statistics for chain choice model, continued

Variable name	Number of valid obs.	Minimum	Maximum	Mean	Std. Deviation
LOGSCH12	13545	5.05752	18.927	10.52149	1.664331
LOGSC12X	13545	2.39955	13.0705	5.765192	0.999335
LOGSCH13	13545	-0.24236	6.78756	3.22245	0.918195
LOGSCH14	13545	2.92546	10.9404	6.852148	1.227517
LOGSCX13	13545	-0.76254	11.1536	5.755142	1.683609
LOGSC214	13545	0.33965	5.15085	2.380954	0.589192
LOGSC314	13545	2.74733	14.571	7.70653	1.427617
LOGSC123	13545	1.45665	13.716	5.763292	1.297668
LOGSC23X	13545	-0.80962	10.111	4.405863	1.191738
LOGSC24X	13545	2.22225	17.9327	9.218446	1.771788
LOGSC134	13545	-0.21215	6.27593	2.70214	0.868789