

HAYDÉE SVAB

Evolução dos padrões de deslocamento na Região Metropolitana de São Paulo: a necessidade de uma análise de gênero

São Paulo
2016

HAYDÉE SVAB

Evolução dos padrões de deslocamento na Região Metropolitana de São Paulo: a necessidade de uma análise de gênero

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Engenharia.

Área de Concentração:
Planejamento de Transportes

Orientador: Prof. Dr. Orlando Strambi

São Paulo
2016

Este trabalho é dedicado a todas as pessoas que acreditam que o planejamento de transportes só será bem-sucedido se houver conexão com as dimensões sociais e urbanas que influencia e pelas quais é influenciado.

Agradecimentos

Um trabalho de pesquisa sempre é fruto direta ou indiretamente da colaboração de muitas pessoas. Agradecê-las nominalmente é um risco por ser possível deixar indevidamente alguém de fora. Assim, deixo registrado um sincero agradecimento a todas e todos que contribuíram para esta dissertação tornar-se realidade: minha família de nascença e também aquela que escolhemos e vamos compondo ao longo da vida.

Ao Diego Rabatone Oliveira, agradeço pela parceria incondicional, pela tranquilidade e amor fundamentais nos momentos de tensão e pelas horas a fio dispendidas ombro a ombro sobre a mesa de trabalho.

Ao Orlando Strambi, agradeço pela dedicação e orientação, indispensáveis para o andamento deste trabalho. Às professoras Maria Eugenia Boscov, Cintia Borges Margi, Liedi Bernucci e Silvia P. C. Casa Nova, pelo apoio e pelo exemplo que são para mim como profissionais e como mulheres.

Aos docentes Andreina Nigirello, Heloisa Buarque de Almeida e Luiz Paulo Fávero, exemplos de honestidade e coerência acadêmica, cujas contribuições foram valiosas e com quem gostaria de estabelecer parcerias intelectuais futuramente.

À Angela Buscema, à Enaege Sant'ana e à Patrícia Santana por me auxiliarem a encontrar meus caminhos, não só burocráticos, dentro da Escola Politécnica da USP.

À Cia. do Metropolitano de São Paulo, por meio de seus funcionários Maria Cecilia M. A. Oliveira, Maria Cecilia M. Laiza, Luiz Claudio Sposito, José de França Bueno, Nelson Lucio Nunes e, especialmente, Emilia Mayumi Hiroi, pela paciência, prestatividade e abertura de portas (e dados).

Ao Felipe Dias, à Glaucia Pereira e ao Dionísio M. S. Gutierrez sou grata pela amizade, pela disposição em ajudar em momentos cruciais - sem vocês a realização deste trabalho seria impossível.

À Bianca Alves, ao Fábio Lofrano, à Patricia Cornils e à Daniela Rozados pelas conversas encorajadoras sobre o tema e pela admiração intelectual que me provocam.

Ao Giuliano Salcas Olguin, um dia chefe, hoje, amigo e sócio, que guiou minhas primeiras frases produzidas dentro do que hoje se chama formalmente de pesquisa.

Finalmente, agradeço a todos que escreveram, empenharam-se na aprovação e lutam por uma justa implementação da Lei de Acesso à Informação que, mesmo com falhas, materializa o conceito de informação pública e dado aberto não apenas para mim, mas para a sociedade - sem ela o acesso aos dados desta pesquisa não estaria garantido.

“No atlas do seu império, ó Grande Khan, devem constar tanto a grande Fedora de pedra quanto as pequenas Fedoras das esferas de vidro. Não porque sejam igualmente reais, mas porque são todas supostas. Uma reúne o que é considerado necessário, mas ainda não o é; as outras, o que se imagina possível e um minuto mais tarde deixa de sê-lo.”

(Calvino, Ítalo; In: Cidades Invisíveis)

Resumo

O presente trabalho aborda questões relativas a análises de comportamento da demanda por transportes jogando luz sobre as questões de gênero. A revisão de literatura explora os conceitos de gênero, mobilidade e acessibilidade, bem como busca apresentar resultados de estudos que analisaram o que ocorre nas intersecções destes conceitos. Para investigar a sobreposição dessas áreas na Região Metropolitana de São Paulo foram utilizados os dados das Pesquisas Origem Destino de 1977, 1987, 1997 e 2007 da Cia. do Metropolitano de São Paulo. Esta dissertação¹ estuda longitudinalmente os padrões de mobilidade e investiga a hipótese de que existem diferenças entre os padrões de mobilidade feminino e masculino. Foi construído um banco de dados unificado, base necessária para a elaboração de estatísticas descritivas, análises de conglomerados e regressões logísticas que pretendem identificar grupos de comportamentos semelhantes, traçar os perfis dos grupos formados e elencar variáveis dependentes e explicativas relevantes para concepção de modelos de análise desagregada de demanda de transportes. Os grupos formados na análise de conglomerados majoritariamente coincidiram com os anos das pesquisas, comprovando a necessidade de se fazer análises longitudinais. Por fim, foi realizada outra segmentação articulando as variáveis sexo e situação familiar, buscando melhor caracterizar o gênero como categoria de análise. Para esta última segmentação, foram realizadas regressões *quasi-poisson* considerando as variáveis relevantes indicadas nas etapas anteriores. A partir desse conjunto de regressões, assim como algumas estatísticas descritivas já davam pistas, foram encontradas diferenças no número total de viagens de homens e mulheres condicionados a diferentes papéis familiares. A compreensão de que grupos de diferentes perfis sócio-econômicos têm diferentes mobilidades e acessibilidades pode beneficiar a área de planejamento de transportes a desenvolver políticas voltadas a algum segmento de interesse e, assim, fazer uso mais eficientes dos recursos públicos.

Palavras-chaves: gênero; mobilidade; planejamento de transportes; pesquisa origem destino; Região Metropolitana de São Paulo.

¹ Esta dissertação está licenciada com uma Licença Creative Commons - Atribuição CompartilhaIgual 4.0 Internacional (CC BY-SA 4.0 <<http://creativecommons.org/licenses/by-sa/4.0/>>) e foi baseada nos materiais disponíveis em <<https://github.com/hsvab/mestrado-usp-ODs>>, <<https://github.com/hsvab/mestrado-usp-algoritmos>> e <<https://github.com/hsvab/mestrado-usp-texto>>.

Abstract

The study focus on the analysis of travel behavior from the perspective of gender. The literature review explores the concepts of gender, mobility and accessibility, searching for results in the intersection of these concepts. The study uses data from the Origin-Destination surveys for the Metropolitan Area of São Paulo, conducted in 1997, 1987, 1997 and 2007. Longitudinal analyses using different approaches explore the hypothesis that there are systematic differences in travel and activity patterns between genders. A unified database was constructed, allowing the longitudinal analysis of descriptive statistics, to perform cluster analysis and to estimate logistic regressions. The results allowed the identification of groups of similar behavior, to look at their composition and to select relevant dependent and explanatory variables for disaggregate modelling of travel demand. Groups resulting from cluster analysis have coincided with each of the four data collection periods, confirming the need to conduct longitudinal analyses. To better represent the concept of gender, a combination of the variables sex and position in family structure were used to define segments for analysis with quasi-poisson regression, using variables selected from previous stages of analysis. The results indicate significant differences in the daily number of trips between men and women, conditioned by their family roles. The understanding of these differences can improve transportation planning and policy making.

Key-words: gender, mobility, transportation planning, origin-destination survey, Metropolitan Area of São Paulo

Listas de ilustrações

Figura 1 – Mapa dos municípios que compõem a região metropolitana em 2014, divididos por sub-regiões	27
Figura 2 – Prisma Espaço Tempo	40
Figura 3 – Quantidade de espaço viário requerido para transportar 60 pessoas por ônibus, bicicleta e carro.	41
Figura 4 – Frances Willard aprendendo a andar de bicicleta	43
Figura 5 – Simulação de probabilidade de uso do carro para viagens não-trabalho em função do número de crianças na família, na Alemanha	47
Figura 6 – Esquema indicativo da formação dos IDs	66
Figura 7 – Box plot da variável “TOT_FAM”, por ano	90
Figura 8 – Box plot da variável “TOT_PESS”, por ano	99
Figura 9 – Etapas para a realização de análise de conglomerados (<i>clusters</i>)	131
Figura 10 – Dendrograma resultante da análise de conglomerados hierárquico, para o conjunto de atributos de viagens relativas às famílias	136
Figura 11 – Dendrograma resultante da análise de conglomerados hierárquico, para o conjunto de atributos de viagens relativas às pessoas	137
Figura 12 – Pertinência a <i>cluster</i> de famílias, pessoas e viagens	143
Figura 13 – Síntese dos resultados da análise de conglomerados para cada ano . . .	153

Lista de quadros

Quadro 1 – Critérios para Avaliação de Dados Secundários	54
Quadro 2 – Dados para atribuição de renda familiar	61
Quadro 3 – <i>Layout</i> do Banco de Dados Unificado	68
Quadro 4 – <i>Dummies</i> de período	77
Quadro 5 – Variáveis de contagem da utilização dos modos	77
Quadro 6 – <i>Dummies</i> de motivos de viagem	78
Quadro 7 – <i>Dummies</i> de faixa etária	79
Quadro 8 – Categorias da variável Faixa de Renda Individual	80
Quadro 9 – Categorias da variável Faixa de Renda Familiar	81
Quadro 10 – <i>Dummies</i> de presença de crianças na família	83
Quadro 11 – Descrição dos grupos utilizados nas regressões <i>quasi-poisson</i>	159

Lista de gráficos

Gráfico 1 – Percentual de indivíduos que fazem parte da PEA, por sexo, no Brasil, entre 1950 e 2010	35
Gráfico 2 – Jornadas Médias para o Mercado de Trabalho e para Reprodução Social, por sexo, raça/cor e região geográfica, no Brasil, em 2003	36
Gráfico 3 – Distribuição da variável “FE_DOM”, por ano	88
Gráfico 4 – Distribuição da variável “REN_FAM”, por ano	93
Gráfico 5 – Distribuição da variável “FAIXA_REN_FAM”, por ano	94
Gráfico 6 – Proporção de famílias com pessoa responsável do sexo feminino e do sexo masculino, segundo posse de automóveis, por ano	97
Gráfico 7 – Proporção de famílias com presença de dependentes, por ano	100
Gráfico 8 – Distribuição da variável “IDADE” de respondentes, por ano e por sexo	101
Gráfico 9 – Distribuição da variável “SIT_FAM”, por ano e por sexo	103
Gráfico 10 – Distribuição da variável “GRAU_INSTR”, por ano e por sexo	105
Gráfico 11 – Distribuição da variável “FAIXA_REN_IND”, por ano	107
Gráfico 12 – Distribuição da variável “TOT_VIAG” por ano e por sexo	112
Gráfico 13 – Proporção das viagens do sexo feminino e do sexo masculino, segundo o primeiro modo da viagem, por ano	115
Gráfico 14 – Proporção das viagens do sexo feminino e do sexo masculino, de acordo com o segundo modo da viagem, por ano	116
Gráfico 15 – Proporção das viagens do sexo feminino e do sexo masculino, segundo o modo da viagem (agregado), por ano	118
Gráfico 16 – Proporção das viagens do sexo feminino e do sexo masculino, segundo o motivo no destino, por ano	120
Gráfico 17 – Durações médias de viagem por ano e por sexo, segundo os modos (agregados)	122
Gráfico 18 – Durações médias de viagem por ano e por sexo, segundo os modos coletivos	123
Gráfico 19 – Durações médias de viagem por ano e por sexo, segundo o motivo da viagem	124
Gráfico 20 – Distâncias médias de viagem por ano e por sexo, segundo os modos (agregados)	127
Gráfico 21 – Distâncias médias de viagem por ano e por sexo, segundo os modos coletivos	128
Gráfico 22 – Distâncias médias de viagem por ano e por sexo, segundo o motivo da viagem	130

Gráfico 23 – Avaliação do número de <i>clusters</i> para o conjunto de atributos de viagens relativas às famílias	136
Gráfico 24 – Avaliação do número de <i>clusters</i> para o conjunto de atributos de viagens relativas às pessoas	137

Lista de tabelas

Tabela 1 – Consumo dinâmico de espaço por modo e renda na RMSP	42
Tabela 2 – Consumo energético teórico dos modos de transporte em lotação plena	49
Tabela 3 – Características Amostrais das Pesquisas OD	58
Tabela 4 – Características Gerais das Pesquisas OD - amostras	59
Tabela 5 – Características Gerais segundo as Pesquisas OD - expansões	60
Tabela 6 – Quantidade de viagens e de domicílios, por ano, antes e depois da preparação do BDU	76
Tabela 7 – Estatísticas da variável “ANO”	85
Tabela 8 – Estatísticas da variável “DIA_SEM”	86
Tabela 9 – Estatísticas da variável “F_DOM”	86
Tabela 10 – Estatísticas da variável “F_FAM”	87
Tabela 11 – Estatísticas da variável “F_PESS”	87
Tabela 12 – Estatísticas da variável “FE_DOM”	88
Tabela 13 – Estatísticas da variável “TIPO_DOM”	89
Tabela 14 – Estatísticas da variável “TOT_FAM”	89
Tabela 15 – Estatísticas da variável “COND_MORA”	90
Tabela 16 – Deflatores utilizados para correção dos valores monetários para outubro/2007	92
Tabela 17 – Estatísticas da variável “REN_FAM”	93
Tabela 18 – Evolução da frota de automóveis na RMSP segundo as Pesquisas OD	95
Tabela 19 – Venda interna de veículos no Brasil entre 1960 e 2009	95
Tabela 20 – Estatísticas da variável “QT_AUTO”	95
Tabela 21 – Proporção das famílias segundo posse de automóveis e de acordo com sexo da pessoa responsável	97
Tabela 22 – Estatísticas das variáveis “QT_MOTO” e “QT_BICI”	98
Tabela 23 – Estatísticas da variável “TOT_PESS”	99
Tabela 24 – Frequências absoluta e relativa da variável SEXO, por ano	101
Tabela 25 – Evolução das taxas de fecundidade no Brasil, de 1970 a 2010	102
Tabela 26 – Frequências da variável TRABALHA, por ano e sexo	104
Tabela 27 – Frequências da variável ESTUDA, por ano e sexo	104
Tabela 28 – Estatísticas da variável “REN_IND”	106
Tabela 29 – Estatísticas da variável “TOT_VIAG”, por ano e por sexo	109
Tabela 30 – Estatísticas da variável “TOT_VIAG”, por ano e por sexo, considerando apenas quem fez viagem	110
Tabela 31 – Média da variável “TOT_VIAG”, por ano e por sexo, com expansão	111
Tabela 32 – Estatísticas da variável “FAM_VIAG_TOT”, por ano	113

Tabela 33 – Frequência relativa das variáveis “MODO1” e “MODO2”, por ano	113
Tabela 34 – Frequência relativa da variável “TIPO_VIAG”, por ano	116
Tabela 35 – Frequência relativa da variável “MOTIVO_DEST”, por ano	119
Tabela 36 – Estatísticas da variável “DURACAO”, por ano	121
Tabela 37 – Estatísticas da variável “DIST_VIAG”, por ano	126
Tabela 38 – Medidas de dissimilaridade utilizadas em análise de <i>cluster</i>	135
Tabela 39 – Resultado do agrupamento de 4 <i>clusters</i> , por atributos de viagens de família - método Ward	138
Tabela 40 – Resultado do agrupamento de 4 <i>clusters</i> , por atributos de viagens de família - método centroide	138
Tabela 41 – Resultado do agrupamento de 3 <i>clusters</i> , por atributos de viagens de família - método Ward	139
Tabela 42 – Resultado do agrupamento de 3 <i>clusters</i> , por atributos de viagens de família - método centroide	139
Tabela 43 – Resultado do agrupamento de 3 <i>clusters</i> , por atributos de viagens de pessoas - métodos Ward e centroide	139
Tabela 44 – Ordenamento das variáveis pelas maiores diferenças percentuais - para agrupamento FAM_CLUSTER_WARD4	141
Tabela 45 – Ordenamento das variáveis pelas maiores diferenças percentuais - para agrupamento FAM_CLUSTER_CENTROIDE4	141
Tabela 46 – Síntese dos resultados da análise de conglomerados para cada ano	144
Tabela 47 – Ordenamento das variáveis pelas maiores diferenças percentuais - para agrupamento FAM_CLUSTER_CENTROIDE4 apenas de 1977	146
Tabela 48 – Ordenamento das variáveis pelas maiores diferenças percentuais - para agrupamento FAM_CLUSTER_CENTROIDE4 apenas de 1987	148
Tabela 49 – Ordenamento das variáveis pelas maiores diferenças percentuais - para agrupamento FAM_CLUSTER_CENTROIDE4 apenas de 1997	149
Tabela 50 – Ordenamento das variáveis pelas maiores diferenças percentuais - para agrupamento FAM_CLUSTER_CENTROIDE4 apenas de 2007	152
Tabela 51 – Resultado da regressão logística multinomial, com categoria de referência Ano=4 (correspondente a 2007) para os atributos de viagem da família	155
Tabela 52 – Resultado da regressão logística multinomial com categoria de referência Ano=4 (correspondente a 2007) para os atributos de viagem da pessoa	158
Tabela 53 – Valores de média e variância de TOT_VIAG para cada um dos grupos	160
Tabela 54 – Resultado das regressões <i>quasi-poisson</i> para os grupos de mulheres com situações familiares ‘pessoa responsável’, ‘cônjugue’ e ‘filha’	163
Tabela 55 – Resultado das regressões <i>quasi-poisson</i> para os grupos de homens com situações familiares ‘pessoa responsável’, ‘cônjugue’ e ‘filho’	164

Lista de abreviaturas e siglas

BDU	Banco de Dados Unificado
CET	Companhia de Engenharia de Tráfego
FGV	Fundação Getúlio Vargas
IBGE	Instituto Brasileiro de Geografia e Estatística
IPVA	Imposto sobre a Propriedade de Veículos Automotores
Metrô-SP	Companhia do Metropolitano de São Paulo
OD-1977	Pesquisa Origem Destino do Metrô-SP de 1977
OD-1987	Pesquisa Origem Destino do Metrô-SP de 1987
OD-1997	Pesquisa Origem Destino do Metrô-SP de 1997
OD-2007	Pesquisa Origem Destino do Metrô-SP de 2007
ONU	Organização das Nações Unidas
PIB	Produto Interno Bruto
RMSP	Região Metropolitana de São Paulo
UCOD	Unidade de Correspondência entre Zonas Origem-Destino

Sumário

1	INTRODUÇÃO	23
1.1	Objeto e Objetivos	24
1.2	Estrutura do trabalho	25
2	REVISÃO DE LITERATURA	27
2.1	Gênero	28
2.2	Mobilidade Urbana	37
2.3	Intersecções e Sobreposições	43
3	CONSIDERAÇÕES METODOLÓGICAS	53
4	AS PESQUISAS ORIGEM DESTINO DA RMSP	57
4.1	Periodicidade e Objetivos da Pesquisa Origem Destino	57
4.2	Descrição Sucinta	57
4.3	Dados Coletados	58
4.4	Conceitos Adotados	62
4.5	Limitações	63
5	TRATAMENTO DOS DADOS	65
5.1	Preparação da base de dados	65
5.2	Critérios de Validação	75
5.3	Formulação de Novas Variáveis	76
5.3.1	Variáveis relativas às viagens	76
5.3.2	Variáveis relativas às pessoas	79
5.3.3	Variáveis relativas às famílias	81
6	ANÁLISES, RESULTADOS E DISCUSSÕES	85
6.1	Estatísticas Descritivas	85
6.2	Análises Preliminares	108
6.3	Análise de conglomerados para identificação de grupos	131
6.4	Regressão logística para investigar a formação de grupos	154
6.5	Regressões <i>quasi-poisson</i> articulando sexo e situação familiar	159
7	CONSIDERAÇÕES FINAIS	171
	Referências	173

ANEXO A – CORRESPONDÊNCIA ENTRE ZONAS DAS PESQUISAS ORIGEM DESTINO POR MEIO DAS UNIDADES DE CORRESPONDÊNCIA ENTRE ZONAS (UCOD)	187
ANEXO B – MAPAS DE ZONAS DAS PESQUISAS ORIGEM DESTINO 1977 E DE SUBZONAS DE 1987 E 1997	191
ANEXO C – LAYOUTS DOS BANCOS DE DADOS DAS PESQUISAS ORIGEM-DESTINO DO METRÔ-SP (1977, 1987, 1997 E 2007)	195
ANEXO D – SÍNTESE DE RESULTADOS PARA 4 CONGLOMERADOS	213
ANEXO E – SÍNTESE DE RESULTADOS PARA 4 CONGLOMERADOS - SEGMENTAÇÃO POR ANO	215
ANEXO F – RESULTADOS DAS REGRESSÕES LOGÍSTICAS PARA ATRIBUTOS DE VIAGENS DA PESSOA - TODAS VARIÁVEIS	219
ANEXO G – RESULTADOS DAS REGRESSÕES QUASI-POISSON	223
ANEXO H – ROTINAS DE PREPARAÇÃO DAS BASES DE DADOS DAS PESQUISAS ORIGEM-DESTINO DO METRÔ-SP (1977, 1987, 1997 E 2007)	227

1 Introdução

A propagação de megacidades¹, especialmente nos países em desenvolvimento, tem sido acompanhada de fenômenos como aumento da urbanização e da motorização. No Brasil, São Paulo é a megacidade de destaque e objeto deste estudo. O que pouco tem acompanhado o crescimento urbano têm sido as políticas de planejamento urbano e de transportes, de forma integrada, o que tem incorrido em problemas como congestionamentos (KINGHAM; DICKINSON; COPSEY, 2001; STEG, 2005; METZ, 2012), piora das características ambientais (TERTOOLEN; KREVELD; VERSTRATEN, 1998; RICHARDSON, 2005; BANISTER, 2011), e aprofundamento das desigualdades (HODGE, 1995; AHMED; LU; YE, 2008; LEWIS, 2011). A iniquidade não se dá apenas no acesso aos recursos e às oportunidades, mas também na distribuição dos espaços públicos (ALVA, 1997) e, em específico, no espaço destinado à circulação nas cidades (VASCONCELLOS, 2012).

Estudos constatam (VASCONCELLOS, 2001; RUEDA et al., 2007) que os automóveis particulares ocupam maior quantidade de espaço de circulação para transportar a mesma quantidade de pessoas do que outros modos de transporte (motorizados ou não). Tendo em vista esse cenário, recentemente, foi aprovada no Brasil a Política Nacional de Mobilidade Urbana (BRASIL, 2012) que implicitamente indica nos seus “princípios, diretrizes e objetivos” que não é mais aceitável a priorização do automóvel particular no meio urbano, já que se deve promover a “equidade no uso do espaço público de circulação, vias e logradouros”.

A questão de gênero nos transportes tem atraído atenção crescente da comunidade científica (ROSENBLUM, 1978; HANSON; JOHNSTON, 1985; ROSENBLUM, 2006; CRESSWELL; UTENG, 2008; HANSON, 2010). Pesquisadores(as) começaram a examinar os padrões de mobilidade com o recorte de gênero considerando que há acesso desigual a recursos materiais e imateriais (HOWE; O'CONNOR, 1982; HANSON, 1995a; SILVEY; ELMHIRST, 2003; RAJU, 2005) que levam a diferenças nos padrões de atividades e de viagem (FAGNANI, 1983; MCLAFFERTY; PRESTON, 1991; JOHNSTON-ANUMONWO, 1992; ROOT; SCHINTLER, 1999; SCHWANEN; DIJST; DIELEMAN, 2002; LEE; MCDONALD, 2003; MCGUCKIN; ZMUD; NAKAMOTO, 2005; CRANE, 2007; VASCONCELLOS, 2012), em particular na escolha de modo e no uso do automóvel particular (FOX, 1983; HJORTHOL, 2000; POLK, 2003; BEST; LANZENDORF, 2005).

¹ “Megacidade” é um termo, cunhado em 1990 pela ONU, para designar cidades com mais de dez milhões de habitantes. Segundo dados da Divisão de População da ONU, em 2014 existiam 33 megacidades no planeta e São Paulo é a sétima megacidade no ranking. Fonte: <<http://esa.un.org/unpd/ppp/>> Acesso em 30 de outubro de 2014. Segundo Freitag (2007), São Paulo é também uma “megalópole”, isto é, uma megacidade (município com mais de 10 milhões de habitantes) que sofreu um crescimento muito acelerado nas três ou quatro últimas décadas do século XX.

Com uma maior participação de mulheres no mercado de trabalho a diferença entre os rendimentos de homens e mulheres vem (lentamente) diminuindo, o que levaria a crer que o padrão de viagens das mulheres passasse a se assemelhar aos dos homens, pois teriam mais recursos financeiros para dispor de um automóvel. Pela flexibilidade de horário e de trajeto que o carro pode oferecer, seria esperado um aumento no uso do carro pelas mulheres devido à pressão exercida por uma dupla jornada (trabalho formal e doméstico) ainda mais se houver presença de criança na família. Entretanto, segundo estudo de Best e Lanzendorf (2005) na Alemanha, a presença de criança na família gera o seguinte efeito: a maternidade reduz a probabilidade de uso do carro para mulheres enquanto a paternidade a aumenta para homens. Resultados como esse, aparentemente contraintuitivo, demonstram a necessidade de olhar mais atento e que considere complexidades sociais na análise do comportamento das demandas por transportes.

Esta dissertação, portanto, aborda diretamente questões relativas a análises de comportamento da demanda por transportes, jogando luz sobre as questões de gênero e suas articulações, principalmente com políticas de transporte e, em menor medida, com políticas de planejamento urbano. Entender melhor o comportamento da demanda pode colaborar na formulação de políticas de incentivo à troca de modos de transporte (de menos para mais sustentáveis) e à redução da necessidade de viajar.

1.1 Objeto e Objetivos

O objeto desta dissertação de mestrado é a relação entre o gênero e os deslocamentos de indivíduos. Assim, por objetivo geral tem-se investigar como o padrão de viagens de indivíduos é afetado pela categoria de análise gênero, no período de 1977 a 2007 na Região Metropolitana de São Paulo (RMSP). Toma-se por hipótese a ser explorada que pessoas com identidades de gênero masculina e feminina tem se deslocado de forma diferente ao longo do tempo.

Como objetivos específicos estão: (i) analisar padrões de viagens, por sexo, das principais variáveis de viagem de cada *cross-section*²; (ii) comparar os padrões encontrados e verificar se existe alguma tendência ao longo do tempo; (iii) elaborar hipóteses sobre as diferenças entre mulheres e homens do ponto de vista da demanda de transportes; (iv) analisar criticamente hipóteses elencadas e análises de dados realizadas, com base na teoria subjacente.

² Dados em *cross-section* são dados em seções transversais, ou seja, revelam características por meio das variáveis para um dado momento no tempo.

1.2 Estrutura do trabalho

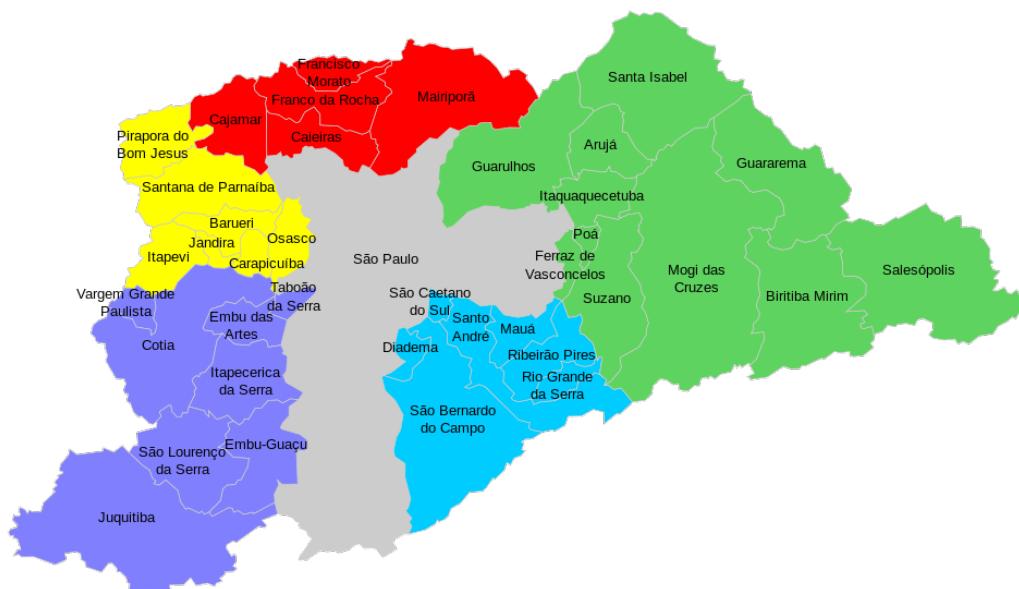
No primeiro capítulo é feita uma breve introdução ao tema, assim como são apresentados objetivos e justificativa deste trabalho. No segundo capítulo é apresentada a revisão de literatura no que tange aos principais conceitos que são explorados, como gênero e mobilidade. Na sequência, são apresentadas algumas reflexões acerca das intersecções e sobreposições desses aspectos. No terceiro capítulo são feitas considerações metodológicas sobre análises de conglomerados, regressões logísticas e *quasi-poisson*, métodos usados no desenvolvimento desta dissertação. No quarto capítulo são apresentadas brevemente as Pesquisas Origem Destino da RMSP. O quinto capítulo contém as principais descrições, considerações e conceitos relativos às bases de dados utilizadas. No sexto capítulo encontram-se as análises realizadas e resultados obtidos: estatísticas descritivas, evolução das frequências absolutas e relativas de variáveis relevantes, análises de conglomerados para formação de grupos, regressão logística para investigar melhor a formação dos grupos e regressões *quasi-poisson* para explicar o número total de viagens da pessoa, segmentando grupos por sexo e situação familiar. Concluindo, são apresentadas algumas considerações finais e apontados caminhos que podem ser seguidos em estudos posteriores.

2 Revisão de Literatura

“Todo enunciado - desde a breve réplica (monolexemática) até o romance ou o tratado científico - comporta um começo absoluto e um fim absoluto: antes de seu início, há os enunciados dos outros, depois de seu fim, há os enunciados-respostas dos outros [...]. O locutor termina seu enunciado para passar a palavra ao outro ou para dar lugar à compreensão responsiva ativa do outro. O enunciado não é uma unidade convencional, mas uma unidade real, estritamente delimitada pela alternância dos sujeitos falantes” (Bakhtin)

Este capítulo tem por objetivo clarificar conceitos considerando a evolução e as intersecções entre as concepções utilizadas, bem como dar um panorama geral de como as questões de gênero e de mobilidade vêm sendo tratadas sob a perspectiva do planejamento de transportes. Buscou-se, sempre que possível, apresentar aspectos ligados à realidade brasileira e quiçá paulistana, pois o escopo espacial de análise do presente trabalho é a Região Metropolitana de São Paulo (RMSP) (ver Figura 1), área coberta pela Pesquisas Origem e Destino (Pesquisas OD) do Metrô-SP.

Figura 1 – Mapa dos municípios que compõem a região metropolitana em 2014, divididos por sub-regiões



Fonte: Mapa elaborado por Marcos Elias Oliveira Júnior, segundo a Lei 1.139/2011 ([SÃO PAULO \(ESTADO\), 2011](#)). Disponível em: <http://pt.wikipedia.org/wiki/Regi%C3%A3o_Metropolitana_de_S%C3%A3o_Paulo#mediaviewer/File:Mapa-RMSP-subregions.svg> Acesso em 10 de novembro de 2014

2.1 Gênero

Ao nascer, uma das primeiras atividades do ser humano é comunicar-se, o que inclui nominar para si o mundo que o cerca. Esse processo não se dá de maneira solitária, a nominação advém de uma interação social que visa compartilhar signos afim de efetivar a comunicação. Por isso, este capítulo visa estabelecer o que se deseja exprimir através de palavras-chave deste trabalho (gênero e mobilidade), considerando a visão de Bakthin, cujo trabalho, segundo [Stella \(2005\)](#), indicava ser necessário

não somente a palavra, mas também a linguagem em geral, ser concebida e tratada de uma outra forma, levando-se em conta sua história, sua historicidade, ou seja, especialmente a linguagem em uso. Isso significa que, no pensamento bakthiniano, a palavra reposiciona-se em relação às concepções tradicionais, passando a ser encarada como um elemento concreto de feitura lógica.

Há um senso comum que confunde e funde, não por acaso, os conceitos de sexo e gênero, muito embora sejam distintos - distinção esta encontrada em maior ou menor grau de acordo com o idioma. Em inglês a palavra *sex* tem sentido mais limitado, ligado à anatomia, e a palavra *gender* tem sentido mais amplo, ligado à construção cultural da identidade. Em francês, a palavra *séxe* e, em alemão, a palavra *Geschlecht*, designam tanto diferenças físicas como psicológicas, sociais e culturais ([FRAISSE, 2001](#)). [Moraes \(1998\)](#) reporta que, em francês, frequentemente utiliza-se *rapports sociaux de séxe* ao invés de *gendre* para se designar *gênero*. Comparado com o termo inglês *gender*, “a palavra gênero, em português, é um substantivo masculino que designa uma classe que se divide em outras, que são chamadas espécies”, definição então retirada do Novo Dicionário Aurélio por [Moraes \(1998, p.101\)](#). Já hoje, em 2014, o Dicionário [Aurélio \(2014\)](#) comporta entre suas definições, aquela em que *gênero* pode ser entendido como o “conjunto de propriedades atribuídas social e culturalmente em relação ao sexo dos indivíduos”. Porém, entre as definições mais gerais apresentadas pelo Dicionário [Michaelis \(2011\)](#), *gênero* é definido da seguinte forma:

s.m. (lat **generu*, por *genus*) 1 Grupo de seres que têm iguais caracteres essenciais. 2 *Lóg.* A classe que tem mais extensão e portanto menor compreensão que a espécie. 3 *Biol.* Grupo morfológico intermediário entre a família e a espécie. 4 *Gram.* Flexão pela qual se exprime o sexo real ou imaginário dos seres. 5 *Gram.* Forma do adjetivo ou pronome com relação ao gênero dos nomes a que se refere. 6 Agrupamento de indivíduos que possuem caracteres comuns. 7 Espécie, casta, raça, variedade, sorte, categoria, estilo etc. 8 Qualidade, espécie, modo.

Percebe-se que o termo gênero designa um conceito em construção e consolidação, não apenas no Brasil, sendo necessário defini-lo sempre que o utilizarmos como denominação de categoria de análise ([MORAES, 1998](#)). Para isso, será feito um breve apanhado do

surgimento e trajetória da palavra *gender* ou *gênero* nas pesquisas acadêmicas, inclusive no Brasil, bem como a evolução do conceito.

O conceito fundido e identitário de sexo e gênero, como se fossem sinônimos, pertence a uma visão binária de mundo que define as mulheres mais próximas da natureza, do trabalho reprodutivo, da passividade e do irracional e, em oposição, define os homens mas próximos à cultura, ao trabalho produtivo, à ação e à racionalidade (HARAWAY, 2004). Estudiosas feministas, rejeitando o determinismo bio-sexual para a situação social das mulheres, precisavam desmontar a naturalização das diferenças entre homens e mulheres que vinculava suas relações sociais, políticas e econômicas a seu aparelho reprodutor (PISCITELLI, 2009). Para Haraway (2004, p.218), as feministas lutaram “para remover as mulheres da categoria da natureza e colocá-las na cultura como sujeitos sociais na história, construídas e auto-construtoras”. Dessa forma, a evolução do conceito de gênero mescla-se à história do feminismo.

A chamada “primeira onda feminista” ocorreu entre o final do século XIX e o início do século XX nos países hoje considerados desenvolvidos da Europa e da América do Norte. A principal bandeira reivindicava direitos iguais, compondo uma ideia de que deveria haver uma igualdade entre os sexos. Em decorrência dessa primeira movimentação, em diversos países, as mulheres conquistaram alguns direitos equivalentes aos dos homens, como o voto. Essa conquista do voto como um direito político caracterizou o movimento sufragista, que não pode ser confundido com o movimento feminista, embora seja parte dele. A Finlândia foi o primeiro país a garantir direito a votar e ser votado(a) igualmente a mulheres e homens, em 1906, quando ainda era um Principado do Império Russo (RAY, 1918). Na Inglaterra, em 1865, John Stuart Mill apresenta ao Parlamento um projeto de lei dando o voto às mulheres, que não foi aprovado. Somente em 1928, o voto feminino é autorizado nas mesmas condições às dos homens (NELSON, 2002). Nos Estados Unidos, em 1920, foi aprovada a 19^a Emenda¹, que proibia o estabelecimento de qualquer restrição ao voto (estadual e federal) baseada no sexo do(a) votante.

Na década de 1930, Mead, uma antropóloga estadunidense, problematiza a fixitude dos conceitos *feminilidade* e *masculinidade* a partir de uma pesquisa comparativa entre três sociedades tribais na Nova Guiné (MEAD, 2000). A pesquisadora conclui não haver um temperamento inato, universal que tenha origem biológica, ligada ao aparelho reprodutor. Ela observa que traços de caráter são aprendidos em sociedade, podendo, portanto, ser modificados e até desaprendidos. Ela deixa legado teórico que suporta a ideia de que existe uma construção cultural da diferença sexual.

¹ Fonte: <<http://www.archives.gov/historical-docs/document.html?doc=13&title.raw=19th+Amendment+to+the+U.S.+Constitution:+Women%27s+Right+to+Vote>> Acesso em 02 de novembro de 2014.

Em 1949, a filósofa francesa Beauvoir lança a obra *O Segundo Sexo*, considerado precursor da “segunda onda feminista”(PISCITELLI, 2009). Ainda que Beauvoir (1967) não cite o conceito de “papel social” ou mesmo “papel sexual”, ela enfatiza logo de início que o “ser uma mulher” é uma construção social:

nenhum destino biológico, psíquico, econômico define a forma que a fêmea humana assume no seio da sociedade; é o conjunto da civilização que elabora esse produto intermediário entre o macho e o castrado que qualificam de feminino. (BEAUVOIR, 1967, p.09)

Em sua obra, Beauvoir (1967) tem por foco questionar a dominação masculina, sem deixar de questionar também a eficácia do movimento feminista, forjado até então no combate a essa dominação. Ela julgava ser possível esse combate ser bem sucedido se fossem combatidos elementos como: forma com que mulheres eram educadas; instituição de casamentos opressores; maternidade compulsória; vigência de um duplo padrão de moralidade sexual que permitia maior liberdade sexual somente aos homens; e falta de trabalhos dignos e bem remunerados que possibilitassem independência econômica às mulheres.

Quase que concomitantemente, nos Estados Unidos, nasce um novo par de categorias de estudos, o sexo-gênero (FRAISSE, 2001; STOLKE, 2004; HARAWAY, 2004). A distinção entre as características biológicas e as características sociais torna-se mais difundida, ou seja, na academia e na sociedade passa a ser considerada a noção de que posturas sociais de identidade masculina ou feminina não estabelecem relação biunívoca com o sexo anatômico. A denominação dessa construção cultural pela palavra gênero ocorre em 1958, na Califórnia, quando foi empreendida uma pesquisa acerca da identidade de gênero no *California Gender Identity Center*. Os resultados foram apresentados pelo psicanalista Robert Stoller em 1963, no Congresso de Psicanálise de Estocolmo. Essa mesma pesquisa embasou a elaboração do primeiro volume de *Sex and Gender* de Stoller (1984). Essa obra expôs o quanto a relação sexo e gênero não é automática, nem estrita, discorrendo ainda sobre casos em que a anatomia da genitália não seria compatível com a identidade masculina ou feminina da pessoa. Assim, Stoller formula um conceito de gênero ligado à cultura, enquanto o conceito de sexo permanece ligado à morfologia corporal.

Em 1970 e 1980, o debate sobre esse par de categorias (sexo-gênero) toma espaço na comunidade acadêmica estadunidense. A antropóloga Rubin (1975) introduz a categoria gênero no debate sobre opressões sociais sofridas pelas mulheres por meio do seu ensaio *The Traffic in Women: Notes on the 'Political Economy' of Sex*. Nessa obra, Rubin faz uma análise marxista sobreposta ao sistema sexo-gênero da qual depreende que no sistema de trocas capitalista, os homens estabelecem-se como vendedores e as mulheres são estabelecidas como mercadorias para serem trocadas. Rubin dialoga com Lévi-Strauss que aponta ser o casamento o dispositivo mais importante de aliança entre as famílias,

inexistente se não fosse pelo *tabu do incesto* (LÉVI-STRAUSS, 2010). Para Rubin (1975 apud PISCITELLI, 2009) esse tabu é precedido por outro, o da *homossexualidade*. Isso porque, mediante a divisão sexual do trabalho² e ao tomar como a menor unidade de sobrevivência econômica a família, tem-se necessariamente um homem e uma mulher, numa relação heterossexual de dependência mútua. Rubin discute também o trabalho doméstico, dando visibilidade a um trabalho que muitas vezes viabiliza o sustento do trabalhador (geralmente homem) sem que seja remunerada (a mulher). Por fim, ela consegue articular teoricamente gênero e sexualidade de forma que o conceito de gênero constituído até então não reside apenas em identificação com um determinado sexo, mas pressupõe que o desejo sexual seja por indivíduo do sexo oposto.

A distinção entre sexo e gênero foi extremamente útil às feministas acadêmicas, pois sinalizava um lastro teórico para embasar os estudos sobre a condição da mulher, muitas vezes inferiorizada por sua condição biológica inerente. Com isso, o questionamento à lógica binária de interpretação do mundo passou a ser menos frequente e incisivo (HARAWAY, 2004, p.218) e, porque não, superada em alguma medida. Conforme pode-se ver no trabalho de Rubin, o conceito de gênero foi além de separar dimensões culturais e biológicas de mulheres e homens. Cada vez mais o conceito de gênero passa a significar também a superação da leitura binária de mundo que só permite feminilidade ou masculinidade. Para Heilborn (1992, p.41):

A categoria de gênero não deve ser acionada como um substituto de referência para homem ou mulher. Seu uso designa, ou deveria fazê-lo, a dimensão inerente de uma escolha cultural e de conteúdo relacional. Por outro lado, traz embutida a articulação desse código, que se apropria da articulação da diferença sexual tematizando-a em masculino e feminino, com outros níveis de significação dos universos.

Se a primeira onda do feminismo reivindicou direitos iguais, a segunda onda avançou e lutou pelo exercício igual dos direitos. Na primeira onda buscava-se provar que as diferenças entre o feminino e o masculino eram de origem social e não biológica. A afirmação não é abandonada na segunda onda, mas passa-se a buscar as origens de tais diferenças sócio-culturais. Nessa construção, segundo Piscitelli (2009, p.133-134):

A categoria “mulher” foi desenvolvida pelo feminismo da segunda onda em leituras segundo as quais a opressão das mulheres está além de questões de classe e raça, atingindo todas mulheres, inclusive as mulheres das classes altas e brancas. [...] O reconhecimento político das mulheres como coletividade ancora-se na ideia de que o que une as mulheres ultrapassa em muito as diferenças entre elas. Isso criava uma “identidade” entre elas.

² A expressão *divisão sexual do trabalho* foi inicialmente utilizada por etnólogos para se referir à repartição das atividades entre homens e mulheres nas sociedades que estudavam (KERGOAT, 2004). Esta autora afirma ainda que “a divisão sexual do trabalho é aquela decorrente das relações sociais de sexo”, o que será explorado mais adiante neste capítulo.

Se essa uniformização entre as mulheres foi útil para forjar uma união na conquista por direitos, em meados da década de 1970 e início dos anos 1980, já era questionada. Feministas negras e mulheres de países subdesenvolvidos (FURTADO, 2009) cada vez menos identificavam-se com o arcabouço teórico hegemônico e homogêneo apresentado por feministas dos países do “norte rico”, inclusive por Rubin. Assim, a “terceira onda feminista” desdobra-se em feminismos diversos. Afinal, as mulheres negras contam com trajetória histórica diferente das mulheres brancas, grande parte das vezes tendo a escravidão e suas consequências como parte determinante da vida de sua ancestralidade (HOOKS, 1990; CRENSHAW, 2002). No caso de países subdesenvolvidos, como o Brasil, não cabe comparar *ipsis literis* a trajetória das mulheres (mesmo brancas) brasileiras com as europeias. A título de ilustração, o estudo de Pinto (2004) apresenta como as mulheres brasileiras são vistas como mais maternais, com vocação para a domesticidade e muito mais “racializadas” do que as portuguesas.

Oferecendo alguma resposta a essas demandas por interseccionalidade³ em 1986, a historiadora pós-estruturalista Joan Scott publica seu artigo *Gender: A Useful Category of Historical Analysis* em que faz uma leitura crítica da utilização do termo *gênero* como categoria de análise e relaciona necessariamente esta categoria a outras como classe e raça, pois demonstra ser o gênero necessariamente imbricado a relações hierarquizadas de poder:

a oposição binária e o processo social das relações de gênero tornam-se, ambos, partes do sentido do próprio poder. Colocar em questão ou mudar um aspecto ameaça o sistema por inteiro. Se as significações de gênero e de poder se constroem reciprocamente, como é que as coisas mudam? [...] o gênero tem que ser redefinido e reestruturado em conjunção com uma visão de igualdade política e social que inclui não só o sexo, mas também, a classe e a raça. (SCOTT, 1986, p.1073,1075)

É então necessário olhar a construção das identidades de gênero à luz das relações de poder e olhar brevemente como se deu a evolução dos direitos, especialmente na sociedade brasileira. As mulheres no Brasil escravocrata dispunham de uma grande imobilidade geográfica e mesmo as mulheres das classes dominantes raramente saíam às ruas e, quando o faziam, nunca estavam desacompanhadas (SAFFIOTI, 2013). Mulheres e homens de então desfrutavam de maneira assimétrica do direito de ir e vir.

Na campo dos direitos políticos, o movimento sufragista das brasileiras não teve tanta capilaridade nem foi um movimento de massas como nos Estados Unidos, Inglaterra ou Rússia. Ele teve início na década de 1910, quando o Partido Republicano Feminino é fundado no Rio de Janeiro com o objetivo de instaurar o debate acerca do voto feminino⁴.

³ Interseccionalidade ou abordagem interseccional, segundo Crenshaw (2002, p.177) “trata especificamente da forma pela qual o racismo, o patriarcalismo, a opressão de classe e outros sistemas discriminatórios criam desigualdades básicas que estruturam as posições relativas de mulheres, raças, etnias, classes e outras”.

⁴ Bertha Lutz, filha do cientista Adolfo Lutz, licenciou-se em Ciências Naturais na Sorbonne de Paris e, ao retornar ao Brasil, funda a Federação Brasileira pelo Progresso Feminino, em 1919, que leva

A igualdade de condições de voto entre homens e mulheres se concretiza em 1932, pelo Decreto nº 21.076 que autoriza o voto a qualquer cidadã ou cidadão com idade superior a 21 anos. A eleição de 1933 foi a primeira em que mulheres puderam participar do pleito, votando e sendo votadas, como Carlota Pereira Queiroz, a primeira deputada brasileira, que participou da Assembleia Nacional Constituinte entre 1934 e 1935 (TABAK, 1989).

Embora o direito ao voto tenha sido emblemático, a ideia de desfrutar de *direitos iguais* na sociedade, mulheres e homens, tratava também de outros direitos como o acesso à educação e poder ter posse de bens - por muito tempo, de acordo com a lei, só homens podiam ser proprietários de casas, por exemplo (PISCITELLI, 2009). Subjacente a esses questionamentos das mulheres tecia-se o conceito de “papel social”, bastante difundido a partir da década de 1930. Para Piscitelli (2009, p.127), a teoria dos papéis sociais buscava:

compreender os fatores que influenciam o comportamento humano. A ideia é que os indivíduos ocupam posições na sociedade, desempenhando papéis de filho, de estudante, de avô. [...] A ideia de posições ocupadas no desempenho dos papéis faz referência a categorias de pessoas que são reconhecidas coletivamente. Um dos atributos que podem servir de base para a definição dessas categorias é a idade. [...] Outro desses atributos pode ser o sexo. Nesse caso, homens e mulheres desempenham papéis culturalmente construídos: os papéis sexuais.

Essa busca por um leque de direitos não foi um movimento só das mulheres, mas um movimento de luta por cidadania⁵. Sob o ponto de vista de gênero e cidadania, Brito (2001) relembra que o conceito clássico de cidadania, o grego, excluía mulheres e escravos. Ela pontua que ao longo da história as identidades de homens e mulheres foram construídas pressupondo uma dicotomia entre o âmbito público e o privado. Blay (2001) relata que até os anos 1960/1970 era um fator negativo para a mulher participar da vida pública. A partir de 1970, com o movimento feminista passa a haver críticas e questionamentos quanto à natureza, à separação e à natural atribuição dessas esfera a um determinado sexo. Assim, elabora-se uma perspectiva de análise a partir do gênero, e não do sexo biológico, pois o conceito de gênero também compreende as dimensões social e política do termo.

adiante a luta pelo sufrágio feminino (PINSKY; PEDRO, 2003). A primeira cidade a autorizar o voto feminino em eleições foi Mossoró (RN), em 1928. Em nível nacional, Getúlio Vargas autoriza em 1931 o voto feminino apenas às mulheres solteiras, viúvas com renda própria ou casadas com a autorização do marido.

⁵ A cidadania, para Carvalho (2002), é entendida como o exercício pleno de três direitos: direitos civis, direitos sociais e direitos políticos. Os civis são aqueles considerados direitos fundamentais, como o direito à vida, à liberdade, à propriedade, à igualdade perante a lei. Eles garantem a vida em sociedade e dependem da existência de uma justiça independente, eficiente, barata e acessível a todos. Os políticos se referem à participação do cidadão no governo da sociedade. Seu exercício é limitado a uma parcela da população definida por idade, por exemplo, e consiste na capacidade de fazer demonstrações políticas, de organizar partidos, de votar, de ser votado. Por fim, os sociais são aqueles que garantem a participação na riqueza coletiva e se baseia na ideia de justiça social. Incluem os direitos à educação, ao trabalho, ao salário justo, à saúde, à aposentadoria.

A partir dos anos 1990 o uso da categoria gênero tornou-se mais frequente no Brasil e cada vez mais influenciado pelas diversas escolas de psicanálise para explicar a produção e a reprodução da identidade de gênero do sujeito. A psicanalista brasileira Kehl (1998) embora não tenha como central esse debate, participa dele e em sua obra *Deslocamentos do Feminino*, ao invés de apartar sexo de gênero, assinala que gênero é um conceito que inclui a dimensão biológica do sexo, não sem somar-lhe atributos que a cultura provê. A partir de Scott é cada vez mais corrente incorporar a dimensão da política e do poder na composição do conceito de gênero, conforme explicita Moraes (1998, p.100):

A expressão relações de gênero, tal como vem sendo utilizada no campo das ciências sociais, designa, primordialmente, a perspectiva culturalista em que as categorias diferenciais de sexo não implicam o reconhecimento de uma essência masculina ou feminina, de caráter abstrato e universal, mas, diferentemente, apontam para a ordem cultural como modeladora de homens e mulheres. Em outras palavras, o que chamamos de homem e mulher não é o produto da sexualidade biológica, mas sim de relações sociais baseadas em distintas estruturas de poder.

Essas relações de poder incidem tanto sobre as relações que se desdobram no espaço público, quanto as do espaço privado. No espaço público, ou não-doméstico, tem relevância para o presente estudo o mercado de trabalho. Ao longo do tempo, a urbanização e a industrialização levaram à ampliação da classe média e ao crescimento do consumo no Brasil. As mulheres entraram também neste processo, embora a maior parte das trabalhadoras tenha sido absorvida, ao menos inicialmente, no setor de serviços e com enorme concentração nos empregos domésticos, de menor rendimento. Constata-se assim que existe aqui uma divisão sexual do trabalho (KERGOAT, 2004), que tem por características a destinação prioritária dos homens à esfera produtiva e das mulheres à esfera reprodutiva. Essa forma de divisão pauta-se em dois princípios: o da separação e o da hierarquização. O princípio da separação explica a ideia de que há “trabalhos de mulheres” e “trabalhos de homens” enquanto o princípio da hierarquização indica existir uma diferença de valoração entre o trabalho do homem (produtivo, mais valioso) e o da mulher (reprodutivo, menos valioso). Blay (2001, p.84) fala sobre a mulher brasileira:

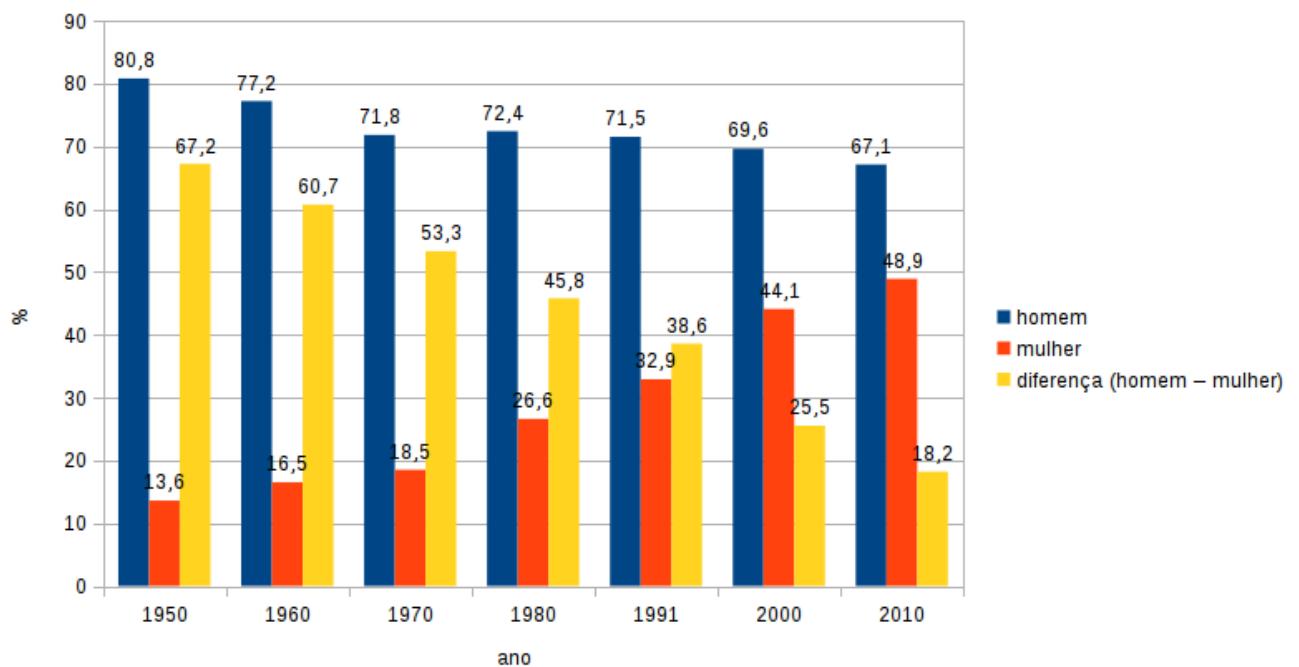
Até a década de 1960 a história, quando focalizava a mulher, atinha-se às supostas atividades femininas fundamentais, isto é, de um ser apêndice da família. A historiografia simplesmente ignorava a participação feminina no mercado de trabalho, a enorme freqüência com que sustentavam economicamente a si e aos seus.

Ao longo do século XX, observou-se aumento de mulheres na população economicamente ativa brasileira⁶ (ver Gráfico 1). O fenômeno permanece no século XXI de acordo com estudo da Fundação Perseu Abramo (2010): (i) observa-se que de 2001 a 2010

⁶ A população economicamente ativa (PEA) é obtida pela soma da população ocupada e desocupada com 16 anos ou mais de idade. “População ocupada” compreende as pessoas que, num determinado

manteve-se a preponderância feminina em ocupações que demandam de 20 a 40 horas semanais; (ii) para homens, manteve-se o predomínio histórico de jornada superior a 40 horas semanais; (iii) o ingresso das mulheres no mercado de trabalho não alterou drasticamente o papel delas na família e, portanto, nas atividades ligadas às tarefas domésticas. Isto é, apesar de muitas mulheres terem entrado no mercado de trabalho algumas décadas atrás, elas ainda são responsáveis pela maior parte do trabalho doméstico. No Gráfico 2 é possível constatar não apenas essa divisão sexual do trabalho - produtivo, no mercado, e reprodutivo, no lar - mas também que as jornadas totais que acabam ficando a cargo da mulher são maiores. Os homens acumulam uma jornada de cerca de 50 horas por semana, as mulheres, 57 horas semanais.

Gráfico 1 – Percentual de indivíduos que fazem parte da PEA, por sexo, no Brasil, entre 1950 e 2010

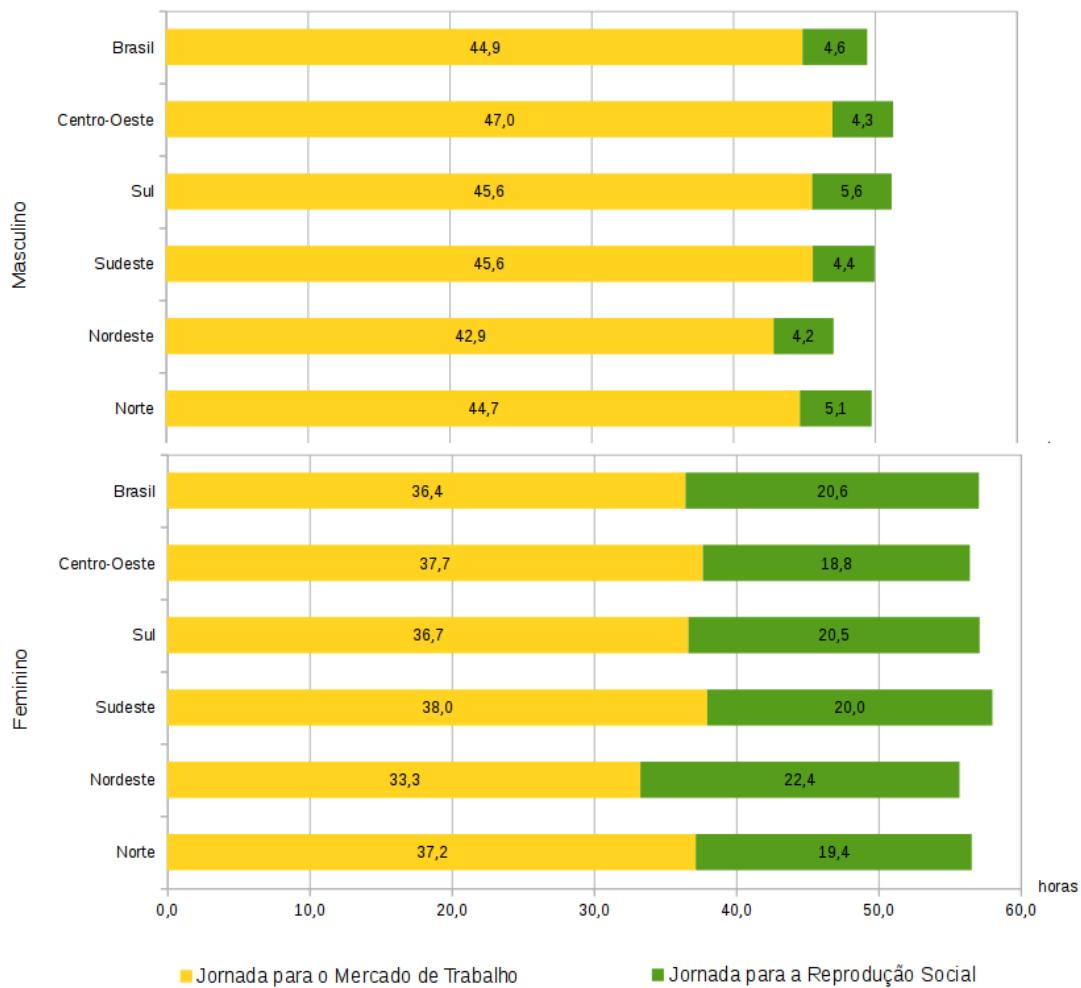


Fonte: Adaptado de (ALVES, 2013)

Esse é o ponto em que a atuação no espaço público e a no espaço privado vincula-se. A mulher passa a poder desempenhar atividades antes tidas como “masculinas”, porém sem ser desonerada de desempenhar as atividades tidas como “femininas”, pois ainda “persistem

período de referência, trabalharam ou tinham trabalho mas não trabalharam (por exemplo, pessoas em férias). “População desocupada” compreende as pessoas que não tinham trabalho, num determinado período de referência, mas estavam dispostas a trabalhar, e que, para isso, tomaram alguma providência efetiva nos últimos 30 dias (consultando pessoas, jornais, etc.). Fonte: IBGE - disponível em <<http://www.ibge.gov.br/apps/snig/v1/?loc=0,355030&cat=118,119,1,2,-2,-3&ind=87>> Acesso em 21 de novembro de 2014

Gráfico 2 – Jornadas Médias para o Mercado de Trabalho e para Reprodução Social, por sexo, raça/cor e região geográfica, no Brasil, em 2003



Fonte: PNAD (2003 apud Soares e Pinheiro, 2003)

Nota: Segundo Dear e Scott (1981), o espaço de reprodução é onde a recuperação da força de trabalho ocorre sendo a residência o local principal a ser considerado; e o espaço de produção é onde o processo de acumulação do capital ocorre, ou seja, no que se denomina mercado de trabalho (indústria, comércio e serviços no geral).

nichos onde vigora uma imagem feminina vinculada à maternidade e ao cuidado da família, à saúde da prole” (BLAY, 2001, p.94). Assim, a ampliação do leque de papéis sociais que a mulher desempenha impacta as relações de poder dentro do ambiente doméstico, dentro da família. Isso molda as necessidades, interesses, atividades e padrão de viagens dos integrantes da família, a partir das identidades de gênero constituídas, forjadas pelos comportamentos de indivíduos e da relação de poder estabelecida entre eles.

2.2 Mobilidade Urbana

A palavra mobilidade, de acordo com o Dicionário Michaelis (2011), significa “(i) propriedade do que é móvel ou do que obedece às leis do movimento; (ii) deslocamento de indivíduos, grupos ou elementos culturais no espaço social; (iii) movimento comunicado por uma força qualquer; (iv) falta de estabilidade, de firmeza ou inconstância”. Tal definição reflete toda uma gama de conceitos relacionados a movimento e/ou deslocamento, o que na área de transportes relaciona-se imediatamente a viagens. No Brasil, há cerca de 100 anos, a maior parte das viagens de pessoas era feita a pé⁷ e, quando muito, usava-se tração animal (cavalo ou boi), especialmente para cargas. Isso incorria em baixas velocidades de deslocamento e, assim, grande parte das pessoas acabavam por desenvolver suas atividades, por toda vida, nas proximidades de onde nasceram. Neste último século o cenário mudou bastante, as mais diversas tecnologias se desenvolveram, os rendimentos aumentaram, a mobilidade aumentou (METZ, 2012, p.06), e como exemplos icônicos dessa maior mobilidade figuram a utilização do carro e do avião.

O conceito de mobilidade pode englobar muitos outros e se desdobrar em uma grande diversidade de temas. Há quem o aborde ligando-o ao turismo (ENLOE, 1989; FROHLICK, 2008), enquanto outros (CHANT, 1992; SILVEY, 2000) abordam-no sob a perspectiva dos movimentos migratórios entre países ou dentro de uma mesma nação. Outro uso do termo é ligado ao intercâmbio de estudantes e pesquisadores de diferentes instituições de origem – o que dá origem à expressão *mobilidade acadêmica* (ENDERS, 1998; TREMBLAY, 2005; HOFFMAN, 2008). Ademais, há um olhar sobre a mobilidade em que as condições geodemográficas são elementos de contorno, delimitando assim as áreas da mobilidade rural e urbana. A delimitação espacial deste trabalho é a RMSP⁸; onde todos municípios englobados contam, atualmente, com áreas consideradas urbanas⁹. E o foco de análise da presente dissertação concentra-se nos deslocamentos intra-urbanos, expressos amplamente pelo conceito de mobilidade urbana.

A mobilidade urbana é um elemento fundamental para que seja possível garantir

⁷ Os primeiros carros foram montados em São Paulo pela Ford na década de 1910. Fonte: <http://www.carroantigo.com/portugues/conteudo/curio_hist_carro_brasileiro.htm> Acesso em 25 de outubro de 2014

⁸ Os 39 municípios que compõem a RMSP são agrupados em 6 regiões de acordo com Lei Complementar estadual nº 1.139, de 16 de junho de 2011. Na região central está São Paulo. Na região Sudoeste encontram-se 8 municípios, a saber, Juquitiba, São Lourenço da Serra, Embu-Guaçu, Itapecerica da Serra, Embu, Tabão da Serra, Cotia e Vargem Grande Paulista. Na Região Oeste encontram-se 7 municípios, a saber, Pirapora do Bom Jesus, Santa de Parnaíba, Barueri, Jandira, Itapevi, Carapicuíba, Osasco. Na região Norte encontram-se 5 municípios, a saber, Cajamar, Caieira, Franco da Rocha, Francisco Morato, Mairiporã. Na região Leste encontram-se 11 municípios, a saber, Santa Isabel, Arujá, Guarulhos, Itaquaquecetuba, Guararema, Poá, Suzano, Ferraz de Vasconcelos, Mogi das Cruzes, Biritiba Mirim, Salesópolis. Na região Sudeste encontram-se 7 municípios, a saber, Santo André, São Bernardo do Campo, São Caetano do Sul, Diadema, Mauá, Ribeirão Pires, Rio Grande da Serra.

⁹ Conceito de urbano adotado no censo do IBGE disponível em: <<http://www.ibge.gov.br/home/estatistica/populacao/contagem/conceitos.shtm>>

aos habitantes de uma cidade acesso aos bens que lhes oferece ([IEMA, 2010](#)). Cabe aqui diferenciar o que seja mobilidade do que seja acessibilidade. A mobilidade exprime a capacidade de se deslocar no espaço, “refere-se à habilidade de mover-se entre dois diferentes locais de atividade” ([HANSON, 1995b](#), p.04). Sobre a acessibilidade e sua ligação com a mobilidade, [Hanson](#) afirma ainda:

A acessibilidade refere-se ao número de oportunidades [...] disponíveis dentro de uma determinada distância ou tempo de viagem. [...] Conforme as distâncias entre os locais de atividades se tornam maiores [...] a acessibilidade passa a depender cada vez mais da mobilidade, particularmente daquela relacionada aos veículos particulares. ([HANSON, 1995b](#), p.04)

Dessa maneira, vinculando a mobilidade à motorização, [Hanson](#) afirma que é possível promover acessibilidade sem incrementar a mobilidade (1995, p.05), afinal, ter toda uma sorte de serviços próximos à residência daria a possibilidade de ir à padaria, ao mercado, à igreja, à escola, à livraria, etc. a pé. A autora ainda se aprofunda na questão da acessibilidade ao classificá-la como (i) de pessoas ou (ii) de lugares. Trata-se apenas de diferentes referenciais, a acessibilidade de uma pessoa indica o quanto fácil ou difícil é para ela chegar a determinado local; a acessibilidade de um lugar mostra o quanto fácil ou difícil pode ser alcançá-lo. Essas expressões “fácil” e “difícil”, entretanto, são formas muita genéricas para caracterizar o que se deseja exprimir. Problema para o qual ela apresenta como resposta uma medida de acessibilidade A_i , onde A_i é o conjunto das oportunidades O_i ponderadas pelas distâncias $d_{i,j}$ da residência da pessoa i :

$$A_i = \sum_i O_i d_{i,j}^{-b} \quad (2.1)$$

Caso considere-se ao invés de i indivíduos, i zonas, a Equação 2.1 referir-se-á à zona i de análise. Como este é um modelo bastante simplificado, indica apenas um potencial de acessibilidade e tem suas limitações, como por exemplo, desconsiderando a dimensão temporal dos deslocamentos. [Vasconcellos \(2012\)](#), por sua vez, pontua a questão temporal em sua definição de acessibilidade:

medida pela quantidade e/ou diversidade de destinos que a pessoa consegue alcançar, por certa forma de transporte, em determinado tempo. Quanto maior for esta quantidade, maior é a acessibilidade, ou seja, mais oportunidades as pessoas terão para realizar atividades desejadas ou necessárias. ([VASCONCELLOS, 2012](#), p.42)

O mesmo autor, em obra anterior, indica como forma de mensurar a acessibilidade, a soma dos tempos: (i) de deslocamento até o meio de transporte; (ii) tempo de espera, caso exista; (iii) tempos(s) dentro do(s) meio(s) de transporte; (iv) tempo de transferência entre diferentes meios de transportes, caso exista; (v) tempo após saída do meio de

transporte até atingir o destino final. Destes tempos, ele classifica os itens (i) e (v) como **microacessibilidade**, ou seja, itens que referem-se “à facilidade relativa de ter acesso aos veículos ou destinos desejados (por exemplo, condições de estacionamento ou acesso ao ponto de ônibus)” (2001, p.91). A **macroacessibilidade** é definida por ele como a:

facilidade relativa de atravessar o espaço e atingir construções e equipamentos urbanos desejados. Ela reflete a variedade de destinos que podem ser alcançados e, consequentemente, o arco de possibilidades de relações sociais, econômicas, políticas e culturais, dos habitantes do local. (VASCONCELLOS, 2001, p.91)

Isto posto, a acessibilidade pode ser entendida como a capacidade de se chegar onde se deseja e, para sua mensuração pode-se usar a distância e/ou o tempo. Uma das formas de reunir essas duas dimensões é através do prisma espaço-tempo. Na Figura 2 pode-se observar o diagrama do tempo em função da distância¹⁰ de um casal hipotético com filho pequeno. Supõe-se que ambos trabalhem das 8 às 18 horas, que seja a mulher a levar e buscar a criança na escola e que haja um carro na família. A Figura 2 descreve uma situação em que o pai fica com o carro da família. Assim, a mulher leva a criança na escola a pé¹¹ e segue para o trabalho de ônibus enquanto seu marido segue para o trabalho de carro, ambos no sentido bairro-centro¹². Na volta, ele retorna de carro diretamente para a residência e ela sai do trabalho de ônibus, apanha a criança na escola, para depois irem a pé para casa; todos no sentido centro-bairro.¹³ São acessíveis à mulher as oportunidades contidas na área de hachura vermelha. A área azul indica o diferencial de acessibilidade que o homem tem neste caso. Ou seja, a cidade possível no que diz respeito a oportunidades de trabalho, escola, lazer, etc. ao alcance dessa mulher é quase a metade daquela desse homem.

A situação narrada é hipotética, mas é ilustrativa de alguns fatores que restringem as liberdades de movimento, segundo Hagerstrand (1970 apud HANSON, 1995b):

- (i) limitações devido ao fato de que não se pode estar em dois lugares ao mesmo tempo e que certas tarefas precisam ser feitas usando um determinado modo de transporte (por algum motivo pode não haver outras opções);

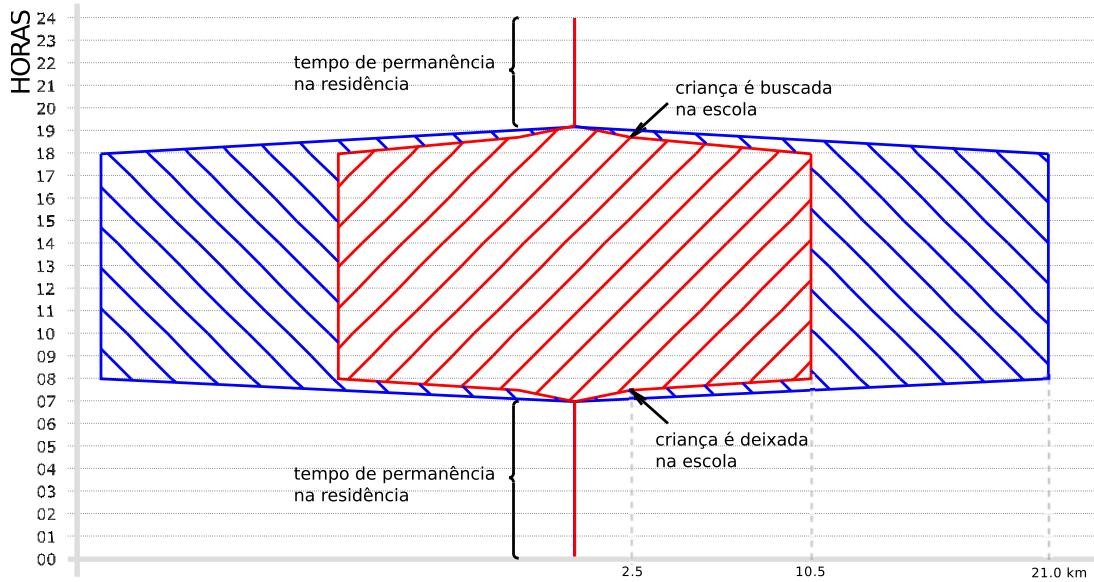
¹⁰ Para ser efetivamente um prisma, tal gráfico deveria ser em três dimensões: na base x e y representariam os deslocamentos no plano, e na altura, z, teríamos o tempo.

¹¹ Considerou-se que a velocidade média ao caminhar seja de uma velocidade média de 5km/h. Fonte: <<http://www.anpet.org.br/xxviiianpet/anais/documents/AC301.pdf>> Acesso em 03 de dezembro de 2014.

¹² Para esta simulação ilustrativa foram usadas velocidades do pico da manhã de 16km/h para ônibus e de 21km/h para carros, de acordo com dados da CET. Fonte: <<http://www.cetsp.com.br/media/228073/2007%20%20volumes%20e%20velocidades.pdf>> Acesso em 06 de dezembro de 2014.

¹³ Para esta simulação ilustrativa foram usadas velocidades do pico da tarde de 11km/h para ônibus e de 19km/h para carros, de acordo com dados da CET. Fonte: <<http://www.cetsp.com.br/media/228073/2007%20%20volumes%20e%20velocidades.pdf>> Acesso em 03 de dezembro de 2014.

Figura 2 – Prisma Espaço Tempo

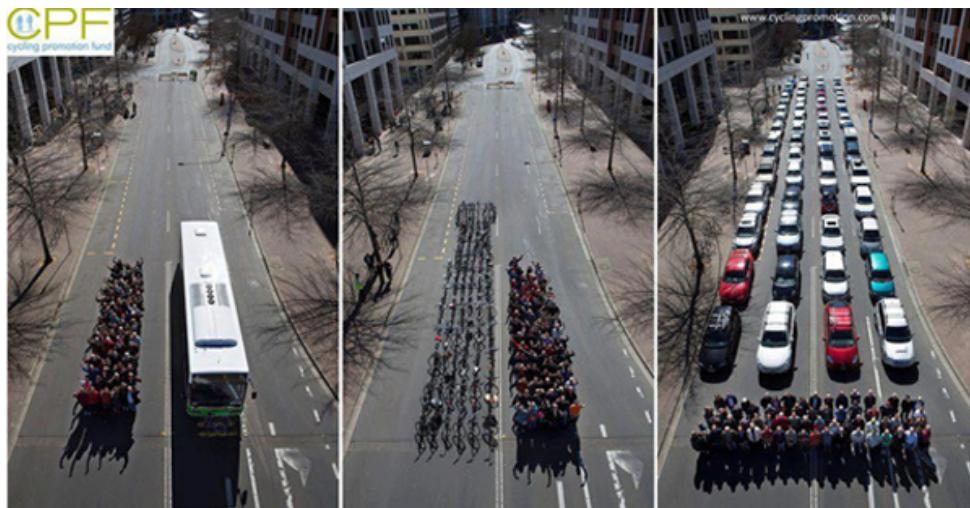


Fonte: Elaboração própria

- (ii) necessidade de encaixar os compromissos de uma pessoa com os de outra(s) pessoa(s), como por exemplo, levar filho(a)(s) à escola, acompanhar idoso(a)(s) ao médico ou almoçar com amigo(a)(s);
- (iii) restrições devido à autoridade social, política e/ou legal no acesso a algum lugar - pode haver regras explícitas (ou implícitas), por exemplo, que impeçam as pessoas de andarem à noite sozinhas num determinado local.

Geralmente, as pessoas não dispõem de várias alternativas modais ou porque não têm recursos (ou para ter um carro ou para morar em área bem servida de transporte público) ou porque se sentem inseguras utilizando algum modo específico (medo de ser assaltado(a) no carro, de ser atropelado(a) de bicicleta, de ser assediado(a) no transporte público, entre outros). Não são todos(as) que podem arranjar horários flexíveis de trabalho e/ou de estudos para conciliar as outras atividades da forma mais eficiente, por exemplo, evitando os deslocamentos nos horários de pico. Enfim, as restrições incidem diferentemente nos grupos sociais criando assimetrias no acesso às oportunidades. Essa iniquidade desdobra-se no espaço: na distribuição desigual de empregos nas cidades, na variação de preço do solo urbano, na densidade heterogênea de serviços públicos oferecidos nos diversos bairros e mesmo na ocupação do espaço de circulação (ver Figura 3).

Figura 3 – Quantidade de espaço viário requerido para transportar 60 pessoas por ônibus, bicicleta e carro.



Fonte: Foto de *Cycling Promotion Fund*, disponível em <<http://www.bhtrans.pbh.gov.br/portal/page/portal/portalpublico/Temas/ObservatorioMobilidade/FiquePorDentro/ObsMobBH%20A%20cidade%20com%20menos%20carros>> - acesso em 22 de novembro de 2014.

Em 1991, o estrato com os 20% de menor renda da população perfazia 9% das milhas viajadas nos Estados Unidos (por carro e ônibus), ao passo que o estrato com os 20% de maior renda concentrava 32% do total de milhas (CAMERON, 1994 apud HANSON, 1995b). Vasconcellos (2001) apresenta o consumo de espaço, por modo de transporte e renda da RMSP em 1987 e 1997 (ver Tabela 1). Observa-se que na RMSP os grupos de maior renda tendem a consumir mais espaço de circulação, o que levanta a questão do quão (in)justo é esse cenário, principalmente no Brasil, onde o sistema de tributação que custeia a infra-estrutura pública urbana é regressivo¹⁴.

Segundo Urry (2004 apud CRESSWELL; UTENG, 2008), tendo em vista as iniquidades urbanas de acesso, estruturadas socialmente, há cinco “mobilidades” bastante interdependentes:

- (i) viagem corpórea das pessoas por motivo de trabalho, lazer, etc.;
- (ii) movimento físico de objetos (cargas);
- (iii) viagem imaginativa a lugares por meio de imagens (fotos ou televisão);
- (iv) viagem virtual mediante uso da internet;
- (v) viagem comunicativa através de mensagens trocadas entre pessoas (cartas, mensagens de celular, telefone).

¹⁴ O Brasil conta com um sistema de tributação regressivo, ou seja, aquele em que a retirada é proporcionalmente maior das pessoas com menor capacidade de contribuir (GRECO; GODOI, 2005).

Tabela 1 – Consumo dinâmico de espaço por modo e renda na RMSP

Renda familiar mensal (1987)	Espaço dinâmico (km*m ² /dia/pessoa) (1987)	Renda familiar mensal (1997)	Espaço dinâmico (km*m ² /dia/pessoa) (1997)
0 a 240	7,6	0 a 250	9,2
241 a 480	13,4	251 a 500	14,6
481 a 900	25,1	501 a 1000	23,7
901 a 1800	42,2	1001 a 1800	36,7
1801 ou mais	74,8	de 1801 a 3600	56,2
-	-	3601 ou mais	81,9

Fonte: Adaptado de ([VASCONCELLOS, 2001](#), p.181;196)

Nota: Considerando consumo médio de $1,0m^2/pessoa$ em transporte público ($30m^2$ de área de ônibus para 30 passageiros em média) e $6,6m^2/pessoa$ em transporte privado ($10m^2$ de área de carro para 1,5 passageiros em média).

Não é possível considerar então a mobilidade do indivíduo, isolando-o do seu contexto social, econômico, político e cultural; muito pelo contrário, só é possível entendê-la se considerarmos os ambientes em que o indivíduo se anora: doméstico, familiar e social ([HANSON, 2010](#)). Portanto, acrescido do significado de urbano definido pelo IBGE, trabalha-se aqui com o conceito de mobilidade relacionado ao item (i) de [Urry \(2004 apud CRESSWELL; UTENG, 2008\)](#), mais detalhado no artigo *Gender and mobility: new approaches for informing sustainability* de [Hanson](#), que emprega o termo **mobilidade** para designar:

o movimento de pessoas de um lugar para outro lugar no decorrer da vida cotidiana [...] [sendo a] principal preocupação com as viagens pessoais que compõem a rotina diária de atividades como o trabalho (remunerado e não remunerado), lazer, socialização e compras. ([HANSON, 2010](#), p.7)

2.3 Intersecções e Sobreposições

Não apenas o movimento feminista e o embrião da concepção de gênero datam do final do século XIX; a intersecção entre gênero e mobilidade. Em 1895, Willard publicou seu livro *A Wheel whithin a Wheel* em que narra como ela, mulher, aos 53 anos, aprendeu a andar de bicicleta (ver Figura 4). Ela não escreveu um livro sobre mobilidade, nem sobre gênero. Porém, ela aborda essas questões a partir dessa sua experiência. Ela toca na questão de gênero, por exemplo, ao falar do vestuário de uma ciclista:

Se as mulheres pedalarem, ao fazê-lo elas devem vestir-se mais racionalmente do que foram acostumadas. E se elas fizerem isso, muitos preconceitos concernentes ao que elas estariam autorizadas a vestir cairão por terra. (Livre tradução de Willard, 1895, p.39)

Figura 4 – Frances Willard aprendendo a andar de bicicleta



Fonte: (WILLARD, 1895, p.56)

O que se esperava que uma mulher vestisse àquela época? Como pedalar com vestidos tão longos que sempre cobriam os pés e contavam com muitos babados, plissados, franjas e passamanarias¹⁵? Andar de bicicleta não foi para ela apenas um desafio por conta da habilidade manual requerida e/ou idade que possuía. Ao longo da obra ela retrata não apenas que ganhou mobilidade, ela adquiriu auto-confiança e vislumbrou possibilidades que antes não reconhecia, como aspirações relativas a seu crescimento pessoal. Ela constatou

¹⁵ Essas características da indumentária feminina utilizada no final do século XIX podem ser percebidas ora numa pintura de Cézanne (*Madam Cézanne num vestido vermelho*, de 1888/1890), ora nas peças expostas no Museu da Moda em Canela (<http://www.museudamodadecanela.com.br/>) - acesso em 15 de novembro de 2014) ou no The Metropolitan Museum of Art (http://www.metmuseum.org/toah/hd/wrth/hd_wrth.htm) - acesso em 15 de novembro de 2014).

que a imobilidade física feminina atava-se a outras imobilidades, sociais. Assim, transformar a forma de se mover no espaço era (e é) também uma forma de transformar as relações de gênero (HANSON, 2010). E aqui o termo gênero é pertinente na análise da obra de Willard, e não anacrônico, porque tem no cerne as relações de poder estabelecidas entre indivíduos (SCOTT, 1986), em função do que se entende por “feminino” e “masculino”. E embora a preocupação a respeito de sustentabilidade seja recente, extemporânea a Willard, vale notar que a liberdade adquirida por ela dá-se ao aprender a andar de bicicleta, um modo de transporte não motorizado, com emissões nulas de gases do efeito estufa, de manutenção pouco custosa e, cada vez mais, símbolo de sustentabilidade.

Os primeiros artigos que abordam explícita e articuladamente questões de gênero e de transporte datam do fim da década de 1970, como o editorial de Rosenbloom (1978), em que ela problematiza como serão distribuídas as atividades e, por conseguinte, as viagens da população frente ao fato de que a proporção de mulheres na força de trabalho vinha aumentando rapidamente¹⁶. À época havia quem afirmasse que conforme as desigualdades salariais diminuíssem e os papéis sociais se alterassem, as diferenças nos padrões de viagens sumiriam. Rosenbloom discorda e entre seus argumentos figura o de que as variáveis sócio-econômicas tradicionais pouco explicam as relações de poder e os processos de decisão circunscritos ao ambiente doméstico. Outrossim, identifica-se que nas diversas classes socio-econômicas as mulheres que trabalham ainda continuam sendo as principais responsáveis pelas tarefas domésticas e pelo cuidado com as crianças. Um fato retratado pela autora ilustra o desconhecimento completo do fenômeno por parte dos órgãos oficiais de planejamento de transportes: cientistas contratados pelo *U. S. Department of Transportation* chegaram a declarar que estimular os homens a usar o transporte público e deixar o carro em casa poderia aumentar o consumo de energia e os níveis de poluição, pois as mulheres usariam o carro fazendo viagens mais curtas, em baixas velocidades e com maior consumo de combustível.

Assim como foi para Willard, o ganho de mobilidade pode refletir melhores condições de vida para as mulheres. Não se pode esquecer que na maioria das vezes a viagem é atividade meio e não fim, ou seja, as pessoas precisam de um motivo para fazer uma viagem, querem chegar a algum lugar por um propósito. Caso uma mulher saísse de casa sem propósito, por exemplo no Brasil da década de 1970, seria questionada moralmente. Um homem também poderia ser considerado vagabundo na mesma situação - embora, há de se frisar, o julgamento moral sobre ele seria mais condescendente do que sobre ela. D’Incao (2012, p.235) relata que:

cronistas, viajantes e historiadores [...] exibem um quadro em que a menina ou a mulher [burguesa] candidata ao casamento é extremamente bem cuidada, é trancafiada nas casas, etc.

¹⁶ Em 1978, 54% das mulheres casadas dos Estados Unidos eram assalariadas, mais do que o dobro do que se constatou logo após o fim da Segunda Guerra Mundial (ROSENBLOOM, 1978).

Desta forma, uma das maneiras de adquirir liberdade de movimento foi poder ter motivos que não domésticos para sair de casa, o que não se deu - nem se dá - sem resistência, como o editorial de Rosenbloom já dava pistas. Hanson (1995a apud HANSON, 2010) constatara que as mulheres de Massachusetts, Estados Unidos, tinham dificuldade de encontrar trabalho no final dos anos 1980. Mais recentemente, Silvey e Elmhirst (2003) apontam que mulheres sequer são consideradas para certos postos de trabalho na Indonésia porque não se supõem que possam estar fora de casa após escurecer. Em algumas vilas indianas, Raju (2005), ao caracterizar um projeto que visa o empoderamento feminino, constata que uma das mudanças mais significativas detectadas foi o fato das mulheres poderem sair sozinhas de casa, isto é, poderem existir *per si* no espaço público. O “ganhar a rua” feminino é fundamentalmente ligado ao acesso ao trabalho, ao aumento da participação feminina na população economicamente ativa (KÜNZLER, 1994 apud BEST; LANZENDORF, 2005) e ao crescimento da renda individual. Mandel (2004) mostrou que mulheres que têm mais liberdade para fazer viagens têm maior renda em Porto Novo, Benin. Mesmo ao considerar países com menor desigualdade¹⁷, como a Noruega, observar-se-á que as mulheres casadas trabalham em localidades mais próximas das residências e têm menos poder de escolha geográfico do que seus maridos no que tange às oportunidades de trabalho (HJORTHOL, 2000).

Não obstante, é preciso ter cuidado e não fazer uma relação identitária automática entre **mobilidade** e **empoderamento**. Alterar padrões de mobilidade pode significar alterar relações de poder já que é de alguma forma requisito da acessibilidade à escola, ao trabalho, a hospitais, às lojas, às áreas de lazer, etc. Então, maior mobilidade pode significar mais equidade em algumas situações, mas não necessariamente em todas. Isto é, não há uma relação biunívoca em que maior mobilidade leve, sempre, a mais acesso e oportunidades iguais a todas pessoas. Essas nuances podem ser observadas quando se perfaz uma abordagem interseccional, como fizeram McLafferty e Preston (1991), McLafferty e Preston (1992) e Crane (2007). Gilbert (1998) faz essa ponderação e encara como demasiado simplista entender a mobilidade como empoderamento e a imobilidade como sinal de falta de poder; afinal, é preciso considerar a espacialização dessa mobilidade, considerando suas características sociais, culturais e econômicas. Talvez uma pessoa se desloque menos por ter mais acesso a oportunidades num raio próximo do seu lar e este lar assim seja localizado porque essa pessoa desfruta de melhor posição social e econômica.

McLafferty e Preston (1991) indicam que maior mobilidade não significa mais poder, visto as condições socioeconômicas de quem precisa fazer longas viagens para trabalhar em postos de baixa remuneração. Em artigo de 1992, com base em dados referentes ao final

¹⁷ Para medir a desigualdade é comum utilizar o Índice de Gini, cujos valores são tão mais altos quanto maior for a desigualdade da renda familiar. Estados Unidos apresentam um Índice Gini de 34 (2005); Indonésia, de 39,4 (2005); Benin, de 36,5 (2003); Noruega, de 25 (2008); e Brasil, de 55,3 (2001). Fonte: <<https://www.cia.gov/library/publications/the-world-factbook/fields/2172.html>> Acesso em 29 de novembro de 2014.

dos anos 1980 do norte de Nova Jersey (Estados Unidos), [McLafferty e Preston](#) sustentam a hipótese que as diferenças de gênero na segmentação do mercado de trabalho têm consequências sobre a distribuição espacial (desigual) das minorias. As autoras conseguem atingir uma **abordagem interseccional** entre raça, gênero e classe neste trabalho. As mulheres afro-descendentes, latinas e brancas distribuem-se de forma diversa no espaço urbano. As negras contam com tempos de viagem maiores e menor grau de confinamento espacial que as latinas e as brancas. A “cidade da mulher negra” é maior, mas nem por isso, melhor: elas têm menos acesso a oportunidades de trabalho em regiões próximas das suas residências e geralmente são empregadas em postos de baixa remuneração. A maioria desses postos, ligados ao setor de serviços e não ao de processos de manufatura, são menos vulneráveis ao desemprego. Os homens negros e latinos e as mulheres latinas são contratados mais frequentemente pelo setor manufatureiro e, portanto, são mais suscetíveis ao desemprego estrutural. Essa menor vulnerabilidade ao desemprego da mulher negra, contudo, não implica que as oportunidades de emprego lhes sejam mais acessíveis - elas precisam ir mais longe para consegui-las. Entre as mulheres, as latinas apresentam viagens relativamente curtas, próximo ao padrão das brancas. Porém, contam com alto grau de confinamento e a pior remuneração de todos os grupos analisados. Destarte, [McLafferty e Preston](#) mostram que analisar somente a classe (pela renda, poder de compra, posse de bens, grau de instrução associado), ou somente o gênero, ou somente a raça/etnia pode levar a conclusões muito triviais para fenômenos que são mais complexos. Em estudo mais recente, também nos Estados Unidos, que parte de bases de dados nacionais de 1985, 1995 e 2005, [Crane \(2007\)](#) observa que os tempos de viagens vêm convergindo quando consideradas as várias raças/etnias do mesmo gênero, em especial para mulheres. Ou seja, mulheres negras, brancas, asiáticas e latinas tendem a apresentar menos diferenças entre si ao longo do tempo. Analogamente, esse fenômeno também ocorre dentro do grupo masculino.

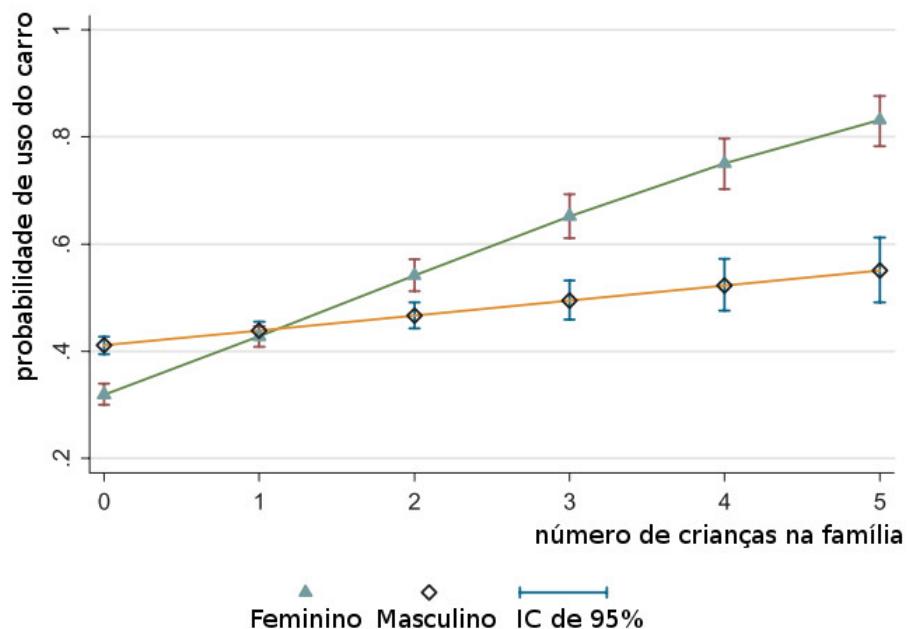
Embora a divisão de trabalho por gênero seja identificada como um fator que influencia a mobilidade, costuma-se ver o trabalho doméstico como uma restrição à participação do mercado de trabalho. Subestima-se o efeito do arranjo familiar no padrão de atividades e de viagens geradas a partir de demandas domésticas. Se no mercado de trabalho vem sendo traçado um caminho que tende a diminuir o desequilíbrio de gênero, no trabalho doméstico ainda é a mulher a grande responsável pela sua execução. Cabe então, analisar as **viagens cujo motivo não seja o trabalho** e outros aspectos relacionados ao arranjo familiar: como a presença de criança interfere na rotina familiar e como se distribui o uso do automóvel entre os membros, quando este está presente.

Em 1997, [Root e Schintler \(1999\)](#) indicam que cerca de 50% das viagens feitas por mulheres, nos Estados Unidos, por motivos pessoais (não trabalho) eram, na realidade, para a família. [Vance e Iovanna \(2007\)](#) trabalham econometricamente com dados em painel da Alemanha, referente ao período de 1996 a 2003, indagando se o sexo (feminino/masculino)

desempenha papel relevante na determinação da probabilidade de utilização do automóvel ou da distância percorrida; e se assim o for, se seria esse papel influenciado por outros atributos socieconômicos do indivíduo ou de seu núcleo familiar. Os autores constataram que as mulheres realizam mais viagens do que os homens quando o motivo não é trabalho. E embora o volume de viagens delas seja maior, a relação de dependência do carro para este tipo de viagem é menor - elas utilizam mais outros modos.

[Vance e Iovanna \(2007\)](#) verificaram que situação ocupacional (empregado(a) ou não), número de crianças na família (ver Figura 5), facilidade de acesso ao transporte público tiveram influência significativa nas viagens feitas por carro (que não para o trabalho), tanto para homens como para mulheres. Essas mesmas variáveis não influenciaram, porém, a distância média dirigida; apenas o acesso (ou não) ao automóvel incidiu sobre esse efeito.

Figura 5 – Simulação de probabilidade de uso do carro para viagens não-trabalho em função do número de crianças na família, na Alemanha



Fonte: Adaptado de ([VANCE; IOVANNA, 2007](#), p.59)

A **presença de criança** é um fator que impacta bastante no padrão de atividades da família, e pode incidir diferentemente sobre pais e mães. Ainda olhando para a Alemanha, o estudo de [Best e Lanzendorf \(2005\)](#) para a região de Cologne indica que mães usam menos frequentemente o carro do que mulheres sem filhos, ao passo que pais usam mais o carro do que homens sem filhos. Este resultado contradiz o encontrado por [Vance e Iovanna](#), indicando que é preciso considerar a diversidade dos contextos socioeconômicos e culturais neste tipo de análise e que não é possível simplesmente transpor conclusões de outros locais para a RMSP.

Goddard et al. (2006) identificaram que a presença de criança na família causa diferença significativa no comportamento de viagens entre homens e mulheres no norte da Califórnia (Estados Unidos). A presença de criança na família tem grande impacto na quantidade de tempo/distância que a mulher dirige automóvel, mas tal efeito não se observa no comportamento masculino. Esse comportamento pode se dever ao fato de que os homens, antes da paternidade, já estão mais familiarizados com o uso do carro e tendem a não trocar sua escolha modal. As mulheres, mães, tendem a ser mais pressionadas pelas atividades a serem cumpridas, que não o trabalho. Elas passam a absorver com mais facilidade a viagem de servir passageiro (a criança). Nesta conjuntura vale ressaltar que as escolas, em especial que atendem as crianças nas primeiras idades, geralmente ficam próximas à residência o que estimula que as viagens para levar crianças à escola sejam feitas a pé, pelas mães. Porém, as metodologias utilizadas nos grandes *surveys*, tanto no Brasil como fora, pecam em detectar viagens curtas a pé. Nas Pesquisas OD da RMSP as viagens feitas integralmente a pé, se estiverem num raio inferior a 500 metros da residência, são desconsideradas se não forem com destino ao trabalho ou à escola.

Fox já apontava que em 1983 as mulheres nos Estados Unidos usavam menos o automóvel e mais o transporte público. Essa priorização masculina no **uso do automóvel** vem persistindo nos Estados Unidos e também na Europa. Hjorthol (2000) ao investigar mulheres e homens casados da região de Oslo, Noruega, observou que em famílias que dispunham de um carro, o marido detinha a prioridade do uso. Polk (2003) indica que os homens usam mais o carro, acumulam mais quilômetros percorridos por ano e fazem mais viagens como ocupantes únicos do que mulheres na Suécia. Embora ter à disposição um carro para uso privado é o fator que mais influencia o seu uso em viagens motivo “manutenção do lar”, segundo Best e Lanzendorf (2005), na Alemanha, isso ainda não é suficiente para que as mulheres usem mais o carro do que os homens no geral. Estes autores elencam ainda outros fatores que estimulam o uso do carro: renda familiar e participação no mercado de trabalho. Assim, se há mais renda familiar, há mais condições financeiras de adquirir um carro para tê-lo à disposição. E se há mais pessoas na família que fazem parte do mercado de trabalho, há mais interesse em comprar um carro, seja pelo aumento da renda familiar, seja pela necessidade de contar com um modo com flexibilidade de rota.

Portanto, as pesquisas apontam para o fato de que, *ceteris paribus*, quando existe a posse do automóvel na família, este fica mais frequentemente à disposição dos homens do que das mulheres. Dessa forma, as mulheres andam mais de transporte público e, ao andarem de automóvel: (i) são com maior frequência passageiras, remetendo à ideia de “não andar desacompanhada fora de casa”, e (ii) servem passageiro mais frequentemente, remetendo às tarefas do cuidado (HIRATA; GUIMARÃES, 2012) com crianças e idosos (ROSENBLUM, 2000; ROSENBLUM, 2003). Se isso por um lado reflete menor autonomia e independência das mulheres por construção histórica, por outro lado, percebe-se também que elas conscientemente indicam estar mais dispostas a uma migração modal para obter um

padrão mais sustentável de deslocamentos do que eles. Mulheres e homens demonstram atitudes diferentes em relação à proteção do meio ambiente e à sustentabilidade, apesar da taxa de motorização feminina vir crescendo nos países mais industrializados (ROOT; SCHINTLER, 1999). Polk (2003) se propõe a estudar se, na Suécia, as mulheres são potencialmente mais adaptáveis a um sistema de transportes mais **sustentável** do que os homens. A autora conclui que sim, as mulheres tendem a expressar mais preocupação em relação às questões ambientais e declaram maior vontade de reduzir o uso do carro. O carro é modo que mais ocupa espaço urbano e com maior consumo energético por pessoa (ver Tabela 2); desta maneira, compreender porque alguém faz uso intensivo do carro e também porque um grupo não o utiliza tanto assim, pode abrir caminhos para políticas de transporte mais sustentáveis.

Tabela 2 – Consumo energético teórico dos modos de transporte em lotação plena

Modo de Transporte	gramas equivalentes de petróleo para mover um passageiro por um quilômetro
ônibus comum	4,1
metrô	4,3
motocicleta	11,0
automóvel	19,3

Fonte: Adaptado de (VASCONCELLOS, 2012, p.84)

Schwanen, Dijst e Dieleman (2002) indicam que são variáveis importantes de análise a **distância** viajada, o **modo** utilizado e o **tempo** de viagem - este último é um fator que pesa bastante nas decisões associadas à viagem (fazê-la ou não, que modo utilizar e que rota percorrer).

Fox indicava os padrões de viagens das mulheres nos Estados Unidos em 1983: elas faziam menos viagens, mais curtas e rápidas. Quase dez anos depois, Johnston-Anumonwo (1992) revisita a hipótese de que mulheres fazem viagens mais curtas que homens em função de suas socialmente construídas atribuições domésticas. O autor foca-se no tipo de família e toma para essa análise a variável número de trabalhadores(as) da família. Assim, constitui-se o grupo das famílias em que há um(a) trabalhador(a) e daquelas com dois trabalhadores(as). Com dados de Baltimore, Estados Unidos, no grupo em que o arranjo familiar conta com mais de uma pessoa que trabalha, as diferenças entre distâncias de viagens de homens e mulheres tendem a ser maiores - ainda que sejam controlados outros fatores como, por exemplo, presença de criança na família.

Também no EUA, já no século XXI, [Crane \(2007\)](#) utiliza dados de 1985, 1995 e 2005 em suas análises e novamente afirma que mulheres fazem viagens mais curtas que homens. [Crane](#) vai além e contesta estudos que apontam que o *gender gap* dos tempos de viagem esteja diminuindo e das distâncias de viagem tenham até sumido em algumas áreas. Constata que ainda persistem diferenças: as distâncias percorridas por homens e por mulheres convergem muito lentamente e os tempos divergem.

Na Europa, o panorama se mantém semelhante ao dos Estados Unidos. [Fagnani \(1983\)](#) estuda os padrões de deslocamento das mulheres em Paris e constata que lá também elas realizam viagens mais curtas. [Schwanen, Dijst e Dieleman \(2002\)](#), a partir do *National Travel Survey* holandês de 1998, analisam tempo, motivo e modo da viagem. Encontram evidências que fatores sociodemográficos (sexo, número de trabalhadores(as) na família, idade e grau de instrução) e contexto espacial da localização da residência influenciam o tempo médio de viagem diário. Os efeitos da posse de carro e da renda se dão de forma indireta e interagem com a escolha modal ou o número de trabalhadores(as) da família, segundo os autores. Do ponto de vista da interação com características urbanas, os tempos de viagem de carro tendem a aumentar quanto maior for o grau de urbanização e quanto mais policêntrica for uma região.

No conjunto Ásia e Oceania, o diagnóstico permanece. Ao analisar as viagens motivo trabalho da região metropolitana de Melbourne, Austrália, [Howe e O'Connor \(1982\)](#) expõem que as taxas de participação feminina na força de trabalho australiana eram baixas à época, levanta a hipótese de que isso se dê por alguma dificuldade de acesso às oportunidades de trabalho. Constata que mulheres fazem viagens mais curtas que homens e conclui que para incrementar a participação feminina no mercado de trabalho é preciso que ofertas de emprego estejam mais distribuídas espacialmente. [Cusset \(1997 apud VASCONCELLOS, 2001\)](#) constata que as mulheres em Hanoi (Vietnam) fazem a maior parte das viagens diárias a pé (54%); para os homens essa taxa cai para cerca de 39%, indicando maior motorização deles. [Lee e McDonald \(2003\)](#) também constata que para Seul, Coreia, tanto os tempos quanto as distâncias de viagens são menores para as trabalhadoras. Nessa cidade, trabalhadores(as) solteiros(as) contam com distâncias de viagem maiores que os(as) casados(as) e, merece destaque o detalhe de que trabalhadores cujas esposas trabalham têm viagens mais curtas do que aqueles cujas esposas não trabalham. Por fim, [Lee e McDonald](#) aponta que a responsabilidade por cuidar das crianças é um fator de encurtamento das viagens das mulheres casadas coreanas.

No Brasil, em específico na RMSP, [Vasconcellos \(2001\)](#) indica também que as mulheres fazem menos viagens e andam mais a pé do que os homens. Como os padrões de deslocamentos das mulheres são muito moldados pelas atividades ligadas ao ambiente doméstico, [Strambi e Bilt \(1998\)](#), com base na Pesquisa OD-1987, lançam olhar sobre o núcleo familiar e constatam que, para categorias de menor renda, o número médio de

viagens cai conforme cresce o tamanho da família, o que pode indicar insuficiência de renda para arcar com as tarifas de transporte dos “membros adicionais”. Percebe-se que o estudo da mulher, da **família** e da mulher na família são relevantes, pois são fortes fatores sociais e culturais condicionantes de comportamentos. [Vasconcellos \(2001, p.119\)](#) endossa essa abordagem ao dizer que:

O papel das mulheres é especialmente relevante para entender os padrões diários de deslocamento nas famílias dos países em desenvolvimento. A melhor forma de entender estes padrões é começando pela divisão de tarefas domésticas e depois examinar condicionantes culturais e religiosos de sua mobilidade.

[McGuckin e Murakami \(1995\)](#), com base no *Nationwide Personal Transportation Survey* de 1995 estadunidense examinam o **encadeamento de viagens** feitas durante a semana de homens e mulheres em fase adulta. Constatam que mulheres fazem cadeias de viagens com mais segmentos para que possam acomodar suas responsabilidades domésticas. Essa segmentação é ainda mais sentida por elas quando se conta com a presença de criança na família. As autoras ainda apontam uma mudança na dinâmica doméstica em relação às tarefas e responsabilidades. Dez anos depois, [Mcguckin, Zmud e Nakamoto \(2005\)](#) publicaram estudo sobre o encadeamento de viagens com início na residência e término no trabalho de acordo com sexo (feminino/masculino) e ciclo de vida a partir de base de dados nacionais dos Estados Unidos de 2001. Constataram que em lares com pai e mãe, em que ambos trabalham, as viagens cujo propósito de servir passageiro é deixar as crianças na escola têm maior probabilidade de serem acomodadas na cadeia de viagens da mulher do que do homem. Na Europa, situação semelhante é encontrada. Na Espanha e no Reino Unido, em 1997, grande parte das viagens femininas era para servir passageiro, em sua maioria, filhos(as) ([ROOT; SCHINTLER, 1999](#)). Em Oslo, mulheres casadas fazem mais encadeamento de viagens por conta de responsabilidades domésticas do que homens casados ([HJORTHOL, 2000](#)). Logo, se houve algum rearranjo na divisão de tarefas domésticas, essa mudança não foi profunda o suficiente para gerar alterações perceptíveis nos padrões de deslocamentos de homens e mulheres em diversos países.

Como o gênero configura e influencia a mobilidade, e como a mobilidade configura e influencia o gênero é o foco da discussão empreendida por [Hanson \(2010\)](#). Entretanto, o mais comum na pesquisa de gênero e mobilidade é um campo de conhecimento contemplar o outro, com pouco esforço de buscar literatura ou metodologia de intersecção entre as áreas e, com isso, parte-se de conceitos e hipóteses que conduzem a resultados pouco convergentes ao final. [Hanson \(2010, p.09\)](#) afirma ainda que em geral o debate ocorre em termos muito genéricos em torno da mobilidade/imobilidade com reflexos de uma ideologia dual que identificam a mulher/feminilidade com a casa, o espaço doméstico, movimentos restritos e o homem/masculinidade com a rua, o espaço público, movimentos livres. Enxergar o mundo a partir de um código binário de gênero foi uma construção

longa e aboli-la por completo é uma tarefa a que este trabalho não se propõe, embora sempre que possível tente considerar a categoria de gênero para além do sexo, binário.

A maior parte da literatura revisada diz respeito aos países do hemisfério Norte, mais industrializados, concentrando-se na Europa e nos Estados Unidos. Isso deve-se principalmente a dois fatores: (i) são países em que os temas gênero, transporte e sustentabilidade são mais sistematicamente investigados; (ii) alguns países da Ásia também têm alguma produção dentro da temática desta pesquisa, mas muitas vezes em língua que não é de grande difusão no ocidente (como mandarim ou coreano). O que realmente chama a atenção é a parca literatura sobre a situação na América Latina. A maioria dos estudos encontrados baseiam-se em grandes bases de dados quantitativos que são tratados de forma agregada. Daí, conclui-se que, agregadamente, os comportamentos de homens e de mulheres são diferentes, em diversos países e continentes. Uma série de aspectos relacionados à viagem são analisados para fazer essa constatação: quantidade por pessoa, tempo, distância, encadeamento, modo e motivo. Outros aspectos, ligados às características de indivíduos, são frequentemente envolvidos nas análises: sexo, idade, estágio do ciclo de vida (individual e familiar), estado civil, presença de filhos na família, papel exercido dentro da família, situação ocupacional e renda.

3 Considerações Metodológicas

Esta pesquisa lida fundamentalmente com dados secundários, principalmente do Metrô-SP através de suas Pesquisas OD e, em menor grau, com dados socioeconômicos advindos de pesquisas PNAD¹ e censos do IBGE². Por dados secundários entendem-se aqueles que “já foram coletados para objetivos que não os do problema” (MALHOTRA, 2001, p.127). Trabalhar com dados secundários traz vantagens e desvantagens. Como vantagens tem-se o acesso relativamente fácil, o baixo custo de coleta e a rapidez de obtenção dos dados. Como desvantagem tem-se que o propósito da coleta dos dados difere daquele para que estão sendo utilizados aqui. Isto é, a formulação dos questionários e o desenho dos bancos de dados buscam responder a perguntas diferentes das propostas por esta dissertação. Essa desvantagem não inviabiliza o uso dos dados, mas é uma informação que deve estar em mente ao manipulá-los e analisá-los. Malhotra (2001, p.128) indica que os dados secundários podem auxiliar a: (i) identificar o problema; (ii) definir melhor o problema; (iii) desenvolver uma abordagem do problema; (iv) formular uma concepção de pesquisa adequada; (v) responder a certas perguntas de pesquisa e testar algumas hipóteses; (vi) interpretar dados primários com mais critério.

Nesta dissertação os bancos de dados e manuais de referência das respectivas Pequenas OD foram requisitados por meio de formulário *online* do e-SIC³ do governo do Estado de São Paulo e disponibilizados pelo Metrô-SP em mídia digital para retirada em dentro do prazo de 20 dias estabelecido pela lei de acesso à informação pública. Os dados provenientes de PNAD, censos ou outras pesquisas foram obtidos por meio de relatórios públicos, disponibilizados *online* e têm suas fontes indicadas ao longo do texto. Entende-se que os dados advindos das Pesquisas OD podem auxiliar na tarefa de identificar se existem diferentes padrões de mobilidade de acordo com o gênero na RMSP, pois contêm informações sobre os deslocamentos de indivíduos, com representatividade em suas respectivas zonas e indicados os respectivos fatores de expansão nos bancos de dados. Para investigar a evolução e os motivos dessas diferenças têm-se como base as hipóteses advindas da **revisão da literatura**: ao gênero feminino é atribuído um determinado papel social e

¹ A PNAD é a Pesquisa Nacional por Amostra de Domicílios, realizada bianualmente pelo IBGE (referência no mês de setembro) com objetivo de investigar características socioeconômicas da população. Fonte:<http://www.previdencia.gov.br/arquivos/office/3_081014-105206-595.pdf> Acesso em 20 de novembro de 2014

² O censo demográfico é uma pesquisa realizada decenalmente pelo IBGE com objetivo de caracterizar sociodemograficamente a população brasileira. O primeiro foi feito em 1872 e o último data de 2010. Fonte:<<http://cod.ibge.gov.br/234lq>> Acesso em 20 de novembro de 2014

³ O e-SIC é o Sistema Eletrônico do Serviço de Informações ao Cidadão que possibilita que qualquer pessoa, física ou jurídica, encaminhe pedidos de acesso à informação, acompanhe o prazo e receba a resposta da solicitação realizada para órgãos e entidades governamentais. Fonte: <<http://www.sic.sp.gov.br/>> Acesso em 20 de abril de 2014

lhe é inputada a realização de uma maior diversidade de trabalhos (remunerados e não remunerados) que precisam ser acomodados no cotidiano. Os dados socieconômicos podem dar indícios daquelas hipóteses ligadas à influência das características socioeconômicas sobre as ações de indivíduos. Dados como idade e ocupação (se trabalha, se estuda, etc.) podem dar indícios para compreender a influência do ciclo de vida no comportamento dos indivíduos. Dados como situação na família (pessoa responsável, cônjuge, etc.) e presença de filhos podem auxiliar a compreender a dinâmica familiar e seu reflexo nos padrões de atividades das pessoas. Dados como posse de automóveis, motocicletas, bicicletas e modos utilizados nas viagens, podem contribuir para compreender a relação que é estabelecida com o transporte público e o transporte privado, mais especificamente o carro.

Os dados secundários das **Pesquisas OD** foram analisados segundo alguns critérios, observados no Quadro 1. Os objetivos primários da coleta dos dados são apresentados na Seção 4.1. Em relação à natureza dos dados, para que fossem melhor utilizados nesta dissertação, foram feitas compatibilizações entre as zonas (geográficas) de análise e os diversos bancos de dados, processo mais detalhadamente explicado no Capítulo 5. Em relação à confiabilidade, os dados foram adquiridos diretamente do Metrô-SP, responsável pela coordenação da coleta e compilação dos dados, e que goza de boa reputação e pioneirismo no Brasil na realização de pesquisas dessa natureza. Em relação à atualidade dos dados, no Capítulo 4 esclarece-se a periodicidade e datas de referência das OD-1977, OD-1987, OD-1997 e OD-2007. Em relação às especificações e metodologia das Pesquisas OD, no Capítulo 4 são descritos brevemente os métodos de coleta, amostragem, conceitos utilizados, além das decorrentes limitações.

Quadro 1 – Critérios para Avaliação de Dados Secundários

Critérios	Questões
Objetivo	Por que os dados foram coletados?
Natureza	Definição de variáveis chave; unidades de medição; categorias usadas e relações examinadas
Confiabilidade	Experiência; credibilidade; reputação e integridade da fonte
Atualidade	Prazo entre coleta e publicação; frequência das atualizações
Especificações e Metodologia	Método de coleta de dados; índice de respostas; qualidade dos dados; técnica de amostragem; tamanho da amostra; criação do questionário e trabalho de campo

Fonte: Adaptado de ([MALHOTRA, 2001](#), p.129)

Numa análise dos padrões de viagem do sudeste de Michigan, entre 1965 e 1980, **KOSTYNIUK e KITAMURA (1984)** concluíram que o padrão de deslocamentos diários não é estável ao longo do tempo. Também nos Estados Unidos, **Mcguckin, Zmud e Nakamoto (2005)** utilizam dois *surveys* nacionais para avaliar tendências nos encadeamentos de

viagem. Frandberg e Vilhelmsen (2011) utilizam dados (intermitentes) de quase 30 anos de *surveys* nacionais para avaliar tendências na mobilidade das pessoas na Suécia, focando no gênero e nos coortes. Cadestin, Dejoux e Roux (2013) usam o *National Travel Survey* de 1973 a 2007 para analisar diferenças de mobilidade entre mulheres e homens com mais de 55 anos na França. Deste modo, além de estar entre os objetivos desta dissertação avaliar **tendências ao longo do tempo**, diversos autores indicam ser relevante elaborar estudos longitudinais, inclusive com o foco de gênero. Para isso ser possível, foi necessário criar um Banco de Dados Unificado (BDU). Para a organização do BDU foi elaborado um *layout* unificador das variáveis e, para cada variável, foi necessário compatibilizar categorias e fazer testes de validação. Essas transformações foram feitas utilizando linguagem *python* e as rotinas estão disponíveis no Anexo H. Também foram feitos testes de validação e de consistência globais do BDU e, por conseguinte, excluídos alguns registros.

É possível fazer análises agregadas (por zonas) ou desagregadas (por famílias ou indivíduos). As **análises desagregadas** permitem, por exemplo, avaliar o comportamento da demanda e considerar modelos baseados em teorias de atividades humanas, que consideram condicionantes familiares e externos aos deslocamentos ao invés das viagens isoladamente (JONES, 1981). A desagregação permite considerar como unidade de estudo a família ou o indivíduo, e tem sido indicada como vantajosa do ponto de vista conceitual (ORTÚZAR; WILLUMSEN, 1994). Para ser possível, neste trabalho, elaborar análises desagregadas, novas variáveis de viagens, pessoas e famílias foram criadas. A descrição mais detalhada de todo este **tratamento de dados** está no Capítulo 5.

O Capítulo 6 contém análises, resultados e discussões - realizados utilizando o software livre de estatística computacional R⁴. Com objetivo de obter alguma familiaridade que essa grande massa de dados que representa o BDU, foram realizadas algumas **estatísticas descritivas** (tabelas de frequências, box plot, histogramas, distribuições, entre outras), segmentadas por ano. Essas análises, apresentadas na Seção 6.1, permitem perceber macro tendências de algumas variáveis no tempo. Na sequência, na Seção 6.2 são apresentadas algumas **análises preliminares** dos dados, feitas simultaneamente para os bancos de dados das quatro *cross-sections* (1977, 1987, 1997 e 2007). Conforme indicado pela literatura, buscou-se compreender o comportamento de pessoas e famílias, ao longo do tempo, segundo modo, motivo, duração, distância e número de viagens. Aqui, já se começa a fazer algumas segmentações das análises pela variável sexo.

A análise das atividades humanas por segmentos da população possibilita identificar necessidades distintas de transporte e, a partir dos padrões de atividades e viagens, tais informações podem ser aplicadas na formulação ou revisão de modelos de projeção de demanda (MAHMASSANI, 1988). Sendo necessário considerar variáveis e categorias relevantes, bem como as interações entre elas, Strambi e Bilt (1998) utilizaram técnicas de

⁴ Página do projeto R: <<https://www.r-project.org/>>

segmentação para identificar grupos razoavelmente homogêneos, viabilizando o estudo do comportamento desses grupos e também dos fatores que diferenciam os grupos. Neste trabalho, a técnica inicialmente utilizada para a segmentação dos grupos foi a **análise de conglomerados** (*clusters*). Como entrada de dados foi utilizados um conjunto de variáveis com atributos de viagem da família. Quais foram essas variáveis e quais os critérios utilizados para definição do número de grupos podem ser observados na Seção 6.3. Como medida de similaridade foi usada a distância euclidiana e como métodos de agrupamentos foram usados Ward e centroide, sendo este último preferido devido à maior robustez a *outliers*.

Realizada uma nova análise de conglomerados a partir dos grupos formados na primeira, foram investigadas as características das famílias e indivíduos dos grupos formados segundo a similaridade de seus padrões de deslocamento. Tal investigação foi empreendida de duas formas. A primeira delas observou as diferenças percentuais entre valores mínimo e máximo de cada variável (quantitativa) ou de cada categoria de variável (qualitativa). Aqui visou-se obter um potencial conjunto de variáveis explicativas para regressões que desenvolvidas mais à frente. A outra forma empregou **regressões logísticas multinomiais** (Seção 6.4) com as mesmas variáveis que foram *input* para a clusterização, visando analisar o peso delas na explicação dos grupos para, daí, eleger uma variável dependente de relevância para explorar melhor na sequência.

Com os resultados da análise de conglomerados e das regressões logísticas foi elencado um conjunto de variáveis explicativas e dependentes. Considerando a revisão de literatura e teoria subjacente, foram escolhidas quais as variáveis a entrar no modelo de regressão e também, quais os grupos alvo de análise. Qual o tipo de regressão utilizada foi função da natureza da variável escolhida. Como a variável dependente escolhida foi TOT_VIAG, um dado de contagem com poucos valores e variância bastante diferente da média, a técnica escolhida foi a **regressão quasi-poisson**, cujos detalhes de aplicação estão na Seção 6.5.

Por fim, esta pesquisa utiliza um banco de dados secundário em que a informação relativa a gênero nasce quase que exclusivamente da variável sexo, codificada binariamente - certamente uma **limitação** - muito embora seja o tipo de pergunta que o(a) entrevistador não faz diretamente, mas marca no papel a partir de julgamento visual. Isso significa que uma pessoa transsexual cuja performatividade seja feminina, provavelmente será identificada como mulher no questionário. Por um lado, este trabalho não traz elemento metodológico inovador que rompa com a leitura binária de gênero, por conta da limitação que os dados de origem impõem. Por outro lado, a articulação da variável sexo com outras (como situação familiar), pode apontar na direção de finalmente compreender o gênero para além do sexo na área dos transportes.

4 As Pesquisas Origem Destino da RMSP

4.1 Periodicidade e Objetivos da Pesquisa Origem Destino

A Pesquisa Origem Destino (Pesquisa OD) vem sendo realizada a cada dez anos pela Companhia do Metropolitano de São Paulo (Metrô-SP), a partir de 1967. Assim, até hoje foram realizadas cinco Pesquisas OD (1967, 1977, 1987, 1997 e 2007), das quais este trabalho abrangerá as quatro últimas, cobrindo uma janela temporal de 30 anos. O intervalo de dez anos foi considerado pelo Metrô-SP muito longo mediante as rápidas transformações no espaço urbano; assim, em 2002 e em 2012 foram feitas Pesquisas de Aferição, com menor amostragem e zonas maiores. Cabe esclarecer que estas pesquisas de aferição não serão objeto de análise do presente estudo.

A Pesquisa OD nasceu com a missão de compor uma base de dados que servisse de suporte a decisões de planejamento de transporte urbano na Região Metropolitana de São Paulo, que hoje abarca 38 municípios, além de São Paulo. Atualmente, além de cumprir esse papel, também é ferramenta de suporte para o planejamento urbano de maneira mais sistemática, bem como para a formulação de políticas públicas segmentadas, nas áreas de educação, saúde e segurança pública, por exemplo ([METRÔ-SP, 2008](#)).

4.2 Descrição Sucinta

A Pesquisa OD do Metrô-SP para a RMSP é composta de duas partes complementares, a saber, a Pesquisa Domiciliar e a Pesquisa de Linha de Contorno. A Pesquisa Domiciliar tem como escopo as viagens internas à Região Metropolitana de São Paulo (RMSP); nela são escolhidos os domicílios por amostragem, cujo critério será melhor discutido adiante, em que todos habitantes respondem a um questionário estruturado referente às viagens feitas no dia útil anterior à visita dos pesquisador. Já a Pesquisa de Linha de Contorno monitora pontos de entrada e saída (limites) da RMSP a fim de captar as viagens com origem dentro da RMSP e destino fora, vice-versa, ou ainda viagens que a atravessam. O presente trabalho tem como foco as viagens feitas internamente à RMSP, portanto, as bases de dados consideradas serão apenas aquelas advindas das Pesquisas Domiciliares.

A Pesquisa OD considera a dimensão espacial dos deslocamentos considerando as zonas de origem e de destino. Tais zonas tiveram seus limites alterados e área de cobertura expandida desde 1967. Na Tabela 3 é possível observar quantos municípios da RMSP foram envolvidos em cada pesquisa e em quantas zonas eram divididos. A correspondência

entre as diversas zonas é feita por uma unidade de compatibilização chamada Unidade de Correspondência de Zona (UCOD), em relação às quais todas zonas têm referência. Para que seja possível realizar uma análise de evolução temporal conjugando dados de diversas OD é preciso organizar todas as informações de maneira coerente, assim, é apresentado no Anexo A as 67 UCOD com as respectivas zonas correspondentes para 1977, 1987, 1997 e 2007. Para tal consolidação ser feita, parte das informações foi recebida do Metrô-SP e parte foi fruto de compilação própria.

Tabela 3 – Características Amostrais das Pesquisas OD

Ano	Municípios da RMSP	Zonas
1977	27	243
1987	38	254
1997	39	389
2007	39	460

Fonte: Compilação a partir de [Metrô-SP\(1977, 1987, 1997, 2007\)](#)

4.3 Dados Coletados

A Pesquisa OD coleta dados referentes a domicílios, famílias, indivíduos e viagens, o que possibilita buscar relações entre características de deslocamentos e de indivíduos (e respectivas famílias e domicílios), e também características socioeconômicas. Em 1987, 1997 e 2007 a amostra de domicílios é do tipo estratificada por faixas de consumo de energia elétrica¹ - isso se dá por dois fatores: (i) as concessionárias possuem bases cadastrais de registro de domicílios mais confiáveis e representativas; (ii) “o consumo de energia elétrica tem correlação com a renda familiar, que por sua vez tem correlação com o número de viagens da família” ([METRÔ-SP, 2008](#), p.10). Esse esquema de amostragem estratificada buscou, em todos anos obter nível de confiança de 95%. Nas zonas em que não foi possível utilizar esse arranjo, foi feita amostra aleatória simples, com erros em torno de 7,5%. Na Tabela 4 é possível observar algum dados relativos às amostras.

Segundo o Metrô-SP, definido o tamanho da amostra total, define-se o tamanho de amostra para cada zona e, a partir daí, procede-se um sorteio de endereços por faixa de consumo energético - etapa esta realizada pelas concessionárias, que fornecem apenas os endereços dos domicílios selecionados, além de alguns adicionais para substituição caso

¹ Na década de 1970 a companhia telefônica TELESP realizou um cadastro de domicílios, que foram categorizados segundo o padrão arquitetônico percebido externamente à residência. Nessa categorização eram considerados critérios como, por exemplo, se a construção é do tipo geminada ou não, etc. Os domicílios eram então classificados de 1 a 5, em que 1 significa o melhor parâmetro e 5 o pior (favela). O Metrô-SP utilizou esse cadastro da TELESP, complementando-o onde necessário, para fazer a estratificação da amostra de domicílios para a Pesquisa OD-1977.

necessário. Os selecionados recebem comunicação oficial por carta do Metrô-SP contendo as informações pertinentes à pesquisa. Quando no domicílio, os(as) pesquisadores(as) aplicam o questionário a todas pessoas que moram ali.

Tabela 4 – Características Gerais das Pesquisas OD - amostras

Ano	Domicílios	Pessoas entrevistadas do sexo feminino	Pessoas entrevistadas do sexo masculino
1977	26.132	55.866	52.162
1987	26.070	57.637	53.176
1997	23.841	51.454	47.326
2007	29.957	49.116	42.289
Total	106.000	214.075	194.954

Fonte: Compilação a partir de [Metrô-SP\(1977, 1987, 1997, 2007\)](#)

Cabe destacar que, segundo as amostras, em todas as Pesquisas OD a quantidade de mulheres entrevistadas é superior à quantidade de homens entrevistados, provavelmente devido ao fato de a pesquisa ser domiciliar e ser mais provável encontrar mulheres do que homens em casa. A Tabela 5, que apresenta os dados expandidos, foi feita para comparar os dados populacionais obtidas a partir das Pesquisas OD com os dos censos do IBGE tendo em vista averiguar se houve um possível viés na coleta dos dados. Todos valores de população das Pesquisas OD foram inferiores aos valores dos censos demográficos, o que era de se esperar já que os censos são realizados cerca de 3 anos depois das Pesquisas OD. A estimativa de proporção da população feminina segundo a Pesquisa OD foi sempre um pouco superior à do censo do IBGE - as diferenças variaram entre 0,2 e 0,8 pontos percentuais, para 1997 e 1987 respectivamente. Assim, entende-se que os dados (e suas expansões) gerados pelas Pesquisas OD são confiáveis para uma análise de gênero.

A coleta, consistência e digitação dos dados são de responsabilidade de institutos de pesquisa contratados pelo Metrô-SP e que variaram ao longo do tempo. Após a consolidação primeira do banco de dados, é calculado e aplicado um fator de expansão aos resultados amostrais dos domicílios segundo a expressão (4.1), segundo fontes oficiais (Metrô-SP). Depois determina-se, por consequência, um fator de correção referente às famílias e às pessoas. As viagens de quem usou o modo metrô são expandidas levando em consideração a entrada de passageiros no sistema Metrô-SP no mês de referência da pesquisa. Situação análoga ocorre com o trem metropolitano. As viagens de quem usou outro modo que não metrô e/ou trem teve seu fator de expansão de viagens determinado pelo total de passageiros transportados pelo sistema de ônibus (em 2007 foram utilizados os dados provenientes de Bilhete Único da SPTrans).

Tabela 5 – Características Gerais segundo as Pesquisas OD - expansões

Ano	Pessoas do sexo feminino (OD)	Pessoas do sexo masculino (OD)	Total de pessoas (OD)	Ano	Pessoas do sexo feminino (IBGE)	Pessoas do sexo masculino (IBGE)	Total de pessoas (IBGE)
1977	5.266.421 (51,3%)	5.006.397 (48,7%)	10.272.818 (100%)	1980	6.280.841 (50,6%)	6.134.327 (49,4%)	12.415.168 (100%)
1987	7.404.822 (52,0%)	6.843.034 (48,0%)	14.247.856 (100%)	1991	7.909.092 (51,2%)	7.535.846 (48,8%)	15.444.938 (100%)
1997	8.709.011 (51,9%)	8.082.041 (48,1%)	16.791.052 (100%)	2000	9.249.192 (51,7%)	8.632.511 (48,3%)	17.881.703 (100%)
2007	10.286.373 (52,7%)	9.248.232 (47,3%)	19.534.605 (100%)	2010	10.250.547 (52,1%)	9.433.428 (47,9%)	19.683.975 (100%)

Fonte: Compilação a partir das Pesquisas OD de 1977, 1987, 1997 e 2007, bem como dos censos demográficos de 1980, 1991, 2000 e 2010.

$$\text{Fator de expansão de domicílio}_i = \frac{\text{Total de domicílios da zona}_i}{\text{domicílios da amostra da zona}_i} \quad (4.1)$$

Vale fazer algumas considerações acerca da renda familiar. Nem todas as pessoas respondem qual é a renda familiar, porém, trata-se de uma das informações mais importantes para descrever o comportamento das pessoas (SHEARMUR, 2006). Assim, nos casos em que a renda não foi informada pelo(a) entrevistado(a), ela é atribuída, mas não sem critério. A atribuição da renda familiar baseou-se na pontuação estabelecida por um critério nacional², que variou ao longo do tempo - tais informações podem ser vistas no Quadro 2.

Com isso, nos casos em que as pessoas não informaram a renda mas declararam os bens de consumo da família, a variável “renda familiar mensal” foi atribuída por meio das equações de regressão³ (4.2), (4.3) e (4.4), cuja função é estimar o poder de compra da família. Nesses critérios de classificação econômica existe a orientação de que a categoria automóvel não deve considerar táxis, vans, *pickups* usadas para fretes ou qualquer veículo usado para atividades profissionais, nem tampouco devem ser considerados veículos de uso misto (lazer e profissional) (ABEP, 2009). Essa mesma orientação em relação aos

² Em 1987 foi usado o Critério ABA, em 1997 foi usado o critério ABIMEPE e, em 2007 foi utilizado o Critério Brasil, todos muitos semelhantes em metodologia que visa a classificação em categorias de capacidade de consumo segundo a posse de bens de consumo e do grau de instrução “do chefe da família” Fonte: <<http://www.abep.org/new/criterioBrasil.aspx>> Acesso em 17 de novembro de 2014.

³ As equações de regressão de 1987, 1997 e 2007 foram obtidas por meio de correspondência eletrônica com Emilia Mayumi Hiroi, Coordenadora de Pesquisa e Avaliação de Transporte do Metrô-SP.

Quadro 2 – Dados para atribuição de renda familiar

Ano	Mês de Referência	Classificação de Referência para Atribuição da Renda
1977	setembro	Função do Salário Mínimo
1987	setembro	Critério ABA/ABIMEPE (análogo ao Critério Brasil)
1997	outubro	Critério Brasil (ABIMEPE)
2007	outubro	Critério Brasil (ABEP)

Fonte: Compilação de informações obtidas por meio de correspondência eletrônica com Emilia Mayumi Hiroi, Coordenadora de Pesquisa e Avaliação de Transporte do Metrô-SP

automóveis é feita pelos manuais das Pesquisas OD ([METRÔ-SP, 1977](#); [METRÔ-SP, 1987](#); [METRÔ-SP, 1997](#); [METRÔ-SP, 2007](#)) tornando o conjunto coerente.

$$RFM_{87} = e^{(9,126 + 0,05051 * PONTUACAO_{ABA})} \quad (4.2)$$

$$RFM_{97} = e^{(5,672 + 0,03259 * PONTUACAO_{ABIMEPE})} \quad (4.3)$$

$$RFM_{07} = e^{(5,864 + 0,084 * PONTUACAO_{BRASIL})} \quad (4.4)$$

Nas famílias em que não se obteve nem declaração da renda, nem informações suficientes sobre bens de consumo, a renda foi atribuída à família a partir da mediana da zona a que pertencia e com mesmo grau de instrução do(a) “chefe da família”.

4.4 Conceitos Adotados

A seguir, são replicados alguns conceitos utilizados pelo Metrô-SP no desenvolvimento das Pesquisas OD, a saber, *zona*, *família*, *respondente qualificado*, *modo coletivo*, *modo individual*, *modo não motorizado*, *modo motorizado*, *modo principal*, *viagem*, *viagem a pé*:

- (i) É considerada *família*: uma pessoa que more só, ou um conjunto de pessoas ligadas por laços de parentesco ou de dependência econômica que morem no mesmo domicílio; ou, ainda, conjunto de, no máximo, cinco pessoas que mesmo não tendo laço de parentesco morem num mesmo domicílio. O(a) empregado(a) doméstico(a) que more com algum outro parente na casa do patrão será considerada como outra família, mas caso o(a) empregado(a) more sozinho(a) na residência onde trabalha, será considerado(a) como parte da família do empregador.
- (ii) Compõem o *modo coletivo* o metrô, o trem, o ônibus, o microônibus, o transporte fretado, o transporte escolar, a lotação, a van, o trólebus.
- (iii) Compõem o *modo individual* o automóvel, o táxi, a motocicleta e a bicicleta.
- (iv) São considerados *modos não motorizados* os modos a pé e bicicleta.
- (v) São considerados *modos motorizados* os demais modos exceto a pé e bicicleta.
- (vi) *Modo principal* é o modo de maior hierarquia dentre os modos utilizados numa mesma viagem. Conforme estabelecido pelo Metrô-SP, a hierarquia desses modos é a seguinte, nesta ordem, do que predomina sobre qual: metrô, trem, ônibus, transporte fretado, transporte escolar, lotação, táxi, dirigindo automóvel, passageiro de automóvel, motocicleta, bicicleta, outros e a pé.
- (vii) *Respondente qualificado* é a pessoa com 10 anos ou mais, residente no domicílio sorteado e capaz de responder às perguntas feitas pelo pesquisador. Uma pessoa responsável pode fornecer informações referentes às pessoas menores de 10 anos; ou pessoas que não fossem capazes de responder ao questionário.
- (viii) *Viagem* refere-se ao deslocamento de uma pessoa, por motivo específico, entre dois pontos determinados (origem e destino), utilizando, para isso, um ou mais modos de transporte. Sendo nominado como origem o local onde a pessoa entrevistada se encontrava quando iniciou o seu deslocamento, e como destino o local para onde a pessoa entrevistada se dirigiu (destino final).
- (ix) *Viagem a pé* é aquela realizada integralmente a pé, da origem ao destino. Além disso, só será contabilizada como viagem a pé se a distância percorrida é superior a 500 metros (ou cinco quadras) ou se o motivo da viagem (na origem ou no destino) é trabalho ou escola, independente da distância percorrida.
- (x) *Zona* de pesquisa é a unidade territorial de levantamento da origem e do destino das viagens

4.5 Limitações

Como este estudo baseia-se em dados secundários é preciso estar ciente das limitações que conceitos e metodologia de pesquisa adotados podem trazer. O conceito de família é bastante centrado na unidade do domicílio, o que pode desconsiderar laços afetivos e redes de solidariedade que as famílias ensejam, mesmo estando em domicílios separados. Por exemplo, uma criança pequena cujos pais precisam trabalhar, pode significar que vá haver viagens motivo escola, com um dos pais, mais provavelmente a mulher, servindo passageiro. Entretanto, a depender da oferta de serviços do local de residência, pode ser que não haja vaga em creche disponível. Pode ser ainda que a família não disponha de condições financeiras para pagar uma escola particular para essa criança. Um arranjo muitas vezes adotado é deixar a criança com avós ou tios que morem próximos. Isso representa impacto no padrão de mobilidade e também uma “economia” que o arranjo familiar proporciona. Esses arranjos e nuances pouco serão percebidos a partir destas bases de dados, pela forma com que foram construídas.

Outra limitação que merece atenção, é a hierarquia estabelecida entre os modos. Muito embora haja a descrição dos modos utilizados em uma viagem (até três em 1977 e 1987 e até quatro em 1997 e 2007), a duração da viagem disponível no banco de dados é a duração total, geralmente atribuída ao modo principal. Contudo, as viagens por modos não motorizados são as menos “fortes” na hierarquia de modos, sendo consideradas praticamente se forem exclusivas. Isso dificulta e às vezes impossibilita analisar devidamente os modos não motorizados dentro das cadeias de viagens. Ademais, existe uma subrepresentatividade das viagens a pé devido ao conceito adotado. E espera-se que estas viagens sejam importantes na descrição diferencial dos padrões de deslocamento de acordo com os gêneros. A mulher é responsável pela maior parte das tarefas ligadas à administração doméstica (ROOT; SCHINTLER, 1999; VANCE; IOVANNA, 2007), o que inclui compras rápidas e próximas à residência ou levar filhos(as) à escola (FOX, 1983; FAGNANI, 1983; JOHNSTON-ANUMONWO, 1992; MCGUCKIN; ZMUD; NAKAMOTO, 2005; SCHWANEN; DIJST; DILEMAN, 2002; LEE; MCDONALD, 2003; CRANE, 2007), muitas vezes, a pé (VASCONCELLOS, 2001).

Por fim, existem limitações advindas da natureza dos estudos longitudinais, como a adoção de diferentes definições e entendimentos para as mesmas palavras ou categorias, ou ainda, a variação no tempo da forma de perguntar a mesma informação, da seleção e treinamento de pesquisadores e dos critérios metodológicos adotados.

5 Tratamento dos Dados

5.1 Preparação da base de dados

A fim de tornar possível comparar dados das diversas Pesquisas OD objetos deste trabalho (1977, 1987, 1997 e 2007) foi necessário conhecer o banco de dados de cada edição, e para que seja possível analisar a evolução de padrões de mobilidade e de comportamento, é preciso que os dados dos diversos bancos sejam comparáveis. Para isso, foi desenvolvido o desenho de um Banco de Dados Unificado (BDU), com função de integrar e compatibilizar as informações julgadas relevantes até este momento, e também aquelas que acredita-se possam vir a ser úteis em etapas posteriores do trabalho. Os *layouts* dos bancos de dados originais podem ser observados no Anexo C, e o do banco integrador pode ser visto no Quadro 3 a seguir.

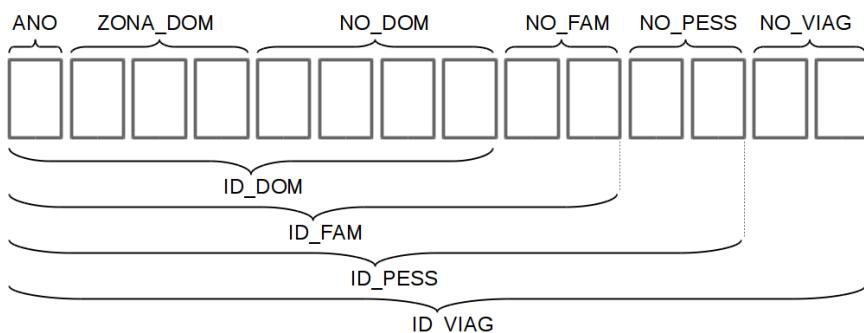
Inicialmente, nos bancos de dados originais, foram identificadas as variáveis correspondentes entre os diversos anos e o BDU. Para cada variável e para cada ano foram feitas transformações para que fosse possível gerar um único banco de dados final, possível de ser “empilhado” - essas transformações estão descritas neste capítulo. Devido ao tamanho dos bancos de dados, sua preparação não foi feita em planilhas eletrônicas convencionais, mas por meio de códigos em linguagem *python*. A estrutura dos códigos apresentados no Anexo H é dividida em 5 blocos: (i) *Set up* inicial com chamadas de bibliotecas e configurações; (ii) Definição de *loggers*; (iii) Definição de funções gerais; (iv) Definição da função principal; e (v) Chamada para execução da função principal.

- (i) *Set up*: foram utilizadas as bibliotecas *math*, para as funções matemáticas, e *pandas*, as análises de dados.
- (ii) Definição de *loggers*¹: foram estabelecidos dois *loggers*, o *log_output* que salva o conteúdo num arquivo de saída (de texto, com extensão .log) e o *log_tela*, que além de salvar o conteúdo no arquivo de saída também mostra esse conteúdo na tela.
- (iii) Definição de funções gerais: são definidas 5 funções gerais assessórias que serão utilizadas pelas funções gerais “passo”. Ou seja, as 66 funções gerais “passo” são as que, de fato, fazem o trabalho de transformação variável a variável. As assessórias servem para que estas possam ler os arquivos .csv e realizar testes de consistência, entre outras tarefas.

¹ Logger é uma rotina utilizada para acompanhar o processamento dos dados.

(iv) Definição da função principal: principia lendo o banco de dados original (arquivo tipo OD____.csv) e também outros arquivos auxiliares de dados estruturados, como o ucod____.csv, o setor_ativ____.csv e o coord_subzonas____.csv – sendo um arquivo desses para cada OD analisada. Logo, temos 4 arquivos .csv principais de entrada e 12 auxiliares. O arquivo ucod____.csv contém a relação de cada zona e sua respectiva UCOD, o arquivo setor_ativ____.csv contém a relação de código do setor de atividade do banco original e o respectivo código no banco unificado. O arquivo coord_subzonas____.csv contém as coordenadas dos centroides das subzonas extraídos a partir dos arquivos de mapas e que não constavam originalmente do banco de dados. No caso de 1977, não se conseguiu o arquivo do MapInfo com a granularidade das subzonas, portanto, foram utilizadas as coordenadas dos centroides das zonas. Na sequência, as colunas que não existiam originalmente no banco de dados são geradas e reordenadas. Existe uma variável de controle (“passo”) que a cada passo executado incrementa 1. São chamadas as funções gerais. Vale notar que ficaram para o final aquelas relacionadas à geração dos IDs, variáveis fundamentais para indexação correta do BDU. Assim, primeiro deve-se executar a função NO_DOM (número do domicílio) e depois a ID_DOM (identificador de domicílio). Analogamente isso ocorre com NO_FAM e ID_FAM (sobre as famílias), NO_PESS e ID_PESS (sobre as pessoas) e NO_VIAG e ID_VIAG (sobre as viagens). Não havia expectativa inicial de que seria necessário gerar essa indexação, porém, a falta de um manual que explicasse a composição dos IDs presentes nos bancos de dados originais inviabilizou sua utilização. A vantagem de gerar nova indexação (ver Figura 6) é que se tem maior controle do seu significado, possibilitando ainda incorporar informações como o ANO, informação não considerada nas bases originais.

Figura 6 – Esquema indicativo da formação dos IDs



(v) Chamada da função principal: é a parte mais enxuta em termos de código, pois trata apenas da chamada da função principal (*main*), que por sua vez chama as funções gerais.

Vale destacar que, todo esse tratamento de dados visou a compatibilização de categorias entre as diversas Pesquisas OD, que às vezes não continham as mesmas variáveis, ou ainda, não continham as mesmas categorias para uma determinada variável. Um dos princípios adotados para nova categorização foi agregar para não perder informação. Por exemplo, se em 1977 havia as duas categorias “sem instrução” e “primário incompleto” e, em 1987, havia a categoria “não alfabetizado/4^a série incompleta”, optou-se por unir as duas categorias de 1977 numa única denominada “não alfabetizado / fundamental incompleto”. Como exemplo, ainda em grau de instrução, do ponto de vista prático houve alterações significativas como: nomenclatura utilizada para os ciclos escolares, formas de ingresso (houve época em que havia um “vestibulinho” em algumas escolas para ingressar no ginásio) e quantidade de anos de cada ciclo. Além disso, os significados sociais atribuídos à “mesma variável” grau de instrução são diferentes ao longo do tempo: ter curso superior no fim da década de 1970 era diferente e muito mais valorado socialmente do que em 2007. Ou seja, foi feito esforço para minimizar a perda de informações, contudo, algumas limitações são inerentes aos estudos longitudinais e informações são inevitavelmente perdidas, seja pela agregação do dado, seja pela descontinuidade na disponibilização dessa informação.

Foi feito um **teste de consistência** que consistiu em identificar, para um mesmo indivíduo, que realizou viagem, se a zona de destino da viagem i é igual à zona de origem da viagem $i+1$. Inicialmente, foram encontradas 548 observações em 1977² que não passaram nesse crivo. Foi feita análise mais minuciosa desses considerando as variáveis de idade, situação familiar, renda individual, horário de saída e horário de chegada. Observou-se que eram casos de linhas i invertidas com as subsequentes, portanto, foram feitas as inversões de linhas e corrigidos 546 registros. Os 2 registros remanescentes foram corrigidos manualmente. Todos os registros de 1987 passaram no teste de consistência. Em 1997, 7 registros não passaram no teste de consistência e, após análise mais cuidadosa, parecia tratar-se de linhas duplicadas, logo, esses 7 registros foram excluídos. Situação semelhante ocorreu com 1 registro de 2007, que também foi excluído.

O Quadro 3 apresenta as variáveis do BDU, suas descrições e também tipificação, onde “Q” indica tratar-se de variável qualitativa, “M”, de variável métrica (quantitativa), “D”, de variável *dummy*, e “ID” variável de indexação (natureza texto).

² É preciso considerar que a base de dados 1977 é bastante antiga, rodada originalmente em computadores de grande porte numa época em que não havia computadores pessoais, e que passou por muitas migrações e manipulações até hoje.

Quadro 3 – *Layout* do Banco de Dados Unificado

Tipo	Variável	Descrição	Qtde. Díg.	Códigos, Categorias e Faixas de Valores Válidas
Q	ANO	Ano de referência da Pesquisa OD	01	1 - OD-1977 2 - OD-1987 3 - OD-1997 4 - OD-2007
D	CD_ENTRE	Código de entrevista	01	0 - Completa sem viagem ou incompleta 1 - Completa com viagem
Q	DIA_SEM	Dia da Semana	01	2 - Segunda-Feira 3 - Terça-Feira 4 - Quarta-Feira 5 - Quinta-Feira 6 - Sexta-Feira
Q	UCOD_DOM	Unidade de Correspondência de Pesquisas OD para domicílio	02	1 a 67
Q	ZONA_DOM	Zona do domicílio da OD original	03	1 a 243 em 1977 1 a 254 em 1987 1 a 389 em 1997 1 a 460 em 2007
Q	SUBZONA_DOM	Subzona do domicílio da OD original	03	1 a 633 em 1977 1 a 9 em 1987 1 a 9 em 1997 não consta em 2007

Fonte: Elaboração própria a partir das OD-1977, OD-1987, OD-1997 e OD-2007

Tipo	Variável	Descrição	Qtde. Díg.	Códigos, Categorias e Faixas de Valores Válidas
Q	MUN_DOM	Município do domicílio	02	1 a 27 em 1977 1 a 38 em 1987 1 a 39 em 1997 1 a 39 em 2007
M	CO_DOM_X	Coordenada X do domicílio	12	12 dígitos, 2 casas decimais
M	CO_DOM_Y	Coordenada Y do domicílio	12	12 dígitos, 2 casas decimais
ID	ID_DOM	Identifica o domicílio	08	Composição de ID com ano, zona e domicílio
D	F_DOM	Identifica o primeiro registro do domicílio	01	0 - Demais registros 1 - Primeiro registro
M	FE_DOM	Fator de expansão do domicílio	10	10 dígitos, 2 casas decimais
M	NO_DOM	Número do domicílio	04	04 dígitos, número inteiro
D	TIPO_DOM	Tipo do domicílio	01	0 - coletivo 1 - particular
M	TOT_FAM	Total de famílias no domicílio	02	02 dígitos, número inteiro
ID	ID_FAM	Identifica a família	10	Composição de ID com ano, zona, domicílio e família
D	F_FAM	Identifica primeiro registro da família	01	0 - Demais registros 1 - Primeiro registro
M	FE_FAM	Fator de expansão da família	10	10 dígitos, 2 casas decimais
M	NO_FAM	Número da família	02	02 dígitos, número inteiro
Q	COND_MORA	Condição de moradia	01	1 - alugada 2 - própria 3 - outros
M	QT_AUTO	Quantidade de automóveis	01	01 dígito, número inteiro
M	QT_BICI	Quantidade de bicicletas	01	01 dígito, número inteiro
M	QT_MOTO	Quantidade de motocicletas	01	01 dígito, número inteiro
Q	CD_RENFAM	Código de renda familiar	01	0 - Renda declarada como zero 1 - Renda declarada maior do que zero 2 - Renda atribuída
M	REN_FAM	Renda familiar	08	08 dígitos, 2 casas decimais (R\$/mês, ref. out/2007)

Fonte: Elaboração própria a partir das OD-1977, OD-1987, OD-1997 e OD-2007

Tipo	Variável	Descrição	Qtde. Díg.	Códigos, Categorias e Faixas de Valores Válidas
ID	ID_PESS	Identifica a pessoa	12	Composição de ID com ano, zona, domicílio, família e pessoa
D	F_PESS	Identifica o primeiro registro da pessoa	01	0 - Demais registros 1 - Primeiro registro
M	FE_PESS	Fator de expansão da pessoa	10	10 dígitos, 2 casas decimais
M	NO_PESS	Número da pessoa	2	02 dígitos, número inteiro
Q	SIT_FAM	Situação familiar	01	1 - Pessoa responsável 2 - Cônjuge/Companheiro(a) 3 - Filho(a)/Enteado(a) 4 - Outro parente / agregado 5 - Empregado residente 6 - Outros
M	IDADE	Idade	02	02 dígitos, número inteiro
D	SEXO	Sexo	01	0 - Masculino 1 - Feminino
D	ESTUDA	A pessoa estuda atualmente?	01	0 - Não 1 - Sim
Q	GRAU_INSTR	Grau de instrução da pessoa	01	1 - Não alfabetizado / Fundamental incompleto 2 - Fundamental completo / Médio incompleto 3 - Médio completo / Superior incompleto 4 - Superior completo
Q	OCUP	Condição de ocupação da pessoa	01	1 - Tem trabalho 2 - Em licença 3 - Aposentado(a) / Pensionista 4 - Desempregado(a) 5 - Sem ocupação 6 - Dono(a) de casa 7 - Estudante

Fonte: Elaboração própria a partir das OD-1977, OD-1987, OD-1997 e OD-2007

Tipo	Variável	Descrição	Qtde. Díg.	Códigos, Categorias e Faixas de Valores Válidas
Q	SETOR_ATIV	Setor de atividade (do 1º trabalho)	01	1 - Agrícola 2 - Construção Civil 3 - Indústria 4 - Comércio 5 - Administração Pública 6 - Serviços de Transporte 7 - Outros Serviços 8 - Outros 9 - Não se aplica
D	CD_RENIND	Condição de Renda Individual	01	0 - Não tem renda 1 - Tem renda
M	REN_IND	Valor da Renda Individual	08	08 dígitos, 2 casas decimais (R\$/mês, ref. out/2007)
Q	UCOD_ESC	Unidade de Correspondência de Pesquisas OD para escola	02	1 a 67
Q	ZONA_ESC	Zona da escola da OD original	03	1 a 243 em 1977 1 a 254 em 1987 1 a 389 em 1997 1 a 460 em 2007
Q	SUBZONA_ESC	Subzona da escola da OD original	03	1 a 633 em 1977 1 a 9 em 1987 1 a 9 em 1997 não consta em 2007
Q	MUN_ESC	Município da escola	02	1 a 27 em 1977 1 a 39 em 1987 1 a 39 em 1997 1 a 39 em 2007
M	CO_ESC_X	Coordenada X da escola	12	12 dígitos, 2 casas decimais
M	CO_ESC_Y	Coordenada Y da escola	12	12 dígitos, 2 casas decimais
Q	UCOD_TRAB1	Unidade de Correspondência de Pesquisas OD para trabalho 1	02	1 a 67

Fonte: Elaboração própria a partir das OD-1977, OD-1987, OD-1997 e OD-2007

Tipo	Variável	Descrição	Qtde. Díg.	Códigos, Categorias e Faixas de Valores Válidas
Q	ZONA_TRAB1	Zona do trabalho 1 da OD original	03	1 a 243 em 1977 1 a 254 em 1987 1 a 389 em 1997 1 a 460 em 2007
Q	SUBZONA_TRAB1	Subzona do trabalho 1 da OD original	03	1 a 633 em 1977 1 a 9 em 1987 1 a 9 em 1997 não consta em 2007
Q	MUN_TRAB1	Município do trabalho 1	02	1 a 27 em 1977 1 a 39 em 1987 1 a 39 em 1997 1 a 39 em 2007
M	CO_TRAB1_X	Coordenada X do trabalho 1	12	12 dígitos, 2 casas decimais
M	CO_TRAB1_Y	Coordenada Y do trabalho 1	12	12 dígitos, 2 casas decimais
Q	UCOD_TRAB2	Unidade de Correspondência de Pesquisas OD para trabalho 2	02	1 a 67
Q	ZONA_TRAB2	Zona do trabalho 2 da OD original	03	1 a 243 em 1977 1 a 254 em 1987 1 a 389 em 1997 1 a 460 em 2007
Q	SUBZONA_TRAB2	Subzona do trabalho 2 da OD original	03	1 a 633 em 1977 1 a 9 em 1987 1 a 9 em 1997 não consta em 2007
Q	MUN_TRAB2	Município do trabalho 2	02	1 a 27 em 1977 1 a 39 em 1987 1 a 39 em 1997 1 a 39 em 2007
M	CO_TRAB2_X	Coordenada X do trabalho 2	12	12 dígitos, 2 casas decimais
M	CO_TRAB2_Y	Coordenada Y do trabalho 2	12	12 dígitos, 2 casas decimais

Fonte: Elaboração própria a partir das OD-1977, OD-1987, OD-1997 e OD-2007

Tipo	Variável	Descrição	Qtd. Díg.	Códigos, Categorias e Faixas de Valores Válidas
ID	ID_VIAG	Identifica a viagem	14	Composição de ID com ano, zona, domicílio, família, pessoa e viagem
D	F_VIAG	Identificador de viagem	01	0 - Não fez viagem 1 - Fez viagem
M	FE_VIAG	Fator de expansão da viagem	10	10 dígitos, 2 casas decimais
M	NO_VIAG	Número da viagem	02	02 dígitos, número inteiro
M	TOT_VIAG	Total de viagens da pessoa	02	02 dígitos, número inteiro
Q	UCOD_ORIG	Unidade de Correspondência de Pesquisas OD para origem	02	1 a 67
Q	ZONA_ORIG	Zona de origem da viagem da OD original	03	1 a 243 em 1977 1 a 254 em 1987 1 a 389 em 1997 1 a 460 em 2007
Q	SUBZONA_ORIG	Subzona de origem da viagem da OD original	03	1 a 633 em 1977 1 a 9 em 1987 1 a 9 em 1997 não consta em 2007
Q	MUN_ORIG	Município de origem da viagem	02	1 a 27 em 1977 1 a 39 em 1987 1 a 39 em 1997 1 a 39 em 2007
M	CO_ORIG_X	Coordenada X da origem	12	12 dígitos, 2 casas decimais
M	CO_ORIG_Y	Coordenada Y da origem	12	12 dígitos, 2 casas decimais
Q	UCOD_DEST	Unidade de Correspondência de Pesquisas OD para destino	02	1 a 67
Q	ZONA_DEST	Zona de destino da viagem da OD original	03	1 a 243 em 1977 1 a 254 em 1987 1 a 389 em 1997 1 a 460 em 2007
Q	SUBZONA_DEST	Subzona de destino da viagem da OD original	03	1 a 633 em 1977 1 a 9 em 1987 1 a 9 em 1997 não consta em 2007

Fonte: Elaboração própria a partir das OD-1977, OD-1987, OD-1997 e OD-2007

Tipo	Variável	Descrição	Qtde. Díg.	Códigos, Categorias e Faixas de Valores Válidas
Q	MUN_DEST	Município de destino da viagem	02	1 a 27 em 1977 1 a 39 em 1987 1 a 39 em 1997 1 a 39 em 2007
M	CO_DEST_X	Coordenada X do destino	12	12 dígitos, 2 casas decimais
M	CO_DEST_Y	Coordenada Y do destino	12	12 dígitos, 2 casas decimais
M	DIST_VIAG	Distância da viagem (m)	08	08 dígitos, 2 casas decimais
D	SERV_PAS_ORIG	Serve passageiro na origem?	01	0 - Não 1 - Sim
D	SERV_PAS_DEST	Serve passageiro no destino?	01	0 - Não 1 - Sim
Q	MOTIVO_ORIG	Motivo na origem	01	1 - Trabalho (indústria) 2 - Trabalho (comércio) 3 - Trabalho (serviços) 4 - Educação 5 - Compras 6 - Saúde 7 - Lazer 8 - Residência 9 - Outros
Q	MOTIVO_DEST	Motivo no destino	02	idem Motivo na origem
Q	MODO1	Modo 1	2	1 - Ônibus de linha 2 - Ônibus escolar / empresa 3 - Dirigindo automóvel 4 - Passageiro de automóvel 5 - Táxi 6 - Lotação / van 7 - Metrô 8 - Trem 9 - Motocicleta 10 - Bicicleta 11 - A pé 12 - Outros

Fonte: Elaboração própria a partir das OD-1977, OD-1987, OD-1997 e OD-2007

Tipo	Variável	Descrição	Qtde. Díg.	Códigos, Categorias e Faixas de Valores Válidas
Q	MODO2	Modo 2	02	idem Modo 1
Q	MODO3	Modo 3	02	idem Modo 1
Q	MODO4	Modo 4	02	idem Modo 1
Q	MODO_PRIN	Modo Principal	02	idem Modo 1
Q	TIPO_VIAG	Tipo de viagem	01	1 - Coletivo 2 - Individual 3 - A pé
M	H_SAIDA	Hora de saída	02	Hora de saída
M	MIN_SAIDA	Minuto de saída	02	Minutos de saída
M	ANDA_ORIG	Tempo andando na origem	02	Tempo andando na origem (minutos)
M	H_CHEG	Hora de chegada	02	Hora de chegada
M	MIN_CHEG	Minuto de chegada	02	Minutos de chegada
M	ANDA_DEST	Tempo andando no destino	02	Tempo andando no destino (minutos)
M	DURACAO	Duração da viagem	02	Duração da viagem (minutos)
Q	TIPO_EST_AUTO	Tipo de estacionamento	01	0 - não estacionou 1 - estacionou em local privado (particular avulso ou mensal, próprio ou patrocinado) 2 - estacionou em local público (na rua)
M	VALOR_EST	Valor do estacionamento	02	06 dígitos, 2 casas decimais (R\$/mês, ref. out/2007)

Fonte: Elaboração própria a partir das OD-1977, OD-1987, OD-1997 e OD-2007

5.2 Critérios de Validação

Algumas variáveis (colunas) inicialmente não identificadas como de interesse para este trabalho foram descartadas dos bancos originais. Dado o interesse em investigar o comportamento de homens e mulheres nos deslocamentos diários, o critério utilizado para a exclusão de observações (linhas) foi a variável “SEXO” ser igual a 0 na base de dados. Assim, após a preparação da base e da execução de testes de consistência, foram excluídas 1561 linhas - todas pertencentes a 1977 e cujo fator de expansão de viagens também era igual a 0. O impacto na quantidade de domicílios, pessoas e viagens (registros) está

expresso na Tabela 6.

Tabela 6 – Quantidade de viagens e de domicílios, por ano, antes e depois da preparação do BDU

Ano	Número de Domicílios		Número de Pessoas		Número de Viagens	
	Antes	Depois	Antes	Depois	Antes	Depois
1977	26.132	24.613	108.069	108.028	230.606	229.045
1987	26.070	26.070	110.813	110.813	223.926	223.926
1997	23.841	23.841	98.780	98.780	199.647	199.647
2007	29.957	29.957	91.405	91.405	196.698	196.698
Total	106.000	104.481	409.067	409.025	850.877	849.316

Fonte: Compilação a partir de [Metrô-SP \(1977, 1987, 1997, 2007\)](#)

Os critérios para a adoção de “NA” foram os seguintes:

- (i) Quando em um ano específico não havia informação levantada para uma determinada variável.
- (ii) Quando, mesmo essa variável tendo sido levantada, não houve preenchimento ou resposta.
- (iii) Quando uma determinada informação não existe pelo fato de que não foi feita viagem.

5.3 Formulação de Novas Variáveis

Foram criadas variáveis de interesse, relativas a viagens, pessoas ou famílias, para gerar informações necessárias para subsidiar as análises subsequentes. Além dessas, também foram criadas 4 *dummies* de ano (ANO_77, ANO_87, ANO_97 e ANO_97) para que fosse possível, nas regressões, tentar captar o efeito do tempo.

5.3.1 Variáveis relativas às viagens

- **Faixa horária e *dummies* de período:** A variável FAIXA_HORARIA foi criada para classificar as viagens em faixas horárias, considerando o horário de término da viagem obtido a partir das variáveis H_CHEG (hora de chegada) e MIN_CHEG (minuto de chegada). Isso porque o horário de chegada corresponde ao horário de início da atividade que motivou a viagem, e são as atividades que, de fato, interessam e motivam o comportamento humano. Foram adotadas sete categorias para a divisão dos períodos do dia, derivados do estudo de [Vespucci \(2003\)](#). Para cada faixa horária, foi gerada uma *dummy* de período correspondente. Tal divisão, também adotada

por Germani (2005), foi a que melhor refletiu as concentrações de atividades e pode ser observada no Quadro 4.

Quadro 4 – *Dummies* de período

Categoría	Descrição	<i>Dummy</i> resultante
0	não fez viagem	-
1	madrugada (entre 0h01 e 5h00)	VIAG_PER_MADRUG
2	começo da manhã (entre 5h01 e 9h00)	VIAG_PER_COM_MAN
3	manhã (entre 9h01 e 12h)	VIAG_PER_MANHA
4	meio-dia tarde (entre 12h01 e 14h)	VIAG_PER_MEIODIA
5	tarde (entre 14h01 e 17h)	VIAG_PER_TARDE
6	começo da noite (17h01 e 22h)	VIAG_PER_COM NOI
7	noite (22h01 e 0h)	VIAG_PER_NOITE

- **Contagem da utilização dos modos:** Como cada viagem pode comportar a utilização de até 4 modos³ é feita a contagem de quantas vezes um determinado modo foi utilizado em uma viagem. Quem não fez viagem terá toda as variáveis listadas a seguir definidas como iguais a 0. A consolidação da quantidade de modos diferentes utilizados na viagem é summarizado na variável VIAG_NO_MODOS. O Quadro 5 apresenta estas variáveis criadas.

Quadro 5 – Variáveis de contagem da utilização dos modos

Variável	Descrição
VIAG_MODO_ONIBUS	viagens feitas por ônibus de linha (municipal ou intermunicipal), ônibus escolar, ônibus de empresa, lotação e van
VIAG_MODO_DIRIG	viagens feitas por automóvel com a pessoa dirigindo
VIAG_MODO_PASS	viagens feitas por automóvel com a pessoa como passageiro, incluindo as viagens de táxi
VIAG_MODO_TREM	viagens feitas por metrô ou trem
VIAG_MODO_MOTO	viagens feitas por motocicleta
VIAG_MODO_BICI	viagens feitas por bicicleta
VIAG_MODO_APE	viagens feitas a pé
VIAG_MODO_OUTROS	viagens não contempladas pelas categorias anteriores
VIAG_NO_MODOS	quantos modos diferentes foram utilizados, por viagem, de acordo com estas categorias

- **Dummies dos motivos de viagem:** Cada viagem só tem um motivo na origem e um motivo no destino. Foram considerados os motivos do destino (que relacionam-se

³ As pesquisas OD-77 e OD-87 previam campo para até 3 modos, já as pesquisas OD-97 e OD-07 previam a utilização de até 4 modos por viagem.

com a atividade fim daquele deslocamento) e criados seis marcadores, conforme pode ser observado a seguir. [Vespucci⁴](#) (2003) e [Germani⁵](#) (2005) agregam as viagens denominadas de manutenção (como compras e saúde) com as de lazer e outros, além de desagregar as viagens de motivo residência em temporária e final. [Dalmaso⁶](#) (2009) verificou que os resultados desta categorização eram muito parecidos. Além disso, este autor constatou que havia grandes diferenças nos resultados mantendo o motivo manutenção único ou desmembrando-o em compras/saúde e lazer/outros. Por fim, ele considerou o motivo “servir passageiro”, que significa que a pessoa ao fazer uma viagem para acompanhar alguém (à escola, ao médico, etc.) não faz essa viagem por um motivo seu, mas acompanhando o motivo da outra pessoa, a quem “serve”. Optou-se por seguir a divisão de categorias de motivos de [Dalmaso](#) (unificação de motivo residência, separação do motivo manutenção e explicitação do motivo servir passageiro) porque a revisão de literatura indica que as mulheres são as principais cuidadoras do núcleo familiar, assim, espera-se que apresentem maior frequência nos motivos “servir passageiro” e, talvez, no “manutenção/compras” do que os homens. O Quadro 6 indica as *dummies* de motivo criadas.

Quadro 6 – *Dummies* de motivos de viagem

<i>Dummy</i>	Descrição
VIAG_MOTIVO_SERV_PAS	indica se o motivo da viagem foi servir passageiro
VIAG_MOTIVO_TRAB	indica se o motivo da viagem foi trabalho
VIAG_MOTIVO_EDUC	indica se o motivo da viagem foi educação
VIAG_MOTIVO_RES	indica se o motivo da viagem foi residência
VIAG_MOTIVO_MANUT_COMPRAS	indica se o motivo da viagem foi manutenção da casa (compras) ou da família (saúde)
VIAG_MOTIVO_LAZER_OUTROS	indica se o motivo da viagem foi lazer ou outros

⁴ Fez estudos sobre a sequência diária de atividades e cadeias de viagens a partir das pesquisas OD-87 e OD-97.

⁵ Fez estudos sobre comportamento de demanda usando método de alinhamento de sequências multidimensionais a partir da pesquisa OD-97.

⁶ Fez estudos sobre a identificação e caracterização de grupos de indivíduos segundo padrões de sequências de atividades multidimensionais a partir da pesquisa OD-97.

5.3.2 Variáveis relativas às pessoas

- **Faixa etária e *dummies* de faixa etária:** A partir da variável IDADE, foi criada a variável FAIXA_ETARIA, seguindo os critérios adotados pelo IBGE, resultando em 21 faixas etárias, conforme pode ser observado a seguir. Para cada faixa etária, foi gerada uma *dummy* correspondente, que pode ser observada no Quadro 7.

Quadro 7 – *Dummies* de faixa etária

Categoría	Descrição	<i>Dummy</i> resultante
0	0 a 4 anos	FX_ET_0
1	5 a 9 anos	FX_ET_1
2	10 a 14 anos	FX_ET_2
3	15 a 19 anos	FX_ET_3
4	20 a 24 anos	FX_ET_4
5	25 a 29 anos	FX_ET_5
6	30 a 34 anos	FX_ET_6
7	35 a 39 anos	FX_ET_7
8	40 a 44 anos	FX_ET_8
9	45 a 49 anos	FX_ET_9
10	50 a 54 anos	FX_ET_10
11	55 a 59 anos	FX_ET_11
12	60 a 64 anos	FX_ET_12
13	65 a 69 anos	FX_ET_13
14	70 a 74 anos	FX_ET_14
15	75 a 79 anos	FX_ET_15
16	80 a 84 anos	FX_ET_16
17	85 a 89 anos	FX_ET_17
18	90 a 94 anos	FX_ET_18
19	95 a 99 anos	FX_ET_19
20	mais de 100 anos	FX_ET_20

- **Faixa de renda individual:** Os valores de renda individual foram levados para out/2007 e classificados em sete categorias explicitadas no Quadro 8, conforme a quantidade de salários mínimos (SM)⁷.

⁷ O valor de um salário mínimo era R\$380,00 em 2007 segundo a Lei do salário mínimo vigente e disponível em <https://www.planalto.gov.br/ccivil_03/_Ato2007-2010/2007/Lei/L11498.htm> Acesso em 30 de setembro de 2015.

Quadro 8 – Categorias da variável Faixa de Renda Individual

Categoría	em R\$	em Salário Mínimo
0	0	sem renda
1	até R\$380 (exclusive)	até 1 SM
2	de R\$380 (inclusive) até R\$760 (exclusive)	de 1 a 2 SM
3	de R\$760 (inclusive) até R\$1140 (exclusive)	de 2 a 3 SM
4	de R\$1140 (inclusive) até R\$1900 (exclusive)	de 3 a 5 SM
5	de R\$1900 (inclusive) até R\$3800 (exclusive)	de 5 a 10 SM
6	de R\$3800 (inclusive) até R\$5700 (exclusive)	de 10 a 15 SM
7	mais de R\$5700 (inclusive)	mais de 15 SM

- **Contagem da utilização dos modos:** Baseadas nas variáveis de contagem de modo foi calculada a variável PESS_NO_MODOS, que sintetiza a quantidade de modos diferentes que a pessoa utilizou ao longo do dia, segundo a Equação (5.1), onde K é o número de categorias dos modos.

$$PESS_NO_MODOS = \sum_{k=1}^K USO_MODO_k, \text{ onde :} \\ \begin{cases} USO_MODO_k = 0 & \text{se } VIAG_MODO_k = 0 \\ USO_MODO_k = 1 & \text{se } VIAG_MODO_k \neq 0 \end{cases} \quad (5.1)$$

- **Contagem dos motivos de viagem:** Baseadas nas *dummies* dos motivos das viagens, foram feitos agrupamentos por pessoas, e somadas a quantidade de vezes que um determinado motivo foi declarado para cada pessoa - ver Equação (5.2), onde n é o número de viagens da pessoa. A variável PESS_NO_MOTIVOS sintetiza a quantidade de motivos diferentes que levou a pessoa a se deslocar ao longo do dia.

$$PESS_NO_MOTIVOS = \sum_{i=1}^n VIAG_MOTIVO_i \quad (5.2)$$

- **Distância total:** A informação das distâncias percorridas por viagem já está armazenada na variável DIST_VIAG do BDU. A variável PESS_DIST_TOT sumariza essa informação para a pessoa, conforme Equação (5.3).

$$PESS_DIST_TOT = \sum_{i=1}^n DIST_VIAG_i \quad (5.3)$$

- **Distância média:** A distância média percorrida pela pessoa é a distância total da pessoa dividida pelo seu número total de viagens - ver Equação (5.4).

$$PESS_DIST_MED = \frac{\sum_{i=1}^n DIST_VIAG_i}{TOT_VIAG} \quad (5.4)$$

- **Duração total:** A informação das distâncias percorridas por viagem já está armazenada na variável DURACAO do BDU. A variável PESS_DURACAO_TOT sumariza essa informação para a pessoa, conforme Equação (5.5).

$$PESS_DURACAO_TOT = \sum_{i=1}^n DURACAO_i \quad (5.5)$$

- **Duração média:** A duração média percorrida pela pessoa é a duração total da pessoa dividida pelo seu número total de viagens - ver Equação (5.6).

$$PESS_DURACAO_MED = \frac{\sum_{i=1}^n DURACAO_i}{TOT_VIAG} \quad (5.6)$$

5.3.3 Variáveis relativas às famílias

- **Faixa de renda familiar:** A partir da variável renda familiar, em valores de outubro de 2007, foi criada a variável FAIXA_REN_FAM, que contém a faixa de renda familiar segundo o critério de classificação do IBGE para classes econômicas⁸ (A, B, C, D e E) – ver Quadro 9. Portanto, as faixas de renda familiar foram construídas a partir da quantidade de salários mínimos, sendo o valor de um salário mínimo R\$380,00 em 2007⁹.

Quadro 9 – Categorias da variável Faixa de Renda Familiar

Categoria	em R\$	Classe Econômica
0	0	sem renda
1	até R\$760 (exclusive)	classe E
2	de R\$760 (inclusive) até R\$1520 (exclusive)	classe D
3	de R\$1520 (inclusive) até R\$3800 (exclusive)	classe C
4	de R\$3800 (inclusive) até R\$7600 (exclusive)	classe B
5	mais de R\$7600 (inclusive)	classe A

⁸ Definição de Classe Econômica – Fonte: <http://www.sae.gov.br/wp-content/uploads/ebook_ClasseMedia1.pdf> Acesso em 30 de setembro de 2015.

⁹ Lei do salário mínimo de 2007: <https://www.planalto.gov.br/ccivil_03/_Ato2007-2010/2007/Lei/L11498.htm> Acesso em 30 de setembro de 2015.

- **Tamanho da família:** O tamanho da família (variável TOT_PESS) é determinado pelo máximo número da pessoa (NO_PESS) de uma determinada família.
- **Número total de viagens da família:** A variável FAM_VIAG_TOT é calculada como a soma do número de viagens de cada pessoa da família - ver Equação (5.7), onde p é o número de pessoas da família.

$$FAM_VIAG_TOT = \sum_{j=1}^p TOT_VIAG_j \quad (5.7)$$

- **Distância total:** A informação das distâncias percorridas por viagem já está armazenada na variável DIST_VIAG do BDU. A variável FAM_DIST_TOT sumariza essa informação para a família, conforme Equação (5.8), onde p é o número de pessoas da família.

$$FAM_DIST_TOT = \sum_{j=1}^p PESS_DIST_TOT_j \quad (5.8)$$

- **Distância média:** A distância média de viagem da família, calculada como a soma das distâncias das pessoas dividida pelo seu total de viagens da família - ver Equação (5.9), onde p é o número de pessoas da família.

$$FAM_DIST_MED = \frac{\sum_{j=1}^p PESS_DIST_TOT_j}{FAM_VIAG_TOT} \quad (5.9)$$

- **Duração total:** A duração total das viagens da família são armazenadas na variável FAM_DURACAO_TOT, que sumariza essa informação para a família, conforme Equação (5.10), onde p é o número de pessoas da família.

$$FAM_DURACAO_TOT = \sum_{j=1}^p PESS_DURACAO_TOT_j \quad (5.10)$$

- **Duração média:** A duração média de uma viagem da família é armazenada na variável FAM_DURACAO_MED, que soma a duração total das pessoas da família e divide pelo seu número total de viagens da família - ver Equação (5.11), onde p é o número de pessoas da família.

$$FAM_DURACAO_MED = \frac{\sum_{j=1}^p PESS_DURACAO_TOT_j}{FAM_VIAG_TOT} \quad (5.11)$$

- **Presença de automóveis na família:** Foi gerada uma *dummy* (PRESENCA_AUTO) para indicar a presença de automóvel na família. Segundo Strambi e Van de Bilt (2001 apud PEIXOTO, 2002) a motorização é um fator de grande influência sobre o padrão de mobilidade. Peixoto (2002), ao estudar a evolução temporal da mobilidade na Região Metropolitana de Porto Alegre, entre 1986 e 1997, detectou ser a posse de veículos uma característica importante para segmentação de grupos de comportamento semelhantes. Geralmente, a utilização do automóvel é decidida dentro de uma determinada dinâmica familiar. No contexto familiar surgem questões como: em não existindo um automóvel para cada pessoa habilitada, quem tem a prioridade no uso? Quem trabalha mais longe de casa? Quem tem uma rotina mais complexa? Quem leva os filhos na escola? Quem é autônomo?
- **Presença de crianças na família:** Como um indicativo do estágio no ciclo de vida familiar (ORTÚZAR; WILLUMSEN, 1994), foram geradas *dummies* para indicar, na família, a presença de crianças de três faixas etárias, conforme Quadro 10.

Quadro 10 – *Dummies* de presença de crianças na família

<i>Dummy</i>	Descrição
PRESENCA_FILH_ate4	indica se a família conta com crianças até 4 anos
PRESENCA_FILH_5a9	indica se a família conta com crianças entre 5 e 9 anos
PRESENCA_FILH_10a14	indica se a família conta com crianças entre 10 e 14 anos

- **Presença de idosos na família:** A presença de idoso na família, também é um indicativo de estágio no ciclo de vida familiar (ORTÚZAR; WILLUMSEN, 1994), tanto mais expressivo quanto mais a população envelhece. Oliveira (2014) ao estudar a correlação e efeitos das composições familiares na mobilidade do idoso, considerando aspectos econômicos e sociais, constatou que o tamanho da família afeta negativamente a mobilidade dos idosos e que quanto maior o número de idosos na mesma família, maior sua mobilidade. Vasconcellos (2001) aponta que aspectos econômicos e sociais influenciam os padrões de mobilidade das pessoas e destaca, dentre os sociais, a família como um dos fatores importantes na tomada de decisões em relação aos deslocamentos e arranjos das atividades diárias. Desta forma, foi criada a *dummy* PRESENCA_IDOSO_60 para indicar se a família conta com pessoa idosa com mais de 60 anos.

6 Análises, Resultados e Discussões

6.1 Estatísticas Descritivas

A seguir serão apresentadas algumas estatísticas descritivas das variáveis, bem como outras observações julgadas pertinentes.

A variável **ANO**, de natureza qualitativa, não possui “NA” e conta com quatro valores únicos, cujas frequências absolutas são apresentadas na Tabela 7. Esta variável não era original dos bancos de dados e foi inserida para poder gerar *dummies* de marcação da *cross-section* observada para análises de caráter longitudinal.

Tabela 7 – Estatísticas da variável “ANO”

Categoria	Frequência Absoluta
1 - 1977	229.046
2 - 1987	223.926
3 - 1997	199.640
4 - 2007	196.698

A variável **DIA_SEM**, de natureza qualitativa, conta com cinco valores únicos, cujas frequências absolutas são apresentada na Tabela 8. Ela possui 296.893 registros “NA”, sendo 229.046 de 1977, ano em que esta variável não foi levantada. Os demais *missing values* foram adotados para o caso em que os valores eram originalmente iguais a 0, dividindo-se entre os anos de 1987 e 2007. Nas transformações efetuadas na preparação da base de dados, em 1987, foram feitas duas correções porque pressupôs-se erro de preenchimento: (i) 187 casos com registro 1 (valor que não consta nos layouts oficiais) que foram transformados para 2 (segunda-feira); e (ii) 170 casos de registro 7 (valor que não consta nos layouts oficiais) que foram transformados para 6 (sexta-feira). Vale observar que a sexta-feira (**DIA_SEM=6**) é superrepresentado devido à metodologia de coleta utilizada: na pesquisa, domiciliar, pregunta-se ao(à) respondente sobre as viagens que fizera no dia anterior e sábado é o dia em que mais se encontram as pessoas em casa, assim, há mais respostas sobre a sexta-feira do que sobre os demais dias da semana.

Tabela 8 – Estatísticas da variável “DIA_SEM”

ANO	1977	1987	1997	2007	Total
DIA_SEM=2	0	3.379	4.516	3.325	11.220
DIA_SEM=3	0	4.121	4.938	3.078	12.137
DIA_SEM=4	0	4.151	5.043	3.444	12.638
DIA_SEM=5	0	3.848	4.411	3.635	11.894
DIA_SEM=6	0	6.475	7.937	8.538	22.950

As variáveis **UCOD** (domicílio, escola, trabalhos, origem e destino), qualitativas, estabelecem correspondência com as zonas conforme Anexo A.

As **ZONAS** e **SUBZONAS** (domicílio, escola, trabalhos, origem e destino), qualitativas, de cada ano podem ser observadas no Anexo B. Não foi disponibilizado o mapa com as subzonas de 1977 e, em 2007, não existem subzonas.

As **coordenadas X e Y** (domicílio, escola, trabalhos, origem e destino), variáveis métricas, foram extraídas diretamente dos mapas pelo uso do software QGIS¹. Em 1977, elas foram determinadas a partir dos centroides das zonas. Em 1987 e 1997, elas foram determinadas a partir dos centroides das subzonas. Já em 2007, o levantamento contou com tecnologia de geoprocessamento e o banco de dados já continha as coordenadas.

As variáveis **F_DOM**, **F_FAM** e **F_PESS** são *dummies* que identificam o primeiro registro do domicílio, família e pessoa, respectivamente. Não contam com “NA” e têm suas frequências absolutas apresentadas pelas Tabelas 9, 10 e 11. Vale observar que a soma das ocorrências de F_DOM=1, F_FAM=1 e F_PESS=1 significam exatamente as quantidades de domicílios, famílias e pessoas entrevistados em cada um dos anos.

Tabela 9 – Estatísticas da variável “F_DOM”

ANO	1977	1987	1997	2007	Total
F_DOM=0	204.433	197.856	175.799	166.741	744.829
F_DOM=1	24.613	26.070	23.841	29.957	104.481

¹ “O QGIS é um Sistema de Informação Geográfica (SIG) de Código Aberto licenciado segundo a Licença Pública Geral GNU.” Fonte: <http://www.qgis.org/pt_BR/site/about/index.html> Acesso em 26 de dezembro de 2015

Tabela 10 – Estatísticas da variável “F_FAM”

ANO	1977	1987	1997	2007	Total
F_FAM=0	202.889	195.709	172.795	165.843	737.236
F_FAM=1	26.157	28.217	26.845	30.855	112.074

Tabela 11 – Estatísticas da variável “F_PESS”

F_PESS=0		ANO				Total
SEXO	1977	1987	1997	2007		
Masculino	72.040	62.264	52.146	51.732	238.182	
Feminino	48.978	50.849	48.714	53.561	202.102	
Total	121.018	113.113	100.860	105.293	440.284	
F_PESS=1		ANO				Total
SEXO	1977	1987	1997	2007		
Masculino	52.162	53.176	47.326	42.289	194.953	
Feminino	55.866	57.637	51.454	49.116	214.073	
Total	108.028	110.813	98.780	91.405	409.026	

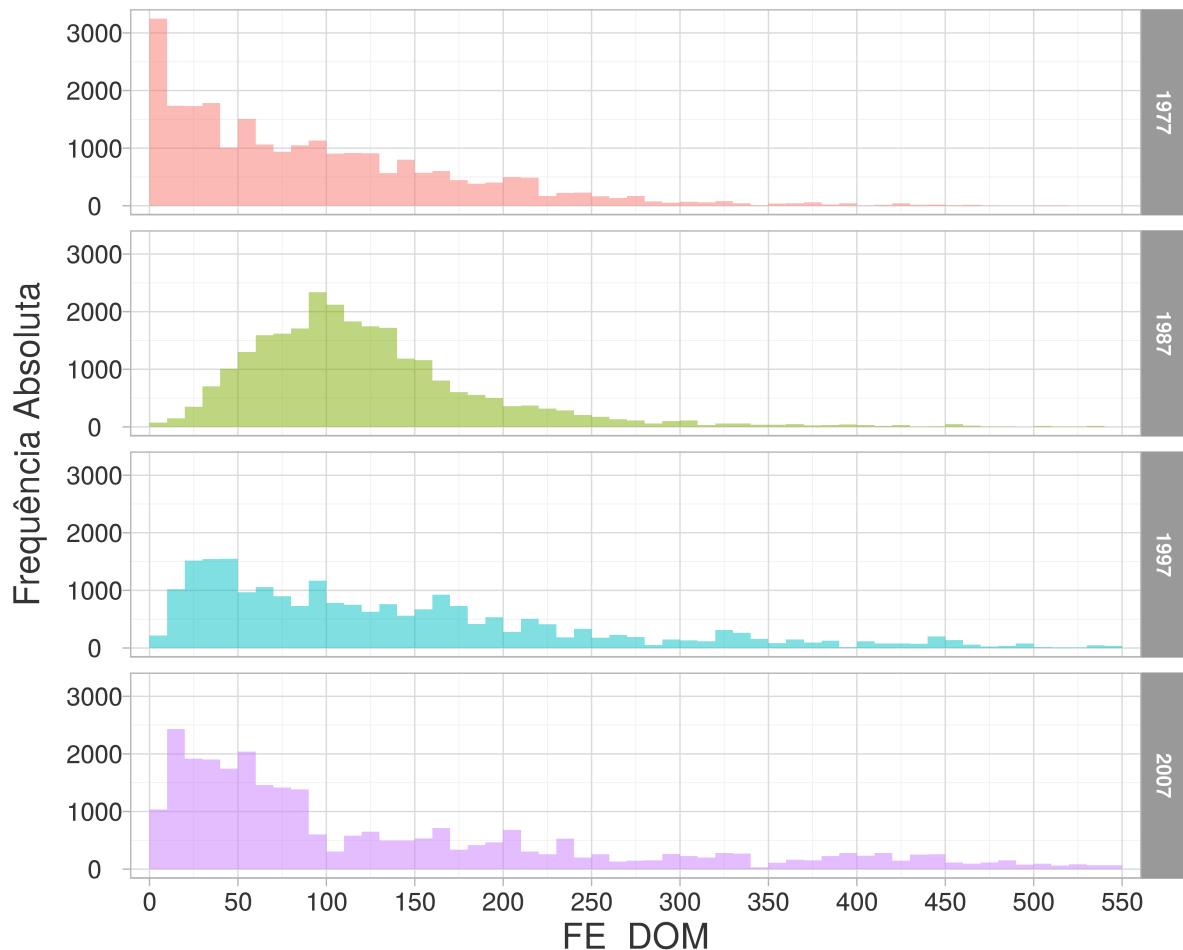
As variáveis **FE_DOM**, **FE_FAM** e **FE_PESS**, variáveis métricas, são fatores de expansão (pesos) de domicílio, família e pessoa, respectivamente, já fornecidos pelas bases de dados e que não sofreram transformação alguma.

Os fatores de expansão da família e da pessoa têm intervalos praticamente iguais aos do domicílio (ver Tabela 12), sendo que as médias, medianas e intervalos inter-quartis se alteram, pouco, dentro do mesmo ano. O Gráfico 3 apresenta as distribuições por ano da variável **FE_DOM**, cujo máximo valor da abscissa ficou em 550 pois o maior valor do intervalo inter quartil foi 535,5 em 2007. As variáveis **FE_FAM** e **FE_PESS** possuem distribuições bastante semelhantes a esta. Foi de se estranhar o ocorrência de fatores de expansão do domicílio inferiores a 1, em 1977. Caso seja um erro da base original, não se tem parâmetro para corrigi-lo e como eram poucas ocorrências, mantiveram-se esses registros.

Tabela 12 – Estatísticas da variável “FE_DOM”

ANO	Mínimo	1º Quartil	Mediana	3º Quartil	Máximo
1977	0,50	25,75	72,09	139,25	1562,75
1987	1,02	78,32	110,49	150,89	2210,00
1997	1.00	51,41	117,14	213,09	2548,63
2007	2,55	40,92	87,42	234,90	2256,41
Geral	0,50	48,67	99,20	174,06	2548,63
ANO	Média	Desvio Padrão	Assimetria	Curtose	Nº de “NA”
1977	94,92	92,42	2,63	18,78	0
1987	127,53	90,95	5,02	54,44	0
1997	178,75	209,77	3,51	20,44	0
2007	183,82	228,96	2,69	10,66	0
Geral	147,67	174,64	3,79	23,63	0

Gráfico 3 – Distribuição da variável “FE_DOM”, por ano



A variável **TIPO_DOM**, em 1977 e 1987, contava somente com as categorias “particular” e “coletivo”; já em 1997 e 2007 passou a existir também a categoria favela, que foi agregada à categoria “coletivo”. Foram adotadas as categorias coletivo e particular, de forma que funcionasse como uma **dummy** indicativa de domicílio particular (1) ou não (0). Quem não respondeu foi tratado como “NA”, encerrando 4 domicílios (23 registros) em 1987, conforme pode ser observado na Tabela 13. Nota-se que o percentual de moradias unifamiliares sempre permaneceu, em média, superior a 90%.

Tabela 13 – Estatísticas da variável “TIPO_DOM”

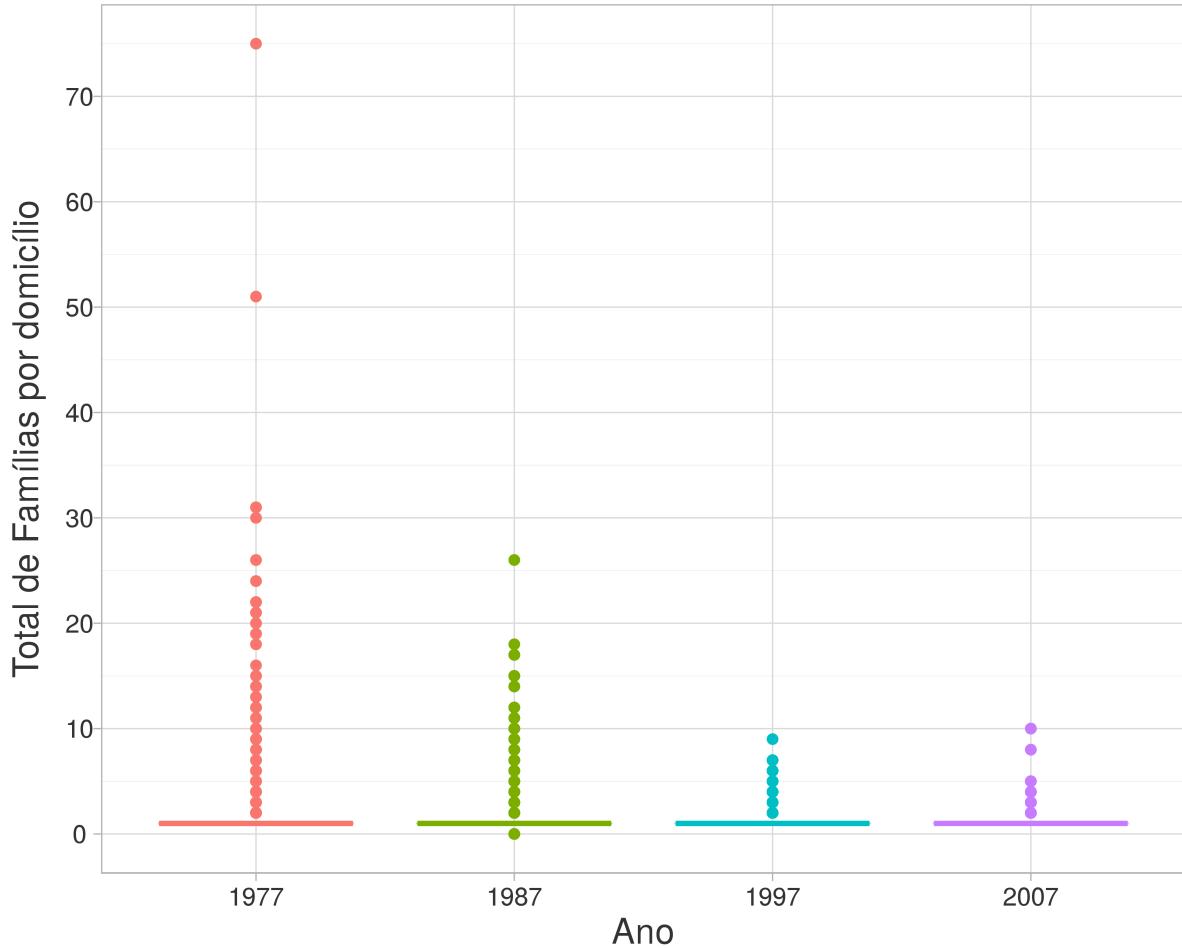
ANO	1977	1987	1997	2007	Total
TIPO_DOM=0	148	207	2.054	1.195	3.604
TIPO_DOM=1	24.465	25.859	21.787	28.762	100.873
TIPO_DOM=“NA”	0	4	0	0	4

A variável **TOT_FAM**, um dado de contagem, indica quantas famílias existem no domicílio e tem suas estatísticas descritivas apresentadas na Tabela 14. O valor médio indica haver pouco mais de uma família por domicílio, sendo a maior média de 1997 - dado coerente com a menor porcentagem de domicílios individuais (91%) indicada pela variável TIPO_DOM. A Figura 7 mostra que a influência dos *outliers* diminui com o tempo, o que também se reflete na queda dos desvios padrão, que indica menor dispersão dos dados.

Tabela 14 – Estatísticas da variável “TOT_FAM”

ANO	Média	Desvio Padrão	Assimetria	Curtose	Máximo
1977	1,08	0,99	33,70	1777,74	75
1987	1,12	0,61	13,72	306,37	26
1997	1,13	0,45	4,85	32,42	9
2007	1,03	0,23	14,00	346,13	10
Geral	1,09	0,62	35,68	2734,92	75

Figura 7 – Box plot da variável “TOT_FAM”, por ano



A variável **COND_MORA**, de natureza qualitativa, conta com três valores únicos, cujas frequências absolutas de registros de família são apresentadas na Tabela 15. A categoria “alugada” (1) que em 1977 representava 35% da amostra, caiu para 23% em 2007. A posse de residência (categoria “própria” - 2), subiu de 57% para 68% da amostra anual. A categoria “outros” (3), que abrange as categorias originais “cedida”, “outros” e “não se aplica”, oscila numa faixa próxima dos 10%. Quem não respondeu foi tratado como “NA”, encerrando 717 domicílios: 9 em 1977, 264 em 1987, 26 em 1997 e 460 em 2007.

Tabela 15 – Estatísticas da variável “COND_MORA”

ANO	1977	1987	1997	2007	Total
COND_MORA=1	9.174	7.967	5.168	7.268	29.577
COND_MORA=2	14.984	17.429	18.040	21.062	71.515
COND_MORA=3	1.990	2.557	3.611	2.065	10.223
COND_MORA=“NA”	9	264	26	460	759

Não foram mantidas no BDU as variáveis relativas aos bens de consumo, pois a função delas era servir de *input* para determinar a renda atribuída (na ausência de declaração da renda), informação de que já se dispõe no banco de dados. Apenas os bens que são meios de transporte foram mantidos (automóveis, motocicletas e bicicletas) e serão explorados mais adiante.

Os valores monetários de renda familiar, renda individual e valor de estacionamento encontravam-se, em cada banco de dados, em um mês de referência diferente (segundo Quadro 2). Foram estudadas as possibilidades de correção pelos seguintes índices:

- IPC-Brasil: índice de preços ao consumidor, que mede a variação de preços de um conjunto fixo de bens e serviços componentes de despesas habituais de famílias com nível de renda situado entre 1 e 33 salários mínimos mensais. Sua pesquisa de preços se desenvolve diariamente, cobrindo as sete principais capitais do país. Suas variações (DI, M e 10) referem-se ao período da coleta dos preços. A série histórica que se obteve de fontes oficiais (página da FGV e do Banco Central) não disponibilizava valores anteriores a 1989.
- IGP: índice geral de preços, (a partir de 1950) é a média aritmética ponderada de três outros índices de preços (Índice de Preços ao Produtor Amplo (50%), Índice de Preços ao Consumidor (30%) e Índice Nacional de Custo da Construção (10%). O IGP-M/FGV se refere ao período do dia vinte e um do mês anterior ao dia vinte do mês de referência e o IGP-DI/FGV se refere ao período do dia um ao dia trinta do mês em referência. A série histórica que se obteve de fontes oficiais (página da FGV e do Banco Central) não disponibilizava valores anteriores a 1989 para o IGP-M, ao passo que o IGP-DI² dispunha de valores para correção desde a década de 1940.
- IPCA: índice geral de preços ao consumidor amplo, que mede a variação de preços de um conjunto fixo de bens e serviços componentes de despesas habituais de famílias com nível de renda situado entre 1 e 40 salários mínimos mensais. Sua pesquisa de preços se desenvolve diariamente, cobrindo as dez principais capitais do país e Brasília. A série histórica que se obteve de fontes oficiais (página do IBGE e do Banco Central) divulga valores após dezembro de 1979. Sua variação IPCA-E é ainda mais recente (1991) e tem divulgação trimestral.
- INPC: índice nacional de preços ao consumidor amplo, que mede a variação de preços de um conjunto fixo de bens e serviços componentes de despesas habituais de famílias com nível de renda situado entre 1 e 5 salários mínimos mensais. Sua pesquisa de preços se desenvolve continuamente, cobrindo as dez principais capitais do país e Brasília. A série histórica que se obteve de fontes oficiais (página do IBGE e do Banco Central) divulga valores após março de 1979.

² Metodologia do IGP-DI disponível em: <<http://portalibre.fgv.br/lumis/portal/file/fileDownload.jsp?fileId=8A7C82C54DB5CA9F014DD9322ADD306E>> Acesso em 30 de agosto de 2015

Assim, pela abrangência histórica foi utilizado o IPG-DI para levar os valores de setembro de 1977 para setembro de 1987. Estes valores, bem como os advindos da OD de 1987, foram levados a outubro de 2007 sofrendo correção do INPC. Este mesmo índice foi utilizado para levar os valores de outubro de 1997 a outubro de 2007. Escolheu-se o INPC, dentro os demais disponíveis, devido à faixa de abrangência da renda familiar utilizada em sua determinação. Não se quis priorizar índices cujas cestas tivessem um perfil de consumo alto, dado que as políticas de transporte público devem priorizar especialmente as menores faixas de renda. A composição final dos deflatores utilizados é apresentada na Tabela 16.

Tabela 16 – Deflatores utilizados para correção dos valores monetários para outubro/2007

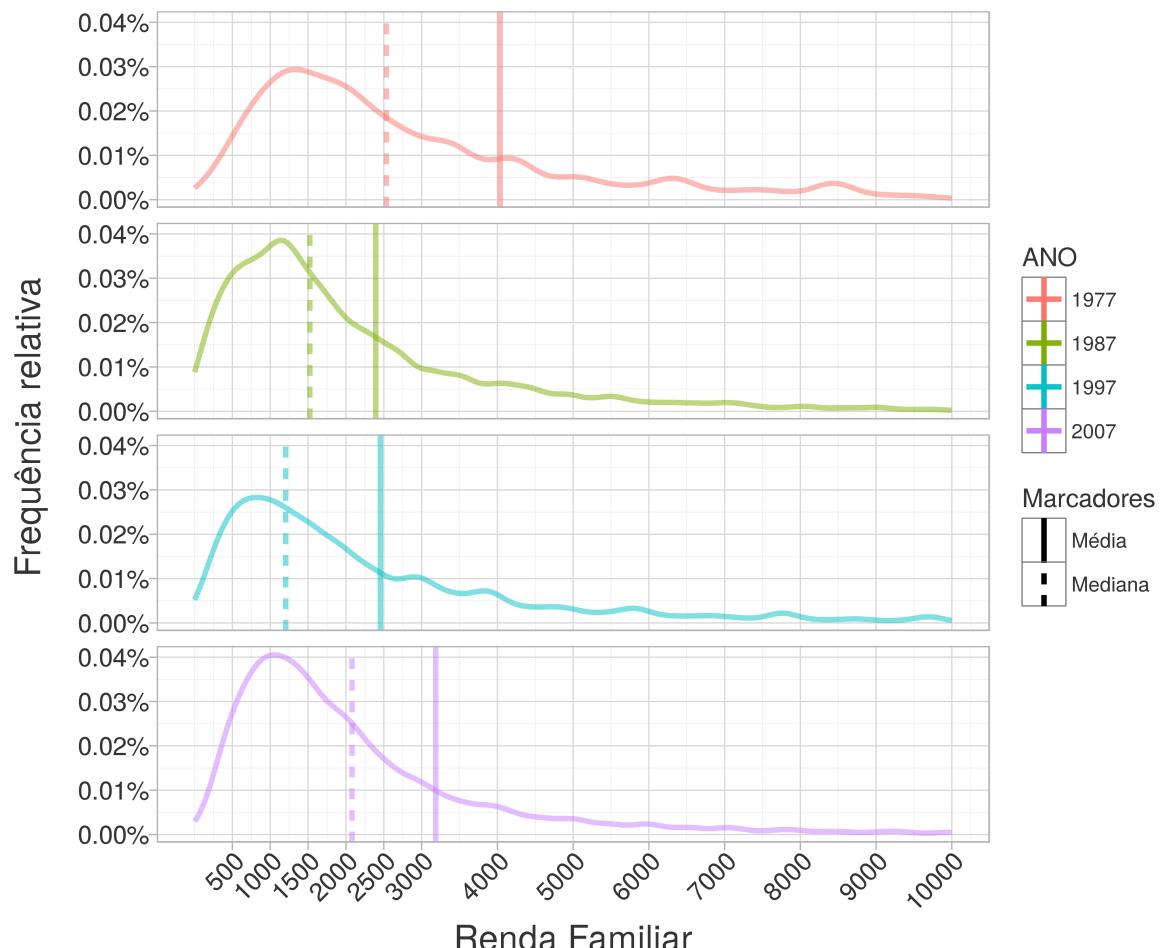
ANO	1977	1987	1997	2007
Deflator	0,44234590	0,09664666	1,94136464	1

A renda familiar (variável **REN_FAM**) tem seus maiores valores de média e mediana em 1977, conforme Tabela 17, talvez por reflexo da época do “milagre econômico brasileiro”, comumente atribuído ao período do final dos anos 1960 até a metade dos anos 1970. Após isso, 1987 apresenta a menor média e também queda da mediana; pode ser devido à recessão que marcou o Brasil nos anos 1980, quando se registraram baixo crescimento do PIB, alto nível de desemprego, perda do poder de compra da população, e índices de inflação extremamente elevados. Em 1997, a média da renda familiar sobe, porém a mediana continua a cair, o que pode significar uma recuperação do aquecimento econômico, porém, sem frear o aprofundamento das desigualdades. Com o valor da média de 2007, parece realmente estar ocorrendo uma recuperação econômica (aumento da renda média familiar) e também uma redução das desigualdades (aumento da mediana da renda média familiar) - ver Gráfico 4.

Tabela 17 – Estatísticas da variável “REN_FAM”

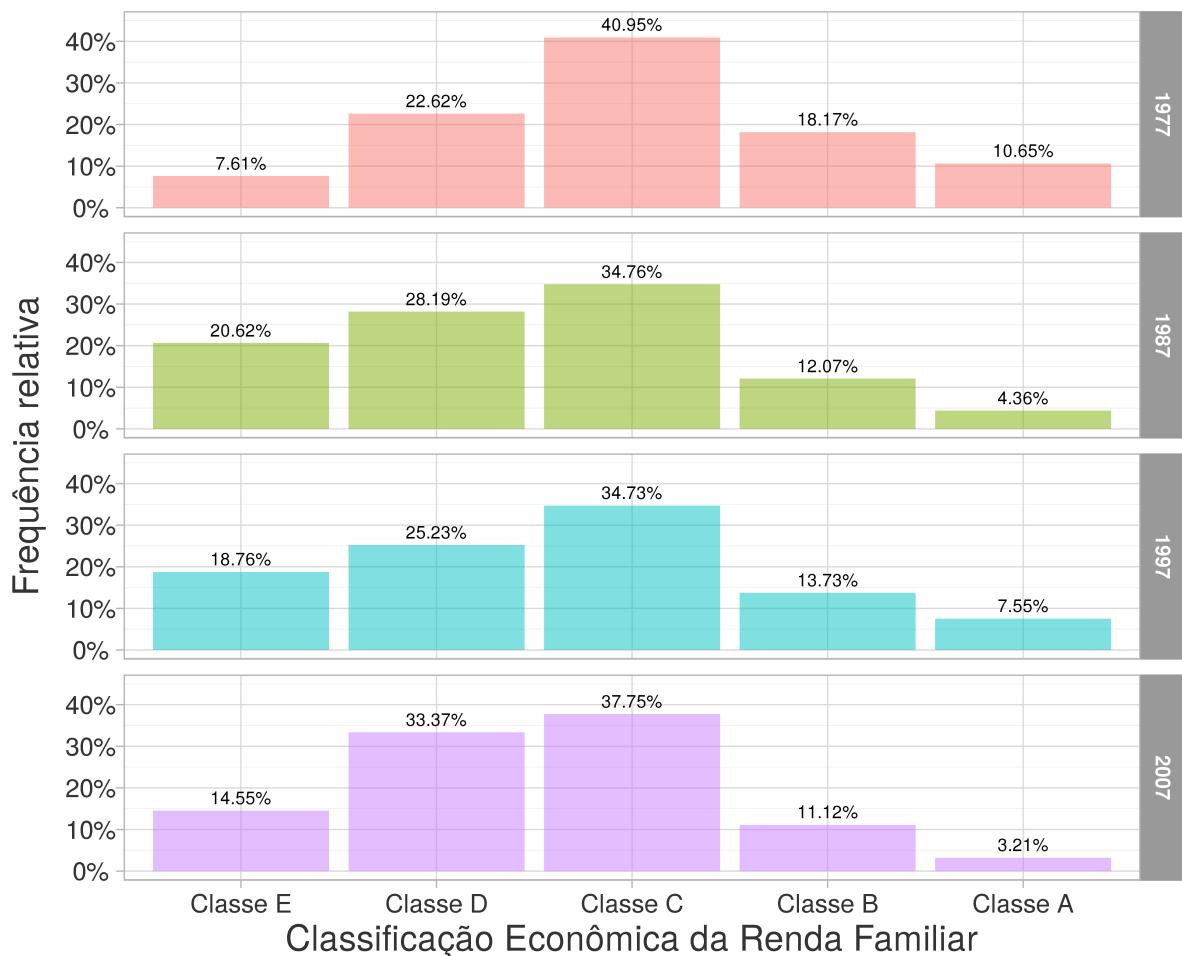
ANO	Mínimo	1º Quartil	Mediana	3º Quartil	Máximo
1977	0	1.351,51	2.534,08	4.796,58	42.234,2
1987	0	875,62	1.523,44	2.801,21	61.830,1
1997	0	330,03	1.203,65	2.912,05	172.781,0
2007	0	1.144,82	2.080,00	4.000,00	46.000,0
Geral	0	935,73	1.801,63	3.582,21	172.181,0
ANO	Média	Desvio Padrão	Assimetria	Curtose	Nº de “NA”
1977	4.037,27	4.724,88	3,56	18,74	0
1987	2.393,58	2.826,96	4,06	29,85	0
1997	2.453,89	4.148,06	7,20	149,78	0
2007	3.186,19	3.312,25	2,85	13,73	0
Geral	3.009,86	3.845,58	4,74	63,49	0

Gráfico 4 – Distribuição da variável “REN_FAM”, por ano



Para a construção do Gráfico 5, a partir da variável **FAIXA_REN_FAM**, foram retiradas as famílias que declararam renda nula. Posto isso, vê-se que as classes E e D, em 1977, correspondiam a 29% da amostra. Em 1987 esse valor vai a 49% e depois começa a decrescer para 45% em 1997 e 37% em 2007. A classe C (classe média) decresce de 1977 (37%) até 1997 (33%) e só aumenta novamente em 2007 (36%). Para estas classificações, foram utilizados sempre os valores em R\$ de outubro de 2007, bem como a classificação de classes econômicas vigente neste mesmo período.

Gráfico 5 – Distribuição da variável “FAIXA_REN_FAM”, por ano



Uma das variáveis que são *proxy* da renda é a quantidade de automóveis na família (**QT_AUTO**). As Tabelas 18 e 19 mostram que a frota de automóveis de uso familiar vem crescendo tanto no Brasil como na RMSP, assim como a motorização familiar (número de veículos por família) ao longo do tempo. Isto é, mesmo com o crescimento populacional do período, a aquisição de carros pelas famílias aumenta em taxa ainda maior. A média de automóveis por família aumentou com o passar do tempo, conforme observa-se na coluna *média* da Tabela 20.

Tabela 18 – Evolução da frota de automóveis na RMSP segundo as Pesquisas OD

ANO	1977	1987	1997	2007
Frota	1.391.832	2.014.474	3.092.238	3.545.263

Nota: Segundo o Departamento Estadual de Trânsito de São Paulo, em 2007 o município de São Paulo contava com cerca de 4,5 milhões de automóveis. Tal discrepância pode ocorrer porque os dados da CET levam em consideração, além dos veículos particulares, veículos de empresas, órgãos públicos, etc; e também por basear-se no cadastro do IPVA.

Tabela 19 – Venda interna de veículos no Brasil entre 1960 e 2009

Ano	Autos	Total	Fator de crescimento (total)
1960	40.980	131.499	1
1970	308.024	416.704	3,2
1980	739.028	980.261	7,5
1990	532.906	712.741	5,4
2000	1.176.774	1.489.481	11,3
2009	2.474.764	3.141.240	23,9

Fonte: Adaptado de ([VASCONCELLOS, 2012](#), p.29)

Tabela 20 – Estatísticas da variável “QT_AUTO”

ANO	Mínimo	1º Quartil	Mediana	3º Quartil	Máximo
1977	0	0	0	1	7
1987	0	0	0	1	9
1997	0	0	1	1	9
2007	0	0	1	1	8
Geral	0	0	0	1	9

ANO	Média	Desvio Padrão	Assimetria	Curtose	Nº de “NA”
1977	0,65	0,81	1,36	2,28	0
1987	0,57	0,77	1,59	4,09	0
1997	0,72	0,89	1,57	3,78	0
2007	0,78	0,86	1,19	1,99	0
Geral	0,68	0,84	1,43	3,03	0

Diversos estudos foram realizados para a construção de modelos desagregados que explicassem a posse de autos (RYAN; HAN, 1999; DARGAY; GATELY, 1999; DARGAY, 2001; CHU, 2002; KARLAFTIS; GOLIAS, 2002; PFEIFFER; STRAMBI, 2005). É comum a utilização de variáveis como renda, número de trabalhadores, tamanho da família, número de estudantes, presença ou não de crianças na família, sexo e idade da pessoa responsável pela família. Embora não seja o foco deste estudo modelar a posse de autos pelas famílias, compreender o que influencia a motorização é importante porque a motorização influencia diretamente a mobilidade dos indivíduos. Pfeiffer e Strambi (2005), que analisam a evolução da motorização da RMSP entre 1987 e 1997, concluem que, para explicar a posse de autos “*as variáveis de estrutura familiar utilizadas na análise e modelagem da posse de autos perderam importância ao longo do tempo, assim como a própria renda familiar*”. Eles sugerem que isso pode ter se dado pela maior facilidade de financiamentos para aquisição de automóveis por uma família, ou ainda pela evolução das condições do transporte público, assim como da organização espacial da região metropolitana.

Segundo a Tabela 21, percebe-se que a proporção de famílias sem automóvel particular oscilou entre os anos, mantendo ainda assim uma tendência de queda entre 1977 e 2007. Oscilação entre os anos também ocorreu na proporção das famílias com dois ou mais automóveis, com tendência de crescimento entre 1977 e 2007. O comportamento mais consistente com os cenários macro econômicos brasileiros foi o das famílias com um automóvel: tiveram leve queda na posse de um automóvel em 1987 (década de crise econômica), aumento de pouco mais de 10% em 1997 (após a estabilização da moeda em 1994), e aumento tímido (0,3%) em 2007.

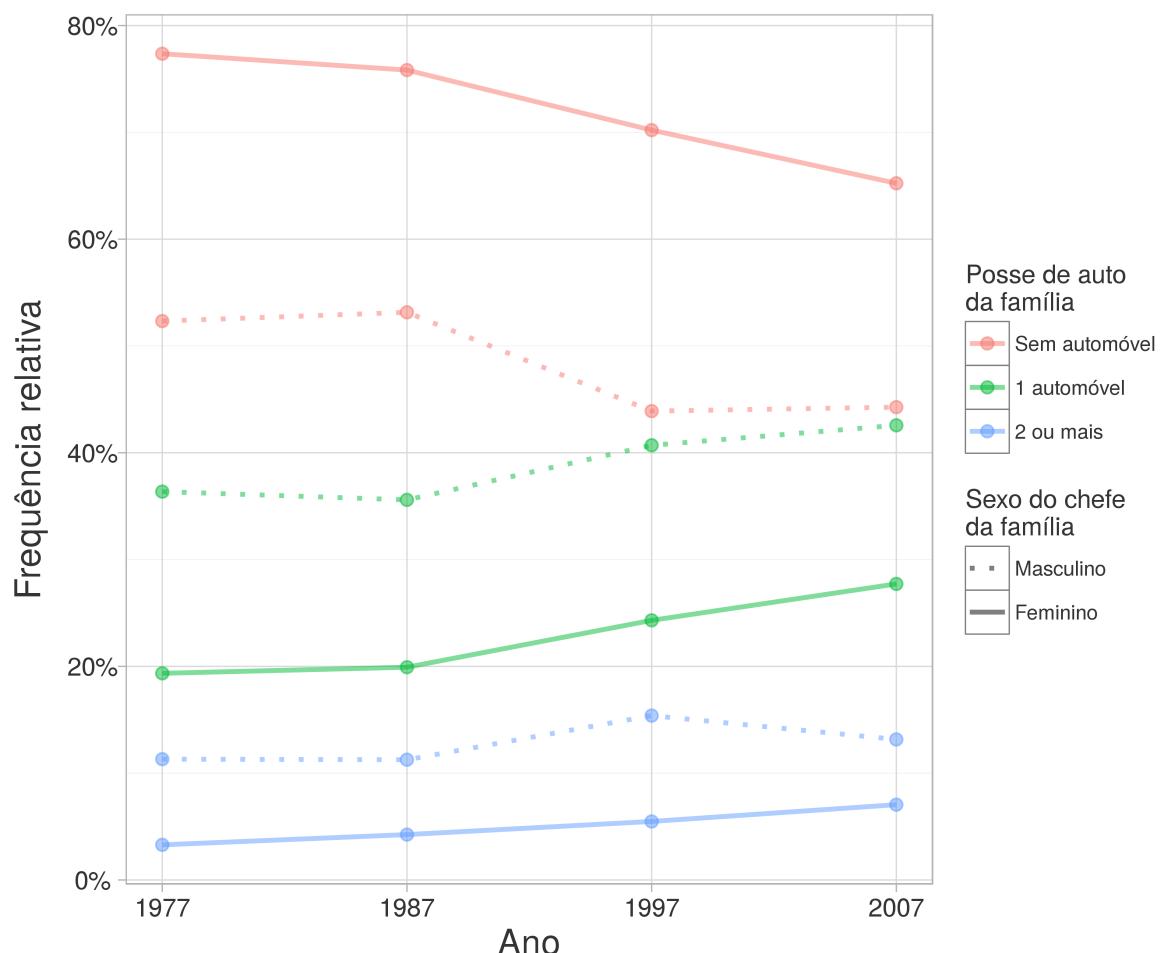
Foram criadas duas *dummies* relativas à presença de automóveis na famílias: (i) presença de 1 automóvel e (ii) presença de 2 ou mais automóveis. Segmentando as famílias segundo o sexo da pessoa responsável e analisando as frequências relativas da posse de autos através dessas *dummies*, obtém-se o Gráfico 6. Nele observa-se que as variações da posse de autos das famílias cujos responsáveis são homens são semelhantes às daquelas cujas responsáveis são mulheres. Porém, as famílias que possuem 2 ou mais automóveis e cujos responsáveis são homens, tiveram uma leve queda de 2 pontos percentuais, enquanto as com dois ou mais automóveis chefiadas por mulheres apresentaram um aumento de 4 pontos percentuais.

Mas o grande destaque no que se refere à posse de automóvel das famílias pode ser dado ao par formado pela queda de 12 pontos percentuais para famílias sem automóvel e pelo aumento de 8 pontos percentuais para famílias com posse de um automóvel, ambos casos são de famílias chefiadas por mulheres e com tendência a continuar seus movimentos de queda e crescimento, respectivamente.

Tabela 21 – Proporção das famílias segundo posse de automóveis e de acordo com sexo da pessoa responsável

Sexo da Pessoa Responsável	Posse de auto	ANO				Geral (%)
		1977 (%)	1987 (%)	1997 (%)	2007 (%)	
Feminino	0 autos	77,36	75,83	70,21	65,23	69,21
	1 auto	19,35	19,92	24,31	27,72	24,90
	2 ou + autos	3,29	4,25	5,48	7,05	5,89
Masculino	0 autos	52,33	53,15	43,90	44,27	47,64
	1 auto	36,36	35,59	40,70	42,57	39,32
	2 ou + autos	11,31	11,26	15,39	13,16	13,04
Geral	0 autos	55,70	56,92	49,37	51,25	52,65
	1 auto	34,07	33,00	37,30	37,62	35,98
	2 ou + autos	10,23	10,08	13,33	11,12	11,38

Gráfico 6 – Proporção de famílias com pessoa responsável do sexo feminino e do sexo masculino, segundo posse de automóveis, por ano



A quantidade de motocicletas (**QT_MOTO**) e bicicletas (**QT_BICI**) foram levantadas apenas na Pesquisa OD de 2007, e têm suas estatísticas descritivas apresentadas na Tabela 22. Percebe-se que, embora um meio de transporte individual motorizado mais barato, a incidência da posse de motocicletas é uma fenômeno mais raro, provavelmente devido ao risco associado a esse meio de transporte.

Essa falta de segurança na condução do meio de transporte também ocorre com a bicicleta, porém, além de ser ainda mais barata que a motocicleta, ela não é motorizada e, por isso, comumente trafega nos passeios públicos e calçadas, onde o(a) condutor(a) sofre menor risco de colisão com meios motorizados de transporte (motocicletas, carros, ônibus e caminhões). Com a recente política municipal de incentivo à bicicleta como meio de transporte, por meio de investimentos em infraestrutura cicloviária, é possível que na Pesquisa OD de 2017 haja evoluções destes dados que mereçam investigação. Vale lembrar que existe também a questão do status associado muito mais fortemente à posse do automóvel do que da motocicleta; e no caso das bicicletas, seu uso é associado à ideia de pobreza e falta de recursos suficientes para comprar um carro.

Tabela 22 – Estatísticas das variáveis “QT_MOTO” e “QT_BICI”

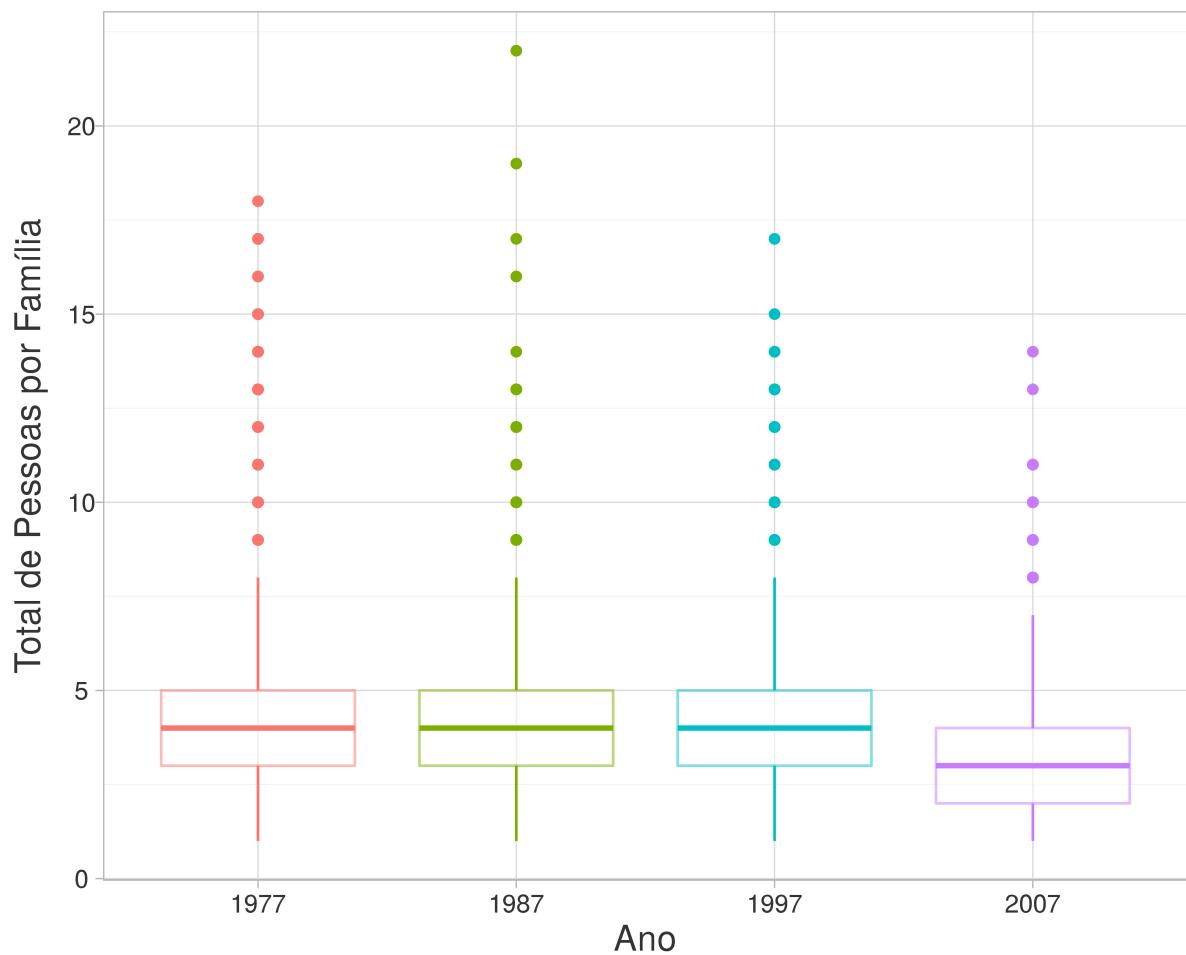
2007	Mínimo	1º Quartil	Mediana	3º Quartil	Máximo
motocicleta	0	0	0	0	9
bicicleta	0	0	0	1	9
2007	Média	Desvio Padrão	Assimetria	Curtose	Nº de “NA”
motocicleta	0,07	0,29	5,86	74,31	0
bicicleta	0,46	0,8	2,27	7,5	0

A variável **TOT_PESS**, um dado de contagem, indica quantas pessoas existem na família e tem suas estatísticas descritivas apresentadas na Tabela 23. Não havia *missing values* e o valor médio indica que o tamanho da família diminuiu, passando de pouco mais de 4 pessoas em 1977 para pouco menos de 3 pessoas por família em 2007. Em comportamento análogo a **TOT_FAM**, a Figura 8 indica que a influência dos *outliers* de **TOT_PESS** diminui com o tempo, o que também se reflete na queda dos desvios padrão, que indica menor dispersão dos dados.

Tabela 23 – Estatísticas da variável “TOT_PESS”

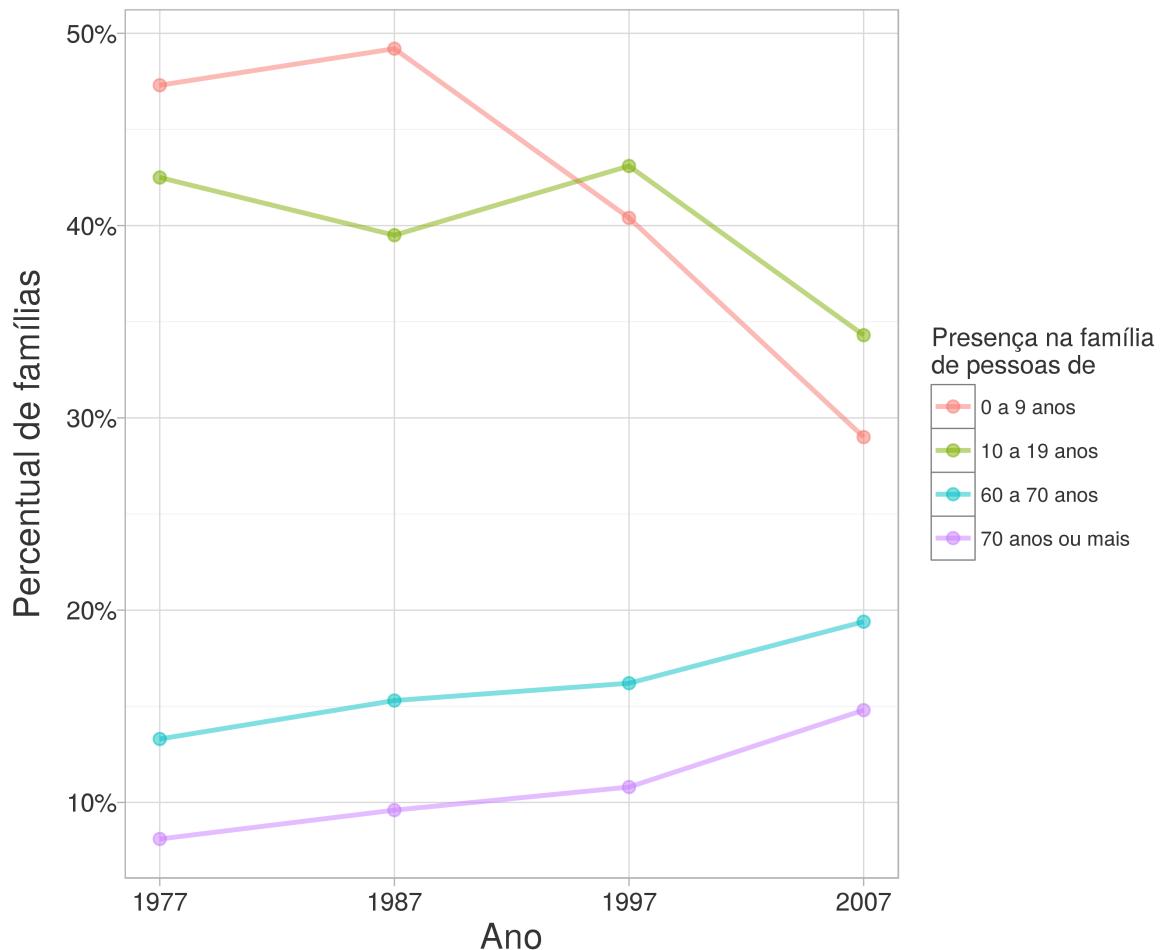
ANO	Média	Desvio Padrão	Assimetria	Curtose	Máximo
1977	4,13	2,09	0,99	1,78	18
1987	3,93	1,80	0,92	2,03	22
1997	3,68	1,73	0,82	1,68	17
2007	2,96	1,46	0,78	1,00	14
Geral	3,65	1,83	1,00	2,10	22

Figura 8 – Box plot da variável “TOT_PESS”, por ano



Percebe-se pelo Gráfico 7 a tendência de diminuição geral nas porcentagens de famílias com filhos pequenos (até 9 anos) a partir de 1987, efeito que será percebido na faixa etária seguinte (entre 10 e 19 anos) em 1997. Essa diminuição da presença de dependentes jovens (crianças/adolescentes) mantém-se em 2007. Nos mesmos períodos de análise, ocorre o envelhecimento da população, efeito capturado no gráfico pela presença de idosos (acima de 60 ou de 70 anos) com taxas positivas de crescimento.

Gráfico 7 – Proporção de famílias com presença de dependentes, por ano



Um dos fatores do indivíduo que influencia seu padrão de deslocamentos é o estágio no “ciclo de vida” (Van de Bilt, 1997). Isto é, as atividades desenvolvidas por uma pessoa dependem em que fase da vida ela, e também os demais membros da família, se encontram. A variável **IDADE** é uma das variáveis que definem o ciclo de vida das pessoas. É possível perceber no Gráfico 8 que houve uma transição da pirâmide etária da RMSP, indicando envelhecimento da população tanto masculina como feminina.

Embora não exista relação identitária entre sexo e gênero, conforme já exposto na revisão de literatura, a variável **SEXO** é uma componente importante para compreender a categoria de análise gênero. Percebe-se na Tabela 24 que em todos os anos a proporção de mulheres entrevistadas é superior à de homens, provavelmente devido ao fato de a pesquisa ser domiciliar e ser mais provável encontrar mulheres do que homens em casa.

Gráfico 8 – Distribuição da variável “IDADE” de respondentes, por ano e por sexo

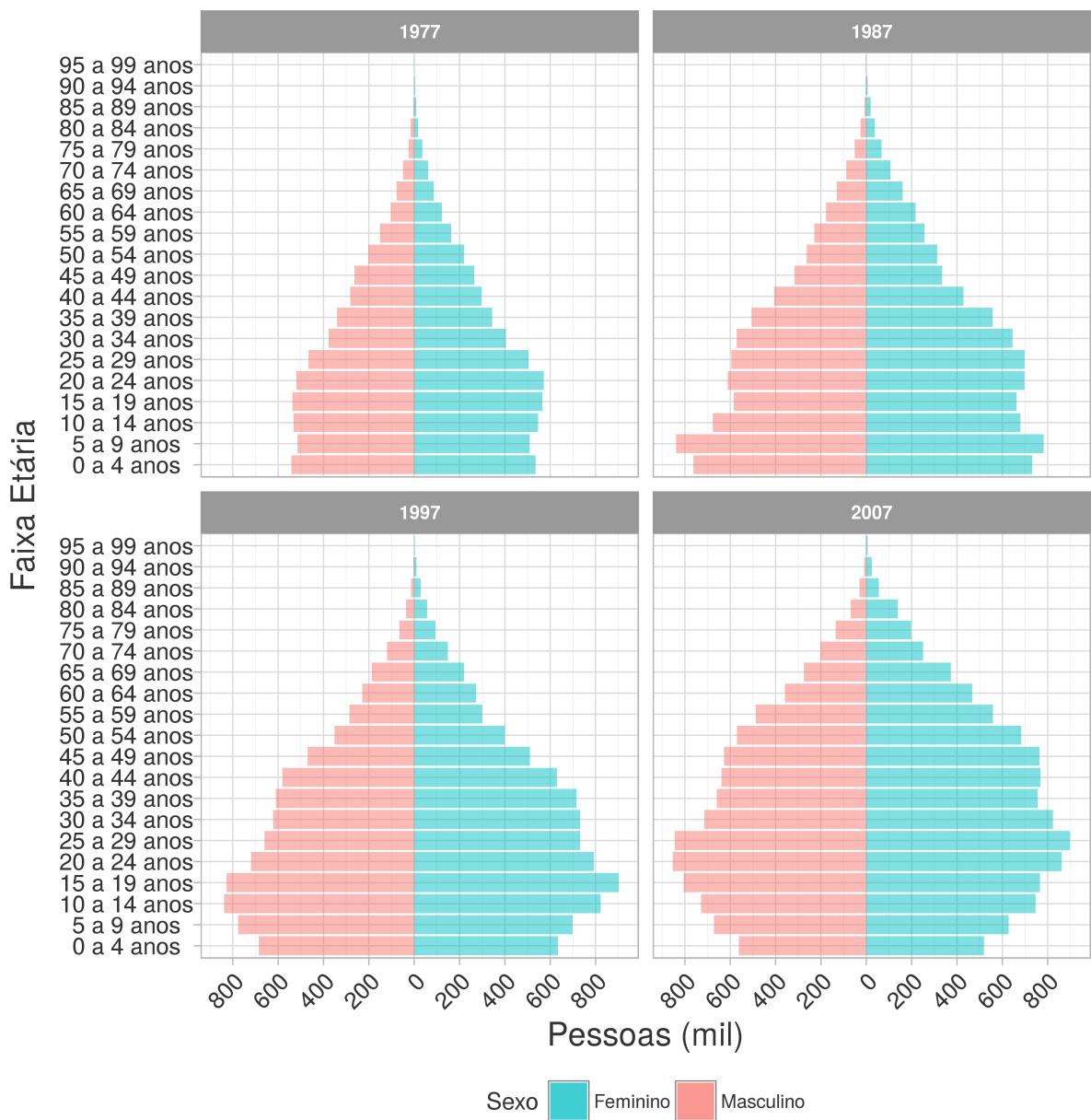


Tabela 24 – Frequências absoluta e relativa da variável SEXO, por ano

ANO	1977	1987	1997	2007
quantidade de pessoas do sexo masculino na amostra	52.162	53.176	47.326	42.289
% de pessoas do sexo masculino	48,3	48,0	47,9	46,3
quantidade de pessoas do sexo feminino na amostra	55.866	57.637	51.454	49.116
% de pessoas do sexo feminino	51,7	52,0	52,1	53,7

Observando a variável situação familiar (**SIT_FAM**), nota-se que para as mulheres houve uma mudança ao longo dessas três décadas - ver Gráfico 9. Em 1977, era mais frequente elas ocuparem a posição de filhas (41,8%), em seguida de cônjuges (36,3%). A posição de “pessoa responsável” pela família é a quarta categoria mais frequente (6,3%), de seis. Tal distribuição permanece semelhante em 1987. Em 1997, no entanto, a posição de “pessoa responsável” pela família (11,1%) já quase se equipara à posição de “outro parente/agregado(a)” (11,3%). Em 2007, se aproxima de um quarto proporção das mulheres entrevistadas que são responsáveis por suas famílias (22,6%), representando aumento de mais de 3,5 vezes em relação aos percentuais de 1977. O percentual de mulheres cônjuges/companheiras pouco se altera ao longo do tempo, permanecendo na faixa dos 35%. Há diminuição da posição de empregado(a) doméstico(a) para as mulheres (da ordem da metade). Existe, também, queda da frequência daqueles que se declaram na posição de filho(a) ou enteado(a) tanto para homens como para mulheres - em ordem de grandeza próxima: cerca de 10% para mulheres e 9% para homens. Isso pode ser reflexo da diminuição das taxas de fecundidade³ da população (ver Tabela 25). Entre os homens percebe-se que houve crescimento entre aqueles com posição de “pessoas responsável” de cerca de 3,5%, e também dos que declaram-se cônjuge/companheiro (cerca de 20 vezes) - esta última constatação é coerente com o fato de mais mulheres serem a principal fonte de renda doméstica, ou seja, serem consideradas a “pessoa responsável” da família.

Tabela 25 – Evolução das taxas de fecundidade no Brasil, de 1970 a 2010

Ano	1970	1980	1991	2000	2010
Taxa de fecundidade (Brasil)	5,8	4,4	2,7	2,4	1,9
Taxa de fecundidade (Sudeste)	4,6	3,2	2,4	2,1	1,7
Taxa de fecundidade (São Paulo)	3,94	3,24	2,28	2,05	1,67

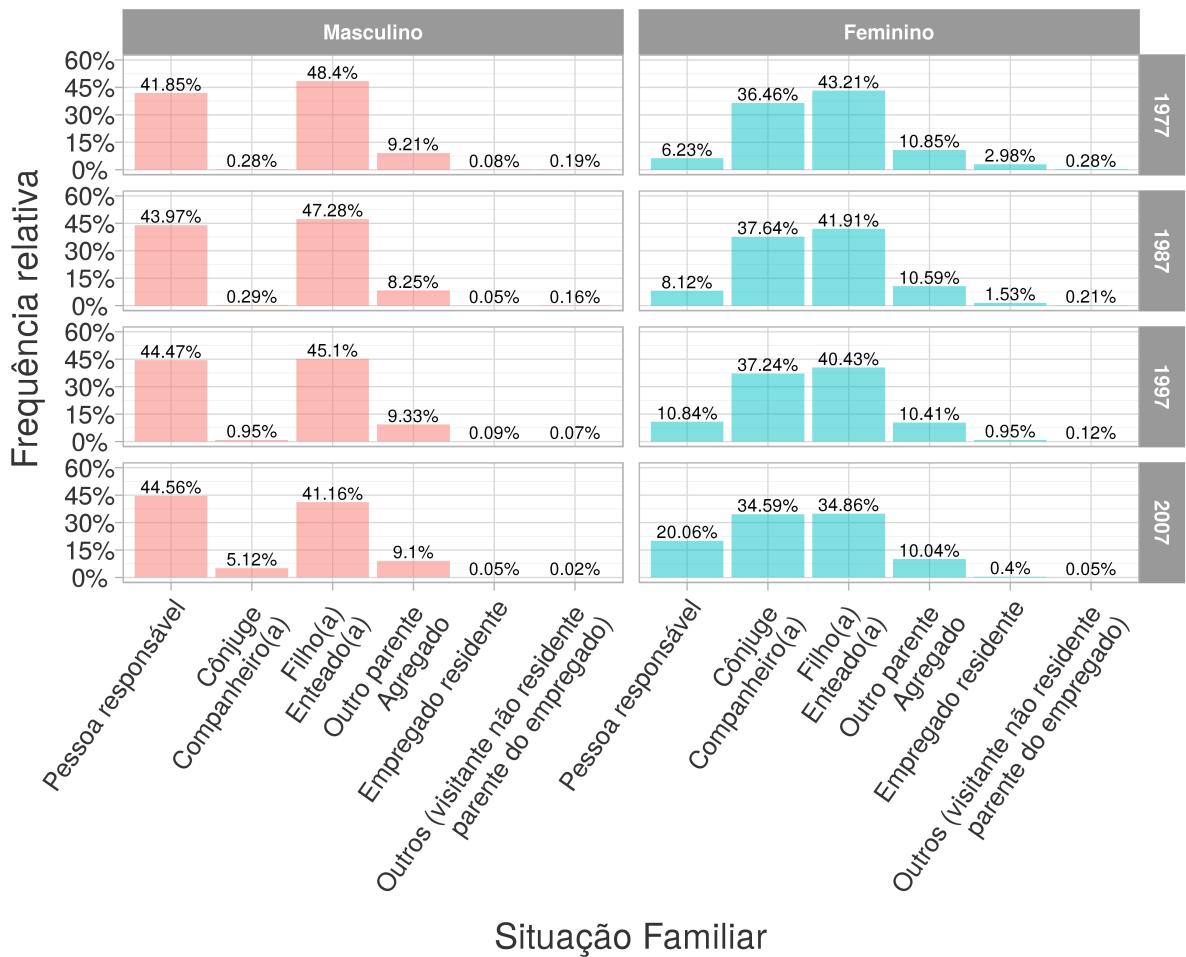
Fonte: Compilação a partir de dados dos censos do IBGE disponíveis em <<http://seculoxx.ibge.gov.br/populacionais-sociais-politicas-e-culturais/busca-por-palavra-chave/populacao/810-fecundidade>> Acesso em 17 de novembro de 2014

Nota: Ao analisar as taxas de fecundidades para as Grandes Regiões, nota-se que o Sudeste tem os menores percentuais de mulheres que tiveram filhos em todos os subgrupos etários.

Tem-se como uma das hipótese deste trabalho que houve evolução dos padrões de mobilidade por gênero. Articular as variáveis sexo e situação familiar pode ser uma estratégia para tentar utilizar o gênero como categoria de análise.

³ Por “taxa de fecundidade total” entende-se o número médio de filhos que teria uma mulher de uma coorte hipotética (15 e 49 anos de idade) ao final de seu período reprodutivo. Fonte: IBGE. Disponível em: <<http://www.ibge.gov.br/home/estatistica/populacao/condicaodevida/indicadoresminimos/conceitos.shtml#tf>>

Gráfico 9 – Distribuição da variável “SIT_FAM”, por ano e por sexo



[Van de Bilt \(1997\)](#), a partir da análise de trabalhos de diversos autores, para compreender melhor o comportamento das pessoas em relação à participação em atividades e a consequente geração de viagens, elenca como conceito importante, além do estágio no ciclo de vida familiar e dos papéis sociais, o estilo de vida. Duas das variáveis que auxiliam a caracterizar o estilo de vida da pessoa respondente é saber se ela estuda e se ela trabalha.

A variável **TRABALHA** foi construída a partir da variável OCUP da seguinte maneira: se a categoria da ocupação fosse 1, correspondente a “tem trabalho”, a variável recebia valor 1, caso contrário, recebia valor 0. A variável **ESTUDA** conta com as categorias sim (1) e não (0). Em 1987, 1997 e 2007 esta variável existe, mas somente com a categoria “não” em comum. Assim, as demais alternativas de resposta tornaram-se simplesmente “sim”, independente das subdivisões que apresentassem. Para 1977, ano em que essa variável não existe, ela foi preenchida segundo o seguinte critério: a pessoa foi considerada estudante se o campo “Zona da escola” dela não fosse vazio ou igual a zero. Preferiu-se não utilizar a categoria “estudante” da variável ocupação para não perder a informação de quem estuda e trabalha, pois neste caso, estudante não seria a ocupação

principal da pessoa e sim “tem trabalho”.

As Tabelas 26 e 27 indicam as frequências das *dummies* TRABALHA e ESTUDA, por ano e por sexo, em relação à população (aplicados os fatores de expansão). Percebe-se que o percentual de trabalhadores(as) aumenta na população devido à maior participação feminina no mercado de trabalho, dado que os percentuais dos trabalhadores permanecem no mesmo patamar (24% da população). Segmentando por sexo, vê-se que houve um aumento de quase 20% na participação feminina no mercado de trabalho e pouco mais de 2% de incremento na masculina. Analisando as proporções de mulheres e homens estudantes, nota-se que houve pequeno crescimento entre 1977 e 1997 (3,2% feminino e 2,5% masculino) seguido de leve queda (1,5% pra ambos sexos). A expectativa era de ter havido um grande crescimento percentual do número de estudantes, especialmente mulheres, para o período observado. O crescimento tímido pode dever-se ao fato de que, embora a população da RMSP tenha elevado seus níveis de escolarização, o crescimento (desacelerando) e o envelhecimento (em ascensão) da população implicam haver mais gente fora da faixa etária escolar do que dentro dela. As quedas no percentual de estudantes pode estar relacionada tanto ao envelhecimento da população quanto à saturação do sistema de ensino, cuja obrigatoriedade de oferta pública limita-se à Educação Básica (9 anos de Ensino Fundamental e 3 anos de Ensino Médio).

Tabela 26 – Frequências da variável TRABALHA, por ano e sexo

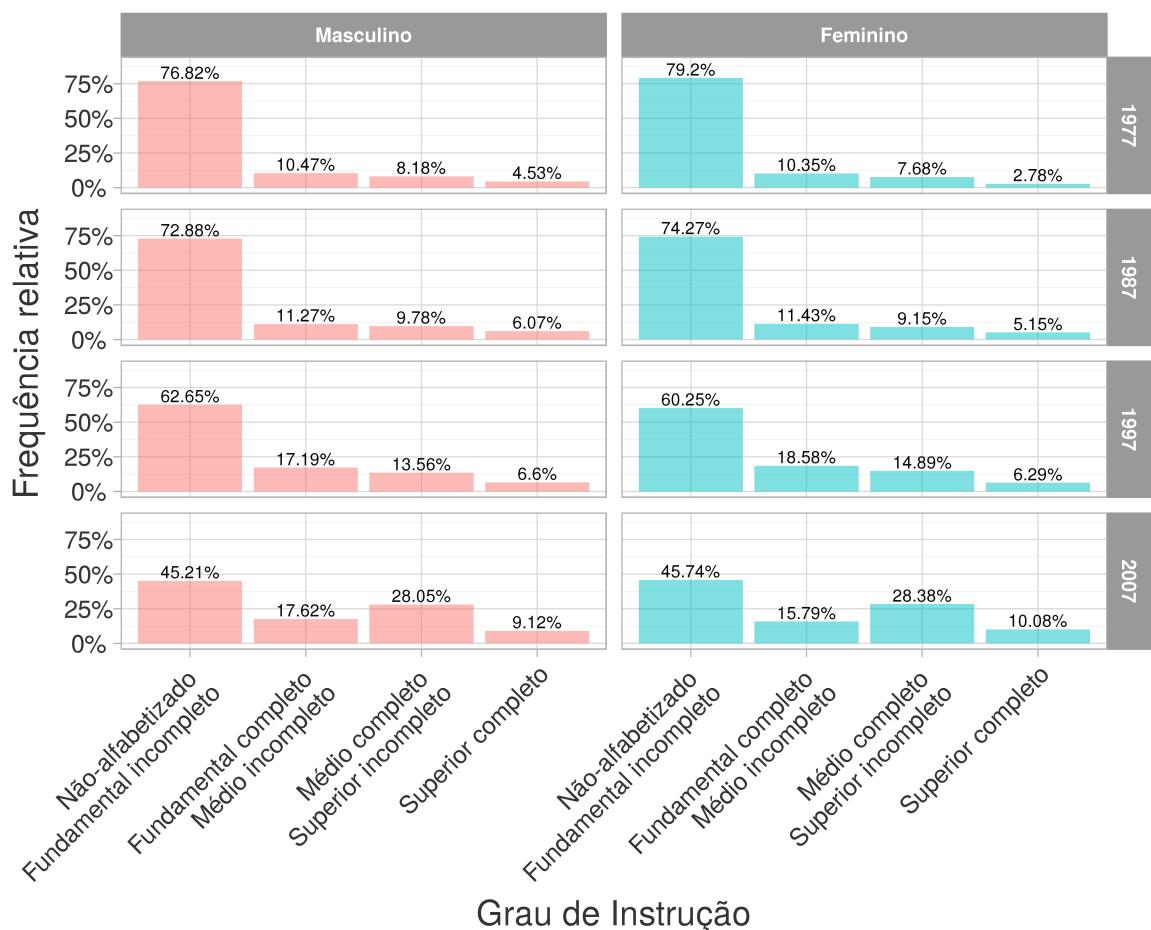
ANO	1977	1987	1997	2007
% de trabalhadoras relativo ao total da população	9,78	12,18	16,27	19,88
% de trabalhadores relativo ao total da população	24,42	24,73	24,61	24,82
% de trabalhadoras relativo ao total de mulheres	19,08	23,44	31,38	37,76
% de trabalhadores relativo ao total de homens	50,12	51,50	51,13	52,42
% de trabalhadores(as) relativo ao total da população	34,20	36,91	40,89	44,70

Tabela 27 – Frequências da variável ESTUDA, por ano e sexo

ANO	1977	1987	1997	2007
% de estudantes mulheres relativo ao total da população	11,94	12,88	15,17	13,75
% de estudantes homens relativo ao total da população	12,55	12,92	14,99	13,42
% de estudantes mulheres relativo ao total de mulheres	23,30	24,79	29,25	26,12
% de estudantes homens relativo ao total de homens	25,74	26,90	31,15	28,35
% de estudantes relativo ao total da população	24,49	25,80	30,17	27,18

No Gráfico 10 nota-se que em 1977 tanto homens como mulheres dispunham de pouco tempo de escolaridade - mais de três quartos da população ou era analfabeto ou possuía no máximo o fundamental incompleto. Nessa época, nos três níveis de instrução superiores a esse os homens tinham índices maiores que as mulheres. O grau de instrução (**GRAU_INSTR**) da população vai aumentando e em 1987, o grau de escolarização feminino é levemente superior ao masculino nas categorias “fundamental completo / médio incompleto” e “médio completo / superior incompleto”. Na categoria “superior completo” o grau de instrução masculino é um pouco superior, situação que se inverte em 2007. Neste último ano de análise, as mulheres apresentam maiores percentuais nos dois níveis de maior grau de instrução.

Gráfico 10 – Distribuição da variável “GRAU_INSTR”, por ano e por sexo



A elevação do grau de instrução entre 1977 e 2007 influencia não apenas as viagens motivo trabalho (por eventual aumento da empregabilidade) mas também as viagens motivo escola, realizadas por um contingente de pessoas cada vez maior, mais diverso e contendo mais faixas etárias. A maior participação feminina no mercado de trabalho além de impactar as rendas (individual e familiar) deve influenciar bastante as viagens motivo trabalho.

A renda individual (variável **REN_IND**) tem seu comportamento de médias e medianas análogo ao da renda familiar. Explora-se aqui, então, as rendas individuais de quem tem 10 anos ou mais⁴, segmentando por sexo e por duas situações familiares: pessoa responsável e cônjuge/companheiro(a). Em todos os anos, a renda individual média masculina é superior à feminina para a mesma situação familiar, conforme Tabela 28. Em 1977, quando pessoa responsável pela família, o homem ganhava 2,5 vezes mais que a mulher em mesma condição. Essa marca vem caindo (cada vez mais devagar) até que, em 2007, eles ganham 1,5 vezes mais do que elas. Para a situação de cônjuges, a relação entre as rendas médias masculina e feminina giram em torno de dois, oscilando para 1,8 em 1987 e 2,4 em 1997. Ou seja, homens cônjuges em geral ganham próximo do dobro das mulheres cônjuges, e essa situação pouco se alterou com o passar das décadas.

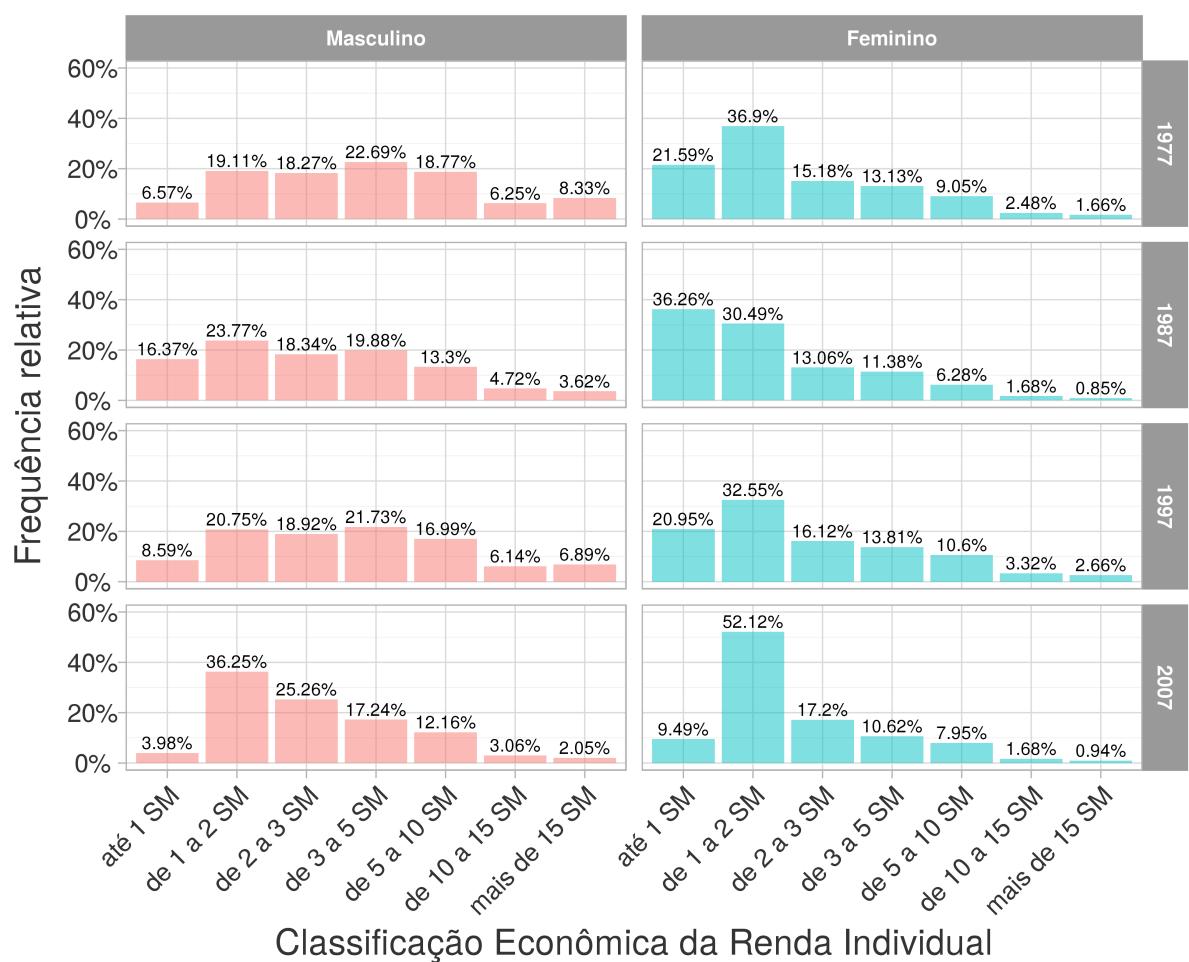
Tabela 28 – Estatísticas da variável “REN_IND”

Homem		pessoa responsável			cônjuge/companheiro		
ANO	Média	Desvio Padrão	Mediana	Média	Desvio Padrão	Mediana	
1977	2.762,85	3.910,71	1.478,21	703,24	1.461,49	0,00	
1987	1.610,76	2.146,33	985,80	522,67	953,85	99,64	
1997	1.769,16	3.272,09	873,61	1.143,30	2.115,36	485,34	
2007	1.265,37	2.330,39	600,00	1.147,71	2.396,79	300,00	
Total	1.869,37	3.059,03	970,68	1.098,29	2.280,25	291,73	
Mulher		pessoa responsável			cônjuge/companheira		
ANO	Média	Desvio Padrão	Mediana	Média	Desvio Padrão	Mediana	
1977	1.098,95	2.108,50	506,81	306,95	1.097,29	0,00	
1987	807,79	1.398,68	397,99	284,87	872,94	0,00	
1997	1.034,83	1.923,79	465,93	478,78	1.332,07	0,00	
2007	866,43	1.661,52	380,00	498,39	1.204,76	0,00	
Total	926,53	1.752,55	388,27	383,26	1.131,58	0,00	

Pelo Gráfico 11 percebe-se que houve um aumento na quantidade de pessoas que ganhava até 1 salário mínimo entre 1977 e 1987. De 1987 para 2007 a proporção de pessoas que têm rendimentos nessa faixa salarial vem decrescendo. De 1977 a 1997 a faixa de rendimento de 1 a 2 salários mínimos ficou próxima dos 25% e, em 2007, cresceu aproximadamente 10 pontos percentuais. Já a faixa de rendimento de 2 a 3 salários mínimos subiu cerca de 3 pontos percentuais entre 1977 e 2007, o mesmo que a faixa de rendimentos de 3 a 5 salários mínimos diminuiu no mesmo período.

⁴ Foi adotada essa idade como limite de corte porque o IBGE produz estatísticas de “valor do rendimento nominal médio mensal e mediano mensal para pessoas de 10 anos ou mais de idade, total e com rendimento”. Fonte: <<http://www.sidra.ibge.gov.br/bda/tabela/listabl.asp?c=1381&z=cd&o=7>> Acesso em 15 de janeiro de 2016

Gráfico 11 – Distribuição da variável “FAIXA_REN_IND”, por ano



6.2 Análises Preliminares

O grupo de análises que se segue busca compreender como é o comportamento das pessoas e das famílias em termo de viagens realizadas, em cada ano e diferencialmente entre os anos, olhando para tanto variáveis como:

- número de viagens;
- modos de viagem;
- motivos de viagens;
- duração das viagens;
- distâncias das viagens.

A variável **TOT_VIAG** representa o número total de viagens realizadas pela pessoa. Não existem *missing values* neste campo, os valores mínimos para todos anos e ambos sexo são 0, bem como também são 0 os valores do primeiro quartil (25%). Os valores das demais estatísticas (sem considerar os fatores de expansão) estão apresentados na Tabela 29. Conforme já era de se esperar, para quem faz viagem no dia da pesquisa (número de viagem é não nulo) existe a predominância do valor 2, ou seja, são pessoas que saem de suas residências com um propósito único (trabalhar, estudar, fazer compras) e depois retornam à residência após a atividade. Considerando toda a amostra, mesmo quem não fez viagem, percebe-se que, independente do sexo, o número médio de viagens por pessoa em relação a 1977 caiu um pouco em 1987 e 1997 (de 1,74 para 1,66) e subiu novamente em 2007 (para 1,86). Os desvios padrão caíram ao longo do tempo, indicando tendência de menor dispersão dos dados. Os valores de assimetria são positivos, indicando maior concentração à esquerda e cauda longa à direita da distribuição. Os valores de curtose evidenciam não tratar-se de distribuição normal.

Analizando esses dados segmentados por sexo, vê-se que as medianas são iguais. Os números médio e máximo de viagens para mulheres são sempre inferiores ao dos homens, para o mesmo ano. Os valores de assimetria para o sexo feminino e o masculino são positivos e convergem para o valor geral com o passar das décadas. A diferença entre o número médio de viagens de mulheres e homens vem diminuindo.

Ao considerar apenas quem relatou viagens no dia da pesquisa, os valores mínimos para todos anos e ambos sexo passam para 1, bem como também são 1 os valores do primeiro quartil (25%). Os valores das demais estatísticas (considerando os fatores de expansão para a população) estão apresentados na Tabela 30. Para todos os anos e para ambos sexos, as médias passaram para valores superiores a 2 e há uma pequena tendência de diminuição do número de viagens por pessoa com o passar do tempo. Os desvios padrão caíram ao longo do tempo para as mulheres e para o conjunto de homens e mulheres, indicando menor dispersão dos dados. Os valores de assimetria aumentaram e continuaram positivos, indicando maior concentração à esquerda e cauda longa à direita da distribuição.

Tabela 29 – Estatísticas da variável “TOT_VIAG”, por ano e por sexo

Grupo	ANO	Média	Desvio Padrão	Mediana	Máximo	Assimetria	Curtose
Geral	1977	1,74	1,82	2,00	24	1,52	4,91
	1987	1,65	1,62	2,00	24	1,35	4,71
	1997	1,66	1,63	2,00	21	1,36	4,34
	2007	1,86	1,61	2,00	18	1,24	4,00
Feminino	1977	1,40	1,64	2,00	16	1,40	3,32
	1987	1,43	1,59	2,00	19	1,38	3,92
	1997	1,53	1,64	2,00	18	1,45	4,61
	2007	1,75	1,64	2,00	17	1,23	3,48
Masculino	1977	2,10	1,92	2,00	24	1,56	5,55
	1987	1,89	1,61	2,00	24	1,41	5,91
	1997	1,79	1,60	2,00	21	1,29	4,22
	2007	1,98	1,58	2,00	18	1,29	4,80

Nota: Tabela elaborada considerando todas as pessoas da amostra, mesmo aquelas que não realizaram viagem, e sem realizar qualquer expansão

Os valores de curtose também aumentaram e a distribuição continua não sendo aderente à normalidade. Na segmentação por sexo, as medianas e a quantidade máxima de viagens permanecem iguais. Entretanto, o número médio de viagens para mulheres era inferior ao dos homens em 1977 e 1987. Já em 1997 em 2007, a média delas passa a ser superior à deles.

Foram feitos testes t para avaliar se as médias de mulheres e de homens eram estatisticamente diferentes, em cada ano, tanto considerando quem não fez viagem ($TOT_VIAG=0$), como desconsiderando esse caso. Os p-valores resultantes foram todos inferiores a 0,05, logo, rejeitou-se a hipótese nula de que as médias seriam iguais (nível de confiança de 95%). Foram feitos testes t para avaliar se as médias entre os anos eram diferentes, para o grupo de mulheres, tanto considerando quem não fez viagem ($TOT_VIAG=0$), como desconsiderando esse caso. Também aqui os p-valores resultantes foram todos inferiores a 0,05, logo, rejeitou-se a hipótese nula de que as médias seriam iguais (nível de confiança de 95%). Portanto, ao considerar o efeito de quem não sai de casa, o número médio de viagens das mulheres sempre é menor que o dos homens, para um dado ano, e realmente houve aumento no número médio de viagens das mulheres, da ordem de 0,1 viagem/década. Ao desconsiderar o efeito de quem não sai de casa, o número médio de viagens das mulheres tornou-se maior que o dos homens em 1997 e o número médio de viagens de homens e mulheres cai com o tempo.

Tabela 30 – Estatísticas da variável “TOT_VIAG”, por ano e por sexo, considerando apenas quem fez viagem

Grupo	ANO	Média	Desvio Padrão	Mediana	Máximo	Assimetria	Curtose
Geral	1977	2,82	1,52	2,00	24	2,65	11,39
	1987	2,62	1,27	2,00	24	3,95	14,38
	1997	2,61	1,30	2,00	21	2,83	12,28
	2007	2,64	1,28	2,00	18	2,73	11,08
Feminino	1977	2,67	1,31	2,00	16	2,48	11,19
	1987	2,61	1,25	2,00	19	2,81	15,66
	1997	2,63	1,31	2,00	18	2,92	12,39
	2007	2,65	1,29	2,00	17	2,64	12,18
Masculino	1977	2,93	1,65	2,00	24	2,61	11,09
	1987	2,62	1,30	2,00	24	3,05	15,66
	1997	2,59	1,29	2,00	21	2,75	11,39
	2007	2,62	1,28	2,00	18	2,82	12,18

Nota: Tabela elaborada considerando apenas as pessoas da amostra que realizaram viagem, e sem realizar qualquer expansão

O Gráfico 12 apresenta a distribuição de viagens (considerando os fatores de expansão para a população) até o limite de 6 viagens, corte feito apenas para melhorar a visualização do gráfico, já que a cauda é bastante longa. Deste gráfico, vale destacar a relação entre as viagens nulas (quem não sai de casa) e as viagens de ida e volta (valores iguais a 2). Para homens, o número de viagens nulo é menos frequente que o número de viagens de valor 2 para todos anos de análise. Já para as mulheres, em 1977 as viagens nulas eram a maioria, indicando certa fixitude delas na residência. Essa porcentagem vai diminuindo e a porcentagem no número de viagens igual a 2 vai crescendo, ficam próximas em 1997 e, em 2007, inverte-se a situação observada em 1977. Provavelmente devido à maior participação no mercado de trabalho, as mulheres ganharam mobilidade, restringindo-se menos ao espaço doméstico.

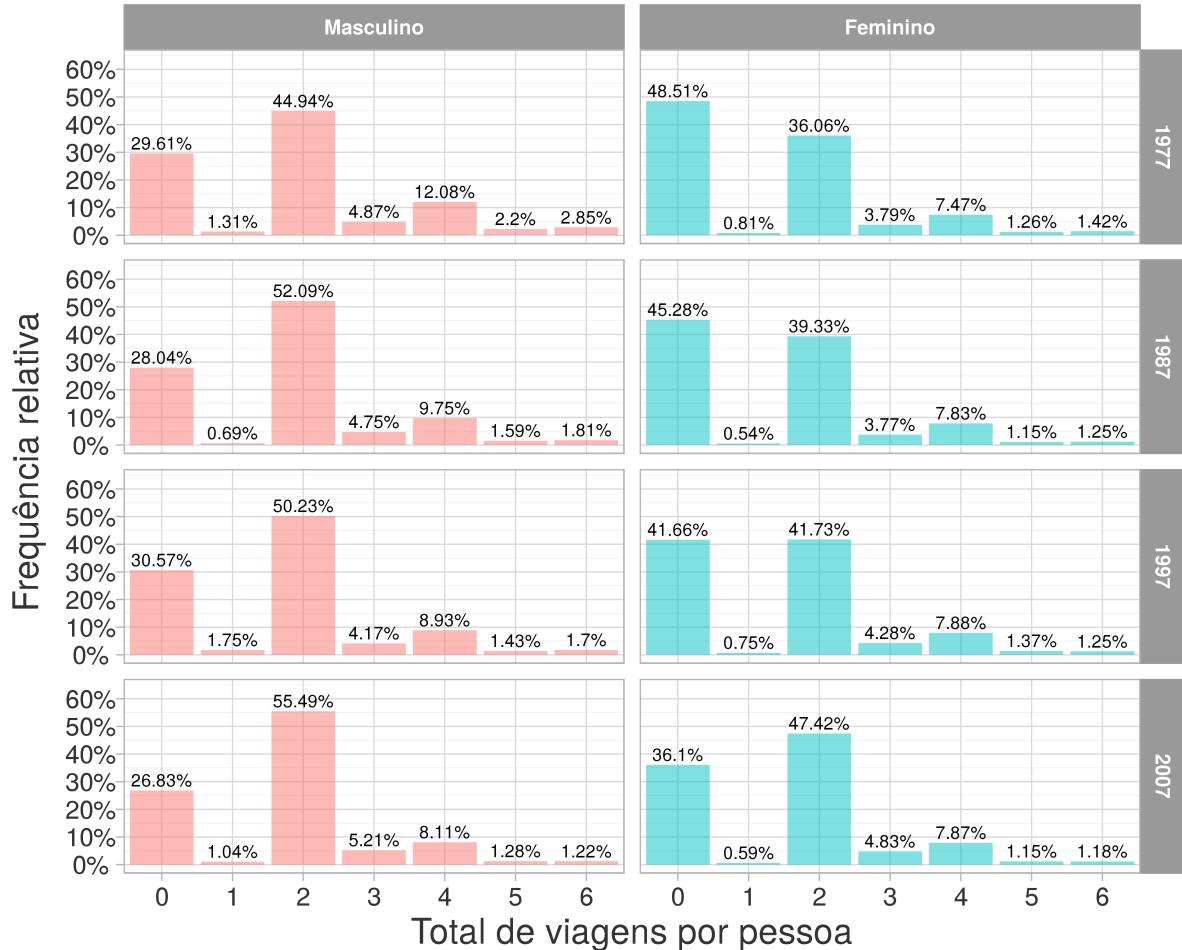
Tabela 31 – Média da variável “TOT_VIAG”, por ano e por sexo, com expansão

Grupo	ANO	Pessoas com ou sem viagem	Apenas Pessoas com viagem
Geral	1977	2,07	3,42
	1987	2,06	3,28
	1997	1,87	2,94
	2007	1,95	2,86
Feminino	1977	1,67	3,24
	1987	1,79	3,27
	1997	1,73	2,96
	2007	1,85	2,89
Masculino	1977	2,50	3,55
	1987	2,36	3,28
	1997	2,03	2,92
	2007	2,07	2,82

Nota: Dados obtidos pela expansão utilizada nos relatórios oficiais das Pesquisas OD, que consiste em calcular a média com base na equação $\frac{\sum(FE_VIAG)}{\sum(FE_PESS*F_PESS)}$

Ao focar no agregado da família, os valores das principais estatísticas (considerando os fatores de expansão para a população) da variável **FAM_VIAG_TOT** estão apresentados na Tabela 32. O número médio de viagens da família com ao longo do tempo, comportamento consistente tanto com a queda do número médio de viagens por pessoa (Tabelas 29 e 30) quanto com a diminuição do tamanho da família (Tabela 23). Aqui também os desvios padrão caem ao longo do tempo (tendência de menor dispersão de dados) e os valores de assimetria são positivos (maior concentração à esquerda e cauda longa à direita da distribuição). Os valores de curtose indicam que a distribuição não é normal. Ao retirar aquelas famílias com total de viagens nulo (ninguém fez viagem), não se observam diferenças nas tendências das estatísticas, à exceção da mediana que em 1997 era 7 em 1997 e passa para 6 ao inserir famílias cujo total de viagens é zero.

Gráfico 12 – Distribuição da variável “TOT_VIAG” por ano e por sexo



Nos campos MODO1, MODO2, MODO3 e MODO4 a categoria “ônibus de linha” inclui as categorias originais “ônibus trólebus”, “trólebus”, “ônibus diesel”, “ônibus”, “ônibus município de São Paulo”, “ônibus outros municípios” e “ônibus metropolitano”. A categoria “ônibus escolar/empresa” inclui também as categorias originais “ônibus fretado”, “escolar”, “transporte escolar”. A categoria “lotação/van” inclui as categorias originais “lotação/perua”, “microônibus/van município de São Paulo”, “microônibus/van outros municípios” e “microônibus/van metropolitano”. Vale lembrar que para os anos de 1977 e 1987 foram levantados no máximo três modos, e para os anos 1997 e 2007, no máximo quatro modos para cada viagem.

Na Tabela 33 foram agrupados em “alta capacidade” os modos metroferroviários (metrô e trem), em “ônibus” todos os tipos de ônibus (de linha, escolar, de empresas, lotações e vans), em “passageiro de automóvel” os passageiros de automóvel particular e também de táxis, em “Outros” as viagens realizadas por motocicletas e bicicletas pois estas foram diagnosticadas apenas para 2007. Nesta tabela estão apresentadas as frequências relativas destes agrupamentos para o **MODO1** (primeiro modo utilizado na viagem) e

Tabela 32 – Estatísticas da variável “FAM_VIAG_TOT”, por ano

Considerando famílias em que há pela menos uma viagem						
ANO	Média	Desvio Padrão	Mediana	Máximo	Assimetria	Curtose
1977	8,97	6,00	8,00	59	1,36	2,90
1987	8,04	5,14	7,00	50	1,39	3,43
1997	7,68	4,80	7,00	45	1,29	2,85
2007	7,11	4,31	6,00	68	1,52	6,84
Considerando inclusive família em que ninguém fez viagem						
ANO	Média	Desvio Padrão	Mediana	Máximo	Assimetria	Curtose
1977	8,58	6,14	8,00	59	1,27	2,62
1987	7,71	5,28	7,00	50	1,27	3,05
1997	7,22	5,00	6,00	45	1,14	2,39
2007	6,62	4,53	6,00	68	1,28	5,38

também para o **MODO2** (segundo modo utilizado), buscando avaliar a divisão modal por ano, para quem realizou viagem. Assim, para o primeiro modo o total soma 100% em todos anos, mas o total por ano do segundo modo em diante não necessariamente, porque nem todas viagens utilizaram mais de um modo.

Tabela 33 – Frequência relativa das variáveis “MODO1” e “MODO2”, por ano

MODO 1		Alta		Dirigindo	Passageiro		
ANO	Capacidade	Ônibus	Automóvel	de Automóvel	A pé	Outros	Total
1977	3,0%	41,7%	15,4%	11,3%	28,0%	0,8%	100%
1987	4,7%	30,5%	17,2%	9,7%	36,2%	1,7%	100%
1997	6,4%	28,9%	20,5%	10,8%	34,4%	1,3%	100%
2007	7,0%	31,8%	19,2%	8,6%	33,1%	2,9%	100%
MODO 2		Alta		Dirigindo	Passageiro		
ANO	Capacidade	Ônibus	Automóvel	de Automóvel	A pé	Outros	Total
1977	2,0%	7,0%	0,1%	0,2%	-	0,0%	9,2%
1987	3,6%	6,5%	0,1%	0,1%	-	0,0%	10,3%
1997	3,5%	5,9%	0,0%	0,1%	-	0,0%	9,5%
2007	4,2%	8,2%	0,1%	0,1%	-	0,0%	12,5%

Em relação à utilização do automóvel, sua proporção sobe cerca de 5% de 1977 até 1997 e cai pouco mais de 1% em 2007, talvez por conta dos congestionamentos cada vez mais frequentes e da evolução do sistema de transporte coletivo público da RMSP. Quando

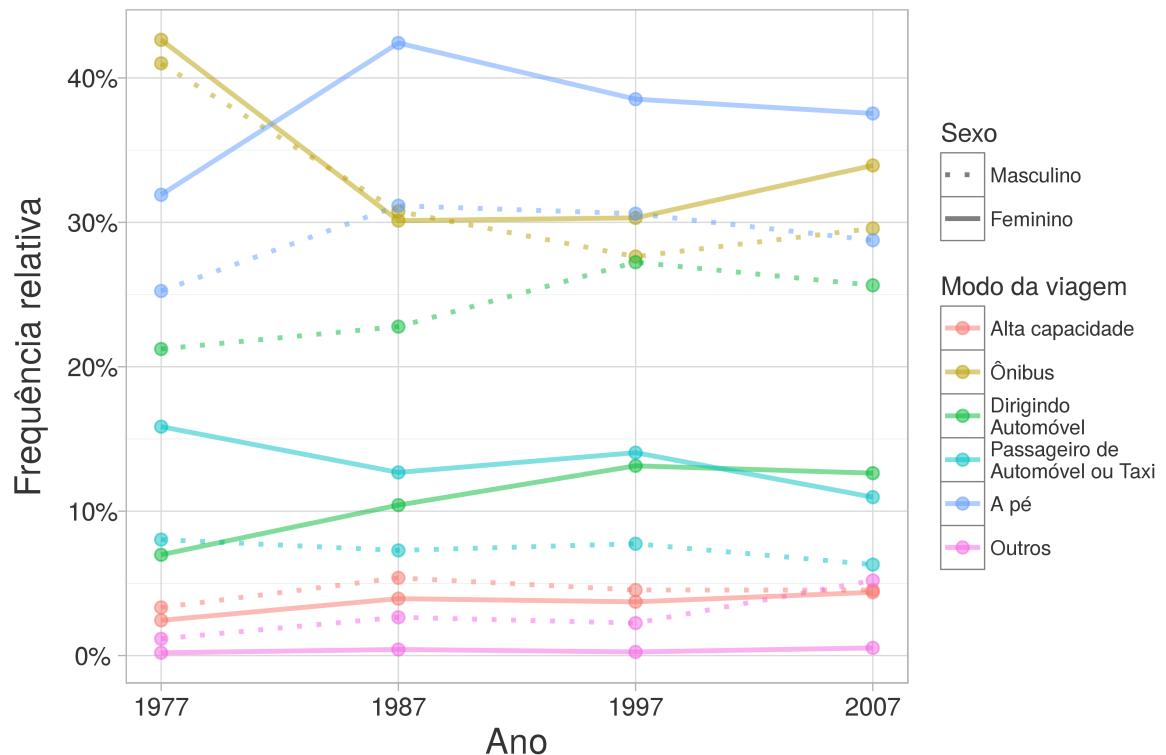
utilizado, o carro é quase sempre o único modo da viagem. Os ônibus têm uma queda de $\sim 13\%$ pontos percentuais entre 1977 e 1997, com recuperação de $\sim 3\%$ em 2007. Quem deixou de utilizar o ônibus nas primeiras 3 décadas passou a utilizar a caminhada como método de deslocamento ($\sim 6\%$), ou o carro ($\sim 5\%$) ou ainda o transporte coletivo de alta capacidade ($\sim 3\%$). Os modos metrô e trem vêm recebendo um incremento de viagens década a década, saindo de 3% em 1977 para 7% em 2077. A contribuição baixa deste modo, apesar de sua alta capacidade, deve-se provavelmente à cobertura insuficiente e pouca capilaridade no tecido urbano. Vale ressaltar que “alta capacidade” é o único modo que tem sua utilização ainda muito presente como segundo modo (relação MODO1/MODO2 $\sim 1,5$) - as viagens de ônibus são reduzidas pelo menos da ordem de um quarto, as de carro e outros caem a quase 0% no segundo modo. Isto significa que as viagens com metrô e trem são mais frequentemente precedidas de viagens com outro modo (principalmente ônibus), funcionando como tronco numa lógica de sistema tronco-alimentador. As viagens a pé aumentaram percentualmente entre 1977 e 1997 e caíram em 2007, talvez pelas distâncias necessárias de viagem dado o crescimento da área metropolitana de São Paulo. Conforme o conceito de viagem a pé anteriormente exposto, não haverá modo a pé nos modos 2, 3 ou 4. As tabelas de contingência dos modos 3 e 4 não serão apresentadas por serem pouco significativos no contexto geral: o modo 3 varia de 0,9% (em 1977) a 2,8% (2007) do total de viagens, sendo predominante o modo ônibus; o modo 4, só existente em 1997 e 2007, aparece em 0,4% das viagens realizadas.

Os Gráficos 13 e 14 apresentam a segmentação por sexo dos modos 1 e 2 de viagem, respectivamente. No primeiro modo de viagem, as viagens a pé são o modo mais frequente para as mulheres em 1987, 1997 e 2007, somente em 1977 o ônibus era mais frequentemente utilizado por elas. Para os homens, o modo mais frequente em 1977 e 2007 é o ônibus, e em 1987 e 1997, a pé. O modo outros é o menos frequente em 1977, 1987 e 1997 para ambos os sexos; em 2007, eles superam a alta capacidade, provavelmente porque o transporte por motocicleta e bicicleta passou a ser mais expressivo. Comparando ambos os sexos por categoria do primeiro modo:

- (i) As mulheres sempre fizeram mais viagens a pé que os homens: a diferença de 6,7 pontos percentuais de 1977 aumenta para 11,3 em 1987, cai para 7,9 em 1997 e sobe novamente em 2007 para 8,8%.
- (ii) A utilização feminina do ônibus é quase sempre superior à masculina, exceto em 1987, ano em que as proporções praticamente se igualam e desde quando a diferença vem aumentando com o tempo.
- (iii) É na utilização do automóvel em que residem as diferenças mais gritantes entre os gêneros: homens predominantemente motoristas e mulheres, passageiras. O destaque aqui reside no fato que entre 1997 e 2007, dentro do grupo de mulheres, elas passaram a dirigir mais do que ser passageiras de automóvel.
- (iv) A participação do transporte de alta capacidade gira em torno de 2,5% a 5,5% do

share modal, com o homem tendo uma utilização um pouco mais frequente dentro do seu grupo do que as mulheres, mais devido ao trem (cujos percentuais dos homens são sempre superiores aos das mulheres) do que ao Metrô (onde os percentuais das mulheres supera o dos homens a partir de 1997).

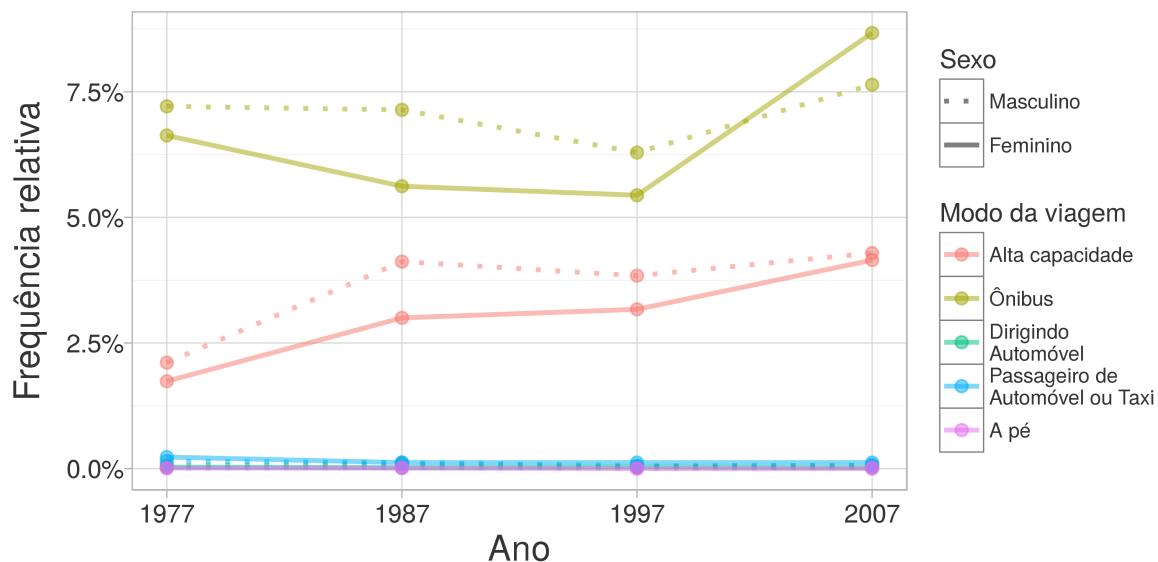
Gráfico 13 – Proporção das viagens do sexo feminino e do sexo masculino, segundo o primeiro modo da viagem, por ano



Analizando agora as categorias do segundo modo, por sexo:

- As duas categorias que apresentam relevância como segundo modo de transporte são os modos coletivos ônibus e alta capacidade (metrô e trem); os demais ou não foram considerados (por exemplo, a pé) ou não são muito significativos (automóvel e outros).
- A utilização feminina do ônibus, como segundo modo da viagem, é inferior à masculina entre 1977 e 1997, ao contrário do que ocorria com o primeiro modo. Em 2007 a situação se altera, quando pessoas do sexo feminino passam a utilizar mais frequentemente o ônibus.
- A frequência do uso de alta capacidade é mais expressiva no conjunto do modo 2, embora ainda seja menor que a do ônibus em todos os anos e para ambos sexos.
- Para as mulheres, de 1977 para 1987, parece ter havido uma troca modal no segundo modo: houve queda de ~1% no uso do ônibus e aumento também de ~1% no uso de alta capacidade. Foi nesse período em que houve a primeira expansão da rede metro-ferroviária para a zona leste (trecho Sé-Penha).

Gráfico 14 – Proporção das viagens do sexo feminino e do sexo masculino, de acordo com o segundo modo da viagem, por ano



É possível também analisar os modos agregando-os em “coletivo”, “individual” e “a pé”, o que já fora feito na variável **TIPO_VIAG**, cujas frequências relativas são apresentadas na Tabela 34. O transporte individual cresceu de 1977 a 1997 e recuou um pouco em 2007. O transporte coletivo decresceu entre 1977 e 1987, mas a uma taxa bem maior que o crescimento do individual, o que significa que essas viagens deixaram de ser feitas de transporte coletivo para, principalmente, serem feitas a pé ou, com menor frequência, de carro. Entre 1987 a 1997 tanto o transporte coletivo como o modo a pé sofrem ligeiras quedas (em torno de 2 pontos percentuais), período em que o transporte individual aumenta sua taxa de crescimento. Neste ano o cenário da divisão modal fica quase equitativamente dividido com cerca de um terço para cada uma das três categorias. Em 2007 a forma de deslocamento a pé sofre ligeira queda ($\sim 1\%$), o transporte individual também cai ($\sim 2,5\%$) e essas viagens passam a ser feitas pelo transporte coletivo que assume proporção um pouco superior à que tinha em 1987.

Tabela 34 – Frequência relativa da variável “TIPO_VIAG”, por ano

ANO	Coletivo	Individual	A pé
1977	45,0%	27,0%	28,0%
1987	35,6%	28,2%	36,2%
1997	33,3%	32,3%	34,4%
2007	36,5%	29,5%	33,1%

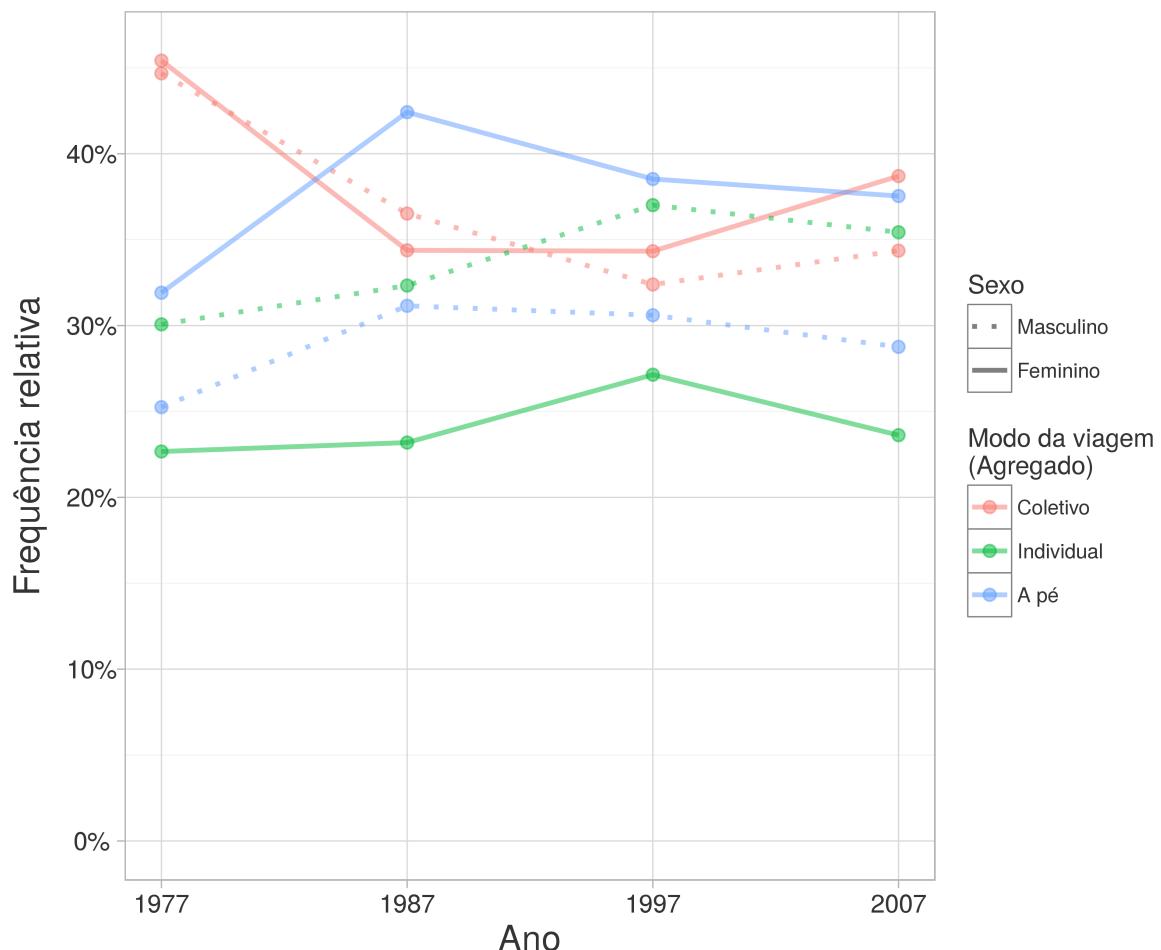
O Gráfico 15 apresenta a segmentação por sexo do modo (agregado) de viagem. Comparando ambos os sexos por categoria do primeiro modo:

- (i) Em 1977, 45,4% das mulheres usavam o transporte coletivo, 31,9% deslocavam-se a pé e 22,7% usavam transporte individual. Em 1987, para elas, o transporte individual permanece no mesmo patamar ($\sim 23,2\%$) e ocorre uma migração do coletivo para o a pé com 34,4% e 42,4%, respectivamente.
- (ii) Em 1977, 44,7% dos homens usavam o transporte coletivo, 30,1% o individual e 25,3% deslocavam-se a pé. Em 1987, para eles, o transporte individual cresceu ($\sim 32,3\%$) e as viagens a pé também ($\sim 31,2\%$) indicando uma migração do transporte coletivo (36,5%) para estes modos.
- (iii) Entre 1987 e 1997, a proporção de mulheres a usar o transporte coletivo permanece inalterada, mas ocorre uma migração do modo a pé para o transporte individual.
- (iv) Entre 1987 e 1997, a proporção de homens a usar o transporte individual continua aumentando (para 37%), superando a participação do coletivo (32,4%), enquanto as viagens a pé permanecem no mesmo patamar.
- (v) Entre 1997 e 2007, a proporção do uso feminino do transporte coletivo cresce (para 38,7%) indicando migração para este modo das viagens advindas, especialmente, do transporte individual (que cai para 23,6%) e, em menor medida, do modo a pé (com 37,5%).
- (vi) Entre 1997 e 2007, a proporção do uso masculino do transporte coletivo também cresce (para 34,4%) indicando migração para este modo das viagens advindas tanto do transporte individual (que cai para 35,4%) como do modo a pé (com 28,8%).

Para as variáveis de motivo (**MOTIVO_ORIG** e **MOTIVO_DEST**) foi criada a categoria “servir passageiro”. Para tanto, olhava-se a variável “servir passageiro no destino” de cada pesquisa OD; caso fosse afirmativo (1), a categoria adotada é “servir passageiro”, porque o que motiva esse deslocamento é o motivo de outrem, não o da pessoa respondente. Caso contrário, adota-se o motivo de origem indicado originalmente na base de dados. Tal procedimento foi realizado com as bases de 1997 e 2007. A base de 1977 já conta com a categoria “servir passageiro”. A base de 1987 é a única que não possui informações suficientes para identificar esse motivo de viagem.

Será explorada a variável motivo no destino porque essa variável indica a atividade fim que gerou o deslocamento. A Tabela 35 foi elaborada expandindo as viagens pelo **FE_VIAG**, consequentemente foram consideradas apenas as viagens realizadas (cujos **FE_VIAG** não são iguais a zero). Observa-se que o motivo “residência” corresponde à maior parte das viagens realizadas (cerca de 45%), se alterando pouco ao longo dos anos - resultado próximo ao esperado (pouco menos de 50%) dado que o comportamento de deslocamentos da demanda tem a residência como base, ou seja, é para onde a maior parte das pessoas retornam após executar alguma outra atividade. O segundo motivo mais

Gráfico 15 – Proporção das viagens do sexo feminino e do sexo masculino, segundo o modo da viagem (agregado), por ano



frequente é “trabalho” girando em torno dos 23,5% e também oscilando pouco (1%) ao longo dos anos, seguido por “educação”, que cresce de 1977 (13,2%) para 1987 (16,9%) e depois decresce em 1997 (14%) e se mantém em 2007 (14%). Assim, trabalho, educação e residência são os motivos de pouco mais de 80% das viagens em todos os anos. A proporção das viagens motivo “manutenção-compras” cresce de 1977 (3,9%) para 1987 (4,5%) e praticamente retorna ao mesmo patamar em 1997 (3,8%), caindo um pouco em 2007 (3,6%). O percentual de viagens motivo “lazer/outros” vem diminuindo com o tempo, cerca de 2 pontos percentuais por década. O percentual de viagens “servir passageiro”, por sua vez, vem aumentando com o tempo, saindo de 1,0% em 1977 para 7,0% em 2007.

Ao observar o Gráfico 16, que dentro de cada ano segmenta por sexo os motivos de viagens, percebe-se que as proporções das viagens motivo “trabalho” femininas são sempre inferiores às masculinas e essa diferença vem diminuindo com o tempo por conta da maior participação das mulheres no mercado de trabalho. As proporções das viagens motivo “educação” femininas são sempre superiores às masculinas e essa diferença vem diminuindo,

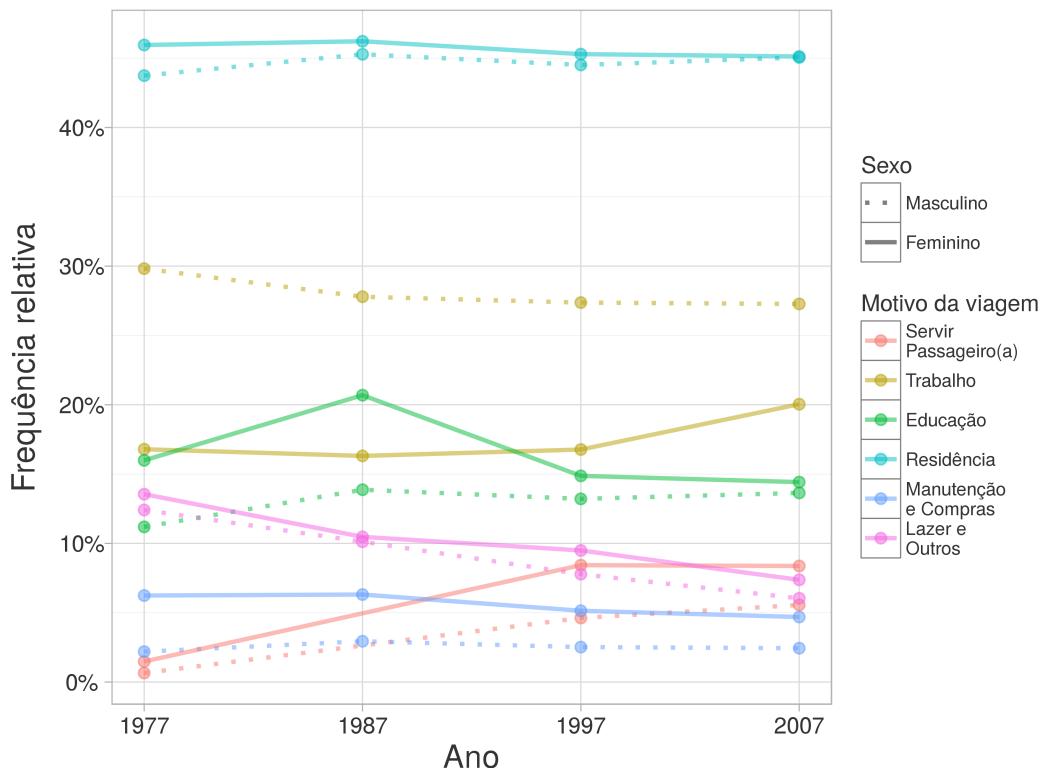
Tabela 35 – Frequência relativa da variável “MOTIVO_DEST”, por ano

ANO	Servir				Manutenção/ compras	Lazer/ Outros
	Passageiro	Trabalho	Educação	Residência		
1977	1,0%	24,4%	13,2%	44,6%	3,9%	12,9%
1987	-	22,6%	16,9%	45,7%	4,5%	10,3%
1997	6,4%	22,3%	14,0%	44,9%	3,8%	8,6%
2007	7,0%	23,7%	14,0%	45,0%	3,6%	6,7%

de forma que em 2007 não chega a 1%. As viagens motivo “lazer / outros” cai em ambos sexos, sendo as porcentagens das viagens femininas superiores às masculinas em todos os períodos. As viagens motivo “manutenção / compras” são sempre mais frequentes para mulheres do que para homens e a diferença entre ambos caiu de 4,05 ponto percentuais em 1977, quando as mulheres faziam 2,8 vezes mais viagens deste tipo do que os homens, para 2,25 pontos percentuais em 2007, quando as mulheres passaram a fazer quase o dobro (1,9 vezes) deste tipo de viagem que os homens. As viagens motivo “servir passageiro” são menos representativas do total para ambos sexos e sempre mais frequentes para mulheres do que para homens. Excluindo 1987, cujos dados não estavam disponíveis nesta categoria, a relação entre o percentual feminino e o masculino era de 2,3 em 1977, passou para 1,8 em 1997 e para 1,5 em 2007.

A variável **DURACAO** tem suas principais estatísticas apresentadas na Tabela 36. Não existem *missing values* neste campo e, desconsiderando quem não fez viagem (duração igual a 0 minutos), o valor mínimo para todos anos e ambos os sexos é 1 minuto. As medianas da duração, independente do sexo, saem de 30 minutos em 1977 para 20 minutos em 1987 e 1997 e retornam para o valor 30 em 2007. Os valores de assimetria são todos positivos, indicando maior concentração à esquerda e cauda longa à direita da distribuição. Os valores de curtose evidenciam não se tratar de distribuição normal. O tempo médio geral de viagem decresce entre 1977 e 1987 e daí em diante só aumenta, comportamento semelhante ao segmento feminino. Os tempos médios das viagens do homens cresce sistematicamente década a década, da ordem de 1 minuto entre 1977, 1987 e 1997. Já 2007 o acréscimo no tempo médio de viagem masculino subiu 4,5 minutos - tal efeito pode ser explicado pela expansão urbana da RMSP. Analisando esses dados segmentados por sexo, vê-se que as medianas das mulheres são sempre 5 minutos a menos que as dos homens. A diferença entre as durações médias das viagens de mulheres e de homens cresce de quase 0,5 minuto em 1977 para pouco mais de 5,5 minutos em 1987 e vem diminuindo desde então. Foram feitos teste t para avaliar se eram estatisticamente significantes (intervalo de confiança de 95%) as médias entre os sexos, no mesmo ano; e entre os anos, para o mesmo sexo. Todas médias foram estatisticamente diferentes umas das outras.

Gráfico 16 – Proporção das viagens do sexo feminino e do sexo masculino, segundo o motivo no destino, por ano



Com o intuito de melhor explorar as durações médias das viagens analisando modos e motivos, foram elaborados os Gráficos 17, 18 e 19. O Gráfico 17 apresenta as durações médias do transporte coletivo, individual e a pé, para homens e para mulheres. Verifica-se que:

- As durações médias de homens e mulheres são muito próximas para as viagens a pé, girando em torno de 15 minutos para ambos.
- A duração média no transporte individual cai um pouco de 1977 para 1987 (de 23,0 para 20,4 min para mulheres e de 27,07 para 25,7 min para homens).
- A duração média no transporte individual aumenta de 1987 para 2007 (7,3 min para mulheres e 8,1 min para homens).
- A duração média no transporte coletivo aumenta entre 1977 e 2007 para ambos sexos, sendo a taxa mais acentuada de 1997 para 2007.
- As diferenças nas durações médias nas viagens feitas por transporte coletivo entre mulheres e homens aumenta de 1977 (4,2 min) para 1987 (6,3 min) e depois decresce nas décadas seguintes (6,0 min em 1997 e 3,2 min em 2007).

O Gráfico 18 apresenta as durações médias dos modos de transporte coletivo (ônibus de linha, ônibus de empresa/escolar, lotação/perua/van/microônibus, Metrô e trem), para homens e para mulheres - foram separados os modos de alta capacidade dos demais para

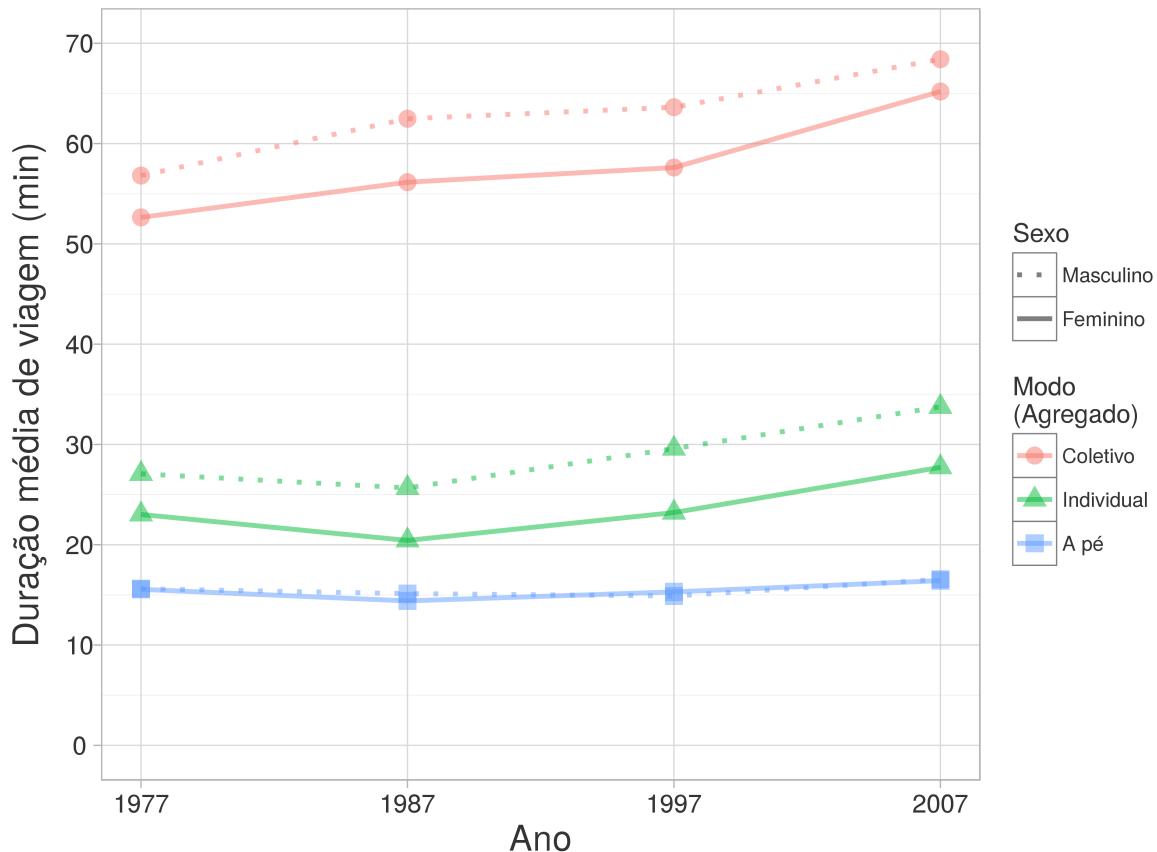
Tabela 36 – Estatísticas da variável “DURACAO”, por ano

Total						
ANO	Média	Desvio Padrão	Mediana	Máximo	Assimetria	Curtose
1977	36,07	31,99	30	240	1,74	3,78
1987	33,27	31,99	20	360	1,92	4,97
1997	34,13	33,52	20	370	1,94	4,54
2007	39,29	37,22	30	299	1,74	3,37
Sexo feminino						
ANO	Média	Desvio Padrão	Mediana	Máximo	Assimetria	Curtose
1977	34,09	30,54	25	240	1,81	4,12
1987	30,15	29,76	20	360	2,08	5,89
1997	31,97	31,67	20	315	2,02	4,78
2007	37,96	36,72	25	299	1,81	3,68
Sexo Masculino						
ANO	Média	Desvio Padrão	Mediana	Máximo	Assimetria	Curtose
1977	34,47	32,89	30	240	1,69	3,53
1987	35,83	33,50	25	360	1,80	4,36
1997	36,11	35,02	25	370	1,86	4,25
2007	40,61	37,67	30	270	1,67	3,11

facilitar a compreensão dos gráficos. Verifica-se que:

- (i) A duração média do trem é superior a todos outros modos, inclusive o Metrô, com médias gerais caindo de 83,3 min em 1977 para 80,8 min em 1987 e crescendo para 89,5 min em 1997. Em 2007, esse valor permanece no mesmo patamar de 1997 (91,2 min).
- (ii) Para o trem, a duração média feminina é inferior à masculina em 1977 (diferença de 2,9 min) e 1987 (diferença de 2,1 min), supera a masculina em 1997 (por 1,5 min) e distancia da masculina em 2007 (diferença de 8,0 min).
- (iii) A duração média das viagens de Metrô crescem sistematicamente, pelo menos 6 min por década, de 1977 (53,8 min) a 2007 (74,4 min).
- (iv) Para o Metrô, a duração média feminina é superior à masculina em 1977 (diferença de 1,3 min) e em 2007 (diferença de 1,7 min). A situação é inversa, com durações médias das viagens masculinas superiores às femininas em 1987 (diferença de 3,8 min) e 1997 (diferença de 5,0 min).
- (v) A duração média das viagens dos ônibus de linha crescem sistematicamente, 5 min entre 1977 e 1987, 0,9 min entre 1987 e 1997, e 8,8 min entre 1997 e 2007.
- (vi) Para ônibus de linha, a duração média feminina é inferior à masculina - a diferença é

Gráfico 17 – Durações médias de viagem por ano e por sexo, segundo os modos (agregados)



de 4,0 min em 1977, de 6,9 min em 1987, de 6,3 min em 1997 e de 3,9 min em 2007.

(vii) A duração média das viagens de lotações/vans crescem de 1977 (54,1 min) para 1987 (60,8 min), caem em 1997 (49,9 min) e sobem novamente em 2007 (66,3 min).

(viii) Para viagens de lotações/vans, a duração média feminina era inferior à masculina em 1977 (diferença de 6,4 min) e em 2007 (diferença 9,4 de min). A situação é inversa em 1987, com diferenças de 7,7 min entre os sexos. E, em 1997, as durações médias são estatisticamente iguais.

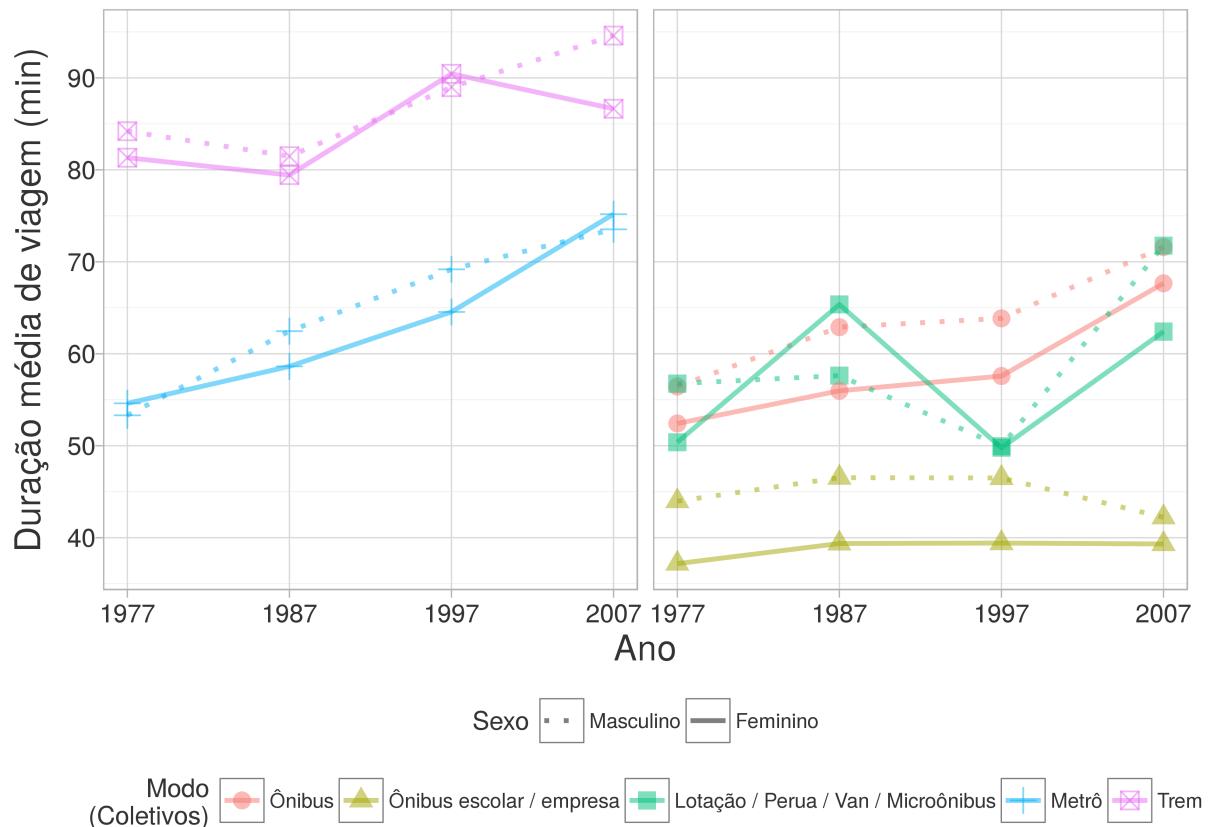
(ix) A duração média das viagens de ônibus escolar/fretado são as menores entre os transportes coletivos, para ambos sexos variando numa faixa de 40,9 a 44,2 min.

(x) Para viagens de ônibus escolar/fretado, a duração média feminina é sempre inferior à masculina - a diferença é de 6,8 min em 1977, de 7,2 min em 1987, de 7,1 min em 1997 e de 2,9 min em 2007.

O Gráfico 19 apresenta as durações médias por motivo de viagem (trabalho, educação, servir passageiro, manutenção/compras e lazer/outros), para homens e para mulheres. Verifica-se que:

(i) A duração média da viagem motivo trabalho (de natureza compulsória) é superior a de todos outros motivos. Ela permanece no mesmo patamar entre 1977 (40,9 min) e 1987

Gráfico 18 – Durações médias de viagem por ano e por sexo, segundo os modos coletivos



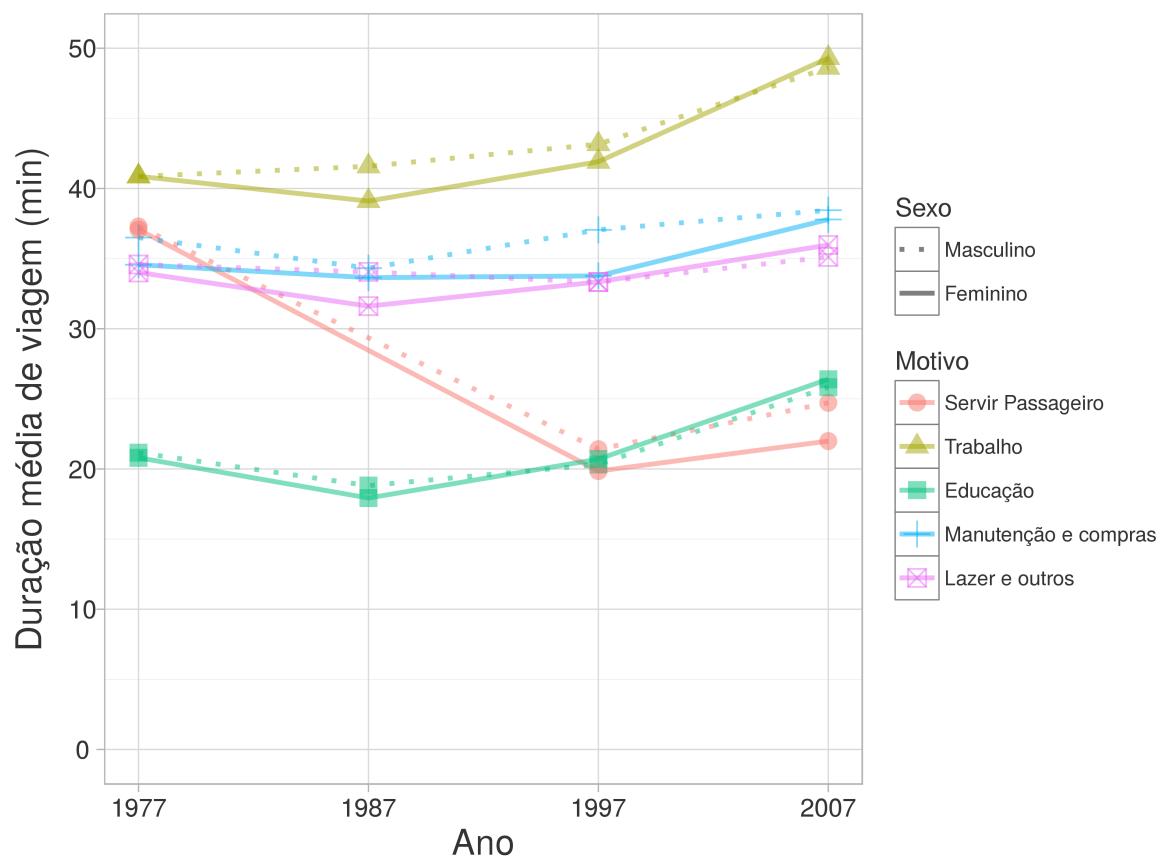
- (40,8 min), sobe um pouco em 1997 (42,7 min) e continua a subir em 2007 (48,9 min).
- (ii) Para o motivo trabalho, a duração média feminina é estatisticamente igual à masculina em 1977, inferior, em 1987 (diferença de 2,5 min) e em 1997 (diferenças de 1,3 min) e levemente superior em 2007 (0,7 min).
- (iii) A duração média das viagens motivo educação decresce de 1977 (20,9 min) para 1987 (18,3 min), a partir de quando sobem até 2007 (26,1 min).
- (iv) Para o motivo educação, a duração média feminina é praticamente igual à masculina em 1977 e em 1997 (diferenças de menos de 0,4 min), levemente inferior à masculina em 1987 (diferença de 0,9 min) e superior em 2007 (diferença de 0,6 min).
- (v) A duração média das viagens motivo manutenção/compras decresce de 1977 (35,2 min) para 1987 (33,9 min), a partir de quando sobem até 2007 (38,0 min).
- (vi) Para viagens motivo manutenção/compras, a duração média feminina é inferior à masculina - a diferença é de 1,9 min em 1977, de 0,7 min em 1987, de 3,3 min em 1997 e de 0,7 min em 2007.
- (vii) A duração média das viagens motivo lazer/outros decresce de 1977 (34,3 min) para 1987 (32,9 min), a partir de quando sobem até 2007 (35,6 min).
- (viii) Para o motivo lazer/outros, a duração média feminina é praticamente igual à

masculina em 1997, sendo antes disso inferior à masculina (diferenças de 0,6 e 2,4 min para 1977 e 1987, respectivamente) e depois disso superior à masculina (diferença de 0,9 min em 2007).

(ix) A duração média das viagens motivo servir passageiro decresce de 1977 (37,2 min) para 1997 (20,4 min), a partir de quando sobem até 2007 (23,1 min). Vale destacar que em 1987 não havia o modo servir passageiro.

(x) Para viagens motivo servir passageiro, a duração média feminina é estatisticamente igual à masculina em 1977, e inferior em 1997 (diferenças de 1,6 min) e em 2007 (diferenças de 2,7 min).

Gráfico 19 – Durações médias de viagem por ano e por sexo, segundo o motivo da viagem



Em “Estatísticas sob Suspeita”, Carrasco (2012, p.100-101) propõe indicadores com base na experiência das mulheres e, especificamente, no capítulo relativo ao acesso à mobilidade e ao planejamento territorial, recomenda a formulação e consideração dos seguintes indicadores quando da formulação de políticas públicas: motivos dos deslocamentos, meio utilizado nos deslocamentos e distâncias dos deslocamentos, entre outros. Anteriormente foram apresentados os panoramas de motivos e modos, agora, serão exploradas as distâncias de deslocamento.

A variável de distância da viagem (**DIST_VIAG**) contém a distância euclidiana entre as coordenadas de origem e as coordenadas de destino, com as limitações que a determinação dessas coordenadas impõem, já que foram calculadas a partir dos centroides das subzonas ou zonas (na ausência das subzonas). Ou seja, para 1977, as distâncias representadas são apenas as inter-zonais, pois não haviam dados disponíveis nem de subzonas nem de coordenadas. Para 1987 e 1997, cujos mapas de subzonas foram disponibilizados, as distâncias representadas são as inter-subzonas. Ainda para 1997, diversas distâncias de viagens não foram possíveis de ser calculadas porque o *shapefile* obtido junto ao Metrô-SP não continha todas subzonas indicadas no banco de dados da Pesquisa OD correspondente. E para 2007, mediante a disponibilidade de melhores recursos tecnológicos, as distâncias apresentadas são mais precisas e obtidas diretamente das coordenadas. Assim, os valores, análises e eventuais resultados decorrentes das distâncias de viagem precisam ser olhados com parcimônia. As principais estatísticas desta variável, cuja unidade é quilômetros, são apresentadas na Tabela 37.

Não existem *missing values* neste campo e, considerando apenas viagens com distância superior a zero, independentemente do sexo, os valores mínimos são: 0,37 km para 1977, 0,35 km para 1987, 0,23 km para 1997 e 0,01 para 2007. As medianas da distância de viagem, também independente do sexo, crescem ao longo do período analisado, saindo de 5,62 km em 1977 para 5,79 km em 2007. Os valores de assimetria são todos positivos, indicando maior concentração à esquerda e cauda longa à direita da distribuição. Os valores de curtose evidenciam não se tratar de distribuição normal. A distância média geral de viagem sempre aumenta no período analisado: 0,5 km de 1977 para 1987, 0,13 km de 1987 para 1997 e 0,44 km de 1997 para 2007.

Analizando esses dados segmentados por sexo, observa-se que as distâncias médias das viagens do homens cresce década a década e é sempre superior às das mulheres. As distâncias médias das viagens das mulheres também cresce no período saindo de 6,64 km em 1977 e chegando a 7,80 km em 2007, valor inferior média masculina em 1977. As diferenças entre os valores médios de mulheres e homens gira em torno de 1,5 km: 1,2 km para 1977, 1,6 km para 1987, 1,7 km para 1997 e 1,3 km para 2007. Foram feitos teste t para avaliar se as médias entre os sexos, para o mesmo ano, e entre os anos, para o mesmo sexo, eram diferentes. Com um intervalo de confiança de 95%, todos o p-valores obtidos indicavam ser possível rejeitar a hipótese nula de que a diferença entre as médias eram iguais a zero.

Com o intuito de melhor explorar as distâncias médias das viagens analisando modos e motivos, foram elaborados os Gráficos 20, 21 e 22. O Gráfico 20 apresenta as distâncias médias de viagem do transporte coletivo, individual e a pé, para homens e para mulheres. Verifica-se que:

- (i) As distâncias médias de homens e mulheres são muito próximas para as viagens a pé

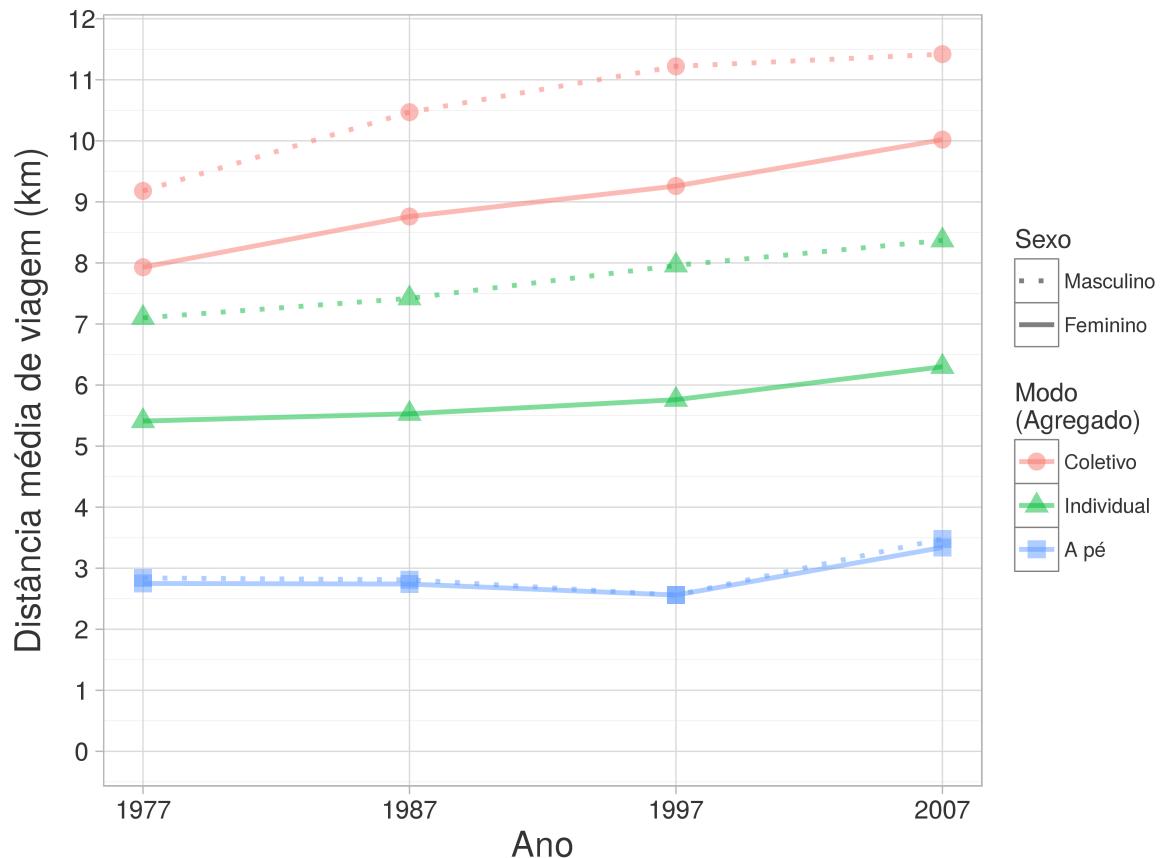
Tabela 37 – Estatísticas da variável “DIST_VIAG”, por ano

Total						
ANO	Média	Desvio Padrão	Mediana	Máximo	Assimetria	Curtose
1977	7,40	6,06	5,62	70,38	1,91	5,71
1987	7,90	6,92	5,67	107,33	1,90	5,31
1997	8,03	7,64	5,69	82,27	1,86	4,85
2007	8,47	8,15	5,79	84,10	1,73	4,05
Sexo feminino						
ANO	Média	Desvio Padrão	Mediana	Máximo	Assimetria	Curtose
1977	6,64	5,47	5,04	65,91	2,00	6,62
1987	6,92	6,26	4,82	107,33	2,09	6,75
1997	7,09	6,89	4,90	82,27	1,95	5,27
2007	7,80	7,73	5,16	82,67	1,80	4,42
Sexo Masculino						
ANO	Média	Desvio Padrão	Mediana	Máximo	Assimetria	Curtose
1977	7,89	6,37	6,08	70,38	1,83	5,15
1987	8,57	7,26	6,31	81,19	1,78	4,59
1997	8,82	8,13	6,36	81,05	1,76	4,34
2007	9,08	8,48	6,44	84,10	1,66	3,71

(~ 75 m), sequer sendo estatisticamente significativa a diferença em 1997.

- (ii) A distância média no transporte individual sobe em taxas crescentes a cada década (0,2 km de 1977 para 1987, 0,3 km de 1987 para 1997, e 0,5 km de 1997 para 2007).
- (iii) As diferenças nas distâncias médias de viagem feitas por transporte individual entre mulheres e homens aumenta de 1977 (1,7 km) para 1987 (1,9 km) e para 1997 (2,2 km), depois decresce em 2007 (2,1 km).
- (iv) A distância média no transporte coletivo aumenta entre 1977 e 2007 para ambos sexos, sendo que o ritmo de crescimento vem diminuindo: diferença de 1,1 km entre 1977 e 1987, diferença de 0,5 km entre 1987 e 1997 e diferença de 0,4 km entre 1997 e 2007.
- (v) As diferenças nas distâncias médias de viagem feitas por transporte coletivo entre mulheres e homens aumenta de 1977 (1,3 km) para 1987 (1,7 km) e para 1997 (2,0 km), depois decresce em 2007 (1,4 km).

Gráfico 20 – Distâncias médias de viagem por ano e por sexo, segundo os modos (agregados)

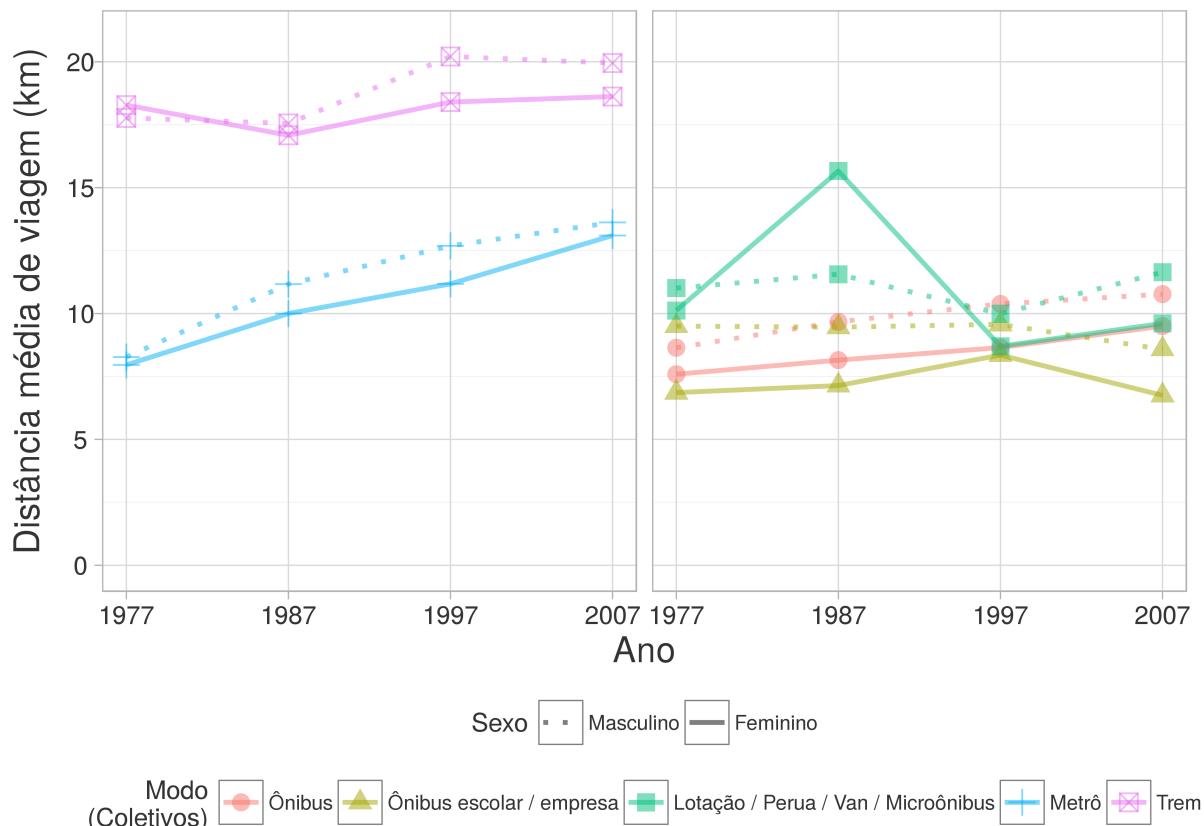


O Gráfico 21 apresenta as distâncias médias dos modos de transporte coletivo (ônibus de linha, ônibus de empresa/escolar, lotação/perua/van/microônibus, Metrô e trem), para homens e para mulheres - foram separados os modos de alta capacidade dos demais para facilitar a compreensão dos gráficos. Verifica-se que:

- As distâncias médias de viagem de trem são superiores às de todos outros modos, inclusive o Metrô, com médias gerais caindo de 17,9 km em 1977 para 17,4 km em 1987, crescendo para 19,6 km em 1997 e tornando a cair em 2007 (19,4 km).
- Para o trem, a distância média feminina só é superior à masculina em 1977 (diferença de 0,5 km), em 1987, 1997 e 2007 as médias delas são inferiores às deles (diferenças de 0,5 km, 1,8 km e 1,3 km, respectivamente).
- As distâncias médias de viagem de Metrô crescem década a década, saindo de 8,2 km em 1977, passando por 10,7 km em 1987, 12,0 km em 1997 e chegando a 13,4 km em 2007.
- Para o metrô, a distância média de viagem feminina é inferior à masculina, com diferenças de 0,3 km em 1977, 1,2 km em 1987, 1,5 km em 1997 e 0,5 km em 2007.
- As distâncias médias de viagem de ônibus de linha crescem sistematicamente, 0,8 km entre 1977 e 1987, 0,5 km entre 1987 e 1997, e 0,6 km entre 1997 e 2007.

- (vi) Para ônibus de linha, as distâncias médias de viagem femininas são inferiores às masculinas, com diferenças de 1,1 km em 1977, 1,5 km em 1987, 1,7 km em 1997 e 1,3 km em 2007.
- (vii) As distâncias médias de viagem de lotações/vans oscilam em torno de 10,9 km: sobem de 10,7 km em 1977 para 13,2 km em 1987, caem para 9,3 km em 1997 e tornando a aumentar em 2007 (10,5 km).
- (viii) Para lotações/vans, as distâncias médias de viagem femininas são inferiores às masculinas exceto em 1987, com diferenças de 0,9 km em 1977, 4,1 km em 1987 (a mais para mulheres), 1,3 km em 1997 e 2,0 km em 2007.
- (ix) As distâncias médias de viagem de ônibus escolar/fretado entre 1977 e 1987 não são estatisticamente diferentes. Há um crescimento nos valores de 1987 (8,8 km) para 1997 (9,1 km) e queda em 2007 (7,8 km).
- (x) Para ônibus escolar/fretado, as distâncias médias de viagem femininas são inferiores às masculinas, com diferenças de 2,6 km em 1977, 2,3 km em 1987, 1,2 km em 1997 e 1,8 km em 2007.

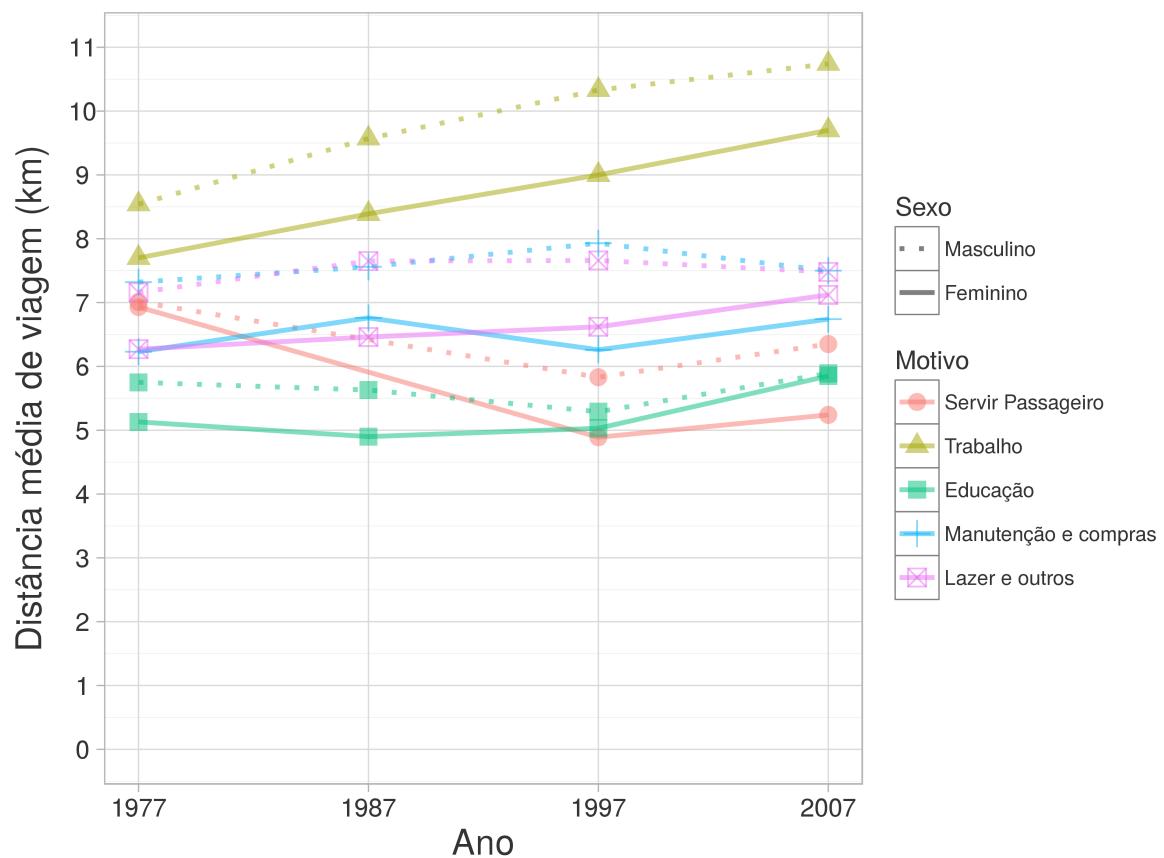
Gráfico 21 – Distâncias médias de viagem por ano e por sexo, segundo os modos coletivos



O Gráfico 22 apresenta as distâncias médias por motivo de viagem (trabalho, educação, servir passageiro, manutenção/compras e lazer/outros), para homens e para mulheres. Verifica-se que:

- (i) As distâncias médias de viagem motivo trabalho (de natureza compulsória) são superiores a de todos outros motivos. Elas crescem ano a ano: 8,3 km em 1977, 9,2 km em 1987, 9,9 km em 1997 e 10,3 km em 2007.
- (ii) Para viagens motivo trabalho (natureza compulsória), as distâncias médias femininas são sempre inferiores às masculinas, com diferenças girando em torno de 1,1 km: 0,8 km em 1977, 1,2 km em 1987, 1,3 km em 1997 e 1,0 km em 2007.
- (iii) As distâncias médias de viagem motivo educação são as mais baixas frente ao demais motivos, exceto em 2007 quando servir passageiro assume média menor. Quando o motivo é escola, as distâncias caem de 1977 (5,44 km) para 1987 (5,23 km), e de 1987 para 1997 (5,16 km). Em 2007, os valores sobem novamente (5,87 km).
- (iv) Para viagens motivo educação, as distâncias médias femininas são sempre inferiores às masculinas, com diferenças girando em torno de 0,4 km: 0,6 km em 1977, 0,7 km em 1987, 0,3 km em 1997 e, em 2007, a diferença não é estatisticamente significativa.
- (v) As distâncias médias de viagem motivo manutenção/compras oscilam em torno de 6,9 km: passam de 6,6 km em 1977 para 7,1 km em 1987, caem para 6,9 km em 1997 e sobem novamente para 7,0 km em 2007.
- (vi) Para viagens motivo manutenção/compras, as distâncias médias femininas são sempre inferiores às masculinas, com diferenças girando em torno de 1,1 km: 1,1 km em 1977, 0,8 km em 1987, 1,7 km em 1997 e 0,8 km em 2007.
- (vii) As distâncias médias de viagem motivo lazer/outros oscilam são crescentes no tempo: passam de 6,8 km em 1977 para 7,1 km em 1987, mantém o valor de 7,1 km em 1997 e sobem para 7,3 km em 2007.
- (viii) Para viagens motivo lazer/outros, as distâncias médias femininas são sempre inferiores às masculinas, com diferenças girando em torno de 0,9 km: 0,9 km em 1977, 1,2 km em 1987, 1,0 km em 1997 e 0,4 km em 2007.
- (ix) As distâncias médias de viagem motivo servir passageiro decrescem de 1977 (7,0 km) para 1987 (5,3 km), a partir de quando sobem até 2007 (5,8 km). Lembrando que em 1987 não havia o modo servir passageiro.
- (x) Para viagens motivo servir passageiro, as distâncias médias femininas são sempre inferiores às masculinas, com diferenças crescentes: 0,1 km em 1977, 0,9 km em 1987 e 1,1 km em 2007.

Gráfico 22 – Distâncias médias de viagem por ano e por sexo, segundo o motivo da viagem

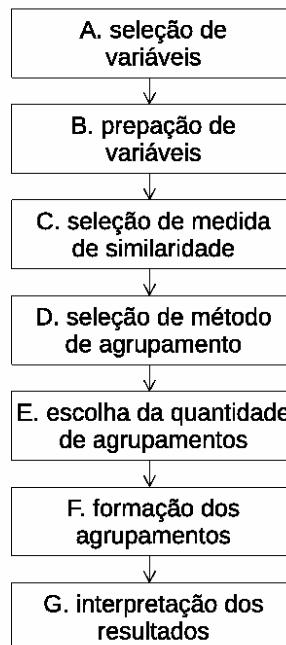


6.3 Análise de conglomerados para identificação de grupos

A análise de conglomerados tem como objetivo segregar elementos (observações) “em grupos homogêneos internamente, heterogêneos entre si e mutuamente exclusivos, a partir de determinados parâmetros conforme uma medida de similaridade ou distância” (FÁVERO et al., 2009, p.196). Esta técnica estatística de interdependência foi escolhida pois se deseja agrupar as pessoas, ou mesmo famílias, em grupos homogêneos em função da similaridade do comportamento de viagens.

Devido ao tamanho da base de dados e ao baixo desempenho em termos computacionais do pacote *stats*⁵ (R Core Team, 2011), preferiu-se a função *hclust* do pacote *fastcluster*⁶. Segundo Müllner (2013), no pior caso, a função do pacote *stats* tem seu tempo de execução proporcional a N^3 , já a mesma função no pacote *fastcluster*, é proporcional a N^2 . As análises de conglomerados não hierárquicas foram executadas pela função *kmeans* do pacote *stats*. Na Figura 9 pode-se observar os passos realizados nesta etapa de análise.

Figura 9 – Etapas para a realização de análise de conglomerados (*clusters*)



⁵ Também existe a função *agnes*, do pacote *cluster*, cuja documentação está disponível em <<https://cran.r-project.org/web/packages/cluster/cluster.pdf>> e que se propõe a implementar o mesmo algoritmo. Porém, o resultado obtido para os três pacotes (stats, fastcluster e cluster) não foram iguais para o mesmo procedimento. Murtagh e Legendre (2014) já apontavam divergência entre as funções *hclust* e *agnes* quando aplicadas à mesma matriz de distâncias.

⁶ Detalhes da implementação disponível em Müllner (2011).

Na **etapa A**, foram delimitados dois conjuntos iniciais de variáveis de análise:

- **conjunto I:** De atributos de viagem, relativas à **família**, a saber:

FAM_DIST_TOT - distância total das viagens da família;
FAM_DIST_MED - distância média de viagem da família;
FAM_DURACAO_TOT - duração total das viagens da família;
FAM_DURACAO_MED - duração média das viagens da família;
FAM_VIAG_TOT - número total de viagens da família.

- **conjunto II:** De atributos de viagem, relativas à **pessoa**, a saber:

PESS_DIST_TOT - distância total das viagens da pessoa;
PESS_DIST_MED - distância média de viagem da pessoa;
PESS_DURACAO_TOT - duração total das viagens da pessoa;
PESS_DURACAO_MED - duração média das viagens da pessoa;
PESS_MODO_ONIBUS - número de vezes que a pessoa utilizou o ônibus (de linha, escolar, van, etc.);
PESS_MODO_DIRIG - número de vezes que a pessoa viajou dirigindo automóvel;
PESS_MODO_PASS - número de vezes que a pessoa viajou como passageira de automóvel;
PESS_MODO TREM - número de vezes que a pessoa utilizou o trem ou metrô;
PESS_MODO_MOTO - número de vezes que a pessoa utilizou a motocicleta;
PESS_MODO_BICI - número de vezes que a pessoa utilizou a bicicleta;
PESS_MODO_APE - número de vezes que a pessoa viajou a pé;
PESS_MODO_OUTROS - número de vezes que a pessoa utilizou algum outro modo;
PESS_NO_MODOS - número de modos diferentes utilizados pela pessoa;
PESS_MOTIVO_TRAB - número de viagens da pessoa com motivo trabalho;
PESS_MOTIVO_EDUC - número de viagens da pessoa com motivo educação;
PESS_MOTIVO_RES - número de viagens da pessoa com motivo residência;
PESS_MOTIVO_SERV_PAS - número de viagens da pessoa com motivo servir passageiro;
PESS_MOTIVO_MANUT_COMPRA - número de viagens da pessoa com motivo manutenção/compras;
PESS_MOTIVO_LAZER_OUTROS - número de viagens da pessoa com motivo lazer/outros;
PESS_NO_MOTIVOS - número de motivos diferentes das viagens da pessoa;
PESS_PER_MADRUG - número de viagens da pessoa entre 0h01 e 5h00;
PESS_PER_COM_MAN - número de viagens da pessoa entre 5h01 e 9h00;
PESS_PER_MANHA - número de viagens da pessoa entre 9h01 e 12h00;
PESS_PER_MEIODIA - número de viagens da pessoa entre 12h01 e 14h00;
PESS_PER_TARDE - número de viagens da pessoa entre 14h01 e 17h00;

PESS_PER_COM NOI - número de viagens da pessoa entre 17h01 e 22h00;

PESS_PER_NOITE - número de viagens da pessoa entre 22h01 e 0h00;

PESS_NO_PERIODOS - número de períodos diferentes em que a pessoa realizou viagem;

TOT_VIAG - número total de viagens da pessoa.

Ocorre em diversos bancos de dados, inclusive neste, de se desejar analisar variáveis cujas unidades e ordens de grandezas não são as mesmas. Faria (2009) indica que há duas razões para padronizar dados: (i) “evitar que as unidades escolhidas para mensurar as características afetem arbitrariamente a similaridade entre indivíduos”, e (ii) fazer com “que as características contribuam igualmente na avaliação da similaridade entre indivíduos”. Se houver alguma unidade de medição que tenha uma amplitude maior que as demais (como é o caso das distâncias totais da família em relação à distância média da pessoa), ela terá maior peso na análise de *cluster*. Então, para mitigar o efeito dessas diferenças é indicado padronizar os dados antes de submetê-los à análise de conglomerados. Ao fazer isso, o pesquisador assume que a importância da variável decresce conforme a variabilidade aumenta (EVERITT et al., 2011). Há diversas formas de fazer essa padronização, as mais comuns são *z-scores* e normalização (FÁVERO et al., 2009). Há também possibilidades como dividir pela mediana dos desvios absolutos ou pelos intervalos de valores da variável (GNANADESIKAN; KETTENRING; TSAO, 1995 apud EVERITT et al., 2011), porém, a complexidade deste últimos métodos não se justifica frente ao conjunto de dados em discussão. Assim, na **etapa B** do presente trabalho, foi feita padronização das variáveis pelo método *z-scores* conforme Equações (6.1), (6.2) e (6.3).

$$Z(x)_i = \frac{x_i - \bar{x}}{\sigma(x)} \quad (6.1)$$

sendo:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (6.2)$$

$$\sigma(x) = \sqrt{\frac{1}{(n-1)} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (6.3)$$

Um dos principais problemas das técnicas de aglomeração não hierárquica (K-means) é definir de início o número de *clusters* desejado (HARTIGAN, 1985; FÁVERO et al., 2009; EVERITT et al., 2011). Existem algumas técnicas para essa determinação:

- (i) A ***upper tail rule*** que considera os valores de fusão como uma série. Calcula-se a média, desvio padrão, estatística t como o desvio normalizado a partir da média. Em seguida, calcula-se o desvio padrão para cada valor de fusão (assumida distribuição normal), seleciona o primeiro “significativo” como sendo aquele cuja estatística t excede

o nível de 5% de significância. Assim, a hipótese nula é que o valor fusão do k-ésimo termo advém da distribuição normal dos valores de fusão (MOJENA, 1977).

(ii) O **índice RMSSTD** (*root mean square standard deviation*), ou raiz quadrada do desvio padrão médio (ver Equação (6.4)), calcula a homogeneidade dos agrupamentos (SHARMA, 1996), de maneira que quanto mais compactos os grupos, menor o valor desta estatística.

$$RMSSTD = \sqrt{\frac{\sum_{j=1}^{nc} \sum_{k=1}^{n_j} (x_k - \bar{x}_j)^2}{\sum_{i=1}^{nc} (n_{ij} - 1)}} \quad (6.4)$$

(iii) O **índice R^2 ajustado** (ver Equação (6.5)), indica dissimilaridade entre agrupamentos (SHARMA, 1996), assim, quanto mais alto o valor de R^2 ajustado, mais dissimilaridade existe entre os grupos. O pesquisador pode estabelecer um valor desejado para R^2 e, a partir daí, determinar o número de *clusters*.

$$R^2 = \frac{\left[\sum_{i=1}^{nc} \sum_{k=1}^{n_j} (x_i - \bar{x}_j)^2 \right] - \left[\sum_{j=1}^{nc} \sum_{k=1}^{n_j} (x_k - \bar{x}_j)^2 \right]}{\sum_{j=1}^{nc} \sum_{k=1}^{n_j} (x_k - \bar{x}_j)^2} \quad (6.5)$$

(iv) O **best cut** é um método que se baseia em um dendrograma (representação bidimensional em forma de árvore) que deve ser cortado quando as diferenças entre grupos forem visualmente mais significativas. Nesta representação, as linhas são ligadas segundo níveis de similaridade que aggregará os indivíduos (EVERITT et al., 2011).

Então, primeiro foi realizada a aglomeração hierárquica para definir as quantidades de grupos. No método hierárquico, se há n observações (linhas), parte-se de n grupos, ou seja, existe uma observação por grupo. A partir de medidas de similaridade, num processo iterativo, as observações vão sendo agrupadas até que se chegue a um único grupo no final. Segundo Maxwell (1977), primeiro é feita a conversão da matriz n versus p de dados em uma matriz n versus n de medidas de similaridade ou dissimilaridade, tendo-se n unidades amostrais e p características. Optou-se por primeiro utilizar o **best cut** com os dendrogramas⁷ e, em seguida, seriam avaliados os índices **RMSSTD** e **R^2 ajustado**.

As medidas de (dis)similaridade podem ser de distância, correlação ou associação, esta última indicada para variáveis qualitativas. Como as variáveis selecionadas na **etapa A** são métricas ou *dummies*, são indicadas medidas de distância ou correlação. Na Tabela 38 é possível observar algumas das principais medidas de dissimilaridade comumente

⁷ Os dendrogramas gerados admitiram escala não-monotônica.

utilizadas. Optou-se, na **etapa C**, pela medida de distância euclidiana, indicada pela literatura (Hair Jr et al., 2005 apud FÁVERO et al., 2009), a ser adotada em conjunto com os métodos de agrupamento Ward (Ward Jr, 1963) e centroide. O arcabouço teórico indica que tanto as distâncias euclidianas quanto as euclidianas quadráticas resultarão nos mesmos *clusters* e, no caso da função *hclust* utilizada, a distância padrão calculada é a euclidiana.

Tabela 38 – Medidas de dissimilaridade utilizadas em análise de *cluster*

Medida	Fórmula
Distância Euclidiana	$d_{ij} = \left[\sum_{k=1}^p w_k^2 (x_{ik} - x_{jk})^2 \right]^{1/2}$
Distância <i>city block</i>	$d_{ij} = \sum_{k=1}^p w_k x_{ik} - x_{jk} $
Distância de Minkowski	$d_{ij} = \left(\sum_{k=1}^p w_k^r x_{ik} - x_{jk} ^r \right)^{1/r} \quad (r \geq 1)$
Distância de Canberra	$d_{ij} = \begin{cases} 0 & \text{for } x_{ik} = x_{jk} = 0 \\ \sum_{k=1}^p w_k x_{ik} - x_{jk} / (x_{ik} + x_{jk}) & \text{for } x_{ik} \neq 0 \text{ or } x_{jk} \neq 0 \end{cases}$

Fonte: (EVERITT et al., 2011)

Na **etapa D**, foi feita a aglomeração para o **conjunto I** de variáveis da família e para o **conjunto II** de variáveis da pessoa, utilizando tanto o método Ward quanto o centroide, utilizando filtros que captassem a ocorrência da família (F_FAM=1) ou da pessoa (F_PESS=1), respectivamente, para que não houvesse repetição indevida de famílias ou pessoas. Essas aglomerações foram feitas sem distinção dos anos e resultaram nos dendrogramas apresentados nas Figuras 10 e 11. Percebe-se que a forma do dendrograma difere muito pouco entre os métodos Ward e centroide para o mesmo conjunto de variáveis.

Considerando o **best cut** observa-se que os dendrogramas que partiram das variáveis de família, indicam 4 como sendo um número de *clusters* interessante de ser dado como *input* do método K-means - observar seções S da Figura 10. Os índices **RMSSTD** e R^2 ajustado, conforme pode ser visto no Gráfico 23, também corroboram para a divisão em 4 grupos, representando 90% da variância.

Partindo do conjunto de atributos de viagens relativas às pessoas, o **best cut** dos dendrogramas resultantes também indicam 4 *clusters*, seja pelo método Ward, seja pelo centroide - observar seções S da Figura 11. Os índices **RMSSTD** e R^2 ajustado, conforme pode ser visto no Gráfico 24, também corroboram para a divisão em 4 grupos, representando mais de 90% da variância.

Figura 10 – Dendrograma resultante da análise de conglomerados hierárquico, para o conjunto de atributos de viagens relativas às famílias

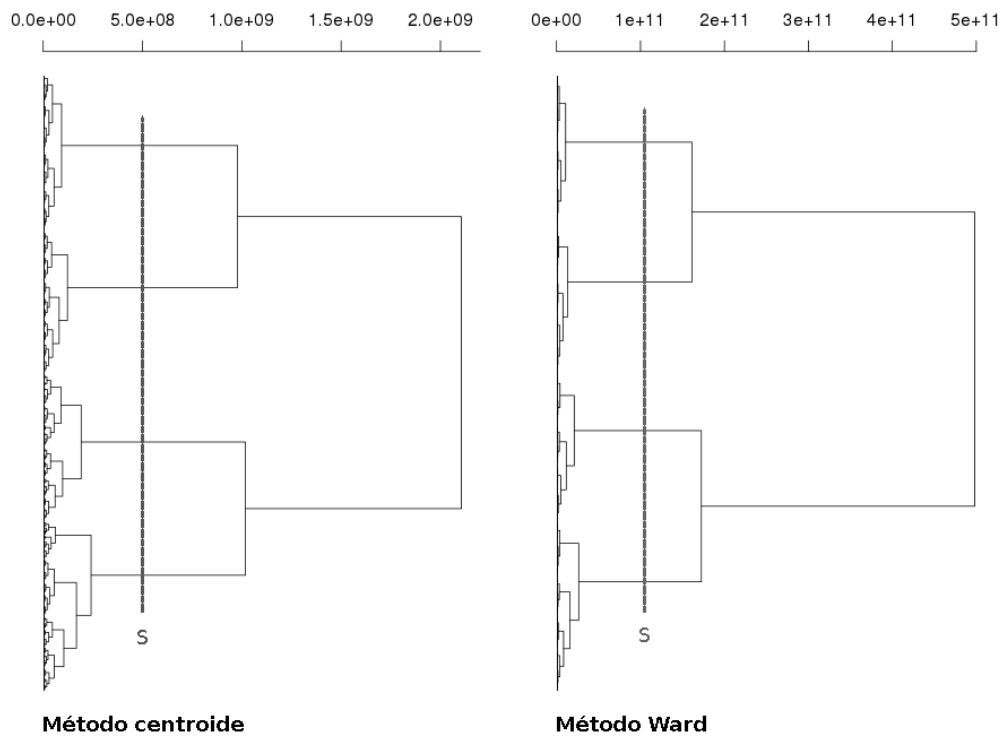


Gráfico 23 – Avaliação do número de *clusters* para o conjunto de atributos de viagens relativas às famílias

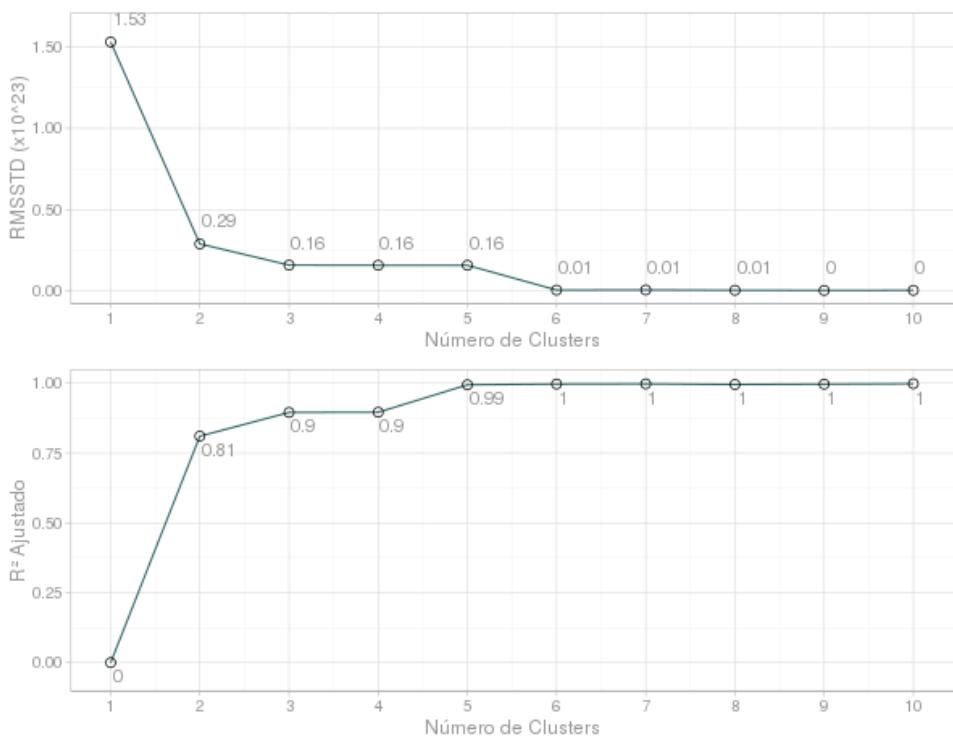


Figura 11 – Dendrograma resultante da análise de conglomerados hierárquico, para o conjunto de atributos de viagens relativas às pessoas

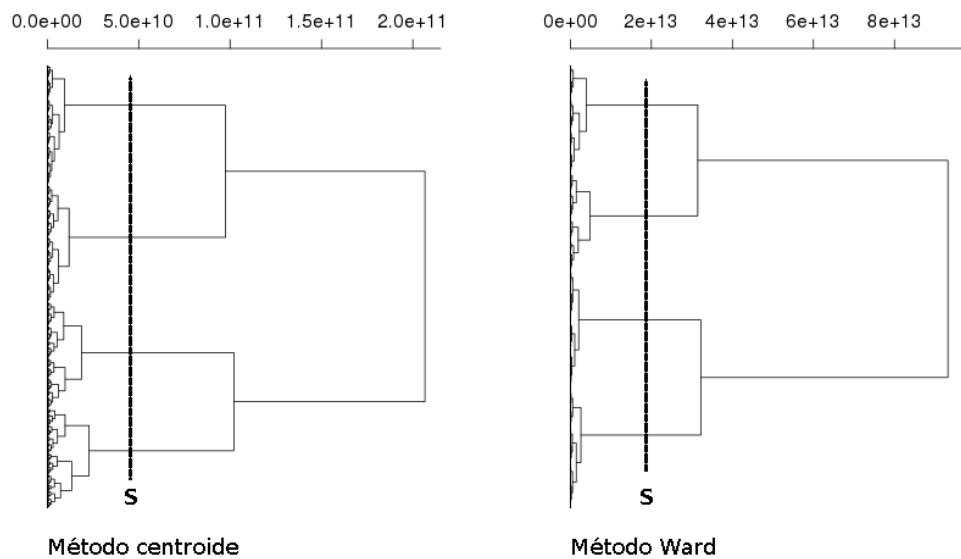
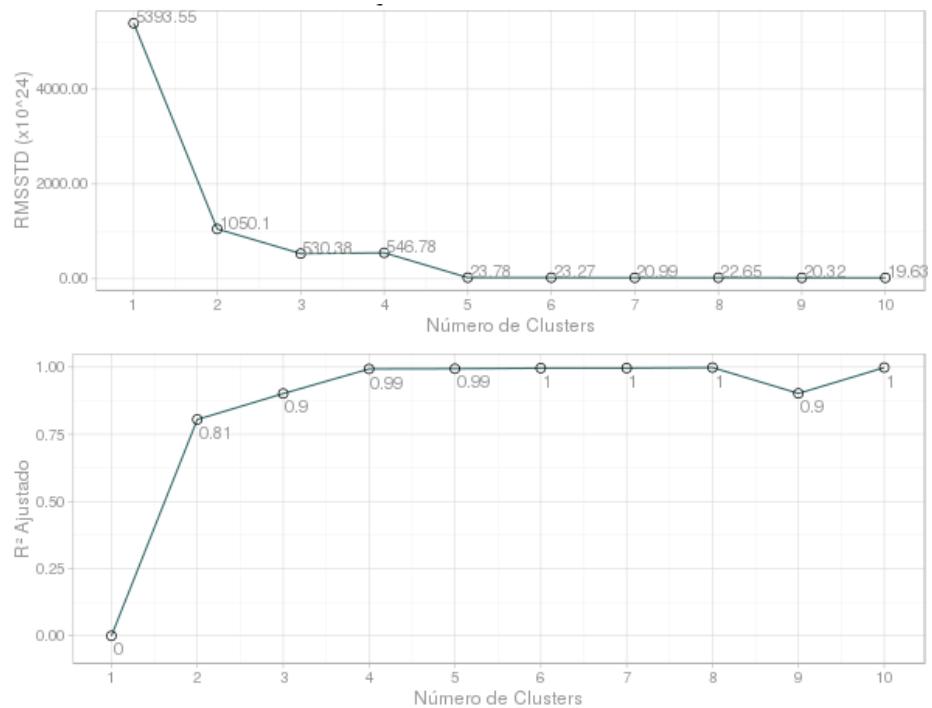


Gráfico 24 – Avaliação do número de *clusters* para o conjunto de atributos de viagens relativas às pessoas



Assim, na **etapa E**, definiu-se que seriam adotados 4 *clusters*, tanto para o conjunto de variáveis I (de família) como o II (de pessoas) no método de aglomeração K-means⁸ que, segundo Gouvea e La Plata (2006 apud FÁVERO et al., 2009), minimiza a variância interna aos grupos e maximiza a variância entre grupos.

Na **etapa F**, foram formados então quatro agrupamentos para atributos de viagens de **família**, pelo método Ward. Observa-se que os grupos formados correspondem exatamente às observações de cada ano. Ou seja, o *cluster* 1 agregou as observações de 1977, o *cluster* 2 agregou as observações de 1987, e assim por diante - ver Tabela 39. Entretanto, utilizando o método centroide, houve a união de 1997 e 2007, além da separação de 1987 em dois grupos distintos - ver Tabela 40. Ao analisar os quatro agrupamentos para atributos de viagens de **pessoas**, tanto o método centroide quanto o Ward, formam grupos que correspondem exatamente às observações de cada ano, como na Tabela 39.

Tabela 39 – Resultado do agrupamento de 4 *clusters*, por atributos de viagens de família - método Ward

<i>Cluster</i> nº	% de famílias de 1977	% de famílias de 1987	% de famílias de 1997	% de famílias de 2007
1	100	0	0	0
2	0	100	0	0
3	0	0	100	0
4	0	0	0	100

Tabela 40 – Resultado do agrupamento de 4 *clusters*, por atributos de viagens de família - método centroide

<i>Cluster</i> nº	% de famílias de 1977	% de famílias de 1987	% de famílias de 1997	% de famílias de 2007
1	100	0	0	0
2	0	100	0	0
3	0	100	0	0
4	0	0	51,94	48,06

Nota: Do total de famílias de 1987, 55,6% pertencem ao grupo 2 e 44,4% pertencem ao grupo 3.

⁸ A função *kmeans* do pacote *stats* do software R implementa o algoritmo de Hartigan e Wong (1979) por padrão para sua execução, que utiliza também a distância euclidiana como medida de similaridade.

Foi feita uma nova análise, agora com três agrupamentos, tendo em foco a **família**, para observar quais grupos se juntariam e quais continuariam separados. Pelo método Ward, os grupos que se unem são os anos de 1997 e 2007. Já pelo método centroide, 1997 destaca-se de 2007, enquanto 1987 se agrega completamente a 1977, conforme pode ser observado nas Tabelas 41 e 42.

Tabela 41 – Resultado do agrupamento de 3 *clusters*, por atributos de viagens de família - método Ward

<i>Cluster</i>	% de famílias de 1977	% de famílias de 1987	% de famílias de 1997	% de famílias de 2007
nº				
1	100	0	0	0
2	0	100	0	0
3	0	0	46,5	53,5

Tabela 42 – Resultado do agrupamento de 3 *clusters*, por atributos de viagens de família - método centroide

<i>Cluster</i>	% de famílias de 1977	% de famílias de 1987	% de famílias de 1997	% de famílias de 2007
nº				
1	48,1	51,9	0	0
2	0	0	100	0
3	0	0	0	100

Ao focar os atributos de viagens das **pessoas**, agrupadas em três *clusters*, não houve diferença no resultado utilizando Ward ou centroide, tal como ocorreu com quatro grupos, conforme Tabela 43. Nota-se que 1977 e 1987 se mantêm separados, e 1997 se une a 2007.

Tabela 43 – Resultado do agrupamento de 3 *clusters*, por atributos de viagens de pessoas - métodos Ward e centroide

<i>Cluster</i>	% de famílias de 1977	% de famílias de 1987	% de famílias de 1997	% de famílias de 2007
nº				
1	100	0	0	0
2	0	100	0	0
3	0	0	51,9	48,1

Os resultados mostram que a variável ANO, embora não inserida nas análises de conglomerados, acabou sendo a grande diferenciadora dos grupos. Esses resultados levantam questões como:

- (i) o tempo (em si ou como *proxy* de outras variáveis) é uma categoria de análise relevante;
- (ii) pode ter havido alterações no método de pesquisa (conceitos, definições, categorias de variáveis, etc.) e o agrupamento temporal esteja captando este efeito;
- (iii) deve haver semelhanças entre 2007 e 1997;
- (iv) deve haver semelhanças entre 1987 e 1977, talvez mais fracas que aquelas entre 2007 e 1997.

Sob a perspectiva de gênero, retomando os dados de participação das mulheres na PEA, apresentados no Gráfico 1 (página 35), percebe-se o $\Delta_{1991-1980} = 6,3\%$, $\Delta_{2000-1991} = 11,2\%$ e $\Delta_{2010-2000} = 4,8\%$. Embora, os dados da PEA refiram-se aos anos 1980, 1991, 2000 e 2010, não coincidentes com os das Pesquisas OD (1977, 1987, 1997 e 2007), se tomarmos os anos mais próximos como referencial de análise, percebemos que o maior salto na participação feminina no mercado de trabalho ocorreu entre 1991 e 2000, e que as pesquisas que parecem apresentar a maior dissemelhança são as OD-1987 e OD-1997. Não é possível fazer muitas afirmações a partir somente deste paralelo, entretanto, isto pode ser um indicativo de que os padrões de mobilidade se alteraram sob efeito do tempo e também considerando o gênero como categoria de análise.

Para explorar melhor o que ocorre dentro de cada grupo e as diferenças entre grupos foram analisadas as características de viagens, das pessoas e das famílias dos quatro *clusters* resultantes - ver Anexo D.

No agrupamento feito por atributos de viagens da **família** pelo método **Ward**, observa-se que a maior diferença percentual entre valores mínimo e máximo entre grupos ocorreu com a variável “% de pessoas com superior completo” (76%) e a menor diferença ocorreu com a variável “sexo” (4%). Se por um lado já se esperava que o grau de instrução fosse uma variável relevante para explicar o comportamento da demanda, por outro, contrariando expectativas iniciais, a variável sexo sozinha não se mostrou tão relevante assim. A Tabela 44 apresenta as variáveis que apresentaram as diferenças percentuais entre valores máximos e mínimos superiores a 50%. Para estas nove variáveis foi realizado teste qui-quadrado de Pearson (de independência) em relação à variável “nº do *cluster*” e todas mostraram-se significativas considerando com um nível de significância de 5%.

No agrupamento feito por atributos de viagens da **família** pelo método **centroide**, observa-se que a maior diferença percentual entre valores mínimo e máximo entre grupos ocorreu com a variável “% de famílias na Classe A” (79%) e a menor diferença ocorreu com a variável “Média da quantidade de trabalhadores (as) na família” (2%).

Buscando entender o fato curioso de 1987 ter se dividido entre dois grupos quando da formação de quatro grupos com método centroide (Tabela 40), foram levantadas as maiores diferenças percentuais entre valores mínimo e máximo (ver Anexo D) que ocorreram entre os grupos 2 e 3 de 1987. Estas principais diferenças eram de categorias relacionadas à **distância, sexo e situação familiar**, a saber:

Tabela 44 – Ordenamento das variáveis pelas maiores diferenças percentuais - para agrupamento FAM_CLUSTER_WARD4

Ranking	Variável	Diferença %
1º	% de pessoas com superior completo	76
2º	% de pessoas com situação familiar ‘outros’	71
3º	% de famílias na Classe A	67
4º	% de pessoas com médio completo ou superior incompleto	66
5º	% de pessoas que servem passageiro no destino	65
6º	% de famílias na Classe E	61
7º	% de famílias com presença de criança entre 0 e 4 anos	60
8º	% de pessoas com situação familiar ‘empregado(a)’	59
9º	% de famílias com presença de criança entre 5 e 9	56

- (i) distâncias médias de viagem (da pessoa e da família);
- (ii) % de pessoas do sexo feminino e
- (iii) % de outros parentes / agregados.

A Tabela 45 apresenta as variáveis que apresentaram as diferenças percentuais entre valores máximos e mínimos superiores a 50%. Para estas nove variáveis foi realizado teste qui-quadrado de Pearson (de independência) em relação à variável “nº do cluster” e todas mostraram-se significativas considerando um nível de significância de 5%.

Tabela 45 – Ordenamento das variáveis pelas maiores diferenças percentuais - para agrupamento FAM_CLUSTER_CENTROIDE4

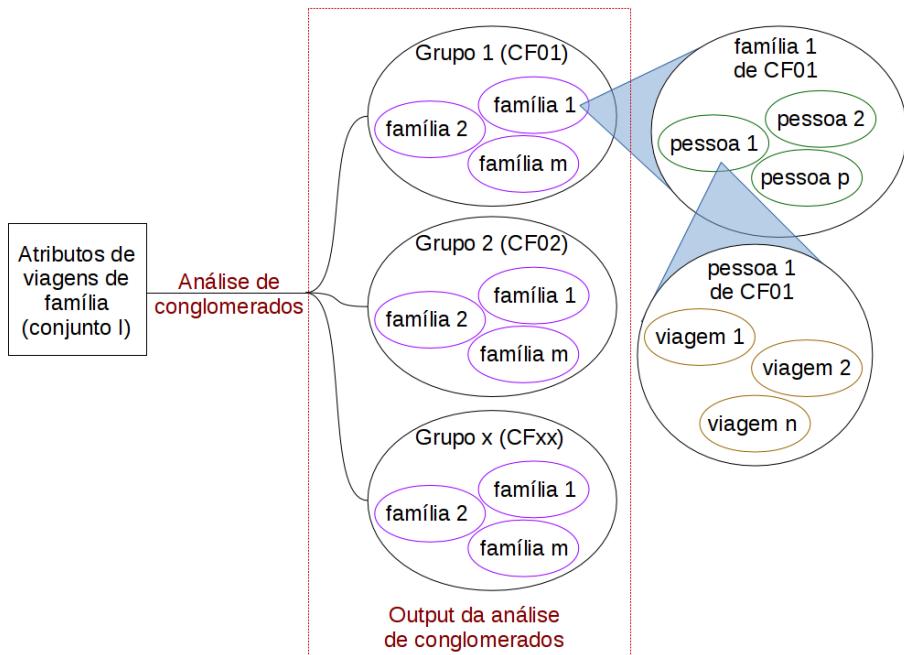
Ranking	Variável	Diferença %
1º	% de famílias na Classe A	79
2º	% de pessoas com situação familiar ‘empregado(a)’	76
3º	% de pessoas com superior completo	73
4º	% de pessoas com situação familiar ‘outros’	71
5º	% de famílias na Classe E	66
6º	% de pessoas com médio completo ou superior incompleto	65
7º	% de pessoas que servem passageiro no destino	61
8º	% de famílias na Classe B	57
9º	% de famílias com presença de criança entre 0 e 4 anos	51

Oito dentre as nove variáveis listadas nas Tabelas 44 e 45 repetem-se, indicando mais consistência do que divergência entre os métodos utilizados. Ao se dar importância para a “% de pessoas com superior completo” ou “% de pessoas com ensino médio completo

ou superior incompleto”, na realidade, se está elencando a variável **grau de instrução** como relevante para explicar os agrupamentos. As porcentagens de famílias nas classes A, B ou E, colocam a **renda familiar** como outra variável relevante. Raciocínio análogo se aplica à “% de pessoas com situação familiar outros” ou “empregado(a)” e à **situação familiar**, e à “% de pessoas que servem passageiro no destino” e ao **motivo no destino**. Aqui cabe pontuar que a presença de situação familiar “empregado(a)” pode ser mais uma *proxy* da renda familiar, do que da composição familiar em si - afinal manter um(a) ou mais empregado(a) domésticos(as) é um custo que nem toda família consegue custear. Sobre a situação familiar “outros” fica a ressalva de que pode estar englobado aqui toda uma sorte de diferentes definições adotadas ao longo da história das Pesquisas OD. *Dummies* que indicam se há **crianças até 9 anos** na família também são relevantes para explicar os agrupamentos.

Se tomado o mesmo procedimento com os *clusters* resultantes dos atributos de viagem das pessoas (método centroide ou Ward), os resultados seriam idênticos aos obtidos no Anexo D (atributos de família, método Ward) e nas Tabelas 39 e 44, posto que o resultado dos agrupamentos foi o mesmo (por ano).

Seguindo recomendação de Vespucci (2003) foi feita uma nova análise de conglomerados a partir dos resultados da primeira análise de conglomerados. Ou seja, o procedimento de análise de *clusters* (hierárquico e não hierárquico) foi feito separadamente para 1977, 1987, 1997 e 2007, para os conjuntos de variáveis I (famílias). Analogamente ao Anexo D , o Anexo E apresenta o resumo dos resultados quando formados os conglomerados a partir de atributos de viagens das famílias (método centroide) para cada ano. Assim, a partir dos conglomerados formados apenas a partir dos atributos de viagem, formaram-se grupos de famílias. Então, cada família passou a pertencer a um determinado *cluster*, e como a cada família pertencem pessoas, as pessoas das famílias também passaram a pertencer aos *clusters*, bem como as viagens que realizaram - ver Figura 12.

Figura 12 – Pertinência a *cluster* de famílias, pessoas e viagens

A Tabela 46 apresenta a síntese dos principais resultados das análises de conglomerados realizadas para cada ano, tais como número de famílias, número de viagens, número de pessoas, número médio de viagens por pessoa, número médio de viagens por família, duração média e distância média de viagem da pessoa; valores estes calculados considerando apenas quem fez viagem e sem fazer qualquer tipo de expansão ou ajuste.

Em 1977, foram formados os grupos CF01, CF02, CF03 e CF04, que serão analisados comparativamente (médias e percentuais de variáveis ou categorias). As proporções de algumas categorias são preponderantes em todos grupos, pouco servido como fator de discernimento entre os *clusters*. Por exemplo, para as viagens feitas de transporte coletivo, o percentual de viagens por ônibus de linha sempre são superiores aos demais modos, logo, nas análises, foca-se nas diferenças entre os agrupamentos para poder captar as contribuições dos demais modos na formação dos agrupamentos. Raciocínio análogo se aplica às categorias, sempre mais frequentes: viagens motivo trabalho, sexo feminino, situação familiar ‘filho(a)/enteado(a)’, grau de instrução ‘não alfabetizado(a) / fundamental incompleto’, pessoas que trabalham, famílias que possuem um automóvel e famílias na classe C.

Tabela 46 – Síntese dos resultados da análise de conglomerados para cada ano

Ano	1977				1987				1997				2007			
Cluster nº	CF01	CF02	CF03	CF04	CF05	CF06	CF07	CF08	CF09	CF10	CF11	CF12	CF13	CF14	CF15	CF16
Nº de famílias	6584	6587	6290	6696	5719	7489	8196	6813	6424	6294	7240	6887	7505	6406	10147	6797
Nº de pessoas	27059	29837	20933	30199	23684	30618	28482	28029	23506	24365	23667	27242	22712	20233	26373	22087
Nº de viagens	52400	46314	40576	48151	35545	48545	54134	44894	41104	38226	45399	38805	41909	35977	55980	35799
Nº médio de viagens por família	8,5	7,5	7,1	7,7	6,6	7,0	7,2	7,1	7,1	6,7	7,1	6,3	6,3	6,3	6,3	6,1
Nº médio de viagens por pessoa	3,0	2,7	3,0	2,7	2,5	2,5	2,8	2,6	2,7	2,5	2,8	2,4	2,6	2,5	2,9	2,5
Duração média de viagens da pessoa (min)	28,79	33,77	27,43	34,11	35,47	36,78	27,34	32,59	31,45	36,6	27,4	38,54	39,7	39,66	30,17	36,54
Distância média por viagem da pessoa (m)	4166,39	5699,65	3391,92	5141,26	5264,61	5599,07	3547,45	4435,31	3687,21	4482,88	3514,06	4590,57	5385,75	5802,02	3812,34	5693,99

O CF01 é o grupo com o maior número de viagens. As pessoas e famílias deste *cluster* fazem em média mais viagens por dia e utilizam mais frequentemente o transporte individual - este grupo apresenta a maior taxa de motorização familiar. De quem usa o transporte público, frente aos demais *clusters*, há preferência pelo Metrô, seguido dos ônibus escolares/fretados. A situação familiar mais frequente no CF01 é “filhos/enteados(as)” assim como para todos os *clusters*, entretanto, entre os *clusters* é o que apresenta o maior percentual de cônjuges. Com maior concentração nas classes A e B, as famílias deste *cluster* apresentam as maiores rendas familiares médias e as maiores porcentagens de famílias com automóveis (1, 2 ou mais autos).

O CF02 é o grupo com o menor número médio de viagens por pessoa. Ele apresenta as maiores distâncias (média e total) de viagem (para pessoa e para a família). É o grupo com o maior percentual de viagens a pé⁹ e menor percentual de viagens por transporte individual. De quem usa o transporte público, frente aos demais *clusters*, há maior utilização de trem - modo bastante utilizado quando trata-se de grandes distâncias a serem percorridas. Maiores percentuais de pessoas do sexo masculino, filhos(as) e as menores médias etárias levam a crer que este grupo concentra os estudantes de baixas idades. Assim, conta com maiores percentuais de pessoas que estudam e concentra as menores rendas (individual e familiar) - a baixas idades correspondem baixos rendimentos e baixos níveis de escolaridade, em média. As famílias deste *cluster* são maiores e mais jovens, com mais presença de crianças e menos presença de idosos.

O grupo de estudantes, entretanto, divide-se entre o CF02 (mais novo) e o CF04 (mais velho). O CF04 tem o maior número de pessoas e de famílias, só que ao invés de corresponderem às maiores distâncias, correspondem às maiores durações (média/total, de pessoas/famílias). As viagens neste *cluster* são mais frequentemente feitas por transporte coletivo (principalmente ônibus de linha) e por motivo educação no destino da viagem (não tão distante do CF02). Frente a outros *clusters*, este tem uma proporção maior de pessoas que estudam e de famílias com a presença de crianças 10 a 14 anos. As famílias são maiores e com maior quantidade média de trabalhadores na família e a maior proporção da classe C. Aqui provavelmente estão os que trabalham e estudam e/ou os que vêm de famílias maiores em que irmãos(as) mais velhos(as) já trabalham e viabilizam economicamente o estudo dos mais novos.

Finalizando 1977, temos o grupo CF03, o menor *cluster* de todos (menos viagens, pessoas e famílias). É o grupo em que as pessoas têm o maior número médio de viagens por pessoa, juntamente com o grupo 1, só que com as menores durações e distâncias. Não se destaca dos demais *clusters* (máximo ou mínimo) no uso do transporte individual, coletivo ou a pé, mas apresenta mais viagens de transporte individual, seguido pelas viagens a pé e,

⁹ Só é considerada a pé a viagem realizada integralmente a pé, da origem ao destino, com distância percorrida superior a 500 metros (ou cinco quadras); ou se o motivo da viagem (na origem ou no destino) é trabalho ou escola, independente da distância percorrida.

por fim, vêm as viagens de transporte coletivo. Ao utilizar o transporte público, apresenta o maior percentual de uso do metrô num análise entre *clusters*. O principal motivo no destino é o trabalho e, consequentemente, o *cluster* conta com o maior percentual de pessoas que trabalham e/ou que são responsáveis pela família. Tem o maior percentual de pessoas do sexo feminino, com estruturas familiares menores (menos filhos e mais idosos) e maiores rendas individuais médias. Este grupo parece ser o com maior participação de mulheres que, ao adquirirem independência financeira (é o grupo com o menor índice de donas de casa), adquirem também mobilidade. Lembrando que durações e distância iguais a zero foram descartadas nos cálculos das médias. Isso corrobora a ideia de que quando as mulheres fazem viagens (expurgado o efeito da fixitude domiciliar), ela fazem mais que os homens.

A Tabela 47 indica as maiores diferenças (superiores a 50%) percentuais entre as categorias das mesmas variáveis analisadas anteriormente. Foram realizados testes qui-quadrado de Pearson (de independência) em relação à variável “nº do *cluster*” e todas mostraram-se significativas considerando um nível de significância de 5%. Percebe-se que são mais relevantes as variáveis **modo, situação familiar, grau de instrução, faixa de renda familiar, presença de dois ou mais automóveis na família e presença de filhos até 14 anos**.

Tabela 47 – Ordenamento das variáveis pelas maiores diferenças percentuais - para agrupamento FAM_CLUSTER_CENTROIDE4 apenas de 1977

Ranking	Variável	Diferença %
1º	% de viagens realizadas por trem	91
2º	% de viagens realizadas por metrô	90
3º	% de pessoas com situação familiar ‘empregado’	80
4º	% de pessoas com superior completo	74
5º	% de pessoas com ensino médio completo ou superior incompleto	64
6º	% de famílias na classe A	66
7º	% de famílias que têm 2 ou mais automóveis	58
8º	% com presença de criança entre 5 e 9 anos	58
9º	% de famílias com presença de criança entre 0 e 4 anos	52

Em 1987, foram formados os grupos CF05, CF06, CF07 e CF08, que serão analisados comparativamente (médias e percentuais de variáveis ou categorias). Assim como em 1977, as proporções de algumas categorias são preponderantes em todos grupos, pouco servido como fator de discernimento entre os *clusters*. São estas categorias, sempre mais frequentes: modo ônibus de linha, viagens motivo trabalho, sexo feminino, situação familiar ‘filho(a)/enteado(a)’, grau de instrução ‘não alfabetizado(a) / fundamental incompleto’, pessoas que trabalham, famílias que possuem um automóvel e famílias na classe C.

O CF05, grupo com menos viagens, pessoas e famílias é o grupo de estudantes, análogo ao CF02 de 1977. Isto é, aqui a maior porcentagem de viagens é feita a pé e, entre os que utilizam o transporte público, o trem se destaca dos outros modos. As viagens motivo educação são mais frequentes frente aos demais *clusters*. Aqui ocorrem os maiores percentuais de pessoas do sexo masculino, de quem estuda e de quem é filho(a)/enteado(a), bem como as menores rendas (individual e familiar). As famílias do CF05 são em média maiores, com maior presença de crianças até 14 anos, com menores taxas de motorização e mais frequentemente das classes D e E.

CF06 é o grupo com o maior número de pessoas, cujas viagens têm as maiores distâncias e durações (médias e totais). A maior parte utiliza o transporte público em seus deslocamentos, geralmente o ônibus de linha A. Maioria das viagens tem motivo trabalho no destino e este grupo conta com a maior média de trabalhadores no núcleo familiar. É o grupo de trabalhadores(as), sem paralelo evidente em 1977. Parece que o grupo CF04 de 1977 dividiu-se em 1987: estudantes migrando para o CF05 e trabalhadores(as) engordando o CF06.

CF07 é o grupo com o maior número de viagens e de famílias. É o grupo de maior mobilidade, com os maiores números médios de viagem (de pessoa e da família). As viagens apresentam menores distâncias e durações (médias e totais) e são mais frequentemente feitas por transporte individual. Quando da utilização do transporte público, o metrô se destaca dos outros modos no conjunto dos *clusters*. Os maiores percentuais do sexo feminino, de pessoa responsável pela família ou cônjuges, ou mesmo parente/agregados, aparecem neste grupo, de maior média etária. Trata-se do *cluster* com maior porcentagem de pessoas que trabalham, menor porcentagem de pessoas que estudam, menores tamanhos médios de família e maiores percentuais de presença de idosos. Parece o CF07 ter sido a fusão de grande parte dos elementos dos grupos CF01 e CF03 de 1977: mais velho, mais rico e mais motorizado.

CF08 não tem muitas características marcantes e talvez tenha sido o grupo dos “sem grupo”. Frente aos demais, este *cluster* apresenta os menores percentuais de pessoas com superior completo, de pessoas responsáveis pela família e de famílias com dois ou mais autos. De quem usa o transporte coletivo, apresenta o maior percentual de utilização do trem, junto do CF05, e o maior percentual de utilização de ônibus escolar/fretado.

A Tabela 48 indica as maiores diferenças (superiores a 50%) percentuais entre as categorias das mesmas variáveis analisadas anteriormente, com cada ano como um grupo. Foram realizados testes qui-quadrado de Pearson (de independência) em relação à variável “nº do *cluster*” e todas mostraram-se significativas considerando um nível de significância de 5%. Percebe-se que são mais relevantes as variáveis **modo, situação familiar, grau de instrução, faixa de renda familiar e presença de dois ou mais automóveis na família**.

Tabela 48 – Ordenamento das variáveis pelas maiores diferenças percentuais - para agrupamento FAM_CLUSTER_CENTROIDE4 apenas de 1987

Ranking	Variável	Diferença %
1º	% de viagens realizadas por metrô	90
2º	% de viagens realizadas por trem	86
3º	% de pessoas com situação familiar ‘empregado’	77
4º	% de pessoas com superior completo	71
5º	% de famílias na classe A	66
6º	% de famílias na classe B	60
7º	% de pessoas com ensino médio completo ou superior incompleto	58
8º	% de famílias que têm 2 ou mais automóveis	58

Em 1997, foram formados os grupos CF09, CF10, CF11 e CF12, que serão analisados comparativamente (médias e percentuais de variáveis ou categorias) de maneira análoga aos anos anteriores. Ou seja, o foco não serão as categorias mais frequentes quando elas ocorrerem em todos *clusters* deste ano, como é o caso de: modo ônibus de linha, viagens motivo trabalho, sexo feminino, situação familiar ‘filho(a)/enteado(a)’, grau de instrução ‘não alfabetizado(a) / fundamental incompleto’, pessoas que trabalham, famílias que possuem um automóvel e famílias na classe C.

CF09 é o grupo de 1997 sem grandes características marcantes. Frente aos demais, este *cluster* apresenta os maiores percentuais de cônjuges e de pessoas que servem passageiro. O CF09 conta com o maior número médio de viagens por família e o menor percentual de viagens feitas a pé.

CF10 é o grupo com o menor número de viagens e de famílias, e suas viagens têm as maiores médias das distâncias totais (para pessoa e para família) e da duração total da família. A maior parte utiliza o transporte público em seus deslocamentos, geralmente o ônibus de linha, com a maioria das viagens motivo trabalho no destino. Este grupo conta com a maior média de trabalhadores no núcleo familiar e o maior percentual de pessoas que estudam, frente aos outros grupos. As famílias deste *cluster* apesenta os menores percentuais de presença de idosos e a maior participação na classe C. É o grupo de trabalhadores(as), da classe C, com presença marcante de estudantes, relacionando-se com o CF04 de 1977 e, parcialmente, com o CF06 de 1987.

CF11 é o grupo com o maior número de viagens e de famílias. É o grupo com os maiores números médios de viagem da pessoa, sendo as viagens com menores distâncias e durações (médias e totais, para pessoas e para famílias) e mais frequentemente feitas por

transporte individual. Quando da utilização do transporte público, o metrô se destaca dos outros modos no conjunto dos *clusters*. Os maiores percentuais do sexo feminino, de pessoa responsável pela família ou mesmo parente/agregados, aparecem neste grupo, de maior média etária. Trata-se do *cluster* com maior porcentagem de pessoas que trabalham, menor porcentagem de pessoas que estudam, menores tamanhos médios de família e maiores percentuais de presença de idosos. Parece ser a continuação do CF07, com a diferença importante que é o menor destaque da classe C no CF11 do que no CF07.

O CF12, grupo com o maior número de pessoas e é o grupo de estudantes jovens, análogo ao CF02 de 1977 e ao CF05 de 1987. Ou seja, marcam este grupo os maiores percentuais deste *cluster* frente aos demais de: viagens a pé, destaque para o trem dentre quem utiliza o transporte público, viagens motivo educação, pessoas do sexo masculino, situação familiar filho(a)/entendo(a), analfabetos e pessoas com fundamental incompleto, pessoas que estudam, presença de crianças até 14 anos na família e participação nas classes D e E. Assim, com menor média etária, também apresenta as menores médias das rendas (individual e familiar) e menor taxa de motorização. Este grupo apresenta as maiores distâncias médias (para pessoas e para famílias) e também as maiores durações (média e total, pra pessoas e para famílias).

A Tabela 49 indica as maiores diferenças (superiores a 50%) percentuais entre as categorias das mesmas variáveis analisadas anteriormente, com cada ano como um grupo. Foram realizados testes qui-quadrado de Pearson (de independência) em relação à variável “nº do *cluster*” e todas mostraram-se significativas considerando um nível de significância de 5%. Percebe-se que são mais relevantes as variáveis **modo, situação familiar, grau de instrução, renda individual, renda familiar, faixa de renda familiar e presença de dois ou mais automóveis na família**.

Tabela 49 – Ordenamento das variáveis pelas maiores diferenças percentuais - para agrupamento FAM_CLUSTER_CENTROIDE4 apenas de 1997

Ranking	Variável	Diferença %
1º	% de pessoas com situação familiar ‘empregado’	90
2º	% de viagens realizadas por trem	87
3º	% de pessoas com superior completo	85
4º	% de viagens realizadas por metrô	80
5º	% de famílias na classe A	80
6º	% de famílias que têm 2 ou mais automóveis	65
7º	média da renda individual	60
8º	% de famílias na classe B	58
9º	% de pessoas com ensino médio completo ou superior incompleto	57
10º	média da renda familiar	52

Em 2007, foram formados os grupos CF13, CF14, CF15 e CF16, que também serão analisados comparativamente (médias e percentuais de variáveis ou categorias). O foco continua não sendo as categorias mais frequentes quando elas ocorrerem em todos *clusters* deste ano. Aqui, porém, essas categorias são diferentes dos anos anteriores: modo ônibus de linha, viagens motivo trabalho, sexo feminino, pessoas que trabalham, famílias que possuem um automóvel e famílias na classe C.

CF13 é o grupo de trabalhadores de 2007, semelhante ao CF06 de 1987. Aqui está o maior o número médio de viagens por família e as maiores durações (média e total, por pessoa e por família). Frente aos demais, este *cluster* apresenta: o maior percentual de pessoas que trabalham, a maior média de trabalhadores na família, o menor percentual de viagens feitas a pé e, mediante a utilização do transporte público, o predomínio do ônibus.

CF14 é o grupo com o menor número de pessoas e de famílias, e suas viagens têm as maiores distâncias (médias e totais, para pessoa e para família). A maior parte utiliza o transporte público em seus deslocamentos, geralmente o ônibus de linha. Frente aos outros *clusters*, o CF14 apresenta as maiores porcentagens do motivo servir passageiro no destino. Este grupo conta com o menor percentual de pessoas com situação familiar “empregado(a)” e com o maior de pessoas com ensino médio completo/superior incompleto. O conjunto de famílias deste *cluster* apresenta menos posse de autos, menores taxas de motorização familiar e a maior participação na classe C. É o grupo da classe C, já com presença menos marcante de trabalhadores e estudantes que os grupos ‘classe C’ dos demais anos - relaciona-se com o CF10 de 1997, parcialmente com o CF06 de 1987, e com o CF04 de 1977.

CF15 é o grupo com o maior número de viagens, de pessoas e de famílias. É o grupo com os maiores números médios de viagem (da pessoa e da família), sendo as viagens com menores distâncias e durações (médias e totais, para pessoas e para famílias) e realizadas mais frequentemente por transporte individual. Quando da utilização do transporte público, o metrô se destaca dos outros modos no conjunto dos *clusters*. Os maiores percentuais do sexo feminino, de pessoa responsável pela família ou mesmo parente/agregados, aparecem neste grupo, de maior média etária. Trata-se do *cluster* com menor porcentagem de pessoas que estudam, menores tamanhos médios de família e maiores percentuais de presença de idosos. É o grupo de maior renda média (individual e familiar) e com maior taxa de motorização familiar. Quem está neste grupo realiza mais viagens, que são mais curtas e mais rápidas que as dos demais conglomerados. Como este grupo concentra as classes média alta e alta, provavelmente as pessoas/famílias têm residência localizadas em bairros mais bem servidos de infraestruturas e serviços, somado ao fato de disporem de carro (muitas vezes mais de um). Isso estimula a mobilidade, ou seja, possibilita fazer mais viagens (de menor distância a duração). O CF15 praticamente replica o perfil do CF11 de 1997, que por sua vez, deve ser a continuação do CF07 (à

exceção da classe C) de 1987 que advém, principalmente, dos CF01 e CF03 de 1977.

O CF16, grupo com o menor número de viagens e é o grupo de estudantes jovens, análogo ao CF02 de 1977, ao CF05 de 1987 e ao CF12 de 1997. Quase todas características do CF12 de 1997 são iguais para o CF16, sendo que neste tem-se: (i) além do destaque para o trem entre os que usam o transporte público, o maior percentual de uso de ônibus escolar/fretado; (ii) a maior taxa de viagens motivo trabalho (além de educação); (iii) a maior porcentagem de cônjuges (além da de filhos(as)); (iv) a maior proporção de pessoas com ensino fundamental completo / ensino médio incompleto (além do nível de escolaridade imediatamente anterior). Caracterizam estes *clusters* de “estudantes”: famílias maiores, com maior percentual de crianças em sua composição, menor percentual de idosos e baixas taxas de motorização. A presença marcante das viagens a pé poderiam levar a crer que tratam-se de viagens curtas e pouco demoradas, nas proximidades da residência, e por isso feitas a pé. Entretanto, a maior ocorrência das viagens a pé não é concomitante a baixas distâncias ou durações em nenhum ano. Assim, pode ser que esse efeito (viagens a pé serem curtas e rápidas) não seja suficientemente forte para predominar sobre as longas viagens feitas por outros modos, principalmente por transporte público.

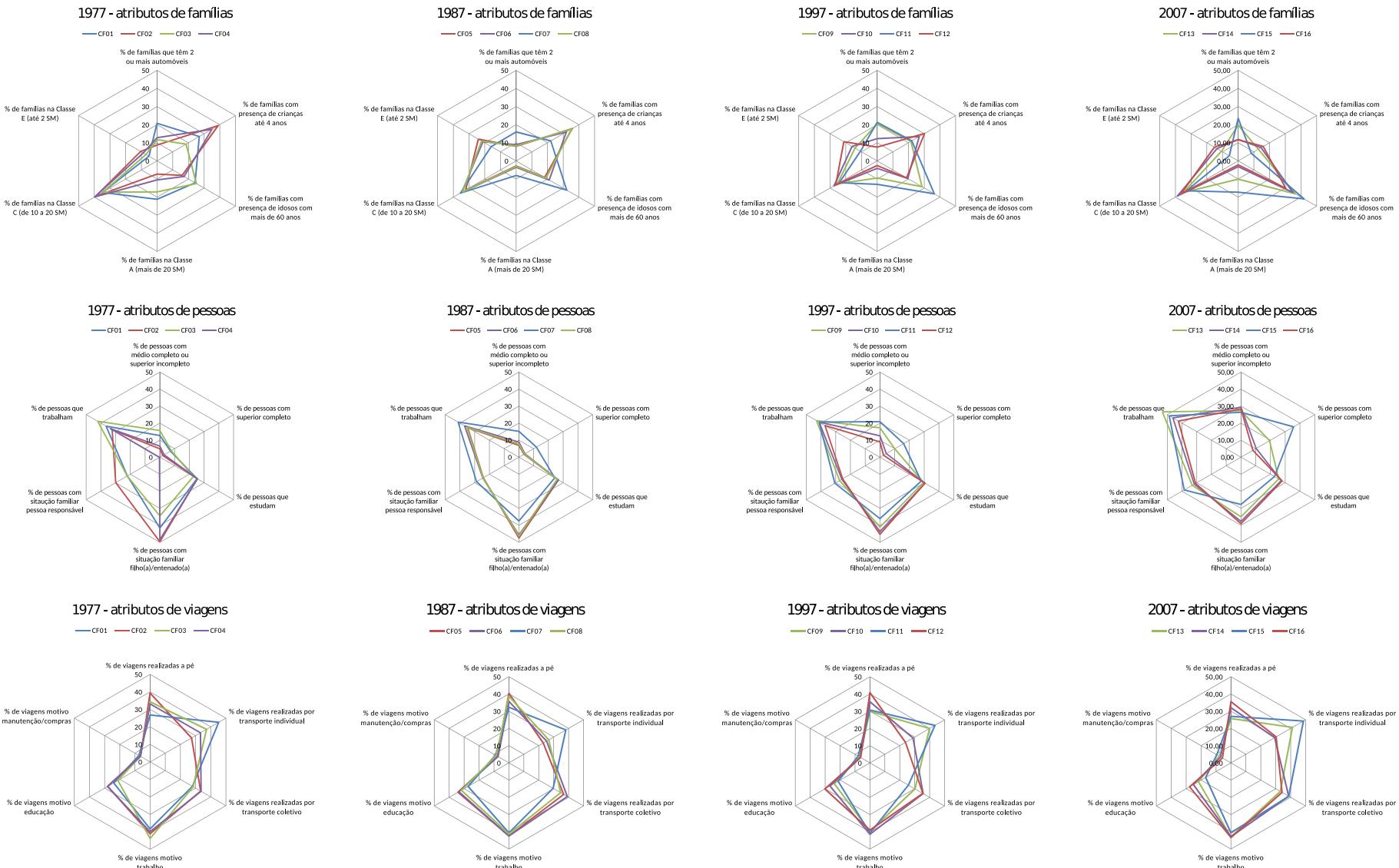
A Tabela 50 indica as maiores diferenças (superiores a 50%) percentuais entre as categorias das mesmas variáveis analisadas anteriormente, com cada ano como um grupo. Foram realizados testes qui-quadrado de Pearson (de independência) em relação à variável “nº do *cluster*” e todas mostraram-se significativas considerando um nível de significância de 5%. Percebe-se que são mais relevantes as variáveis **modo, situação familiar, grau de instrução, renda individual, renda familiar, faixa de renda familiar e presença de filhos entre 5 e 14 anos**.

As variáveis **modo** (de viagem), **situação familiar** e **grau de instrução** (do indivíduo) e **faixa de renda familiar** (da família) apresentaram grandes diferenças percentuais nos quatro anos analisados, sendo portanto, as principais candidatas a variáveis explicativas de um modelo que busque explicar número de viagens, distâncias ou durações, que foram as naturezas das variáveis alvo das análises de conglomerados. Também são boas candidatas a serem acrescidas em eventual modelo: (i) presença de 2 ou mais autos na família, que apareceu nos rankings em três dos quatro anos analisados; (ii) presença de filhos até 14 anos, que apareceu em 1977 e em 2007; (iii) rendas individual e familiar, que apareceram em 1997 e 2007. Partindo desse diagnóstico foi elaborado um quadro síntese com gráficos tipo radar, por ano, e por conjunto de atributos de famílias, pessoas e viagens—ver Figura 13.

Tabela 50 – Ordenamento das variáveis pelas maiores diferenças percentuais - para agrupamento FAM_CLUSTER_CENTROIDE4 apenas de 2007

Ranking	Variável	Diferença %
1º	% de pessoas com situação familiar ‘empregado’	97
2º	% de famílias na classe A	87
3º	% de viagens realizadas por metrô	85
4º	% de pessoas com superior completo	77
5º	% de viagens realizadas por trem	68
6º	% de famílias na classe B	64
7º	% de famílias na classe E	63
8º	média da renda individual	61
9º	% de viagens realizadas por ônibus escolar/fretado	55
10º	% de famílias com presença de crianças entre 10 e 14 anos	55
11º	média da renda familiar	54
12º	% de famílias na classe D	53
13º	% de famílias com presença de crianças entre 5 e 9 anos	53

Figura 13 – Síntese dos resultados da análise de conglomerados para cada ano



6.4 Regressão logística para investigar a formação de grupos

Com o objetivo de investigar as formações de grupos da análise de conglomerados não pelos perfis de quem acabou por compor os *clusters*, como feito na Seção 6.3, mas a partir das variáveis que entraram na clusterização em si, recorreu-se à regressão logística¹⁰ multinomial, uma técnica estatística que busca investigar a relação entre uma variável dependente categórica (neste caso a variável ANO) e variáveis explicativas métricas ou não métricas. Não é um método de classificação, mas neste caso, a técnica está sendo combinada com a análise de conglomerados, para tentar responder quais foram os pesos de cada variável na formação dos grupos (coincidente com os anos, na maior parte das aglomerações).

As probabilidades são estimadas usando a função logística definida conforme Equações (6.6) e (6.7), em que Z (logit) assume valores entre menos e mais infinito, levando $f(Z)$ a assumir valores entre 0 e 1, respectivamente. A ideia, analogamente à regressão linear, é construir uma função de predição que pondere as importâncias das variáveis independentes na explicação de um determinado evento. Só é preciso ressaltar que na regressão linear, os estimadores correspondem às probabilidades, diretamente; já na regressão logística, o que se obtém, são scores do tipo $(\mathbf{X}_i, k) = \beta_k \cdot \mathbf{X}_i$, onde X_i é o vetor das variáveis descritivas por observação i, k é o vetor de pesos (ou coeficientes da regressão) correspondente à escolha k . No presente caso, as observações correspondem às famílias ou às pessoas, que foram agrupadas num determinado *cluster* k . O termo $(p/1-p)$ é que representa, de fato, a chance de ocorrência do evento.

$$f(Z_i) = \frac{1}{1 + e^{-Z}} \quad (6.6)$$

sendo:

$$Z_i = \ln \left(\frac{p}{1 - p} \right) = \alpha + \sum \beta_k \cdot X_i \quad (6.7)$$

Na regressão logística multinomial a variável dependente é categórica com duas ou mais categorias, de natureza ordinal ou nominal. Tomemos por linha de análise o caso em que os *clusters* coincidem com os anos. Assim, tomando 2007 (categoria 4 da variável ANO) por referência, temos as Expressões (6.8), (6.9) e (6.10).

$$Z_i = \ln \left(\frac{P(\text{cluster} = 1|X)}{P(\text{cluster} = 4|X)} \right) = \alpha_1 + \sum \beta_1 \cdot X_i \quad (6.8)$$

$$Z_i = \ln \left(\frac{P(\text{cluster} = 2|X)}{P(\text{cluster} = 4|X)} \right) = \alpha_2 + \sum \beta_2 \cdot X_i \quad (6.9)$$

¹⁰ Técnica desenvolvida inicialmente por Cox (1958).

$$Z_i = \ln \left(\frac{P(\text{cluster} = 3|X)}{P(\text{cluster} = 4|X)} \right) = \alpha_3 + \sum \beta_3.X_i \quad (6.10)$$

Foram utilizadas tanto a função *mlogit* da versão 11 do software estatístico STATA quanto a função *multinom* do pacote *nnet*¹¹ do R, para o cálculo da regressão logística multinomial de ANO, em função dos conjuntos de variáveis I e II (ver página 132). Os resultados de ambos softwares foram iguais. Analogamente ao que foi feito na análise de conglomerados, para as regressões logísticas, as variáveis foram padronizadas pelo método *z-scores* conforme Equações (6.1), (6.2) e (6.3) (ver página 133).

Os resultados relativos aos atributos de viagem da **família** (conjunto de variáveis I) são apresentados na Tabela 51.

Tabela 51 – Resultado da regressão logística multinomial, com categoria de referência Ano=4 (correspondente a 2007) para os atributos de viagem da família

Variável	ANO		1977		1987		1997	
	Coef.	Erro Padrão	Coef.	Erro Padrão	Coef.	Erro Padrão	Coef.	Erro Padrão
intercepto	-0,166 (***)	0,0086	-0,073 (***)	0,0083	-0,137 (***)	0,0085		
FAM_VIAG_TOT	0,488 (***)	0,0155	0,387 (***)	0,0155	0,270 (***)	0,0159		
FAM_DIST_TOT	0,102 (***)	0,0209	0,014 (-)	0,0211	-0,184 (***)	0,0234		
FAM_DIST_MED	0,018 (-)	0,0190	0,017 (-)	0,0183	-0,124 (***)	0,0203		
FAM_DURACAO_TOT	-0,286 (***)	0,0241	-0,235 (***)	0,0237	-0,047 (***)	0,0240		
FAM_DURACAO_MED	0,017 (-)	0,0191	0,091 (***)	0,0179	0,135 (***)	0,0178		

Nota: *** nível de significância de 0,01 | ** nível de significância de 0,05 | * nível de significância de 0,10 | - não significante

O grau de explicação do modelo, expresso pelo pseudo R^2 de McFadden, foi de 0,01, um valor baixo. Entretanto, o foco aqui não é criar um modelo preditivo, mas diagnosticar a influência de cada variável na formação dos agrupamentos. Não foram significativamente diferentes de 2007 (nível de significância de 5%), *ceteris paribus*, os coeficientes:

¹¹ Documentação do pacote *nnet* disponível em <<https://cran.r-project.org/web/packages/nnet/nnet.pdf>> - acesso em 27 de janeiro de 2016.

- (i) de 1977, a distância média de viagem da família (FAM_DIST_MED) e a duração média de viagem da família (FAM_DURACAO_MED);
- (ii) de 1987, a distância média de viagem da família (FAM_DIST_MED) e a distância total de viagens da família (FAM_DIST_TOT).

Para todos anos, FAM_VIAG_TOT é a variável cujos coeficientes apresentam os maiores valores absolutos, sendo sempre positivos, e que decrescem quanto mais nos aproximamos do ano de referência, 2007. Ao que parece, o **total de viagens da família** é o número que mais contribuiu para explicar os *clusters* formados a partir dos atributos de viagem da família.

Foi formulada, então, uma regressão logística multinomial para as variáveis relativas aos atributos de viagem da **pessoa**, também tendo como referência 2007. Os detalhes deste primeiro resultado podem ser observados no Anexo F). O grau de explicação do modelo, expresso pelo pseudo R^2 de McFadden, foi de 0,03, também um valor baixo, mas superior ao obtido na regressão feita com o conjunto I de dados (de atributos de viagem da família). Não foram significativamente diferentes de 2007 (nível de significância de 5%), *ceteris paribus*, os coeficientes:

- (i) de 1977: duração média de viagem da pessoa (PESS_DURACAO_MED) e a quantidade de vezes que a pessoa utiliza o modo ‘outros’ (PESS_MODO_OUTROS);
- (ii) de 1987: todas variáveis relativas aos motivos de viagem (PESS_MOTIVO_) e todas relativas ao período de viagem (PESS_PER_);
- (iii) de 1997: quantidade de vezes que a pessoa utiliza o modo ônibus (PESS_MODO_ONIBUS), todas variáveis relativas ao motivos de viagem (PESS_MOTIVO_) e todas relativas ao período de viagem (PESS_PER_).

Foi feita nova regressão logística multinomial retirando as variáveis que não foram significativas para distinguir algum dos anos de 2007, e reinserindo a variável com o número de viagens diárias da pessoa (TOT_VIAG), pois na retirada daquelas, esta poderia não mais apresentar multicolinearidade. Novamente foi tomado como referência o ano de 2007 e os resultados são apresentados na Tabela 52. O grau de explicação do modelo, expresso pelo pseudo R^2 de McFadden, também foi de 0,01, o mesmo grau de explicação do obtido com o conjunto de variáveis I. Não foram significativamente diferentes de 2007 (nível de significância de 5%), *ceteris paribus*, os coeficientes:

- (i) de 1987, o número de modos diferentes utilizados pela pessoa (PESS_NO_MODOS);
- (ii) de 1997, a duração total da viagem da pessoa (PESS_DURACAO_TOT) e o número de períodos diferentes em que a pessoa fez viagem (PESS_NO_PERIODOS).

A variável **total de viagens** diárias da pessoa (TOT_VIAG), deixou de ser excluída por problemas de multicolinearidade e passou a ser a que possui o coeficiente de maior contribuição em todos anos, sempre assumindo valores positivos e que decrescem

conforme nos aproximamos do ano referência. A variável PESS_DIST_TOT também apresenta coeficiente positivos que decrescem conforme nos aproximamos do ano de referência. Assim, é a segunda mais relevante em 1977 e 1987, após TOT_VIAG. Já em 1997, PESS_DIST_TOT passa a ter coeficiente negativo e PESS_NO_MODOS passa a ser a segunda variável de maior impacto positivo da distinção dos *clusters*, tendo como referência 2007.

O coeficientes positivos têm um efeito amplificador (fator multiplicador é superior a 1), os coeficientes negativos têm efeito redutor (fator multiplicador está entre 0 e 1). Posto isto, PESS_MODO_DIRG e PESS_NO_MOTIVOS são as variáveis de coeficientes mais negativos e, portanto, as que mais diminuem a probabilidade de pertencer a um determinado grupo.

Com as regressões logísticas chega-se a um potencial conjunto de variáveis de atributos de viagens de maior peso na formação dos *clusters*, a saber:

- (i) total de viagens da família;
- (ii) total de viagens da pessoa;
- (iii) distância total percorrida pela pessoa;
- (iv) número de motivos;
- (v) quantidade de vezes que a pessoa utiliza o automóvel.

Tabela 52 – Resultado da regressão logística multinomial com categoria de referência Ano=4 (correspondente a 2007) para os atributos de viagem da pessoa

Variável	ANO	1977		1987		1997	
		Coef.	Erro Padrão	Coef.	Erro Padrão	Coef.	Erro Padrão
intercepto		0,170 (***)	0,0046	0,209 (***)	0,0045	0,094 (***)	0,0047
PESS_DIST_TOT		0,436 (***)	0,0144	0,274 (***)	0,0150	-0,037 (***)	0,0166
PESS_DIST_MED		-0,248 (***)	0,0130	-0,138 (***)	0,0131	-0,107 (***)	0,0145
PESS_DUR_TOT		-0,255 (***)	0,0090	-0,271 (***)	0,0089	-0,013 (-)	0,0087
PESS_MODO_DIRIG		-0,345 (***)	0,0084	-0,377 (***)	0,0084	-0,141 (***)	0,0086
PESS_MODO_PASS		-0,081 (***)	0,0059	-0,161 (***)	0,0061	-0,047 (***)	0,0062
PESS_MODO_TREM		-0,249 (***)	0,0061	-0,080 (***)	0,0052	-0,065 (***)	0,0055
PESS_MODO_MOTO		-0,182 (***)	0,0069	-0,097 (***)	0,0041	-0,087 (***)	0,0050
PESS_MODO_BICI		-0,069 (***)	0,0046	-0,073 (***)	0,0046	-0,032 (***)	0,0041
PESS_MODO_APE		-0,132 (***)	0,0081	-0,123 (***)	0,0079	-0,065 (***)	0,0083
PESS_NO_MODOS		0,140 (***)	0,0102	0,003 (-)	0,0102	0,058 (***)	0,0105
PESS_NO_MOTIVOS		-0,408 (***)	0,0138	-0,299 (***)	0,0141	-0,231 (***)	0,0144
PESS_NO_PERIODOS		-0,271 (***)	0,0140	0,047 (***)	0,0144	-0,020 (***)	0,0146
TOT_VIAG		0,758 (***)	0,0186	0,497 (***)	0,0192	0,250 (***)	0,0196

Nota: *** nível de significância de 0,01 | ** nível de significância de 0,05 | * nível de significância de 0,10 | - não significante

6.5 Regressões quasi-poisson articulando sexo e situação familiar

Está entre os objetivos deste trabalho compreender as diferenças entre mulheres e homens do ponto de vista da demanda de transportes. Por um lado, contrariando expectativas iniciais, a variável sexo pouco se destacou nas análises de conglomerados. Por outro lado, a teoria subjacente apresentada na revisão de literatura indica que sexo não é sinônimo de gênero, mas um de seus componentes. Assim, para considerar gênero como categoria de análise no planejamento de transportes é necessária uma articulação da variável sexo com outras. Considerando a gama de dados disponíveis nas Pesquisas Origem Destino da RMSP e que este trabalho não avançará sobre métodos de pesquisa qualitativos, optou-se por elaborar a seguir uma sequência de regressões considerando segmentações por sexo (feminino e masculino) e situação familiar (pessoa responsável, cônjuge, filho(a)/enteado(a)). Optou-se por esta aproximação para a categoria gênero porque um dos principais *lócus* onde se estabelecem os papéis sociais para os indivíduos é a família. Assim, foram feitas segmentações de seis grupos, descritos no Quadro 11.

Quadro 11 – Descrição dos grupos utilizados nas regressões *quasi-poisson*

Grupo	Descrição	Qtde. de indivíduos
FPR	Pessoas do sexo feminino, com mais de 14 anos, cuja situação familiar é ‘pessoa responsável’	25.076
MPR	Pessoas do sexo masculino, com mais de 14 anos, cuja situação familiar é ‘pessoa responsável’	86.496
FCJ	Pessoas do sexo feminino, com mais de 14 anos, cuja situação familiar é ‘cônjuge’	75.812
MCJ	Pessoas do sexo masculino, com mais de 14 anos, cuja situação familiar é ‘cônjuge’	3.101
FFL	Pessoas do sexo feminino, com mais de 14 anos, cuja situação familiar é ‘filha/enteada’	35.409
MFL	Pessoas do sexo masculino, com mais de 14 anos, cuja situação familiar é ‘filho/enteado’	37.516

Interessa agora formular um modelo em que a variável dependente seja um atributo de viagem e que as explicativas sejam atributos de pessoas e de famílias. Isto porque deseja-se explicar o comportamento da demanda de transportes em função das características de indivíduos.

Destarte, do conjunto de variáveis indicado pela regressão logística, foi escolhida como **variável dependente** o número total de viagens da pessoa (TOT_VIAG). Primeiramente, porque a literatura utiliza largamente essa variável para compor índices de mobilidade. Também seria possível utilizar o número total de viagens da família, mas optou-se por analisar desagregadamente os indivíduos, considerando neles os efeitos da

estrutura familiar.

A partir dos resultados da análise de conglomerados, optou-se por não incluir como variável independente a situação familiar porque ela foi usada como critério de filtro na segmentação usada nesta série de análises. Também não se incluiu a renda familiar, pois esta informação já está contemplada pela variável de faixa de renda familiar. Foram consideradas, então, como **variáveis explicativas**:

- (i) presença de crianças até 14 anos (PRESENCA_FILH_ATE4, PRESENCA_FILH_5a9, PRESENCA_FILH_10a14);
- (ii) presença de 2 ou mais automóveis na família (PRESENCA_AUTO2);
- (iii) faixa de renda familiar (FAIXA_REN_FAM);
- (iv) renda individual (REN_IND);
- (v) grau de instrução da pessoa (GRAU_INSTR);

TOT_VIAG é uma variável de contagem, discreta, cuja distribuição não tem aderência à normalidade (ver Gráfico 12). Assim, não está atendido um dos principais pressupostos para a utilização da regressão linear múltipla. Embora seja comum fazer regressões por mínimos quadrados ordinários, RAMALHO (1996 apud FÁVERO, 2015), entre outros, alerta que o modelo clássico de regressão linear não é adequado quando a variável dependente é discreta assumindo um pequeno número de valores positivos.

Desta forma, avaliou-se a utilização do modelo de regressão *poisson*, que conta com a premissa expressa na Equação (6.11). Porém, os valores de variância foram sempre superiores à média de *TOT_VIAG* para cada um dos grupos A a F (ver Tabela 53), configurando superdispersão, fenômeno bastante comum em estudos empíricos segundo Tadano, Ugaya e Franco (2009 apud FÁVERO, 2015).

$$E(Y) = Var(Y) = \lambda \quad (6.11)$$

Tabela 53 – Valores de média e variância de *TOT_VIAG* para cada um dos grupos

Grupo	Média	Variância
Grupo FPR - sexo feminino, mais de 14 anos, pessoa responsável	1,70	2,8962
Grupo MPR - sexo masculino, mais de 14 anos, pessoa responsável	2,20	3,2592
Grupo FCJ - sexo feminino, mais de 14 anos, cônjuge	1,45	3,2764
Grupo MCJ - sexo masculino, mais de 14 anos, cônjuge	1,84	2,9190
Grupo FFL - sexo feminino, mais de 14 anos, filha ou enteada	2,16	2,4868
Grupo MFL - sexo masculino, mais de 14 anos, filho ou enteado	2,31	3,0170

Nota: Foram considerados todos os valores de *TOT_VIAG*, inclusive quando esta variável assume valor 0.

Posto isso, frente à inadequabilidade dos modelos de regressão linear e *poisson*, tem-se como técnicas possíveis para a formulação das regressões os modelos binomial negativo e *quasi-poisson* (GARDNER; MULVEY; SHAW, 1995; HOEF; BOVENG, 2007), ambos pertencentes à família dos modelos lineares generalizados (GLM). Ambos modelos têm igual número de parâmetros e, embora resultem valores similares para os coeficientes, podem apresentar grandes diferenças de covariância (HOEF; BOVENG, 2007) - sendo a variância do modelo *quasi-poisson* uma função linear da média e a do modelo binomial negativo, função quadrática da média.

Hoef e Boveng (2007), ao aplicar modelos *quasi-poisson* e binomial negativo para estudos de ecologia, apontam ser o primeiro mais indicado do que o segundo, por conta do melhor ajuste da relação média-variância. Gardner, Mulvey e Shaw (1995) recomendam utilizar o modelo *overdispersed Poisson* (ou *quasi-poisson*) se o principal interesse de pesquisa for avaliar a força dos coeficientes por meio de testes de hipóteses. Os autores acrescentam ainda que o binomial negativo é recomendado nos casos em que o principal interesse seja estimar a distribuição de probabilidade de um determinado valor de contagem da variável dependente.

Foi feita a escolha pelo modelo de regressão *quasi-poisson*, que é caracterizado apenas por sua média (ver Equação 6.12) e sua variância (ver Equação 6.13), sem necessariamente ter uma forma típica de distribuição. Por isso, dele não se obtém R^2 , pseudo R^2 ou mesmo valores de AIC (*Akaike Information Criteria*) e BIC (*Bayesian Information Criteria*) para efeito de comparação de modelos.

$$E(Y) = \mu \quad (6.12)$$

$$Var(Y) = \theta\mu \quad (6.13)$$

Da Tabela 53 depreende-se que a condição de filho(a)/enteado(a) para pessoas com mais de 14 anos confere mais mobilidade se comparado às situações de pessoa responsável e cônjuge, para ambos sexos. Isso porque: (i) o grupos dos(as) filhos(as)/enteados(as) concentra os(as) estudantes que, ao ir e voltar da escola, realizam pelo menos duas viagens (compulsórias) por dia; (ii) em geral, na RMSP, as escolas são de meio período, o que deixa um tempo livre diário para que façam outras atividades (e decorrentes deslocamentos); e (iii) foram selecionadas pessoas acima de 14 anos, logo, ela não são tão dependentes de responsáveis para sair de casa como as crianças. Vê-se também que a situação de cônjuge, muitas vezes dependente financeiramente da pessoa responsável, é a que menos estimula os deslocamentos e é mais frequente para o sexo feminino do que para o masculino. Percebe-se também que os números médios de viagem para pessoas do sexo feminino são sempre inferiores aos masculinos, para uma mesma situação familiar.

Os resultados completos das regressões *quasi-poisson* encontram-se no Anexo G e a síntese está expressa nas Tabelas 54 e 55. Antes de analisar os resultados cabe fazer algumas considerações:

- (i) Há 3,45 vezes mais homens que mulheres responsáveis pela família.
- (ii) Há 24,5 vezes mais mulheres do que homens cônjuges.
- (iii) Como não foi feita padronização de variáveis, a renda individual (única quantitativa) tem sua escala muito diferente das demais variáveis e, portanto, seus coeficientes são bem menores que os demais.

Não foram significativos (nível de significância de 5%), *ceteris paribus*, os coeficientes das seguintes (categorias de) variáveis:

- (i) para o grupo FPR: faixa de renda - classe A e presença de dois ou mais autos na família;
- (ii) para o grupo MPR: faixa de renda - classe E;
- (iii) para o grupo MCJ: faixa de renda - classes D e E;
- (iv) para o grupo MFL: grau de instrução - ensino superior completo;

Para a mesma situação familiar, são maiores os coeficientes dos grupos femininos das seguintes variáveis: renda individual, todas as categorias de grau de instrução e presença de crianças entre 5 e 9 anos na família. Isto significa que essas (categorias de) variáveis impactam mais no total de viagens das mulheres do que dos homens. Para a mesma situação familiar, são maiores os coeficientes dos grupos masculinos das seguintes variáveis: classes A, B, C e D de faixa de renda familiar, e presença de crianças na família com idades entre 0 e 4 anos e entre 10 e 14 anos. Isto significa que essas (categorias de) variáveis incrementam mais o total de viagens dos homens do que das mulheres.

A presença de dois um mais automóveis particulares na família aumenta mais o número de viagens para o sexo masculino do que para o feminino, exceto quando a mulher é cônjuge. Nesta condição familiar o coeficiente desta *dummy* para ela tem seu valor máximo. Isso pode se dever ao fato de que a prioridade de uso do primeiro automóvel seja do homem, responsável pela família e, havendo um segundo carro, este fica disponível para a mulher cônjuge.

Tabela 54 – Resultado das regressões *quasi-poisson* para os grupos de mulheres com situações familiares ‘pessoa responsável’, ‘cônjugue’ e ‘filha’

Variável	Mulher responsável		Mulher cônjuge		Mulher filha	
	Coef.	Erro Padrão	Coef.	Erro Padrão	Coef.	Erro Padrão
intercepto	-0,004 (-)	0,0279	-0,228 (***)	0,0203	0,369 (***)	0,0200
REN_IND	3,86.10 ⁻⁵ (***)	2,93.10 ⁻⁶	5,62.10 ⁻⁵ (***)	2,38.10 ⁻⁶	6,46.10 ⁻⁵ (***)	2,48.10 ⁻⁶
PRESENCA_AUTO2	0,043 (*)	0,0224	0,178 (***)	0,0117	0,064 (***)	0,0106
FAIXA_REN_FAM Classe E	0,066 (**)	0,0301	-0,134 (***)	0,0251	-0,111 (-)	0,0251
FAIXA_REN_FAM Classe D	0,127 (***)	0,0140	0,047 (**)	0,0212	0,074 (***)	0,0204
FAIXA_REN_FAM Classe C	0,095 (***)	0,0284	0,110 (***)	0,0199	0,197 (***)	0,0188
FAIXA_REN_FAM Classe B	0,130 (***)	0,0314	0,183 (***)	0,0208	0,277 (***)	0,0195
FAIXA_REN_FAM Classe A	0,069 (*)	0,0402	0,237 (***)	0,0224	0,319 (***)	0,0211
GRAU_INSTR Fundamental completo / Médio incompleto	0,355 (***)	0,0190	0,355 (***)	0,0125	0,246 (***)	0,0107
GRAU_INSTR Médio completo / Superior incompleto	0,524 (***)	0,0166	0,540 (***)	0,0116	0,214 (***)	0,139
GRAU_INSTR Superior completo	0,720 (***)	0,0183	0,706 (***)	0,0139	0,139 (***)	0,0148
PRESENCA_FILH_ate4 TRUE	0,100 (***)	0,0186	-0,026 (***)	0,0095	-0,094 (***)	0,0116
PRESENCA_FILH_5a9 TRUE	0,276 (***)	0,0164	0,366 (***)	0,0091	0,051 (***)	0,053
PRESENCA_FILH_10a14 TRUE	0,057 (***)	0,0062	0,141 (***)	0,0094	0,053 (***)	0,0087

Nota: *** nível de significância de 0,01 | ** nível de significância de 0,05 | * nível de significância de 0,10 | - não significante

Tabela 55 – Resultado das regressões *quasi-poisson* para os grupos de homens com situações familiares ‘pessoa responsável’, ‘cônjuge’ e ‘filho’

Variável	Homem responsável		Homem cônjuge		Homem filho	
	Coef.	Erro Padrão	Coef.	Erro Padrão	Coef.	Erro Padrão
intercepto	0,330 (***)	0,0133	-0,039 (-)	0,1029	0,439 (***)	0,0198
REN_IND	2,49.10 ⁻⁵ (***)	7,51.10 ⁻⁷	2,72.10 ⁻⁵ (***)	6,59.10 ⁻⁶	5,24.10 ⁻⁵ (***)	1,94.10 ⁻⁶
PRESENCA_AUTO2	0,069 (***)	0,0078	0,156 (***)	0,0464	0,098 (***)	0,0102
FAIXA_REN_FAM Classe E	-0,022 (***)	0,0158	0,087 (-)	0,1204	-0,139 (-)	0,0260
FAIXA_REN_FAM Classe D	0,131 (***)	0,0290	0,161 (-)	0,1060	0,094 (***)	0,0209
FAIXA_REN_FAM Classe C	0,233 (***)	0,0133	0,347 (***)	0,1026	0,244 (***)	0,0193
FAIXA_REN_FAM Classe B	0,333 (***)	0,0141	0,405 (***)	0,1071	0,336 (***)	0,0198
FAIXA_REN_FAM Classe A	0,286 (***)	0,0164	0,308 (***)	0,1163	0,346 (***)	0,0214
GRAU_INSTR Fundamental completo / Médio incompleto	0,151 (***)	0,0080	0,169 (***)	0,0520	0,190 (***)	0,0099
GRAU_INSTR Médio completo / Superior incompleto	0,236 (***)	0,0074	0,313 (***)	0,0459	0,133 (***)	0,0106
GRAU_INSTR Superior completo	0,228 (***)	0,0086	0,329 (***)	0,0545	0,004 (-)	0,0161
PRESENCA_FILH_ate4 TRUE	0,140 (***)	0,0060	0,091 (**)	0,0424	-0,047 (***)	0,0124
PRESENCA_FILH_5a9 TRUE	0,105 (***)	0,0061	0,181 (***)	0,0407	0,043 (***)	0,0106
PRESENCA_FILH_10a14 TRUE	0,123 (***)	0,0155	0,146 (***)	0,0379	0,059 (***)	0,0087

Nota: *** nível de significância de 0,01 | ** nível de significância de 0,05 | * nível de significância de 0,10 | - não significante

O coeficiente da variável que indica pertinência à classe E ou não é significativo, ou é quase sempre negativo (exceção para mulheres responsáveis pela família), indicando que a baixa renda é um fator que gera imobilidade na maioria dos casos.

Outro caso em que aparecem coeficientes negativos, cujos fatores multiplicadores resultantes serão redutores (valores entre 0 e 1), são da *dummy* que indica presença de crianças até 4 anos para mulheres cônjuges e filhas/enteadas. Provavelmente porque é socialmente atribuída à mulher, especialmente no papel de cônjuge, a tarefa de cuidado das crianças nas primeiras idades. Assim, as viagens geradas pela demanda dos(as) pequenos(as) acabam ficando a cargo do homem responsável pela família, quando essa figura existe. Caso seja a mulher a pessoa responsável, ela precisa dar conta das tarefas do cuidado e também daquelas que exigem deslocamentos, por conseguinte, o coeficiente delas nesta situação é positivo, embora inferior ao deles na mesma situação. Para os(as) filhos(as)/enteados(as), com mais de 14 anos, o impacto da presença de crianças de 0 a 4 anos é de reduzir o número de viagens feitas. Provavelmente porque os(as) filhos(as)/enteados(as) mais velhos ficam responsáveis por cuidar dos(as) mais novos(as). Sendo este impacto maior sobre as filhas do que sobre os filhos.

Avaliando o impacto da renda individual, caso fossem consideradas as rendas médias de cada grupo, teríamos que os resultados dos produtos dos coeficientes por essas médias para as pessoas do sexo feminino seriam: 0,036 (pessoa responsável), 0,022 (cônjuge) e 0,029 (filha/enteada). Para o grupo masculino os valores desse produto seriam: 0,046 (pessoa responsável), 0,030 (cônjuge) e 0,032 (filho/enteado). A renda individual, para a mesma situação familiar, aumenta mais o número de viagens de homens do que se mulheres. Avaliando esses coeficientes no conjunto de cada grupo, eles são, de fato, menos representativos que os demais.

Em relação aos valores do intercepto, o único valor negativo aparece para a mulher cônjuge, indicando, de partida, uma penalização da mobilidade para esta específica articulação entre sexo e situação familiar. Os valores (positivos) para as condição de filho(a)/enteado(a) são os maiores, indicando uma possível omissão de variável explicativa importante para estes grupos. Não foram significativos os interceptos nem para mulheres responsáveis pela família, nem para homens cônjuges. Isso pode ter ocorrido porque talvez o conjunto de variáveis escolhido descreva suficientemente os seus comportamentos.

Os modelos resultantes para cada grupo são expressos pelas Equações 6.14, 6.15, 6.16, 6.17, 6.18 e 6.19. Para a variável GRAU_INSTR foi tomada como referência a categoria “não alfabetizado / fundamental incompleto”. Para a variável FAIXA_REN_FAM foi tomada como referência a categoria dos sem renda (renda familiar declarada nula). Para cada equação serão avaliados os impactos das variáveis cujos coeficientes forem superiores a 0,2.

$$TOT_VIAG_{FRP} = e^{RI+FR+GI+PC} \quad \text{onde:} \quad (6.14)$$

$$RI = 3,86 * 10^{-5} * REN_IND$$

$$\begin{aligned} FR = & 0,066 * FAIXA_REN_FAMClasseE + \\ & 0,131 * FAIXA_REN_FAMClasseD + \\ & 0,095 * FAIXA_REN_FAMClasseC + \\ & 0,130 * FAIXA_REN_FAMClasseB \end{aligned}$$

$$\begin{aligned} GI = & 0,355 * GRAU_INSTRFundamentalcompleto/Médioincompleto + \\ & 0,524 * GRAU_INSTRMédiocompleto/Superiorincompleto + \\ & 0,720 * GRAU_INSTRSuperiorcompleto \end{aligned}$$

$$\begin{aligned} PC = & 0,100 * PRESENCA_FILH_ate4_TRUE + \\ & 0,276 * PRESENCA_FILH_5a9_TRUE + \\ & 0,123 * PRESENCA_FILH_10a14_TRUE \end{aligned}$$

Para as pessoas do sexo feminino na condição de pessoa responsável pela família (grupo FRP) a variável que mais impacta o total de viagens é o grau de instrução. Quanto mais instruída elas são, mais viagens elas fazem. Em seguida, o que mais impacta é a presença de filho(a)/enteado(a) pequeno entre 5 e 9 anos (não na primeira infância). Nesta fase as crianças ingressam no ambiente escolar e talvez ainda sejam muito pequenas para ir à escola sozinhas, necessitando de alguém que as acompanhe.

$$TOT_VIAG_{MRP} = e^{0,330+PA+RI+FR+GI+PC} \quad \text{onde:} \quad (6.15)$$

$$PA = 0,069 * PRESENCA_AUTO2$$

$$RI = 2,49 * 10^{-5} * REN_IND$$

$$\begin{aligned} FR = & 0,127 * FAIXA_REN_FAMClasseD + \\ & 0,233 * FAIXA_REN_FAMClasseC + \\ & 0,333 * FAIXA_REN_FAMClasseB + \\ & 0,286 * FAIXA_REN_FAMClasseA \end{aligned}$$

$$\begin{aligned} GI = & 0,151 * GRAU_INSTRFundamentalcompleto/Médioincompleto + \\ & 0,236 * GRAU_INSTRMédiocompleto/Superiorincompleto + \\ & 0,228 * GRAU_INSTRSuperiorcompleto \end{aligned}$$

$$\begin{aligned} PC = & 0,140 * PRESENCA_FILH_ate4_TRUE + \\ & 0,105 * PRESENCA_FILH_5a9_TRUE + \\ & 0,057 * PRESENCA_FILH_10a14_TRUE \end{aligned}$$

Para as pessoas do sexo masculino na condição de pessoa responsável pela família (grupo MRP) a variável que mais impacta o total de viagens é a faixa de renda familiar. Quanto maior a renda (até a classe B), mais viagens realizam. Ser da classe A implica realizar um pouco menos viagens que ser da classe B, talvez pela gama de serviços de que se pode dispor quando quem se desloca é o prestador do serviço e não o cliente. Em seguida, o que mais impacta é o grau de instrução e não há uma tendência de quanto mais escolarizado mais viagens realiza.

$$TOT_VIAG_{FCJ} = e^{-0,228+PA+RI+FR+GI+PC} \quad \text{onde:} \quad (6.16)$$

$$PA = 0,178 * PRESENCA_AUTO2$$

$$RI = 5,62 * 10^{-5} * REN_IND$$

$$\begin{aligned} FR = & -0,134 * FAIXA_REN_FAMClasseE + \\ & 0,047 * FAIXA_REN_FAMClasseD + \\ & 0,110 * FAIXA_REN_FAMClasseC + \\ & 0,183 * FAIXA_REN_FAMClasseB + \\ & 0,237 * FAIXA_REN_FAMClasseA \end{aligned}$$

$$\begin{aligned} GI = & 0,355 * GRAU_INSTRFundamentalcompleto / Médioincompleto + \\ & 0,540 * GRAU_INSTRMédioicompleto / Superiorincompleto + \\ & 0,706 * GRAU_INSTRSuperiorcompleto \end{aligned}$$

$$\begin{aligned} PC = & -0,026 * PRESENCA_FILH_ate4_TRUE + \\ & 0,366 * PRESENCA_FILH_5a9_TRUE + \\ & 0,141 * PRESENCA_FILH_10a14_TRUE \end{aligned}$$

Assim como para o grupo FRP, para as pessoas do sexo feminino na condição cônjuge (FCJ) a variável que mais impacta o total de viagens é o grau de instrução, seguido pela presença de filho(a)/enteado(a) pequeno entre 5 e 9 anos (não na primeira infância). A tendência deste grupo, diferentemente do FRP, é que quanto maior a renda, maior o número de viagens realizadas.

$$TOT_VIAG_{MCJ} = e^{PA+RI+FR+GI+PC} \quad \text{onde:} \quad (6.17)$$

$$PA = 0,156 * PRESENCA_AUTO2$$

$$RI = 2,72 * 10^{05} * REN_IND$$

$$\begin{aligned} FR = & 0,347 * FAIXA_REN_FAMClasseC + \\ & 0,405 * FAIXA_REN_FAMClasseB + \\ & 0,308 * FAIXA_REN_FAMClasseA \end{aligned}$$

$$GI = 0,169 * GRAU_INSTRFundamentalcompleto/Médioincompleto + 0,313 * GRAU_INSTRMédiocompleto/Superiorincompleto + 0,329 * GRAU_INSTRSuperiorcompleto$$

$$PC = 0,091 * PRESENCA_FILH_ate4_TRUE + 0,181 * PRESENCA_FILH_5a9_TRUE + 0,146 * PRESENCA_FILH_10a14_TRUE$$

Assim como para o grupo MRP, para as pessoas do sexo masculino na condição cônjugue (MCJ) a variável que mais impacta o total de viagens é a faixa de renda familiar, com o mesmo comportamento de queda do coeficiente da classe A em relação ao da classe B. Também aqui, em seguida, o que mais impacta é o grau de instrução, havendo, porém, uma tendência de que quanto mais escolarizado mais viagens realiza.

$$TOT_VIAG_{FFL} = e^{0,369+PA+RI+FR+GI+PC} \quad \text{onde:} \quad (6.18)$$

$$PA = 0,064 * PRESENCA_AUTO2$$

$$RI = 6,46 * 10^{05} * REN_IND$$

$$FR = -0,111 * FAIXA_REN_FAMClasseE + 0,074 * FAIXA_REN_FAMClasseD + 0,197 * FAIXA_REN_FAMClasseC + 0,277 * FAIXA_REN_FAMClasseB + 0,319 * FAIXA_REN_FAMClasseA$$

$$GI = 0,246 * GRAU_INSTRFundamentalcompleto/Médioincompleto + 0,214 * GRAU_INSTRMédiocompleto/Superiorincompleto + 0,139 * GRAU_INSTRSuperiorcompleto$$

$$PC = -0,094 * PRESENCA_FILH_ate4_TRUE + 0,051 * PRESENCA_FILH_5a9_TRUE + 0,053 * PRESENCA_FILH_10a14_TRUE$$

Diferentemente do que ocorre para os grupos FRP e FCJ, para as pessoas do sexo feminino na condição de filha/enteada (FFL) a variável que mais impacta o total de viagens é a renda familiar (classes A e B), sendo que quanto maior a renda, maior o número de viagens realizadas. O grau de instrução também é bastante relevante, mas apresenta tendência oposta ao dos grupos FRP e FCJ: quanto maior a escolarização, menor o peso da variável.

$$TOT_VIAG_{MFL} = e^{0,439+PA+RI+FR+GI+PC} \quad \text{onde:} \quad (6.19)$$

$$PA = 0,098 * PRESENCA_AUTO2$$

$$RI = 5,25 * 10^5 * REN_IND$$

$$\begin{aligned} FR = & -0,139 * FAIXA_REN_FAMClasseE + \\ & 0,094 * FAIXA_REN_FAMClasseD + \\ & 0,244 * FAIXA_REN_FAMClasseC + \\ & 0,336 * FAIXA_REN_FAMClasseB + \\ & 0,346 * FAIXA_REN_FAMClasseA \end{aligned}$$

$$\begin{aligned} GI = & 0,190 * GRAU_INSTRFundamentalcompleto/Médioincompleto + \\ & 0,133 * GRAU_INSTRMédiocompleto/Superiorincompleto \end{aligned}$$

$$\begin{aligned} PC = & -0,047 * PRESENCA_FILH_ate4_TRUE + \\ & 0,043 * PRESENCA_FILH_5a9_TRUE + \\ & 0,059 * PRESENCA_FILH_10a14_TRUE \end{aligned}$$

Assim como ocorre para os grupos MRP e MCJ, para as pessoas do sexo masculino na condição de filho/enteado (MFL) a variável que mais impacta o total de viagens é a renda familiar (classes A, B e C), só que sem a queda na classe A. Ou seja, quanto maior a renda, mais viagens realiza. Com coeficientes inferiores a 0,2, o grau de instrução segue a mesma tendência do grupo FFL: quanto maior a escolarização, menor o peso (e até a relevância) da variável.

7 Considerações Finais

Um dos principais indicadores de mobilidade, o total de viagens que uma pessoa realiza num dia, é fortemente influenciado pela renda familiar, pelo grau de instrução do indivíduo e, em menores medidas, pelas presenças de crianças até 14 anos e de dois ou mais automóveis na família. Essas influências, porém, não se dão de forma homogênea em toda a população, incidindo diferentemente sobre os grupos sociais. Essas conclusões foram possíveis após análise de um conjunto de modelos de regressões *quasi-poisson* aplicados a diferentes grupos, determinados pela articulação das variáveis sexo e situação familiar, numa primeira aproximação do gênero como categoria de análise. Está na origem desta pesquisa a limitação do próprio conjunto de dados (secundários) que aborda especificamente o sexo como variável de interesse, mas não o gênero como categoria de análise. Assim, a partir deste trabalho, recomenda-se estudar outras combinações de variáveis desse mesmo banco de dados para aprimorar a abordagem de gênero na compreensão da mobilidade na RMSP. Além disso, é possível elaborar pesquisas de caráter qualitativo para esmiuçar melhor como é o processo decisório dentro do núcleo familiar relativo à compra e ao uso do(s) carro(s), ou ainda, relativo aos deslocamentos decorrentes do cuidados com as crianças. Com isso, poder-se-á ir além das constatações deste trabalho em relação ao comportamento de demanda, em busca das motivações desses comportamentos. Será um grande avanço ainda se houver coleta de dados sobre raça/etnia, possibilitando abordagens interseccionais.

A divisão do trabalho de acordo com o gênero implica diferentes padrões de atividades e, assim, diferentes padrões de viagens. Analisando os dados expandidos do total de viagens da pessoa, observou-se que, ao considerar inclusive quem não fez viagens, as mulheres sempre tinham médias inferiores aos homens. Ao passo que se considerarmos apenas quem fez viagem, essa situação se mantém para 1977 e 1987, mas em 1997 e 2007, as mulheres móveis, o são mais do que os homens. Elas têm conseguido diminuir as desigualdades no mercado de trabalho ao longo do tempo, mas não vêm obtendo o mesmo êxito em relação ao trabalho doméstico. Além da obrigatoriedade na realização das viagens motivo trabalho, elas não ficaram desobrigadas daquelas ligadas ao espaço doméstico, como se pode ver nos maiores percentuais femininos de viagens motivo manutenção/compras para todos os anos.

As mulheres vem utilizando mais o transporte coletivo do que os homens (com exceção do período de 1987), sendo o uso do modo ônibus o a mais representativo. Então, embora tenham uma diversidade maior de atividades a cumprir, ao se deslocar por meio motorizado, elas utilizam mais frequentemente um modo cujas rotas não são flexíveis. Isso significa que é possível cumprir uma agenda mais complexa na RMSP sem que seja

necessário dispor de um automóvel e que, talvez, o modo que confira maior flexibilidade de rota não seja o carro, seja a pé. As mulheres caminham mais, considerando o conceito de viagem a pé adotado pelas Pesquisas OD, o que provavelmente sub-representa este tipo de viagem. Entretanto, para estimular o modo a pé em substituição ao uso do carro, é preciso considerar a principal limitação do modo a pé: as grandes distâncias. Assim, políticas de transporte precisam necessariamente ser articuladas com o planejamento urbano para melhor distribuir as oportunidades na cidade. O acesso às oportunidades (de trabalho, estudo, lazer, compras, saúde) de forma mais equânime no espaço urbano torna possível a utilização de modos não motorizados, mais sustentáveis. E dada a existência de oportunidades mais próximas à residência, é preciso também que o ambiente construído seja convidativo a realizar as viagens a pé ou de bicicleta, ou seja, as pessoas de qualquer gênero devem sentir-se seguras e acolhidas pela cidade que as cerca. Posto isso, ao abordar gênero e mobilidade, recomenda-se fortemente estudos focados nas viagens a pé, que invariavelmente precisarão de coletas de dados para além das Pesquisas OD, com ênfase na subjetividade e na preocupação com o contexto urbano.

Neste estudo, observou-se que, apesar da variável ano não entrar nas clusterizações, ela foi muito relevante na formação dos grupos, indicando que o efeito do tempo pode ter grande peso nos padrões de deslocamentos. Pelos dados das quatro Pesquisas OD observou-se que o número total de viagens por família diminuiu enquanto os tempos e distâncias de viagem por pessoa aumentaram. As distâncias de viagem aumentaram mais para o usuário do transporte coletivo do que para o individual e, especificamente dentro dos modos coletivos, o efeito do alongamento das viagens é mais sentido na alta capacidade (metrô e trem) - provavelmente devido à expansão da rede metroferroviária ocorrida no período. Quando ocorre, a taxa de incremento das durações e distâncias é maior para viagens de motivação compulsória (trabalho e educação). As durações de viagem aumentaram tanto para quem usa modos coletivos como para quem usa os individuais, afetando todos motivos. O quadro de tendência de aumento de durações e distâncias indica que houve expansão da RMSP, mas sobretudo mostra que a capacidade da malha de transportes oferecida não tem acompanhado a demanda. Cabem, por conseguinte, estudos futuros que aprofundem as análises longitudinais, buscando avaliar os impactos de efeitos fixos e aleatórios e orientar o planejamento de transportes na busca por mais eficiência.

Os padrões de viagem se alteram conforme o tempo, o que foi explorado nas análises de *clusters* e com regressões logísticas, e conforme o gênero (situações exploradas com regressões *quasi-poisson*). Portanto, o objetivo principal desta dissertação foi atingido ao constatar que a transformação dos papéis sociais desempenhados por homens e mulheres dentro do núcleo familiar e na sociedade, ao longo das últimas décadas, alterou de maneira significativa a maneira como pessoas com identidades de gênero masculina e feminina têm se deslocado.

Referências

- ABEP. *Critério Brasil*. São Paulo, 2009. 4 p. Citado na página 60.
- AHMED, Q. I.; LU, H.; YE, S. Urban transportation and equity: A case study of Beijing and Karachi. *Transportation Research Part A: Policy and Practice*, v. 42, n. 1, p. 125–139, jan 2008. ISSN 09658564. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0965856407000559>>. Citado na página 23.
- ALVA, E. N. *Metrópoles (in)sustentáveis*. Rio de Janeiro: Relume-Dumará, 1997. 152 p. Disponível em: <<http://livraria.folha.com.br/livros/urbanismo/metropoles-in-sustentaveis-eduardo-neira-alva-1043600.html>>. Citado na página 23.
- ALVES, J. E. D. *O crescimento da PEA e a redução do hiato de gênero nas taxas de atividade no mercado de trabalho*. 2013. 1–5 p. Disponível em: <<http://www.ie.ufrj.br/aparte/1>>. Citado na página 35.
- AURÉLIO. *Dicionário Online*. 2014. Disponível em: <<http://www.dicionariodoaurelio.com/genero>>. Citado na página 28.
- BANISTER, D. The trilogy of distance, speed and time. *Journal of Transport Geography*, Elsevier Ltd, v. 19, n. 4, p. 950–959, jul 2011. ISSN 09666923. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0966692310001973>>. Citado na página 23.
- BEAUVOIR, S. *O Segundo Sexo: a experiência vivida*. 2^a. ed. Rio de Janeiro: Difusão Europeia do Livro, 1967. 500 p. Citado na página 30.
- BEST, H.; LANZENDORF, M. Division of labour and gender differences in metropolitan car use: An empirical study in Cologne, Germany. *Journal of Transport Geography*, v. 13, n. 2, p. 109–121, jun 2005. ISSN 09666923. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0966692304000201>>. Citado 5 vezes nas páginas 23, 24, 45, 47 e 48.
- BLAY, E. A. Um caminho ainda em construção: a igualdade de oportunidades para as mulheres. *Revista USP*, São Paulo, n. 49, p. 82–97, 2001. Disponível em: <<http://www.usp.br/revistausp/49/06-evablay.pdf>>. Citado 3 vezes nas páginas 33, 34 e 36.
- BRASIL. *Lei nº 12.587 de 03 de janeiro de 2012. Institui as diretrizes da Política Nacional de Mobilidade Urbana*. 2012. 1–11 p. Disponível em: <http://www.planalto.gov.br/ccivil__03__ato2011-2014/2012/lei/l125>. Citado na página 23.
- BRITO, M. N. C. Gênero e cidadania: referenciais analíticos. *Revista Estudos Feministas*, v. 9, n. 1, p. 291–298, 2001. Disponível em: <<http://www.redalyc.org/pdf/381/38109117.pdf>>. Citado na página 33.
- CADESTIN, C.; DEJOUX, V.; ROUX, S. *The mobility of elderly people: what are the differences between men and women?* Rio de Janeiro: [s.n.], 2013. 1–14 p. Citado na página 55.

- CAMERON, M. *Efficiency and fairness on the road*. Oakland, 1994. Citado na página 41.
- CARRASCO, C. *Estatísticas sob suspeita: proposta de novos indicadores com base na experiência das mulheres*. 1. ed. São Paulo: SOF - Sempreviva Organização Feminista, 2012. 160 p. Disponível em: <<http://www.sof.org.br/wp-content/uploads/2015/07/Estatisticas.pdf>>. Citado na página 124.
- CARVALHO, J. M. de. *Cidadania no Brasil - O longo Caminho*. Rio de Janeiro: Civilização Brasileira, 2002. Citado na página 33.
- CHANT, S. *Gender and Migration in Developing Countries*. London: Belhaven Press, 1992. Citado na página 37.
- CHU, Y.-L. Automobile Ownership Analysis Using Ordered Probit Models. *Transportation Research Record*, v. 1805, n. 1, p. 60–67, 2002. ISSN 0361-1981. Citado na página 96.
- COX, D. R. The regression analysis of binary sequences (with discussion). *Journal of the Royal Statistical Society, Series B* (, n. 20, p. 215–242, 1958. Citado na página 154.
- CRANE, R. Is There a Quiet Revolution in Women's Travel? Revisiting the Gender Gap in Commuting. *Journal of the American Planning Association*, v. 73, n. 3, p. 298–316, sep 2007. ISSN 0194-4363. Disponível em: <<http://www.tandfonline.com/doi/abs/10.1080/01944360708977979>>. Citado 5 vezes nas páginas 23, 45, 46, 50 e 63.
- CRENSHAW, K. Documento para o encontro de especialistas em aspectos da discriminação racial relativos ao gênero. *Estudos Feministas*, v. 1, p. 171–188, 2002. Citado na página 32.
- CRESSWELL, T.; UTENG, T. P. Gendered Mobilities: towards an holistic understanding. In: UTENG, T. P.; CRESSWELL, T. (Ed.). *Gendered Mobilities*. 1^a. ed. Hampshire: Ashgate Publishing Limited, 2008. cap. 1, p. 1–12. Citado 3 vezes nas páginas 23, 41 e 42.
- CUSSET, J.-M. Mobility deux roues et politique de transport à Ouagadougou et à Hanoi. In: INRETS (Ed.). *Mobilité et politiques de transport dans les villes en développement: journées spécialisées INRETS*. [S.l.: s.n.], 1997. p. 87–104. Citado na página 50.
- DALMASO, R. C. *Identificação e caracterização de grupos de indivíduos segundo padrões de seqüências de atividades multidimensionais*. 153 p. Tese (Dissertação de Mestrado) — Escola Politécnica da Universidade de São Paulo, 2009. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3138/tde-21072009-144859/en.php>>. Citado na página 78.
- DARGAY, J.; GATELY, D. Income's effect on car and vehicle ownership, worldwide: 1960 - 2015. *Transportation Research Part A*, v. 33, p. 101–138, 1999. Citado na página 96.
- DARGAY, J. M. The effect of income on car ownership: Evidence of asymmetry. *Transportation Research Part A: Policy and Practice*, v. 35, n. 9, p. 807–821, 2001. ISSN 09658564. Citado na página 96.
- DEAR, M.; SCOTT, A. J. *Urbanization and Urban Planning in Capitalist Societies*. Nova Iorque: Methuen, 1981. Citado na página 36.

- D'INCAO, M. Â. Mulher e Família Burguesa. In: PRIORE, M. D.; PINSKY, C. B. (Ed.). *História das Mulheres no Brasil*. 10ª. ed. São Paulo: Contexto, 2012. cap. 7, p. 677. Citado na página 44.
- ENDERS, J. Academic Staff Mobility in the European Community: The ERASMUS Experience. *Comparative Education Review*, The University of Chicago Press, v. 42, n. 1, p. 46–60, 1998. Disponível em: <<http://www.jstor.org/stable/1188786>>. Citado na página 37.
- ENLOE, C. H. *Bananas, Beaches and Bases: Making Feminist Sense of International Politics*. London: Pandora, 1989. Citado na página 37.
- EVERITT, B. S. et al. *Cluster Analysis*. 5. ed. West Sussex: Wiley Series in probability and statistics, 2011. 330 p. Citado 3 vezes nas páginas 133, 134 e 135.
- FAGNANI, J. Women's commuting patterns in the Paris region. *Tijdschrift voor Economische en Sociale Geografie*, v. 24, p. 12–24, 1983. Citado 3 vezes nas páginas 23, 50 e 63.
- FARIA, P. N. *Avaliação de Métodos para Determinação do Número Ótimo de Clusters em Estudo de Divergência Genética Entre Acessos de Pimenta*. 54 p. Tese (Mestrado) — Universidade Federal de Viçosa, 2009. Disponível em: <http://www.tede.ufv.br/tesesimplificado/tde{_\}arquivos/41/TDE-2009-07-01T150546Z-1744/Publico/textocompleto>. Citado na página 133.
- FÁVERO, L. P. *Análise de Dados: Modelos de Regressão com Excel, STATA e SPSS*. 1. ed. Rio de Janeiro: Elsevier, 2015. 504 p. Citado na página 160.
- FÁVERO, L. P. et al. *Análise de Dados: modelagem multivariada para tomada de decisões*. Rio de Janeiro: Elsevier, 2009. 650 p. Citado 4 vezes nas páginas 131, 133, 135 e 138.
- FOX, M. B. Working Women and Travel: The Access of Women to Work and Community Facilities. *Journal of the American Planning Association*, v. 49, n. 2, p. 156–170, 1983. Citado 4 vezes nas páginas 23, 48, 49 e 63.
- FRAISSE, G. *El concepto filosófico de género*. 2001. Disponível em: <http://www.europarl.europa.eu/transl{_\}es/plataforma/pagina/celter/art2fraise>. Citado 2 vezes nas páginas 28 e 30.
- FRANDBERG, L.; VILHELMSON, B. More or less travel: Personal mobility trends in the Swedish population focusing gender and cohort. *Journal of Transport Geography*, v. 19, n. 6, p. 1235–1244, 2011. Citado na página 55.
- FREITAG, B. *Teorias da Cidade*. 2. ed. Campinas: Papirus, 2007. 190 p. Citado na página 23.
- FROHLICK, S. 'I'm More Sexy Here': Erotic Subjectivities of Female Tourists in the 'Sexual Paradise' of the Costa Rican Caribbean. In: UTENG, T. P.; CRESSWELL, T. (Ed.). *Gendered Mobilities*. 1. ed. Hampshire: Ashgate Publishing Limited, 2008. cap. 9, p. 129–142. Citado na página 37.
- Fundação Perseu Abramo. *Mulheres brasileiras e gênero nos espaços público e privado*. 2010. 300 p. Disponível em: <<http://www.fpa.org.br/sites/default/files/pesquisaintegra.pdf>>. Citado na página 34.

- FURTADO, C. *Desenvolvimento e subdesenvolvimento*. Rio de Janeiro: Centro Celso Furtado / Contraponto, 2009. 234 p. ISBN 978-85-7866-019-2. Disponível em: <http://www.centrocelfurtado.org.br/interna.php?ID{_\}M=>>. Citado na página 32.
- GARDNER, W.; MULVEY, E. P.; SHAW, E. C. Regression analyses of counts and rates: Poisson, overdispersed poisson, and negative binomial models. *Psychological Bulletin*, v. 118, n. 3, p. 392–404, 1995. ISSN 0033-2909. Citado na página 161.
- GERMANI, E. B. *Análise do Comportamento da Demanda pos Transportes Utilizando Métodos de Alinhamento de Sequências Multidimensionais: Uma Aplicação à Região Metropolitana de São Paulo*. 120 p. Tese (Mestrado) — Universidade de São Paulo, 2005. Citado 2 vezes nas páginas 77 e 78.
- GILBERT, M. R. ‘Race’, space and power: The survival strategies of working poor women. *Annals of the Association of American Geographers*, v. 88, n. 4, p. 595–621, 1998. Citado na página 45.
- GNANADESIKAN, R.; KETTENRING, J. R.; TSAO, S. L. Weighting and selection of variables. *Journal of Classification*, v. 12, p. 113–136, 1995. Citado na página 133.
- GODDARD, T. B. et al. Voyage of the SS Minivan: Women’s Travel Behavior in Traditional and Suburban Neighborhoods. *Journal of the Transportation Research Board*, n. 1956, p. 141–148, 2006. Disponível em: <<http://trb.metapress.com/content/2731180920015rml/fulltext.pdf>>. Citado na página 48.
- GOUVEA, M. A.; La Plata, J. P. F. *Segmentos Médicos para a categoria de produtos cirúrgicos no Brasil*. São Paulo: [s.n.], 2006. Citado na página 138.
- GRECO, M. A.; GODOI, M. S. *Solidariedade Social e Tributação*. São Paulo: Dialética, 2005. Citado na página 41.
- HAGERSTRAND, T. What about people in regional science? *Papers, Regional Science Association*, v. 24, p. 7–21, 1970. Citado na página 39.
- Hair Jr, J. F. et al. *Análise Multivariada de Dados*. 5. ed. Porto Alegre: Bookman, 2005. Citado na página 135.
- HANSON, S. *Gender, work and space*. London: Routledge, 1995. Citado 2 vezes nas páginas 23 e 45.
- HANSON, S. Getting There: Urban Trnasportation in Context. In: *The Geography of Urban Transportation*. 2^a. ed. Nova Iorque: The Guilford Press, 1995. p. 478. Citado 3 vezes nas páginas 38, 39 e 41.
- HANSON, S. Gender and mobility: new approaches for informing sustainability. *Gender, Place & Culture*, v. 17, n. 1, p. 5–23, feb 2010. ISSN 0966-369X. Disponível em: <<http://www.tandfonline.com/doi/abs/10.1080/09663690903498225>>. Citado 5 vezes nas páginas 23, 42, 44, 45 e 51.
- HANSON, S.; JOHNSTON, I. Gender Differences in Work-Trip Length: Explanations and Implications. *Urban Geography*, v. 6, n. 3, p. 193–219, may 1985. ISSN 0272-3638. Disponível em: <<http://www.tandfonline.com/doi/abs/10.2747/0272-3638.6.3.193>>. Citado na página 23.

- HARAWAY, D. "Gênero" para um dicionário marxista: a política sexual de uma palavra. *Cadernos Pagu*, v. 22, p. 201–246, 2004. Disponível em: <<http://www.scielo.br/pdf/cpa/n22/n22a09.pdf>>. Citado 3 vezes nas páginas 29, 30 e 31.
- HARTIGAN, J. A. Statistical Theory in Clustering. *Journal of Classification*, v. 2, n. 1, p. 63–76, 1985. Disponível em: <<http://link.springer.com/article/10.1007/BF01908064>>. Citado na página 133.
- HARTIGAN, J. A.; WONG, M. A. A K-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, v. 28, n. 1, p. 100–108, 1979. Citado na página 138.
- HEILBORN, M. L. Usos e Abusos da Categoria de Gênero. In: HOLLANDA, H. H. O. B. de (Ed.). *Y Nosotras latinoamericanas? Estudos sobre Gênero e Raça*. São Paulo: Fundação Memorial da América Latina, 1992. p. 39–44. Disponível em: <http://www.clam.org.br/bibliotecadigital/uploads/publicacoes/114{_}1042{_}usoseabusosdacategoriadegenero.p>. Citado na página 31.
- HIRATA, H.; GUIMARÃES, N. A. *Cuidado e cuidadoras: as várias faces do trabalho do care*. São Paulo: Atlas, 2012. 236 p. Citado na página 48.
- HJORTHOL, R. J. Same city, different options: An analysis of the work trips of married couples in the metropolitan area of Oslo. *Journal of Transport Geography*, v. 8, p. 213–220, 2000. Citado 4 vezes nas páginas 23, 45, 48 e 51.
- HODGE, D. My fair share: equity issues in urban transportation. In: HANSON, S. (Ed.). *The Geography of Urban Transportation*. Nova Iorque: The Guilford Press, 1995. Citado na página 23.
- HOEF, J. M.; BOVENG, P. L. Quasi-Poisson vs. Negative Binomial Regression: how shold we model overdispersed count data? *Ecology*, v. 88, n. 11, p. 2766–2772, 2007. ISSN 00129658. Disponível em: <<http://onlinelibrary.wiley.com/doi/10.1890/07-0043.1/abstract>>. Citado na página 161.
- HOFFMAN, D. M. Changing Academic Mobility Patterns and International Migration: What Will Academic Mobility Mean in the 21st Century? *Journal of Studies in International Education*, v. 13, n. 3, p. 347–364, jul 2008. ISSN 1028-3153. Disponível em: <<http://jsi.sagepub.com/cgi/doi/10.1177/1028315308321374>>. Citado na página 37.
- HOOKS, B. *Yearning: Race, gender, and cultural politics*. Boston: South End Press, 1990. 248 p. Citado na página 32.
- HOWE, A.; O'CONNOR, K. Travel to work and labour force participation of men and women in an Australian metropolitan area. *Professional Geographer*, v. 34, p. 50–64, 1982. Citado 2 vezes nas páginas 23 e 50.
- IEMA, I. d. E. e. M. A. *A Bicicleta e as Cidades: como inserir a bicicleta na política de mobilidade urbana*. 2^a. ed. São Paulo: IEMA, 2010. 84 p. Disponível em: <<http://www.energiaeambiente.org.br/>>. Citado na página 38.
- JOHNSTON-ANUMONWO, I. The Influence of Household Type on Gender Differences in Work Trip Distance". *Professional Geographer*, v. 44, n. 2, p. 161–169, 1992. Citado 3 vezes nas páginas 23, 49 e 63.

JONES, P. M. Activity approaches to understanding travel behavior. In: (eds) . Lexington Books, Cap. 13, p.253-266. In: STOPHER, P. R.; MEYBURG, A. H.; BRÖG, W. (Ed.). *New horizons in travel behavior*. Lexington Books, 1981. p. 253–266. Disponível em: <<http://trid.trb.org/view.aspx?id=320615>>. Citado na página 55.

KARLAFTIS, M.; GOLIAS, J. Automobile Ownership, Households Without Automobiles, and Urban Traffic Parameters: Are They Related? *Transportation Research Record*, v. 1792, n. 1, p. 29–35, 2002. ISSN 0361-1981. Citado na página 96.

KEHL, M. R. *Deslocamentos do Feminino- A Mulher Freudiana na Passagem para a Modernidade*. Rio de Janeiro: Imago, 1998. Citado na página 34.

KERGOAT, D. Division sexuelle du travail et rapports sociaux de sexe. In: HIRATA, H. et al. (Ed.). *Dictionnaire critique du féminisme*. 2. ed. Paris: Presses Universitaires de France, 2004. p. 35–44. Citado 2 vezes nas páginas 31 e 34.

KINGHAM, S.; DICKINSON, J.; COPSEY, S. Travelling to work: will people move out of their cars. *Transport Policy*, v. 8, p. 151–160, 2001. Citado na página 23.

KOSTYNIUK, L. P.; KITAMURA, R. Trip chains and activity sequences: test of temporal stability. *Transportation Research Record*, v. 987, p. 29–39, 1984. Citado na página 54.

KÜNZLER, J. *Familiale Arbeitsteilung: Die Beteiligung von Männern an der Hausarbeit*. Bielefeld: Kleine, 1994. Citado na página 45.

LEE, B. S.; McDONALD, J. Determinants of commuting time and distance for Seoul residents: the impact of family status on the commuting of women. *Urban Studies*, v. 40, n. 7, p. 1283–1302, jun 2003. ISSN 0042-0980. Disponível em: <<http://usj.sagepub.com/cgi/doi/10.1080/0042098032000084604>>. Citado 3 vezes nas páginas 23, 50 e 63.

LÉVI-STRAUSS, C. *As Estruturas Elementares do Parentesco*. 6^a. ed. Petrópolis: Vozes, 2010. 544 p. Disponível em: <<http://classicos12011.files.wordpress.com/2011/03/lc3a9vi-strauss-claude-as-estruturas-elementares-do-parentesco.pdf>>. Citado na página 31.

LEWIS, D. *Economic Perspectives on Transport and Equality*. Leipzig, 2011. 30 p. Disponível em: <<http://www.internationaltransportforum.org/>>. Citado na página 23.

MAHMASSANI, H. S. Some comments on activity-based approaches to the analysis and prediction of travel behavior. *Transportation*, v. 15, n. 1-2, p. 35–40, 1988. Citado na página 55.

MALHOTRA, N. K. *Pesquisa de Marketing: Uma Orientação*. 3^a. ed. Porto Alegre: Bookman, 2001. 720 p. Citado 2 vezes nas páginas 53 e 54.

MANDEL, J. Mobility matters: Women's livelihood strategies in Porto Novo, Benin. *Gender, Place & Culture*, v. 11, n. 2, p. 257–87, 2004. Disponível em: <<http://www.tandfonline.com/doi/pdf/10.1080/0966369042000218482>>. Citado na página 45.

MAXWELL, A. E. *Multivariate Analysis in Behavioural Reserach*. 1. ed. London: Chapman & Hall, 1977. Citado na página 134.

- MCGUCKIN, N.; MURAKAMI, E. Examining Trip-Chaining Behavior. *Transportation Reserach Record*, v. 1693, n. 99, p. 79–85, 1995. Citado na página 51.
- MCGUCKIN, N.; ZMUD, J.; NAKAMOTO, Y. Trip-Chaining Trends in the United States: Understanding Travel Behavior for Policy Making. *Transportation Research Record: Journal of the Transportation Research Board*, v. 1917, p. 199–204, 2005. Citado 4 vezes nas páginas 23, 51, 54 e 63.
- MCLAFFERTY, S.; PRESTON, V. Gender, race and commuting among service sector workers. *The Professional Geographer*, v. 43, p. 1–14, 1991. Citado 2 vezes nas páginas 23 e 45.
- MCLAFFERTY, S.; PRESTON, V. Spatial Mismatch and Labor Market Segmentation for African-American and Latina Women. *Economic Geography*, v. 68, n. 4, p. 406–431, 1992. Citado 2 vezes nas páginas 45 e 46.
- MEAD, M. *Sexo e temperamento em três sociedades primitivas*. 4^a. ed. São Paulo: Perspectiva, 2000. 316 p. Disponível em: <<http://www.livrariacultura.com.br/scripts/resenha/resenha.asp?nitem=72303http://pt.scribd.com/doc/178229042/Resumo-Sexo-e-Temperamento-Margareth-Mead>>. Citado na página 29.
- METRÔ-SP. *Pesquisa Origem Destino 1977*. São Paulo, 1977. Citado 4 vezes nas páginas 58, 59, 61 e 76.
- METRÔ-SP. *Pesquisa Origem Destino 1987*. São Paulo, 1987. Citado 4 vezes nas páginas 58, 59, 61 e 76.
- METRÔ-SP. *Pesquisa Origem Destino 1997*. São Paulo, 1997. Citado 4 vezes nas páginas 58, 59, 61 e 76.
- METRÔ-SP. *Pesquisa Origem Destino 2007*. São Paulo, 2007. Citado 4 vezes nas páginas 58, 59, 61 e 76.
- METRÔ-SP. *Pesquisa Origem Destino 2007 - Região Metropolitana de São Paulo: Síntese das Informações*. São Paulo, 2008. 83 p. Citado 2 vezes nas páginas 57 e 58.
- METZ, D. Demographic determinants of daily travel demand. *Transport Policy*, Elsevier, v. 21, p. 20–25, may 2012. ISSN 0967070X. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0967070X1200008X>>. Citado 2 vezes nas páginas 23 e 37.
- MICHAELIS. *Dicionário Prático da Língua Portuguesa*. São Paulo: Melhoramentos, 2011. Disponível em: <<http://michaelis.uol.com.br/moderno/portugues/index.php>>. Citado 2 vezes nas páginas 28 e 37.
- MOJENA, R. Hierarchical Grouping Methods and Stopping Rules: an Evaluation. *The Computer Journal*, v. 20, n. 4, p. 359–363, 1977. Disponível em: <<http://comjnl.oxfordjournals.org/content/20/4/359.full.pdf+html>>. Citado na página 134.
- MORAES, M. L. Q. de. Usos e Limites da categoria gênero. *Cadernos Pagu*, Campinas, n. 11, p. 99–106, 1998. Disponível em: <www.bibliotecadigital.unicamp.br/document/?down=51203>. Citado 2 vezes nas páginas 28 e 34.

MÜLLNER, D. Modern hierarchical, agglomerative clustering algorithms. n. 1973, p. 29, 2011. Disponível em: <<http://arxiv.org/abs/1109.2378>>. Citado na página 131.

MÜLLNER, D. Fastcluster: Fast Hierarchical, Agglomerative Clustering Routines for R and Python. *Journal of Statistical Software*, v. 53, n. 9, p. 1–18, 2013. Disponível em: <<http://www.jstatsoft.org/v53/i09/>>. Citado na página 131.

MURTAGH, F.; LEGENDRE, P. Ward’s Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward’s Criterion? *Journal of Classification*, v. 31, p. 274–295, 2014. Citado na página 131.

NELSON, C. C. *Literature of the women’s suffrage campaign in England*. [S.l.]: Broardview Press, 2002. Citado na página 29.

OLIVEIRA, A. G. de. *Efeitos das composições familiares na mobilidade dos idosos – uma análise multinível*. 128 p. Tese (Mestrado) — Universidade de Brasília, 2014. Citado na página 83.

ORTÚZAR, J. d. D.; WILLUMSEN, L. G. *Modelling transport*. New York: John Wiley and Sons, 1994. Citado 2 vezes nas páginas 55 e 83.

PEIXOTO, N. M. O. *A evolução Temporal da Mobilidade da População na Região Metropolitana de Porto Alegre, entre 1986 e 1997*. 160 p. Tese (Mestrado) — Universidade Federal do Rio Grande do Sul, 2002. Citado na página 83.

PFEIFFER, L. M.; STRAMBI, O. Análise e modelagem da evolução temporal da posse de autos na Região Metropolitana de São Paulo. *Revista Transportes*, XIII, n. 1, p. 21–29, 2005. Citado na página 96.

PINSKY, C. B.; PEDRO, J. M. Mulheres – Igualdade e Especificidade. In: PINSKY, J.; PINSKY, C. B. (Ed.). *História da Cidadania*. 2ª. ed. São Paulo: Contexto, 2003. Citado na página 33.

PINTO, L. P. Mulheres brasileiras na mídia portuguesa. *Cadernos Pagu*, v. 23, p. 229–257, 2004. Disponível em: <<http://www.scielo.br/pdf/cpa/n23/n23a08.pdf>>. Citado na página 32.

PISCITELLI, A. Gênero: a história de um conceito. In: ALMEIDA, H. B. de; SZWAKO, J. (Ed.). *Diferenças, Igualdade*. São Paulo: Berlendis & Vertecchia, 2009. cap. 4, p. 239. Citado 4 vezes nas páginas 29, 30, 31 e 33.

POLK, M. Are women potentially more accommodating than men to a sustainable transportation system in Sweden? *Transportation Research Part D: Transport and Environment*, v. 8, p. 75–95, 2003. Disponível em: <[R Core Team. *R: A Language and Environment for Statistical Computing*. Viena: R Foundation for Statistical Computing, 2011. Citado na página 131.](http://ac.els-cdn.com/S1361920902000342/1-s2.0-S1361920902000342-main.pdf?{|_}tid=74a6c252-78c6-11e4-a00c-00000aacb362{&}acdnat=1417375628{|_}1084fe2f2f94333a3b7d59a>}. Citado 3 vezes nas páginas 23, 48 e 49.</p></div><div data-bbox=)

- RAJU, S. Gender and empowerment: Creating “thus far and no further” supportive structures. A case from India. In: NELSON, L.; SEAGER, J. (Ed.). *A Companion to Feminist Geography*. Oxford: Blackwell Publishing, 2005. cap. 14. Disponível em: <<http://onlinelibrary.wiley.com/doi/10.1002/9780470996898.ch14/pdf>>. Citado 2 vezes nas páginas 23 e 45.
- RAMALHO, J. J. D. S. *Modelos De Regressão Para Dados De Contagem*. 137 p. Tese (Dissertação de Mestrado) — Universidade Técnica de Lisboa, 1996. Citado na página 160.
- RAY, P. O. Woman Suffrage in Foreign Countries. *American Political Science Review*, v. 12, n. 3, p. 469–474, 1918. Citado na página 29.
- RICHARDSON, B. C. Sustainable transport: analysis frameworks. *Journal of Transport Geography*, v. 13, n. 1, p. 29–39, mar 2005. ISSN 09666923. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0966692304000857>>. Citado na página 23.
- ROOT, A.; SCHINTLER, L. Women, motorization and the environment. *Transportation Research Part D: Transport and Environment*, v. 4, n. 5, p. 353–355, sep 1999. ISSN 13619209. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S1361920999000127>>. Citado 5 vezes nas páginas 23, 46, 49, 51 e 63.
- ROSENBLoom, S. Editorial: The Need for Study of Women’s Travel Issues Recently. *Transportation*, v. 7, n. 1978, p. 347–350, 1978. Citado 3 vezes nas páginas 23, 44 e 45.
- ROSENBLoom, S. Trends in women’s travel patterns. In: *Women’s Travel Issues Second National Conference*. [s.n.], 2000. p. 16–34. Disponível em: <<http://trid.trb.org/view.aspx?id=720092>>. Citado na página 48.
- ROSENBLoom, S. The mobility needs of older Americans. *Taking the High Road: A Transportation Agenda of Strengthening Metropolitan Areas*, Washington DC, p. 227–54, 2003. Disponível em: <http://www.brookings.edu/\~/media/research/files/reports/2003/7/transportationrosenbloom/20030807__rosenbloom>. Citado na página 48.
- ROSENBLoom, S. Understanding Women’s and Men’s Travel Patterns: The Research Challenge. In: TRANSPORTATION RESEARCH BOARD OF THE NATIONAL ACADEMIES. *Research on Women’s Issues in Transportation: Volume 1 - Conference Overview and Plenary Papers*. Washington DC: National Research Council, 2006. p. 7–28. ISBN 0309099560. Disponível em: <<http://onlinepubs.trb.org/Onlinepubs/conf/CP35v1.pdf>>. Citado na página 23.
- RUBIN, G. S. The Traffic in Women: Notes on the ‘Political Economy’ of Sex. In: REITER, R. (Ed.). *Toward an Anthropology of Women*. Nova Iorque: Monthly Review Press, 1975. p. 157–211. Disponível em: <<http://summermeetings2013.files.wordpress.com/2013/04/rubin-traffic.pdf>>. Citado 2 vezes nas páginas 30 e 31.
- RUEDA, S. et al. *Libro verde de medio ambiente urbano – Tomo I*. Barcelona, 2007. 473 p. Disponível em: <<http://bcnecologia.net/es/proyectos/libro-verde-de-medio-ambiente-urbano-tomo-i-y-ii>>. Citado na página 23.
- RYAN, J. M.; HAN, G. Vehicle-Ownership Model Using Family Structure and Accessibility Application to Honolulu, Hawaii. *Transportation Research Record*, v. 1676, n. 99, p. 1–10, 1999. ISSN 0361-1981. Citado na página 96.

SÃO PAULO (ESTADO). *Lei nº 1.139, de 16 de junho de 2011. Reorganiza a Região Metropolitana da Grande São Paulo, cria o respectivo Conselho de Desenvolvimento e dá providências correlatas.* 2011. Citado na página 27.

SAFFIOTI, H. I. B. *A Mulher na Sociedade Classes: Mito e Realidade.* 3^a. ed. São Paulo: Expressão Popular, 2013. 528 p. Citado na página 32.

SCHWANEN, T.; DIJST, M.; DIELEMAN, F. M. A microlevel analysis of residential context and travel time. *Environment and Planning A*, v. 34, n. 8, p. 1487–1507, 2002. ISSN 0308-518X. Disponível em: <<http://www.envplan.com/abstract.cgi?id=a34159>>. Citado 4 vezes nas páginas 23, 49, 50 e 63.

SCOTT, J. W. Gender: A Useful Category of Historical Analysis. *The American Historical Review*, IE/UFRGS, Porto Alegre, v. 91, n. 5, p. 1053–1075, 1986. Disponível em: <<http://www.jstor.org/stable/1864376>>. Citado 2 vezes nas páginas 32 e 44.

SHARMA, S. *Applied Multivariate Techniques.* New York: John Wiley, 1996. 493 p. Citado na página 134.

SHEARMUR, R. Travel from home: An economic geography of commuting distances in Montreal. *Urban Geography*, v. 27, n. 4, p. 330–59, 2006. Citado na página 60.

SILVEY, R.; ELMHIRST, R. Engendering Social Capital: Women Workers and Rural–Urban Networks in Indonesia’s Crisis. *World Development*, v. 31, n. 5, p. 865–879, may 2003. ISSN 0305750X. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0305750X03000135>>. Citado 2 vezes nas páginas 23 e 45.

SILVEY, R. M. Stigmatized spaces: gender and mobility under crisis in South Sulawesi, Indonesia. *Gender, Place & Culture*, v. 7, n. 2, p. 143–61, 2000. Citado na página 37.

SOARES, V.; PINHEIRO, L. S. *Dados das desigualdades: gênero e raça.* [S.l.], 2003. 15 p. Disponível em: <http://www.planalto.gov.br/seppir/pesquisas{_\}indicadores/raca/presskit/unifem>. Citado na página 36.

STEG, L. Car use: lust and must. Instrumental, symbolic and affective motives for car use. *Transportation Research Part A: Policy and Practice*, v. 39, n. 2-3, p. 147–162, feb 2005. ISSN 09658564. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0965856404001016>>. Citado na página 23.

STELLA, P. R. Palavra. In: BRAIT, B. (Ed.). *Bakthin - Conceitos-Chave.* São Paulo: Contexto, 2005. cap. 10. Citado na página 28.

STOLKE, V. La mujer es puro cuento: la cultura del género. *Revista Estudos Feministas*, v. 12, n. 2, p. 77–105, 2004. Citado na página 30.

STOLLER, R. J. *Sex and Gender.* Londres: Karnac, 1984. 400 p. Citado na página 30.

STRAMBI, O.; BILT, K.-A. van de. Trip Generation Modeling Using CHAID, a Criterion-Based Segmentation Modeling Tool. *Transportation Research Record*, v. 1645, n. 98, p. 24–31, 1998. Citado 2 vezes nas páginas 50 e 55.

STRAMBI, O.; Van de Bilt, K. Mobility in São Paulo: a temporal perspective. *IRF ROAD WORLD CONGRESS*, 14, Paris, 2001. Citado na página 83.

- TABAK, F. *A mulher brasileira no Congresso Nacional*. 1^a. ed. Brasília: Câmara dos Deputados, Centro de Documentação e Informação, Coordenação de Publicações, 1989. Citado na página 33.
- TADANO, Y. D. S.; UGAYA, C. M. L.; FRANCO, A. T. Método de regressão de Poisson: metodologia para avaliação do impacto da poluição atmosférica na saúde populacional. *Ambiente & sociedade*, v. 12, n. 2, p. 241–255, 2009. ISSN 1414-753X. Citado na página 160.
- TERTOOLEN, G.; KREVELD, D. V.; VERSTRATEN, B. Psychological resistance against attempts to reduce private car use. *Transportation Research Part A: Policy and Practice*, v. 32, n. 3, p. 171–181, 1998. Citado na página 23.
- TREMBLAY, K. Academic Mobility and Immigration. *Journal of Studies in International Education*, v. 9, n. 3, p. 196–228, sep 2005. ISSN 1028-3153. Disponível em: <<http://jsi.sagepub.com/cgi/doi/10.1177/1028315305277618>>. Citado na página 37.
- URRY, J. Connections. *Environment and Planning D*, v. 22, p. 143–61, 2004. Citado 2 vezes nas páginas 41 e 42.
- Van de Bilt, K.-A. *Análise de taxas de produção de viagens urbanas utilizando modelagem de segmentação*. 131 p. Tese (Mestrado) — Universidade de São Paulo, 1997. Citado 2 vezes nas páginas 100 e 103.
- VANCE, C.; IOVANNA, R. Gender and the Automobile: Analysis of Nonwork Service Trips. *Transportation Research Record: Journal of the Transportation Research Board*, v. 2013, p. 54–61, dec 2007. ISSN 0361-1981. Disponível em: <<http://trb.metapress.com/openurl.asp?genre=article&id=doi:10.3141/2013>>. Citado 3 vezes nas páginas 46, 47 e 63.
- VASCONCELLOS, E. *Transporte Urbano, Espaço e Equidade: Análise das Políticas Públicas*. 1^a. ed. São Paulo: Annablume, 2001. 218 p. Citado 8 vezes nas páginas 23, 39, 41, 42, 50, 51, 63 e 83.
- VASCONCELLOS, E. *Mobilidade Urbana e Cidadania*. Rio de Janeiro: Senac, 2012. 213 p. ISBN 978-85-7458-318-1. Citado 4 vezes nas páginas 23, 38, 49 e 95.
- VESPUCCI, K. M. *Sequências de atividades e cadeias de viagens na Região Metropolitana de São Paulo – uma investigação comparativa do período 1987-1997*. 191 p. Tese (Mestrado) — Universidade de São Pauçlo, 2003. Citado 3 vezes nas páginas 76, 78 e 142.
- Ward Jr, J. H. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American statistical association*, v. 58, n. 301, p. 236–244, 1963. Disponível em: <<http://www.tandfonline.com/doi/abs/10.1080/01621459.1963.10500845>>. Citado na página 135.
- WILLARD, F. E. *Wheel Within a Wheel: How I Learned to Ride a Bicycle, With Some Reflections by The Way*. Chicago: Fleming H. Revell Company, 1895. 75 p. Disponível em: <<https://archive.org/details/wheelwithinwheel00williala>>. Citado 2 vezes nas páginas 43 e 44.

Anexos

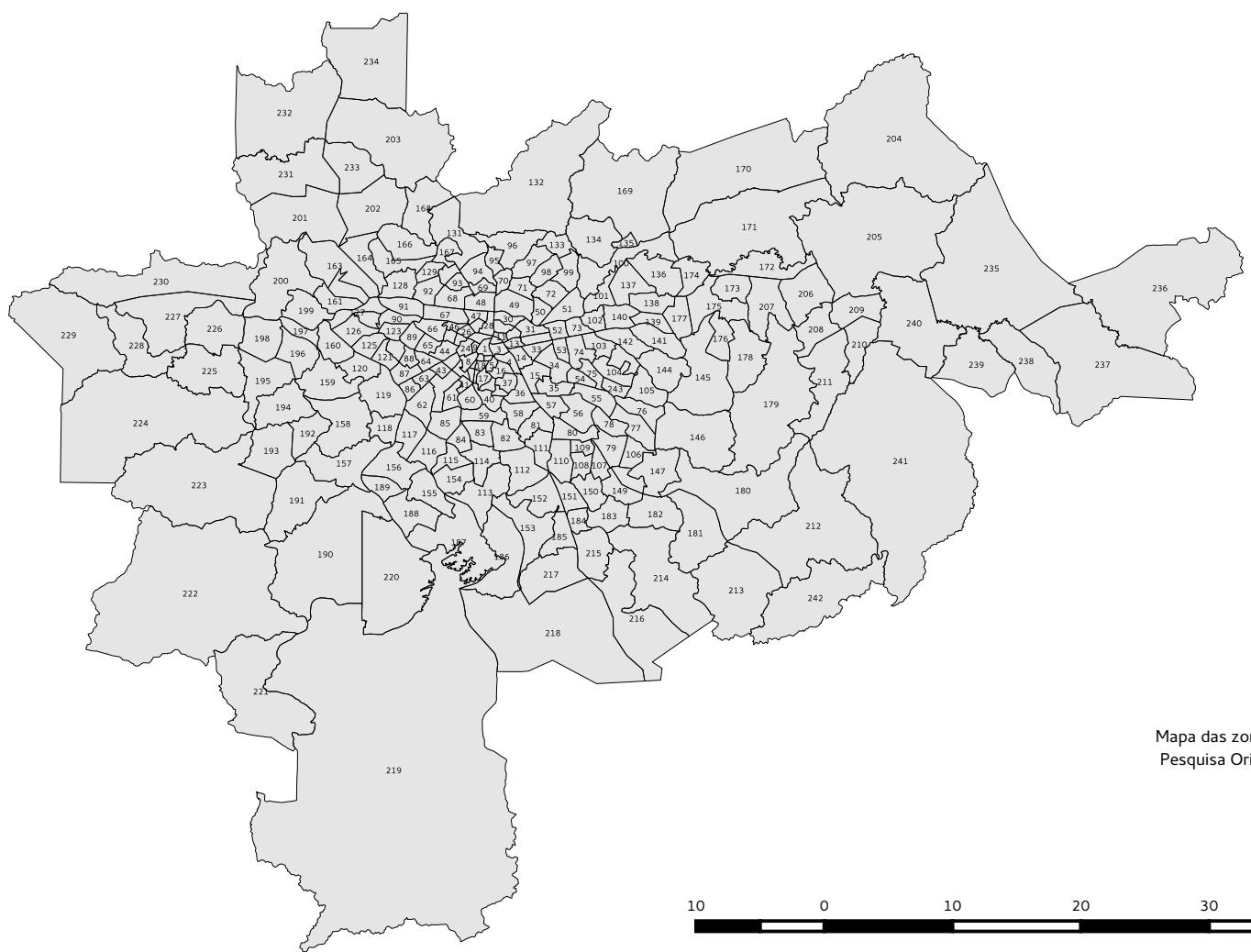
ANEXO A – Correspondência entre Zonas
das Pesquisas Origem Destino por meio das
Unidades de Correspondência entre Zonas
(UCOD)

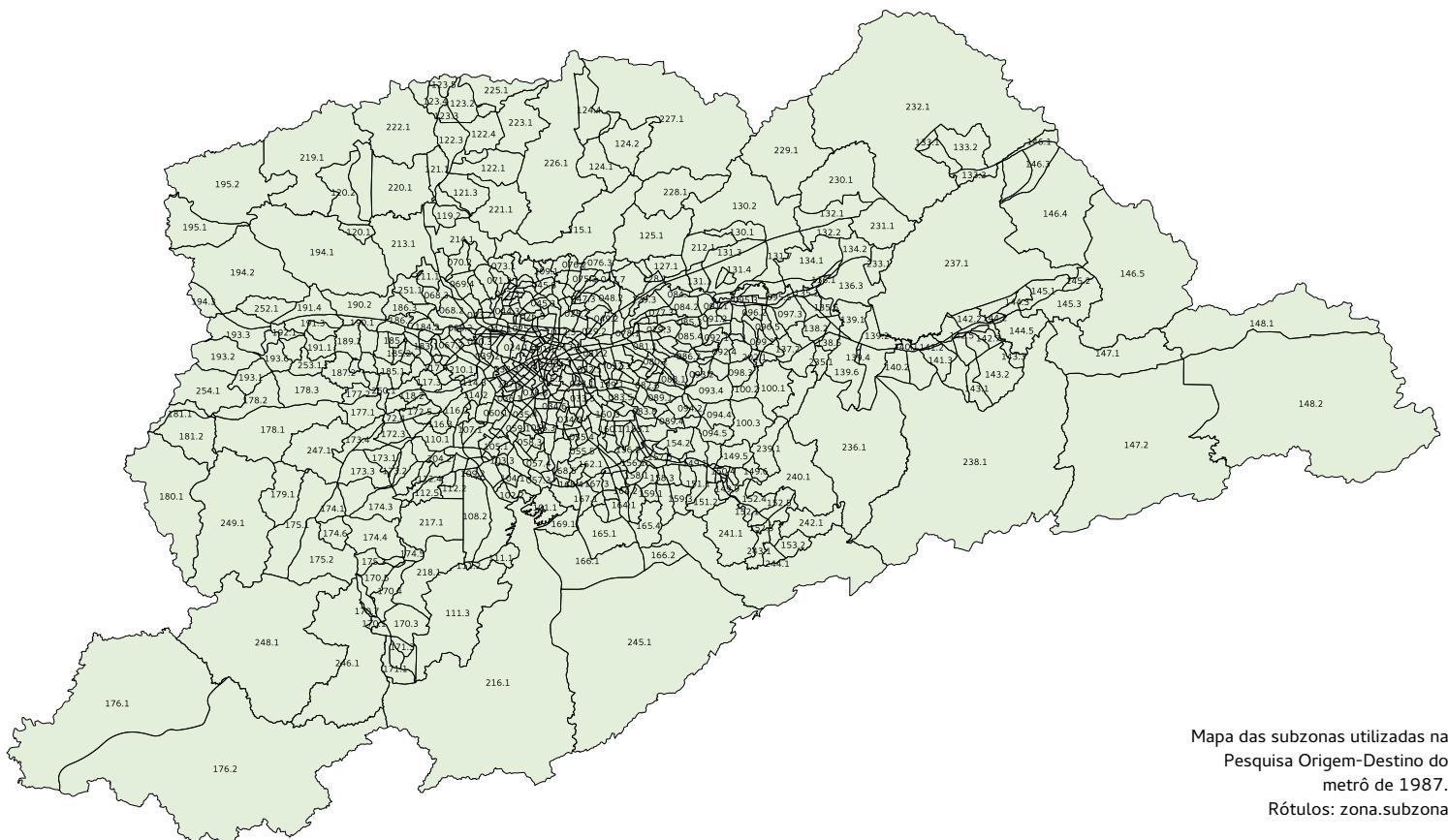
UCOD	Nome	Zoneamento da OD 1977												Zoneamento da OD 1987										
		1	6	7	9	10	24	5	8	17	18	19	20	21	22	23	39	1	2	3	4	5	6	7
1	Sé/República							44	65	66	89							9	10					
2	Liberdade/Bela Vista							25	26	27	28	45	46	47	67			11	12	13	14	15	16	17
3	Consolação/Perdizes							2	3	11	12	13	29	30	31	32		18	19	20	21			
4	Barra Funda/Bom Retiro/Santa Cecília							14	15	33	34	35	54					22	23	24	25	26	27	
5	Brás/Pari/Belém							4	16	36	37	58	81					28	29	30				
6	Mooca/Águia Rasa							38	40	41	60	61						31	32	33				
7	Cambuci/Ipiranga							59	83									34						
8	Vila Mariana							10	85	115	116							35	36	37				
9	Saúde							11	156	189								38	39					
10	Moema/Campo Belo							12	62	117								40	41	42	43	44		
11	Santo Amaro							13	42	43	63	64						45	46	47				
12	Itaim Bibi							14	86	87	88	121	122	125				48	49	50	51			
13	Jardim Paulista							15	90	91	123	124	126	127				52	53	54	55	56	57	
14	Pinheiros/Alto de Pinheiros							16	161	162	163	164						58	59					
15	Lapa/Leopoldina							17	92	128	129	165						60	61	62				
16	Pirituba/Jaguara/São Domingos							18	166	168								63	64					
17	Freguesia do Ó/Limão							19	48	68	69	70	93	94	95	130		65	66	67	68	69		
18	Brasilândia							20	131	167								70	71	72	73			
19	Santana/Casa Verde							21	96	97	132							74	75	76	77			
20	Mandaqui/Cachoeirinha							22	98	99	133							78						
21	Tremembé/Tucuruvi							23	49	50	51	71	72					79	80	81	82			
22	Jacanã/Vila Medeiros							24	52	53	73	74	103	143				83	84	85				
23	Vila Maria/Vila Guilherme							25	75	104	105	243						86	87					
24	Tatuapé/Carrão							26	55	56	57	78						88						
25	Vila Formosa/Aricanduva							27	76	77								89						
26	Vila Prudente/São Lucas							28	101	136	137	174						90	91					
29	Penha/Ponte Rasa							29	102	138	139	140	177					92	93	94				
30	Vila Matilde/Artur Alvim							31	141	142								95	96					
31	Cidade Líder/Parque do Carmo							32	144	145								97	98					
32	São Mateus/Guatemi/São Rafael							33	146									99						
33	Vila Jacuí/São Miguel Paulista							34	173	175							100	101						
34	Itaquera/José Bonifácio							35	176	178							102	103						
35	Jardim Helena/Vila Curuçá/Itaim Paulista							36	172	206	207						104	105	106					
36	Lajeado/Guaianases/Cidade Tiradentes							37	179								107	108	109					
37	Cursino/Sacomã							38	82	111	112						110	111						
38	Jabaquara							39	113	114	154						112	113						
39	Campo Grande/Cidade Ademar/Pedreira							40	155	187	188						114	115	116	117	118			
40	Socorro/Cidade Dutra							41	220								119	120						
41	Jd. São Luís/Capão Redondo/Jd. Ângela							42	190	191							121	122	123					
42	Campo Limpo/Vila Andrade							43	157								124	125	126					
43	Butantã/Morumbi/Vila Sônia							44	118	119	120	158					127	128	129	130				
44	Jacuara/Rio Pequeno/Raposo Tavares							45	159	160	194						131	132	133					
45	Taboão da Serra							46	192	193							134							
46	Osasco(centro)							47	196	197							135	136						
47	Novo Osasco							48	195	198							137	138						
48	Muttinga/Presidente Altino							49	199	200							139	140	141					
49	Carapicuíba							50	225	226							142	143	144					
50	Guarulhos(centro)							51	100	134	135						145	146	147					
51	F. de Vasconcelos/Poá/Itaquaquecetuba/Suzano							52	205	208	209	210	211	240			148	149	150	151	152	153	154	155
52	Mogi das Cruzes(centro)							53	237	238	239						157	158	159	160				
53	São Caetano do Sul							54	80	108	109	110					161	162						
54	Santo André(centro)							55	79	106	107	147	148	149	150		163	164	165	166				
55	Vila Pires/Pedroso							56	182	183	214						167	168						
56	Mauá							57	180	181							169	170	171	172				
57	Rudge Ramos							58	151	152							173							
58	São Bernardo do Campo(centro)							59	184	185	215	217					174	175	176					
59	Diadema							60	153	186							177	178	179					
60	Grajaú/Parelheiros/Marsilac							61	219								180	181	182					
61	Jaraguá/Perus/Anhangüera							62	201	202	231	233					183	184	185	186	187			
62	Macro-zona Norte (1)							63	203	232	234						188	189	190	191	192	193	194	
63	Macro-zona Nordeste (2)							64	169	170	171	204					195	196	197	198	199	200	201	
64	Macro-zona Leste (3)							65	235	236	241						202	203	204	205	206	207	208	
65	Macro-zona Sudeste (4)							66	212	213	216	218	242					214	215	216	217	218	219	220
66	Macro_zona Sudoeste (5)							67	221	222	223						222	223	224	225	226	227	228	
67	Macro_zona Oeste (6)								224	227	228	229	230					231	232	233	234	235	236	237
																	240	241	242	243	244	245	246	
																	247	248	249	250	251	252	253	
																	254							

UCOD	Nome	Zoneamento da OD 1997					
		1	2	3	4	5	6
1	Sé/República	11	12	13	14	15	16
2	Liberdade/Bela Vista	17	36	37	74	76	
3	Consolação/Perdizes	7	18	19	38	39	40
4	Barra Funda/Bom Retiro/Santa Cecília	8	20	21	22	23	24
5	Brás/Pari/Belém	25	26	27	53	55	
6	Mococa/Águia Rasa	9	10	28	29	58	59
7	Cambuci/Ipiranga	30	31	32	62	64	
8	Vila Mariana	63	65	109	111		
9	Saúde	66	67	112	114	116	117
10	Moema/Campo Belo	190	191	192	193	118	119
11	Santo Amaro	69	121	122	123	124	
12	Itaim Bibi	33	34	35	68		
13	Jardim Paulista	70	71	72	73	75	130
14	Pinheiros/Alto de Pinheiros	77	78	79	132	133	
15	Lapa/Leopoldina	136	137	138	139	140	
16	Pirituba/Jaguara/São Domingos	80	81	141	142	143	
17	Freguesia do O/Limão	144	145	210	211		
18	Brasilândia	42	43	44	45	82	83
19	Santana/Casa Verde	146	147	148	149	212	213
20	Mandaqui/Cachoeirinha	86	150	151	152	153	155
21	Tremembé/Tucuruvi	144	145	210	211	214	215
22	Jaçanã/Vila Medeiros	87	88	89	154	156	
23	Vila Maria/Vila Guilherme	46	47	48	49	50	
24	Tatuapé/Carrão	51	52	95	96		
25	Vila Formosa/Aricanduva	54	97	169	171		
26	Vila Prudente/São Lucas	56	57	98	99	100	175
27	Sapopemba	173	174	176	235		
28	Cangaíba/Ermelino Matarazzo	92	159	160	161	218	
29	Penha/Ponte Rasa	91	93	162	163	164	
30	Vila Matilde/Artur Alvim	94	165	166	167		
31	Cidade Líder/Parque do Carmo	168	170	228	229	230	231
32	São Mateus/Iguatemi/São Rafael	172	232	233	234	236	294
33	Vila Jacuí/São Miguel Paulista	219	220	221	222	225	
34	Itaquera/José Bonifácio	223	224	226	227	289	292
35	Jardim Helena/Vila Curuçá/Itaim Paulista	279	280	281	282	283	284
36	Lajeado/Guaianases/Cidade Tiradentes	286	287	290	291	293	
37	Cursino/Sacomã	60	61	103	104	106	107
38	Jabaquara	110	113	115	182		
39	Campo Grande/Cidade Ademar/Pedreira	183	184	185	186	187	188
40	Socorro/Cidade Dutra	246	247	248	249	250	303
41	Jd. São Luís/Capão Redondo/Jd. Ângela	251	252	253	254	255	304
42	Campo Limpo/Vila Andrade	194	195	196	256	257	258
43	Butantã/Morumbi/Vila Sônia	125	126	127	128	129	131
44	Jacuare/Rio Pequeno/Raposo Tavares	134	135	201	202	262	263
45	Taboão da Serra	260	261				
46	Osasco(centro)	203	204				
47	Novo Osasco	265	266				
48	Mutinga/Presidente Altino	205	206	207			
49	Carapicuíba	267	313				
50	Guarulhos(centro)	90	157	158			
51	F. de Vasconcelos/Poá/Itaquaquecetuba/Suzano	288	329	331	332	334	335
52	Mogi das Cruzes(centro)	339	371	372			
53	São Caetano do Sul	102	105	178			
54	Santo André(centro)	177	179	237			
55	Vila Pires/Pedroso	239	298				
56	Mauá	238	297	342			
57	Rudge Ramos	180					
58	São Bernardo do Campo(centro)	240	241	299			
59	Diadema	181	242	243			
60	Grajaú/Parelheiros/Marsilac	301	302	347	348	349	382
61	Jaraguá/Perus/Anhanguera	208	209	270	271	272	
62	Macro-zona Norte (1)	273	274	322	323	324	325
63	Macro-zona Nordeste (2)	216	217	275	276	277	278
64	Macro-zona Leste (3)	333	341	369	370	373	374
65	Macro-zona Sudeste (4)	300	343	344	345	346	378
66	Macro_zona Sudoeste (5)	309	310	311	350	351	352
67	Macro_zona Oeste (6)	268	269	312	314	315	316

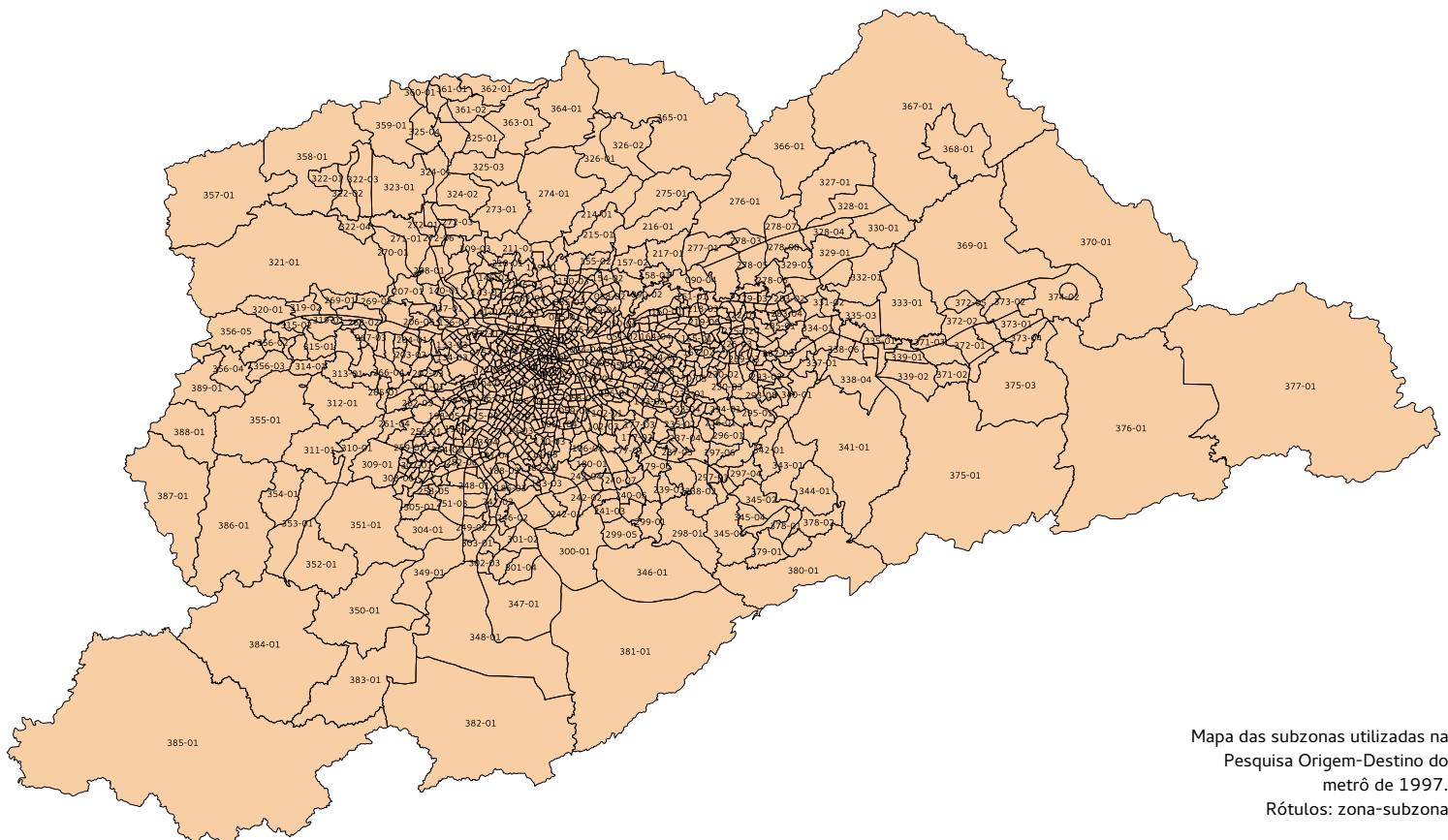
UCOD	Nome	Zoneamento da OD 2007																												
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
1	Sé/República	20	21	22	23	24	25	26	27	28	29																			
2	Liberdade/Bela Vista	30	31	32	33	34	83	84	85	86	87	88	89																	
3	Consolação/Perdizes	7	8	9	35	36	37	90	91	92	93																			
4	Barra Funda/Bom Retiro/Santa Cecília	10	11	12	13	14	15	16	17	38	39	40	41	42																
5	Brás/Pari/Belém	43	44	45	46	47	48																							
6	Mooca/Água Rasa	18	19	222	223	224	225	226	227	228																				
7	Cambuci/Ipiranga	49	50	51	52	53	54	55	56	57																				
8	Vila Mariana	58	59	60	61																									
9	Saúde																													
10	Moema/Campo Belo	62	63	64	65	66	251	252	253	254	255																			
11	Santo Amaro	281	282	283	284																									
12	Itaim Bibi	67	68	69	70	71	72																							
13	Jardim Paulista	73	74	75	76	77	78																							
14	Pinheiros/Alto de Pinheiros	79	80	81	82	94	95	96																						
15	Lapa/Leopoldina	97	98	99	100	101	102	103	104	105	106	107																		
16	Pirituba/Jaguara/São Domingos	108	109	110	111	112																								
17	Freguesia do Ó/Limão	123	124	125	136	137																								
18	Brasilândia	119	120	121	122																									
19	Santana/Casa Verde	126	127	128	129	130	131	132	133	134	135																			
20	Mandaqui/Cachoeirinha	138	139	140	141	142	143																							
21	Tremembé/Tucuruvi	144	145	146	147	148	149	150	151																					
22	Jaçanã/Vila Medeiros	152	153	154	155	156																								
23	Vila Maria/Vila Guilherme	157	158	159	160	161	162																							
24	Tatuapé/Carrão	163	164	165	166	197	198																							
25	Vila Formosa/Aricanduva	199	200	201	202																									
26	Vila Prudente/São Lucas	237	238	239	240	241	242																							
27	Sapopemba	243	244	245	246																									
28	Cangaíba/Ermelino Matarazzo	170	171	172	173	174	175	176																						
29	Penha/Ponte Rasa	167	168	169	177	178																								
30	Vila Matilde/Artur Alvim	203	204	205	206																									
31	Cidade Líder/Parque do Carmo	207	208	209	210	211	212																							
32	São Mateus/Iguatemi/São Rafael	218	219	220	221	247	248	249	250																					
33	Vila Jacuí/São Miguel Paulista	179	180	185	186	187																								
34	Itaquera/José Bonifácio	181	182	183	184	213	214																							
35	Jardim Helena/Vila Curuçá/Itaim Paulista	188	189	190	191	194	195	196																						
36	Lajeado/Guaianases/Cidade Tiradentes	192	193	215	216	217																								
37	Cursino/Sacomã	229	230	231	232	233	234	235	236																					
38	Jabotacaba	256	257	258	259																									
39	Campo Grande/Cidade Ademar/Pedreira	260	261	262	263	264	265	266	267	268																				
40	Socorro/Cidade Dutra	269	270	271	272	273	274																							
41	Jd. São Luís/Capão Redondo/Jd. Ângela	285	286	287	288	289	290	291	292	293	294																			
42	Campo Limpo/Vila Andrade	299	300	301	302	303	304	305																						
43	Butantã/Morumbi/Vila Sônia	295	296	297	298	306	307	308	309	317	318	319	320																	
44	Jaquaré/Rio Pequeno/Raposo Tavares	310	311	312	313	314	315	316																						
45	Taboão da Serra	411	412																											
46	Osasco (centro)	436	437	438	439																									
47	Novo Osasco	440	441	442																										
48	Mutings/Presidente Altino	443	444	445	446	447																								
49	Carapicuíba	432	433	434	435																									
50	Guarulhos (centro)	336	337	338	339	340	341																							
51	F. de Vasconcelos/Poá/Itaquaquecetuba/Suzano	354	355	356	357	358	359	360	361	362	363																			
52	Mogi das Cruzes (centro)	366	367	368	369	370																								
53	São Caetano do Sul	379	380	381	382																									
54	Santo André (centro)	383	384	385	386	387	388																							
55	Vila Pires/Pedroso	389	390																											
56	Mauá	392	393	394	395																									
57	Rudge Ramos	402																												
58	São Bernardo do Campo (centro)	401	403	404																										
59	Diadema	408	409	410																										
60	Grajaú/Parelheiros/Marsilac	275	276	277	278	279	280																							
61	Jaraguá/Perus/Anhanguera	113	114	115	116	117	118																							
62	Macro-zona Norte (1)	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335														
63	Macro-zona Nordeste (2)	342	343	344	345	346	347	348	349	350	351	352	353																	
64	Macro-zona Leste (3)	364	365	371	372	373	374	375	376	377	378																			
65	Macro-zona Sudeste (4)	391	396	397	398	399	400	405	406	407																				
66	Macro_zona Sudoeste (5)	413	414	415	416	417</																								

ANEXO B – Mapas de Zonas das Pesquisas
Origem Destino 1977 e de Subzonas de 1987 e
1997





10 0 10 20 30 40 km



10 0 10 20 30 40 km

ANEXO C – Layouts dos bancos de dados
das Pesquisas Origem-Destino do Metrô-SP
(1977, 1987, 1997 e 2007)

LAYOUT PESQUISA ORIGEM DESTINO 1977

Metrô - SP

Variável	Conteúdo	Início	Fim	Compr	Códigos
1	zona	Zona do Domicílio	1	3	3
2	subzona	Subzona do Domicílio	4	6	3
3	munires	Município do Domicílio	7	12	6
4	nota		13	13	1
5	conglome	Conglomerado	14	17	4
6	bolsao	Bolsão	18	18	1
7	setores	Setores	13	14	2
8	id_dom	Identifica Domicílio	15	34	20
9	f_dom	Identifica Primeiro Registro do Domicílio	35	35	1
					0 - Registros Complementares do Dom.
					1- Primeiro registro do domicílio
10	fe_dom	Fator de Expansão do Domicílio	36	43	8
11	domicil	Número do Domicílio	44	45	2
12	tipo_dom	Tipo de Domicílio	46	46	1
					1 - particular
					2 - Coletivo
13	pess_fam	Pessoa Residente da Família	47	48	2
14	não_res	Pessoas Não Residentes	49	50	2
15	condmora	Condição de Moradia	51	51	1
					1 - Própria paga
					2 - Própria em pagamento
					3 - Alugada
					4 - Cedida
					5 - Outro
16	aluguel	Prestação ou Aluguel	52	56	5
17	f_fam	Identifica Primeiro Registro da Família	57	57	1
					0 - Demais registros
					1- Primeiro registro da família
18	fe_fam	Fator de Expansão da Família	58	66	9
19	no_fam	Número da Família	67	68	2
20	tot_fam	Total de Famílias	69	70	2
21	qt_auto	Número de Veículos	71	71	1
22	codrendf	Código de renda familiar	72	72	1
					0 - Respondeu
					1 - Não sabe
					2 - Não respondeu
					3 - Não se aplica
23	redfam	Renda Familiar Mensal	73	77	5
24	f_pess	Identifica Primeiro Registro da Pessoa	78	78	1
					0 - Demais registros
					1- Primeiro registro da pessoa
25	fe_pess	Fator de Expansão da Pessoa	79	87	9
26	pessoa	Número da Pessoa	88	89	2
27	sit_fam	Situação Familiar	90	90	1
					1 - Chefe
					2 - Cônjugue
					3 - Filho(a)
					4 - Parente
					5 - Agregado
					6 - Empregado Residente
					7 - Visitante não Res.
28	idade	Idade	91	92	2
29	sexo	Sexo	93	93	1
					1 - Masculino
					2 - Feminino
30	grau_ins	Grau de Escolariedade	94	94	1
					1 - Sem instrução
					2 - Primário Incompleto
					3 - Primário Completo
					4 - Ginásial Incompleto
					5 - Ginásial Completo
					6 - Colegial Incompleto
					7 - Colegial Completo
					8 - Universitário Incompleto
					9 - Universitário Completo

LAYOUT PESQUISA ORIGEM DESTINO 1977

Metrô - SP

Variável	Conteúdo	Início	Fim	Compr	Códigos
31	ocupacao	95	96	2	1 - Estudante 2 - Prendas Domésticas 3 -Aposentado 4 - Sem Ocupação (nunca trabalhou) 5 - Desempregado 6 -Em Licença 11 - Serviços não Especializados 12 - Serviços semi-especializados 13 - Escriturários 14 - Empregados de Comércio/Vendedor/Corretor/Outros 15 - Chefe/Supervisor de Operários Especializados 16 - Empregado de Escritório c/ Nível Supervisor 17 - Empregados em Empresas c/ Nível Gerente 18 - Empregados c/ Nível Universitário 21 - Operários sem Especialização 22 - Operários semi-especializados 23 - Operários Especializados Trabalhando p/ Terceiros 24 - Operários Especializados Trabalhando em Fábricas 25 - Chefe/Supervisor de Operários Especializados 26 - Trabalhadores Rurais 27 - Trabalhadores Rurais c/ Nível de Capataz 31- Prestadores de Serviços Autônomos 32- Prestadores de Serviços por Cta Própria 33- Operários Especializados Trabalhando por Cta Própria 34- Prof.Lib.Trab. por Cta Própria c/ Trabalho Intelect/Mental 35- Prof.Lib.Trab. por Cta Própria c/ Nível Altam. Especializado 36- Prof.Lib.Trab. por Cta Própria c/ Nível Universitário 41- Func.Púb.Trab.p/o Governo em Serv.Não Especializados 42- Func.Púb.Trab.p/o Governo em Serv.semi-especializados 43 - Escriturários e Aux.de Escritório Trab.p/o Governo 44- Func.Púb.Trab.p/o Governo com Nível de Supervisor 45- Func.Púb.Trab.p/o Governo com Nível de Diretor 46 - Func.Púb.Trab.p/o Governo com Nível Universitário 51 - Sócios ou Donos de Pequeno Comércio 52 - Téc.,Prof., Artesões, Artífices, Sócios/Donos Firma Pequena 53 - Industriais e Comerciantes, Sócios ou Donos Firma Pequena 54 - Téc.,Prof., Artesões, Artífices, Sócios/Donos Firma Grande/Média 55 - Industriais e Comerciantes, Sócios ou Donos Firma Média 56 - Industriais e Comerciantes, Sócios ou Donos Firma Grande 57 - Agricultores Arrendários de Fazendas, Hortas, Granjas, etc. 58 - Agricultores Proprietários de Fazendas 59 - Grandes Fazendeiros
32	ramoativ	Ramo de Atividade	97	98	2
					1 -Agrícola 2 - Constr.Civil 3 - Industrial 4 - Comercial 5 - Func.Público 6 - Serv.Transportes 7 - Empr.Serviço 8 - Serv.Autônomos 9 - Outro 10 -Não se Aplica

LAYOUT PESQUISA ORIGEM DESTINO 1977

Metrô - SP

Variável	Conteúdo	Início	Fim	Compr	Códigos
33	co_ren_i Código de Renda Individual	99	99	1	1 - Tem Renda 2 - Não Tem Renda 3 - Não Declarou
34	vl_ren_i Renda Individual Mensal	100	104	5	
35	zonaesc1 Zona da Escola	105	107	3	1 a 243
36	subesc1 Subzona da Escola	108	110	3	1 a 633
37	muniesc1 Município da Escola 1	111	116	6	1 a 27
38	zonaesc2 Zona da 2 Escola	117	119	3	1 a 243
39	subesc2 Subzona da 2 Escola	120	122	3	1 a 633
40	zonatra1 Zona do Primeiro Trabalho	117	119	3	1 a 243
41	subtra1 Subzona do Primeiro Trabalho	120	122	3	1 a 633
42	munitra1 Município do Primeiro Trabalho	123	128	6	1 a 27
43	zonatra2 Zona do Segundo Trabalho	129	131	3	1 a 243
44	subtra2 Subzona do Segundo Trabalho	132	134	3	1 a 633
45	munitra2 Município do Segundo Trabalho	135	140	6	1 a 27
46	id_viage Identificador do Registro de Viagem	141	141	1	0 - Não Tem Viagem 1 - Tem Viagem
47	fe_via Fator de Expansão com Aferição	142	150	9	9 dígitos 2 casas decimais
48	zonao Zona de Origem	151	153	3	1 a 243
49	subor Subzona de Origem	154	156	3	1 a 633
50	munorig Município de Origem	157	162	6	1 a 27
51	bolsaor	163	163	1	0 a 4
52	setoror	164	165	2	1 a 81
53	zonad Zona de Destino	166	168	3	1 a 243
54	subdes Subzona de Destino	169	171	3	1 a 633
55	munidest Município de Destino	172	177	6	1 a 27
56	bolsaode	178	178	1	0 a 4
57	setorde	179	180	2	1 a 81
58	motivo_o Motivo da Viagem na Origem	181	182	2	1 - Trabalho Indústria 2 - Trabalho Comércio 3 - Trabalho Serviços 4 - Escola/Educação 5 - Compras 6 - Negócios 7 - Médico/Dentista/Saúde 8 - Recreação/Visitas 9 - Servir Passageiro 10 - Residência
59	motivo_d Motivo da Viagem no Destino	183	184	2	idem ao anterior
60	modo1 Primeiro Modo	185	186	2	1 - Ônibus Trólebus 2 - Ônibus Escolar/Empresa 3 - Dirigindo Automóvel 4 - Passageiro de Automóvel 5 - Táxi 6 - Lotação/Perua 7 - Metrô 8 - Trem 9 - Motocicleta 10 - Bicicleta 11 - A Pé 12 - Outros

LAYOUT PESQUISA ORIGEM DESTINO 1977

Metrô - SP

Variável	Conteúdo	Início	Fim	Compr	Códigos	
61	modo2	Segundo Modo	187	188	2	idem ao anterior
62	modo3	Terceiro Modo	189	190	2	idem ao anterior
63	modoprin	Modo principal	191	192	2	idem ao anterior
64	tipo_vg	Tipo de Viagem	193	193	1	1 - Coletivo 2 - individual 3 - A pé
65	hsaida	Hora da Saída	191	192	2	
66	minsaida	Minutos da Saída	193	194	2	
67	anda_o	Tempo Andando na Origem	195	196	2	
68	h_cheg	Hora da Chegada	197	198	2	
69	mincheg	Minutos da Chegada	199	200	2	
70	anda_d	Tempo Andando até o Destino	201	202	2	
71	duracao	Duração da Viagem (em minutos)	203	205	3	
72	tipoestc	Tipo de Estacionamento	206	206	1	1 - Zona Azul/Parquímetro 2 - Estacionamento Avulso 3 - Estacionamento Mensal 4 - Estacionamento Próprio 5 - Meio Fio/Logradouro 6 - Estacionamento Patrocinado 7 - Não Estacionou
73	custoest	Custo do Estacionamento	207	210	4	
74	pqnaocar	Código de Não Utilização do Carro	211	211	1	1 -Não Disponível 2 - Estacionamento Caro 3 - Difícil de Estacionar 4 - Condução mais Barata 5 -Condução mais Conveniente 6 - Outros
75	numviag	Número da Viagem	212	213	2	
76	ordem		214	221	8	1 a 230.606

LAYOUT PESQUISA ORIGEM DESTINO 1987

Metrô - SP

Variável	Conteúdo	Início	Fim	Compr	Códigos
1 ZONA	Zona do Domicílio	1	3	3	1 a 204
2 SZ	Subzona do Domicílio	4	4	1	1 a 9
3 SZSEQ	Subzona Sequencial do Domicílio	5	8	4	1 a 1.012
4 MUNI_DOM	Município do Domicílio	9	10	2	1 a 38
5 FAIXA	Faixa de Consumo de Energia Elétrica	11	11	1	
6 ID_DOM	Identifica Domicílio	12	29	18	
7 F_DOM	Identifica Primeiro Registro do Domicílio	30	30	1	0 - Registros Complementares do Dom. 1- Primeiro registro do domicílio
8 FE_DOM	Fator de Expansão do Domicílio	31	36	6	6 dígitos 2 casas decimais
9 DOMICIL	Número do Domicílio	37	40	4	
10 DATA	Data da Entrevista	41	46	6	
11 TIPO_DOM	Tipo de Domicílio	47	47	1	1 - Individual 2 - Coletivo
12 FAM_DOM	Número de Famílias no Domicílio	48	49	2	
13 F_FAM	Identifica Primeiro Registro da Família	50	50	1	0 - Demais registros 1- Primeiro registro da família
14 FE_FAM	Fator de Expansão da Família	51	56	6	6 dígitos 2 casas decimais
15 FAMILIA	Número da Família	57	58	2	
16 CD_ENTRE	Código da Entrevista	59	59	1	1 - Recusa Total 2 - Moradores Ausentes 3 - Domicílio Vago 4 - Incompleta 5 - Completa sem viagem 6 - Completa com viagem
17 TP_RBAIR	Tempo de Residência no Bairro	60	61	2	
18 CONDMORA	Condição de Moradia	62	62	1	1 - Não se Aplica 2 - Não respondeu 3 - Alugada 4 - Casa Própria
19 ALUG_SM	Valor do Aluguel em Salários Mínimos	63	65	3	
20 VALUGUEL	Valor do Aluguel Ajustado - set/87	66	73	8	
21 INS_CHEF	Grau de Instrução do Chefe da Família	74	74	1	1 - Analfabeto/4ª Série Incompleta 2 - 4ª Série Completa 3 - 1º Grau Completo 4 - Coelgial Completo 5 - Supreior Completo
22 QT_TV	Quantidade de Televisores	75	75	1	
23 QT_RADIO	Quantidade de Rádios	76	76	1	
24 QT_BANHO	Quantidade de Banheiros	77	77	1	
25 QT_AUTO	Quantidade de Automóveis	78	78	1	
26 QT_EMPRE	Quantidade de Empregados Domésticos	79	79	1	
27 QT_ASPIR	Quantidade de Aspiradores de Pó	80	80	1	
28 QT_MLAVA	Quantidade de Máquinas de Lavar	81	81	1	
29 ABAABIPE	Classificação Socioeconômica ABA-ABIPEME	82	82	1	1 - A 2 - B 3 - C 4 - D 5 - E
30 PONTOABA	Ponto-Aba - Total de Todos os Critérios ABA	83	84	2	
31 RENDA_FA	Renda Familiar Ajustado - set/87	85	92	8	
32 CD_RENDA	Código de Renda Familiar	93	93	1	1 - Não Tem Renda 2 - Renda Familiar Incompleta 3 - Renda Familiar Completa
33 RENDATRI	Renda Familiar Atribuída	94	103	10	10 dígitos 2 casas decimais
34 CD_ATRI	Código de Renda Familiar Atribuída	104	104	1	1 - Não Tem Renda

2 - Renda Familiar Atribuída pelo Crit.ABA-ABIPEME

3 - Renda Familiar Pesquisada

35	TOT_PESS	Total de Pessoas na Família	104	105	2	
36	TOT_V_FA	Total de Viagens na Família	106	107	2	
37	F_PESS	Identifica Primeiro Registro da Pessoa	108	108	1	0 - Demais registros 1- Primeiro registro da pessoa
38	FE_PESS	Fator de Expansão da Pessoa	109	114	6	6 dígitos 2 casas decimais
39	PESSOA	Número da Pessoa	115	116	2	
40	SIT_FAMI	Situação Familiar	117	117	1	1 - Chefe 2 - Cônjuge 3 - Filho(a) 4 - Parente 5 - Agregado 6 - Empregado Residente 7 - Visitante
41	IDADE	Idade	118	119	2	(anos)
42	FXETOD	Faixa Etária (anos)	120	121	2	1 - até 3 2 - 4 a 6 3 - 7 a 10 4 - 11 a 14 5 - 15 a 17 6 - 18 a 22 7 - 23 a 29 8 - 30 a 39 9 - 40 a 49 10 - 50 a 59 11 - 60 e mais
43	SEXO	Sexo	122	122	1	1 - Masculino 2 - Feminino
44	ESTUDA	Estuda Atualmente ?	123	123	1	1 - Sim 2 - Não
45	GRAU_INS	Grau de Instrução	124	124	1	0 - Não Declarou 1 - Não alfabetizado/4ª Série Incompleta 2 - 4ª Série Completa 3 - 1º Grau Completo 4 - Colegial Completo 5 - Superior Completo
46	CLAS_ATI	Classe de Ativ.da Empresa que Trabalha	125	126	2	0 - Não Declarou 1 - Agrícola 2 - Construção Civil 3 - Indústria 4 - Comércio 5 - Funcionalismo Público 6 - Serviços de Transporte 7 - Empresa de Serviços 8 - Serviços Autônomo 9 - Outros 10 - Não se Aplica
47	SET_ATIV	Setor de Atividade Agregado	127	127	1	1 - Indústria 2 - Comércio 3 - Serviços 4 - Outros
48	OCUPACAO	Ocupação Principal	128	129	2	
49	CD_REN_I	Código de Renda Individual	130	130	1	1 - Não Tem Renda 2 - Não Declarou 3 - Declarou
50	VL_REN_I	Renda Individual em Salários Mínimos	131	132	2	2 dígitos 1 casas decimais
51	FX_REN_I	Faixa de Renda Individual	133	134	2	0 - Não Declarou 1 - até 1 SM

2 - 1-2 SM

3 - 2-3 SM

4 - 3-4 SM

5 - 4-5 SM

6 - 5-6 SM

7 - 6-7 SM

8 - 7-8 SM

9 - 8-9 SM

10 - 9-10 SM

11 - 10-12 SM

12 - 12-15 SM

13 - 15-20 SM

14 - 20-25 SM

15 - 25-30 SM

16 - 30-40 SM

17 - 40-50 SM

18 - mais de 50 SM

52	VREN_IND	Valor da Renda Individual Ajustado-set/87	135	140	6	
53	ZONAESC	Zona da Escola	141	143	3	1 a 254
54	SZESCOLA	Subzona da Escola	144	144	1	1 a 9
55	SZESEQ	Subzona Sequencial da Escola	145	148	4	1 a 1.012
56	MUNIESC	Município da Escola	149	150	2	1 a 38
57	ZONATRA1	Zona do Primeiro Trabalho	151	153	3	1 a 254
58	SZTRAB1	Subzona do Primeiro Trabalho	154	154	1	1 a 9
59	SZT1SEQ	Subzona Sequencial do Primeiro Trabalho	155	158	4	1 a 1.012
60	MUNITRA1	Município do Primeiro Trabalho	159	160	2	1 a 38
61	ZONATRA2	Zona do Segundo Trabalho	161	163	3	1 a 254
62	SZTRAB2	Subzona do Segundo Trabalho	164	164	1	1 a 9
63	SZT2SEQ	Subzona Sequencial do Segundo Trabalho	165	168	4	1 a 1.012
64	MUNITRA2	Município do Segundo Trabalho	169	170	2	1 a 38
65	VIG_PESS	Número de Viagem da Pessoa	171	172	2	
66	F_VIA	Identifica Primeiro Registro da Viagem	173	173	1	0 - Demais registros 1- Primeiro registro da pessoa
67	FE_VIA	Fator de Expansão da Viagem	174	179	6	6 dígitos 2 casas decimais
68	DIA_SEM	Dia da Semana	180	180	1	2 - Segunda-feira 3 - Terça-feira 4 - Quarta-feira 5 - Quinta-feira 6 - Sexta-feira
69	ZONA_O	Zona de Origem	181	183	3	1 a 254
70	SZO	Subzona de Origem	184	184	1	1 a 9
71	SZOSEQ	Subzona Sequencial de Origem	185	188	4	1 a 1.012
72	MUNIORIG	Município de Origem	189	190	2	1 a 38
73	ZONA_D	Zona de Destino	191	193	3	1 a 254
74	SZD	Subzona de Destino	194	194	1	1 a 9
75	SZDSEQ	Subzona Sequencial de Destino	195	198	4	1 a 1.012
76	MUNIDEST	Município de Destino	199	200	2	1 a 38
77	MOTIVO_O	Motivo na Origem	201	201	1	1 - Trabalho Indústria 2 - Trabalho Comércio 3 - Trabalho Serviços 4 - Escola/Educação 5 - Compras 6 - Negócios 7 - Médico/Dentista/Saúde 8 - Recreação/Visitas 9 - Residência
78	MOTIVO_D	Motivo no Destino	202	202	1	idem ao anterior
79	MOT_SRES	Motivo de Destino sem Residência	203	203	1	1 - Trabalho Indústria 2 - Trabalho Comércio

							3 - Trabalho/Serviços
							4 - Escola/Educação
							5 - Compras
							6 - Negócios
							7 - Médico/Dentista/Saúde
							8 - Recreação/Visitas
80	MODO1	Modo 1	204	205	2	1	1 - Ônibus Diesel
						2	2 - Trólebus
						3	3 - Ônibus Fretado
						4	4 - Escolar
						5	5 - Dirigindo Automóvel
						6	6 - Passageiro de Automóvel
						7	7 - Táxi
						8	8 - Lotação/Perua
						9	9 - Metrô
						10	10 - Trem
						11	11 - Moto
						12	12 - Bicicleta
						13	13 - A Pé
						14	14 - Caminhão
						15	15 - Outros
81	MODO2	Modo 2	206	207	2		idem ao anterior
82	MODO3	Modo 3	208	209	2		idem ao anterior
83	H_SAIDA	Hora da Saída	210	211	2		
84	MINSAIDA	Minutos da Saída	212	213	2		
85	ANDA_O	Tempo Andando desde a Origem	214	215	2		
86	H_CHEG	Hora da Chegada	216	217	2		
87	MINCHEG	Minutos da Chegada	218	219	2		
88	ANDA_D	Tempo Andando até o Destino	220	221	2		
89	ESTAC	Tipo de Estacionamento	222	222	1	1	1 - Zona Azul/Parquímetro
						2	2 - Estacionamento Particular
						3	3 - Estacionamento Próprio
						4	4 - Estacionamento Patrocinado
						5	5 - Meio Fio
						6	6 - Não Estacionou
90	DURACAO	Duração da Viagem (em minutos)	223	225	3		
91	MODOPRIN	Modo Principal	226	227	2	1	1 - Ônibus Diesel
						2	2 - Trólebus
						3	3 - Ônibus Fretado
						4	4 - Escolar
						5	5 - Dirigindo Automóvel
						6	6 - Passageiro de Automóvel
						7	7 - Táxi
						8	8 - Lotação/Perua
						9	9 - Metrô
						10	10 - Trem
						11	11 - Moto
						12	12 - Bicicleta
						13	13 - A Pé
						14	14 - Caminhão
						15	15 - Outros
92	TIPO_VG	Tipo de Viagem	228	228	1	1	1 - Coletivo
						2	2 - Individual
						3	3 - A pé
93	ID_ORDEM	Número de Ordem do Registro	229	235	7	1 a 223.926	

LAYOUT PESQUISA ORIGEM DESTINO 1997 - ZONA

Metrô - SP

Variável	Conteúdo	Início	Fim	Compr	Códigos
1 ZONA	Zona do Domicílio	1	3	3	1 a 389
2 SZ	Subzona do Domicílio	4	4	1	1 a 9
3 SZSEQ	Subzona Sequencial do Domicílio	5	8	4	1 a 1.400
4 MUNI_DOM	Município do Domicílio	9	10	2	1 a 39
5 DISTRDOM	Distrito do Domicílio (São Paulo)	11	12	2	1 a 96
6 DTRMUN	Distrito+Município do Domicílio	13	15	3	1 a 134
7 ID_DOM	Identifica Domicílio	16	23	8	
8 F_DOM	Identifica Primeiro Registro do Domicílio	24	24	1	0 - Demais registros 1- Primeiro registro do domicílio
9 FE_DOM	Fator de Expansão do Domicílio	25	34	10	10 dígitos 5 casas decimais
10 DOMICIL	Número do Domicílio	35	38	4	
11 TIPO_DOM	Tipo de Domicílio	39	39	1	1 - Particular 2 - Coletivo 3 - Favela
12 TOT_FAM	Total de Famílias no Domicílio	40	41	2	
13 NO_MORAD	Número de Moradores do Domicílio	42	43	2	
14 RESULDOM	Resultado do Domicílio	44	44	1	6 - Completa sem Viagem 7 - Completa com Viagem
15 ID_FAM	Identifica Família	45	54	10	
16 F_FAM	Identifica Primeiro Registro da Família	55	55	1	0 - Demais registros 1- Primeiro registro da família
17 FE_FAM	Fator de Expansão da Família	56	65	10	10 dígitos 5 casas decimais
18 FAMILIA	Número da Família	66	67	2	
19 RESULFAM	Resultado da Família	68	68	1	6 - Completa sem Viagem 7 - Completa com Viagem
20 NO_MORAF	Número de Moradores da Família	69	70	2	
21 TP_RMUN	Tempo de Residência no Município	71	72	2	
22 TP_RBAIR	Tempo de Residência no Bairro	73	74	2	
23 CONDMORA	Condição de Moradia	75	75	1	1 - Alugada 2 - Própria 3 - Cedida 4 - Outros 5 - Não Respondeu
24 QT_RADIO	Rádios	76	76	1	
25 QT_GEL	Geladeiras	77	77	1	
26 QT_TV	Tv. a Cores	78	78	1	
27 QT_VIDEO	Vídeo Cassetes	79	79	1	
28 QT_BANHO	Banheiros	80	80	1	
29 QT_AUTO	Automóveis	81	81	1	
30 QT_ASPIR	Aspiradores de Pó	82	82	1	
31 QT_MLAVA	Máquinas de Lavar	83	83	1	
32 QT_EMPRE	Empregados Domésticos	84	84	1	
33 QT_MICRO	Microcomputadores	85	85	1	
34 QT_TEL	Telefones	86	86	1	
35 QT_CEL	Celulares	87	87	1	
36 DIA_SEMA	Dia da Semana	88	88	1	2 - Segunda-feira 3 - Terça-feira 4 - Quarta-feira 5 - Quinta-feira 6 - Sexta-feira
37 RENDA_FA	Renda Familiar	89	96	8	8 dígitos 2 casas decimais
38 CD_RENFA	Código de Renda Familiar	97	97	1	1 - Renda Familiar Completa 2 - Não Tem Renda 3 - Renda Familiar Incompleta
39 RENDATRI	Renda Familiar Atribuída	98	105	8	8 dígitos 2 casas decimais
40 ABIPEME	Classificação ABIPEME	106	106	1	1 - A 2 - B

						3 - C
						4 - D
						5 - E
41	ID_PESS	Identifica Pessoa	107	118	12	
42	F_PESS	Identifica Primeiro Registro da Pessoa	119	119	1	0 - Demais registros 1- Primeiro registro da pessoa
43	FE_PESS	Fator de Expansão da Pessoa	120	129	10	10 dígitos 5 casas decimais
44	PESSOA	Número da Pessoa	130	131	2	
45	SIT_FAM	Situação Familiar	132	132	1	1 - Chefe 2 - Cônjugue 3 - Filho(a) 4 - Parente/Agregado 5 - Empregado Residente 6 - Visitante não Residente na RMSP
46	IDADE	Idade	133	134	2	(anos)
47	SEXO	Sexo	135	135	1	1 - Masculino 2 - Feminino
48	SE_ESTUD	Estuda Atualmente ?	136	136	1	1 - Não 2 - Creche/Pré-Escola 3 - 1o./2o./3o. Graus 4 - Outros
49	GRAU_INS	Grau de Instrução	137	137	1	1 - Não-alfabetizado 2 - Pré-Escola 3 - 1o. Grau Incompleto 4 - 1o. Grau Completo 5 - 2o. Grau Incompleto 6 - 2o. Grau Completo 7 - Superior Incompleto 8 - Superior Completo
50	CD_ATIVI	Condição de Atividade	138	138	1	1 - Ocupado 2 - Ocupado Eventualmente 3 - Em Licença 4 - Não Ocupado 5 - Aposentado/Pensionista 6 - Nunca Trabalhou 7 - Dona de Casa 8 - Estudante
51	OCUP_PRI	Ocupação Principal	139	140	2	01 - Assalariado com Carteira 02 - Assalariado sem Carteira 03 - Funcionário Público 04 - Autônomo 05 - Empregador 06 - Profissional Liberal 07 - Trab. Doméstico com Carteira 08 - Trab. Doméstico sem Carteira 09 - Dono de Negócio Familiar 10 - Trabalhador Familiar 11 - Não se Aplica
52	SET_ATIV	Setor de Atividade	141	142	2	01 - Agrícola 02 - Construção Civil 03 - Indústria 04 - Comércio 05 - Serviços de Transp. Carga 06 - Serviços de Transp. de Passag. 07 - Serviços Creditícios/Financeiros 08 - Serviços Pessoais 09 - Serviços de Alimentação 10 - Serviços de Saúde 11 - Serviços de Educação 12 - Serviços Especializados

						13 - Serviços da Adm. Pública
						14 - Outros
						15 - Não se Aplica
53	CO_REN_I	Condição de Renda Individual	143	143	1	1 - Tem Renda
						2 - Não Tem Renda
						3 - Não Respondeu
54	VL_REN_I	Renda Individual	144	151	8	8 dígitos 2 casas decimais
55	USA_VTRA	Usa Vale-Transporte ?	152	152	1	1 - Sim
						2 - Não
56	ZONAESC	Zona da Escola	153	155	3	1 a 389
57	SZESCOLA	Subzona da Escola	156	156	1	1 a 9
58	SZESEQ	Subzona Sequencial da Escola	157	160	4	1 a 1.400
59	DISTRESC	Distrito da Escola (São Paulo)	161	162	2	1 a 96
60	MUNIESC	Município da Escola	163	164	2	1 a 39
61	DTRMUESC	Distrito+Município da Escola	165	167	3	1 a 134
62	ZONATRA1	Zona do Primeiro Trabalho	168	170	3	1 a 389
63	SZTRAB1	Subzona do Primeiro Trabalho	171	171	1	1 a 9
64	SZT1SEQ	Subzona Sequencial do Primeiro Trabalho	172	175	4	1 a 1.400
65	DISTRAB1	Distrito do Primeiro Trabalho (São Paulo)	176	177	2	1 a 96
66	MUNITRA1	Município do Primeiro Trabalho	178	179	2	1 a 39
67	DTRMUTR1	Distrito+Município do Primeiro Trabalho	180	182	3	1 a 134
68	TRAB1_SN	Primeiro Trabalho é igual a Residência ?	183	183	1	1 - Sim
						2 - Não
69	ZONATRA2	Zona do Segundo Trabalho	184	186	3	1 a 389
70	SZTRAB2	Subzona do Segundo Trabalho	187	187	1	1 a 9
71	SZT2SEQ	Subzona Sequencial do Segundo Trabalho	188	191	4	1 a 1.400
72	DISTRAB2	Distrito do Segundo Trabalho (São Paulo)	192	193	2	1 a 96
73	MUNTRA2	Município do Segundo Trabalho	194	195	2	1 a 39
74	DTRMUTR2	Distrito+Município do Segundo Trabalho	196	198	3	1 a 134
75	TRAB2_SN	Segundo Trabalho é igual a Residência ?	199	199	1	1 - Sim
						2 - Não
76	ID_VIAGE	Identifica Viagem	200	205	6	
77	FE_VIA	Fator de Expansão da Viagem	206	215	10	10 dígitos 5 casas decimais
78	ZONA_O	Zona de Origem	216	218	3	1 a 389
79	SZO	Subzona de Origem	219	219	1	1 a 9
80	SZOSEQ	Subzona Sequencial de Origem	220	223	4	1 a 1.400
81	DISTRORG	Distrito de Origem	224	225	2	1 a 96
82	MUNIORIG	Município de Origem	226	227	2	1 a 39
83	DTRMUNO	Distrito+Município de Origem	228	230	3	1 a 134
84	ID_POLOO	Pólo Gerador na Origem	231	231	1	
85	ZONA_D	Zona de Destino	232	234	3	1 a 389
86	SZD	Subzona de Destino	235	235	1	1 a 9
87	SZDSEQ	Subzona Sequencial de Destino	236	239	4	1 a 1.400
88	DISTRD	Distrito de Destino	240	241	2	1 a 96
89	MUNIDEST	Município de Destino	242	243	2	1 a 39
90	DTRMUND	Distrito+Município de Destino	244	246	3	1 a 134
91	ID_POLOD	Pólo Gerador no Destino	247	247	1	
92	MOTIVO_O	Motivo na Origem	248	248	1	1 - Trabalho Indústria 2 - Trabalho Comércio 3 - Trabalho Serviços 4 - Escola/Educação 5 - Compras 6 - Médico/Dentista/Saúde 7 - Recreação/Visitas 8 - Residência 9 - Outros
93	MOTIVO_D	Motivo no Destino	249	249	1	idem ao anterior
94	SERVIR_O	Servir Passageiro na Origem	250	250	1	1 - Sim 2 - Não
95	SERVIR_D	Servir Passageiro no Destino	251	251	1	1 - Sim

						2 - Não
96	MODO1	Modo 1	252	253	2	01 - Ônibus
						02 - Ônibus Fretado
						03 - Transporte Escolar
						04 - Dirigindo Automóvel
						05 - Passageiro de Automóvel
						06 - Táxi
						07 - Lotação/Perua
						08 - Metrô
						09 - Trem
						10 - Moto
						11 - Bicicleta
						12 - A Pé
						13 - Outros
97	MODO2	Modo 2	254	255	2	idem ao anterior
98	MODO3	Modo 3	256	257	2	idem ao anterior
99	MODO4	Modo 4	258	259	2	idem ao anterior
100	PQ_VIAPE	Motivo da Viagem a Pé	260	260	1	1 - Condução Cara 2 - Condução Desconfortável 3 - Ponto/Estação Distante 4 - Condução Demora a Passar 5 - Condução Lotada 6 - Viagem Demorada 7 - Horário Irregular (da condução) 8 - Pequena Distância 9 - Outros Motivos
101	H_SAIDA	Hora da Saída	261	262	2	
102	MINSAIDA	Minutos da Saída	263	264	2	
103	ANDA_O	Minutos Andando desde a Origem	265	266	2	
104	H_CHEG	Hora da Chegada	267	268	2	
105	MINCHEG	Minutos da Chegada	269	270	2	
106	ANDA_D	Minutos Andando até o Destino	271	272	2	
107	DURACAO	Duração da Viagem (em minutos)	273	275	3	
108	MODOPRIN	Modo Principal	276	277	2	01 - Ônibus 02 - Ônibus Fretado 03 - Transporte Escolar 04 - Dirigindo Automóvel 05 - Passageiro de Automóvel 06 - Táxi 07 - Lotação/Perua 08 - Metrô 09 - Trem 10 - Moto 11 - Bicicleta 12 - A Pé 13 - Outros
109	TIPO_VG	Tipo de Viagem	278	278	1	1 - Coletivo 2 - Individual 3 - A pé
110	ID_ORDEM	Número de Ordem do Registro	279	284	6	

LAYOUT PESQUISA ORIGEM DESTINO 2007

Variável	Conteúdo	Início	Fim	Compr	Códigos
1 ZONA	Zona do Domicílio	1	3	3	1 a 460
2 MUNI_DOM	Município de Domicílio	4	5	2	1 a 39
3 CO_DOM_X	Coordenada X Domicílio	6	17	12	12 dígitos 2 casas decimais
4 CO_DOM_Y	Coordenada Y Domicílio	18	29	12	12 dígitos 2 casas decimais
5 ID_DOM	Identifica Domicílio	30	36	7	
6 F_DOM	Identifica Primeiro Registro do Domicílio	37	37	1	0 - Demais Registros 1- Primeiro Registro do Domicílio
7 FE_DOM	Fator de Expansão do Domicílio	38	47	10	10 dígitos 5 casas decimais
8 DOM	Número do Domicílio	48	51	4	
9 CD_ENTRE	Código de Entrevista	52	52	1	5 - Completa sem Viagem 6 - Completa com Viagem
10 DATA	Data da Entrevista	53	60	8	
11 TIPO_DOM	Tipo de Domicílio	61	61	1	1 – Particular 2 – Coletivo 3 – Favela
12 NO_MORAD	Total de Moradores no Domicílio	62	63	2	
13 TOT_FAM	Total de Famílias no Domicílio	64	65	2	
14 ID_FAM	Identifica Família	66	74	9	
15 F_FAM	Identifica Primeiro Registro da Família	75	75	1	0 - Demais Registros 1- Primeiro Registro da Família
16 FE_FAM	Fator de Expansão da Família	76	85	10	10 dígitos 5 casas decimais
17 FAMILIA	Número da Família	86	87	2	
18 NO_MORAF	Total de Moradores na Família	88	89	2	
19 CONDMORA	Condição de Moradia	90	90	1	1 - Alugada 2 - Própria 3 - Cedida 4 - Outros 5 - Não Respondeu
20 QT_RADIO	Rádios	91	91	1	
21 QT_GEL1	Geladeiras de 1 porta	92	92	1	
22 QT_GEL2	Geladeiras de 2 portas	93	93	1	
23 QT_TVCOR	Tv. a Cores	94	94	1	
24 QT_FREEZ	Freezer	95	95	1	
25 QT_VIDEO	Vídeo Cassetes/DVD	96	96	1	
26 QT_BANHO	Banheiros	97	97	1	
27 QT_MOTO	Motos	98	98	1	
28 QT_AUTO	Automóveis	99	99	1	
29 QT_ASPIR	Aspiradores de Pó	100	100	1	
30 QT_MLAVA	Máquinas de Lavar	101	101	1	
31 QT_EMPRE	Empregados Domésticos	102	102	1	
32 QT_MICRO	Microcomputadores	103	103	1	
33 QT_BICICLE	Bicicletas	104	104	1	
34 NAO_DCL_IT	Código de Declaração de Itens de Conforto	105	105	1	
35 CRITERIO_B	Crítario de Classificação Econômica Brasil	106	106	1	1 – A1 2 – A2 3 – B1 4 – B2 5 – C1 6 – C2 7 – D 8 – E
36 ANO_AUTO1	Ano Fabricação - Auto 1	107	110	4	
37 ANO_AUTO2	Ano Fabricação - Auto 2	111	114	4	
38 ANO_AUTO3	Ano Fabricação - Auto 3	115	118	4	
39 RENDA_FA	Renda Familiar	119	126	8	8 dígitos 2 casa decimais
40 CD_RENFA	Código de Renda Familiar	127	127	1	1 - Renda Familiar Declarada e Maior que Zero 2 - Renda Familiar Declarada como Zero 3 - Renda Atribuída pelo Critério Brasil 4 - Renda Atribuída pela Média da Zona
41 ID_PESS	Identifica Pessoa	128	139	12	
42 F_PESS	Identifica Primeiro Registro da Pessoa	140	140	1	0 - Demais registros 1- Primeiro registro da pessoa
43 FE_PESS	Fator de Expansão da Pessoa	141	150	10	10 dígitos 5 casas decimais
44 PESSOA	Número da Pessoa	151	152	2	
45 SIT_FAM	Situação Familiar	153	153	1	1 – Pessoa Responsável 2 – Cônjuge/Companheiro(a)

						3 – Filho(a)/Enteado(a)
						4 - Outro Parente
						5 - Agregado
						6 - Empregado Residente
						7 - Parente do Empregado
46	IDADE	Idade	154	155	2	(anos)
47	SEXO	Gênero	156	156	1	1 - Masculino 2 - Feminino
48	ESTUDA	Estuda Atualmente?	157	157	1	1 - Não 2 - Creche/Pré-Escola 3 - 1º Grau /Fundamental 4 - 2º Grau/Médio 6 – Superior/Universitário 7 - Outros
49	GRAU_INS	Grau de Instrução	158	158	1	1 - Não-Alfabetizado/Primário Incompleto 2 - Primário Completo/Ginásio Incompleto 3 - Ginásio Completo/Colegial Incompleto 4 - Colegial Completo/Superior Incompleto 5 - Superior Completo
50	CD_ATIVI	Condição de Atividade	159	159	1	1 - Tem trabalho 2 - Faz bico 3 - Em Licença Médica 4 - Aposentado/Pensionista 5 - Sem Trabalho 6 - Nunca Trabalhou 7 - Dona de Casa 8 - Estudante
51	CO_REN_I	Condição de Renda Individual	160	160	1	1 - Tem Renda 2 - Não Tem Renda 3 - Não Respondeu
52	VL_REN_I	Renda Individual	161	168	8	8 dígitos
53	ZONA_ESC	Zona da Escola	169	171	3	1 a 460
54	MUNIESC	Município da Escola	172	173	2	
55	CO_ESC_X	Coordenada X Escola	174	185	12	12 dígitos 2 casas decimais
56	CO_ESC_Y	Coordenada Y Escola	186	197	12	12 dígitos 2 casas decimais
57	TIPO_ESC	Tipo de Escola	198	198	1	1 - Pública 2 - Particular
58	ZONATRA1	Zona do Primeiro Trabalho	199	201	3	1 a 460
59	MUNITRA1	Município do Primeiro Trabalho	202	203	2	1 a 39
60	CO_TR1_X	Coordenada X 1º Trabalho	204	215	12	12 dígitos 2 casas decimais
61	CO_TR1_Y	Coordenada Y 1º Trabalho	216	227	12	12 dígitos 2 casas decimais
62	TRAB1_RE	Primeiro Trabalho é igual a Residência ?	228	228	1	1 - Sim 2 - Não 3 - Sem endereço fixo
63	TRABEXT1	Realiza Trabalho Externo-1º Trabalho	229	229	1	1 - Sim 2 - Não
64	OCUP1	Ocupação do 1º Trabalho	230	232	3	
65	SETOR1	Setor de Atividade do 1º Trabalho	233	234	2	
66	VINC1	Vínculo Empregatício do 1º Trabalho	235	235	1	1 - Assalariado com carteira 2 - Assalariado sem carteira 3 - Funcionário Público 4 - Autônomo 5 - Empregador 6 - Profissional Liberal 7 – Dono de Negócio Familiar 8 - Trabalho Familiar
67	ZONATRA2	Zona do Segundo Trabalho	236	238	3	1 a 460
68	MUNITRA2	Município do Segundo Trabalho	239	240	2	1 a 39
69	CO_TR2_X	Coordenada X 2º Trabalho	241	252	12	12 dígitos 2 casas decimais
70	CO_TR2_Y	Coordenada Y 2º Trabalho	253	264	12	12 dígitos 2 casas decimais
71	TRAB2_RE	Segundo Trabalho é igual a Residência ?	265	265	1	1 - Sim 2 - Não 3 - Sem endereço fixo
72	TRABEXT2	Realiza Trabalho Externo 2º Trabalho	266	266	1	1 - Sim 2 - Não
73	OCUP2	Ocupação do 2º Trabalho	267	269	3	
74	SETOR2	Setor de Atividade do 2º Trabalho	270	271	2	
75	VINC2	Vínculo Empregatício do 2º Trabalho	272	272	1	idem ao 1º Trabalho
76	N_VIAG	Número da Viagem	273	274	2	

77	FE_VIA	Fator de Expansão da Viagem	275	284	10	10 dígitos 5 casas decimais
78	DIA_SEM	Dia da Semana	285	285	1	2 - Segunda-Feira 3 - Terça-Feira 4 - Quarta-Feira 5 - Quinta-Feira 6 - Sexta-Feira
79	TOT_VIAG	Total de Viagens da Pessoa	286	287	2	
80	ZONA_O	Zona de Origem	288	290	3	1 a 460
81	MUNI_O	Município de Origem	291	292	2	1 a 39
82	CO_O_X	Coordenada X Origem	293	304	12	12 dígitos 2 casas decimais
83	CO_O_Y	Coordenada Y Origem	305	316	12	12 dígitos 2 casas decimais
84	ZONA_D	Zona de Destino	317	319	3	1 a 460
85	MUNI_D	Município de Destino	320	321	2	1 a 39
86	CO_D_X	Coordenada X Destino	322	333	12	12 dígitos 2 casas decimais
87	CO_D_Y	Coordenada Y Destino	334	345	12	12 dígitos 2 casas decimais
88	ZONA_T1	Zona da 1ª Transferência	346	348	3	1 a 460
89	MUNI_T1	Município 1ª Transferência	349	350	2	1 a 39
90	CO_T1_X	Coordenada X 1ª Transferência	351	362	12	12 dígitos 2 casas decimais
91	CO_T1_Y	Coordenada Y 1ª Transferência	363	374	12	12 dígitos 2 casas decimais
92	ZONA_T2	Zona da 2ª Transferência	375	377	3	1 a 460
93	MUNI_T2	Município 2ª Transferência	378	379	2	1 a 39
94	CO_T2_X	Coordenada X 2ª Transferência	380	391	12	12 dígitos 2 casas decimais
95	CO_T2_Y	Coordenada Y 2ª Transferência	392	403	12	12 dígitos 2 casas decimais
96	ZONA_T3	Zona da 3ª Transferência	404	406	3	1 a 460
97	MUNI_T3	Município 3ª Transferência	407	408	2	1 a 39
98	CO_T3_X	Coordenada X 3ª Transferência	409	420	12	12 dígitos 2 casas decimais
99	CO_T3_Y	Coordenada Y 3ª Transferência	421	432	12	12 dígitos 2 casas decimais
100	MOTIVO_O	Motivo na Origem	433	434	2	1 - Trabalho/Indústria 2 - Trabalho/Comércio 3 - Trabalho/Serviços 4 - Educação 5 - Compras 6 - Saúde 7 - Lazer 8 - Residência 9 - Procurar Emprego 10 – Assuntos Pessoais
101	MOTIVO_D	Motivo no Destino	435	436	2	idem ao anterior
102	SERVIR_O	Servir Passageiro na Origem	437	437	1	1 - Sim 2 - Não
103	SERVIR_D	Servir Passageiro no Destino	438	438	1	1 - Sim 2 - Não
104	MODO1	Modo 1	439	440	2	01 – Ônibus Município S.Paulo 02 – Ônibus Outros Municípios 03 – Ônibus Metropolitano 04 – Ônibus Fretado 05 - Escolar 06 - Dirigindo Automóvel 07 - Passageiro de Automóvel 08 - Táxi 09 – Microônibus/Van Município de S.Paulo 10 – Microônibus/Van Outros Município 11 – Microônibus/Van Metropolitano 12 - Metrô 13 - Trem 14 - Moto 15 - Bicicleta 16 - A Pé 17 - Outros
105	MODO2	Modo 2	441	442	2	idem ao anterior
106	MODO3	Modo 3	443	444	2	idem ao anterior
107	MODO4	Modo 4	445	446	2	idem ao anterior
108	H_SAIDA	Hora Saída	447	448	2	Hora de Saída
109	MIN_SAIDA	Minuto Saída	449	450	2	Minuto de Saída
110	ANDA_O	Tempo Andando na Origem	451	452	2	Em minutos
111	H_CHEG	Hora Chegada	453	454	2	Hora de Chegada
112	MIN_CHEG	Minuto Chegada	455	456	2	Minuto de Chegada
113	ANDA_D	Tempo Andando no Destino	457	458	2	Em minutos
114	DURACAO	Duração da Viagem (em minutos)	459	461	3	

115 MODOPRIN	Modo Principal	462	463	2	01 – Ônibus Município S.Paulo 02 – Ônibus Outros Municípios 03 – Ônibus Metropolitano 04 - Ônibus Fretado 05 - Escolar 06 - Dirigindo Automóvel 07 - Passageiro de Automóvel 08 - Táxi 09 – Microônibus/Van Município de S.Paulo 10 – Microônibus/Van Outros Município 11 – Microônibus/Van Metropolitano 12 - Metrô 13 - Trem 14 - Moto 15 - Bicicleta 16 - A Pé 17 - Outros
116 TIPOVG	Tipo de Viagem	464	464	1	1 - Coletivo 2 - Individual 3 - A pé
117 PAG_VIAG	Quem Pagou a Viagem	465	465	1	1 – Você/Sua Família 2 – Empregador 3 - Isento 4 - Outros
118 TP_ESAUTO	Tipo de Estacionamento Automóvel	466	466	1	1 – Não Estacionou 2 – Zona azul/Zona marrom 3 – Patrocinado 4 – Proprio 5 – Meio-Fio 6 – Avulso 7 – Mensal 8 – Não respondeu
119 VL_EST	Valor do Estacionamento Automóvel	467	470	4	
120 PE_BICI	Por Que Viajou A Pé ou Bicicleta	471	471	1	1 – Pequena Distância 2 – Condução Cara 3 – Ponto/Estação Distante 4 – Condução Demora a Passar 5 – Viagem Demorada 6 – Condução Lotada 7 – Atividade Física 8 – Outros Motivos
121 TP_ESBICI	Estacionamento Bicicleta	472	472	1	1 – Bicicletário Gratuito 2 – Bicicletário Pago 3 – Local Privado 4 – Rua/Local Público 5 – Outros
122 ID_ORDEM	Número de Ordem do Registro	473	478	6	1 a 196.698

ANEXO D – Síntese de resultados para 4 conglomerados

RESUMO DE RESULTADOS PARA AGRUPAMENTO POR ATRIBUTOS DE VIAGENS DA FAMÍLIA - MÉTODO WARD		CARACTERÍSTICAS DE VIAGENS																			
Cluster nº (ward)	Nº de pessoas	% de pessoas de 1977	% de pessoas de 1987	% de pessoas de 1997	% de pessoas de 2007	Nº de registros	Nº de viagens	Nº médio de viagens por pessoa	Nº médio de viagens por família	Média da distância total da pessoa (m)	Média da distância total da família (m)	Distância média por viagem da pessoa (m)	Distância média por viagem da família (m)	Média da duração total de viagens da pessoa (min)	Média da duração total de viagens da família (min)	Média da duração de viagens da pessoa (min)	Média da duração de viagens da família (min)	Duração média de viagens da pessoa (min)	Duração média de viagens da família (min)	Duração média de viagens da pessoa no destino (min)	% de pessoas que servem passageiro no destino
1	100	0	0	0	0	229046	187441	2.82	7.70	13049.87	35647.09	4629.49	4631.85	87.68	239.51	31.09	31.11	31.11	2.9		
2	0	100	0	0	0	229026	183114	2.82	7.00	12022.88	34465.88	4517.86	4518.86	87.72	232.94	31.44	31.44	32.67			
3	0	0	100	0	0	198640	163534	6.81	6.81	10541.04	27493.84	4038.71	4037.27	86.66	226.04	33.20	33.19	33.19			
4	0	0	0	100	100	198698	169665	2.64	6.25	13228.73	31576.57	5010.88	5020.25	94.56	224.28	35.82	35.88	35.88	8.19		
Diferenças % entre Min e Máx		7%		19%		20%		23%		19%		20%		10%		6%		13%			
RESUMO DE RESULTADOS PARA AGRUPAMENTO POR ATRIBUTOS DE PESSOAS - MÉTODO WARD		CARACTERÍSTICAS DE PESSOAS																			
Cluster nº (ward)	Nº de pessoas	% de pessoas do sexo feminino	% de pessoas do sexo masculino	Média de idade (anos)	% de pessoa responsável	% de cônjuges	% filhos / enteados	% de outros parentes / agregados	% de empregados	% de outros	% de pessoas não alfabetizadas ou com fundamental incompleto	% de pessoas com fundamental incompleto ou médio incompleto	% de pessoas com médio completo ou superior incompleto	% de pessoas com superior completo	% de pessoas que estudam	% de pessoas que trabalham	Média da Renda Individual (R\$ - Out/2007)				
1	108028	51.71	48.29	27.17	24.13	18.9	44.36	10.11	0.24	74.81	11.16	9.40	4.83	24.85	35.59	851.24					
2	110813	52.01	47.99	27.84	25.45	19.67	44.17	9.80	0.19	72.92	9.78	5.84	25.89	37.17	555.39						
3	98780	52.09	47.91	30.1	27.18	19.47	41.66	10.66	0.08	60.03	11.47	14.71	7.97	29.66	41.99		668.88				
4	91405	53.73	46.27	37.08	33.78	24.49	10.00	1.17	0.07	37.91	14.97	27.76	19.37	25.73	47.98		616.90				
Diferenças % entre Min e Máx		4%		27%		29%		8%		59%		71%		35%		76%		35%			
RESUMO DE RESULTADOS PARA AGRUPAMENTO POR ATRIBUTOS DE FAMÍLIAS - MÉTODO WARD		CARACTERÍSTICAS DE FAMÍLIAS																			
Cluster nº (ward)	Nº de famílias	Tamanho médio da família	% de famílias com presença de criança entre 0 e 4 anos	% de famílias com presença de criança entre 5 e 9 anos	% de famílias com presença de crianças entre 10 e 14 anos	% de famílias com presença de idosos com 60 anos ou mais	% de famílias com presença de idosos entre 10 e 14 anos	Média da quantidade de trabalhador(es) na família	% de famílias que têm 1 automóvel	% de famílias que têm 2 ou mais automóveis	Média da quantidade de autos na família	% de famílias na Classe E (até 2 SM)	% de famílias na Classe D (de 2 a 4 SM)	% de famílias na Classe C (de 4 a 10 SM)	% de famílias na Classe B (de 10 a 20 SM)	% de famílias na Classe A (mais de 20 SM)	Média da Renda Família (R\$ - Out/2007)				
1	26157	4.13	29.99	27.65	28.11	89.96	1.46	34.06	13.53	0.65	7.88	20.94	37.09	19.92	4037.27						
2	28217	3.93	30.70	32.42	23.23	87.22	1.45	32.93	10.65	0.57	27.55	34.27	11.89	4.58		2393.88					
3	26560	3.98	29.96	24.59	19.04	82.56	1.54	38.25	15.56	0.72	25.78	25.94	7.39	7.39		2656.99					
4	30855	2.96	12.40	14.41	36.57	80.00	1.42	38.25	17.55	0.78	10.58	25.26	35.41	17.93		3196.19					
Diferenças % entre Min e Máx		28%		60%		56%		43%		11%		39%		27%		30%		47%			
RESUMO DE RESULTADOS PARA AGRUPAMENTO POR ATRIBUTOS DE PESSOAS - MÉTODO CENTROIDE		CARACTERÍSTICAS DE VIAGENS																			
Cluster nº (centr)	Nº de registros	Nº de viagens	% de pessoas de 1977	% de pessoas de 1987	% de pessoas de 1997	% de pessoas de 2007	Nº médio de viagens	Nº médio de viagens por pessoa	Média da distância total da pessoa (m)	Média da distância total da família (m)	Distância média por viagem da pessoa (m)	Distância média por viagem da família (m)	Média da duração total de viagens da pessoa (min)	Média da duração total de viagens da família (min)	Média da duração de viagens da pessoa (min)	Média da duração de viagens da família (min)	Duração média de viagens da pessoa (min)	Duração média de viagens da família (min)	Duração média de viagens da pessoa no destino (min)	% de pessoas que servem passageiro no destino	
1	229046	187441	100	0	0	0	2.82	7.70	13049.87	35647.09	4629.49	4631.85	87.68	239.51	31.09	31.11	31.11	2.9			
2	123485	102679	0	100	0	0	2.68	7.10	12107.87	32053.83	4517.86	4518.86	85.24	225.66	31.81	31.78					
3	100441	80439	0	100	0	0	2.54	6.87	12175.88	32699.81	4793.65	4799.10	85.86	232.94	33.80	33.84					
4	396338	333199	0	0	51.94	48.06	8.51	8.51	11902.85	29593.24	4543.07	4539.67	90.66	225.11	34.60	34.58					
Diferenças % entre Min e Máx		10%		7%		7%		6%		6%		6%		10%		61%		7.48			
RESUMO DE RESULTADOS PARA AGRUPAMENTO POR ATRIBUTOS DE PESSOAS - MÉTODO CENTROIDE		CARACTERÍSTICAS DE PESSOAS																			
Cluster nº (centr)	Nº de pessoas	% de pessoas do sexo feminino	% de pessoas do sexo masculino	Média de idade (anos)	% de pessoa responsável	% de cônjuges	% filhos / enteados	% de outros parentes / agregados	% de empregados	% de outros	% de pessoas não alfabetizadas ou com fundamental incompleto	% de pessoas com fundamental incompleto ou médio incompleto	% de pessoas com médio completo ou superior incompleto	% de pessoas com superior completo	% de pessoas que estudam	% de pessoas que trabalham	Média da Renda Individual (R\$ - Out/2007)				
1	108028	51.71	48.29	27.17	24.13	18.9	44.36	10.11	0.24	74.81	11.16	9.40	4.83	24.85	35.59	851.24					
2	59103	52.01	47.99	27.84	25.45	19.67	44.17	9.80	0.19	72.92	9.78	5.84	25.89	37.17	555.39						
3	51713	52.09	47.91	30.1	27.18	19.47	41.66	10.66	0.08	60.03	11.47	14.71	7.97	29.66	41.99						
4	190185	52.88	47.12	33.46	30.34	19.98	38.21	10.34	0.07	49.40	16.18	20.98	13.45	35.08	44.86						
Diferenças % entre Min e Máx		4%		23%		20%		5%		19%		76%		71%		73%		22%			
RESUMO DE RESULTADOS PARA AGRUPAMENTO POR ATRIBUTOS DE FAMÍLIAS - MÉTODO CENTROIDE		CARACTERÍSTICAS DE FAMÍLIAS																			
Cluster nº (centr)	Nº de famílias	Tamanho médio da família	% de famílias com presença de criança entre 0 e 4 anos	% de famílias com presença de criança entre 5 e 9 anos	% de famílias com presença de crianças entre 10 e 14 anos	% de famílias com presença de idosos com 60 anos ou mais	% de famílias com presença de idosos entre 10 e 14 anos	Média da quantidade de trabalhador(es) na família	% de famílias que têm 1 automóvel	% de famílias que têm 2 ou mais automóveis	Média da quantidade de autos na família	% de famílias na Classe E (até 2 SM)	% de famílias na Classe D (de 2 a 4 SM)	% de famílias na Classe C (de 4 a 10 SM)	% de famílias na Classe B (de 10 a 20 SM)	% de famílias na Classe A (mais de 20 SM)	Média da Renda Família (R\$ - Out/2007)				
1	26157	4.13	29.99	27.65	28.11	89.96	1.46	34.06	13.53	0.65	7.88	20.94	37.09	19.92	4037.27						
2	15685	3.77	26.83	28.63	23.77	26.90	1.46	34.27	12.65	0.63	18.38	35.08	14.56	5.96		2677.95					
3	125520	3.13	33.95	37.16	19.33	18.92	1.43	34.26	8.59	0.50	22.95	33.24	14.57	2.89		2845.49					
4	57700	3.30	17.46	20.94	31.66	20.94	1.48	36.67	15.51	0.75	12.85	23.93	14.68	8.41		50%					
Diferenças % entre Min e Máx		20%		51%		48%		31%		41%		8%		33%		66%		77%			

ANEXO E – Síntese de resultados para 4 conglomerados - segmentação por ano

RESUMO DE RESULTADOS PARA AGRUPAMENTO POR ATRIBUTOS DE VIAGENS DA FAMÍLIA												CARACTERÍSTICAS DE VIAGENS DAS PESSOAS E DAS FAMÍLIAS											
Cluster (1977)	% de pessoas de 1977	% de pessoas de 1987	% de pessoas de 1997	% de pessoas de 2007	Nº de registos	Nº médio de viagens por pessoa	Nº médio de viagens por família	Média da distância total da pessoa (m)	Média da distância total da família (m)	Distância média por viagem da pessoa	Média da duração total da viagem da pessoa (min)	Média da duração total de viagens da família (min)	Duração média de viagens da pessoa	Duração média de viagens da família									
CF01	100	0	0	0	52400	3,39	9,46	12457,5	32392,24	146,39	417,78	56,0	243,98	28,79	28,83								
CF02	100	0	0	0	58756	2,66	5,52	15161,07	42866,07	5699,78	506,0	253,97	33,77	33,77									
CF03	100	0	0	0	47948	2,99	7,06	10141,83	23954,91	3391,92	3393,05	82,01	193,69	27,43	27,43								
CF04	100	0	0	0	60403	2,68	7,71	13778,57	39635,88	5141,26	5140,84	91,42	263	34,11	34,11								
Diferenças % entre mínimos e máximos												10%	26%	20%	20%								
CARACTERÍSTICAS DE VIAGENS												CARACTERÍSTICAS DE VIAGENS											
Cluster (1977)	Nº de viagens	Nº de pessoas	Nº de famílias	% de viagens realizadas a pé	% de viagens realizadas por transporte individual	% de viagens realizadas por trem	% de viagens realizadas por metro	% de viagens realizadas por ônibus escolar / fretado	% de viagens realizadas por motivo trabalho	% de viagens realizadas por motivo educação	% de viagens realizadas por motivo servir / comércio	% de viagens realizadas por motivo manutenção/ comensal	% de viagens realizadas por motivo lazer/ outros										
CF01	52400	27059	6584	26,8	45,3	27,9	1,3	7,9	83,1	7,4	38,3	24,0	4,8	25,6	21,1	21,1							
CF02	46314	29837	6587	38,3	27,4	33,4	14,4	0,8	78,0	6,2	41,1	3,5	6,4	24,8	21,1	21,1							
CF03	40576	20933	6290	34,4	37,2	28,4	1,2	8,2	84,9	5,5	43,8	21,3	3,8	6,3	27,43	27,43							
CF04	48151	30199	6696	33,2	33,2	33,6	4,3	6,3	86,1	9,9	39,8	28,1	4,2	6,7	21,1	21,1							
Diferenças % entre mínimos e máximos												9%	25%	27%	14%	17%							
CARACTERÍSTICAS DE PESSOAS												CARACTERÍSTICAS DE PESSOAS											
Cluster (1977)	% de pessoas do sexo feminino	% de pessoas do sexo masculino	Média de idade (anos)	% de pessoa responsável	% de conjuges	% filhos / enteados	% de outros parentes / agregados	% de empregados	% de pessoas com fundamental incompleto	% de pessoas com médio completo ou superior incompleto	% de pessoas com superior completo	% de pessoas que estudam	% de pessoas que trabalham	Média de Renda Família (R\$ - out/2007)									
CF01	52,9	47,1	29,08	24,2	19,6	41,6	10,3	3,9	66,8	12,7	2,7	25,2	36,4	110,32									
CF02	50,3	49,6	24,7	22,0	18,2	49,7	9,1	8,0	8,0	4,9	2,0	32,3	42,2	104,98									
CF03	53,3	46,7	31,44	29,9	19,4	34,5	12,7	32,0	14,7	15,7	7,7	22,7	24,4	111,84									
CF04	51,0	49,0	25,51	22,1	18,6	48,4	9,3	8,0	9,9	6,5	2,9	25,5	33,6	698,59									
Difs % mdx-min												27%	11%	47%									
CARACTERÍSTICAS DE FAMÍLIAS												CARACTERÍSTICAS DE FAMÍLIAS											
Cluster (1977)	Tamanho médio da família	% de famílias com presença de crianças entre 0 e 4 anos	% de famílias com presença de crianças entre 5 e 9 anos	% de famílias com presença de crianças entre 10 e 14 anos	% de famílias com presença de idosos com 60 anos ou mais	Média da quantidade de trabalhadores (as) na família	% de famílias que têm 1 automóvel	Média da quantidade de pessoas na família	% de famílias na Classe E (até 2 SM)	% de famílias na Classe D (de 2 a 4 SM)	% de famílias na Classe C (de 4 a 10 SM)	% de famílias na Classe B (de 10 a 20 SM)	% de famílias na Classe A (mais de 20 SM)	Média da Renda Família (R\$ - out/2007)									
CF01	4,11	26,9	24,1	25,9	24,2	1,49	38,7	20,8	5,2	14,6	24,3	21,2	21,2	21,00									
CF02	4,53	39,0	36,5	34,3	16,0	1,45	29,2	8,8	10,4	27,8	39,2	15,2	7,3	293,97									
CF03	3,33	18,5	15,4	17,3	25,0	1,37	35,2	11,7	7,1	16,9	34,5	23,7	17,1	4654,63									
CF04	4,51	35,0	33,9	34,3	17,2	1,51	32,1	12,8	8,8	24,2	39,7	16,7	10,5	3486,43									
Difs % mdx-min												5%	19%	43%									
RESUMO DE RESULTADOS PARA AGRUPAMENTO POR ATRIBUTOS DE VIAGENS DA FAMÍLIA												CARACTERÍSTICAS DE VIAGENS DAS PESSOAS E DAS FAMÍLIAS											
Cluster (1987)	% de pessoas de 1977	% de pessoas de 1987	% de pessoas de 1997	% de pessoas de 2007	Nº de registos	Nº médio de viagens por pessoa	Nº médio de viagens por família	Média da distância total da pessoa (m)	Média da distância total da família (m)	Distância média por viagem da pessoa	Média da duração total da viagem da pessoa (min)	Média da duração total de viagens da família (min)	Duração média de viagens da pessoa	Duração média de viagens da família									
CF05	0	100	0	0	44830	2,47	6,63	13003,59	34939,12	5264,61	5269,85	87,61	35,47	35,50									
CF06	0	100	0	0	60060	2,54	6,96	14221,63	35997,06	5599,07	55997,06	93,43	36,78	36,77									
CF07	0	100	0	0	63425	2,82	7,23	10003,00	25626,49	3547,45	3544,74	77,09	197,50	27,34									
CF08	0	100	0	0	55611	2,59	7,07	11487,45	31008,35	4435,31	4428,34	84,41	230,04	32,54									
Diferenças % entre mínimos e máximos												37%	17%	28%									
CARACTERÍSTICAS DE VIAGENS												CARACTERÍSTICAS DE VIAGENS											
Cluster (1987)	Nº de viagens	Nº de pessoas	Nº de famílias	% de viagens realizadas a pé	% de viagens realizadas por transporte individual	% de viagens realizadas por trem	% de viagens realizadas por metro	% de viagens realizadas por ônibus de linha	% de viagens realizadas por ônibus escolar / fretado	% de viagens realizadas por motivo trabalho	% de viagens realizadas por motivo educação	% de viagens realizadas por motivo servir / comércio	% de viagens realizadas por motivo manutenção/ comensal	% de viagens realizadas por motivo lazer/ outros									
CF05	35545	23684	5719	40,3	23,0	9,0	2,3	77,1	11,4	42,4	34,1	-	15,8										
CF06	30618	28482	7489	357	24,5	36,1	5,7	75,1	12,2	42,5	36,78	-	16,8										
CF07	54134	28482	8196	28,7	24,2	20,0	3,7	74,2	6,9	40,6	37,2	-	12,2										
CF08	44894	28029	6813	38,7	26,7	34,7	9,0	72,2	11,8	41,3	31,8	8,5	18,4										
Difs % mdx-min												5%	20%	32%									
CARACTERÍSTICAS DE PESSOAS												CARACTERÍSTICAS DE PESSOAS											
Cluster (1987)	% de pessoas do sexo feminino	% de pessoas do sexo masculino	Média de idade (anos)	% de pessoas responsável	% de conjuges	% filhos / enteados	% de outros parentes / agregados	% de empregados	% de pessoas com fundamental incompleto	% de pessoas com médio completo ou superior incompleto	% de pessoas com superior completo	% de pessoas que estudam	% de pessoas que trabalham	Média de Renda Individual (R\$ - out/2007)									
CF05	58,8	42,2	25,44	24,1	19,4	45,7	8,3	0,5	9,2	6,5	3,4	26,8	34,7	420,5									
CF06	58,8	48,2	23,04	24,5	19,5	45,7	8,3	0,5	11,7	8,8	3,4	484,13	486,43										
CF07	54,3	45,7	22,35	28,7	20,0	37,5	11,5	2,0	13,5	15,2	11,9	24,3	26,7	826,65									
CF08	51,0	49,0	26,15	24,3	19,8	46,4	8,7	0,6	7,7	10,8	7,7	3,7	25,9	35,4	471,57								
Difs % mdx-min												21%	16%	49%									
CARACTERÍSTICAS DE FAMÍLIAS												CARACTERÍSTICAS DE FAMÍLIAS											
Cluster (1987)	Tamanho médio da família	% de famílias com presença de crianças entre 0 e 4 anos	% de famílias com presença de crianças entre 5 e 9 anos	% de famílias com presença de crianças entre 10 e 14 anos	% de famílias com presença de idosos com 60 anos ou mais	Média da quantidade de trabalhadores (as) na família	% de famílias que têm 1 automóvel	Média da quantidade de pessoas na família	% de famílias na Classe E (até 2 SM)	% de famílias na Classe D (de 2 a 4 SM)	% de famílias na Classe C (de 4 a 10 SM)	% de famílias na Classe B (de 10 a 20 SM)	% de famílias na Classe A (mais de 20 SM)	Média da Renda Família (R\$ - out/2007)									
CF05	4,14	35,8	37,8	31,6	18,2	1,44	28,7	8,0	0,5	31,8	32,2	7,6	2,8	1948,48									
CF06	4,09	32,0	35,0	29,6	21,1	1,50	31,9	9,0	0,5	21,1	29,6	34,6	3,5	2157,43									
CF07	3,48	22,1	22,8	18,4	32,3	1,43	33,4	8,2	0,5	15,9	20,8	35,5	8,1	3152,57									
CF08	4,11	38,4	36,6	29,7	19,0	1,45	33,4	8,2	0,5	22,0	29,8	34,3	3,1	2111,51									
Difs % mdx-min												3%	9%	6%									

RESUMO DE RESULTADOS PARA AGRUPAMENTO POR ATRIBUTOS DE VIAGENS DA FAMÍLIA												CARACTERÍSTICAS DE VIAGENS DAS PESSOAS E DAS FAMÍLIAS											
Cluster (1997)	% de pessoas de 1977	% de pessoas de 1987	% de pessoas de 1997	% de pessoas de 2007	Nº de registros	Nº médio de viagens por pessoa	Nº médio de viagens por família	Média da distância total da pessoa (m)	Média da distância total da família (m)	Média da distância viagem da pessoa (m)	Média da distância viagem da família (m)	Distância media por total da viagem da pessoa (min)	Média da duração total da viagem da pessoa (min)	Média da duração total da viagem da família (min)	Duração media de viagens da pessoa (min)	Duração media de viagens da família (min)	Duração media de viagens da pessoa (min)	Duração media de viagens da família (min)					
CF09	0	0	100	0	49400	2,71	7,12	30,24	3887,21	85,24	22,62	1,45	31,45	1,45	31,45	1,45	31,45	1,45	31,45	1,45	31,45		
CF10	0	0	100	0	47294	2,71	7,12	30,24	3887,21	85,24	22,62	1,45	31,45	1,45	31,45	1,45	31,45	1,45	31,45	1,45	31,45		
CF11	0	0	100	0	53042	2,83	7,09	30,67	3944,79	84,88	24,95	1,45	31,45	1,45	31,45	1,45	31,45	1,45	31,45	1,45	31,45		
CF12	0	0	100	0	49884	2,40	6,34	11017,37	29087,50	4590,57	4587,21	92,49	24,18	38,54	2,40	38,54	2,40	38,54	2,40	38,54			
Diferenças % entre mínimos e máximos												15%	11%	11%	23%	16%	29%	29%	29%	29%	29%	29%	
Cluster (1997)												CARACTERÍSTICAS DE VIAGENS											
	Nº de viagens	Nº de pessoas	Nº de famílias	% de viagens realizadas a pé	% de viagens realizadas por transporte individual	% de viagens realizadas por transporte coletivo	% de viagens realizadas por trem	% de viagens realizadas por ônibus de linha	% de viagens realizadas por ônibus escolar / fretado	% de viagens realizadas por trabalho	% de viagens realizadas por motivo educação	% de viagens realizadas por motivo servir passageiro	% de viagens realizadas motivo manutenção/ compras	% de viagens realizadas motivo lazer outros									
CF09	41104	38226	23506	6424	29,9	39,9	30,2	0,9	12,4	75,6	39,7	23,9	13,1	16,0	16,0	51,4	495,02	495,02	13,9	13,9	13,9	13,9	
CF10	38226	24365	23667	7240	35,5	28,9	35,6	2,5	5,6	79,7	8,9	41,4	26,8	6,3	6,3	1028,99	1028,99	28,0	28,0	28,0	28,0		
CF11	45399	38805	27242	6887	30,7	43,5	25,8	0,8	21,6	70,0	5,6	40,7	21,3	12,2	12,2	30,5	411,11	411,11	17,9	17,9	17,9	17,9	
CF12	38805	38805	27242	6887	40,8	23,8	35,4	6,3	4,3	77,9	8,3	39,0	30,2	9,9	9,9	30,5	412,23	412,23	14,6	14,6	14,6	14,6	
Diferenças % entre mínimos e máximos												27%	45%	87%	37%	29%	21%	22%	22%	22%	22%	22%	
Cluster (1997)												CARACTERÍSTICAS DE VIAGENS											
	Nº de viagens	Nº de pessoas	Nº de famílias	% de viagens realizadas a pé	% de viagens realizadas por transporte individual	% de viagens realizadas por transporte coletivo	% de viagens realizadas por trem	% de viagens realizadas por ônibus de linha	% de viagens realizadas por ônibus escolar / fretado	% de viagens realizadas por trabalho	% de viagens realizadas por motivo educação	% de viagens realizadas por motivo servir passageiro	% de viagens realizadas motivo manutenção/ compras	% de viagens realizadas motivo lazer outros									
CF09	41104	38226	23506	6424	29,9	39,9	30,2	0,9	12,4	75,6	39,7	23,9	13,1	16,0	16,0	51,4	495,02	495,02	13,9	13,9	13,9	13,9	
CF10	38226	24365	23667	7240	35,5	28,9	35,6	2,5	5,6	79,7	8,9	41,4	26,8	6,3	6,3	1028,99	1028,99	28,0	28,0	28,0	28,0		
CF11	45399	38805	27242	6887	30,7	43,5	25,8	0,8	21,6	70,0	5,6	40,7	21,3	12,2	12,2	30,5	411,11	411,11	17,9	17,9	17,9	17,9	
CF12	38805	38805	27242	6887	40,8	23,8	35,4	6,3	4,3	77,9	8,3	39,0	30,2	9,9	9,9	30,5	412,23	412,23	14,6	14,6	14,6	14,6	
Diferenças % entre mínimos e máximos												27%	45%	87%	37%	29%	21%	22%	22%	22%	22%	22%	
Cluster (1997)												CARACTERÍSTICAS DE PESSOAS											
	% de pessoas do sexo feminino	% de pessoas do sexo masculino	Média de idade (anos)	% de pessoa responsável	% de cônjuges	% filhos / enteados	% de outros parentes / agregados	% de empregados	% de pessoas com alfabetização não fundamental ou com fundamental incompleto	% de pessoas com fundamental completo ou médio incompleto	% de pessoas com médio completo ou superior incompleto	% de pessoas com superior completo	% de pessoas que estudam	% de pessoas que trabalham	Média da Renda Individual (R\$ - out/2007)								
CF09	52,8	47,2	31,19	27,3	19,5	41,0	10,7	1,3	54,9	17,8	17,3	10,0	29,6	43,0	776,86	776,86	495,02	495,02	495,02	495,02	495,02		
CF10	51,0	49,0	28,06	30,6	19,5	43,7	10,4	2,0	56,4	17,7	12,6	4,3	30,5	41,3	1916,28	1916,28	1028,99	1028,99	1028,99	1028,99	1028,99		
CF11	54,3	45,7	34,87	30,6	19,5	43,7	11,8	0,5	44,0	15,1	20,9	16,0	28,0	41,3	3363,68	3363,68	411,11	411,11	411,11	411,11	411,11		
CF12	50,5	49,5	26,85	25,3	19,4	45,3	9,8	0,2	73,6	15,0	9,0	2,0	30,5	41,3	1626,19	1626,19	412,23	412,23	412,23	412,23	412,23		
Difs % mês-min	7%	8%	23%	17%	15%	20%	17%	17%	40%	21%	21%	21%	21%	21%	52%	52%	52%	52%	52%	52%	52%		
Cluster (1997)												CARACTERÍSTICAS DE FAMÍLIAS											
	Tamanho médio da família	% de famílias com presença de criança entre 0 e 4 anos	% de famílias com presença de criança entre 5 e 9 anos	% de famílias com presença de criança entre 10 e 14 anos	% de famílias com presença de idosos com 60 anos ou mais	Média da quantidade de trabalhadores (as) na família	% de famílias que têm 1 automóvel	% de famílias que têm 2 ou mais automóveis	Média da quantidade de autos na família	% de famílias na Classe E (até 2 SM)	% de famílias na Classe D (de 2 a 4 SM)	% de famílias na Classe C (de 4 a 10 SM)	% de famílias na Classe B (de 10 a 20 SM)	% de famílias na Classe A (mais de 20 SM)	Média da Renda Família (R\$ - out/2007)								
CF09	3,66	21,5	23,1	25,8	30,0	19,5	1,60	1,60	6,30	14,056,80	33901,95	5381,26	17,7	25,4	9,5	2842,61	2842,61	495,02	495,02	495,02	495,02	495,02	
CF10	3,87	26,8	28,8	30,0	30,0	19,5	1,60	1,60	6,30	14563,08	36484,03	5802,02	16,0	27,5	8,7	1916,28	1916,28	1028,99	1028,99	1028,99	1028,99	1028,99	
CF11	3,27	15,1	16,6	19,2	36,4	19,5	1,54	1,54	6,36	3079,54	24004,40	3816,28	10,7	24,1	4,1	3363,68	3363,68	411,11	411,11	411,11	411,11	411,11	
CF12	3,96	30,2	32,2	32,2	32,2	18,7	1,47	1,47	6,34	14007,21	34705,45	5689,58	7,6	22,7	6,3	1626,19	1626,19	412,23	412,23	412,23	412,23	412,23	
Difs % mês-min	17%	8%	49%	40%	5%	8%	9%	9%	3%	34%	34%	34%	34%	34%	34%	52%	52%	52%	52%	52%	52%		
Cluster (2007)												CARACTERÍSTICAS DE VIAGENS DAS PESSOAS E DAS FAMÍLIAS											
	Nº de viagens	Nº de pessoas	Nº de famílias	% de viagens realizadas a pé	% de viagens realizadas por transporte individual	% de viagens realizadas por transporte coletivo	% de viagens realizadas por trem	% de viagens realizadas por ônibus de linha	% de viagens realizadas por ônibus escolar / fretado	% de viagens realizadas por trabalho	% de viagens realizadas por motivo educação	% de viagens realizadas por motivo servir passageiro	% de viagens realizadas motivo manutenção/ compras	% de viagens realizadas motivo lazer outros									
CF13	41909	36977	22712	7025	25,74	41,21	33,05	2,36	7,67	67,88	13,95	24,96	21,77	7,77	14,76	3274,09	3274,09	495,02	495,02	495,02	495,02	495,02	
CF14	35977	36977	22712	7025	46,80	35,40	31,66	20,87	37,68	54,46	12,70	25,45	22,75	6,45	12,00	3274,09	3274,09	495,02	495,02	495,02	495,02	495,02	
CF15	55980	55980	26379	10147	26,69	48,33	38,47	19,94	27,76	59,50	13,72	24,08	22,29	6,45	12,23	3274,09	3274,09	495,02	495,02	495,02	495,02	495,02	
CF16	35799	35799	22807	6797	35,52	30,26	34,22	7,39	4,03	67,57	18,89	43,28	27,70	6,45	10,99	3274,09	3274,09	495,02	495,02	495,02	495,02	495,02	
Difs % mês-min	28%	40%	40%	29%	37%	37%	37%	68%	37%	34%	34%	34%	34%	34%	34%	46%	46%	46%	46%	46%	46%	46%	
Cluster (2007)												CARACTERÍSTICAS DE PESSOAS											
	% de pessoas do sexo feminino	% de pessoas do sexo masculino	Média de idade (anos)	% de pessoa responsável	% de cônjuges	% filhos / enteados	% de outros parentes / agregados	% de empregados	% de pessoas com alfabetização não fundamental ou com fundamental incompleto	% de pessoas com fundamental completo ou médio incompleto	% de pessoas com médio completo ou superior incompleto	% de pessoas com superior completo	% de pessoas que estudam	% de pessoas que trabalham	Média da Renda Individual (R\$ - out/2007)								
CF13	53,60	46,40	36,87	33,04	20,67	35,00	9,76	0,07	54,46	17,27	19,69	18,40	25,59	53,31	532,10	532,10	495,02	495,02	495,02	495,02	495,02		
CF14	53,20	46,80	35,40	31,66	20,87	37,68	9,76	0,17	44,10	16,23	26,39	26,39	27,37	47,38	498,15	498,15	495,02	495,02	495,02	495,02	495,02		
CF15	55,78	44,22	41,32	38,47	19,94	27,76	11,07	2,62	25,92	11,98	34,25	34,25	35,71	48,70	48,70	48,70	495,02	495,02	495,02	495,02	495,02		
CF16	51,92	48,08	33,78	30,77	20,93	39,08	9,14	1,43	46,60	16,93	28,36	8,11	7,76	53,31	42,43	387,72	387,72	495,02	495,02				

ANEXO F – Resultados das regressões logísticas para atributos de viagens da pessoa – todas variáveis

Chamada da função:

```
mlogit ANO PESS_DIST_TOT PESS_DIST_MED PESS_DURACAO_TOT PESS_DURACAO_MED
PESS_MODO_ONIBUS PESS_MODO_DIRIG PESS_MODO_PASS PESS_MODO_TREM
PESS_MODO_MOTO PESS_MODO_BICI PESS_MODO_APE PESS_MODO_OUTROS PESS_NO_MODOS
PESS_MOTIVO_TRAB PESS_MOTIVO_EDUC PESS_MOTIVO_RES PESS_MOTIVO_SERV_PAS
PESS_MOTIVO_MANUT_COMPRAIS PESS_MOTIVO_LAZER_OUTROS PESS_NO_MOTIVOS
PESS_PER_MADRUG PESS_PER_COM_MAN PESS_PER_MANHA PESS_PER_MEIODIA
PESS_PER_TARDE PESS_PER_COM NOI PESS_PER_NOITE PESS_NO_PERIODOS TOT_VIAG, base(4)
```

Número de observações = 409026

LR chi2(84) = 33814,11775391

Prob > chi2 = 0

Log likelihood = -548958

Pseudo R2 (McFadden) = 0,0299

Output (betas) – ANO = 1

Coefficientes:	Estimador	Desvio Padrão	Pr(> z)	Sig
Intercepto	0,178	0,0047	0,000	***
PESS_DIST_TOT	0,420	0,0204	0,000	***
PESS_DIST_MED	-0,237	0,0193	0,000	***
PESS_DURACAO_TOT	-0,172	0,0227	0,000	***
PESS_DURACAO_MED	-0,040	0,0209	0,058	*
PESS_MODO_ONIBUS	-0,106	0,0123	0,000	***
PESS_MODO_DIRIG	-0,405	0,0158	0,000	***
PESS_MODO_PASS	-0,132	0,0097	0,000	***
PESS_MODO_TREM	-0,260	0,0068	0,000	***
PESS_MODO_MOTO	-0,206	0,0074	0,000	***
PESS_MODO_BICI	-0,091	0,0050	0,000	***
PESS_MODO_APE	-0,243	0,0143	0,000	***
PESS_MODO_OUTROS	0,016	0,0083	0,059	*
PESS_NO_MODOS	0,103	0,0106	0,000	***
PESS_MOTIVO_TRAB	1,505	0,2111	0,000	***
PESS_MOTIVO_EDUC	1,114	0,1497	0,000	***
PESS_MOTIVO_RES	2,072	0,2250	0,000	***
PESS_MOTIVO_SERV_PAS	0,439	0,1148	0,000	***
PESS_MOTIVO_MANUT_COMPRAIS	0,568	0,0929	0,000	***
PESS_MOTIVO_LAZER_OUTROS	1,327	0,1727	0,000	***
PESS_NO_MOTIVOS	-0,492	0,0161	0,000	***
PESS_PER_MADRUG	-0,157	0,0329	0,000	***
PESS_PER_COM_MAN	-0,524	0,1104	0,000	***
PESS_PER_MANHA	-0,512	0,1002	0,000	***
PESS_PER_MEIODIA	-0,647	0,1051	0,000	***
PESS_PER_TARDE	-0,495	0,1050	0,000	***
PESS_PER_COM NOI	-0,754	0,1379	0,000	***
PESS_PER_NOITE	-0,306	0,0551	0,000	***
PESS_NO_PERIODOS	-0,646	0,0168	0,000	***
TOT_VIAG	omitted			

Output (betas) – ANO = 2				
Coefficientes:	Estimador	Desvio Padrão	Pr(> z)	Sig
Intercepto	-0,264	0,0683	0,000	***
PESS_DIST_TOT	0,289	0,0221	0,000	***
PESS_DIST_MED	-0,162	0,0202	0,000	***
PESS_DURACAO_TOT	-0,324	0,0241	0,000	***
PESS_DURACAO_MED	0,054	0,0216	0,012	**
PESS_MODO_ONIBUS	0,104	0,0113	0,000	***
PESS_MODO_DIRIG	-0,171	0,0149	0,000	***
PESS_MODO_PASS	-0,084	0,0094	0,000	***
PESS_MODO_TREM	-0,034	0,0060	0,000	***
PESS_MODO_MOTO	-0,076	0,0048	0,000	***
PESS_MODO_BICI	-0,060	0,0050	0,000	***
PESS_MODO_APE	-0,028	0,0134	0,038	**
PESS_MODO_OUTROS	0,097	0,0079	0,000	***
PESS_NO_MODOS	-0,073	0,0108	0,000	***
PESS_MOTIVO_TRAB	0,021	0,2350	0,927	-
PESS_MOTIVO_EDUC	0,164	0,1667	0,325	-
PESS_MOTIVO_RES	0,406	0,2505	0,105	-
PESS_MOTIVO_SERV_PAS	-3,121	0,3944	0,100	-
PESS_MOTIVO_MANUT_COMPRA	0,013	0,1034	0,903	-
PESS_MOTIVO_LAZER_OUTROS	0,125	0,1923	0,517	-
PESS_NO_MOTIVOS	-0,422	0,0167	0,000	***
PESS_PER_MADRUG	0,005	0,0366	0,884	-
PESS_PER_COM_MAN	0,175	0,1230	0,155	-
PESS_PER_MANHA	0,100	0,1116	0,370	-
PESS_PER_MEIODIA	-0,057	0,1171	0,624	-
PESS_PER_TARDE	0,151	0,1170	0,197	-
PESS_PER_COM NOI	0,137	0,1536	0,373	-
PESS_PER_NOITE	-0,008	0,0614	0,901	-
PESS_NO_PERIODOS	-0,301	0,0173	0,000	***
TOT_VIAG	<i>omitted</i>			

Output (betas) – ANO = 3				
Coefficientes:	Estimador	Desvio Padrão	Pr(> z)	Sig
Intercepto	0,102	0,0048	0,000	***
PESS_DIST_TOT	0,074	0,0221	0,001	***
PESS_DIST_MED	-0,216	0,0205	0,000	***
PESS_DURACAO_TOT	-0,160	0,0223	0,000	***
PESS_DURACAO_MED	0,153	0,0200	0,000	***
PESS_MODO_ONIBUS	-0,008	0,0117	0,490	-
PESS_MODO_DIRIG	-0,142	0,0152	0,000	***
PESS_MODO_PASS	-0,051	0,0095	0,000	***
PESS_MODO_TREM	-0,064	0,0063	0,000	***
PESS_MODO_MOTO	-0,090	0,0054	0,000	***
PESS_MODO_BICI	-0,033	0,0044	0,000	***
PESS_MODO_APE	-0,075	0,0139	0,000	***
PESS_MODO_OUTROS	0,058	0,0083	0,000	***
PESS_NO_MODOS	0,051	0,0107	0,000	***
PESS_MOTIVO_TRAB	0,052	0,2304	0,821	-
PESS_MOTIVO_EDUC	0,066	0,1634	0,685	-
PESS_MOTIVO_RES	0,096	0,2455	0,696	-
PESS_MOTIVO_SERV_PAS	0,005	0,1252	0,970	-
PESS_MOTIVO_MANUT_COMPRA	-0,007	0,1014	0,948	-
PESS_MOTIVO_LAZER_OUTROS	0,061	0,1885	0,746	-
PESS_NO_MOTIVOS	-0,267	0,0159	0,000	***
PESS_PER_MADRUG	-0,013	0,0359	0,718	-
PESS_PER_COM_MAN	0,014	0,1205	0,908	-
PESS_PER_MANHA	0,007	0,1094	0,950	-
PESS_PER_MEIODIA	0,025	0,1147	0,830	-
PESS_PER_TARDE	0,075	0,1147	0,515	-
PESS_PER_COM NOI	0,079	0,1506	0,601	-
PESS_PER_NOITE	0,042	0,0602	0,482	-
PESS_NO_PERIODOS	-0,072	0,0170	0,000	***
TOT_VIAG	omitted			

Output (betas) – ANO = 4				
Referência				
*** significância de 0,01				
** significância de 005				
* significância de 0,10				
- não significante				

*** significância de 0,01

** significância de 005

* significância de 0,10

- não significante

ANEXO G – Resultados das regressões quasi-poisson

Mulher, pessoa responsável, com mais de 14 anos

Fórmula do modelo (formula_atual):

TOT_VIAG ~ REN_IND + GRAU_INSTR + FAIXA_REN_FAM + PRESENCA_AUTO2 + PRESENCA_FILH_ate4 + PRESENCA_FILH_5a9 + PRESENCA_FILH_10a14

Chamada da função:

glm(family = 'quasipoisson', formula=formula.atual)

Número de Observações:

25076

Filtros:

F_PESS=1

SEXO=1

SIT_FAM=1

FAIXA_ETÁRIA>2

Variáveis	Estimador	Desvio Padrão	t valor	Pr(> t)	Sig
Intercesto	-0,004	0,0279	0	0,8818	-
REN_IND	3,86E-05	2,93E-06	13	0,0000	***
GRAU_INSTRFundamental completo / Médio incompleto	0,355	0,0190	19	0,0000	***
GRAU_INSTRMédio completo / Superior incompleto	0,524	0,0166	32	0,0000	***
GRAU_INSTRSuperior completo	0,720	0,0183	39	0,0000	***
FAIXA_REN_FAMClasse E	0,066	0,0301	2	0,0283	**
FAIXA_REN_FAMClasse D	0,127	0,0140	9	0,0000	***
FAIXA_REN_FAMClasse C	0,095	0,0284	3	0,0008	***
FAIXA_REN_FAMClasse B	0,130	0,0314	4	0,0000	***
FAIXA_REN_FAMClasse A	0,069	0,0402	2	0,0855	*
PRESERCA_AUTO2	0,043	0,0224	2	0,0564	*
PRESERCA_FILH_ate4TRUE	0,100	0,0186	5	0,0000	***
PRESERCA_FILH_5a9TRUE	0,276	0,0164	17	0,0000	***
PRESERCA_FILH_10a14TRUE	0,057	0,0062	9	0,0000	***

Homem, pessoa responsável, com mais de 14 anos

Fórmula do modelo (formula_atual):

TOT_VIAG ~ REN_IND + GRAU_INSTR + FAIXA_REN_FAM + PRESENCA_AUTO2 + PRESENCA_FILH_ate4 + PRESENCA_FILH_5a9 + PRESENCA_FILH_10a14

Chamada da função:

glm(family = 'quasipoisson', formula=formula.atual)

Número de Observações:

86871

Filtros:

F_PESS=1

SEXO=0

SIT_FAM=1

FAIXA_ETÁRIA>2

Variáveis	Estimador	Desvio Padrão	t valor	Pr(> t)	Sig
Intercesto	0,330	0,0133	25	0,0000	***
REN_IND	2,49E-05	7,51E-07	33	0,0000	***
GRAU_INSTRFundamental completo / Médio incompleto	0,151	0,0080	19	0,0000	***
GRAU_INSTRMédio completo / Superior incompleto	0,236	0,0074	32	0,0000	***
GRAU_INSTRSuperior completo	0,228	0,0086	26	0,0000	***
FAIXA_REN_FAMClasse E	-0,022	0,0158	-1	0,1560	-
FAIXA_REN_FAMClasse D	0,131	0,0290	4	0,0000	***
FAIXA_REN_FAMClasse C	0,233	0,0133	17	0,0000	***
FAIXA_REN_FAMClasse B	0,333	0,0141	24	0,0000	***
FAIXA_REN_FAMClasse A	0,286	0,0164	17	0,0000	***
PRESERCA_AUTO2	0,069	0,0078	9	0,0000	***
PRESERCA_FILH_ate4TRUE	0,140	0,0060	23	0,0000	***
PRESERCA_FILH_5a9TRUE	0,105	0,0061	17	0,0000	***
PRESERCA_FILH_10a14TRUE	0,123	0,0155	8	0,0000	***

*** nível de significância de 0,01 | ** nível de significância de 0,05 | * nível de significância de 0,10 | - não significante

Mulher, cônjuge, com mais de 14 anos

Fórmula do modelo (formula_atual):

TOT_VIAG ~ REN_IND + GRAU_INSTR + FAIXA_REN_FAM + PRESENCA_AUTO2 + PRESENCA_FILH_ate4 + PRESENCA_FILH_5a9 + PRESENCA_FILH_10a14

Chamada da função:

glm(family = 'quasipoisson', formula=formula.atual)

Número de Observações:

76946

Filtros:

F_PESS=1

SEXO=1

SIT_FAM=2

FAIXA_ETÁRIA>2

Variáveis	Estimador	Desvio Padrão	t valor	Pr(> t)	Sig
Intercesto	-0,228	0,0203	-11	0,0000	***
REN_IND	5,62E-05	2,38E-06	24	0,0000	***
GRAU_INSTRFundamental completo / Médio incompleto	0,355	0,0125	28	0,0000	***
GRAU_INSTRMédio completo / Superior incompleto	0,540	0,0116	46	0,0000	***
GRAU_INSTRSuperior completo	0,706	0,0139	51	0,0000	***
FAIXA_REN_FAMClasse E	-0,134	0,0251	-5	0,0000	***
FAIXA_REN_FAMClasse D	0,047	0,0212	2	0,0278	**
FAIXA_REN_FAMClasse C	0,110	0,0199	6	0,0000	***
FAIXA_REN_FAMClasse B	0,183	0,0208	9	0,0000	***
FAIXA_REN_FAMClasse A	0,237	0,0224	11	0,0000	***
PRESERCA_AUTO2	0,178	0,0117	15	0,0000	***
PRESERCA_FILH_ate4TRUE	-0,026	0,0095	-3	0,0067	***
PRESERCA_FILH_5a9TRUE	0,366	0,0091	40	0,0000	***
PRESERCA_FILH_10a14TRUE	0,141	0,0094	15	0,0000	***

Homem, cônjuge, com mais de 14 anos

Fórmula do modelo (formula_atual):

TOT_VIAG ~ REN_IND + GRAU_INSTR + FAIXA_REN_FAM + PRESENCA_AUTO2 + PRESENCA_FILH_ate4 + PRESENCA_FILH_5a9 + PRESENCA_FILH_10a14

Chamada da função:

glm(family = 'quasipoisson', formula=formula.atual)

Número de Observações:

3136

Filtros:

F_PESS=1

SEXO=0

SIT_FAM=2

FAIXA_ETÁRIA>2

Variáveis	Estimador	Desvio Padrão	t valor	Pr(> t)	Sig
Intercesto	-0,039	0,1029	0	0,7079	-
REN_IND	2,72E-05	6,59E-06	4	0,0000	***
GRAU_INSTRFundamental completo / Médio incompleto	0,169	0,0520	3	0,0012	***
GRAU_INSTRMédio completo / Superior incompleto	0,313	0,0459	7	0,0000	***
GRAU_INSTRSuperior completo	0,329	0,0545	6	0,0000	***
FAIXA_REN_FAMClasse E	0,087	0,1204	1	0,4697	-
FAIXA_REN_FAMClasse D	0,161	0,1060	2	0,1279	-
FAIXA_REN_FAMClasse C	0,347	0,1026	3	0,0007	***
FAIXA_REN_FAMClasse B	0,405	0,1071	4	0,0002	***
FAIXA_REN_FAMClasse A	0,308	0,1163	3	0,0082	***
PRESERCA_AUTO2	0,156	0,0464	3	0,0008	***
PRESERCA_FILH_ate4TRUE	0,091	0,0424	2	0,0326	**
PRESERCA_FILH_5a9TRUE	0,181	0,0407	4	0,0000	***
PRESERCA_FILH_10a14TRUE	0,146	0,0379	4	0,0001	***

*** nível de significância de 0,01 | ** nível de significância de 0,05 | * nível de significância de 0,10 | - não significante

Mulher, filha ou enteada, com mais de 14 anos

Fórmula do modelo (formula_atual):

TOT_VIAG ~ REN_IND + GRAU_INSTR + FAIXA_REN_FAM + PRESENCA_AUTO2 + PRESENCA_FILH_ate4 + PRESENCA_FILH_5a9 + PRESENCA_FILH_10a14

Chamada da função:

glm(family = 'quasipoisson', formula=formula.atual)

Número de Observações:

35409

Filtros:

F_PESS=1

SEXO=1

SIT_FAM=3

FAIXA_ETÁRIA>2

Variáveis	Estimador	Desvio Padrão	t valor	Pr(> t)	Sig
Intercepto	0,369	0,0200	18	0,0000	***
REN_IND	6,46E-05	2,48E-06	26	0,0000	***
GRAU_INSTRFundamental completo / Médio incompleto	0,246	0,0107	23	0,0000	***
GRAU_INSTRMédio completo / Superior incompleto	0,214	0,0111	19	0,0000	***
GRAU_INSTRSuperior completo	0,139	0,0148	9	0,0000	***
FAIXA_REN_FAMClasse E	-0,111	0,0251	-4	0,0000	***
FAIXA_REN_FAMClasse D	0,074	0,0204	4	0,0003	***
FAIXA_REN_FAMClasse C	0,197	0,0188	10	0,0000	***
FAIXA_REN_FAMClasse B	0,277	0,0195	14	0,0000	***
FAIXA_REN_FAMClasse A	0,319	0,0211	15	0,0000	***
PRESSENCA_AUTO2	0,064	0,0106	6	0,0000	***
PRESSENCA_FILH_ate4TRUE	-0,094	0,0116	-8	0,0000	***
PRESSENCA_FILH_5a9TRUE	0,051	0,0102	5	0,0000	***
PRESSENCA_FILH_10a14TRUE	0,053	0,0087	6	0,0000	***

Homem, filho ou enteado, com mais de 14 anos

Fórmula do modelo (formula_atual):

TOT_VIAG ~ REN_IND + GRAU_INSTR + FAIXA_REN_FAM + PRESENCA_AUTO2 + PRESENCA_FILH_ate4 + PRESENCA_FILH_5a9 + PRESENCA_FILH_10a14

Chamada da função:

glm(family = 'quasipoisson', formula=formula.atual)

Número de Observações:

37516

Filtros:

F_PESS=1

SEXO=0

SIT_FAM=3

FAIXA_ETÁRIA>2

Variáveis	Estimador	Desvio Padrão	t valor	Pr(> t)	Sig
Intercepto	0,439	0,0198	22	0,0000	***
REN_IND	5,24E-05	1,94E-06	27	0,0000	***
GRAU_INSTRFundamental completo / Médio incompleto	0,190	0,0099	19	0,0000	***
GRAU_INSTRMédio completo / Superior incompleto	0,133	0,0106	13	0,0000	***
GRAU_INSTRSuperior completo	0,004	0,0161	0	0,8108	-
FAIXA_REN_FAMClasse E	-0,139	0,0260	-5	0,0000	***
FAIXA_REN_FAMClasse D	0,094	0,0209	4	0,0000	***
FAIXA_REN_FAMClasse C	0,244	0,0193	13	0,0000	***
FAIXA_REN_FAMClasse B	0,336	0,0198	17	0,0000	***
FAIXA_REN_FAMClasse A	0,346	0,0214	16	0,0000	***
PRESSENCA_AUTO2	0,098	0,0102	10	0,0000	***
PRESSENCA_FILH_ate4TRUE	-0,047	0,0124	-4	0,0001	***
PRESSENCA_FILH_5a9TRUE	0,043	0,0106	4	0,0000	***
PRESSENCA_FILH_10a14TRUE	0,059	0,0087	7	0,0000	***

*** nível de significância de 0,01 | ** nível de significância de 0,05 | * nível de significância de 0,10 | - não significante

ANEXO H – Rotinas de Preparação das Bases de Dados das Pesquisas Origem-Destino do Metrô-SP (1977, 1987, 1997 e 2007)

rotina_final_1977

January 4, 2016

1 Setup Inicial

```
In [ ]: # coding: utf-8
        import math
        import logging
        import time
        import pandas as pd

        # Faz os gráficos de output ficarem um pouco melhores
        pd.set_option('display.mpl_style', 'default')
```

2 Definindo Loggers

Define loggers

Estes ‘loggers’ serão utilizados para salvar as saídas (outputs) em um arquivo de texto no diretório ‘outputs’.

ref: <http://stackoverflow.com/questions/17035077/python-logging-to-multiple-log-files-from-different-classes>

ATENÇÃO: RODAR O BLOCO ABAIXO APENAS UMA VEZ, SE ESTE BLOCO FOR EXECUTADO MAIS DE UMA VEZ OS LOGs SERÃO DUPLICADOS.

```
In [ ]: log_formatter = logging.Formatter('%(asctime)s | %(levelname)s: %(message)s')

log_output = logging.getLogger('log_output')
FH_output = logging.FileHandler(
            'outputs/1977_output.log', mode='w')
FH_output.setFormatter(log_formatter)
log_output.setLevel(logging.INFO)
log_output.addHandler(FH_output)
log_output.propagate = False

# Este logger (log_tela) loga na tela e também
# no arquivo de output, junto com o conteúdo do
# logger log_output
log_tela = logging.getLogger('log_tela')
SH_tela = logging.StreamHandler()
SH_tela.setFormatter(log_formatter)
log_tela.addHandler(SH_tela)
log_tela.addHandler(FH_output)
log_tela.setLevel(logging.INFO)
log_tela.propagate = False
```

3 Funções de 1977

3.1 Funções gerais assessórias

Função	Status
verifica_dummy	OK
verifica_range	OK
pre_processamento	OK
coordenadas	OK

```
In [ ]: log_tela.info('Definindo as funções gerais assessórias')
log_output.info('\n'=====\\n')

def verifica_dummy(df, variavel):
    """
    Verifica se uma variável, dummy, contém algum valor diferente de 0 ou de 1.
    :param df: dataframe com os dados a serem verificados
    :param variavel: string com o nome da variável (coluna) que tem os dados
        que devem ser verificados.
    :return: Sem retorno, apenas salva as avaliações nos logs.
    Uso:
        verifica_dummy(dataframe, 'coluna a ser verificada')
    """
    contador_de_erros = 0
    log_tela.info('Verificando a variável Dummy: ' + variavel)

    df_erros = df[(df[variavel] != 0) & (df[variavel] != 1)]
    if len(df_erros[variavel].value_counts()) > 0:
        log_tela.warning(variavel + ": " +
                          str(len(df_erros[variavel].value_counts())) +
                          " erros encontrados:\\n" +
                          str(df_erros[variavel].value_counts()))
        # Imprimindo as linhas com problemas no log_output
        #log_output.warning("Linhas com erro:\\n -----")
        #for index, linha in df_erros.iterrows():
        #    log_output.warning("Linha " + str(index+1) + ':\\n => ' +
        #                      str(linha))
    else:
        log_tela.info(variavel + ": Nenhum erro encontrado.")

def verifica_range(df, variavel, valor_menor, valor_maior):
    """
    Verifica se uma variável, do tipo número inteiro, contém algum valor menor
    que "valor_menor" ou maior que "valor_maior".
    :param df: dataframe com os dados a serem verificados
    :param variavel: string com o nome da variável (coluna) que tem os dados
        que devem ser verificados
    :param valor_menor: Valor mínimo esperado na variável (int ou float)
    :param valor_maior: Valor máximo esperado na variável (int ou float)
```

```

:rtype: Sem retorno, apenas salva as avaliações nos logs.

Uso:
    verifica_range(dataframe, 'nome_variavel', valor_menor, valor_maior)
"""

log_tela.info('Verificando Range da variável: ' + variavel)
# Obs: Registros inválidos: None (equivalente a NA)
log_output.info('\n\n' +
    ',      - ' + 'Mínimo esperado: ' + str(valor_menor) + '\n' +
    ',      - ' + 'Máximo esperado: ' + str(valor_maior) + '\n' +
    ',      - ' + 'Total de registros: ' + str(len(df[variavel])) + '\n' +
    ',      - ' + 'Registros nulos (NA): ' + str(df[variavel].isnull().sum()) + '\n' +
    #'      - ' + 'Descrição da variável: \n' +
    #str(df[variavel].describe())
)

df_errores = df[(df[variavel] < valor_menor) | (df[variavel] > valor_maior)]

if len(df_errores[variavel].value_counts()) > 0:

    result = df_errores[variavel].value_counts().sort_index()
    #Verificando limite inferior
    if result.first_valid_index() < valor_menor:
        log_tela.warning(
            variavel + ': ' + 'Valor inteiro mínimo encontrado: ' +
            str(result.first_valid_index()) + ' - abaixo do esperado!')
    #Verificando limite superior
    if result.last_valid_index() > valor_maior:
        log_tela.warning(
            variavel + ': ' + 'Valor inteiro máximo encontrado: ' +
            str(result.last_valid_index()) + " - acima do esperado!")

    log_tela.warning(variavel + ': ' +
                    str(len(df_errores[variavel].value_counts())) +
                    ' valor(es) incorreto(s) ' +
                    'encontrado(s) nesta variável:\n' +
                    str(df_errores[variavel].value_counts()))
    #log_output.warning("Linhas com erro:\n-----")
    #for index, linha in df_errores.iterrows():
    #    log_output.warning("Linha " + str(index+1) + ': \n => ' +
    #                      str(linha))
else:
    log_tela.info(variavel + ": Nenhum erro encontrado.")

def pre_processamento(df):
    """
    Realiza algumas ações prévias ao processamento dos dados,
    removendo alguns registros e ajustando o dataframe.
    """

log_tela.info('Criando/Renomeando colunas no dataframe principal')

```

```

log_output.info('Renomeando coluna UCOD para UCOD_DOM')
df.rename(columns={'UCOD':'UCOD_DOM'}, inplace=True)

log_tela.info('Verificando se todas as variaveis esperadas'+
              'existem no dataframe.\n' +
              'Caso não exista alguma, ela é criada.')
variaveis = ['ANO', 'CD_ENTRE', 'DIA_SEM', 'UCOD_DOM', 'ZONA_DOM',
             'SUBZONA_DOM', 'MUN_DOM', 'CO_DOM_X', 'CO_DOM_Y', 'ID_DOM',
             'F_DOM', 'FE_DOM', 'NO_DOM', 'TIPO_DOM', 'TOT_FAM', 'ID_FAM',
             'F_FAM', 'FE_FAM', 'NO_FAM', 'COND_MORA', 'QT_AUTO', 'QT_BICI',
             'QT_MOTO', 'CD_RENFAM', 'REN_FAM', 'ID_PESS', 'F_PESS',
             'FE_PESS', 'NO_PESS', 'SIT_FAM', 'IDADE', 'SEXO', 'ESTUDA',
             'GRAU_INSTR', 'OCUP', 'SETOR_ATIV', 'CD_RENIND', 'REN_IND',
             'UCOD_ESC', 'ZONA_ESC', 'SUBZONA_ESC', 'MUN_ESC', 'CO_ESC_X',
             'CO_ESC_Y', 'UCOD_TRAB1', 'ZONA_TRAB1', 'SUBZONA_TRAB1',
             'MUN_TRAB1', 'CO_TRAB1_X', 'CO_TRAB1_Y', 'UCOD_TRAB2',
             'ZONA_TRAB2', 'SUBZONA_TRAB2', 'MUN_TRAB2', 'CO_TRAB2_X',
             'CO_TRAB2_Y', 'ID_VIAG', 'F_VIAG', 'FE_VIAG', 'NO_VIAG',
             'TOT_VIAG', 'UCOD_ORIG', 'ZONA_ORIG', 'SUBZONA_ORIG',
             'MUN_ORIG', 'CO_ORIG_X', 'CO_ORIG_Y', 'UCOD_DEST', 'ZONA_DEST',
             'SUBZONA_DEST', 'MUN_DEST', 'CO_DEST_X', 'CO_DEST_Y',
             'DIST_VIAG', 'SERV_PAS_ORIG', 'SERV_PAS_DEST', 'MOTIVO_ORIG',
             'MOTIVO_DEST', 'MODO1', 'MODO2', 'MODO3', 'MODO4', 'MODO_PRIN',
             'TIPO_VIAG', 'H_SAIDA', 'MIN_SAIDA', 'ANDA_ORIG', 'H_CHEG',
             'MIN_CHEG', 'ANDA_DEST', 'DURACAO', 'TIPO_EST_AUTO',
             'VALOR_EST_AUTO']

for variavel in variaveis:
    if variavel not in df.columns:
        # Se a variável não existe no dataframe então ela é criada
        # com valor padrão de NONE (NA)
        df[variavel] = None
    log_tela.info('Criando a variavel ' + variavel + ' no dataframe')

log_output.info('Reordenando as variáveis')
df = df[['ANO', 'CD_ENTRE', 'DIA_SEM', 'UCOD_DOM', 'ZONA_DOM',
          'SUBZONA_DOM', 'MUN_DOM', 'CO_DOM_X', 'CO_DOM_Y', 'ID_DOM',
          'F_DOM', 'FE_DOM', 'NO_DOM', 'TIPO_DOM', 'TOT_FAM', 'ID_FAM',
          'F_FAM', 'FE_FAM', 'NO_FAM', 'COND_MORA', 'QT_AUTO', 'QT_BICI',
          'QT_MOTO', 'CD_RENFAM', 'REN_FAM', 'ID_PESS', 'F_PESS',
          'FE_PESS', 'NO_PESS', 'SIT_FAM', 'IDADE', 'SEXO', 'ESTUDA',
          'GRAU_INSTR', 'OCUP', 'SETOR_ATIV', 'CD_RENIND', 'REN_IND',
          'UCOD_ESC', 'ZONA_ESC', 'SUBZONA_ESC', 'MUN_ESC', 'CO_ESC_X',
          'CO_ESC_Y', 'UCOD_TRAB1', 'ZONA_TRAB1', 'SUBZONA_TRAB1',
          'MUN_TRAB1', 'CO_TRAB1_X', 'CO_TRAB1_Y', 'UCOD_TRAB2',
          'ZONA_TRAB2', 'SUBZONA_TRAB2', 'MUN_TRAB2', 'CO_TRAB2_X',
          'CO_TRAB2_Y', 'ID_VIAG', 'F_VIAG', 'FE_VIAG', 'NO_VIAG',
          'TOT_VIAG', 'UCOD_ORIG', 'ZONA_ORIG', 'SUBZONA_ORIG',
          'MUN_ORIG', 'CO_ORIG_X', 'CO_ORIG_Y', 'UCOD_DEST', 'ZONA_DEST',
          'SUBZONA_DEST', 'MUN_DEST', 'CO_DEST_X', 'CO_DEST_Y',
          'DIST_VIAG', 'SERV_PAS_ORIG', 'SERV_PAS_DEST', 'MOTIVO_ORIG',
          'MOTIVO_DEST', 'MODO1', 'MODO2', 'MODO3', 'MODO4', 'MODO_PRIN',
          'TIPO_VIAG', 'H_SAIDA', 'MIN_SAIDA', 'ANDA_ORIG', 'H_CHEG',
          'MIN_CHEG', 'ANDA_DEST', 'DURACAO', 'TIPO_EST_AUTO',
          'VALOR_EST_AUTO']]

```

```

        'VALOR_EST_AUTO']]]

log_output.info('\'\n\n=====
log_output.info('\'\n\n=====\\n')

return df

def coordenadas(passo, df):
    """
    Itera sobre os registros do dataframe coords,
    a cada iteração, utiliza o valor da coluna ZONA
    para filtrar o dataframe df em cada um dos tipos de ZONA
    (DOM, ESC, TRAB1, TRAB2, ORIG, DEST) e substitui os valores
    das coordenadas X e Y de cada tipo (CO_DOM_X, CO_ESC_Y, etc) com
    os valores das colunas CO_X e CO_Y do dataframe coords.

    :param passo: número do passo para o log
    :param df: Dafaframe a ser modificado (ex.: od1977)
    :return: retorna o dataframe modificado com todas as coordenadas aplicadas
    """
    log_tela.info("### PASSO " + str(passo) + " - COORDENADAS")

    def coord_aux(row):
        """
        Esta função irá receber uma linha (row) do dataframe coords e
        utilizar os valores desta linha para realizar as alterações
        de coordenadas no dataframe df.
        """
        # Começando pelo tipo DOM, alterando CO_DOM_X e depois CO_DOM_Y
        df.loc[df['ZONA_DOM'] == row['ZONA'], 'CO_DOM_X'] = row['CO_X']
        df.loc[df['ZONA_DOM'] == row['ZONA'], 'CO_DOM_Y'] = row['CO_Y']
        # Agora com o tipo ESC, alterando CO_ESC_X e depois CO_ESC_Y
        df.loc[df['ZONA_ESC'] == row['ZONA'], 'CO_ESC_X'] = row['CO_X']
        df.loc[df['ZONA_ESC'] == row['ZONA'], 'CO_ESC_Y'] = row['CO_Y']
        # Agora com o tipo TRAB1, alterando CO_TRAB1_X e depois CO_TRAB1_Y
        df.loc[df['ZONA_TRAB1'] == row['ZONA'], 'CO_TRAB1_X'] = row['CO_X']
        df.loc[df['ZONA_TRAB1'] == row['ZONA'], 'CO_TRAB1_Y'] = row['CO_Y']
        # Agora com o tipo TRAB2, alterando CO_TRAB2_X e depois CO_TRAB2_Y
        df.loc[df['ZONA_TRAB2'] == row['ZONA'], 'CO_TRAB2_X'] = row['CO_X']
        df.loc[df['ZONA_TRAB2'] == row['ZONA'], 'CO_TRAB2_Y'] = row['CO_Y']
        # Agora com o tipo ORIG, alterando CO_ORIG_X e depois CO_ORIG_Y
        df.loc[df['ZONA_ORIG'] == row['ZONA'], 'CO_ORIG_X'] = row['CO_X']
        df.loc[df['ZONA_ORIG'] == row['ZONA'], 'CO_ORIG_Y'] = row['CO_Y']
        # Agora com o tipo DEST, alterando CO_DEST_X e depois CO_DEST_Y
        df.loc[df['ZONA_DEST'] == row['ZONA'], 'CO_DEST_X'] = row['CO_X']
        df.loc[df['ZONA_DEST'] == row['ZONA'], 'CO_DEST_Y'] = row['CO_Y']

    log_output.info('Lendo arquivo auxiliar de Coordenadas das subzonas')
    coords = pd.read_csv('auxiliares/coord_zonas_1977.csv', sep=';', decimal=',')

    # Esta variável out não é utilizada para nada além de evitar um
    # monte de output que não será utilizado e que é gerado pelo método apply.
    out = coords.apply(coord_aux, axis=1)

```

```
log_output.info('\'\n\n=====\\n')  
return df
```

3.2 Funções Gerais

Função/Variável	Status
passo_ano	OK
passo_dia_sem	OK
passo_ucods	OK

```
In [ ]: log_tela.info('Definindo as funções Gerais')
log_output.info('\n\n=====\\n')
```

```
def passo_ano(passo, df):
    """
    Preenche a coluna "ANO" com valor 1 em todas células
    Categorias:
    /valor/ano_correspondente/
    /-----/-----
    /1/1977/
    /2/1987/
    /3/1997/
    /4/2007/

    :param passo: Número do passo atual para registro/log
    :param df: dataframe a ser modificado
    :return: retorna o dataframe modificado
    """
    log_tela.info("### PASSO " + str(passo) + " - ANO")

    # Definindo valor '1' para todas as células da coluna ANO
    df["ANO"] = 1

    return df

def passo_dia_sem(passo, df):
    """
    Não existe essa informação no banco de dados de 1977, logo, este campo será
    preenchido com 'None' (NA).

    # #####Categorias:
    # Valor/Descrição
    # -----/-----
    # 2/Segunda-Feira
    # 3/Terça-Feira
    # 4/Quarta-Feira
    # 5/Quinta-Feira
    # 6/Sexta-Feira

    :param passo: Número do passo atual para registro/log
    :param df: dataframe a ser modificado
```

```

:return: retorna o dataframe modificado
"""
log_tela.info("### PASSO " + str(passo) + " - DIA_SEM")

# Atribuindo "None" (Nulo/NA) para todos os registros
df['DIA_SEM']= None

return df

def passo_ucods(passo, df):
    """
    Itera sobre os registros do dataframe ucods,
    a cada iteração utiliza o valor da coluna ZONA_REF
    para filtrar o dataframe df em cada um dos tipos de ZONA
    (DOM, ESC, TRAB1, TRAB2, ORIG, DEST) e substitui o valor
    da respectiva UCOD.

:param passo: número do passo para o log
:param df: Dafaframe a ser modificado (ex.: od1977)
:return: retorna o dataframe modificado com todas as UCODS aplicadas
"""

log_tela.info("### PASSO " + str(passo) + " - UCODS")

def ucod_aux(row):
    df.loc[df['ZONA_DOM']==row['ZONA_REF'], 'UCOD_DOM'] = row['UCOD_BUSCADA']
    df.loc[df['ZONA_ESC']==row['ZONA_REF'], 'UCOD_ESC'] = row['UCOD_BUSCADA']
    df.loc[df['ZONA_TRAB1']==row['ZONA_REF'], 'UCOD_TRAB1'] = row['UCOD_BUSCADA']
    df.loc[df['ZONA_TRAB2']==row['ZONA_REF'], 'UCOD_TRAB2'] = row['UCOD_BUSCADA']
    df.loc[df['ZONA_ORIG']==row['ZONA_REF'], 'UCOD_ORIG'] = row['UCOD_BUSCADA']
    df.loc[df['ZONA_DEST']==row['ZONA_REF'], 'UCOD_DEST'] = row['UCOD_BUSCADA']

log_output.info('Lendo arquivo auxiliar UCOD')
ucods = pd.read_csv('auxiliares/UCOD-1977.csv', sep=';', decimal=',')

# Esta variável out não é utilizada para nada além de evitar um
# monte de output que não será utilizado e que é gerado pelo método apply.
out = ucods.apply(ucod_aux, axis=1)

# Verificando intervalo de valores - condições:
# "UCOD_XXX >= 1" E "UCOD_XXX <= 67"
for tipo in ['DOM', 'ESC', 'TRAB1', 'TRAB2', 'ORIG', 'DEST']:
    verifica_range(df, 'UCOD_' + tipo, 1, 67)
log_output.info('\n\n=====\\n')

return df

```

3.3 Funções do Domicílio

Função/Variável	Status
passo_zona_dom	OK
passo_subzona_dom	OK

Função/Variável	Status
passo_mun_dom	OK
passo_f_dom	OK
passo_fe_dom	OK
passo_tipo_dom	OK

```
In [ ]: log_tela.info('Definindo as funções do Domicílio')
log_output.info('\n=====\\n')

def passo_zona_dom(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 243

    [Teste: Checar se existe algum número < 1 ou > 243.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: passo
    :param df: dataframe
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - ZONA_DOM")

    # Substituindo valor 0 por None
    df.loc[df['ZONA_DOM']==0,'ZONA_DOM'] = None

    # Verificando intervalo de valores - condições:
    # "ZONA_DOM >= 1" E "ZONA_DOM <= 243"
    verifica_range(df, 'ZONA_DOM', 1, 243)
    log_output.info('\n=====\\n')

    return df

def passo_subzona_dom(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 633

    [Teste: Checar se existe algum número < 1 ou > 633.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - SUBZONA_DOM")
```

```

# Substituindo valor 0 por None
df.loc[df['SUBZONA_DOM']==0,'SUBZONA_DOM'] = None

# Verificando intervalo de valores - condições:
# "SUBZONA_DOM >= 1" E "SUBZONA_DOM <= 633"
verifica_range(df, 'SUBZONA_DOM', 1, 633)
log_output.info('\n\n=====\\n')

return df

def passo_mun_dom(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias
    # > 1 a 27

    [Teste: Checar se existe algum número < 1 ou > 27.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - MUN_DOM")

    # Substituindo valor 0 por None
    df.loc[df['MUN_DOM']==0,'MUN_DOM'] = None

    # Verificando intervalo de valores - condições:
    # "MUN_DOM >= 1" E "MUN_DOM <= 27"
    verifica_range(df, 'MUN_DOM', 1, 27)
    log_output.info('\n\n=====\\n')

    return df

def passo_f_dom(passo, df):
    """
    Checar se existe algum erro na coluna "F_DOM"

    # #####Categorias
    # Valor/Descrição
    # ----/----
    # 0/Demais registros
    # 1/Primeiro Registro do Domicilio

    [Teste: Checar se existe algum número diferente de 0 ou 1.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - F_DOM")

```

```

verifica_dummy(df, 'F_DOM')
log_output.info('\n\n=====\\n')

return df

def passo_fe_dom(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "FE_DOM"

    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - FE_DOM")
    log_output.info('\n\n=====\\n')

    return df

def passo_tipo_dom(passo, df):
    """
    Substituir **0** por **None (NA)**
    Substituir **2** por **0**

    # #####Categorias anteriores
    # Valor / Descrição
    # ----/----
    # 0/Não respondeu
    # 1/Particular
    # 2/Coletivo

    # #####Categorias novas
    # Valor / Descrição
    # ----/----
    # 0/Coletivo
    # 1/Particular

    [Teste: Checar se existe algum número < 0 ou > 1.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - TIPO_DOM")

    df.loc[df['TIPO_DOM']==0,'TIPO_DOM'] = None
    df.loc[df['TIPO_DOM']==2,'TIPO_DOM'] = 0

    # Verificando intervalo de valores - condições:
    # "TIPO_DOM >= 0" E "TIPO_DOM <= 1"
    verifica_dummy(df, 'TIPO_DOM')
    log_output.info('\n\n=====\\n')

```

```
return df
```

3.4 Funções da Família

Função/Variável	Status
passo_tot_fam	OK
passo_f_fam	OK
passo_fe_fam	OK
passo_cond_mora	OK
passo_qt_auto	OK
passo_qt_bici	OK
passo_qt_moto	OK
passo_ren_fam	OK
passo_cd_renfam	OK

```
In [ ]: log_tela.info('Definindo as funções da Família')
log_output.info('\n\n=====\\n')

def passo_tot_fam(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "TOT_FAM"
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - TOT_FAM")
    log_output.info('\\n\\n=====\\n')

    return df

def passo_f_fam(passo, df):
    """
    Checar se existe algum erro na coluna "F_FAM"

    # #####Categorias
    # Valor/Descrição
    # ----/----
    # 0/Demais registros
    # 1/Primeiro Registro da Família

    [Teste: Checar se existe algum número diferente de 0 ou 1.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - F_FAM")

    verifica_dummy(df, 'F_FAM')
```

```

log_output.info('\n\n=====\\n')
return df

def passo_fe_fam(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "FE_FAM"
    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - FE_FAM")
    log_output.info('\n\n=====\\n')

    return df

def passo_cond_mora(passo, df):
    """
    Substituir valores da coluna "COND_MORA"

    # * Substituir todos valores **1** por **2**
    # * Substituir todos valores **3** por **1**
    # * Substituir todos valores **4** por **3**
    # * Substituir todos valores **5** por **3**
    # * Substituir todos valores **9** por **None**
    # * Substituir todos valores **0** por **None**

    # #####Categorias anteriores
    # Valor/Descrição
    # ----/----
    # 0/Não respondeu
    # 1/Própria paga
    # 2/Própria em pagamento
    # 3/Alugada
    # 4/Cedida
    # 5/Outro
    # 9/Erro no preenchimento do banco de dados

    # #####Categorias novas
    # Valor/Descrição
    # ----/----
    # 1/Alugada
    # 2/Própria
    # 3/Outros

    [Teste: Checar se existe algum número < 1 ou > 3.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: passo
    :param df: dataframe
    :return: dataframe modificado
    """
    log_tela.info("### PASSO " + str(passo) + " - COND_MORA")

```

```

# Substituindo valor 1 por 2
df.loc[df['COND_MORA']==1, 'COND_MORA'] = 2
# Substituindo valor 3 por 1
df.loc[df['COND_MORA']==3, 'COND_MORA'] = 1
# Substituindo valor 4 por 3
df.loc[df['COND_MORA']==4, 'COND_MORA'] = 3
# Substituindo valor 5 por 3
df.loc[df['COND_MORA']==5, 'COND_MORA'] = 3
# Substituindo valor 9 por None
df.loc[df['COND_MORA']==9, 'COND_MORA'] = None
# Substituindo valor 0 por None
df.loc[df['COND_MORA']==0, 'COND_MORA'] = None

# Verificando intervalo de valores - condições:
# "COND_MORA >= 1" E "COND_MORA <= 3"
verifica_range(df, 'COND_MORA', 1, 3)
log_output.info('\n\n=====\\n')

return df

def passo_qt_auto(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "QT_AUTO"
    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - QT_AUTO")
    log_output.info('\n\n=====\\n')

    return df

def passo_qt_bici(passo, df):
    """
    Não existe essa informação no banco de dados de 1977, logo,
    este campo será preenchido com 'None'.
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: dataframe modificado
    """
    df['QT_BICI'] = None

    # Contando "QT_BICI" para verificar os valores após o procedimento
    log_tela.info('### PASSO ' + str(passo) + ' - QT_BICI' + '\n'
                  'Total de registros: ' +
                  str(len(df['QT_BICI'])) + '\n' +
                  'Soma de bicicletas "nulas": ' +
                  str(df['QT_BICI'].isnull().sum()))

    log_output.info('\n\n=====\\n')

```

```

    return df

def passo_qt_moto(passo, df):
    """
    Não existe essa informação no banco de dados de 1977, logo,
    este campo será preenchido com 'None'.
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: dataframe modificado
    """
    df['QT_MOTO'] = None

    # Contando "QT_MOTO" para verificar os valores após o procedimento
    log_tela.info('### PASSO ' + str(passo) + ' - QT_MOTO' + '\n'
                  , ' ' + 'Total de registros: ' +
                  str(len(df['QT_MOTO'])) + '\n' +
                  ' ' + 'Soma de motos "nulas": ' +
                  str(df['QT_MOTO'].isnull().sum()))

    log_output.info('\n\n=====\\n')

    return df

def passo_ren_fam(passo, df, deflator):
    """
    Corriga a renda familiar de acordo com o deflator passado na função.
    :param passo: Número do passo atual para registro/log
    :param df: dataframe
    :param deflator: Deflator utilizado para correção da renda.
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - REN_FAM")

    df['REN_FAM'] = df['REN_FAM'] * deflator

    return df

def passo_cd_renfam(passo, df):
    """
    Substituir valores da coluna "CD_RENFAM"

    * Substituir todos valores **1** por **2**
    * Substituir todos valores **3** por **2**
        Quando categoria original for 0 (respondeu)
            checar no campo REN_FAM se o valor é nulo.
        Se for nulo, manter na categoria 0 (Renda Familiar Declarada como Zero),
        senão, mover para a categoria 1 (Renda Familiar Declarada e
        Maior que Zero).

    # #####Categorias anteriores
    # Valor/Descrição
    """

```

```

# ----/----
# 0/Respondeu
# 1/Não Sabe
# 2/Não Respondeu
# 3/Não se Aplica

# #####Categorias novas
# Valor/Descrição
# ----/----
# 0/Renda Familiar Declarada como Zero
# 1/Renda Familiar Declarada e Maior que Zero
# 2/Renda Atribuída

[Teste: Checar se existe algum número < 0 ou > 2.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe corrigido
"""
log_tela.info("### PASSO " + str(passo) + " - CD_RENFAM")

# Substituindo valor 1 (Não sabe) por 2 (Atribuída)
df.loc[df['CD_RENFAM']==1,'CD_RENFAM'] = 2

# Substituindo valor 3 (Não se Aplica) por 2 (Atribuída)
df.loc[df['CD_RENFAM']==3,'CD_RENFAM'] = 2

# Correção de REN_FAM
# REN_FAM só pode ser igual a zero se tiver sido declarada assim.
df.loc[df['REN_FAM']==0,'CD_RENFAM'] = 0

# Dividindo a categoria '0', "Respondeu", em:
#   - 0 "Renda Familiar Declarada como Zero" e
#   - 1 "Renda Familiar Declarada e Maior que Zero"
df.loc[(df['CD_RENFAM'] == 0) &
       (df['REN_FAM'] != 0) &
       (df['REN_FAM'].notnull()), 'CD_RENFAM'] = 1

# Verificando intervalo de valores - condições:
# "CD_RENFAM >= 0" E "CD_RENFAM <= 2"
verifica_range(df, 'CD_RENFAM', 0, 2)
log_output.info('\n\n=====\\n')

return df

```

3.5 Funções da pessoa

Função/Variável	Status
passo_f_pess	OK
passo_fe_pess	OK
passo_sit_fam	OK
passo_idade	OK
passo_sexo	OK
passo_grau_instr	OK
passo_ocup	OK
passo_setor_ativ	OK
passo_ren_ind	OK
passo_cd_renind	OK
passo_estuda	OK

```
In [ ]: log_tela.info('Definindo as funções da Pessoa')
log_output.info('\n\n=====\\n')

def passo_f_pess(passo, df):
    """
    Checar se existe algum erro na coluna "F_PESS"

    # #####Categorias
    # Valor/Descrição
    # ----/----
    # 0/Demais registros
    # 1/Primeiro Registro da Pessoa

    [Teste: Checar se existe algum número diferente de 0 ou 1.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - F_PESS")

    verifica_dummy(df, 'F_PESS')
    log_output.info('\n\n=====\\n')

    return df

def passo_fe_pess(passo, df):
    """
    Nada há que se fazer em relação aos dados das colunas "FE_PESS"
    :param passo: Número do passo atual para registro/log
    :param df:
```

```

:return:
"""
log_tela.info("### PASSO " + str(passo) + " - FE_PESS")
log_output.info('\\n\\n=====\\n')

return df

def passo_sit_fam(passo, df):
"""
Substituir todos valores **5** por **4**
Substituir todos valores **6** por **5**
Substituir todos valores **7** por **6**
Substituir todos valores **9** por **6**
Substituir todos valores **0** por **6** quando DURACAO!=0

# #####Categorias anteriores
# Valor/Descrição
# ----/----
# 1/Chefe
# 2/Cônjuge
# 3/Filho(a)
# 4/Parente
# 5/Agregado
# 6/Empregado Residente
# 7/Visitante não Residente

# #####Categorias novas:
# Valor/Descrição
# ----/----
# 0/ Não Respondeu/Não fez viagem
# 1/ Pessoa Responsável
# 2/ Cônjuge/Companheiro(a)
# 3/ Filho(a)/Enteado(a)
# 4/ Outro Parente / Agregado
# 5/ Empregado Residente
# 6/ Outros (visitante não residente / parente do empregado)

[Teste: Checar se existe algum número < 0 ou > 6.
 Se encontrar, retornar erro indicando em qual linha.]"""

:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe modificado
"""

log_tela.info("### PASSO " + str(passo) + " - SIT_FAM")

# Substituindo valor 5 por 4
df.loc[df['SIT_FAM']==5,'SIT_FAM'] = 4
# Substituindo valor 6 por 5
df.loc[df['SIT_FAM']==6,'SIT_FAM'] = 5
# Substituindo valor 7 por 6
df.loc[df['SIT_FAM']==7,'SIT_FAM'] = 6
# Substituindo valor 9 por 6
df.loc[df['SIT_FAM']==9,'SIT_FAM'] = 6

```

```

# Substituindo valor 0 por 6 quando DURACAO != 0
df.loc[(df['SIT_FAM']==9) & (df['DURACAO']!=0), 'SIT_FAM'] = 6

# Verificando intervalo de valores - condições:
# "SIT_FAM >= 0" E "SIT_FAM <= 6"
verifica_range(df, 'SIT_FAM', 0, 6)
log_output.info('\n\n=====\\n')

return df

def passo_idade(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "IDADE"
    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - IDADE")
    log_output.info('\n\n=====\\n')

    return df

def passo_sexo(passo, df):
    """
    #####Categorias anteriores
    Valor/Descrição
    ----/----
    0/Não respondeu (-> NA)
    1/Masculino
    2/Feminino

    #####Categorias novas
    Valor/Descrição
    ----/----
    0/Masculino
    1/Feminino

    [Teste: Checar se existe algum número diferente de 0 ou 1.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe modificado
    """
    log_tela.info("### PASSO " + str(passo) + " - SEXO")

    # Substituindo valor 0 por None
    df.loc[df['SEXO']==0, 'SEXO'] = None
    # Substituindo valor 1 por 0
    df.loc[df['SEXO']==1, 'SEXO'] = 0
    # Substituindo valor 2 por 1
    df.loc[df['SEXO']==2, 'SEXO'] = 1

```

```

# Verificando intervalo de valores - condições:
# "SEXO >= 0" E "SEXO <= 1"
verifica_dummy(df, 'SEXO')
log_output.info('\'\n\'=====\'\n')

return df


def passo_grau_instr(passo, df):
    """
    Substituir valores da coluna "GRAU_INSTR"

    Substituir todos valores **2** por **1**
    Substituir todos valores **3** por **1**
    Substituir todos valores **4** por **1**
    Substituir todos valores **5** por **2**
    Substituir todos valores **6** por **2**
    Substituir todos valores **7** por **3**
    Substituir todos valores **8** por **3**
    Substituir todos valores **9** por **4**
    Substituir todos valores **0** por **None**

    #####Categorias anteriores:
    # Valor/Descrição
    # ----/----
    # 1/Sem Instrução
    # 2/Primário Incompleto
    # 3/Promocional Completo
    # 4/Ginasial Incompleto
    # 5/Ginasial Completo
    # 6/Colegial Incompleto
    # 7/Colegial Completo
    # 8/Universitário Incompleto
    # 9/Universitário Completo

    #####Categorias novas
    # Valor/Descrição
    # ----/----
    # 1/Não-Alfabetizado/Fundamental Incompleto
    # 2/Fundamental Completo/Médio Incompleto
    # 3/Médio Completo/Superior Incompleto
    # 4/Superior completo

[Teste: Checar se existe algum número < 1 ou > 4.
Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna dataframe modificado
"""

log_tela.info("### PASSO " + str(passo) + " - GRAU_INSTR")

# Substituindo valor 2 por 1
df.loc[df['GRAU_INSTR']==2, 'GRAU_INSTR'] = 1
# Substituindo valor 3 por 1

```

```

df.loc[df['GRAU_INSTR']==3,'GRAU_INSTR'] = 1
# Substituindo valor 4 por 1
df.loc[df['GRAU_INSTR']==4,'GRAU_INSTR'] = 1
# Substituindo valor 5 por 2
df.loc[df['GRAU_INSTR']==5,'GRAU_INSTR'] = 2
# Substituindo valor 6 por 2
df.loc[df['GRAU_INSTR']==6,'GRAU_INSTR'] = 2
# Substituindo valor 7 por 3
df.loc[df['GRAU_INSTR']==7,'GRAU_INSTR'] = 3
# Substituindo valor 8 por 3
df.loc[df['GRAU_INSTR']==8,'GRAU_INSTR'] = 3
# Substituindo valor 9 por 4
df.loc[df['GRAU_INSTR']==9,'GRAU_INSTR'] = 4
# Substituindo valor 0 por None
df.loc[df['GRAU_INSTR']==0,'GRAU_INSTR'] = None

# Verificando intervalo de valores - condições:
# "GRAU_INSTR >= 1" E "GRAU_INSTR <= 4"
verifica_range(df, 'GRAU_INSTR', 1, 4)
log_output.info('\n\n=====\\n')

return df

def passo_ocup(passo, df):
    """
    Substituir valores da coluna "OCUP"

    Somar 10 em todos valores (para facilitar a lógica).
    Substituir todos valores **10** por **None**
    Substituir todos valores **11** por **7**
    Substituir todos valores **12** por **6**
    Substituir todos valores **13** por **3**
    Substituir todos valores **14** por **5**
    Substituir todos valores **15** por **4**
    Substituir todos valores **16** por **2**
    Substituir todos valores *maiores do que 16* por **1**

    #####Categorias anteriores:
    # Valor/Descrição
    # ----/
    # 1/Estudante
    # 2/Prendas Domésticas
    # 3/Aposentado
    # 4/Sem Ocupação (nunca trabalhou)
    # 5/Desempregado
    # 6/Em licença
    # 7/em diante/diversas profissões

    #####Categorias novas
    # Valor/Descrição
    # ----/
    # 1/Tem trabalho
    # 2/Em licença médica
    """

    if passo == 1:
        df['OCUP'].replace(10, np.nan, inplace=True)
        df['OCUP'].replace(11, 7, inplace=True)
        df['OCUP'].replace(12, 6, inplace=True)
        df['OCUP'].replace(13, 3, inplace=True)
        df['OCUP'].replace(14, 5, inplace=True)
        df['OCUP'].replace(15, 4, inplace=True)
        df['OCUP'].replace(16, 2, inplace=True)
        df['OCUP'].replace(17, 1, inplace=True)
    else:
        df['OCUP'].replace(10, np.nan, inplace=True)
        df['OCUP'].replace(11, 7, inplace=True)
        df['OCUP'].replace(12, 6, inplace=True)
        df['OCUP'].replace(13, 3, inplace=True)
        df['OCUP'].replace(14, 5, inplace=True)
        df['OCUP'].replace(15, 4, inplace=True)
        df['OCUP'].replace(16, 2, inplace=True)
        df['OCUP'].replace(17, 1, inplace=True)

    return df

```

```

# 3/Aposentado / pensionista
# 4/Desempregado
# 5/Sem ocupação
# 6/Dona de casa
# 7/Estudante

[Teste: Checar se existe algum número < 1 ou > 7.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna dataframe modificado
"""

log_tela.info("### PASSO " + str(passo) + " - OCUP")

df['OCUP'] = df['OCUP'] + 10
# Substituindo valor 10 por None
df.loc[df['OCUP']==10,'OCUP'] = None
# Substituindo valor 11 por 7
df.loc[df['OCUP']==11,'OCUP'] = 7
# Substituindo valor 12 por 6
df.loc[df['OCUP']==12,'OCUP'] = 6
# Substituindo valor 13 por 3
df.loc[df['OCUP']==13,'OCUP'] = 3
# Substituindo valor 14 por 5
df.loc[df['OCUP']==14,'OCUP'] = 5
# Substituindo valor 15 por 4
df.loc[df['OCUP']==15,'OCUP'] = 4
# Substituindo valor 16 por 2
df.loc[df['OCUP']==16,'OCUP'] = 2
# Substituindo valor >16 por 1
df.loc[df['OCUP']>16,'OCUP'] = 1

# Verificando intervalo de valores - condições:
# "OCUP >= 1" E "OCUP <= 7"
verifica_range(df, 'OCUP', 1, 7)
log_output.info('\n\n=====\\n')

return df

def passo_setor_ativ(passo, df):
"""
Substituir valores da coluna "SETOR_ATIV"

Na coluna "SETOR_ATIV", linha i,
ler o valor da linha i da coluna "SETOR_ATIV", daí,
buscar o mesmo valor na coluna "COD" do arquivo setor_ativ-1977.csv.
Ao achar, retornar o valor da mesma linha, só que da coluna "COD_UNIF"

####Categorias anteriores
> ver arquivo .csv

####Categorias novas
Valor/Descrição

```

```

----/----
1/Agrícola
2/Construção Civil
3/Indústria
4/Comércio
5/Administração Pública
6/Serviços de Transporte
7/Doutros serviços
8/Doutros
9/Não se aplica

[Teste: Checar se existe algum número < 1 ou > 9.
Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna dataframe modificado
"""

log_tela.info("### PASSO " + str(passo) + " - SETOR_ATIV")

log_output.info('Lendo arquivo de referência externa setor_ativ-1977.csv')
df_setor = pd.read_csv('auxiliares/setor_ativ-1977.csv', sep=';', decimal=',')

def setor_aux(row):
    df.loc[df['SETOR_ATIV']==row['COD'], 'SETOR_ATIV'] = row['COD_UNIF']

# Esta variável out não é utilizada para nada além de evitar um
# monte de output que não será utilizado e que é gerado pelo método apply.
out = df_setor.apply(setor_aux, axis=1)

# Substituindo valor 0 por None
df.loc[df['SETOR_ATIV']==0, 'SETOR_ATIV'] = None

# Verificando intervalo de valores - condições:
# "SETOR_ATIV >= 1" E "SETOR_ATIV <= 9"
verifica_range(df, 'SETOR_ATIV', 1, 9)
log_output.info('\n\n=====\\n')

return df

def passo_ren_ind(passo, df, deflator):
    """
    Corriga a renda individual de acordo com o deflator passado na função.
    :param passo: Número do passo atual para registro/log
    :param df: dataframe
    :param deflator: Deflator utilizado para correção da renda.
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - REN_IND")

    df['REN_IND'] = df['REN_IND'] * deflator

    return df

```

```

def passo_cd_renind(passo, df):
    """
        Substituir **2** por None (NA)
        Substituir **3** (erro de preenchimento) por None
        Substituir **1** por **2**
        Substituir **0** por **1**
        Substituir **2** por **0**

        ##### Categorias anteriores efetivas
        Valor Real/ Descrição /Valor Layout
        -----/-----/-----
        0      /Tem renda     /1
        1      /Não tem renda/2
        2      /Não declarou /3

        ##### Categorias novas
        Valor/Descrição
        -----/-----
        0      /Não tem renda
        1      /Tem renda

        [Teste: Checar se existe algum número < 0 ou > 1.
         Se encontrar, retornar erro indicando em qual linha.]
        :param passo: Número do passo atual para registro/log
        :param df:
        :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - CD_RENIND")

    df.loc[df['CD_RENIND']==2, 'CD_RENIND'] = None
    df.loc[df['CD_RENIND']==3, 'CD_RENIND'] = None
    df.loc[df['CD_RENIND']==1, 'CD_RENIND'] = 2
    df.loc[df['CD_RENIND']==0, 'CD_RENIND'] = 1
    df.loc[df['CD_RENIND']==2, 'CD_RENIND'] = 0

    # Verificando intervalo de valores - condições:
    # "CD_RENIND >= 0" E "CD_RENIND <= 1"
    verifica_dummy(df, 'CD_RENIND')
    log_output.info('\'\n\'=====\\n')

    return df


def passo_estuda(passo, df):
    """
        Considera-se que ninguém estuda, exceto quem forneceu informação
        de localização da Escola (Zona/SubZona)

        #####
        # ATENÇÃO #
        # ESTA FUNÇÃO SÓ DEVE SER #
        # EXECUTADA APÓS O PASSO #
        # ZONA_ESC      #
    """

```

```

#####
##### Categorias novas
Valor/Descrição
----/----
0/Não estuda
1/Estuda

[Teste: Checar se existe algum número diferente de 0 ou 1.
Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna dataframe com campo ESCOLA resolvido
"""
log_tela.info("### PASSO " + str(passo) + " - ESTUDA")

df['ESTUDA'] = 0

# Substituindo todos que declararam zona escola diferente de zero
# com campo ESTUDA igual a 1.
df.loc[(df['ZONA_ESC'] != 0)&
       (df['ZONA_ESC'].notnull()), 'ESTUDA'] = 1

verifica_dummy(df, 'ESTUDA')
log_output.info('\n\n=====\\n')

return df

```

3.6 Funções da Viagem

Função/Variável	Status
passo_f_viag	OK
passo_fe_viag	OK
passo_zona_esc	OK
passo_subzona_esc	OK
passo_mun_esc	OK
passo_zona_trab1	OK
passo_subzona_trab1	OK
passo_mun_trab1	OK
passo_zona_trab2	OK
passo_subzona_trab2	OK
passo_mun_trab2	OK
passo_zona_orig	OK
passo_subzona_orig	OK
passo_mun_orig	OK
passo_zona_dest	OK
passo_subzona_dest	OK
passo_mun_dest	OK
passo_serv_pas_orig	OK
passo_serv_pas_dest	OK
passo_motivo_orig	OK
passo_motivo_dest	OK
passo_modo1	OK
passo_modo2	OK
passo_modo3	OK
passo_modo4	OK
passo_modo_prin	OK
passo_tipo_est_auto	OK
passo_valor_est_auto	OK
passo_tot_viag	OK

Obs: O passo_tot_viag deve ser executado após os passos que geram id's.

```
In [ ]: log_tela.info('Definindo funções referentes às viagens')
log_output.info('\n=====\\n')
```

```
def passo_f_viag(passo, df):
```

```

"""
Checar se existe algum erro na coluna "F_VIAG"

# #####Categorias
# Valor/Descrição
# ----/----
# 0/Não fez viagem
# 1/Fez viagem

[Teste: Checar se existe algum número diferente de 0 ou 1.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe com F_VIAG modificado
"""
log_tela.info("### PASSO " + str(passo) + " - F_VIAG")

# Setando F_VIAG=0 se DURACAO==0
df.loc[df['DURACAO']==0,'F_VIAG'] = 0

verifica_dummy(df, 'F_VIAG')
log_output.info('\n\n=====\\n')

return df


def passo_fe_viag(passo, df):
"""
# Nada há que se fazer em relação aos dados da coluna "FE_VIAG"
:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - FE_VIAG")
log_output.info('\\n\\n=====\\n')

return df


def passo_zona_esc(passo, df):
"""
Checar se existe algum erro

# #####Categorias:
# > 1 a 243

[Teste: Checar se existe algum número < 1 ou > 243.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: Sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - ZONA_ESC")

```

```

# Substituindo valor 0 por None
df.loc[df['ZONA_ESC']==0,'ZONA_ESC'] = None

# Verificando intervalo de valores - condições:
# "ZONA_ESC >= 1" E "ZONA_ESC <= 243"
verifica_range(df, 'ZONA_ESC', 1, 243)
log_output.info('\n\n=====\\n')

return df

def passo_subzona_esc(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 633

    [Teste: Checar se existe algum número < 1 ou > 633.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - SUBZONA_ESC")

    # Substituindo valor 0 por None
    df.loc[df['SUBZONA_ESC']==0,'SUBZONA_ESC'] = None

    # Verificando intervalo de valores - condições:
    # "SUBZONA_ESC >= 1" E "SUBZONA_ESC <= 633"
    verifica_range(df, 'SUBZONA_ESC', 1, 633)
    log_output.info('\\n\\n=====\\n')

    return df

def passo_mun_esc(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias
    # > 1 a 27

    [Teste: Checar se existe algum número < 1 ou > 27.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - MUN_ESC")

    # Substituindo valor 0 por None
    df.loc[df['MUN_ESC']==0,'MUN_ESC'] = None

```

```

# Verificando intervalo de valores - condições:
# "MUN_ESC >= 1" E "MUN_ESC <= 27"
verifica_range(df, 'MUN_ESC', 1, 27)
log_output.info('\n\n=====\\n')

return df

def passo_zona_trab1(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 243

    [Teste: Checar se existe algum número < 1 ou > 243.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """

    log_tela.info("### PASSO " + str(passo) + " - ZONA_TRAB1")

    # Substituindo valor 0 por None
    df.loc[df['ZONA_TRAB1']==0,'ZONA_TRAB1'] = None

    # Verificando intervalo de valores - condições:
    # "ZONA_TRAB1 >= 1" E "ZONA_TRAB1 <= 243"
    verifica_range(df, 'ZONA_TRAB1', 1, 243)
    log_output.info('\n\n=====\\n')

    return df

def passo_subzona_trab1(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 633

    [Teste: Checar se existe algum número < 1 ou > 633.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """

    log_tela.info("### PASSO " + str(passo) + " - SUBZONA_TRAB1")

    # Substituindo valor 0 por None
    df.loc[df['SUBZONA_TRAB1']==0,'SUBZONA_TRAB1'] = None

    # Verificando intervalo de valores - condições:

```

```

# "SUBZONA_TRAB1 >= 1" E "SUBZONA_TRAB1 <= 633"
verifica_range(df, 'SUBZONA_TRAB1', 1, 633)
log_output.info('\n\n=====\\n')

return df

def passo_mun_trab1(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias
    # > 1 a 27

    [Teste: Checar se existe algum número < 1 ou > 27.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - MUN_TRAB1")

    # Substituindo valor 0 por None
    df.loc[df['MUN_TRAB1']==0, 'MUN_TRAB1'] = None

    # Verificando intervalo de valores - condições:
    # "MUN_TRAB1 >= 1" E "MUN_TRAB1 <= 27"
    # trips that are not work purposed
    verifica_range(df, 'MUN_TRAB1', 1, 27)
    log_output.info('\n\n=====\\n')

    return df

def passo_zona_trab2(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 243

    [Teste: Checar se existe algum número < 1 ou > 243.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - ZONA_TRAB2")

    # Substituindo valor 0 por None
    df.loc[df['ZONA_TRAB2']==0, 'ZONA_TRAB2'] = None

    # Verificando intervalo de valores - condições:
    # "ZONA_TRAB2 >= 1" E "ZONA_TRAB2 <= 243"

```

```

verifica_range(df, 'ZONA_TRAB2', 1, 243)
log_output.info('\n\n=====\\n')

return df

def passo_subzona_trab2(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 633

    [Teste: Checar se existe algum número < 1 ou > 633.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - SUBZONA_TRAB2")

    # Substituindo valor 0 por None
    df.loc[df['SUBZONA_TRAB2']==0, 'SUBZONA_TRAB2'] = None

    # Verificando intervalo de valores - condições:
    # "SUBZONA_TRAB2 >= 1" E "SUBZONA_TRAB2 <= 633"
    verifica_range(df, 'SUBZONA_TRAB2', 1, 633)
    log_output.info('\n\n=====\\n')

    return df

def passo_mun_trab2(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias
    # > 1 a 27

    [Teste: Checar se existe algum número < 1 ou > 27.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - MUN_TRAB2")

    # Substituindo valor 0 por None
    df.loc[df['MUN_TRAB2']==0, 'MUN_TRAB2'] = None

    # Verificando intervalo de valores - condições:
    # "MUN_TRAB2 >= 1" E "MUN_TRAB2 <= 27"
    verifica_range(df, 'MUN_TRAB2', 1, 27)
    log_output.info('\n\n=====\\n')

```

```

    return df

def passo_zona_orig(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 243

    [Teste: Checar se existe algum número < 1 ou > 243.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """

    log_tela.info("### PASSO " + str(passo) + " - ZONA_ORIG")

    # Substituindo valor 0 por None
    df.loc[df['ZONA_ORIG']==0,'ZONA_ORIG'] = None

    # Verificando intervalo de valores - condições:
    # "ZONA_ORIG >= 1" E "ZONA_ORIG <= 243"
    verifica_range(df, 'ZONA_ORIG', 1, 243)
    log_output.info('\n\n=====\\n')

    return df

def passo_subzona_orig(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 633

    [Teste: Checar se existe algum número < 1 ou > 633.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """

    log_tela.info("### PASSO " + str(passo) + " - SUBZONA_ORIG")

    # Substituindo valor 0 por None
    df.loc[df['SUBZONA_ORIG']==0,'SUBZONA_ORIG'] = None

    # Verificando intervalo de valores - condições:
    # "SUBZONA_ORIG >= 1" E "SUBZONA_ORIG <= 633"
    verifica_range(df, 'SUBZONA_ORIG', 1, 633)
    log_output.info('\n\n=====\\n')

    return df

```

```

def passo_mun_orig(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias
    # > 1 a 27

    [Teste: Checar se existe algum número < 1 ou > 27.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - MUN_ORIG")

    # Substituindo valor 0 por None
    df.loc[df['MUN_ORIG']==0,'MUN_ORIG'] = None

    # Verificando intervalo de valores - condições:
    # "MUN_ORIG >= 1" E "MUN_ORIG <= 27"
    verifica_range(df, 'MUN_ORIG', 1, 27)
    log_output.info('\n\n=====\\n')

    return df


def passo_zona_dest(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 243

    [Teste: Checar se existe algum número < 1 ou > 243.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - ZONA_DEST")

    # Substituindo valor 0 por None
    df.loc[df['ZONA_DEST']==0,'ZONA_DEST'] = None

    # Verificando intervalo de valores - condições:
    # "ZONA_DEST >= 1" E "ZONA_DEST <= 243"
    verifica_range(df, 'ZONA_DEST', 1, 243)
    log_output.info('\n\n=====\\n')

    return df

```

```

def passo_subzona_dest(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 633

    [Teste: Checar se existe algum número < 1 ou > 633.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - SUBZONA_DEST")

    # Substituindo valor 0 por None
    df.loc[df['SUBZONA_DEST']==0, 'SUBZONA_DEST'] = None

    # Verificando intervalo de valores - condições:
    # "SUBZONA_DEST >= 1" E "SUBZONA_DEST <= 633"
    verifica_range(df, 'SUBZONA_DEST', 1, 633)
    log_output.info('\n\n=====\\n')

    return df


def passo_mun_dest(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias
    # > 1 a 27

    [Teste: Checar se existe algum número < 1 ou > 27.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - MUN_DEST")

    # Substituindo valor 0 por None
    df.loc[df['MUN_DEST']==0, 'MUN_DEST'] = None

    # Verificando intervalo de valores - condições:
    # "MUN_DEST >= 1" E "MUN_DEST <= 27"
    verifica_range(df, 'MUN_DEST', 1, 27)
    log_output.info('\n\n=====\\n')

    return df


def passo_serv_pas_orig(passo, df):
    """

```

Atribuir valor à variável dummy que identifica se o motivo origem é servir passageiro(a)

```
#####
#      ATENÇÃO      #
#  ESTE PASSO DEVE  #
#  SER EXECUTADO   #
#  ANTES DO PASSO  #
#      MOTIVO_ORIG  #
#####

# Atribuir 1 quando "MOTIVO_ORIG" for igual a 9

####Categorias novas
Valor/Descrição
----/----
0/Não
1/Sim

:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""

log_tela.info("### PASSO " + str(passo) + " - SERV_PAS_ORIG")

df['SERV_PAS_ORIG'] = 0

df.loc[df['MOTIVO_ORIG']==9, 'SERV_PAS_ORIG'] = 1

verifica_dummy(df, 'SERV_PAS_ORIG')
log_output.info('\n\n=====\\n')

return df


def passo_serv_pas_dest(passo, df):
    """
Atribuir valor à variável dummy que identifica se o motivo destino é servir passageiro(a)

#####
#      ATENÇÃO      #
#  ESTE PASSO DEVE  #
#  SER EXECUTADO   #
#  ANTES DO PASSO  #
#      MOTIVO_DEST   #
#####

# Atribuir 1 quando "MOTIVO_DEST" for igual a 9

####Categorias novas
Valor/Descrição
----/----
0/Não
```

```

1 /Sim

:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - SERV_PAS_DEST")

df['SERV_PAS_DEST'] = 0

df.loc[df['MOTIVO_DEST']==9, 'SERV_PAS_DEST'] = 1

verifica_dummy(df, 'SERV_PAS_DEST')
log_output.info('\n\n=====\\n')

return df

def passo_motivo_orig(passo, df):
"""
Substituir todos valores **6** por **11**
Substituir todos valores **7** por **6**
Substituir todos valores **8** por **7**
Substituir todos valores **10** por **8**
Substituir todos valores **11** por **9**
Substituir todos valores **0** por **None**

# #####Categorias anteriores
# Valor/Descrição
# ----/----
# 1/Trabalho/Indústria
# 2/Trabalho/Comércio
# 3/Trabalho/Serviços
# 4/Escola/Educação
# 5/Compras
# 6/Negócios
# 7/Médico/Dentista/Saúde
# 8/Recreação/Visitas
# 9/Servir Passageiro
# 10/Residência

# #####Categorias novas
# Valor/Descrição
# ----/----
# 1/Trabalho/Indústria
# 2/Trabalho/Comércio
# 3/Trabalho/Serviços
# 4/Educação
# 5/Compras
# 6/Saúde
# 7/Lazer
# 8/Residência
# 9/Outros

```

```

[Teste: Checar se existe algum número < 1 ou > 9.
Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: dataframe modificado
"""
log_tela.info("### PASSO " + str(passo) + " - MOTIVO_ORIG")

# Substituindo valor 6 por 11
df.loc[df['MOTIVO_ORIG']==6,'MOTIVO_ORIG'] = 11
# Substituindo valor 7 por 6
df.loc[df['MOTIVO_ORIG']==7,'MOTIVO_ORIG'] = 6
# Substituindo valor 8 por 7
df.loc[df['MOTIVO_ORIG']==8,'MOTIVO_ORIG'] = 7
# Substituindo valor 10 por 8
df.loc[df['MOTIVO_ORIG']==10,'MOTIVO_ORIG'] = 8
# Substituindo valor 11 por 9
df.loc[df['MOTIVO_ORIG']==11,'MOTIVO_ORIG'] = 9
# Substituindo valor 0 por None
df.loc[df['MOTIVO_ORIG']==0,'MOTIVO_ORIG'] = None

# Verificando intervalo de valores - condições:
# "MOTIVO_ORIG >= 1" E "MOTIVO_ORIG <= 9"
verifica_range(df, 'MOTIVO_ORIG', 1, 9)
log_output.info('\n\n=====\\n')

return df

def passo_motivo_dest(passo, df):
"""
Substituir todos valores **6** por **11**
Substituir todos valores **7** por **6**
Substituir todos valores **8** por **7**
Substituir todos valores **10** por **8**
Substituir todos valores **11** por **9**
Substituir todos valores **0** por **None**

# #####Categorias anteriores
# Valor/Descrição
# ----/
# 1/Trabalho/Indústria
# 2/Trabalho/Comércio
# 3/Trabalho/Serviços
# 4/Escola/Educação
# 5/Compras
# 6/Negócios
# 7/Médico/Dentista/Saúde
# 8/Recreação/Visitas
# 9/Servir Passageiro
# 10/Residência

# #####Categorias novas
# Valor/Descrição
"""

```

```

# ----/----
# 1/Trabalho/Indústria
# 2/Trabalho/Comércio
# 3/Trabalho/Serviços
# 4/Educação
# 5/Compras
# 6/Saúde
# 7/Lazer
# 8/Residência
# 9/Outros

[Teste: Checar se existe algum número < 1 ou > 9.
Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: dataframe modificado
"""

log_tela.info("### PASSO " + str(passo) + " - MOTIVO_DEST")

# Substituindo valor 6 por 11
df.loc[df['MOTIVO_DEST']==6,'MOTIVO_DEST'] = 11
# Substituindo valor 7 por 6
df.loc[df['MOTIVO_DEST']==7,'MOTIVO_DEST'] = 6
# Substituindo valor 8 por 7
df.loc[df['MOTIVO_DEST']==8,'MOTIVO_DEST'] = 7
# Substituindo valor 10 por 8
df.loc[df['MOTIVO_DEST']==10,'MOTIVO_DEST'] = 8
# Substituindo valor 11 por 9
df.loc[df['MOTIVO_DEST']==11,'MOTIVO_DEST'] = 9
# Substituindo valor 0 por None
df.loc[df['MOTIVO_DEST']==0,'MOTIVO_DEST'] = None

# Verificando intervalo de valores - condições:
# "MOTIVO_DEST >= 1" E "MOTIVO_DEST <= 9"
verifica_range(df, 'MOTIVO_DEST', 1, 9)
log_output.info(' \n\n=====\\n')

return df

def passo_modo1(passo, df):
"""
Substituir todos valores **0** por **None**

# #####Categorias anteriores
# Valor/Descrição
# ----/----
# 1/Ônibus trólebus
# 2/Ônibus Escolar / Empresa
# 3/Dirigindo Automóvel
# 4/Passageiro de Automóvel
# 5/Táxi
# 6/Lotação / Perua
# 7/Metrô

```

```

# 8/Trem
# 9/Motocicleta
# 10/Bicicleta
# 11/A Pé
# 12/Outros

# #####Categorias novas
# Valor/Descrição
# ----/----
# 1/Ônibus
# 2/Ônibus Escolar / Empresa
# 3/Dirigindo Automóvel
# 4/Passageiro de Automóvel
# 5/Táxi
# 6/Lotação / Perua / Van / Microônibus
# 7/Metrô
# 8/Trem
# 9/Moto
# 10/Bicicleta
# 11/A Pé
# 12/Outros

[Teste: Checar se existe algum número < 1 ou > 12.
Se encontrar, retornar erro indicando em qual linha.]
```

:param passo: Número do passo atual para registro/log

:param df:

:return: sem retorno

"""

```

log_tela.info("### PASSO " + str(passo) + " - MODO1")

# Substituindo valor 0 por None
df.loc[df['MODO1']==0,'MODO1'] = None

# Verificando intervalo de valores - condições:
# "MODO1 >= 1" E "MODO1 <= 12"
verifica_range(df, 'MODO1', 1, 12)
log_output.info('\'\n\'=====\\n')

return df
```

def passo_modo2(passo, df):

"""

Substituir todos valores **0** por **None**

#####Categorias anteriores
Valor/Descrição
----/----
1/Ônibus trólebus
2/Ônibus Escolar / Empresa
3/Dirigindo Automóvel
4/Passageiro de Automóvel
5/Táxi
6/Lotação / Perua

```

# 7/Metrô
# 8/Trem
# 9/Motocicleta
# 10/Bicicleta
# 11/A Pé
# 12/Outros

# #####Categorias novas
# Valor/Descrição
# ----/----
# 1/Ônibus
# 2/Ônibus Escolar / Empresa
# 3/Dirigindo Automóvel
# 4/Passageiro de Automóvel
# 5/Táxi
# 6/Lotação / Perua / Van / Microônibus
# 7/Metrô
# 8/Trem
# 9/Moto
# 10/Bicicleta
# 11/A Pé
# 12/Outros

[Teste: Checar se existe algum número < 1 ou > 12.
Se encontrar, retornar erro indicando em qual linha.]
```

:param passo: Número do passo atual para registro/log

:param df:

:return: sem retorno

"""

```

log_tela.info("### PASSO " + str(passo) + " - MODO2")

# Substituindo valor 0 por None
df.loc[df['MODO2']==0,'MODO2'] = None

# Verificando intervalo de valores - condições:
# "MODO2 >= 1" E "MODO2 <= 12"
verifica_range(df, 'MODO2', 1, 12)
log_output.info(' \n\n=====\\n')

return df

def passo_modo3(passo, df):
    """
Substituir todos valores **0** por **None**

# #####Categorias anteriores
# Valor/Descrição
# ----/----
# 1/Ônibus trólebus
# 2/Ônibus Escolar / Empresa
# 3/Dirigindo Automóvel
# 4/Passageiro de Automóvel
# 5/Táxi
# 6/Lotação / Perua

```

```

# 7/Metrô
# 8/Trem
# 9/Motocicleta
# 10/Bicicleta
# 11/A Pé
# 12/Outros

# #####Categorias novas
# Valor/Descrição
# ----/----
# 1/Ônibus
# 2/Ônibus Escolar / Empresa
# 3/Dirigindo Automóvel
# 4/Passageiro de Automóvel
# 5/Táxi
# 6/Lotação / Perua / Van / Microônibus
# 7/Metrô
# 8/Trem
# 9/Moto
# 10/Bicicleta
# 11/A Pé
# 12/Outros

[Teste: Checar se existe algum número < 1 ou > 12.
Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - MODO3")

# Substituindo valor 0 por None
df.loc[df['MODO3']==0,'MODO3'] = None

# Verificando intervalo de valores - condições:
# "MODO3 >= 1" E "MODO3 <= 12"
verifica_range(df, 'MODO3', 1, 12)
log_output.info('\n\n=====\\n')

return df

def passo_modo4(passo, df):
"""
Nada há que se fazer em relação à coluna "MODO4" - não há dados de 1977,
coluna permanecerá vazia
:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - MODO4")

df['MODO4'] = None

log_output.info('\n\n=====\\n')

```

```

    return df

def passo_modo_prin(passo, df):
    """
    Substituir todos valores **0** por **None** 

    # #####Categorias anteriores
    # Valor/Descrição
    # ----/----
    # 1/Ônibus trólebus
    # 2/Ônibus Escolar / Empresa
    # 3/Dirigindo Automóvel
    # 4/Passageiro de Automóvel
    # 5/Táxi
    # 6/Lotação / Perua
    # 7/Metrô
    # 8/Trem
    # 9/Motocicleta
    # 10/Bicicleta
    # 11/A Pé
    # 12/Outros

    # #####Categorias novas
    # Valor/Descrição
    # ----/----
    # 1/Ônibus
    # 2/Ônibus Escolar / Empresa
    # 3/Dirigindo Automóvel
    # 4/Passageiro de Automóvel
    # 5/Táxi
    # 6/Lotação / Perua / Van / Microônibus
    # 7/Metrô
    # 8/Trem
    # 9/Moto
    # 10/Bicicleta
    # 11/A Pé
    # 12/Outros

    [Teste: Checar se existe algum número < 1 ou > 12.
     Se encontrar, retornar erro indicando em qual linha.] 
:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - MODO_PRIN")

# Substituindo valor 0 por None
df.loc[df['MODO_PRIN']==0,'MODO_PRIN'] = None

# Verificando intervalo de valores - condições:
# "MODO_PRIN >= 1" E "MODO_PRIN <= 12"
verifica_range(df, 'MODO_PRIN', 1, 12)
log_output.info('\'\n\'=====\\n')

```

```

    return df

def passo_tipo_vitag(passo, df):
    """
    * Substituir os valores **0** por **None** 

    # #####Categorias novas
    # Valor/Descrição
    # ----/----
    # 1/Coletivo
    # 2/Individual
    # 3/A pé

    [Teste: Checar se existe algum número < 1 ou > 3.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno

    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - TIPO_VIAG")

    # Substituindo valor 0 por None
    df.loc[df['TIPO_VIAG']==0,'TIPO_VIAG'] = None

    # Verificando intervalo de valores - condições:
    # "MODO_PRIN >= 1" E "MODO_PRIN <= 3"
    verifica_range(df, 'TIPO_VIAG', 1, 3)

    log_output.info('\n\n=====\\n')
    return df


def passo_tipo_est_auto(passo, df):
    """
    Substituir valores da coluna "TIPO_EST_AUTO"

    Substituir todos valores **1** por **5**
    Substituir todos valores **2** por **2**
    Substituir todos valores **3** por **2**
    Substituir todos valores **4** por **3**
    Substituir todos valores **5** por **5**
    Substituir todos valores **6** por **4**
    Substituir todos valores **7** por **1**

    # #####Categorias anteriores
    # Valor/Descrição
    # ----/----
    # 1/Zona Azul / Parqímetro
    # 2/Estacionamento Avulso

```

```

# 3/Estacionamento Mensal
# 4/Estacionamento Próprio
# 5/Meio-Fio / Logradouro
# 6/Estacionamento Patrocinado
# 7/Não estacionou

# #####Categorias novas
# Valor/Descrição
# ----/----
# 0/Não Respondeu
# 1/Não Estacionou
# 2/Estacionamento Particular (Avulso / Mensal)
# 3/Estacionamento Próprio
# 4/Estacionamento Patrocinado
# 5/Rua (meio fio / zona azul / zona marrom / parquímetro)

[Teste: Checar se existe algum número < 0 ou > 5.
Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna dataframe modificado
"""

log_tela.info("### PASSO " + str(passo) + " - TIPO_EST_AUTO")

# Substituindo valor 1 por 5
df.loc[df['TIPO_EST_AUTO']==1, 'TIPO_EST_AUTO'] = 5
# Substituindo valor 3 por 2
df.loc[df['TIPO_EST_AUTO']==3, 'TIPO_EST_AUTO'] = 2
# Substituindo valor 4 por 3
df.loc[df['TIPO_EST_AUTO']==4, 'TIPO_EST_AUTO'] = 3
# Substituindo valor 6 por 4
df.loc[df['TIPO_EST_AUTO']==6, 'TIPO_EST_AUTO'] = 4
# Substituindo valor 7 por 1
df.loc[df['TIPO_EST_AUTO']==7, 'TIPO_EST_AUTO'] = 1

# Verificando intervalo de valores - condições:
# "TIPO_EST_AUTO >= 0" E "TIPO_EST_AUTO <= 5"
verifica_range(df, 'TIPO_EST_AUTO', 0, 5)
log_output.info('\n\n=====\\n')

return df

def passo_valor_est_auto(passo, df, deflator):
"""
Corrigir o valor da coluna "VALOR_EST_AUTO" pelo
deflator passado como parâmetro

:param passo: Número do passo atual para registro/log
:param df:
:param deflator: Deflator a ser utilizado para correção
:return: sem retorno
"""

log_tela.info("### PASSO " + str(passo) + " - VALOR_EST_AUTO")

```

```

df['VALOR_EST_AUTO'] = df['VALOR_EST_AUTO'] * deflator

log_output.info('`n`n=====`n')

return df

def passo_tot_viag(passo, df):
    """
    #####
    #      ATENÇÃO      #
    # ESTA FUNÇÃO SÓ DEVE SER #
    # EXECUTADA APÓS A GERAÇÃO#
    # DE TODOS OS ID'S, POIS #
    # HÁ UMA DESCONFIANÇA   #
    # QUANTO À QUALIDADE   #
    #      DESTE DADO       #
    #####
    Calcula e confere o campo TOT_VIAG,
    baseado no maior valor de NO_VIAG para cada pessoa
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe com o "TOT_VIAG" corrigido
"""

log_tela.info("### PASSO " + str(passo) + " - TOT_VIAG")
log_tela.warning('`n`n#####`n' +
                  'Este passo DEVE ser executado apenas após a `n' +
                  'execução dos passos que geram os novos IDs`n' +
                  '#####`n' )

# 'Calculando' o máximo de viagens para cada indivíduo
# Agrupa por ID_PESS e encontra o máximo para NO_VIAG.
# O resultado é um objeto do tipo "Series" cujo index é
# ID_PESS e a variável é NO_VIAG. Em seguida transforma
# esse objeto num DataFrame e depois renomeia a coluna
# NO_VIAG para MAX_VIAG.
df_max = pd.DataFrame(
    df.groupby('ID_PESS')['NO_VIAG'].max()).rename(
        columns={'NO_VIAG': 'MAX_VIAG'})
# Criando um novo dataframe auxiliar apenas com ID_PESS, apenas para ficar mais leve

# O passo seguinte é atribuir o MAX_VIAG à coluna TOT_VIAG
# do dataframe df, linkando por ID_PESS. Isso é feito usando o
# método 'merge' da biblioteca pandas.
df['TOT_VIAG'] = pd.merge(df, df_max, how='left', left_on='ID_PESS',
                           right_index=True)['MAX_VIAG']

log_output.info('TOT_VIAG:`n`n' +
                '      Situação final dos dados: `n' +
                str(df['TOT_VIAG'].describe()) + '\n' +
                '      TOT_VIAG: Situação final dos dados: `n' +
                str(df['TOT_VIAG'].value_counts()))

```

```

# Agora uma função que irá verificar se para todo "ID_PESS"
# o "TOT_VIAG" é igual ao 'NO_VIAG' máximo.
def verifica_no_viag_tot_viag(row):
    if row['NO_VIAG'] != row['TOT_VIAG']:
        log_output.warning('TOT_VIAG: Erro encontrado na linha '
                           + str(row) + ':\\n'
                           + ' => ' + row
                           )
df.loc[:, ['ID_PESS', 'NO_VIAG', 'TOT_VIAG']]\
    .groupby('ID_PESS')\
    .agg({'NO_VIAG': 'max', 'ID_PESS': 'max', 'TOT_VIAG': 'max'})\
    .apply(verifica_no_viag_tot_viag, axis=1)
log_output.info('\\n\\n=====\\n')

return df

```

3.7 Funções que geram os “NO”s e os “ID”s

Função/Variável	Status
gera_nos_e_ids	OK

```
In [ ]: log_tela.info('Definindo funções que geram os "NO"s e os "ID"s')
log_output.info('\n\n=====\\n')
```

```
def gera_nos_e_ids(passo, df):
```

```
"""

```

Esta função gera todos os IDs e todos os NOs das variáveis:
 DOM (domicílio), FAM (família), PESS (pessoa) e VIAG (viagem)
 A ordem de geração é:

NO_DOM, ID_DOM, NO_FAM, ID_FAM, NO_PESS, ID_PESS, NO_VIAG, ID_VIAG
 Esta ordem é fixa pois cada uma dessas variáveis depende da geração
 da variável anterior.

Os NOs são gerados como se fossem subíndices.

No caso dos domicílios, eles são contabilizados dentro de cada zona.
 Assim, cada novo domicílio que aparece na listagem recebe um número
 que representa sua posição na sequência de domicílios dentro da zona
 na qual ele está contido.

As famílias seguem o mesmo raciocínio, com relação ao domicílio, as
 pessoas com relação às famílias e as viagens com relação às pessoas.

Deve-se apenas tomar cuidado com as viagens, pois existem pessoas que
 não realizaram viagens. Neste caso, estes registros devem contabilizar
 o NO_VIAG como zero, e não como 1. Para contemplar estes casos, vamos
 utilizar a variável F_VIAG, que representa quando há ou não viagem
 naquele registro (F_VIAG==1 tem viagem, F_VIAG==0 não tem viagem).
 Assim, o que precisamos fazer para calcular o NO_VIAG é agrupar os
 registros por pessoa (ID_PESS) e, dentro de cada agrupamento, somar
 o valor de F_VIAG registro a registro (linha por linha)
 cumulativamente. No final do processo, atribui-se zero a NO_VIAG
 quando F_VIAG for igual a zero.

```
"""

```

```
def gera_id_dom(row):
```

```
"""

```

Gera o ID_DOM baseado no 'ANO', na 'ZONA_DOM' e no 'NO_DOM'
 O argumento passado é a "linha".

Uso:

```
df['ID_DOM'] = df.apply(gera_id_dom, axis=1)
```

Retorna: ID_DOM da respectiva linha

```
"""

```

Formatando o ano para string
 ano = str(row['ANO'])

Formatando a zona para ficar como string com 3 caracteres
 zona = str('%03d' % row['ZONA_DOM'])

```

# Formatando no_dom para ficar como string com 4 caracteres
no_dom = str('%04d' % row['NO_DOM'])

# Retornando o número inteiro correspondente à string
# concatenada de ano + zona + no_dom
return ano + zona + no_dom


def gera_id_fam(row):
    """
    Gera o ID_FAM baseado no 'ID_DOM' e no 'NO_FAM'
    O argumento passado é a "linha".
    Uso:
        df['ID_FAM'] = df.apply(gera_ID_FAM, axis=1)
    Retorna: ID_FAM da respectiva linha
    """
    # Formatando id_dom como string
    id_dom = str(row['ID_DOM'])

    # Formatando no_fam como string para ficar com 2 caracteres
    no_fam = str('%02d' % row['NO_FAM'])

    # Retornando o número inteiro correspondente à string
    # concatenada de ID_DOM com NO_FAM
    return id_dom + no_fam


def gera_id_pess(row):
    """
    Gera o ID_PESS baseado no 'ID_FAM' e no 'NO_PESS'
    O argumento passado é a "linha".
    Uso:
        df['ID_PESS'] = df.apply(gera_ID_PESS, axis=1)
    Retorna: ID_PESS da respectiva linha
    """
    # Formatando id_fam como string
    id_fam = str(row['ID_FAM'])

    # Formatando no_pess como string para ficar com 2 caracteres
    no_pess = str('%02d' % row['NO_PESS'])

    # Retornando o número inteiro correspondente à string
    # concatenada de ID_FAM com NO_PESS
    return id_fam + no_pess


def gera_id_viag(row):
    """
    Gera o ID_VIAG baseado no 'ID_PESS' e no 'NO_VIAG'
    O argumento passado é a "linha".
    Uso:
        df['ID_VIAG'] = df.apply(gera_ID_VIAG, axis=1)
    Retorna ID_VIAG da respectiva linha
    """
    # Formatando id_pess como string
    id_pess = str(row['ID_PESS'])

```

```

# Formatando no_viag como string para ficar com 2 caracteres
no_viag = str('%02d' % row['NO_VIAG'])

# Retornando o número inteiro correspondente à string
# concatenada de ID_PESS com NO_VIAG
return id_pess + no_viag

#gera NO_DOM
log_tela.info("Gerando NO_DOM")
df['NO_DOM'] = df.groupby('ZONA_DOM', sort=False)['F_DOM'].cumsum()
log_tela.info("NO_DOM gerado")
log_tela.info("Gerando ID_DOM")
#gera ID_DOM
df['ID_DOM'] = df.apply(gera_id_dom, axis=1)
log_tela.info("ID_DOM gerado")

#gera NO_FAM
log_tela.info("Gerando NO_FAM")
df['NO_FAM'] = df.groupby('ID_DOM', sort=False)['F_FAM'].cumsum()
log_tela.info("NO_FAM gerado")
log_tela.info("Gerando ID_FAM")
#gera ID_FAM
df['ID_FAM'] = df.apply(gera_id_fam, axis=1)
log_tela.info("ID_FAM gerado")

#gera NO_PESS
log_tela.info("Gerando NO_PESS")
df['NO_PESS'] = df.groupby('ID_FAM', sort=False)['F_PESS'].cumsum()
log_tela.info("NO_PESS gerado")
log_tela.info("Gerando ID_PESS")
#gera ID_PESS
df['ID_PESS'] = df.apply(gera_id_pess, axis=1)
log_tela.info("ID_PESS gerado")

#gera NO_VIAG
log_tela.info("Gerando NO_VIAG")
df['NO_VIAG'] = df.groupby('ID_PESS', sort=False)['F_VIAG'].cumsum()
# Verificando se F_VIAG == 0 e zerando NO_VIAG nesse caso
df.loc[df['F_VIAG'] == 0, 'NO_VIAG'] = 0
log_tela.info("NO_VIAG gerado")
log_tela.info("Gerando ID_VIAG")
#gera ID_VIAG
df['ID_VIAG'] = df.apply(gera_id_viag, axis=1)
log_tela.info("ID_VIAG gerado")

return df

```

3.8 Função relativa à entrevista

Função/Variável	Status
passo_cd_entre	OK

Obs: o passo_cd_entre deve ser executado após o passo_tot_vitag.

```
In [ ]: log_tela.info('Definindo função relativa à entrevista')
log_output.info('\n\n=====\\n')

def passo_cd_entre(passo, df):
    """
    -----
    Substituir valores da coluna "CD_ENTRE"
    Todas entrevistas são consideradas "completas", segundo informações do Metrô

    * sem viagem: se TOT_VIAG == 0
    * com viagem: se TOT_VIAG != 0

    # #####Categorias novas
    # | Valor | Descrição |
    # | ----- | ----- |
    # | 0 | Completa sem viagem |
    # | 1 | Completa com viagem |

    [Teste: Checar se existe algum número diferente de 0 ou 1.
     Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return:
"""
    log_tela.info("### PASSO " + str(passo) + " - CD_ENTRE")
    log_tela.warning('\n\n#####\\n' +
                     'Este passo DEVE ser executado apenas após a \\n' +
                     'execução do passo que geram o TOT_VIAG.\\n' +
                     '#####\\n' )

    # Definindo 'CD_ENTRE' baseado no valor de 'TOT_VIAG'
    df.loc[df['TOT_VIAG'] == 0, 'CD_ENTRE'] = 0
    df.loc[df['TOT_VIAG'] != 0, 'CD_ENTRE'] = 1

    verifica_dummy(df, 'CD_ENTRE')
    log_output.info('\n\n=====\\n')

    return df

In [ ]: def passo_correcoes(passo, df):
    log_tela.info("### PASSO " + str(passo) + " - Correções")

    log_tela.info("Corrigindo 'SERV_PAS' caso TOT_VIAG seja nulo.")
    df.loc[df['TOT_VIAG'] == 0, 'SERV_PAS_ORIG'] = 0
```

```
df.loc[df['TOT_VIAG'] == 0, 'SERV_PAS_DEST'] = 0

log_tela.info("Corrigindo motivo_origem, provável domicílio")
df.loc[(df['MOTIVO_ORIG']==0) &
       (df['MOTIVO_DEST']!=0) &
       (df['ZONA_ORIG']==df['ZONA_DOM']) &
       (df['SUBZONA_ORIG']==df['SUBZONA_DOM']), 'MOTIVO_ORIG'] == 8

log_tela.info("Corrigindo motivo_destino, provável domicílio")
df.loc[(df['MOTIVO_ORIG']!=0) &
       (df['MOTIVO_DEST']==0) &
       (df['ZONA_DEST']==df['ZONA_DOM']) &
       (df['SUBZONA_DEST']==df['SUBZONA_DOM']), 'MOTIVO_DEST'] == 8

return df
```

4 Defining the main function

This function will load all needed files, then call all other step functions defined above. On the end this main function will save the transformed data into an csv file.

```
In [ ]: def main():
    start_time = time.time()

    log_tela.info("Horário de início da execução: %s" %
                  time.strftime("%H:%M", time.localtime(start_time)))

    # ----

    log_tela.info('Lendo arquivos CSV')

    log_output.info('Lendo CSV da base original da OD.')
    od = pd.read_csv('bases/OD_1977.csv', sep=';', decimal=',')

    # Definindo variável de deflação de renda:
    deflator = 0.422345901

    # Filtrando dataframe para pegar apenas uma amostra
    # logging.info('\nFiltering the main dataframe to get just a sample')
    #od = od[:1000].copy()

    log_output.info('\n\n=====\\n')
    log_tela.info('Pré-Processamento.')
    od = pre_processamento(od)

    log_tela.info('Imprimindo a lista de variáveis do dataframe da OD.')
    log_tela.debug(od.columns.tolist())

    #Contador de 'PASSO'
    passo = 1

    # ----
    # ##Passo: "ANO"
    od = passo_ano(passo, od)
    passo += 1

    # ----
    # ##Passo: "DIA_SEM"
    od = passo_dia_sem(passo, od)
    passo += 1

    # ----
    # ##Passo: "UCODs"
    od = passo_ucods(passo, od)
    passo += 1

    # ----
    # ##Passo: "ZONA_DOM"
    od = passo_zona_dom(passo, od)
    passo += 1
```

```

# -----
# ##Passo: "SUBZONA_DOM"
od = passo_subzona_dom(passo, od)
passo += 1

# -----
# ##Passo: "MUN_DOM"
od = passo_mun_dom(passo, od)
passo += 1

# -----
# ##Passo: "F_DOM"
od = passo_f_dom(passo, od)
passo += 1

# -----
# ##Passo: "FE_DOM"
od = passo_fe_dom(passo, od)
passo += 1

# -----
# ##Passo: "TIPO_DOM"
od = passo_tipo_dom(passo, od)
passo += 1

# -----
# ##Passo: "TOT_FAM"
od = passo_tot_fam(passo, od)
passo += 1

# -----
# ##Passo: "F_FAM"
od = passo_f_fam(passo, od)
passo += 1

# -----
# ##Passo: "FE_FAM"
od = passo_fe_fam(passo, od)
passo += 1

# -----
# ##Passo: "COND_MORA"
od = passo_cond_mora(passo, od)
passo += 1

# -----
# ##Passo: "QT_AUTO"
od = passo_qt_auto(passo, od)
passo += 1

# -----
# ##Passo: "QT_BICI"
od = passo_qt_bici(passo, od)
passo += 1

```

```

# -----
# ##Passo: "QT_MOTO"
od = passo_qt_moto(passo, od)
passo += 1

# -----
# ##PASSO: "REN_FAM"
od = passo_ren_fam(passo, od, deflator)
passo += 1

# -----
# ##Passo: "CD_RENFAM"
od = passo_cd_renfam(passo, od)
passo += 1

# -----
# ##Passo: "F_PESS"
od = passo_f_pess(passo, od)
passo += 1

# -----
# ##Passo: "FE_PESS"
od = passo_fe_pess(passo, od)
passo += 1

# -----
# ##Passo: "SIT_FAM"
od = passo_sit_fam(passo, od)
passo += 1

# -----
# ##Passo: "IDADE"
od = passo_idade(passo, od)
passo += 1

# -----
# ##Passo: "SEXO"
od = passo_sexo(passo, od)
passo += 1

# -----
# ##Passo: "GRAU_INSTR"
od = passo_grau_instr(passo, od)
passo += 1

# -----
# ##Passo: "OCUP"
od = passo_ocup(passo, od)
passo += 1

# -----
# ##Passo: "SETOR_ATIV"
od = passo_setor_ativ(passo, od)

```

```

    passo += 1

    # -----
    # ##Passo: "REN_IND"
    od = passo_ren_ind(passo, od, deflator)
    passo += 1

    # -----
    # ##Passo: "CD_RENIND"
    od = passo_cd_renind(passo, od)
    passo += 1

## O Passo estuda encontra-se mais abaixo, após MUN_ESC

    # -----
    # ##Passo: "F_VIAG"
    od = passo_f_viag(passo, od)
    passo += 1

    # -----
    # ##Passo: "FE_VIAG"
    od = passo_fe_viag(passo, od)
    passo += 1

    # -----
    # ##Passo: "ZONA_ESC"
    od = passo_zona_esc(passo, od)
    passo += 1

    # -----
    # ##Passo: "SUBZONA_ESC"
    od = passo_subzona_esc(passo, od)
    passo += 1

    # -----
    # ##Passo: "MUN_ESC"
    od = passo_mun_esc(passo, od)
    passo += 1

    # -----
    # ##Passo: "ESTUDA"
    # Este passo deve vir após os passos de localização da escola
    od = passo_estuda(passo, od)
    passo += 1

    # -----
    # ##Passo: "ZONA_TRAB1"
    od = passo_zona_trab1(passo, od)
    passo += 1

    # -----
    # ##Passo: "SUBZONA_TRAB1"
    od = passo_subzona_trab1(passo, od)
    passo += 1

```

```

# -----
# ##Passo: "MUN_TRAB1"
od = passo_mun_trab1(passo, od)
passo += 1

# -----
# ##Passo: "ZONA_TRAB2"
od = passo_zona_trab2(passo, od)
passo += 1

# -----
# ##Passo: "SUBZONA_TRAB2"
od = passo_subzona_trab2(passo, od)
passo += 1

# -----
# ##Passo: "MUN_TRAB2"
od = passo_mun_trab2(passo, od)
passo += 1

# -----
# ##Passo: "ZONA_ORIG"
od = passo_zona_orig(passo, od)
passo += 1

# -----
# ##Passo: "SUBZONA_ORIG"
od = passo_subzona_orig(passo, od)
passo += 1

# -----
# ##Passo: "MUN_ORIG"
od = passo_mun_orig(passo, od)
passo += 1

# -----
# ##Passo: "ZONA_DEST"
od = passo_zona_dest(passo, od)
passo += 1

# -----
# ##Passo: "SUBZONA_DEST"
od = passo_subzona_dest(passo, od)
passo += 1

# -----
# ##Passo: "MUN_DEST"
od = passo_mun_dest(passo, od)
passo += 1

# -----
# ##Passo: "SERV_PAS_ORIG"
od = passo_serv_pas_orig(passo, od)

```

```

    passo += 1

    # -----
    # ##Passo: "SERV_PAS_DEST"
    od = passo_serv_pas_dest(passo, od)
    passo += 1

    # -----
    # ##Passo: "MOTIVO_ORIG"
    od = passo_motivo_orig(passo, od)
    passo += 1

    # -----
    # ##Passo: "MOTIVO_DEST"
    od = passo_motivo_dest(passo, od)
    passo += 1

    # -----
    # ##Passo: "MOD01"
    od = passo_modo1(passo, od)
    passo += 1

    # -----
    # ##Passo: "MOD02"
    od = passo_modo2(passo, od)
    passo += 1

    # -----
    # ##Passo: "MOD03"
    od = passo_modo3(passo, od)
    passo += 1

    # -----
    # ##Passo: "MOD04"
    od = passo_modo4(passo, od)
    passo += 1

    # -----
    # ##Passo: "MODO_PRIN"
    od = passo_modo_prin(passo, od)
    passo += 1

    # -----
    # ##Passo: "TIPO_VIAG"
    od = passo_tipo_viag(passo, od)
    passo += 1

    # -----
    # ##"H_SAIDA"; "MIN_SAIDA"; "ANDA_ORIG"; "H_CHEG"; "MIN_CHEG";
    #     "ANDA_DEST"; "DIST_VIAG" e "DURACAO"
    # Nada há que se fazer em relação aos dados das colunas acima mencionadas

    # -----
    # ##Passo: "TIPO_EST_AUTO"

```

```

od = passo_tipo_est_auto(passo, od)
passo += 1

# -----
# ##Passo: "VALOR_EST_AUTO"
od = passo_valor_est_auto(passo, od, deflator)
passo += 1

# -----
# ##Passo: Coordenadas
od = coordenadas(passo, od)
passo += 1

## O passo TOT_VIAG apenas é chamado após a geração dos IDs.

# -----
# ##Passo: Gerando NO's e ID's
od = gera_nos_e_ids(passo, od)
passo += 1

# -----
# ##Passo: "TOT_VIAG"
od = passo_tot_viag(passo, od)
passo += 1

# -----
# ##Passo: "CD_ENTRE"
od = passo_cd_entre(passo, od)
passo += 1

# -----
# ##Passo: Correções
od = passo_correcoes(passo, od)
passo += 1

log_tela.info('Salvando dataframe como arquivo CSV')
# -----
# ## Salvando o dataframe num arquivo local
od.to_csv('outputs/1977_od.csv', sep=';', decimal=',', index=False)

log_tela.info("Output gerado. Arquivo: outputs/1977_od.csv")

log_tela.info("Tempo total de execução: %s segundos" %
              (time.time() - start_time))

log_tela.info("Horário de finalização: %s" %
              (time.strftime("%H:%M", time.localtime(time.time()))))

log_tela.info("Terminou o main")

```

5 RUN the main() function....

```
In [ ]: if __name__ == "__main__":
          main()
```

rotina_final_1987

January 4, 2016

1 Setup Inicial

```
In [ ]: # coding: utf-8
        import math
        import logging
        import time
        import pandas as pd

        # Faz os gráficos um pouco mais bonitos
        pd.set_option('display.mpl_style', 'default')
```

2 Definindo Loggers

Define os loggers

Estes ‘loggers’ serão utilizados para salvar as saídas (outputs) em um arquivo de texto no diretório ‘outputs’.

ref: <http://stackoverflow.com/questions/17035077/python-logging-to-multiple-log-files-from-different-classes>

ATENÇÃO: RODAR O BLOCO ABAIXO APENAS UMA VEZ, SE ESTE BLOCO FOR EXECUTADO MAIS DE UMA VEZ OS LOGs SERÃO DUPLICADOS.

```
In [ ]: log_formatter = logging.Formatter('%(asctime)s | %(levelname)s: %(message)s')

log_output = logging.getLogger('log_output')
FH_output = logging.FileHandler(
            'outputs/1987_output.log', mode='w')
FH_output.setFormatter(log_formatter)
log_output.setLevel(logging.INFO)
log_output.addHandler(FH_output)
log_output.propagate = False

# Este logger (log_tela) loga na tela e também
# no arquivo de output, junto com o conteúdo do
# logger log_output
log_tela = logging.getLogger('log_tela')
SH_tela = logging.StreamHandler()
SH_tela.setFormatter(log_formatter)
log_tela.addHandler(SH_tela)
log_tela.addHandler(FH_output)
log_tela.setLevel(logging.INFO)
log_tela.propagate = False
```

3 Funções de 1987

3.1 Funções gerais assessórias

Função	Status
verifica_dummy	OK
verifica_range	OK
pre_processamento	OK
coordenadas	OK

```
In [ ]: log_tela.info('Definindo as funções gerais assessórias')
log_output.info('\n'=====\\n')

def verifica_dummy(df, variavel):
    """
    Verifica se uma variável, dummy, contém algum valor diferente de 0 ou de 1.
    :param df: dataframe com os dados a serem verificados
    :param variavel: string com o nome da variável (coluna) que tem os dados
        que devem ser verificados.
    :return: Sem retorno, apenas salva as avaliações nos logs.
    Uso:
        verifica_dummy(dataframe, 'coluna a ser verificada')
    """
    contador_de_erros = 0
    log_tela.info('Verificando a variável Dummy: ' + variavel)

    df_erros = df[(df[variavel] != 0) & (df[variavel] != 1)]
    if len(df_erros[variavel].value_counts()) > 0:
        log_tela.warning(variavel + ": " +
                          str(len(df_erros[variavel].value_counts())) +
                          " erros encontrados:\\n" +
                          str(df_erros[variavel].value_counts()))
    else:
        log_tela.info(variavel + ": Nenhum erro encontrado.")

def verifica_range(df, variavel, valor_menor, valor_maior):
    """
    Verifica se uma variável, do tipo número inteiro, contém algum valor menor
    que "valor_menor" ou maior que "valor_maior".
    :param df: dataframe com os dados a serem verificados
    :param variavel: string com o nome da variável (coluna) que tem os dados
        que devem ser verificados
    :param valor_menor: Valor mínimo esperado na variável (int ou float)
    :param valor_maior: Valor máximo esperado na variável (int ou float)
    :return: Sem retorno, apenas salva as avaliações nos logs.
    Uso:
        verifica_range(dataframe, 'nome_variavel', valor_menor, valor_maior)
    """
    log_tela.info('Verificando Range da variável: ' + variavel)
```

```

# Obs: Registros inválidos: None (equivalente a NA)
nulos = df[variavel].isnull().sum()
nulos = nulos if nulos else 0
log_output.info('\n\n' +
    ',       - ' + 'Mínimo esperado: ' + str(valor_menor) + '\n' +
    ',       - ' + 'Máximo esperado: ' + str(valor_maior) + '\n' +
    ',       - ' + 'Total de registros: ' + str(len(df[variavel])) +
    '\n' +
    ',       - ' + 'Registros nulos (NA): ' +
    str(df[variavel].isnull().sum()) + '\n'
)
df_errores = df[(df[variavel] < valor_menor) | (df[variavel] > valor_maior)]

if len(df_errores[variavel].value_counts()) > 0:

    result = df_errores[variavel].value_counts().sort_index()
    # Verificando limite inferior
    if result.first_valid_index() < valor_menor:
        log_tela.warning(
            variavel + ': ' + 'Valor inteiro mínimo encontrado: ' +
            str(result.first_valid_index()) + ' - abaixo do esperado!')
    # Verificando limite superior
    if result.last_valid_index() > valor_maior:
        log_tela.warning(
            variavel + ': ' + 'Valor inteiro máximo encontrado: ' +
            str(result.last_valid_index()) + " - acima do esperado!")

    log_tela.warning(variavel + ': ' +
        str(len(df_errores[variavel].value_counts())) +
        ' valor(es) incorreto(s) ' +
        'encontrado(s) nesta variável:\n' +
        str(df_errores[variavel].value_counts()))
else:
    log_tela.info(variavel + ": Nenhum erro encontrado.")

def pre_processamento(df):
    """
    Realiza algumas ações prévias ao processamento dos dados,
    removendo alguns registros e ajustando o dataframe.
    """
    log_tela.info('Criando/Renomeando colunas no dataframe principal')

    log_output.info('Renomeando coluna UCOD para UCOD_DOM')
    df.rename(columns={'UCOD': 'UCOD_DOM'}, inplace=True)

    log_output.info('Renomeando coluna ANDA_CHEG para ANDA_DEST')
    df.rename(columns={'ANDA_CHEG': 'ANDA_DEST'}, inplace=True)

    log_tela.info('Verificando se todas as variáveis esperadas'+
        'existem no dataframe.\n' +
        'Caso não exista alguma, ela é criada.')

```

```

variaveis = ['ANO', 'CD_ENTRE', 'DIA_SEM', 'UCOD_DOM', 'ZONA_DOM',
             'SUBZONA_DOM', 'MUN_DOM', 'CO_DOM_X', 'CO_DOM_Y', 'ID_DOM',
             'F_DOM', 'FE_DOM', 'NO_DOM', 'TIPO_DOM', 'TOT_FAM', 'ID_FAM',
             'F_FAM', 'FE_FAM', 'NO_FAM', 'COND_MORA', 'QT_AUTO', 'QT_BICI',
             'QT_MOTO', 'CD_RENFAM', 'REN_FAM', 'ID_PESS', 'F_PESS',
             'FE_PESS', 'NO_PESS', 'SIT_FAM', 'IDADE', 'SEXO', 'ESTUDA',
             'GRAU_INSTR', 'OCUP', 'SETOR_ATIV', 'CD_RENIND', 'REN_IND',
             'UCOD_ESC', 'ZONA_ESC', 'SUBZONA_ESC', 'MUN_ESC', 'CO_ESC_X',
             'CO_ESC_Y', 'UCOD_TRAB1', 'ZONA_TRAB1', 'SUBZONA_TRAB1',
             'MUN_TRAB1', 'CO_TRAB1_X', 'CO_TRAB1_Y', 'UCOD_TRAB2',
             'ZONA_TRAB2', 'SUBZONA_TRAB2', 'MUN_TRAB2', 'CO_TRAB2_X',
             'CO_TRAB2_Y', 'ID_VIAG', 'F_VIAG', 'FE_VIAG', 'NO_VIAG',
             'TOT_VIAG', 'UCOD_ORIG', 'ZONA_ORIG', 'SUBZONA_ORIG',
             'MUN_ORIG', 'CO_ORIG_X', 'CO_ORIG_Y', 'UCOD_DEST', 'ZONA_DEST',
             'SUBZONA_DEST', 'MUN_DEST', 'CO_DEST_X', 'CO_DEST_Y',
             'DIST_VIAG', 'SERV_PAS_ORIG', 'SERV_PAS_DEST', 'MOTIVO_ORIG',
             'MOTIVO_DEST', 'MODO1', 'MODO2', 'MODO3', 'MODO4', 'MODO_PRIN',
             'TIPO_VIAG', 'H_SAIDA', 'MIN_SAIDA', 'ANDA_ORIG', 'H_CHEG',
             'MIN_CHEG', 'ANDA_DEST', 'DURACAO', 'TIPO_EST_AUTO',
             'VALOR_EST_AUTO']

for variavel in variaveis:
    if variavel not in df.columns:
        # Se a variável não existe no dataframe então ela é criada
        # com valor padrão de NONE (NA)
        df[variavel] = None
    log_tela.info('Criando a variavel ' + variavel + ' no dataframe')

log_output.info('Reordenando as variáveis')
df = df[['ANO', 'CD_ENTRE', 'DIA_SEM', 'UCOD_DOM', 'ZONA_DOM',
          'SUBZONA_DOM', 'MUN_DOM', 'CO_DOM_X', 'CO_DOM_Y', 'ID_DOM',
          'F_DOM', 'FE_DOM', 'NO_DOM', 'TIPO_DOM', 'TOT_FAM', 'ID_FAM',
          'F_FAM', 'FE_FAM', 'NO_FAM', 'COND_MORA', 'QT_AUTO', 'QT_BICI',
          'QT_MOTO', 'CD_RENFAM', 'REN_FAM', 'ID_PESS', 'F_PESS',
          'FE_PESS', 'NO_PESS', 'SIT_FAM', 'IDADE', 'SEXO', 'ESTUDA',
          'GRAU_INSTR', 'OCUP', 'SETOR_ATIV', 'CD_RENIND', 'REN_IND',
          'UCOD_ESC', 'ZONA_ESC', 'SUBZONA_ESC', 'MUN_ESC', 'CO_ESC_X',
          'CO_ESC_Y', 'UCOD_TRAB1', 'ZONA_TRAB1', 'SUBZONA_TRAB1',
          'MUN_TRAB1', 'CO_TRAB1_X', 'CO_TRAB1_Y', 'UCOD_TRAB2',
          'ZONA_TRAB2', 'SUBZONA_TRAB2', 'MUN_TRAB2', 'CO_TRAB2_X',
          'CO_TRAB2_Y', 'ID_VIAG', 'F_VIAG', 'FE_VIAG', 'NO_VIAG',
          'TOT_VIAG', 'UCOD_ORIG', 'ZONA_ORIG', 'SUBZONA_ORIG',
          'MUN_ORIG', 'CO_ORIG_X', 'CO_ORIG_Y', 'UCOD_DEST', 'ZONA_DEST',
          'SUBZONA_DEST', 'MUN_DEST', 'CO_DEST_X', 'CO_DEST_Y',
          'DIST_VIAG', 'SERV_PAS_ORIG', 'SERV_PAS_DEST', 'MOTIVO_ORIG',
          'MOTIVO_DEST', 'MODO1', 'MODO2', 'MODO3', 'MODO4', 'MODO_PRIN',
          'TIPO_VIAG', 'H_SAIDA', 'MIN_SAIDA', 'ANDA_ORIG', 'H_CHEG',
          'MIN_CHEG', 'ANDA_DEST', 'DURACAO', 'TIPO_EST_AUTO',
          'VALOR_EST_AUTO']]

log_output.info('\n\n=====\\n')
log_output.info('\\n=====\\n')

return df

```

```

def coordenadas(passo, df):
    """
    Itera sobre os registros do dataframe coords,
    a cada iteração, utiliza o valor das colunas ZONA e SUBZONA
    para filtrar o dataframe df em cada um dos tipos de ZONA E
    SUBZONA (DOM, ESC, TRAB1, TRAB2, ORIG, DEST) e substitui os valores
    das coordenadas X e Y de cada tipo (CO_DOM_X, CO_ESC_Y, etc) com
    os valores das colunas CO_X e CO_Y do dataframe coords.

    :param passo: número do passo para o log
    :param df: Dafaframe a ser modificado (ex.: od1987)
    :return: retorna o dataframe modificado com todas as coordenadas aplicadas
    """
    log_tela.info("### PASSO " + str(passo) + " - COORDENADAS")

    def coord_aux(row):
        """
        Esta função irá receber uma linha (row) do dataframe coords e
        utilizar os valores desta linha para realizar as alterações
        de coordenadas no dataframe df.
        """
        # Começando pelo tipo DOM, alterando CO_DOM_X e depois CO_DOM_Y
        df.loc[(df['ZONA_DOM'] == row['ZONA']) &
               (df['SUBZONA_DOM'] == row['SUBZONA']), 'CO_DOM_X'] = row['CO_X']
        df.loc[(df['ZONA_DOM'] == row['ZONA']) &
               (df['SUBZONA_DOM'] == row['SUBZONA']), 'CO_DOM_Y'] = row['CO_Y']
        # Agora com o tipo ESC, alterando CO_ESC_X e depois CO_ESC_Y
        df.loc[(df['ZONA_ESC'] == row['ZONA']) &
               (df['SUBZONA_ESC'] == row['SUBZONA']), 'CO_ESC_X'] = row['CO_X']
        df.loc[(df['ZONA_ESC'] == row['ZONA']) &
               (df['SUBZONA_ESC'] == row['SUBZONA']), 'CO_ESC_Y'] = row['CO_Y']
        # Agora com o tipo TRAB1, alterando CO_TRAB1_X e depois CO_TRAB1_Y
        df.loc[(df['ZONA_TRAB1'] == row['ZONA']) &
               (df['SUBZONA_TRAB1'] == row['SUBZONA']), 'CO_TRAB1_X'] = row['CO_X']
        df.loc[(df['ZONA_TRAB1'] == row['ZONA']) &
               (df['SUBZONA_TRAB1'] == row['SUBZONA']), 'CO_TRAB1_Y'] = row['CO_Y']
        # Agora com o tipo TRAB2, alterando CO_TRAB2_X e depois CO_TRAB2_Y
        df.loc[(df['ZONA_TRAB2'] == row['ZONA']) &
               (df['SUBZONA_TRAB2'] == row['SUBZONA']), 'CO_TRAB2_X'] = row['CO_X']
        df.loc[(df['ZONA_TRAB2'] == row['ZONA']) &
               (df['SUBZONA_TRAB2'] == row['SUBZONA']), 'CO_TRAB2_Y'] = row['CO_Y']
        # Agora com o tipo ORIG, alterando CO_ORIG_X e depois CO_ORIG_Y
        df.loc[(df['ZONA_ORIG'] == row['ZONA']) &
               (df['SUBZONA_ORIG'] == row['SUBZONA']), 'CO_ORIG_X'] = row['CO_X']
        df.loc[(df['ZONA_ORIG'] == row['ZONA']) &
               (df['SUBZONA_ORIG'] == row['SUBZONA']), 'CO_ORIG_Y'] = row['CO_Y']
        # Agora com o tipo DEST, alterando CO_DEST_X e depois CO_DEST_Y
        df.loc[(df['ZONA_DEST'] == row['ZONA']) &
               (df['SUBZONA_DEST'] == row['SUBZONA']), 'CO_DEST_X'] = row['CO_X']
        df.loc[(df['ZONA_DEST'] == row['ZONA']) &
               (df['SUBZONA_DEST'] == row['SUBZONA']), 'CO_DEST_Y'] = row['CO_Y']

```

```
log_output.info('Lendo arquivo auxiliar de Coordenadas das subzonas')
coords = pd.read_csv('auxiliares/coord_subzonas_1987.csv', sep=';', decimal=',')

# Esta variável out não é utilizada para nada além de evitar um
# monte de output que não será utilizado e que é gerado pelo método apply.
out = coords.apply(coord_aux, axis=1)

log_output.info('\n\n=====\\n')

return df
```

3.2 Funções Gerais

Função/Variável	Status
passo_ano	OK
passo_dia_sem	OK
passo_ucods	OK

```
In [ ]: log_tela.info('Definindo as funções gerais')
log_output.info('\n'=====\\n')

def passo_ano(passo, df):
    """
    Preenche a coluna "ANO" com valor 2 em todas células
    Categorias:
    /valor/ano_correspondente/
    /-----/-----
    /1/1977/
    /2/1987/
    /3/1997/
    /4/2007/

    :param passo: Número do passo atual para registro/log
    :param df: dataframe a ser modificado
    :return: retorna o dataframe modificado
    """

    log_tela.info("### PASSO " + str(passo) + " - ANO")

    # Definindo valor '2' para todas as células da coluna ANO
    df["ANO"] = 2

    return df

def passo_dia_sem(passo, df):
    """
    Assumindo que as respostas iguais a **1** são
    referentes à segunda-feira (2) e
    valores iguais a **7** são referentes à
    sexta-feira (6)

    # #####Categorias:
    # Valor/Descrição
    # -----/-----
    # 2/Segunda-Feira
    # 3/Terça-Feira
    # 4/Quarta-Feira
    # 5/Quinta-Feira
    # 6/Sexta-Feira
```

```

:param passo: Número do passo atual para registro/log
:param df: dataframe a ser modificado
:return: retorna o dataframe modificado
"""
log_tela.info("### PASSO " + str(passo) + " - DIA_SEM")

# Substituindo **0** por **None**
df.loc[df['DIA_SEM']==0,'DIA_SEM'] = None

# Alocando que respondeu "1" para Segunda-feira (2)
df.loc[df['DIA_SEM']==1,'DIA_SEM'] = 2

# Alocando quem respondeu "7" para Sexta-feira (6)
df.loc[df['DIA_SEM']==7,'DIA_SEM'] = 6

# Verificando intervalo de valores - condições:
# "DIA_SEM >= 2" E "DIA_SEM <= 6"
verifica_range(df, 'DIA_SEM', 2, 6)
log_output.info('\n\n=====\\n')

return df

def passo_ucods(passo, df):
"""
Itera sobre os registros do dataframe ucods,
a cada iteração, utiliza o valor da coluna ZONA_REF
para filtrar o dataframe df em cada um dos tipos de ZONA
(DOM, ESC, TRAB1, TRAB2, ORIG, DEST) e substitui o valor
da respectiva UCOD.

:param passo: número do passo para o log
:param df: Dafaframe a ser modificado (ex.: od1987)
:return: retorna o dataframe modificado com todas as UCODS aplicadas
"""
log_tela.info("### PASSO " + str(passo) + " - UCODS")

def ucod_aux(row):
    df.loc[df['ZONA_DOM']==row['ZONA_REF'], 'UCOD_DOM'] = row['UCOD_BUSCADA']
    df.loc[df['ZONA_ESC']==row['ZONA_REF'], 'UCOD_ESC'] = row['UCOD_BUSCADA']
    df.loc[df['ZONA_TRAB1']==row['ZONA_REF'], 'UCOD_TRAB1'] = row['UCOD_BUSCADA']
    df.loc[df['ZONA_TRAB2']==row['ZONA_REF'], 'UCOD_TRAB2'] = row['UCOD_BUSCADA']
    df.loc[df['ZONA_ORIG']==row['ZONA_REF'], 'UCOD_ORIG'] = row['UCOD_BUSCADA']
    df.loc[df['ZONA_DEST']==row['ZONA_REF'], 'UCOD_DEST'] = row['UCOD_BUSCADA']

log_output.info('Lendo arquivo auxiliar UCOD')
ucods = pd.read_csv('auxiliares/UCOD-1987.csv', sep=';', decimal=',')

# Esta variável out não é utilizada para nada além de evitar um
# monte de output que não será utilizado e que é gerado pelo método apply.
out = ucods.apply(ucod_aux, axis=1)

# Verificando intervalo de valores - condições:
# "UCOD_XXX >= 1" E "UCOD_XXX <= 67"

```

```
for tipo in ['DOM', 'ESC', 'TRAB1', 'TRAB2', 'ORIG', 'DEST']:
    verifica_range(df, 'UCOD_' + tipo, 1, 67)
log_output.info('\n\n=====\\n')

return df
```

3.3 Funções do Domicílio

Função/Variável	Status
passo_zona_dom	OK
passo_subzona_dom	OK
passo_mun_dom	OK
passo_f_dom	OK
passo_fe_dom	OK
passo_tipo_dom	OK

```
In [ ]: log_tela.info('Definindo as funções do domicílio')
log_output.info('\n\n=====\\n')

def passo_zona_dom(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 254

    [Teste: Checar se existe algum número < 1 ou > 254.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: passo
    :param df: dataframe
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - ZONA_DOM")

    # Substituindo valor 0 por None
    df.loc[df['ZONA_DOM']==0,'ZONA_DOM'] = None

    # Verificando intervalo de valores - condições:
    # "ZONA_DOM >= 1" E "ZONA_DOM <= 254"
    verifica_range(df, 'ZONA_DOM', 1, 254)
    log_output.info('\n\n=====\\n')

    return df

def passo_subzona_dom(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 9

    [Teste: Checar se existe algum número < 1 ou > 9.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
```

```

:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - SUBZONA_DOM")

# Substituindo valor 0 por None
df.loc[df['SUBZONA_DOM']==0,'SUBZONA_DOM'] = None

# Verificando intervalo de valores - condições:
# "SUBZONA_DOM >= 1" E "SUBZONA_DOM <= 9"
verifica_range(df, 'SUBZONA_DOM', 1, 9)
log_output.info('\n\n=====\\n')

return df


def passo_mun_dom(passo, df):
"""
Checar se existe algum erro

# #####Categorias
# > 1 a 38

[Teste: Checar se existe algum número < 1 ou > 38.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - MUN_DOM")

# Substituindo valor 0 por None
df.loc[df['MUN_DOM']==0,'MUN_DOM'] = None

# Verificando intervalo de valores - condições:
# "MUN_DOM >= 1" E "MUN_DOM <= 38"
verifica_range(df, 'MUN_DOM', 1, 38)
log_output.info('\n\n=====\\n')

return df


def passo_f_dom(passo, df):
"""
Checar se existe algum erro na coluna "F_DOM"

# #####Categorias
# Valor/Descrição
# ----/----
# 0/Demais registros
# 1/Primeiro Registro do Domicílio

[Teste: Checar se existe algum número diferente de 0 ou 1.
 Se encontrar, retornar erro indicando em qual linha.]

```

```

:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - F_DOM")

verifica_dummy(df, 'F_DOM')
log_output.info('\n\n=====\\n')

return df

def passo_fe_dom(passo, df):
"""
Nada há que se fazer em relação aos dados da coluna "FE_DOM"

:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - FE_DOM")
log_output.info('\n\n=====\\n')

return df

def passo_tipo_dom(passo, df):
"""
Substituir **0** por **None (NA)**
Substituir **2** por **0**

# #####Categorias anteriores
# Valor / Descrição
# ----/----
# 1/Individual
# 2/Coletivo

# #####Categorias novas
# Valor / Descrição
# ----/----
# 0/Coletivo
# 1/Particular

[Teste: Checar se existe algum número < 0 ou > 1.
Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return:
"""
log_tela.info("### PASSO " + str(passo) + " - TIPO_DOM")

df.loc[df['TIPO_DOM']==0, 'TIPO_DOM'] = None
df.loc[df['TIPO_DOM']==2, 'TIPO_DOM'] = 0

```

```
# Verificando intervalo de valores - condições:  
# "TIPO_DOM >= 0" E "TIPO_DOM <= 1"  
verifica_dummy(df, 'TIPO_DOM')  
log_output.info('\\n\\n=====\\n')  
  
return df
```

3.4 Funções da Família

Função/Variável	Status
passo_tot_fam	OK
passo_f_fam	OK
passo_fe_fam	OK
passo_cond_mora	OK
passo_qt_auto	OK
passo_qt_bici	OK
passo_qt_moto	OK
passo_ren_fam	OK
passo_cd_renfam	OK

```
In [ ]: log_tela.info('Definindo as funções da família')
log_output.info('\n\n=====\\n')

def passo_tot_fam(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "TOT_FAM"
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - TOT_FAM")
    log_output.info('\\n\\n=====\\n')

    return df

def passo_f_fam(passo, df):
    """
    Checar se existe algum erro na coluna "F_FAM"

    # #####Categorias
    # Valor/Descrição
    # ----/----
    # 0/Demais registros
    # 1/Primeiro Registro da Família

    [Teste: Checar se existe algum número diferente de 0 ou 1.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - F_FAM")

    verifica_dummy(df, 'F_FAM')
```

```

log_output.info('\n\n=====\\n')
return df

def passo_fe_fam(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "FE_FAM"
    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - FE_FAM")
    log_output.info('\n\n=====\\n')

    return df

def passo_cond_mora(passo, df):
    """
    Substituir valores da coluna "COND_MORA"

    * Substituir todos valores **2** por **0**
    * Substituir todos valores **0** por **None**
    * Substituir todos valores **4** por **2**
    * Substituir todos valores **3** por **5**
    * Substituir todos valores **1** por **3**
    * Substituir todos valores **5** por **1**

    ##### Categorias anteriores

    Valor/Descrição
    ----/----
    1/Não se aplica
    2/Não respondeu
    3/Alugada
    4/Casa Própria

    ##### Categorias novas

    Valor/Descrição
    ----/----
    1/Alugada
    2/Própria
    3/Outros

    [Teste: Checar se existe algum número < 1 ou > 3.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: passo
    :param df: dataframe
    :return: dataframe modificado
    """
    log_tela.info("### PASSO " + str(passo) + " - COND_MORA")

```

```

# Substituindo valor 2 por 0
df.loc[df['COND_MORA']==2,'COND_MORA'] = 0
# Substituindo valor 0 por None
df.loc[df['COND_MORA']==0,'COND_MORA'] = None
# Substituindo valor 4 por 2
df.loc[df['COND_MORA']==4,'COND_MORA'] = 2
# Substituindo valor 3 por 5
df.loc[df['COND_MORA']==3,'COND_MORA'] = 5
# Substituindo valor 1 por 3
df.loc[df['COND_MORA']==1,'COND_MORA'] = 3
# Substituindo valor 5 por 1
df.loc[df['COND_MORA']==5,'COND_MORA'] = 1

# Verificando intervalo de valores - condições:
# "COND_MORA >= 1" E "COND_MORA <= 3"
verifica_range(df, 'COND_MORA', 1, 3)
log_output.info('\'\n\'=====\\n')

return df

def passo_qt_auto(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "QT_AUTO"
    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - QT_AUTO")
    log_output.info('\'\n\'=====\\n')

    return df

def passo_qt_bici(passo, df):
    """
    Não existe essa informação no banco de dados de 1987, logo,
    este campo será preenchido com 'None'.
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: dataframe modificado
    """
    df['QT_BICI'] = None

    log_tela.info('\'\n\'=====\\n')
    log_output.info('\'\n\'=====\\n')

    return df

def passo_qt_moto(passo, df):
    """
    Não existe essa informação no banco de dados de 1987, logo,
    este campo será preenchido com 'None'.

```

```

:param passo: Número do passo atual para registro/log
:param df:
:return: dataframe modificado
"""
df['QT_MOTO'] = None

log_tela.info('### PASSO ' + str(passo) + ' - QT_MOTO')
log_output.info('\n\n=====\\n')

return df

def passo_ren_fam(passo, df, deflator):
    """
    Corrigir a renda familiar de acordo com o deflator passado na função.
    :param passo: Número do passo atual para registro/log
    :param df: dataframe
    :param deflator: Deflator utilizado para correção da renda.
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - REN_FAM")

    df['REN_FAM'] = df['REN_FAM'] * deflator

    return df

def passo_cd_renfam(passo, df):
    """
    Substituir valores da coluna "CD_RENFAM"

    * Substituir todos valores **1** por **0**
    * Substituir todos valores **3** por **1**

    ##### Categorias anteriores
    Valor/Descrição
    ----/----
    1/Não Tem Renda
    2/Renda Familiar Incompleta
    3/Renda Familiar Completa

    ##### Categorias novas
    Valor/Descrição
    ----/----
    0/Renda Familiar Declarada como Zero
    1/Renda Familiar Declarada e Maior que Zero
    2/Renda Atribuída

    [Teste: Checar se existe algum número < 0 ou > 2.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe corrigido
    """

```

```

log_tela.info("### PASSO " + str(passo) + " - CD_RENFAM")

# Substituindo valor 1 por 0
df.loc[df['CD_RENFAM']==1,'CD_RENFAM'] = 0
# Substituindo valor 3 por 1
df.loc[df['CD_RENFAM']==3,'CD_RENFAM'] = 1

# Dividindo a categoria '0', "Respondeu", em:
# - 0 "Renda Familiar Declarada como Zero" e
# - 1 "Renda Familiar Declarada e Maior que Zero"
df.loc[(df['CD_RENFAM'] == 0) &
       (df['REN_FAM'] != 0) &
       (df['REN_FAM'].notnull()), 'CD_RENFAM'] = 1

# Verificando intervalo de valores - condições:
# "CD_RENFAM >= 0" E "CD_RENFAM <= 2"
verifica_range(df, 'CD_RENFAM', 0, 2)
log_output.info('\'\n\'=====\\n')

return df

```

3.5 Funções da pessoa

Função/Variável	Status
passo_f_pess	OK
passo_fe_pess	OK
passo_sit_fam	OK
passo_idade	OK
passo_sexo	OK
passo_grau_instr	OK
passo_ocup	OK
passo_setor_ativ	OK
passo_ren_ind	OK
passo_cd_renind	OK
passo_estuda	OK

Obs.: o passo estuda deve ser executado após o passo ZONA_ESC.

```
In [ ]: log_tela.info('Definindo as funções da pessoa')
log_output.info('\n\n=====\\n')

def passo_f_pess(passo, df):
    """
    Checar se existe algum erro na coluna "F_PESS"

    # #####Categorias
    # Valor/Descrição
    # ----/----
    # 0/Demais registros
    # 1/Primeiro Registro da Pessoa

    [Teste: Checar se existe algum número diferente de 0 ou 1.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - F_PESS")

    verifica_dummy(df, 'F_PESS')
    log_output.info('\n\n=====\\n')

    return df

def passo_fe_pess(passo, df):
    """
    Nada há que se fazer em relação aos dados das colunas "FE_PESS"
```

```

:param passo: Número do passo atual para registro/log
:param df:
:return:
"""
log_tela.info("### PASSO " + str(passo) + " - FE_PESS")
log_output.info('\n\n=====\\n')

return df

def passo_sit_fam(passo, df):
    """
    * Substituir todos valores **5** por **4**
    * Substituir todos valores **6** por **5**
    * Substituir todos valores **7** por **6**

    ##### Categorias anteriores
    Valor/Descrição
    ----/----
    1/Chefe
    2/Cônjuge
    3/Filho(a)
    4/Parente
    5/Agregado
    6/Empregado Residente
    7/Visitante

    ##### Categorias novas:
    Valor/Descrição
    ----/----
    0/ Não Respondeu/Não fez viagem
    1/ Pessoa Responsável
    2/ Cônjuge/Companheiro(a)
    3/ Filho(a)/Enteado(a)
    4/ Outro Parente / Agregado
    5/ Empregado Residente
    6/ Outros (visitante não residente / parente do empregado)

[Teste: Checar se existe algum número < 0 ou > 6.
Se encontrar, retornar erro indicando em qual linha.]]

:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe modificado
"""
log_tela.info("### PASSO " + str(passo) + " - SIT_FAM")

# Substituindo valor 5 por 4
df.loc[df['SIT_FAM']==5,'SIT_FAM'] = 4
# Substituindo valor 6 por 5
df.loc[df['SIT_FAM']==6,'SIT_FAM'] = 5
# Substituindo valor 7 por 6
df.loc[df['SIT_FAM']==7,'SIT_FAM'] = 6

# Verificando intervalo de valores - condições:

```

```

        # "SIT_FAM >= 0" E "SIT_FAM <= 6"
verifica_range(df, 'SIT_FAM', 0, 6)
log_output.info('\n\n=====\\n')

return df

def passo_idade(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "IDADE"
    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - IDADE")
    log_output.info('\n\n=====\\n')

return df

def passo_sexo(passo, df):
    """
    #####Categorias anteriores
    # Valor/Descrição
    # ---/---
    # 0/Não Respondeu (-> None)
    # 1/Masculino
    # 2/Feminino

    #####Categorias novas
    # Valor/Descrição
    # ---/---
    # 0/Masculino
    # 1/Feminino

    [Teste: Checar se existe algum número diferente de 0 ou 1.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe modificado
    """
    log_tela.info("### PASSO " + str(passo) + " - SEXO")

    # Substituindo valor 0 por None
    df.loc[df['SEXO']==0,'SEXO'] = None
    # Substituindo valor 1 por 0
    df.loc[df['SEXO']==1,'SEXO'] = 0
    # Substituindo valor 2 por 1
    df.loc[df['SEXO']==2,'SEXO'] = 1

    # Verificando intervalo de valores - condições:
    # "SEXO >= 0" E "SEXO <= 1"
    verifica_dummy(df, 'SEXO')
    log_output.info('\n\n=====\\n')

```

```

    return df

def passo_grau_instr(passo, df):
    """
    Substituir valores da coluna "GRAU_INSTR"

    * Substituir todos valores **2** por **1**
    * Substituir todos valores **3** por **2**
    * Substituir todos valores **4** por **3**
    * Substituir todos valores **5** por **4**
    * Substituir todos valores **0** por **None**

    ##### Categorias anteriores:
    Valor/Descrição
    ----/----
    0/Não declarou
    1/Não-alfabetizado/4ª série Incompleta
    2/4ª Série Completa
    3/1º Grau completo
    4/Colegial completo
    5/Superior Completo

    ##### Categorias novas
    Valor/Descrição
    ----/----
    1/Não-Alfabetizado/Fundamental Incompleto
    2/Fundamental Completo/Médio Incompleto
    3/Médio Completo/Superior Incompleto
    4/Superior completo

    [Teste: Checar se existe algum número < 1 ou > 4.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna dataframe modificado
    """
    log_tela.info("### PASSO " + str(passo) + " - GRAU_INSTR")

    # Substituindo valor 2 por 1
    df.loc[df['GRAU_INSTR']==2, 'GRAU_INSTR'] = 1
    # Substituindo valor 3 por 2
    df.loc[df['GRAU_INSTR']==3, 'GRAU_INSTR'] = 2
    # Substituindo valor 4 por 3
    df.loc[df['GRAU_INSTR']==4, 'GRAU_INSTR'] = 3
    # Substituindo valor 5 por 4
    df.loc[df['GRAU_INSTR']==5, 'GRAU_INSTR'] = 4
    # Substituindo valor 0 por None
    df.loc[df['GRAU_INSTR']==0, 'GRAU_INSTR'] = None

    # Verificando intervalo de valores - condições:
    # "GRAU_INSTR >= 1" E "GRAU_INSTR <= 4"
    verifica_range(df, 'GRAU_INSTR', 1, 4)

```

```

log_output.info('\n\n=====\\n')
return df

def passo_ocup(passo, df):
    """
    Substituir valores da coluna "OCUP"

    Somar 10 em todos valores.
    Substituir todos valores **10** por **None**
    Substituir todos valores **11** por **7**
    Substituir todos valores **12** por **6**
    Substituir todos valores **13** por **3**
    Substituir todos valores **14** por **5**
    Substituir todos valores **15** por **4**
    Substituir todos valores **16** por **2**
    Substituir todos valores *maiores do que 16* por **1**

    # #####Categorias anteriores:
    # Valor/Descrição
    # ----/----
    # 1/Estudante
    # 2/Prendas Domésticas
    # 3/Aposentado
    # 4/Sem Ocupação (nunca trabalhou)
    # 5/Desempregado
    # 6/Em licença
    # 7 em diante/diversas profissões

    # #####Categorias novas
    # Valor/Descrição
    # ----/----
    # 1/Tem trabalho
    # 2/Em licença médica
    # 3/Aposentado / pensionista
    # 4/Desempregado
    # 5/Sem ocupação
    # 6/Dona de casa
    # 7/Estudante

    [Teste: Checar se existe algum número < 1 ou > 7.
     Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna dataframe modificado
"""
log_tela.info("### PASSO " + str(passo) + " - OCUP")

df['OCUP'] = df['OCUP'] + 10
# Substituindo valor 10 por None
df.loc[df['OCUP']==10,'OCUP'] = None
# Substituindo valor 11 por 7
df.loc[df['OCUP']==11,'OCUP'] = 7

```

```

# Substituindo valor 12 por 6
df.loc[df['OCUP']==12,'OCUP'] = 6
# Substituindo valor 13 por 3
df.loc[df['OCUP']==13,'OCUP'] = 3
# Substituindo valor 14 por 5
df.loc[df['OCUP']==14,'OCUP'] = 5
# Substituindo valor 15 por 4
df.loc[df['OCUP']==15,'OCUP'] = 4
# Substituindo valor 16 por 2
df.loc[df['OCUP']==16,'OCUP'] = 2
# Substituindo valor >16 por 1
df.loc[df['OCUP']>16,'OCUP'] = 1

# Verificando intervalo de valores - condições:
# "OCUP >= 1" E "OCUP <= 7"
verifica_range(df, 'OCUP', 1, 7)
log_output.info(' \n\n===== \n')

return df


def passo_setor_ativ(passo, df):
    """
    Substituir valores da coluna "SETOR_ATIV"

    Na coluna "SETOR_ATIV", linha i,
        ler o valor da linha i da coluna "SETOR_ATIV", daí,
        buscar o mesmo valor na coluna "COD" do arquivo setor_ativ-1987.csv.
    Ao achar, retornar o valor da mesma linha, só que da coluna "COD_UNIF"

    #####Categorias anteriores
    > ver arquivo .csv

    #####Categorias novas
    Valor/Descrição
    ----/-----
    1/Agrícola
    2/Construção Civil
    3/Indústria
    4/Comércio
    5/Administração Pública
    6/Serviços de Transporte
    7/Outros serviços
    8/Outros
    9/Não se aplica

    [Teste: Checar se existe algum número < 0 ou > 9.
     Se encontrar, retornar erro indicando em qual linha.]"
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna dataframe modificado
    """
    log_tela.info("### PASSO " + str(passo) + " - SETOR_ATIV")

```

```

log_output.info('Lendo arquivo de referência externa setor_ativ-1987.csv')
df_setor = pd.read_csv('auxiliares/setor_ativ-1987.csv', sep=';', decimal=',')

def setor_aux(row):
    df.loc[df['SETOR_ATIV']==row['COD'], 'SETOR_ATIV'] = row['COD_UNIF']

# Esta variável out não é utilizada para nada além de evitar um
# monte de output que não será utilizado e que é gerado pelo método apply.
out = df_setor.apply(setor_aux, axis=1)

# Substituindo valor 0 por None
df.loc[df['SETOR_ATIV']==0, 'SETOR_ATIV'] = None

# Verificando intervalo de valores - condições:
# "SETOR_ATIV >= 0" E "SETOR_ATIV <= 9"
verifica_range(df, 'SETOR_ATIV', 0, 9)
log_output.info('\n\n=====\\n')

return df

def passo_ren_ind(passo, df, deflator):
    """
    Corriga a renda individual de acordo com o deflator passado na função.
    :param passo: Número do passo atual para registro/log
    :param df: dataframe
    :param deflator: Deflator utilizado para correção da renda.
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - REN_IND")
    df['REN_IND'] = df['REN_IND'] * deflator
    return df

def passo_cd_renind(passo, df):
    """
    Substituir valores da coluna "CD_RENIND"

    * Substituir todos valores **2** por None
    * Substituir todos valores **1** por **0**
    * Substituir todos valores **3** por **1**

    ##### Categorias anteriores:
    Valor/Descrição
    ----/----
    1/Não tem renda
    2/Não Declarou
    3/Declarou

    ##### Categorias novas
    Valor/Descrição
    ----/-----
    """

```

```

0      /Não tem renda
1      /Tem renda

[Teste: Checar se existe algum número < 0 ou > 1.
   Se encontrar, retornar erro indicando em qual linha.]
```

:param passo: Número do passo atual para registro/log

:param df:

:return: sem retorno

"""

```

log_tela.info("### PASSO " + str(passo) + " - CD_RENIND")

df.loc[df['CD_RENIND']==2,'CD_RENIND'] = None
df.loc[df['CD_RENIND']==1,'CD_RENIND'] = 0
df.loc[df['CD_RENIND']==3,'CD_RENIND'] = 1

# Verificando intervalo de valores - condições:
# "CD_RENIND >= 0" E "CD_RENIND <= 1"
verifica_dummy(df, 'CD_RENIND')
log_output.info('\n\n=====\\n')

return df
```

def passo_estuda(passo, df):

"""

Se zona da escola for zero (0) então não estuda (0), senão, estuda (1)

```

#####
#      ATENÇÃO      #
# ESTA FUNÇÃO SÓ DEVE SER #
# EXECUTADA APÓS A GERAÇÃO#
# DE TODOS OS ID'S, POIS #
# HÁ UMA DESCONFIANÇA    #
# QUANTO À QUALIDADE    #
#      DESTE DADO        #
#####

* Substituir todos valores **2** por **0**
```

Categorias anteriores

Valor/Descrição

----/----

1/Sim

2/Não

Categorias novas

Valor/Descrição

----/----

0/Não estuda

1/Estuda

[Teste: Checar se existe algum número diferente de 0 ou 1.
 Se encontrar, retornar erro indicando em qual linha.]

:param passo: Número do passo atual para registro/log

```

:param df:
:return: retorna dataframe com campo ESCOLA resolvido
"""

log_tela.info("### PASSO " + str(passo) + " - ESTUDA")

# Substitui 2 por 0
df.loc[df['ESTUDA'] == 2, 'ESTUDA'] = 0

# Substituindo todos que declararam zona escola diferente de zero
# com campo ESTUDA igual a 1.
df.loc[(df['ZONA_ESC'] != 0)&
       (df['ZONA_ESC'].notnull()), 'ESTUDA'] = 1

verifica_dummy(df, 'ESTUDA')
log_output.info('\n\n=====\\n')

return df

```

3.6 Funções referentes da Viagem

Função/Variável	Status
passo_f_vitag	OK
passo_fe_vitag	OK
passo_zona_esc	OK
passo_subzona_esc	OK
passo_mun_esc	OK
passo_zona_trab1	OK
passo_subzona_trab1	OK
passo_mun_trab1	OK
passo_zona_trab2	OK
passo_subzona_trab2	OK
passo_mun_trab2	OK
passo_zona_orig	OK
passo_subzona_orig	OK
passo_mun_orig	OK
passo_zona_dest	OK
passo_subzona_dest	OK
passo_mun_dest	OK
passo_serv_pas_orig	OK
passo_serv_pas_dest	OK
passo_motivo_orig	OK
passo_motivo_dest	OK
passo_modo1	OK
passo_modo2	OK
passo_modo3	OK
passo_modo4	OK
passo_modo_prin	OK
passo_tipo_est_auto	OK
passo_valor_est_auto	OK
passo_tot_vitag	OK

Obs: O passo_tot_vitag só deve ser executado após a produção dos ID's e NO's.

```
In [ ]: log_tela.info('Definindo funções referentes às viagens')
log_output.info('\n'=====\\n')
```

```
def passo_f_vitag(passo, df):
```

```

"""
Checar se existe algum erro na coluna "F_VIAG"

# #####Categorias
# Valor/Descrição
# ----/----
# 0/Não fez viagem
# 1/Fez viagem

[Teste: Checar se existe algum número diferente de 0 ou 1.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - F_VIAG")

verifica_dummy(df, 'F_VIAG')
log_output.info('\n\n=====\\n')

return df


def passo_fe_viag(passo, df):
"""
# Nada há que se fazer em relação aos dados da coluna "FE_VIAG"
:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - FE_VIAG")
log_output.info('\n\n=====\\n')

return df


def passo_zona_esc(passo, df):
"""
Checar se existe algum erro

# #####Categorias:
# > 1 a 254

[Teste: Checar se existe algum número < 1 ou > 254.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: Sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - ZONA_ESC")

# Substituindo valor 0 por None
df.loc[df['ZONA_ESC']==0, 'ZONA_ESC'] = None

```

```

# Verificando intervalo de valores - condições:
# "ZONA_ESC >= 1" E "ZONA_ESC <= 254"
verifica_range(df, 'ZONA_ESC', 1, 254)
log_output.info('\'\n\'=====\'\n')

return df


def passo_subzona_esc(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 9

    [Teste: Checar se existe algum número < 1 ou > 9.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - SUBZONA_ESC")

    # Substituindo valor 0 por None
    df.loc[df['SUBZONA_ESC']==0,'SUBZONA_ESC'] = None

    # Verificando intervalo de valores - condições:
    # "SUBZONA_ESC >= 1" E "SUBZONA_ESC <= 9"
    verifica_range(df, 'SUBZONA_ESC', 1, 9)
    log_output.info('\'\n\'=====\'\n')

    return df


def passo_mun_esc(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias
    # > 1 a 38

    [Teste: Checar se existe algum número < 1 ou > 38.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - MUN_ESC")

    # Substituindo valor 0 por None
    df.loc[df['MUN_ESC']==0,'MUN_ESC'] = None

    # Verificando intervalo de valores - condições:
    # "MUN_ESC >= 1" E "MUN_ESC <= 38"

```

```

verifica_range(df, 'MUN_ESC', 1, 38)
log_output.info('\n\n=====\\n')

return df

def passo_zona_trab1(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 254

    [Teste: Checar se existe algum número < 1 ou > 254.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - ZONA_TRAB1")

    # Substituindo valor 0 por None
    df.loc[df['ZONA_TRAB1']==0,'ZONA_TRAB1'] = None

    # Verificando intervalo de valores - condições:
    # "ZONA_TRAB1 >= 1" E "ZONA_TRAB1 <= 254"
    verifica_range(df, 'ZONA_TRAB1', 1, 254)
    log_output.info('\n\n=====\\n')

    return df

def passo_subzona_trab1(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 9

    [Teste: Checar se existe algum número < 1 ou > 9.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - SUBZONA_TRAB1")

    # Substituindo valor 0 por None
    df.loc[df['SUBZONA_TRAB1']==0,'SUBZONA_TRAB1'] = None

    # Verificando intervalo de valores - condições:
    # "SUBZONA_TRAB1 >= 1" E "SUBZONA_TRAB1 <= 9"
    verifica_range(df, 'SUBZONA_TRAB1', 1, 9)
    log_output.info('\n\n=====\\n')

```

```

    return df

def passo_mun_trab1(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias
    # > 1 a 38

    [Teste: Checar se existe algum número < 1 ou > 38.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - MUN_TRAB1")

    # Substituindo valor 0 por None
    df.loc[df['MUN_TRAB1']==0,'MUN_TRAB1'] = None

    # Verificando intervalo de valores - condições:
    # "MUN_TRAB1 >= 1" E "MUN_TRAB1 <= 38"
    verifica_range(df, 'MUN_TRAB1', 1, 38)
    log_output.info('\n\n=====\\n')

    return df

def passo_zona_trab2(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 254

    [Teste: Checar se existe algum número < 1 ou > 254.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - ZONA_TRAB2")

    # Substituindo valor 0 por None
    df.loc[df['ZONA_TRAB2']==0,'ZONA_TRAB2'] = None

    # Verificando intervalo de valores - condições:
    # "ZONA_TRAB2 >= 1" E "ZONA_TRAB2 <= 254"
    verifica_range(df, 'ZONA_TRAB2', 1, 254)
    log_output.info('\n\n=====\\n')

    return df

```

```

def passo_subzona_trab2(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 9

    [Teste: Checar se existe algum número < 1 ou > 9.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - SUBZONA_TRAB2")

    # Substituindo valor 0 por None
    df.loc[df['SUBZONA_TRAB2']==0, 'SUBZONA_TRAB2'] = None

    # Verificando intervalo de valores - condições:
    # "SUBZONA_TRAB2 >= 1" E "SUBZONA_TRAB2 <= 9"
    verifica_range(df, 'SUBZONA_TRAB2', 1, 9)
    log_output.info('\n\n=====\\n')

    return df


def passo_mun_trab2(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias
    # > 1 a 38

    [Teste: Checar se existe algum número < 1 ou > 38.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - MUN_TRAB2")

    # Substituindo valor 0 por None
    df.loc[df['MUN_TRAB2']==0, 'MUN_TRAB2'] = None

    # Verificando intervalo de valores - condições:
    # "MUN_TRAB2 >= 1" E "MUN_TRAB2 <= 38"
    verifica_range(df, 'MUN_TRAB2', 1, 38)
    log_output.info('\n\n=====\\n')

    return df


def passo_zona_orig(passo, df):

```

```

"""
Checar se existe algum erro

# #####Categorias:
# > 1 a 254

[Teste: Checar se existe algum número < 1 ou > 254.
 Se encontrar, retornar erro indicando em qual linha.] 
:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""

log_tela.info("### PASSO " + str(passo) + " - ZONA_ORIG")

# Substituindo valor 0 por None
df.loc[df['ZONA_ORIG']==0,'ZONA_ORIG'] = None

# Verificando intervalo de valores - condições:
# "ZONA_ORIG >= 1" E "ZONA_ORIG <= 254"
verifica_range(df, 'ZONA_ORIG', 1, 254)
log_output.info('\n\n=====\\n')

return df


def passo_subzona_orig(passo, df):
"""
Checar se existe algum erro

# #####Categorias:
# > 1 a 9

[Teste: Checar se existe algum número < 1 ou > 9.
 Se encontrar, retornar erro indicando em qual linha.] 
:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""

log_tela.info("### PASSO " + str(passo) + " - SUBZONA_ORIG")

# Substituindo valor 0 por None
df.loc[df['SUBZONA_ORIG']==0,'SUBZONA_ORIG'] = None

# Verificando intervalo de valores - condições:
# "SUBZONA_ORIG >= 1" E "SUBZONA_ORIG <= 9"
verifica_range(df, 'SUBZONA_ORIG', 1, 9)
log_output.info('\n\n=====\\n')

return df


def passo_mun_orig(passo, df):
"""
Checar se existe algum erro

```

```

# #####Categorias
# > 1 a 38

[Teste: Checar se existe algum número < 1 ou > 38.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - MUN_ORIG")

# Substituindo valor 0 por None
df.loc[df['MUN_ORIG']==0,'MUN_ORIG'] = None

# Verificando intervalo de valores - condições:
# "MUN_ORIG >= 1" E "MUN_ORIG <= 38"
verifica_range(df, 'MUN_ORIG', 1, 38)
log_output.info('\n\n=====\\n')

return df


def passo_zona_dest(passo, df):
"""
Checar se existe algum erro

# #####Categorias:
# > 1 a 254

[Teste: Checar se existe algum número < 1 ou > 254.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - ZONA_DEST")

# Substituindo valor 0 por None
df.loc[df['ZONA_DEST']==0,'ZONA_DEST'] = None

# Verificando intervalo de valores - condições:
# "ZONA_DEST >= 1" E "ZONA_DEST <= 254"
verifica_range(df, 'ZONA_DEST', 1, 254)
log_output.info('\n\n=====\\n')

return df


def passo_subzona_dest(passo, df):
"""
Checar se existe algum erro

# #####Categorias:

```

```

# > 1 a 9

[Teste: Checar se existe algum número < 1 ou > 9.
 Se encontrar, retornar erro indicando em qual linha.]
```

:param passo: Número do passo atual para registro/log

:param df:

:return: sem retorno

"""

```
log_tela.info("### PASSO " + str(passo) + " - SUBZONA_DEST")
```

Substituindo valor 0 por None

```
df.loc[df['SUBZONA_DEST']==0,'SUBZONA_DEST'] = None
```

Verificando intervalo de valores - condições:

"SUBZONA_DEST >= 1" E "SUBZONA_DEST <= 9"

```
verifica_range(df, 'SUBZONA_DEST', 1, 9)
```

```
log_output.info('\n\n=====\\n')
```

```
return df
```

def passo_mun_dest(passo, df):

"""

Checar se existe algum erro

#####Categorias

> 1 a 38

[*Teste: Checar se existe algum número < 1 ou > 38.*
 Se encontrar, retornar erro indicando em qual linha.]

:param passo: Número do passo atual para registro/log

:param df:

:return: sem retorno

"""

```
log_tela.info("### PASSO " + str(passo) + " - MUN_DEST")
```

Substituindo valor 0 por None

```
df.loc[df['MUN_DEST']==0,'MUN_DEST'] = None
```

Verificando intervalo de valores - condições:

"MUN_DEST >= 1" E "MUN_DEST <= 38"

```
verifica_range(df, 'MUN_DEST', 1, 38)
```

```
log_output.info('\n\n=====\\n')
```

```
return df
```

def passo_serv_pas_orig(passo, df):

"""

*Substituir **0** por None*

*Substituir **2** por **0***

#####Categorias antigas

Valor/Descrição

```

-----/-----
0/Não Respondido
1/Sim
2/Não

#####Categorias novas
Valor/Descrição
-----/-----
0/Não
1/Sim

:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - SERV_PAS_ORIG")

df.loc[df['SERV_PAS_ORIG']==0, 'SERV_PAS_ORIG'] = None
df.loc[df['SERV_PAS_ORIG']==2, 'SERV_PAS_ORIG'] = 0

verifica_dummy(df, 'SERV_PAS_ORIG')
log_output.info('\n\n=====\\n')

return df


def passo_serv_pas_dest(passo, df):
"""
Substituir **0** por None
Substituir **2** por **0**

#####Categorias antigas
Valor/Descrição
-----/-----
0/Não Respondido
1/Sim
2/Não

#####Categorias novas
Valor/Descrição
-----/-----
0/Não
1/Sim

:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - SERV_PAS_DEST")

df.loc[df['SERV_PAS_DEST']==0, 'SERV_PAS_DEST'] = None
df.loc[df['SERV_PAS_DEST']==2, 'SERV_PAS_DEST'] = 0

verifica_dummy(df, 'SERV_PAS_DEST')

```

```

log_output.info('\n\n=====\\n')
return df

def passo_motivo_orig(passo, df):
    """
    * Substituir todos valores **6** por **10**
    * Substituir todos valores **7** por **6**
    * Substituir todos valores **8** por **7**
    * Substituir todos valores **9** por **8**
    * Substituir todos valores **10** por **9**
    * Substituir todos valores **0** por **None**

    ##### Categorias anteriores
    Valor/Descrição
    ----/----
    1/Trabalho/Indústria
    2/Trabalho/Comércio
    3/Trabalho/Serviços
    4/Escola/Educação
    5/Compras
    6/Negócios
    7/Médico/Dentista/Saúde
    8/Recreação/Visitas
    9/Residência

    ##### Categorias novas
    Valor/Descrição
    ----/----
    1/Trabalho/Indústria
    2/Trabalho/Comércio
    3/Trabalho/Serviços
    4/Educação
    5/Compras
    6/Saúde
    7/Lazer
    8/Residência
    9/Outros

    [Teste: Checar se existe algum número < 1 ou > 9.
     Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: dataframe modificado
"""
log_tela.info("### PASSO " + str(passo) + " - MOTIVO_ORIG")

# Substituindo valor 6 por 10
df.loc[df['MOTIVO_ORIG']==6, 'MOTIVO_ORIG'] = 10
# Substituindo valor 7 por 6
df.loc[df['MOTIVO_ORIG']==7, 'MOTIVO_ORIG'] = 6
# Substituindo valor 8 por 7
df.loc[df['MOTIVO_ORIG']==8, 'MOTIVO_ORIG'] = 7

```

```

# Substituindo valor 9 por 8
df.loc[df['MOTIVO_ORIG']==9,'MOTIVO_ORIG'] = 8
# Substituindo valor 10 por 9
df.loc[df['MOTIVO_ORIG']==10,'MOTIVO_ORIG'] = 9
# Substituindo valor 0 por None
df.loc[df['MOTIVO_ORIG']==0,'MOTIVO_ORIG'] = None

# Verificando intervalo de valores - condições:
# "MOTIVO_ORIG >= 1" E "MOTIVO_ORIG <= 9"
verifica_range(df, 'MOTIVO_ORIG', 1, 9)
log_output.info('\n\n=====\\n')

return df

def passo_motivo_dest(passo, df):
"""
Substituir todos valores **6** por **10**
Substituir todos valores **7** por **6**
Substituir todos valores **8** por **7**
Substituir todos valores **9** por **8**
Substituir todos valores **10** por **9**
Substituir todos valores **0** por **None**

#### Categorias anteriores
Valor/Descrição
----/----
1/Trabalho/Indústria
2/Trabalho/Comércio
3/Trabalho/Serviços
4/Escola/Educação
5/Compras
6/Negócios
7/Médico/Dentista/Saúde
8/Recreação/Visitas
9/Residência

#### Categorias novas
Valor/Descrição
----/----
1/Trabalho/Indústria
2/Trabalho/Comércio
3/Trabalho/Serviços
4/Educação
5/Compras
6/Saúde
7/Lazer
8/Residência
9/Outros

[Teste: Checar se existe algum número < 1 ou > 9.
Se encontrar, retornar erro indicando em qual linha.] 
:param passo: Número do passo atual para registro/log
:param df:

```

```

: return: dataframe modificado
"""

log_tela.info("### PASSO " + str(passo) + " - MOTIVO_DEST")

# Substituindo valor 6 por 10
df.loc[df['MOTIVO_DEST']==6,'MOTIVO_DEST'] = 10
# Substituindo valor 7 por 6
df.loc[df['MOTIVO_DEST']==7,'MOTIVO_DEST'] = 6
# Substituindo valor 8 por 7
df.loc[df['MOTIVO_DEST']==8,'MOTIVO_DEST'] = 7
# Substituindo valor 9 por 8
df.loc[df['MOTIVO_DEST']==9,'MOTIVO_DEST'] = 8
# Substituindo valor 10 por 9
df.loc[df['MOTIVO_DEST']==10,'MOTIVO_DEST'] = 9
# Substituindo valor 0 por None
df.loc[df['MOTIVO_DEST']==0,'MOTIVO_DEST'] = None

# Verificando intervalo de valores - condições:
# "MOTIVO_DEST >= 1" E "MOTIVO_DEST <= 9"
verifica_range(df, 'MOTIVO_DEST', 1, 9)
log_output.info('\n\n=====\\n')

return df

def passo_modo1(passo, df):
    """
    Substituir valores da coluna "MOD01"

    * Substituir todos valores **2** por **1**
    * Substituir todos valores **3** por **2**
    * Substituir todos valores **4** por **2**
    * Substituir todos valores **5** por **3**
    * Substituir todos valores **6** por **4**
    * Substituir todos valores **7** por **5**
    * Substituir todos valores **8** por **6**
    * Substituir todos valores **9** por **7**
    * Substituir todos valores **10** por **8**
    * Substituir todos valores **11** por **9**
    * Substituir todos valores **12** por **10**
    * Substituir todos valores **13** por **11**
    * Substituir todos valores **14** por **12**
    * Substituir todos valores **15** por **12**
    * Substituir todos valores **0** por **None**

    #### Categorias anteriores
    Valor/Descrição
    ----/----
    1/Ônibus diesel
    2/Trólebus
    3/Ônibus Fretado
    4/Escolar
    5/Dirigindo Automóvel
    6/Passageiro de Automóvel

```

```

7/Táxi
8/Lotação/Perua
9/Metrô
10/Trem
11/Moto
12/Bicicleta
13/A Pé
14/Caminhão
15/Outros

##### Categorias novas
Valor/Descrição
----/----
1/Ônibus
2/Ônibus Escolar / Empresa
3/Dirigindo Automóvel
4/Passageiro de Automóvel
5/Táxi
6/Lotação / Perua / Van / Microônibus
7/Metrô
8/Trem
9/Moto
10/Bicicleta
11/A Pé
12/Outros

[Teste: Checar se existe algum número < 1 ou > 12.
Se encontrar, retornar erro indicando em qual linha.]
```

:param passo: Número do passo atual para registro/log

:param df:

:return: sem retorno

"""

```

log_tela.info("### PASSO " + str(passo) + " - MODO1")

df.loc[df['MODO1']==2,'MODO1'] = 1
df.loc[df['MODO1']==3,'MODO1'] = 2
df.loc[df['MODO1']==4,'MODO1'] = 2
df.loc[df['MODO1']==5,'MODO1'] = 3
df.loc[df['MODO1']==6,'MODO1'] = 4
df.loc[df['MODO1']==7,'MODO1'] = 5
df.loc[df['MODO1']==8,'MODO1'] = 6
df.loc[df['MODO1']==9,'MODO1'] = 7
df.loc[df['MODO1']==10,'MODO1'] = 8
df.loc[df['MODO1']==11,'MODO1'] = 9
df.loc[df['MODO1']==12,'MODO1'] = 10
df.loc[df['MODO1']==13,'MODO1'] = 11
df.loc[df['MODO1']==14,'MODO1'] = 12
df.loc[df['MODO1']==15,'MODO1'] = 12
df.loc[df['MODO1']==0,'MODO1'] = None

# Verificando intervalo de valores - condições:
# "MODO1 >= 1" E "MODO1 <= 12"
verifica_range(df, 'MODO1', 1, 12)
log_output.info('\'\n\'=====\\n')
```

```

    return df

def passo_modo2(passo, df):
    """
    Substituir valores da coluna "MODO2"

    * Substituir todos valores **2** por **1**
    * Substituir todos valores **3** por **2**
    * Substituir todos valores **4** por **2**
    * Substituir todos valores **5** por **3**
    * Substituir todos valores **6** por **4**
    * Substituir todos valores **7** por **5**
    * Substituir todos valores **8** por **6**
    * Substituir todos valores **9** por **7**
    * Substituir todos valores **10** por **8**
    * Substituir todos valores **11** por **9**
    * Substituir todos valores **12** por **10**
    * Substituir todos valores **13** por **11**
    * Substituir todos valores **14** por **12**
    * Substituir todos valores **15** por **12**
    * Substituir todos valores **0** por **None**

    ##### Categorias anteriores
    Valor/Descrição
    ----/----
    1/Ônibus diesel
    2/Trólebus
    3/Ônibus Fretado
    4/Escolar
    5/Dirigindo Automóvel
    6/Passageiro de Automóvel
    7/Táxi
    8/Lotação/Perua
    9/Metrô
    10/Trem
    11/Moto
    12/Bicicleta
    13/A Pé
    14/Caminhão
    15/Outros

    ##### Categorias novas
    Valor/Descrição
    ----/----
    1/Ônibus
    2/Ônibus Escolar / Empresa
    3/Dirigindo Automóvel
    4/Passageiro de Automóvel
    5/Táxi
    6/Lotação / Perua / Van / Microônibus
    7/Metrô
    8/Trem

```

9/Moto
 10/Bicicleta
 11/A Pé
 12/Outros

[Teste: Checar se existe algum número < 1 ou > 12.
 Se encontrar, retornar erro indicando em qual linha.]

```

:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""

log_tela.info("### PASSO " + str(passo) + " - MODO3")

df.loc[df['MODO2']==2,'MODO2'] = 1
df.loc[df['MODO2']==3,'MODO2'] = 2
df.loc[df['MODO2']==4,'MODO2'] = 2
df.loc[df['MODO2']==5,'MODO2'] = 3
df.loc[df['MODO2']==6,'MODO2'] = 4
df.loc[df['MODO2']==7,'MODO2'] = 5
df.loc[df['MODO2']==8,'MODO2'] = 6
df.loc[df['MODO2']==9,'MODO2'] = 7
df.loc[df['MODO2']==10,'MODO2'] = 8
df.loc[df['MODO2']==11,'MODO2'] = 9
df.loc[df['MODO2']==12,'MODO2'] = 10
df.loc[df['MODO2']==13,'MODO2'] = 11
df.loc[df['MODO2']==14,'MODO2'] = 12
df.loc[df['MODO2']==15,'MODO2'] = 12
df.loc[df['MODO2']==0,'MODO2'] = None

# Verificando intervalo de valores - condições:
# "MODO2 >= 1" E "MODO2 <= 12"
verifica_range(df, 'MODO2', 1, 12)
log_output.info('\n\n=====\\n')

return df
  
```

def passo_modo3(passo, df):

Substituir valores da coluna "MODO3"

- * Substituir todos valores **2** por **1**
- * Substituir todos valores **3** por **2**
- * Substituir todos valores **4** por **2**
- * Substituir todos valores **5** por **3**
- * Substituir todos valores **6** por **4**
- * Substituir todos valores **7** por **5**
- * Substituir todos valores **8** por **6**
- * Substituir todos valores **9** por **7**
- * Substituir todos valores **10** por **8**
- * Substituir todos valores **11** por **9**
- * Substituir todos valores **12** por **10**
- * Substituir todos valores **13** por **11**
- * Substituir todos valores **14** por **12**

```
* Substituir todos valores **15** por **12**
* Substituir todos valores **0** por **None**
```

```
##### Categorias anteriores
```

```
Valor/Descrição
```

```
----/----
```

```
1/Ônibus diesel
```

```
2/Trólebus
```

```
3/Ônibus Fretado
```

```
4/Escolar
```

```
5/Dirigindo Automóvel
```

```
6/Passageiro de Automóvel
```

```
7/Táxi
```

```
8/Lotação/Perua
```

```
9/Metrô
```

```
10/Trem
```

```
11/Moto
```

```
12/Bicicleta
```

```
13/A Pé
```

```
14/Caminhão
```

```
15/Outros
```

```
##### Categorias novas
```

```
Valor/Descrição
```

```
----/----
```

```
1/Ônibus
```

```
2/Ônibus Escolar / Empresa
```

```
3/Dirigindo Automóvel
```

```
4/Passageiro de Automóvel
```

```
5/Táxi
```

```
6/Lotação / Perua / Van / Microônibus
```

```
7/Metrô
```

```
8/Trem
```

```
9/Moto
```

```
10/Bicicleta
```

```
11/A Pé
```

```
12/Outros
```

```
[Teste: Checar se existe algum número < 1 ou > 12.
```

```
Se encontrar, retornar erro indicando em qual linha.]
```

```
:param passo: Número do passo atual para registro/log
```

```
:param df:
```

```
:return: sem retorno
```

```
"""
```

```
log_tela.info("### PASSO " + str(passo) + " - MOD03")
```

```
df.loc[df['MOD03']==2,'MOD03'] = 1
```

```
df.loc[df['MOD03']==3,'MOD03'] = 2
```

```
df.loc[df['MOD03']==4,'MOD03'] = 2
```

```
df.loc[df['MOD03']==5,'MOD03'] = 3
```

```
df.loc[df['MOD03']==6,'MOD03'] = 4
```

```
df.loc[df['MOD03']==7,'MOD03'] = 5
```

```
df.loc[df['MOD03']==8,'MOD03'] = 6
```

```
df.loc[df['MOD03']==9,'MOD03'] = 7
```

```

df.loc[df['MODO3']==10,'MODO3'] = 8
df.loc[df['MODO3']==11,'MODO3'] = 9
df.loc[df['MODO3']==12,'MODO3'] = 10
df.loc[df['MODO3']==13,'MODO3'] = 11
df.loc[df['MODO3']==14,'MODO3'] = 12
df.loc[df['MODO3']==15,'MODO3'] = 12
df.loc[df['MODO3']==0,'MODO3'] = None

# Verificando intervalo de valores - condições:
# "MODO3 >= 1" E "MODO3 <= 12"
verifica_range(df, 'MODO3', 1, 12)
log_output.info('\n\n=====\\n')

return df

def passo_modo4(passo, df):
    """
    Nada há que se fazer em relação à coluna "MODO4" - não há dados de 1987,
    coluna permanecerá vazia
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - MODO4")

    df['MODO4'] = None

    log_output.info('\n\n=====\\n')

    return df

def passo_modo_prin(passo, df):
    """
    Substituir valores da coluna "MODO_PRIN"

    * Substituir todos valores **2** por **1**
    * Substituir todos valores **3** por **2**
    * Substituir todos valores **4** por **2**
    * Substituir todos valores **5** por **3**
    * Substituir todos valores **6** por **4**
    * Substituir todos valores **7** por **5**
    * Substituir todos valores **8** por **6**
    * Substituir todos valores **9** por **7**
    * Substituir todos valores **10** por **8**
    * Substituir todos valores **11** por **9**
    * Substituir todos valores **12** por **10**
    * Substituir todos valores **13** por **11**
    * Substituir todos valores **14** por **12**
    * Substituir todos valores **15** por **12**
    * Substituir todos valores **0** por **None**

    #### Categorias anteriores

```

Valor/Descrição

----/----

1/*Ônibus diesel*
 2/*Trólebus*
 3/*Ônibus Fretado*
 4/*Escolar*
 5/*Dirigindo Automóvel*
 6/*Passageiro de Automóvel*
 7/*Táxi*
 8/*Lotação/Perua*
 9/*Metrô*
 10/*Trem*
 11/*Moto*
 12/*Bicicleta*
 13/*A Pé*
 14/*Caminhão*
 15/*Outros*

Categorias novas

Valor/Descrição

----/----

1/*Ônibus*
 2/*Ônibus Escolar / Empresa*
 3/*Dirigindo Automóvel*
 4/*Passageiro de Automóvel*
 5/*Táxi*
 6/*Lotação / Perua / Van / Microônibus*
 7/*Metrô*
 8/*Trem*
 9/*Moto*
 10/*Bicicleta*
 11/*A Pé*
 12/*Outros*

[*Teste: Checar se existe algum número < 1 ou > 12.*
Se encontrar, retornar erro indicando em qual linha.]

```
:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""

log_tela.info("### PASSO " + str(passo) + " - MODO_PRIN")

df.loc[df['MODO_PRIN']==2, 'MODO_PRIN'] = 1
df.loc[df['MODO_PRIN']==3, 'MODO_PRIN'] = 2
df.loc[df['MODO_PRIN']==4, 'MODO_PRIN'] = 2
df.loc[df['MODO_PRIN']==5, 'MODO_PRIN'] = 3
df.loc[df['MODO_PRIN']==6, 'MODO_PRIN'] = 4
df.loc[df['MODO_PRIN']==7, 'MODO_PRIN'] = 5
df.loc[df['MODO_PRIN']==8, 'MODO_PRIN'] = 6
df.loc[df['MODO_PRIN']==9, 'MODO_PRIN'] = 7
df.loc[df['MODO_PRIN']==10, 'MODO_PRIN'] = 8
df.loc[df['MODO_PRIN']==11, 'MODO_PRIN'] = 9
df.loc[df['MODO_PRIN']==12, 'MODO_PRIN'] = 10
df.loc[df['MODO_PRIN']==13, 'MODO_PRIN'] = 11
```

```

df.loc[df['MODO_PRIN']==14, 'MODO_PRIN'] = 12
df.loc[df['MODO_PRIN']==15, 'MODO_PRIN'] = 12
df.loc[df['MODO_PRIN']==0, 'MODO_PRIN'] = None

# Verificando intervalo de valores - condições:
# "MODO_PRIN >= 1" E "MODO_PRIN <= 12"
verifica_range(df, 'MODO_PRIN', 1, 12)
log_output.info('\n\n=====\\n')

return df

def passo_tipo_vitag(passo, df):
    """
    * Substituir os valores **0** por **None**

    # #####Categorias novas
    # Valor/Descrição
    # ----/----
    # 1/Coletivo
    # 2/Individual
    # 3/A pé

    [Teste: Checar se existe algum número < 1 ou > 3.
     Se encontrar, retornar erro indicando em qual linha.]"""

:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno

:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""

log_tela.info("### PASSO " + str(passo) + " - TIPO_VIAG")

# Substituindo valor 0 por None
df.loc[df['TIPO_VIAG']==0, 'TIPO_VIAG'] = None

# Verificando intervalo de valores - condições:
# "MODO_PRIN >= 1" E "MODO_PRIN <= 3"
verifica_range(df, 'TIPO_VIAG', 1, 3)

log_output.info('\n\n=====\\n')
return df

def passo_tipo_est_auto(passo, df):
    """
    Substituir valores da coluna "TIPO_EST_AUTO"

    * Substituir todos valores **1** por **5**
    * Substituir todos valores **6** por **1**

    # #####Categorias anteriores
    Valor/Descrição

```

```

----/----
1/Zona Azul / Parquímetro
2/Estacionamento Particular
3/Estacionamento Próprio
4/Estacionamento Patrocinado
5/Meio-Fio
6/Não estacionou

# #####Categorias novas
# Valor/Descrição
# ----/----
# 0/Não Respondeu
# 1/Não Estacionou
# 2/Estacionamento Particular (Avulso / Mensal)
# 3/Estacionamento Próprio
# 4/Estacionamento Patrocinado
# 5/Rua (meio fio / zona azul / zona marrom / parquímetro)

[Teste: Checar se existe algum número < 0 ou > 5.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna dataframe modificado
"""
log_tela.info("### PASSO " + str(passo) + " - TIPO_EST_AUTO")

# Substituindo valor 1 por 5
df.loc[df['TIPO_EST_AUTO']==1, 'TIPO_EST_AUTO'] = 5
# Substituindo valor 6 por 1
df.loc[df['TIPO_EST_AUTO']==6, 'TIPO_EST_AUTO'] = 1

# Verificando intervalo de valores - condições:
# "TIPO_EST_AUTO >= 0" E "TIPO_EST_AUTO <= 5"
verifica_range(df, 'TIPO_EST_AUTO', 0, 5)
log_output.info('\n\n=====\\n')

return df

def passo_valor_est_auto(passo, df, deflator):
"""
Nada há que se fazer em relação à coluna "VALOR_EST_AUTO".
Não há dados de 1987, coluna permanecerá vazia
:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - VALOR_EST_AUTO")

df['VALOR_EST_AUTO'] = None

log_output.info('\n\n=====\\n')

```

```

    return df

def passo_tot_viang(passo, df):
    """
    ##### ATENÇÃO #####
    # ESTA FUNÇÃO SÓ DEVE SER #
    # EXECUTADA APÓS A GERAÇÃO#
    # DE TODOS OS ID'S, POIS #
    # HÁ UMA DESCONFIANÇA   #
    # QUANTO À QUALIDADE   #
    # DESTE DADO           #
    #####
    Calcula e confere o campo TOT_VIAG,
    baseado no maior valor de NO_VIAG para cada pessoa
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe com o "TOT_VIAG" corrigido
"""
log_tela.info("### PASSO " + str(passo) + " - TOT_VIAG")
log_tela.warning('\n\n#####\n' +
                 'Este passo DEVE ser executado apenas após a \n' +
                 'execução dos passos que geram os novos IDs\n' +
                 '#####\n')

# 'Calculando' o máximo de viagens para cada indivíduo
# Agrupa por ID_PESS e encontra o máximo para NO_VIAG.
# O resultado é um objeto do tipo "Series" cujo index é
# ID_PESS e a variável é NO_VIAG. Em seguida transforma
# esse objeto num DataFrame e depois renomeia a coluna
# NO_VIAG para MAX_VIAG.
df_max = pd.DataFrame(
    df.groupby('ID_PESS')['NO_VIAG'].max()).rename(
        columns={'NO_VIAG': 'MAX_VIAG'})
# Criando um novo dataframe auxiliar apenas com ID_PESS, apenas para ficar mais leve

# O passo seguinte é atribuir o MAX_VIAG à coluna TOT_VIAG
# do dataframe df, linkando por ID_PESS. Isso é feito usando o
# método 'merge' da biblioteca pandas.
df['TOT_VIAG'] = pd.merge(df, df_max, how='left', left_on='ID_PESS',
                           right_index=True)[['MAX_VIAG']]

log_output.info('TOT_VIAG:\n\n' +
                '      Situação final dos dados: \n' +
                str(df['TOT_VIAG'].describe()) + '\n' +
                '      TOT_VIAG: Situação final dos dados: \n' +
                str(df['TOT_VIAG'].value_counts()))

# Agora uma função que irá verificar se para todo "ID_PESS" o "TOT_VIAG"
# é igual ao 'NO_VIAG' máximo.
def verifica_no_viang_tot_viang(row):
    if row['NO_VIAG'] != row['TOT_VIAG']:
        log_output.warning('TOT_VIAG: Erro encontrado na linha '

```

```
+ str(row) + ':\\n'
+ ' => ' + row
)
df.loc[:, ['ID_PESS', 'NO_VIAG', 'TOT_VIAG']]\
.groupby('ID_PESS')\
.agg({'NO_VIAG': 'max', 'ID_PESS': 'max', 'TOT_VIAG': 'max'})\
.apply(verifica_no_viag_tot_viag, axis=1)
log_output.info('\\n\\n=====\\n')
return df
```

3.7 Funções que geram os “NO”s e os “ID”s

Função/Variável	Status
gera_nos_e_ids	Verificar NO VIAG

```
In [ ]: log_tela.info('Definindo funções que geram os "NO"s e os "ID"s')
log_output.info('\n\n=====\\n')
```

```
def gera_nos_e_ids(passo, df):
    """
    Esta função gera todos os IDs e todos os NOs das variáveis:
    DOM (domicílio), FAM (família), PESS (pessoa) e VIAG (viagem)
    A ordem de geração é:
    NO_DOM, ID_DOM, NO_FAM, ID_FAM, NO_PESS, ID_PESS, NO_VIAG, ID_VIAG
    Esta ordem é fixa pois cada uma dessas variáveis depende da geração
    da variável anterior.

    Os NOs são gerados como se fossem subíndices.
    No caso dos domicílios, eles são contabilizados dentro de cada zona.
    Assim, cada novo domicílio que aparece na listagem recebe um número
    que representa sua posição na sequência de domicílios dentro da zona
    na qual ele está contido.
    As famílias seguem o mesmo raciocínio, com relação ao domicílio, as
    pessoas com relação às famílias e as viagens com relação às pessoas.

    Deve-se apenas tomar cuidado com as viagens, pois existem pessoas que
    não realizaram viagens. Neste caso, estes registros devem contabilizar
    o NO_VIAG como zero, e não como 1. Para contemplar estes casos, vamos
    utilizar a variável F_VIAG, que representa quando há ou não viagem
    naquele registro (F_VIAG==1 tem viagem, F_VIAG==0 não tem viagem).
    Assim, o que precisamos fazer para calcular o NO_VIAG é agrupar os
    registros por pessoa (ID_PESS) e, dentro de cada agrupamento, somar
    o valor de F_VIAG registro a registro (linha por linha)
    cumulativamente. No final do processo, atribui-se zero a NO_VIAG
    quando F_VIAG for igual a zero.
    """

def gera_id_dom(row):
    """
    Gera o ID_DOM baseado no 'ANO', na 'ZONA_DOM' e no 'NO_DOM'
    O argumento passado é a "linha".
    Uso:
        df['ID_DOM'] = df.apply(gera_ID_DOM, axis=1)
    Retorna: ID_DOM da respectiva linha
    """
    # Formatando o ano para string
    ano = str(row['ANO'])

    # Formatando a zona para ficar como string com 3 caracteres
    zona = str('%03d' % row['ZONA_DOM'])
```

```

# Formatando no_dom para ficar como string com 4 caracteres
no_dom = str('%04d' % row['NO_DOM'])

# Retornando o número inteiro correspondente à string
# concatenada de ano + zona + no_dom
return ano + zona + no_dom


def gera_id_fam(row):
    """
    Gera o ID_FAM baseado no 'ID_DOM' e no 'NO_FAM'
    O argumento passado é a "linha".
    Uso:
        df['ID_FAM'] = df.apply(gera_ID_FAM, axis=1)
    Retorna: ID_FAM da respectiva linha
    """
    # Formatando id_dom como string
    id_dom = str(row['ID_DOM'])

    # Formatando no_fam como string para ficar com 2 caracteres
    no_fam = str('%02d' % row['NO_FAM'])

    # Retornando o número inteiro correspondente à string
    # concatenada de ID_DOM com NO_FAM
    return id_dom + no_fam


def gera_id_pess(row):
    """
    Gera o ID_PESS baseado no 'ID_FAM' e no 'NO_PESS'
    O argumento passado é a "linha".
    Uso:
        df['ID_PESS'] = df.apply(gera_ID_PESS, axis=1)
    Retorna: ID_PESS da respectiva linha
    """
    # Formatando id_fam como string
    id_fam = str(row['ID_FAM'])

    # Formatando no_pess como string para ficar com 2 caracteres
    no_pess = str('%02d' % row['NO_PESS'])

    # Retornando o número inteiro correspondente à string
    # concatenada de ID_FAM com NO_PESS
    return id_fam + no_pess


def gera_id_viag(row):
    """
    Gera o ID_VIAG baseado no 'ID_PESS' e no 'NO_VIAG'
    O argumento passado é a "linha".
    Uso:
        df['ID_VIAG'] = df.apply(gera_ID_VIAG, axis=1)
    Retorna ID_VIAG da respectiva linha
    """
    # Formatando id_pess como string
    id_pess = str(row['ID_PESS'])

```

```

# Formatando no_viag como string para ficar com 2 caracteres
no_viag = str('%02d' % row['NO_VIAG'])

# Retornando o número inteiro correspondente à string
# concatenada de ID_PESS com NO_VIAG
return id_pess + no_viag

#gera NO_DOM
log_tela.info("Gerando NO_DOM")
df['NO_DOM'] = df.groupby('ZONA_DOM', sort=False)['F_DOM'].cumsum()
log_tela.info("NO_DOM gerado")
log_tela.info("Gerando ID_DOM")
#gera ID_DOM
df['ID_DOM'] = df.apply(gera_id_dom, axis=1)
log_tela.info("ID_DOM gerado")

#gera NO_FAM
log_tela.info("Gerando NO_FAM")
df['NO_FAM'] = df.groupby('ID_DOM', sort=False)['F_FAM'].cumsum()
log_tela.info("NO_FAM gerado")
log_tela.info("Gerando ID_FAM")
#gera ID_FAM
df['ID_FAM'] = df.apply(gera_id_fam, axis=1)
log_tela.info("ID_FAM gerado")

#gera NO_PESS
log_tela.info("Gerando NO_PESS")
df['NO_PESS'] = df.groupby('ID_FAM', sort=False)['F_PESS'].cumsum()
log_tela.info("NO_PESS gerado")
log_tela.info("Gerando ID_PESS")
#gera ID_PESS
df['ID_PESS'] = df.apply(gera_id_pess, axis=1)
log_tela.info("ID_PESS gerado")

#gera NO_VIAG
log_tela.info("Gerando NO_VIAG")
df['NO_VIAG'] = df.groupby('ID_PESS', sort=False)['F_VIAG'].cumsum()
# Verificando se FE_VIAG == 0 e zerando NO_VIAG nesse caso
df.loc[df['F_VIAG'] == 0, 'NO_VIAG'] = 0
log_tela.info("NO_VIAG gerado")
log_tela.info("Gerando ID_VIAG")
#gera ID_VIAG
df['ID_VIAG'] = df.apply(gera_id_viag, axis=1)
log_tela.info("ID_VIAG gerado")

return df

```

3.8 Função relativa à entrevista

Função/Variável	Status
passo_cd_entre	OK

Obs: o passo_cd_entre deve ser executado após o passo_tot_vitag.

```
In [ ]: log_tela.info('Definindo função relativa à entrevista')
log_output.info('\n\n=====\\n')

def passo_cd_entre(passo, df):
    """
    #####
    #      ATENÇÃO      #
    # ESTA FUNÇÃO SÓ DEVE SER #
    # EXECUTADA APÓS A GERAÇÃO#
    #      DO TOT_VIAG     #
    #####
    ----
    Substituir valores da coluna "CD_ENTRE"
    Todas viagens são consideradas "completas", segundo informações do Metrô

    * sem viagem: se TOT_VIAG == 0
    * com viagem: se TOT_VIAG != 0

    # #####Categorias novas
    # | Valor | Descrição |
    # | ----- | ----- |
    # | 0 | Completa sem viagem |
    # | 1 | Completa com viagem |

    [Teste: Checar se existe algum número diferente de 0 ou 1.
     Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return:
"""

log_tela.info("### PASSO " + str(passo) + " - CD_ENTRE")
log_tela.warning('\n\n#####\n' + 'Este passo DEVE ser executado apenas após a\n' +
                 'execução do passo que geram o TOT_VIAG.\n' +
                 '#####\n')

# Definindo 'CD_ENTRE' baseado no valor de 'TOT_VIAG'
df.loc[df['TOT_VIAG'] == 0, 'CD_ENTRE'] = 0
df.loc[df['TOT_VIAG'] != 0, 'CD_ENTRE'] = 1

verifica_dummy(df, 'CD_ENTRE')
log_output.info('\n\n=====\\n')

return df
```

```
In [ ]: def passo_correcoes(passo, df):
    log_tela.info("### PASSO " + str(passo) + " - Correções")
    log_tela.info("Corrigindo 'SERV_PAS' caso TOT_VIAG seja nulo.")
    df.loc[df['TOT_VIAG'] == 0, 'SERV_PAS_ORIG'] = 0
    df.loc[df['TOT_VIAG'] == 0, 'SERV_PAS_DEST'] = 0

    return df
```

4 Definindo a função principal (main)

Esta função vai ler os arquivos (csv) necessários e, então, irá efetuar cada passo, chamando as respectivas funções que foram definidas acima. Ao final, esta função irá salvar o resultado das transformações realizadas num arquivo csv.

```
In [ ]: def main():
    start_time = time.time()

    log_tela.info("Horário de início da execução: %s" %
                  time.strftime("%H:%M", time.localtime(start_time)))

    # ----
    log_tela.info('Lendo arquivos CSV')

    log_output.info('Lendo CSV da base original da OD.')
    od = pd.read_csv('bases/OD_1987.csv', sep=';', decimal=',')

    deflator = 0.0966466592714303
    log_output.info('Deflator a ser utilizado para correção monetária: ' +
                    str(deflator))

    # Filtrando dataframe para pegar apenas uma amostra
    #od = od[:1000].copy()

    log_output.info('\n\n=====\\n')
    log_tela.info('Pré-Processamento.')
    od = pre_processamento(od)

    log_tela.info('Imprimindo a lista de variáveis do dataframe da OD.')
    log_tela.debug(od.columns.tolist())

    #Contador de 'PASSO'
    passo = 1

    # ----
    # ##Passo: "ANO"
    od = passo_ano(passo, od)
    passo += 1

    # ----
    # ##Passo: "DIA_SEM"
    od = passo_dia_sem(passo, od)
    passo += 1

    # ----
    # ##Passo: "UCODs"
    od = passo_ucods(passo, od)
    passo += 1

    # ----
    # ##Passo: "ZONA_DOM"
    od = passo_zona_dom(passo, od)
    passo += 1
```

```

# -----
# ##Passo: "SUBZONA_DOM"
od = passo_subzona_dom(passo, od)
passo += 1

# -----
# ##Passo: "MUN_DOM"
od = passo_mun_dom(passo, od)
passo += 1

# -----
# ##Passo: "F_DOM"
od = passo_f_dom(passo, od)
passo += 1

# -----
# ##Passo: "FE_DOM"
od = passo_fe_dom(passo, od)
passo += 1

# -----
# ##Passo: "TIPO_DOM"
od = passo_tipo_dom(passo, od)
passo += 1

# -----
# ##Passo: "TOT_FAM"
od = passo_tot_fam(passo, od)
passo += 1

# -----
# ##Passo: "F_FAM"
od = passo_f_fam(passo, od)
passo += 1

# -----
# ##Passo: "FE_FAM"
od = passo_fe_fam(passo, od)
passo += 1

# -----
# ##Passo: "COND_MORA"
od = passo_cond_mora(passo, od)
passo += 1

# -----
# ##Passo: "QT_AUTO"
od = passo_qt_auto(passo, od)
passo += 1

# -----
# ##Passo: "QT_BICI"
od = passo_qt_bici(passo, od)
passo += 1

```

```

# -----
# ##Passo: "QT_MOTO"
od = passo_qt_moto(passo, od)
passo += 1

# -----
# ##PASSO: "REN_FAM"
od = passo_ren_fam(passo, od, deflator)
passo += 1

# -----
# ##Passo: "CD_RENFAM"
od = passo_cd_renfam(passo, od)
passo += 1

# -----
# ##Passo: "F_PESS"
od = passo_f_pess(passo, od)
passo += 1

# -----
# ##Passo: "FE_PESS"
od = passo_fe_pess(passo, od)
passo += 1

# -----
# ##Passo: "SIT_FAM"
od = passo_sit_fam(passo, od)
passo += 1

# -----
# ##Passo: "IDADE"
od = passo_idade(passo, od)
passo += 1

# -----
# ##Passo: "SEXO"
od = passo_sexo(passo, od)
passo += 1

# -----
# ##Passo: "GRAU_INSTR"
od = passo_grau_instr(passo, od)
passo += 1

# -----
# ##Passo: "OCUP"
od = passo_ocup(passo, od)
passo += 1

# -----
# ##Passo: "SETOR_ATIV"
od = passo_setor_ativ(passo, od)

```

```

    passo += 1

    # ##Passo: "REN_IND"
    od = passo_ren_ind(passo, od, deflator)
    passo += 1

    # ##Passo: "CD_RENIND"
    od = passo_cd_renind(passo, od)
    passo += 1

    # -----
    # ##Passo: "F_VIAG"
    od = passo_f_viag(passo, od)
    passo += 1

    # -----
    # ##Passo: "FE_VIAG"
    od = passo_fe_viag(passo, od)
    passo += 1

    # -----
    # ##Passo: "ZONA_ESC"
    od = passo_zona_esc(passo, od)
    passo += 1

    # -----
    # ##Passo: "SUBZONA_ESC"
    od = passo_subzona_esc(passo, od)
    passo += 1

    # -----
    # ##Passo: "MUN_ESC"
    od = passo_mun_esc(passo, od)
    passo += 1

    # -----
    # ##Passo: "ESTUDA"
    # Este passo deve vir após os passos de localização da escola
    od = passo_estuda(passo, od)
    passo += 1

    # -----
    # ##Passo: "ZONA_TRAB1"
    od = passo_zona_trab1(passo, od)
    passo += 1

    # -----
    # ##Passo: "SUBZONA_TRAB1"
    od = passo_subzona_trab1(passo, od)
    passo += 1

    # -----
    # ##Passo: "MUN_TRAB1"
    od = passo_mun_trab1(passo, od)

```

```

    passo += 1

    # -----
    # ##Passo: "ZONA_TRAB2"
    od = passo_zona_trab2(passo, od)
    passo += 1

    # -----
    # ##Passo: "SUBZONA_TRAB2"
    od = passo_subzona_trab2(passo, od)
    passo += 1

    # -----
    # ##Passo: "MUN_TRAB2"
    od = passo_mun_trab2(passo, od)
    passo += 1

    # -----
    # ##Passo: "ZONA_ORIG"
    od = passo_zona_orig(passo, od)
    passo += 1

    # -----
    # ##Passo: "SUBZONA_ORIG"
    od = passo_subzona_orig(passo, od)
    passo += 1

    # -----
    # ##Passo: "MUN_ORIG"
    od = passo_mun_orig(passo, od)
    passo += 1

    # -----
    # ##Passo: "ZONA_DEST"
    od = passo_zona_dest(passo, od)
    passo += 1

    # -----
    # ##Passo: "SUBZONA_DEST"
    od = passo_subzona_dest(passo, od)
    passo += 1

    # -----
    # ##Passo: "MUN_DEST"
    od = passo_mun_dest(passo, od)
    passo += 1

    # -----
    # ##Passo: "SERV_PAS_ORIG"
    od = passo_serv_pas_orig(passo, od)
    passo += 1

    # -----
    # ##Passo: "SERV_PAS_DEST"

```

```

od = passo_serv_pas_dest(passo, od)
passo += 1

# -----
# ##Passo: "MOTIVO_ORIG"
od = passo_motivo_orig(passo, od)
passo += 1

# -----
# ##Passo: "MOTIVO_DEST"
od = passo_motivo_dest(passo, od)
passo += 1

# -----
# ##Passo: "MOD01"
od = passo_modo1(passo, od)
passo += 1

# -----
# ##Passo: "MOD02"
od = passo_modo2(passo, od)
passo += 1

# -----
# ##Passo: "MOD03"
od = passo_modo3(passo, od)
passo += 1

# -----
# ##Passo: "MOD04"
od = passo_modo4(passo, od)
passo += 1

# -----
# ##Passo: "MODO_PRIN"
od = passo_modo_prin(passo, od)
passo += 1

# -----
# ##Passo: "TIPO_VIAG"
od = passo_tipo_vitag(passo, od)
passo += 1

# -----
# ##"H_SAIDA"; "MIN_SAIDA"; "ANDA_ORIG"; "H_CHEG"; "MIN_CHEG";
# "ANDA_DEST"; "DIST_VIAG" e "DURACAO"
# Nada há que se fazer em relação aos dados das colunas acima mencionadas

# -----
# ##Passo: "TIPO_EST_AUTO"
od = passo_tipo_est_auto(passo, od)
passo += 1

# -----

```

```

# ##Passo: "VALOR_EST_AUTO"
od = passo_valor_est_auto(passo, od, deflator)
passo += 1

# -----
# ##Passo: Coordenadas
od = coordenadas(passo, od)
passo += 1

## O passo TOT_VIAG apenas é chamado após a geração dos IDs.

# -----
# ##Passo: Gerando NO's e ID's
od = gera_nos_e_ids(passo, od)
passo += 1

# -----
# ##Passo: "TOT_VIAG"
od = passo_tot_viag(passo, od)
passo += 1

# -----
# ##Passo: "CD_ENTRE"
od = passo_cd_entre(passo, od)
passo += 1

# -----
# ##Passo: Correções
od = passo_correcoes(passo, od)
passo += 1

log_tela.info('Salvando dataframe como arquivo CSV')
# -----
# ## Salvando o dataframe num arquivo local
od.to_csv('outputs/1987_od.csv', sep=';', decimal=',', index=False)

log_tela.info("Base gerada. Arquivo: outputs/1987_od.csv")

log_tela.info("Tempo total de execução: %s segundos" %
              (time.time() - start_time))

log_tela.info("Horário de finalização: %s" %
              (time.strftime("%H:%M", time.localtime(time.time()))))

log_tela.info("Terminou o main")

```

5 RUN the main() function....

```
In [ ]: if __name__ == "__main__":
    main()
```

rotina_final_1997

January 4, 2016

1 Setup Inicial

```
In [ ]: # coding: utf-8
        import math
        import logging
        import time
        import pandas as pd

        # Faz os gráficos um pouco mais bonitos
        pd.set_option('display.mpl_style', 'default')
```

2 Definindo Loggers

Define os loggers

Estes ‘loggers’ serão utilizados para salvar as saídas (outputs) em um arquivo de texto no diretório ‘outputs’.

ref: <http://stackoverflow.com/questions/17035077/python-logging-to-multiple-log-files-from-different-classes>

ATENÇÃO: RODAR O BLOCO ABAIXO APENAS UMA VEZ, SE ESTE BLOCO FOR EXECUTADO MAIS DE UMA VEZ OS LOGs SERÃO DUPLICADOS.

```
In [ ]: log_formatter = logging.Formatter('%(asctime)s | %(levelname)s: %(message)s')

log_output = logging.getLogger('log_output')
FH_output = logging.FileHandler(
            'outputs/1997_output.log', mode='w')
FH_output.setFormatter(log_formatter)
log_output.setLevel(logging.INFO)
log_output.addHandler(FH_output)
log_output.propagate = False

# Este logger (log_tela) loga na tela e também
# no arquivo de output, junto com o conteúdo do
# logger log_output
log_tela = logging.getLogger('log_tela')
SH_tela = logging.StreamHandler()
SH_tela.setFormatter(log_formatter)
log_tela.addHandler(SH_tela)
log_tela.addHandler(FH_output)
log_tela.setLevel(logging.INFO)
log_tela.propagate = False
```

3 Funções de 1997

3.1 Funções gerais assessórias

Função	Status
consulta_refext	OK
verifica_dummy	OK
verifica_range	OK
pre_processamento	OK
coordenadas	OK

```
In [ ]: log_tela.info('Definindo as funções gerais assessórias')
log_output.info('\n\n=====\\n')

def verifica_dummy(df, variavel):
    """
    Verifica se uma variável, dummy, contém algum valor diferente de 0 ou de 1.
    :param df: dataframe com os dados a serem verificados
    :param variavel: string com o nome da variável (coluna) que tem os dados
        que devem ser verificados.
    :return: Sem retorno, apenas salva as avaliações nos logs.
    Uso:
        verifica_dummy(dataframe, 'coluna a ser verificada')
    """
    contador_de_erros = 0
    log_tela.info('Verificando a variável Dummy: ' + variavel)

    df_erros = df[(df[variavel] != 0) & (df[variavel] != 1)]
    if len(df_erros[variavel].value_counts()) > 0:
        log_tela.warning(variavel + ": " +
                          str(len(df_erros[variavel].value_counts())) +
                          " erros encontrados:\\n" +
                          str(df_erros[variavel].value_counts()))
    else:
        log_tela.info(variavel + ": Nenhum erro encontrado.")

def verifica_range(df, variavel, valor_menor, valor_maior):
    """
    Verifica se uma variável, do tipo número inteiro, contém algum valor menor
    que "valor_menor" ou maior que "valor_maior".
    :param df: dataframe com os dados a serem verificados
    :param variavel: string com o nome da variável (coluna) que tem os dados
        que devem ser verificados
    :param valor_menor: Valor mínimo esperado na variável (int ou float)
    :param valor_maior: Valor máximo esperado na variável (int ou float)
    :return: Sem retorno, apenas salva as avaliações nos logs.
    Uso:
        verifica_range(dataframe, 'nome_variavel', valor_menor, valor_maior)
```

```

"""
log_tela.info('Verificando Range da variável: ' + variavel)
# Obs: Registros inválidos: None (equivalente a NA)
nulos = df[variavel].isnull().sum()
nulos = nulos if nulos else 0
log_output.info('\n\n' +
    ' - ' + 'Mínimo esperado: ' + str(valor_menor) + '\n' +
    ' - ' + 'Máximo esperado: ' + str(valor_maior) + '\n' +
    ' - ' + 'Total de registros: ' + str(len(df[variavel])) +
    '\n' +
    ' - ' + 'Registros nulos (NA): ' +
    str(df[variavel].isnull().sum()) + '\n'
)
df_erro = df[(df[variavel] < valor_menor) | (df[variavel] > valor_maior)]

if len(df_erro[variavel].value_counts()) > 0:

    result = df_erro[variavel].value_counts().sort_index()
    # Verificando limite inferior
    if result.first_valid_index() < valor_menor:
        log_tela.warning(
            variavel + ': ' + 'Valor inteiro mínimo encontrado: ' +
            str(result.first_valid_index()) + " - abaixo do esperado!")
    # Verificando limite superior
    if result.last_valid_index() > valor_maior:
        log_tela.warning(
            variavel + ': ' + 'Valor inteiro máximo encontrado: ' +
            str(result.last_valid_index()) + " - acima do esperado!")

    log_tela.warning(variavel + ': ' +
        str(len(df_erro[variavel].value_counts())) +
        ' valor(es) incorreto(s)' +
        'encontrado(s) nesta variável:\n' +
        str(df_erro[variavel].value_counts()))
else:
    log_tela.info(variavel + ": Nenhum erro encontrado.")

def pre_processamento(df):
"""
Realiza algumas ações prévias ao processamento dos dados,
removendo alguns registros e ajustando o dataframe.
"""

log_tela.info('Criando/Renomeando colunas no dataframe principal')

log_output.info('Renomeando coluna UCOD para UCOD_DOM')
df.rename(columns={'UCOD': 'UCOD_DOM'}, inplace=True)

log_output.info('Renomeando coluna ANDA_CHEG para ANDA_DEST')
df.rename(columns={'ANDA_CHEG': 'ANDA_DEST'}, inplace=True)

log_tela.info('Verificando se todas as variáveis esperadas' +

```

```

        'existem no dataframe.\n' +
        'Caso não exista alguma, ela é criada.')
variaveis = ['ANO', 'CD_ENTRE', 'DIA_SEM', 'UCOD_DOM', 'ZONA_DOM',
             'SUBZONA_DOM', 'MUN_DOM', 'CO_DOM_X', 'CO_DOM_Y', 'ID_DOM',
             'F_DOM', 'FE_DOM', 'NO_DOM', 'TIPO_DOM', 'TOT_FAM', 'ID_FAM',
             'F_FAM', 'FE_FAM', 'NO_FAM', 'COND_MORA', 'QT_AUTO', 'QT_BICI',
             'QT_MOTO', 'CD_RENFAM', 'REN_FAM', 'ID_PESS', 'F_PESS',
             'FE_PESS', 'NO_PESS', 'SIT_FAM', 'IDADE', 'SEXO', 'ESTUDA',
             'GRAU_INSTR', 'OCUP', 'SETOR_ATIV', 'CD_RENIND', 'REN_IND',
             'UCOD_ESC', 'ZONA_ESC', 'SUBZONA_ESC', 'MUN_ESC', 'CO_ESC_X',
             'CO_ESC_Y', 'UCOD_TRAB1', 'ZONA_TRAB1', 'SUBZONA_TRAB1',
             'MUN_TRAB1', 'CO_TRAB1_X', 'CO_TRAB1_Y', 'UCOD_TRAB2',
             'ZONA_TRAB2', 'SUBZONA_TRAB2', 'MUN_TRAB2', 'CO_TRAB2_X',
             'CO_TRAB2_Y', 'ID_VIAG', 'F_VIAG', 'FE_VIAG', 'NO_VIAG',
             'TOT_VIAG', 'UCOD_ORIG', 'ZONA_ORIG', 'SUBZONA_ORIG',
             'MUN_ORIG', 'CO_ORIG_X', 'CO_ORIG_Y', 'UCOD_DEST', 'ZONA_DEST',
             'SUBZONA_DEST', 'MUN_DEST', 'CO_DEST_X', 'CO_DEST_Y',
             'DIST_VIAG', 'SERV_PAS_ORIG', 'SERV_PAS_DEST', 'MOTIVO_ORIG',
             'MOTIVO_DEST', 'MODO1', 'MODO2', 'MODO3', 'MODO4', 'MODO_PRIN',
             'TIPO_VIAG', 'H_SAIDA', 'MIN_SAIDA', 'ANDA_ORIG', 'H_CHEG',
             'MIN_CHEG', 'ANDA_DEST', 'DURACAO', 'TIPO_EST_AUTO',
             'VALOR_EST_AUTO']

for variavel in variaveis:
    if variavel not in df.columns:
        # Se a variável não existe no dataframe então ela é criada
        # com valor padrão de NONE (NA)
        df[variavel] = None
    log_tela.info('Criando a variavel ' + variavel + ' no dataframe')

log_output.info('Reordenando as variáveis')
df = df[['ANO', 'CD_ENTRE', 'DIA_SEM', 'UCOD_DOM', 'ZONA_DOM',
          'SUBZONA_DOM', 'MUN_DOM', 'CO_DOM_X', 'CO_DOM_Y', 'ID_DOM',
          'F_DOM', 'FE_DOM', 'NO_DOM', 'TIPO_DOM', 'TOT_FAM', 'ID_FAM',
          'F_FAM', 'FE_FAM', 'NO_FAM', 'COND_MORA', 'QT_AUTO', 'QT_BICI',
          'QT_MOTO', 'CD_RENFAM', 'REN_FAM', 'ID_PESS', 'F_PESS',
          'FE_PESS', 'NO_PESS', 'SIT_FAM', 'IDADE', 'SEXO', 'ESTUDA',
          'GRAU_INSTR', 'OCUP', 'SETOR_ATIV', 'CD_RENIND', 'REN_IND',
          'UCOD_ESC', 'ZONA_ESC', 'SUBZONA_ESC', 'MUN_ESC', 'CO_ESC_X',
          'CO_ESC_Y', 'UCOD_TRAB1', 'ZONA_TRAB1', 'SUBZONA_TRAB1',
          'MUN_TRAB1', 'CO_TRAB1_X', 'CO_TRAB1_Y', 'UCOD_TRAB2',
          'ZONA_TRAB2', 'SUBZONA_TRAB2', 'MUN_TRAB2', 'CO_TRAB2_X',
          'CO_TRAB2_Y', 'ID_VIAG', 'F_VIAG', 'FE_VIAG', 'NO_VIAG',
          'TOT_VIAG', 'UCOD_ORIG', 'ZONA_ORIG', 'SUBZONA_ORIG',
          'MUN_ORIG', 'CO_ORIG_X', 'CO_ORIG_Y', 'UCOD_DEST', 'ZONA_DEST',
          'SUBZONA_DEST', 'MUN_DEST', 'CO_DEST_X', 'CO_DEST_Y',
          'DIST_VIAG', 'SERV_PAS_ORIG', 'SERV_PAS_DEST', 'MOTIVO_ORIG',
          'MOTIVO_DEST', 'MODO1', 'MODO2', 'MODO3', 'MODO4', 'MODO_PRIN',
          'TIPO_VIAG', 'H_SAIDA', 'MIN_SAIDA', 'ANDA_ORIG', 'H_CHEG',
          'MIN_CHEG', 'ANDA_DEST', 'DURACAO', 'TIPO_EST_AUTO',
          'VALOR_EST_AUTO']]

log_output.info('\n\n=====\\n')
log_output.info('\\n\\n=====\\n')

```

```

    return df

def coordenadas(passo, df):
    """
    Itera sobre os registros do dataframe coords,
    a cada iteração, utiliza o valor das colunas ZONA e SUBZONA
    para filtrar o dataframe df em cada um dos tipos de ZONA E
    SUBZONA (DOM, ESC, TRAB1, TRAB2, ORIG, DEST) e substitui os valores
    das coordenadas X e Y de cada tipo (CO_DOM_X, CO_ESC_Y, etc) com
    os valores das colunas CO_X e CO_Y do dataframe coords.

    :param passo: número do passo para o log
    :param df: Dafaframe a ser modificado (ex.: od1997)
    :return: retorna o dataframe modificado com todas as coordenadas aplicadas
    """
    log_tela.info("### PASSO " + str(passo) + " - COORDENADAS")

    def coord_aux(row):
        """
        Esta função irá receber uma linha (row) do dataframe coords e
        utilizar os valores desta linha para realizar as alterações
        de coordenadas no dataframe df.
        """
        # Começando pelo tipo DOM, alterando CO_DOM_X e depois CO_DOM_Y
        df.loc[(df['ZONA_DOM'] == row['ZONA']) &
               (df['SUBZONA_DOM'] == row['SUBZONA']), 'CO_DOM_X'] = row['CO_X']
        df.loc[(df['ZONA_DOM'] == row['ZONA']) &
               (df['SUBZONA_DOM'] == row['SUBZONA']), 'CO_DOM_Y'] = row['CO_Y']
        # Agora com o tipo ESC, alterando CO_ESC_X e depois CO_ESC_Y
        df.loc[(df['ZONA_ESC'] == row['ZONA']) &
               (df['SUBZONA_ESC'] == row['SUBZONA']), 'CO_ESC_X'] = row['CO_X']
        df.loc[(df['ZONA_ESC'] == row['ZONA']) &
               (df['SUBZONA_ESC'] == row['SUBZONA']), 'CO_ESC_Y'] = row['CO_Y']
        # Agora com o tipo TRAB1, alterando CO_TRAB1_X e depois CO_TRAB1_Y
        df.loc[(df['ZONA_TRAB1'] == row['ZONA']) &
               (df['SUBZONA_TRAB1'] == row['SUBZONA']), 'CO_TRAB1_X'] = row['CO_X']
        df.loc[(df['ZONA_TRAB1'] == row['ZONA']) &
               (df['SUBZONA_TRAB1'] == row['SUBZONA']), 'CO_TRAB1_Y'] = row['CO_Y']
        # Agora com o tipo TRAB2, alterando CO_TRAB2_X e depois CO_TRAB2_Y
        df.loc[(df['ZONA_TRAB2'] == row['ZONA']) &
               (df['SUBZONA_TRAB2'] == row['SUBZONA']), 'CO_TRAB2_X'] = row['CO_X']
        df.loc[(df['ZONA_TRAB2'] == row['ZONA']) &
               (df['SUBZONA_TRAB2'] == row['SUBZONA']), 'CO_TRAB2_Y'] = row['CO_Y']
        # Agora com o tipo ORIG, alterando CO_ORIG_X e depois CO_ORIG_Y
        df.loc[(df['ZONA_ORIG'] == row['ZONA']) &
               (df['SUBZONA_ORIG'] == row['SUBZONA']), 'CO_ORIG_X'] = row['CO_X']
        df.loc[(df['ZONA_ORIG'] == row['ZONA']) &
               (df['SUBZONA_ORIG'] == row['SUBZONA']), 'CO_ORIG_Y'] = row['CO_Y']
        # Agora com o tipo DEST, alterando CO_DEST_X e depois CO_DEST_Y
        df.loc[(df['ZONA_DEST'] == row['ZONA']) &
               (df['SUBZONA_DEST'] == row['SUBZONA']), 'CO_DEST_X'] = row['CO_X']
        df.loc[(df['ZONA_DEST'] == row['ZONA']) &
               (df['SUBZONA_DEST'] == row['SUBZONA']), 'CO_DEST_Y'] = row['CO_Y']

```

```
(df['SUBZONA_DEST'] == row['SUBZONA']), 'CO_DEST_Y'] = row['CO_Y']

log_output.info('Lendo arquivo auxiliar de Coordenadas das subzonas')
coords = pd.read_csv('auxiliares/coord_subzonas_1997.csv', sep=';', decimal=',')

# Esta variável out não é utilizada para nada além de evitar um
# monte de output que não será utilizado e que é gerado pelo método apply.
out = coords.apply(coord_aux, axis=1)

log_output.info('\n\n=====\\n')

return df
```

3.2 Funções Gerais

Função/Variável	Status
passo_ano	OK
passo_dia_sem	OK
passo_ucods	OK

```
In [ ]: log_tela.info('Definindo as funções gerais')
log_output.info('\n'=====\\n')

def passo_ano(passo, df):
    """
    Preenche a coluna "ANO" com valor 3 em todas células
    Categorias:
    /valor/ano_correspondente/
    /-----/-----
    /1/1977/
    /2/1987/
    /3/1997/
    /4/2007/

    :param passo: Número do passo atual para registro/log
    :param df: dataframe a ser modificado
    :return: retorna o dataframe modificado
    """

    log_tela.info("### PASSO " + str(passo) + " - ANO")

    # Definindo valor '3' para todas as células da coluna ANO
    df["ANO"] = 3

    return df

def passo_dia_sem(passo, df):
    """
    Apenas verificar os valores existentes
    # #####Categorias:
    # Valor/Descrição
    # -----/-----
    # 2/Segunda-Feira
    # 3/Terça-Feira
    # 4/Quarta-Feira
    # 5/Quinta-Feira
    # 6/Sexta-Feira

    :param passo: Número do passo atual para registro/log
    :param df: dataframe a ser modificado
    :return: retorna o dataframe modificado
    """


```

```

log_tela.info("### PASSO " + str(passo) + " - DIA_SEM")

# Substituindo **0** por **None**
df.loc[df['DIA_SEM']==0,'DIA_SEM'] = None

# Verificando intervalo de valores - condições:
# "DIA_SEM >= 2" E "DIA_SEM <= 6"
verifica_range(df, 'DIA_SEM', 2, 6)
log_output.info('\n\n=====\\n')

return df

def passo_ucods(passo, df):
    """
    Itera sobre os registros do dataframe ucods,
    a cada iteração, utiliza o valor da coluna ZONA_REF
    para filtrar o dataframe df em cada um dos tipos de ZONA
    (DOM, ESC, TRAB1, TRAB2, ORIG, DEST) e substitui o valor
    da respectiva UCOD.

    :param passo: número do passo para o log
    :param df: Dataframe a ser modificado (ex.: od1997)
    :return: retorna o dataframe modificado com todas as UCODS aplicadas
    """
    log_tela.info("### PASSO " + str(passo) + " - UCODS")

    def ucod_aux(row):
        df.loc[df['ZONA_DOM']==row['ZONA_REF'], 'UCOD_DOM'] = row['UCOD_BUSCADA']
        df.loc[df['ZONA_ESC']==row['ZONA_REF'], 'UCOD_ESC'] = row['UCOD_BUSCADA']
        df.loc[df['ZONA_TRAB1']==row['ZONA_REF'], 'UCOD_TRAB1'] = row['UCOD_BUSCADA']
        df.loc[df['ZONA_TRAB2']==row['ZONA_REF'], 'UCOD_TRAB2'] = row['UCOD_BUSCADA']
        df.loc[df['ZONA_ORIG']==row['ZONA_REF'], 'UCOD_ORIG'] = row['UCOD_BUSCADA']
        df.loc[df['ZONA_DEST']==row['ZONA_REF'], 'UCOD_DEST'] = row['UCOD_BUSCADA']

    log_output.info('Lendo arquivo auxiliar UCOD')
    ucods = pd.read_csv('auxiliares/UCOD-1997.csv', sep=';', decimal=',')

    # Esta variável out não é utilizada para nada além de evitar um
    # monte de output que não será utilizado e que é gerado pelo método apply.
    out = ucods.apply(ucod_aux, axis=1)

    # Verificando intervalo de valores - condições:
    # "UCOD_XXX >= 1" E "UCOD_XXX <= 67"
    for tipo in ['DOM', 'ESC', 'TRAB1', 'TRAB2', 'ORIG', 'DEST']:
        verifica_range(df, 'UCOD_' + tipo, 1, 67)
    log_output.info('\n\n=====\\n')

    return df

```

3.3 Funções do Domicílio

Função/Variável	Status
passo_zona_dom	OK
passo_subzona_dom	OK
passo_mun_dom	OK
passo_f_dom	OK
passo_fe_dom	OK
passo_tipo_dom	OK

```
In [ ]: log_tela.info('Definindo as funções do domicílio')
log_output.info('\n\n=====\\n')

def passo_zona_dom(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 389

    [Teste: Checar se existe algum número < 1 ou > 389.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: passo
    :param df: dataframe
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - ZONA_DOM")

    # Substituindo valor 0 por None
    df.loc[df['ZONA_DOM']==0,'ZONA_DOM'] = None

    # Verificando intervalo de valores - condições:
    # "ZONA_DOM >= 1" E "ZONA_DOM <= 389"
    verifica_range(df, 'ZONA_DOM', 1, 389)
    log_output.info('\n\n=====\\n')

    return df

def passo_subzona_dom(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 9

    [Teste: Checar se existe algum número < 1 ou > 9.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
```

```

:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - SUBZONA_DOM")

# Substituindo valor 0 por None
df.loc[df['SUBZONA_DOM']==0,'SUBZONA_DOM'] = None

# Verificando intervalo de valores - condições:
# "SUBZONA_DOM >= 1" E "SUBZONA_DOM <= 9"
verifica_range(df, 'SUBZONA_DOM', 1, 9)
log_output.info('\n\n=====\\n')

return df


def passo_mun_dom(passo, df):
"""
Checar se existe algum erro

# #####Categorias
# > 1 a 39

[Teste: Checar se existe algum número < 1 ou > 39.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - MUN_DOM")

# Substituindo valor 0 por None
df.loc[df['MUN_DOM']==0,'MUN_DOM'] = None

# Verificando intervalo de valores - condições:
# "MUN_DOM >= 1" E "MUN_DOM <= 39"
verifica_range(df, 'MUN_DOM', 1, 39)
log_output.info('\n\n=====\\n')

return df


def passo_f_dom(passo, df):
"""
Checar se existe algum erro na coluna "F_DOM"

# #####Categorias
# Valor/Descrição
# ----/----
# 0/Demais registros
# 1/Primeiro Registro do Domicílio

[Teste: Checar se existe algum número diferente de 0 ou 1.
 Se encontrar, retornar erro indicando em qual linha.]

```

```

:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - F_DOM")

verifica_dummy(df, 'F_DOM')
log_output.info('\n\n=====\\n')

return df

def passo_fe_dom(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "FE_DOM"

    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - FE_DOM")
    log_output.info('\n\n=====\\n')

    return df

def passo_tipo_dom(passo, df):
    """
    * Substituir **0** por **None (NA)**
    * Substituir **3** por **2**
    * Substituir **2** por **0**

    # #####Categorias anteriores / novas
    # Valor / Descrição
    # ----/----
    # 1/Particular
    # 2/Coletivo
    # 3/Favela

    # #####Categorias novas
    # Valor / Descrição
    # ----/----
    # 0/Coletivo
    # 1/Particular

    [Teste: Checar se existe algum número < 0 ou > 1.
     Se encontrar, retornar erro indicando em qual linha.]"""

    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - TIPO_DOM")

    df.loc[df['TIPO_DOM']==0,'TIPO_DOM'] = None

```

```
df.loc[df['TIPO_DOM']==3,'TIPO_DOM'] = 2
df.loc[df['TIPO_DOM']==2,'TIPO_DOM'] = 0

# Verificando intervalo de valores - condições:
    # "TIPO_DOM >= 0" E "TIPO_DOM <= 1"
verifica_dummy(df, 'TIPO_DOM')
log_output.info('\'\n\'=====\\n')

return df
```

3.4 Funções da Família

Função/Variável	Status
passo_tot_fam	OK
passo_f_fam	OK
passo_fe_fam	OK
passo_cond_mora	OK
passo_qt_auto	OK
passo_qt_bici	OK
passo_qt_moto	OK
passo_ren_fam	OK
passo_cd_renfam	OK

```
In [ ]: log_tela.info('Definindo as funções da família')
log_output.info('\n\n=====\\n')

def passo_tot_fam(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "TOT_FAM"
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - TOT_FAM")
    log_output.info('\\n\\n=====\\n')

    return df

def passo_f_fam(passo, df):
    """
    Checar se existe algum erro na coluna "F_FAM"

    # #####Categorias
    # Valor/Descrição
    # ----/----
    # 0/Demais registros
    # 1/Primeiro Registro da Família

    [Teste: Checar se existe algum número diferente de 0 ou 1.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - F_FAM")

    verifica_dummy(df, 'F_FAM')
```

```

log_output.info('`n`n=====`n')

return df

def passo_fe_fam(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "FE_FAM"
    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - FE_FAM")
    log_output.info('`n`n=====`n')

    return df

def passo_cond_mora(passo, df):
    """
    Substituir valores da coluna "COND_MORA"

    * Substituir todos valores **5** por **0**
    * Substituir todos valores **0** por **None**
    * Substituir todos valores **4** por **3**

    ##### Categorias anteriores

    Valor/Descrição
    ----/----
    1/Alugada
    2/Própria
    3/Cedida
    4/Outros
    5/Não respondeu

    ##### Categorias novas

    Valor/Descrição
    ----/----
    1/Alugada
    2/Própria
    3/Outros

    [Teste: Checar se existe algum número < 1 ou > 3.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: passo
    :param df: dataframe
    :return: dataframe modificado
    """

    log_tela.info("### PASSO " + str(passo) + " - COND_MORA")

    # Substituindo valor 5 por 0
    df.loc[df['COND_MORA']==5,'COND_MORA'] = 0

```

```

# Substituindo valor 0 por None
df.loc[df['COND_MORA']==0,'COND_MORA'] = None
# Substituindo valor 4 por 3
df.loc[df['COND_MORA']==4,'COND_MORA'] = 3

# Verificando intervalo de valores - condições:
# "COND_MORA >= 1" E "COND_MORA <= 3"
verifica_range(df, 'COND_MORA', 1, 3)
log_output.info('\n\n=====\\n')

return df


def passo_qt_auto(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "QT_AUTO"
    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - QT_AUTO")
    log_output.info('\n\n=====\\n')

    return df


def passo_qt_bici(passo, df):
    """
    Não existe essa informação no banco de dados de 1997, logo,
    este campo será preenchido com 'None'.
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: dataframe modificado
    """
    df['QT_BICI'] = None

    log_tela.info('### PASSO ' + str(passo) + ' - QT_BICI')
    log_output.info('\n\n=====\\n')

    return df


def passo_qt_moto(passo, df):
    """
    Não existe essa informação no banco de dados de 1997, logo,
    este campo será preenchido com 'None'.
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: dataframe modificado
    """
    df['QT_MOTO'] = None

    log_tela.info('### PASSO ' + str(passo) + ' - QT_MOTO')
    log_output.info('\n\n=====\\n')

```

```

    return df

def passo_ren_fam(passo, df, deflator):
    """
    Corrigie a renda familiar de acordo com o deflator passado na função.
    :param passo: Número do passo atual para registro/log
    :param df: dataframe
    :param deflator: Deflator utilizado para correção da renda.
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - REN_FAM")
    df['REN_FAM'] = df['REN_FAM'] * deflator
    return df

def passo_cd_renfam(passo, df):
    """
    Substituir valores da coluna "CD_RENFAM"

    * Substituir todos valores **2** por **0**
    * Substituir todos valores **3** por **2**

    ##### Categorias anteriores
    Valor/Descrição
    ----/----
    1/Renda familiar completa
    2/Não tem renda
    3/Renda familiar incompleta

    ##### Categorias novas
    Valor/Descrição
    ----/----
    0/Renda Familiar Declarada como Zero
    1/Renda Familiar Declarada e Maior que Zero
    2/Renda Atribuída

    [Teste: Checar se existe algum número < 0 ou > 2.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe corrigido
    """
    log_tela.info("### PASSO " + str(passo) + " - CD_RENFAM")

    # Substituindo valor 2 por 0
    df.loc[df['CD_RENFAM']==2, 'CD_RENFAM'] = 0
    # Substituindo valor 3 por 2
    df.loc[df['CD_RENFAM']==3, 'CD_RENFAM'] = 2

    # Dividindo a categoria '0', "Respondeu", em:

```

```
#     - 0 "Renda Familiar Declarada como Zero" e
#     - 1 "Renda Familiar Declarada e Maior que Zero"
df.loc[(df['CD_RENFAM'] == 0) &
       (df['REN_FAM'] != 0) &
       (df['REN_FAM'].notnull()), 'CD_RENFAM'] = 1

# Verificando intervalo de valores - condições:
# "CD_RENFAM >= 0" E "CD_RENFAM <= 2"
verifica_range(df, 'CD_RENFAM', 0, 2)
log_output.info('\n\n=====\\n')

return df
```

3.5 Funções da pessoa

Função/Variável	Status
passo_f_pess	OK
passo_fe_pess	OK
passo_sit_fam	OK
passo_idade	OK
passo_sexo	OK
passo_grau_instr	OK
passo_ocup	OK
passo_setor_ativ	OK
passo_ren_ind	OK
passo_cd_renind	OK
passo_estuda	Verificar

Obs.: Se ESTUDA for ser alterado pela verificação do ZONA_ESC, no main() ele deverá ser chamado após ZONA_ESC. Atualmente ele já está localizado após o ZONA_ESC na chamada do main(). Após confirmar o que será feito com ele, se for necessário alterar sua localização.

```
In [ ]: log_tela.info('Definindo as funções da pessoa')
log_output.info('\n\n=====\\n')

def passo_f_pess(passo, df):
    """
    Checar se existe algum erro na coluna "F_PESS"

    # #####Categorias
    # Valor/Descrição
    # ----/----
    # 0/Demais registros
    # 1/Primeiro Registro da Pessoa

    [Teste: Checar se existe algum número diferente de 0 ou 1.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - F_PESS")

    verifica_dummy(df, 'F_PESS')
    log_output.info('\n\n=====\\n')

    return df

def passo_fe_pess(passo, df):
```

```

"""
Nada há que se fazer em relação aos dados das colunas "FE_PESS"
:param passo: Número do passo atual para registro/log
:param df:
:return:
"""
log_tela.info("### PASSO " + str(passo) + " - FE_PESS")
log_output.info('\n\n=====\\n')

return df

def passo_sit_fam(passo, df):
    """
    Não é preciso realizar nenhuma alteração

    ##### Categorias anteriores
    Valor/Descrição
    ----/----
    1/Chefe
    2/Cônjuge
    3/Filho(a)
    4/Parente / Agregado
    5/Empregado Residente
    6/Visitante não residente na RMSP

    ##### Categorias novas:
    Valor/Descrição
    ----/----
    1/ Pessoa Responsável
    2/ Cônjuge/Companheiro(a)
    3/ Filho(a)/Enteado(a)
    4/ Outro Parente / Agregado
    5/ Empregado Residente
    6/ Outros (visitante não residente / parente do empregado)

    [Teste: Checar se existe algum número < 1 ou > 6.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe modificado
    """
    log_tela.info("### PASSO " + str(passo) + " - SIT_FAM")

    # Verificando intervalo de valores - condições:
    # "SIT_FAM >= 1" E "SIT_FAM <= 6"
    verifica_range(df, 'SIT_FAM', 1, 6)
    log_output.info('\n\n=====\\n')

    return df

def passo_idade(passo, df):
    """

```

```

Nada há que se fazer em relação aos dados da coluna "IDADE"
:param passo: Número do passo atual para registro/log
:param df:
:return:
"""
log_tela.info("### PASSO " + str(passo) + " - IDADE")
log_output.info('\n\n=====\\n')

return df

def passo_sexo(passo, df):
    """
    # #####Categorias anteriores
    # Valor/Descrição
    # ----/----
    # 0/Não Respondeu (-> None)
    # 1/Masculino
    # 2/Feminino

    # #####Categorias novas
    # Valor/Descrição
    # ----/----
    # 0/Masculino
    # 1/Feminino

    [Teste: Checar se existe algum número diferente de 0 ou 1.
     Se encontrar, retornar erro indicando em qual linha.]
```

:param passo: Número do passo atual para registro/log

:param df:

:return: retorna o dataframe modificado

"""
log_tela.info("### PASSO " + str(passo) + " - SEXO")

Substituindo valor 0 por None
df.loc[df['SEXO']==0,'SEXO'] = None
Substituindo valor 1 por 0
df.loc[df['SEXO']==1,'SEXO'] = 0
Substituindo valor 2 por 1
df.loc[df['SEXO']==2,'SEXO'] = 1

Verificando intervalo de valores - condições:
"SEXO >= 0" E "SEXO <= 1"
verifica_dummy(df, 'SEXO')
log_output.info('\n\n=====\\n')

return df

```

def passo_grau_instr(passo, df):
    """
    Substituir valores da coluna "GRAU_INSTR"

    * Substituir todos valores **2** por **1**
```

```

* Substituir todos valores **3** por **1**
* Substituir todos valores **4** por **2**
* Substituir todos valores **5** por **2**
* Substituir todos valores **6** por **3**
* Substituir todos valores **7** por **3**
* Substituir todos valores **8** por **4**
* Substituir todos valores **0** por **None**

##### Categorias anteriores:
Valor/Descrição
----/----
1/Não-alfabetizado
2/Pré-escola
3/1º Grau incompleto
4/1º Grau completo
5/2º Grau incompleto
6/2º Grau completo
7/Superior Incompleto
8/Superior Completo

##### Categorias novas
Valor/Descrição
----/----
1/Não-Alfabetizado/Fundamental Incompleto
2/Fundamental Completo/Médio Incompleto
3/Médio Completo/Superior Incompleto
4/Superior completo

[Teste: Checar se existe algum número < 1 ou > 4.
Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna dataframe modificado
"""
log_tela.info("### PASSO " + str(passo) + " - GRAU_INSTR")

# Substituindo valor 2 por 1
df.loc[df['GRAU_INSTR']==2, 'GRAU_INSTR'] = 1
# Substituindo valor 3 por 1
df.loc[df['GRAU_INSTR']==3, 'GRAU_INSTR'] = 1
# Substituindo valor 4 por 2
df.loc[df['GRAU_INSTR']==4, 'GRAU_INSTR'] = 2
# Substituindo valor 5 por 2
df.loc[df['GRAU_INSTR']==5, 'GRAU_INSTR'] = 2
# Substituindo valor 6 por 3
df.loc[df['GRAU_INSTR']==6, 'GRAU_INSTR'] = 3
# Substituindo valor 7 por 3
df.loc[df['GRAU_INSTR']==7, 'GRAU_INSTR'] = 3
# Substituindo valor 8 por 4
df.loc[df['GRAU_INSTR']==8, 'GRAU_INSTR'] = 4
# Substituindo valor 0 por None
df.loc[df['GRAU_INSTR']==0, 'GRAU_INSTR'] = None

# Verificando intervalo de valores - condições:

```

```

# "GRAU_INSTR >= 1" E "GRAU_INSTR <= 4"
verifica_range(df, 'GRAU_INSTR', 1, 4)
log_output.info('\n\n=====\\n')

return df

def passo_ocup(passo, df):
    """
    Substituir valores da coluna "OCUP"

    Substituir todos valores **2** por **1**
    Substituir todos valores **3** por **2**
    Substituir todos valores **5** por **3**
    Substituir todos valores **6** por **5**
    Substituir todos valores **7** por **6**
    Substituir todos valores **8** por **7**
    Substituir todos valores **0** por **None**

    #####Categorias anteriores:
    Valor/Descrição
    ----/----
    1/Ocupado
    2/Ocupado eventualmente
    3/Em licença
    4/Não ocupada
    5/Aposentado/Pensionista
    6/Nunca trabalhou
    7/Dona de casa
    8/Estudante

    #####Categorias novas
    Valor/Descrição
    ----/----
    1/Tem trabalho
    2/Em licença médica
    3/Aposentado / pensionista
    4/Desempregado
    5/Sem ocupação
    6/Dona de casa
    7/Estudante

[Teste: Checar se existe algum número < 1 ou > 7.
Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna dataframe modificado
"""
log_tela.info("### PASSO " + str(passo) + " - OCUP")

# Substituindo valor 2 por 1
df.loc[df['OCUP']==2,'OCUP'] = 1
# Substituindo valor 3 por 2
df.loc[df['OCUP']==3,'OCUP'] = 2
# Substituindo valor 5 por 3

```

```

df.loc[df['OCUP']==5,'OCUP'] = 3
# Substituindo valor 6 por 5
df.loc[df['OCUP']==6,'OCUP'] = 5
# Substituindo valor 7 por 6
df.loc[df['OCUP']==7,'OCUP'] = 6
# Substituindo valor 8 por 7
df.loc[df['OCUP']==8,'OCUP'] = 7
# Substituindo valor 0 por None
df.loc[df['OCUP']==0,'OCUP'] = None

# Verificando intervalo de valores - condições:
# "OCUP >= 1" E "OCUP <= 7"
verifica_range(df, 'OCUP', 1, 7)
log_output.info('\n\n=====\\n')

return df

def passo_setor_ativ(passo, df):
    """
    Substituir valores da coluna "SETOR_ATIV"

    Na coluna "SETOR_ATIV", linha i,
        ler o valor da linha i da coluna "SETOR_ATIV", daí,
        buscar o mesmo valor na coluna "COD" do arquivo setor_ativ-1997.csv.
    Ao achar, retornar o valor da mesma linha, só que da coluna "COD_UNIF"

    #####Categorias anteriores
    > ver arquivo .csv

    #####Categorias novas
    Valor/Descrição
    ----/----
    1/Agrícola
    2/Construção Civil
    3/Indústria
    4/Comércio
    5/Administração Pública
    6/Serviços de Transporte
    7/Outros serviços
    8/Doutros
    9/Não se aplica

    [Teste: Checar se existe algum número < 1 ou > 9.
     Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna dataframe modificado
"""
log_tela.info("### PASSO " + str(passo) + " - SETOR_ATIV")

log_output.info('Lendo arquivo de referência externa setor_ativ-1997.csv')
df_setor = pd.read_csv('auxiliares/setor_ativ-1997.csv', sep=';', decimal=',')

```

```

def setor_aux(row):
    df.loc[df['SETOR_ATIV']==row['COD'], 'SETOR_ATIV'] = row['COD_UNIF']

    # Esta variável out não é utilizada para nada além de evitar um
    # monte de output que não será utilizado e que é gerado pelo método apply.
    out = df_setor.apply(setor_aux, axis=1)

    # Substituindo valor 0 por None
    df.loc[df['SETOR_ATIV']==0, 'SETOR_ATIV'] = None

    # Verificando intervalo de valores - condições:
    # "SETOR_ATIV >= 1" E "SETOR_ATIV <= 9"
    verifica_range(df, 'SETOR_ATIV', 1, 9)
    log_output.info('\n\n=====\\n')

    return df


def passo_ren_ind(passo, df, deflator):
    """
    Corriga a renda individual de acordo com o deflator passado na função.
    :param passo: Número do passo atual para registro/log
    :param df: dataframe
    :param deflator: Deflator utilizado para correção da renda.
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - REN_IND")
    df['REN_IND'] = df['REN_IND'] * deflator

    return df


def passo_cd_renid(passo, df):
    """
    Substituir valores da coluna "CD_RENIND"

    * Substituir todos valores **3** por None
    * Substituir todos valores **2** por **0**

    ##### Categorias anteriores
    Valor/Descrição
    ----/----
    1/Tem renda
    2/Não tem renda
    3/Não respondeu

    ##### Categorias novas
    Valor/Descrição
    ----/-----
    0    /Não tem renda
    1    /Tem renda
    """

    [Teste: Checar se existe algum número < 0 ou > 1.

```

```

Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - CD_RENIND")

df.loc[df['CD_RENIND']==3,'CD_RENIND'] = None
df.loc[df['CD_RENIND']==2,'CD_RENIND'] = 0

# Verificando intervalo de valores - condições:
# "CD_RENIND >= 0" E "CD_RENIND <= 1"
verifica_dummy(df, 'CD_RENIND')
log_output.info('\n\n=====\\n')

return df

def passo_estuda(passo, df):
"""
Se zona da escola for zero (0) então não estuda (0), senão, estuda (1)

#####
ATENÇÃO
ESTA FUNÇÃO SÓ DEVE SER #
EXECUTADA APÓS A GERAÇÃO#
DE TODOS OS ID'S, POIS #
HÁ UMA DESCONFIANÇA #
QUANTO À QUALIDADE #
DESTE DADO #
#####

* Substituir todos valores **1** por **0**
* Substituir todos valores **2** por **1**
* Substituir todos valores **3** por **1**
* Substituir todos valores **4** por **1**

#### Categorias anteriores
Valor/Descrição
----/----
1/Não
2/Creche/Pré-Escola
3/1º Grau/2º Grau/3º Graus
4/Outros

#### Categorias novas
Valor/Descrição
----/----
0/Não estuda
1/Estuda

[Teste: Checar se existe algum número diferente de 0 ou 1.
Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log

```

```

:param df:
:return: retorna dataframe com campo ESCOLA resolvido
"""

log_tela.info("### PASSO " + str(passo) + " - ESTUDA")

# Substitui 1 por 0
df.loc[df['ESTUDA'] == 1, 'ESTUDA'] = 0
# Substitui 2 por 1
df.loc[df['ESTUDA'] == 2, 'ESTUDA'] = 1
# Substitui 3 por 1
df.loc[df['ESTUDA'] == 3, 'ESTUDA'] = 1
# Substitui 4 por 1
df.loc[df['ESTUDA'] == 4, 'ESTUDA'] = 1

# Substituindo todos que declararam zona escola diferente de zero
# com campo ESTUDA igual a 1.
df.loc[(df['ZONA_ESC'] != 0)&
       (df['ZONA_ESC'].notnull()), 'ESTUDA'] = 1

verifica_dummy(df, 'ESTUDA')
log_output.info('\'\n\'=====\\n')

return df

```

3.6 Funções referentes da Viagem

Função/Variável	Status
passo_f_vitag	OK
passo_fe_vitag	OK
passo_zona_esc	OK
passo_subzona_esc	OK
passo_mun_esc	OK
passo_zona_trab1	OK
passo_subzona_trab1	OK
passo_mun_trab1	OK
passo_zona_trab2	OK
passo_subzona_trab2	OK
passo_mun_trab2	OK
passo_zona_orig	OK
passo_subzona_orig	OK
passo_mun_orig	OK
passo_zona_dest	OK
passo_subzona_dest	OK
passo_mun_dest	OK
passo_serv_pas_orig	OK
passo_serv_pas_dest	OK
passo_motivo_orig	OK
passo_motivo_dest	OK
passo_modo1	OK
passo_modo2	OK
passo_modo3	OK
passo_modo4	OK
passo_modo_prin	OK
passo_tipo_est_auto	OK
passo_valor_est_auto	OK
passo_tot_vitag	OK

Obs: O passo_tot_vitag só deve ser executado após a produção dos ID's e NO's.

```
In [ ]: log_tela.info('Definindo funções referentes às viagens')
log_output.info('\n=====\\n=====\\n')
```

```
def passo_f_vitag(passo, df):
```

```

"""
Checar se existe algum erro na coluna "F_VIAG"

# #####Categorias
# Valor/Descrição
# ----/----
# 0/Não fez viagem
# 1/Fez viagem

[Teste: Checar se existe algum número diferente de 0 ou 1.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe com F_VIAG modificado
"""

log_tela.info("### PASSO " + str(passo) + " - F_VIAG")

# Setando F_VIAG=1 para todos os casos
df['F_VIAG'] = 1

# Setando F_VIAG=0 se DURACAO==0
df.loc[df['DURACAO']==0, 'F_VIAG'] = 0

verifica_dummy(df, 'F_VIAG')
log_output.info('\n\n=====\\n')

return df


def passo_fe_viag(passo, df):
"""
# Nada há que se fazer em relação aos dados da coluna "FE_VIAG"
:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""

log_tela.info("### PASSO " + str(passo) + " - FE_VIAG")
log_output.info('\n\n=====\\n')

return df


def passo_zona_esc(passo, df):
"""
Checar se existe algum erro

# #####Categorias:
# > 1 a 389

[Teste: Checar se existe algum número < 1 ou > 389.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: Sem retorno
"""

```

```

"""
log_tela.info("### PASSO " + str(passo) + " - ZONA_ESC")

# Substituindo valor 0 por None
df.loc[df['ZONA_ESC']==0,'ZONA_ESC'] = None

# Verificando intervalo de valores - condições:
# "ZONA_ESC >= 1" E "ZONA_ESC <= 389"
verifica_range(df, 'ZONA_ESC', 1, 389)
log_output.info('\n\n=====\\n')

return df
"""

def passo_subzona_esc(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 9

    [Teste: Checar se existe algum número < 1 ou > 9.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - SUBZONA_ESC")

    # Substituindo valor 0 por None
    df.loc[df['SUBZONA_ESC']==0,'SUBZONA_ESC'] = None

    # Verificando intervalo de valores - condições:
    # "SUBZONA_ESC >= 1" E "SUBZONA_ESC <= 9"
    verifica_range(df, 'SUBZONA_ESC', 1, 9)
    log_output.info('\n\n=====\\n')

    return df

def passo_mun_esc(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias
    # > 1 a 39

    [Teste: Checar se existe algum número < 1 ou > 39.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - MUN_ESC")

```

```

# Substituindo valor 0 por None
df.loc[df['MUN_ESC']==0,'MUN_ESC'] = None

# Verificando intervalo de valores - condições:
# "MUN_ESC >= 1" E "MUN_ESC <= 39"
verifica_range(df, 'MUN_ESC', 1, 39)
log_output.info('\n\n=====\\n')

return df


def passo_zona_trab1(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 389

    [Teste: Checar se existe algum número < 1 ou > 389.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - ZONA_TRAB1")

    # Substituindo valor 0 por None
    df.loc[df['ZONA_TRAB1']==0,'ZONA_TRAB1'] = None

    # Verificando intervalo de valores - condições:
    # "ZONA_TRAB1 >= 1" E "ZONA_TRAB1 <= 389"
    verifica_range(df, 'ZONA_TRAB1', 1, 389)
    log_output.info('\\n\\n=====\\n')

    return df


def passo_subzona_trab1(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 9

    [Teste: Checar se existe algum número < 1 ou > 9.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - SUBZONA_TRAB1")

    # Substituindo valor 0 por None

```

```

df.loc[df['SUBZONA_TRAB1']==0, 'SUBZONA_TRAB1'] = None

# Verificando intervalo de valores - condições:
# "SUBZONA_TRAB1 >= 1" E "SUBZONA_TRAB1 <= 9"
verifica_range(df, 'SUBZONA_TRAB1', 1, 9)
log_output.info('\n\n=====\\n')

return df

def passo_mun_trab1(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias
    # > 1 a 39

    [Teste: Checar se existe algum número < 1 ou > 39.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - MUN_TRAB1")

    # Substituindo valor 0 por None
    df.loc[df['MUN_TRAB1']==0, 'MUN_TRAB1'] = None

    # Verificando intervalo de valores - condições:
    # "MUN_TRAB1 >= 1" E "MUN_TRAB1 <= 39"
    verifica_range(df, 'MUN_TRAB1', 1, 39)
    log_output.info('\n\n=====\\n')

    return df

def passo_zona_trab2(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 389

    [Teste: Checar se existe algum número < 1 ou > 389.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - ZONA_TRAB2")

    # Substituindo valor 0 por None
    df.loc[df['ZONA_TRAB2']==0, 'ZONA_TRAB2'] = None

```

```

# Verificando intervalo de valores - condições:
# "ZONA_TRAB2 >= 1" E "ZONA_TRAB2 <= 389"
verifica_range(df, 'ZONA_TRAB2', 1, 389)
log_output.info('\'\n\'=====\'\n')

return df

def passo_subzona_trab2(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 9

    [Teste: Checar se existe algum número < 1 ou > 9.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - SUBZONA_TRAB2")

    # Substituindo valor 0 por None
    df.loc[df['SUBZONA_TRAB2']==0, 'SUBZONA_TRAB2'] = None

    # Verificando intervalo de valores - condições:
    # "SUBZONA_TRAB2 >= 1" E "SUBZONA_TRAB2 <= 9"
    verifica_range(df, 'SUBZONA_TRAB2', 1, 9)
    log_output.info('\'\n\'=====\'\n')

    return df

def passo_mun_trab2(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias
    # > 1 a 39

    [Teste: Checar se existe algum número < 1 ou > 39.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - MUN_TRAB2")

    # Substituindo valor 0 por None
    df.loc[df['MUN_TRAB2']==0, 'MUN_TRAB2'] = None

    # Verificando intervalo de valores - condições:
    # "MUN_TRAB2 >= 1" E "MUN_TRAB2 <= 39"

```

```

verifica_range(df, 'MUN_TRAB2', 1, 39)
log_output.info('\n\n=====\\n')

return df

def passo_zona_orig(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 389

    [Teste: Checar se existe algum número < 1 ou > 389.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - ZONA_ORIG")

    # Substituindo valor 0 por None
    df.loc[df['ZONA_ORIG']==0, 'ZONA_ORIG'] = None

    # Verificando intervalo de valores - condições:
    # "ZONA_ORIG >= 1" E "ZONA_ORIG <= 389"
    verifica_range(df, 'ZONA_ORIG', 1, 389)
    log_output.info('\n\n=====\\n')

    return df

def passo_subzona_orig(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 9

    [Teste: Checar se existe algum número < 1 ou > 9.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - SUBZONA_ORIG")

    # Substituindo valor 0 por None
    df.loc[df['SUBZONA_ORIG']==0, 'SUBZONA_ORIG'] = None

    # Verificando intervalo de valores - condições:
    # "SUBZONA_ORIG >= 1" E "SUBZONA_ORIG <= 9"
    verifica_range(df, 'SUBZONA_ORIG', 1, 9)
    log_output.info('\n\n=====\\n')

```

```

    return df

def passo_mun_orig(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias
    # > 1 a 39

    [Teste: Checar se existe algum número < 1 ou > 39.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - MUN_ORIG")

    # Substituindo valor 0 por None
    df.loc[df['MUN_ORIG']==0,'MUN_ORIG'] = None

    # Verificando intervalo de valores - condições:
    # "MUN_ORIG >= 1" E "MUN_ORIG <= 39"
    verifica_range(df, 'MUN_ORIG', 1, 39)
    log_output.info('\n\n=====\\n')

    return df

def passo_zona_dest(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 389

    [Teste: Checar se existe algum número < 1 ou > 389.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - ZONA_DEST")

    # Substituindo valor 0 por None
    df.loc[df['ZONA_DEST']==0,'ZONA_DEST'] = None

    # Verificando intervalo de valores - condições:
    # "ZONA_DEST >= 1" E "ZONA_DEST <= 389"
    verifica_range(df, 'ZONA_DEST', 1, 389)
    log_output.info('\n\n=====\\n')

    return df

```

```

def passo_subzona_dest(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 9

    [Teste: Checar se existe algum número < 1 ou > 9.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - SUBZONA_DEST")

    # Substituindo valor 0 por None
    df.loc[df['SUBZONA_DEST']==0,'SUBZONA_DEST'] = None

    # Verificando intervalo de valores - condições:
    # "SUBZONA_DEST >= 1" E "SUBZONA_DEST <= 9"
    verifica_range(df, 'SUBZONA_DEST', 1, 9)
    log_output.info('\n\n=====\\n')

    return df


def passo_mun_dest(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias
    # > 1 a 39

    [Teste: Checar se existe algum número < 1 ou > 39.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - MUN_DEST")

    # Substituindo valor 0 por None
    df.loc[df['MUN_DEST']==0,'MUN_DEST'] = None

    # Verificando intervalo de valores - condições:
    # "MUN_DEST >= 1" E "MUN_DEST <= 39"
    verifica_range(df, 'MUN_DEST', 1, 39)
    log_output.info('\n\n=====\\n')

    return df


def passo_serv_pas_orig(passo, df):

```

```

"""
Substituir **0** por None
Substituir **2** por **0**

####Categorias antigas
Valor/Descrição
----/----
0/Não Respondido
1/Sim
2/Não

####Categorias novas
Valor/Descrição
----/----
0/Não
1/Sim

:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - SERV_PAS_ORIG")

df.loc[df['SERV_PAS_ORIG']==0,'SERV_PAS_ORIG'] = None
df.loc[df['SERV_PAS_ORIG']==2,'SERV_PAS_ORIG'] = 0

verifica_dummy(df, 'SERV_PAS_ORIG')
log_output.info('\n\n=====\\n')

return df


def passo_serv_pas_dest(passo, df):
"""
Substituir **0** por None
Substituir **2** por **0**

####Categorias antigas
Valor/Descrição
----/----
0/Não Respondido
1/Sim
2/Não

####Categorias novas
Valor/Descrição
----/----
0/Não
1/Sim

:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""

```

```

log_tela.info("### PASSO " + str(passo) + " - SERV_PAS_DEST")

df.loc[df['SERV_PAS_DEST']==0, 'SERV_PAS_DEST'] = None
df.loc[df['SERV_PAS_DEST']==2, 'SERV_PAS_DEST'] = 0

verifica_dummy(df, 'SERV_PAS_DEST')
log_output.info('\n\n=====\\n')

return df

def passo_motivo_orig(passo, df):
    """
    * Substituir todos valores **0** por **None**

    ##### Categorias anteriores
    Valor/Descrição
    ----/----
    1/Trabalho/Indústria
    2/Trabalho/Comércio
    3/Trabalho/Serviços
    4/Escola/Educação
    5/Compras
    6/Médico/Dentista/Saúde
    7/Recreação/Visitas
    8/Residência
    9/Outros

    ##### Categorias novas
    Valor/Descrição
    ----/----
    1/Trabalho/Indústria
    2/Trabalho/Comércio
    3/Trabalho/Serviços
    4/Educação
    5/Compras
    6/Saúde
    7/Lazer
    8/Residência
    9/Outros

    [Teste: Checar se existe algum número < 1 ou > 9.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: dataframe modificado
    """
    log_tela.info("### PASSO " + str(passo) + " - MOTIVO_ORIG")

    # Substituindo valor 0 por None
    df.loc[df['MOTIVO_ORIG']==0, 'MOTIVO_ORIG'] = None

    # Verificando intervalo de valores - condições:
    # "MOTIVO_ORIG >= 1" E "MOTIVO_ORIG <= 9"

```

```

verifica_range(df, 'MOTIVO_ORIG', 1, 9)
log_output.info('\n\n=====\\n')

return df

def passo_motivo_dest(passo, df):
    """
    * Substituir todos valores **0** por **None**

    ##### Categorias anteriores
    Valor/Descrição
    ----/----
    1/Trabalho/Indústria
    2/Trabalho/Comércio
    3/Trabalho/Serviços
    4/Escola/Educação
    5/Compras
    6/Médico/Dentista/Saúde
    7/Recreação/Visitas
    8/Residência
    9/Outros

    ##### Categorias novas
    Valor/Descrição
    ----/----
    1/Trabalho/Indústria
    2/Trabalho/Comércio
    3/Trabalho/Serviços
    4/Educação
    5/Compras
    6/Saúde
    7/Lazer
    8/Residência
    9/Outros

    [Teste: Checar se existe algum número < 1 ou > 9.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: dataframe modificado
    """
    log_tela.info("### PASSO " + str(passo) + " - MOTIVO_DEST")

    # Substituindo valor 0 por None
    df.loc[df['MOTIVO_DEST']==0, 'MOTIVO_DEST'] = None

    # Verificando intervalo de valores - condições:
    # "MOTIVO_DEST >= 1" E "MOTIVO_DEST <= 9"
    verifica_range(df, 'MOTIVO_DEST', 1, 9)
    log_output.info('\n\n=====\\n')

return df

```

```

def passo_modo1(passo, df):
    """
    Substituir valores da coluna "MODO1"

    * Substituir todos valores **3** por **2**
    * Substituir todos valores **4** por **3**
    * Substituir todos valores **5** por **4**
    * Substituir todos valores **6** por **5**
    * Substituir todos valores **7** por **6**
    * Substituir todos valores **8** por **7**
    * Substituir todos valores **9** por **8**
    * Substituir todos valores **10** por **9**
    * Substituir todos valores **11** por **10**
    * Substituir todos valores **12** por **11**
    * Substituir todos valores **13** por **12**
    * Substituir todos valores **0** por **None**

    ##### Categorias anteriores
    Valor/Descrição
    ----/----
    1/Ônibus
    2/Ônibus Fretado
    3/Transporte Escolar
    4/Dirigindo Automóvel
    5/Passageiro de Automóvel
    6/Táxi
    7/Lotação/Perua
    8/Metrô
    9/Trem
    10/Moto
    11/Bicicleta
    12/A Pé
    13/Outros

    ##### Categorias novas
    Valor/Descrição
    ----/----
    1/Ônibus
    2/Ônibus Escolar / Empresa
    3/Dirigindo Automóvel
    4/Passageiro de Automóvel
    5/Táxi
    6/Lotação / Perua / Van / Microônibus
    7/Metrô
    8/Trem
    9/Moto
    10/Bicicleta
    11/A Pé
    12/Outros

    [Teste: Checar se existe algum número < 1 ou > 12.
     Se encontrar, retornar erro indicando em qual linha.]
```

:param passo: Número do passo atual para registro/log

```

:param df:
:return: sem retorno
"""

log_tela.info("### PASSO " + str(passo) + " - MODO1")

df.loc[df['MODO1']==3,'MODO1'] = 2
df.loc[df['MODO1']==4,'MODO1'] = 3
df.loc[df['MODO1']==5,'MODO1'] = 4
df.loc[df['MODO1']==6,'MODO1'] = 5
df.loc[df['MODO1']==7,'MODO1'] = 6
df.loc[df['MODO1']==8,'MODO1'] = 7
df.loc[df['MODO1']==9,'MODO1'] = 8
df.loc[df['MODO1']==10,'MODO1'] = 9
df.loc[df['MODO1']==11,'MODO1'] = 10
df.loc[df['MODO1']==12,'MODO1'] = 11
df.loc[df['MODO1']==13,'MODO1'] = 12
df.loc[df['MODO1']==0,'MODO1'] = None

# Verificando intervalo de valores - condições:
# "MODO1 >= 1" E "MODO1 <= 12"
verifica_range(df, 'MODO1', 1, 12)
log_output.info('\n\n=====\\n')

return df

def passo_modo2(passo, df):
"""
Substituir valores da coluna "MODO2"

* Substituir todos valores **3** por **2**
* Substituir todos valores **4** por **3**
* Substituir todos valores **5** por **4**
* Substituir todos valores **6** por **5**
* Substituir todos valores **7** por **6**
* Substituir todos valores **8** por **7**
* Substituir todos valores **9** por **8**
* Substituir todos valores **10** por **9**
* Substituir todos valores **11** por **10**
* Substituir todos valores **12** por **11**
* Substituir todos valores **13** por **12**
* Substituir todos valores **0** por **None**

#### Categorias anteriores
Valor/Descrição
----/----
1/Ônibus
2/Ônibus Fretado
3/Transporte Escolar
4/Dirigindo Automóvel
5/Passageiro de Automóvel
6/Táxi
7/Lotação/Perua
8/Metrô

```

```

9/Trem
10/Moto
11/Bicicleta
12/A Pé
13/Outros

##### Categorias novas
Valor/Descrição
----/----
1/Ônibus
2/Ônibus Escolar / Empresa
3/Dirigindo Automóvel
4/Passageiro de Automóvel
5/Táxi
6/Lotação / Perua / Van / Microônibus
7/Metrô
8/Trem
9/Moto
10/Bicicleta
11/A Pé
12/Outros

[Teste: Checar se existe algum número < 1 ou > 12.
Se encontrar, retornar erro indicando em qual linha.]
```

:param passo: Número do passo atual para registro/log

:param df:

:return: sem retorno

"""

```

log_tela.info("### PASSO " + str(passo) + " - MODO3")

df.loc[df['MODO2']==3,'MODO2'] = 2
df.loc[df['MODO2']==4,'MODO2'] = 3
df.loc[df['MODO2']==5,'MODO2'] = 4
df.loc[df['MODO2']==6,'MODO2'] = 5
df.loc[df['MODO2']==7,'MODO2'] = 6
df.loc[df['MODO2']==8,'MODO2'] = 7
df.loc[df['MODO2']==9,'MODO2'] = 8
df.loc[df['MODO2']==10,'MODO2'] = 9
df.loc[df['MODO2']==11,'MODO2'] = 10
df.loc[df['MODO2']==12,'MODO2'] = 11
df.loc[df['MODO2']==13,'MODO2'] = 12
df.loc[df['MODO2']==0,'MODO2'] = None

# Verificando intervalo de valores - condições:
# "MODO2 >= 1" E "MODO2 <= 12"
verifica_range(df, 'MODO2', 1, 12)
log_output.info('`\\n\\n=====\`\\n')

return df

def passo_modo3(passo, df):
"""
Substituir valores da coluna "MODO3"

```

```

* Substituir todos valores **3** por **2**
* Substituir todos valores **4** por **3**
* Substituir todos valores **5** por **4**
* Substituir todos valores **6** por **5**
* Substituir todos valores **7** por **6**
* Substituir todos valores **8** por **7**
* Substituir todos valores **9** por **8**
* Substituir todos valores **10** por **9**
* Substituir todos valores **11** por **10**
* Substituir todos valores **12** por **11**
* Substituir todos valores **13** por **12**
* Substituir todos valores **0** por **None**

```

Categorias anteriores

Valor/Descrição

----/----

- 1/*Ônibus*
- 2/*Ônibus Fretado*
- 3/*Transporte Escolar*
- 4/*Dirigindo Automóvel*
- 5/*Passageiro de Automóvel*
- 6/*Táxi*
- 7/*Lotação/Perua*
- 8/*Metrô*
- 9/*Trem*
- 10/*Moto*
- 11/*Bicicleta*
- 12/*A Pé*
- 13/*Outros*

Categorias novas

Valor/Descrição

----/----

- 1/*Ônibus*
- 2/*Ônibus Escolar / Empresa*
- 3/*Dirigindo Automóvel*
- 4/*Passageiro de Automóvel*
- 5/*Táxi*
- 6/*Lotação / Perua / Van / Microônibus*
- 7/*Metrô*
- 8/*Trem*
- 9/*Moto*
- 10/*Bicicleta*
- 11/*A Pé*
- 12/*Outros*

[Teste: Checar se existe algum número < 1 ou > 12.

Se encontrar, retornar erro indicando em qual linha.]

:param passo: Número do passo atual para registro/log

:param df:

:return: sem retorno

"""

log_tela.info("### PASSO " + str(passo) + " - MOD03")

```

df.loc[df['MODO3']==3, 'MODO3'] = 2
df.loc[df['MODO3']==4, 'MODO3'] = 3
df.loc[df['MODO3']==5, 'MODO3'] = 4
df.loc[df['MODO3']==6, 'MODO3'] = 5
df.loc[df['MODO3']==7, 'MODO3'] = 6
df.loc[df['MODO3']==8, 'MODO3'] = 7
df.loc[df['MODO3']==9, 'MODO3'] = 8
df.loc[df['MODO3']==10, 'MODO3'] = 9
df.loc[df['MODO3']==11, 'MODO3'] = 10
df.loc[df['MODO3']==12, 'MODO3'] = 11
df.loc[df['MODO3']==13, 'MODO3'] = 12
df.loc[df['MODO3']==0, 'MODO3'] = None

# Verificando intervalo de valores - condições:
# "MODO3 >= 1" E "MODO3 <= 12"
verifica_range(df, 'MODO3', 1, 12)
log_output.info('\'\n\'=====\\n')

return df

def passo_modo4(passo, df):
    """
    Substituir valores da coluna "MODO4"

    * Substituir todos valores **3** por **2**
    * Substituir todos valores **4** por **3**
    * Substituir todos valores **5** por **4**
    * Substituir todos valores **6** por **5**
    * Substituir todos valores **7** por **6**
    * Substituir todos valores **8** por **7**
    * Substituir todos valores **9** por **8**
    * Substituir todos valores **10** por **9**
    * Substituir todos valores **11** por **10**
    * Substituir todos valores **12** por **11**
    * Substituir todos valores **13** por **12**
    * Substituir todos valores **0** por **None**

    ##### Categorias anteriores
    Valor/Descrição
    ----/----
    1/Ônibus
    2/Ônibus Fretado
    3/Transporte Escolar
    4/Dirigindo Automóvel
    5/Passageiro de Automóvel
    6/Táxi
    7/Lotação/Perua
    8/Metrô
    9/Trem
    10/Moto
    11/Bicicleta
    12/A Pé

```

13/Outros

```
#### Categorias novas  
Valor/Descrição  
----/----  
1/Ônibus  
2/Ônibus Escolar / Empresa  
3/Dirigindo Automóvel  
4/Passageiro de Automóvel  
5/Táxi  
6/Lotação / Peruá / Van / Microônibus  
7/Metrô  
8/Trem  
9/Moto  
10/Bicicleta  
11/A Pé  
12/Outros
```

[Teste: Checar se existe algum número < 1 ou > 12.

Se encontrar, retornar erro indicando em qual linha.]

:param passo: Número do passo atual para registro/log

:param df:

:return: sem retorno

"""

```
log_tela.info("### PASSO " + str(passo) + " - MODO4")
```

```
df.loc[df['MODO4']==3,'MODO4'] = 2  
df.loc[df['MODO4']==4,'MODO4'] = 3  
df.loc[df['MODO4']==5,'MODO4'] = 4  
df.loc[df['MODO4']==6,'MODO4'] = 5  
df.loc[df['MODO4']==7,'MODO4'] = 6  
df.loc[df['MODO4']==8,'MODO4'] = 7  
df.loc[df['MODO4']==9,'MODO4'] = 8  
df.loc[df['MODO4']==10,'MODO4'] = 9  
df.loc[df['MODO4']==11,'MODO4'] = 10  
df.loc[df['MODO4']==12,'MODO4'] = 11  
df.loc[df['MODO4']==13,'MODO4'] = 12  
df.loc[df['MODO4']==0,'MODO4'] = None
```

Verificando intervalo de valores - condições:

"MODO4 >= 1" E "MODO4 <= 12"

```
verifica_range(df, 'MODO4', 1, 12)
```

```
log_output.info('\n\n=====\\n')
```

```
return df
```

```
def passo_modo_prin(passo, df):  
    """
```

Substituir valores da coluna "MODO_PRIN"

```
* Substituir todos valores **3** por **2**  
* Substituir todos valores **4** por **3**  
* Substituir todos valores **5** por **4**
```

```

* Substituir todos valores **6** por **5**
* Substituir todos valores **7** por **6**
* Substituir todos valores **8** por **7**
* Substituir todos valores **9** por **8**
* Substituir todos valores **10** por **9**
* Substituir todos valores **11** por **10**
* Substituir todos valores **12** por **11**
* Substituir todos valores **13** por **12**
* Substituir todos valores **0** por **None**

```

Categorias anteriores

Valor/Descrição

----/----

1/*Ônibus*

2/*Ônibus Fretado*

3/*Transporte Escolar*

4/*Dirigindo Automóvel*

5/*Passageiro de Automóvel*

6/*Táxi*

7/*Lotação/Perua*

8/*Metrô*

9/*Trem*

10/*Moto*

11/*Bicicleta*

12/*A Pé*

13/*Outros*

Categorias novas

Valor/Descrição

----/----

1/*Ônibus*

2/*Ônibus Escolar / Empresa*

3/*Dirigindo Automóvel*

4/*Passageiro de Automóvel*

5/*Táxi*

6/*Lotação / Perua / Van / Microônibus*

7/*Metrô*

8/*Trem*

9/*Moto*

10/*Bicicleta*

11/*A Pé*

12/*Outros*

[Teste: Checar se existe algum número < 1 ou > 12.

Se encontrar, retornar erro indicando em qual linha.]

:param passo: Número do passo atual para registro/log

:param df:

:return: sem retorno

"""

log_tela.info("### PASSO " + str(passo) + " - MODO_PRIN")

df.loc[df['MODO_PRIN']==3,'MODO_PRIN'] = 2

df.loc[df['MODO_PRIN']==4,'MODO_PRIN'] = 3

df.loc[df['MODO_PRIN']==5,'MODO_PRIN'] = 4

```

df.loc[df['MODO_PRIN']==6, 'MODO_PRIN'] = 5
df.loc[df['MODO_PRIN']==7, 'MODO_PRIN'] = 6
df.loc[df['MODO_PRIN']==8, 'MODO_PRIN'] = 7
df.loc[df['MODO_PRIN']==9, 'MODO_PRIN'] = 8
df.loc[df['MODO_PRIN']==10, 'MODO_PRIN'] = 9
df.loc[df['MODO_PRIN']==11, 'MODO_PRIN'] = 10
df.loc[df['MODO_PRIN']==12, 'MODO_PRIN'] = 11
df.loc[df['MODO_PRIN']==13, 'MODO_PRIN'] = 12
df.loc[df['MODO_PRIN']==0, 'MODO_PRIN'] = None

# Verificando intervalo de valores - condições:
    # "MODO_PRIN >= 1" E "MODO_PRIN <= 12"
verifica_range(df, 'MODO_PRIN', 1, 12)
log_output.info('\n\n=====\\n')

return df

def passo_tipo_vitag(passo, df):
    """
* Substituir os valores **0** por **None**

# #####Categorias novas
# Valor/Descrição
# ----/----
# 1/Coletivo
# 2/Individual
# 3/A pé

[Teste: Checar se existe algum número < 1 ou > 3.
    Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno

:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - TIPO_VIAG")

# Substituindo valor 0 por None
df.loc[df['TIPO_VIAG']==0, 'TIPO_VIAG'] = None

# Verificando intervalo de valores - condições:
    # "MODO_PRIN >= 1" E "MODO_PRIN <= 3"
verifica_range(df, 'TIPO_VIAG', 1, 3)

log_output.info('\n\n=====\\n')
return df

def passo_tipo_est_auto(passo, df):
    """
Nada a fazer, essa informação não existe em 1997

```

```

# Valor/Descrição
# ----/----
# 0/Não Respondeu
# 1/Não Estacionou
# 2/Estacionamento Particular (Avulso / Mensal)
# 3/Estacionamento Próprio
# 4/Estacionamento Patrocinado
# 5/Rua (meio fio / zona azul / zona marrom / parquímetro)

[Teste: Checar se existe algum número < 0 ou > 5.
 Se encontrar, retornar erro indicando em qual linha.]
```

:param passo: Número do passo atual para registro/log

:param df:

:return: retorna dataframe modificado

"""

```

log_tela.info("### PASSO " + str(passo) + " - TIPO_EST_AUTO")

df['TIPO_EST_AUTO'] = None

log_output.info('\n\n=====\\n')

return df
```

```

def passo_valor_est_auto(passo, df, deflator):
"""
Nada há que se fazer em relação à coluna "VALOR_EST_AUTO".
Não há dados de 1997, coluna permanecerá vazia
:param passo: Número do passo atual para registro/log
:param df:
:return: sem retorno
"""
log_tela.info("### PASSO " + str(passo) + " - VALOR_EST_AUTO")

df['VALOR_EST_AUTO'] = None

log_output.info('\n\n=====\\n')

return df
```

```

def passo_tot_viang(passo, df):
"""
#####
#      ATENÇÃO      #
# ESTA FUNÇÃO SÓ DEVE SER #
# EXECUTADA APÓS A GERAÇÃO#
# DE TODOS OS ID'S, POIS #
# HÁ UMA DESCONFIANÇA   #
# QUANTO À QUALIDADE    #
#      DESTE DADO       #
#####

```


3.7 Funções que geram os “NO”s e os “ID”s

Função/Variável	Status
gera_nos_e_ids	Verificar NO VIAG

```
In [ ]: log_tela.info('Definindo funções que geram os "NO"s e os "ID"s')
log_output.info('\n\n=====\\n')
```

```
def gera_nos_e_ids(passo, df):
    """
    Esta função gera todos os IDs e todos os NOs das variáveis:
    DOM (domicílio), FAM (família), PESS (pessoa) e VIAG (viagem)
    A ordem de geração é:
    NO_DOM, ID_DOM, NO_FAM, ID_FAM, NO_PESS, ID_PESS, NO_VIAG, ID_VIAG
    Esta ordem é fixa pois cada uma dessas variáveis depende da geração
    da variável anterior.

    Os NOs são gerados como se fossem subíndices.
    No caso dos domicílios, eles são contabilizados dentro de cada zona.
    Assim, cada novo domicílio que aparece na listagem recebe um número
    que representa sua posição na sequência de domicílios dentro da zona
    na qual ele está contido.
    As famílias seguem o mesmo raciocínio, com relação ao domicílio, as
    pessoas com relação às famílias e as viagens com relação às pessoas.

    Deve-se apenas tomar cuidado com as viagens, pois existem pessoas que
    não realizaram viagens. Neste caso, estes registros devem contabilizar
    o NO_VIAG como zero, e não como 1. Para contemplar estes casos, vamos
    utilizar a variável F_VIAG, que representa quando há ou não viagem
    naquele registro (F_VIAG==1 tem viagem, F_VIAG==0 não tem viagem).
    Assim, o que precisamos fazer para calcular o NO_VIAG é agrupar os
    registros por pessoa (ID_PESS) e, dentro de cada agrupamento, somar
    o valor de F_VIAG registro a registro (linha por linha)
    cumulativamente. No final do processo, atribui-se zero a NO_VIAG
    quando F_VIAG for igual a zero.
    """

def gera_id_dom(row):
    """
    Gera o ID_DOM baseado no 'ANO', na 'ZONA_DOM' e no 'NO_DOM'
    O argumento passado é a "linha".
    Uso:
        df['ID_DOM'] = df.apply(gera_ID_DOM, axis=1)
    Retorna: ID_DOM da respectiva linha
    """
    # Formatando o ano para string
    ano = str(row['ANO'])

    # Formatando a zona para ficar como string com 3 caracteres
    zona = str('%03d' % row['ZONA_DOM'])
```

```

# Formatando no_dom para ficar como string com 4 caracteres
no_dom = str('%04d' % row['NO_DOM'])

# Retornando o número inteiro correspondente à string
# concatenada de ano + zona + no_dom
return ano + zona + no_dom


def gera_id_fam(row):
    """
    Gera o ID_FAM baseado no 'ID_DOM' e no 'NO_FAM'
    O argumento passado é a "linha".
    Uso:
        df['ID_FAM'] = df.apply(gera_ID_FAM, axis=1)
    Retorna: ID_FAM da respectiva linha
    """
    # Formatando id_dom como string
    id_dom = str(row['ID_DOM'])

    # Formatando no_fam como string para ficar com 2 caracteres
    no_fam = str('%02d' % row['NO_FAM'])

    # Retornando o número inteiro correspondente à string
    # concatenada de ID_DOM com NO_FAM
    return id_dom + no_fam


def gera_id_pess(row):
    """
    Gera o ID_PESS baseado no 'ID_FAM' e no 'NO_PESS'
    O argumento passado é a "linha".
    Uso:
        df['ID_PESS'] = df.apply(gera_ID_PESS, axis=1)
    Retorna: ID_PESS da respectiva linha
    """
    # Formatando id_fam como string
    id_fam = str(row['ID_FAM'])

    # Formatando no_pess como string para ficar com 2 caracteres
    no_pess = str('%02d' % row['NO_PESS'])

    # Retornando o número inteiro correspondente à string
    # concatenada de ID_FAM com NO_PESS
    return id_fam + no_pess


def gera_id_viag(row):
    """
    Gera o ID_VIAG baseado no 'ID_PESS' e no 'NO_VIAG'
    O argumento passado é a "linha".
    Uso:
        df['ID_VIAG'] = df.apply(gera_ID_VIAG, axis=1)
    Retorna ID_VIAG da respectiva linha
    """
    # Formatando id_pess como string
    id_pess = str(row['ID_PESS'])

```

```

# Formatando no_viag como string para ficar com 2 caracteres
no_viag = str('%02d' % row['NO_VIAG'])

# Retornando o número inteiro correspondente à string
# concatenada de ID_PESS com NO_VIAG
return id_pess + no_viag

#gera NO_DOM
log_tela.info("Gerando NO_DOM")
df['NO_DOM'] = df.groupby('ZONA_DOM', sort=False)['F_DOM'].cumsum()
log_tela.info("NO_DOM gerado")
log_tela.info("Gerando ID_DOM")
#gera ID_DOM
df['ID_DOM'] = df.apply(gera_id_dom, axis=1)
log_tela.info("ID_DOM gerado")

#gera NO_FAM
log_tela.info("Gerando NO_FAM")
df['NO_FAM'] = df.groupby('ID_DOM', sort=False)['F_FAM'].cumsum()
log_tela.info("NO_FAM gerado")
log_tela.info("Gerando ID_FAM")
#gera ID_FAM
df['ID_FAM'] = df.apply(gera_id_fam, axis=1)
log_tela.info("ID_FAM gerado")

#gera NO_PESS
log_tela.info("Gerando NO_PESS")
df['NO_PESS'] = df.groupby('ID_FAM', sort=False)['F_PESS'].cumsum()
log_tela.info("NO_PESS gerado")
log_tela.info("Gerando ID_PESS")
#gera ID_PESS
df['ID_PESS'] = df.apply(gera_id_pess, axis=1)
log_tela.info("ID_PESS gerado")

#gera NO_VIAG
log_tela.info("Gerando NO_VIAG")
#df['NO_VIAG'] = 1
df['NO_VIAG'] = df.groupby('ID_PESS', sort=False)['F_VIAG'].cumsum()
# Verificando se FE_VIAG == 0 e zerando NO_VIAG nesse caso
df.loc[df['FE_VIAG'] == 0, 'NO_VIAG'] = 0
log_tela.info("NO_VIAG gerado")
log_tela.info("Gerando ID_VIAG")
#gera ID_VIAG
df['ID_VIAG'] = df.apply(gera_id_viag, axis=1)
log_tela.info("ID_VIAG gerado")

return df

```

3.8 Função relativa à entrevista

Função/Variável	Status
passo_cd_entre	OK

Obs: o passo_cd_entre deve ser executado após o passo_tot_vitag.

```
In [ ]: log_tela.info('Definindo função relativa à entrevista')
log_output.info('\n\n=====\\n')

def passo_cd_entre(passo, df):
    """
    #####
    #      ATENÇÃO      #
    # ESTA FUNÇÃO SÓ DEVE SER #
    # EXECUTADA APÓS A GERAÇÃO#
    #      DO TOT_VIAG     #
    #####
    ----
    Substituir valores da coluna "CD_ENTRE"
    Todas viagens são consideradas "completas", segundo informações do Metrô

    * sem viagem: se TOT_VIAG == 0
    * com viagem: se TOT_VIAG != 0

    # #####Categorias novas
    # | Valor | Descrição |
    # | ----- | ----- |
    # | 0 | Completa sem viagem |
    # | 1 | Completa com viagem |

    [Teste: Checar se existe algum número diferente de 0 ou 1.
     Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return:
"""

log_tela.info("### PASSO " + str(passo) + " - CD_ENTRE")
log_tela.warning('\n\n#####\n' + 
                 'Este passo DEVE ser executado apenas após a \\n' +
                 'execução do passo que geram o TOT_VIAG.\\n' +
                 '#####\n')

# Definindo 'CD_ENTRE' baseado no valor de 'TOT_VIAG'
df.loc[df['TOT_VIAG'] == 0, 'CD_ENTRE'] = 0
df.loc[df['TOT_VIAG'] != 0, 'CD_ENTRE'] = 1

verifica_dummy(df, 'CD_ENTRE')
log_output.info('\n\n=====\\n')

return df
```

```
In [ ]: def passo_correcoes(passo, df):
    log_tela.info("### PASSO " + str(passo) + " - Correções")
    log_tela.info("Corrigindo 'SERV_PAS' caso TOT_VIAG seja nulo.")
    df.loc[df['TOT_VIAG'] == 0, 'SERV_PAS_ORIG'] = 0
    df.loc[df['TOT_VIAG'] == 0, 'SERV_PAS_DEST'] = 0

    return df
```

4 Definindo a função principal (main)

Esta função vai ler os arquivos (csv) necessários e, então, irá efetuar cada passo, chamando as respectivas funções que foram definidas acima. Ao final, esta função irá salvar o resultado das transformações realizadas num arquivo csv.

```
In [ ]: def main():
    start_time = time.time()

    log_tela.info("Horário de início da execução: %s" %
                  time.strftime("%H:%M", time.localtime(start_time)))

    # ----
    log_tela.info('Lendo arquivos CSV')

    log_output.info('Lendo CSV da base original da OD.')
    od = pd.read_csv('bases/OD_1997.csv', sep=';', decimal=',')

    deflator = 1.94136464
    log_output.info('Deflator a ser utilizado para correção monetária: ' +
                    str(deflator))

    # Filtrando dataframe para pegar apenas uma amostra
    #od = od[:1000].copy()

    log_output.info('\n\n=====\\n')
    log_tela.info('Pré-Processamento.')
    od = pre_processamento(od)

    log_tela.info('Imprimindo a lista de variáveis do dataframe da OD.')
    log_tela.debug(od.columns.tolist())

    #Contador de 'PASSO'
    passo = 1

    # ----
    # ##Passo: "ANO"
    od = passo_ano(passo, od)
    passo += 1

    # ----
    # ##Passo: "DIA_SEM"
    od = passo_dia_sem(passo, od)
    passo += 1

    # ----
    # ##Passo: "UCODs"
    od = passo_ucods(passo, od)
    passo += 1

    # ----
    # ##Passo: "ZONA_DOM"
    od = passo_zona_dom(passo, od)
    passo += 1
```

```

# -----
# ##Passo: "SUBZONA_DOM"
od = passo_subzona_dom(passo, od)
passo += 1

# -----
# ##Passo: "MUN_DOM"
od = passo_mun_dom(passo, od)
passo += 1

# -----
# ##Passo: "F_DOM"
od = passo_f_dom(passo, od)
passo += 1

# -----
# ##Passo: "FE_DOM"
od = passo_fe_dom(passo, od)
passo += 1

# -----
# ##Passo: "TIPO_DOM"
od = passo_tipo_dom(passo, od)
passo += 1

# -----
# ##Passo: "TOT_FAM"
od = passo_tot_fam(passo, od)
passo += 1

# -----
# ##Passo: "F_FAM"
od = passo_f_fam(passo, od)
passo += 1

# -----
# ##Passo: "FE_FAM"
od = passo_fe_fam(passo, od)
passo += 1

# -----
# ##Passo: "COND_MORA"
od = passo_cond_mora(passo, od)
passo += 1

# -----
# ##Passo: "QT_AUTO"
od = passo_qt_auto(passo, od)
passo += 1

# -----
# ##Passo: "QT_BICI"
od = passo_qt_bici(passo, od)
passo += 1

```

```

# -----
# ##Passo: "QT_MOTO"
od = passo_qt_moto(passo, od)
passo += 1

# -----
# ##PASSO: "REN_FAM"
od = passo_ren_fam(passo, od, deflator)
passo += 1

# -----
# ##Passo: "CD_RENFAM"
od = passo_cd_renfam(passo, od)
passo += 1

# -----
# ##Passo: "F_PESS"
od = passo_f_pess(passo, od)
passo += 1

# -----
# ##Passo: "FE_PESS"
od = passo_fe_pess(passo, od)
passo += 1

# -----
# ##Passo: "SIT_FAM"
od = passo_sit_fam(passo, od)
passo += 1

# -----
# ##Passo: "IDADE"
od = passo_idade(passo, od)
passo += 1

# -----
# ##Passo: "SEXO"
od = passo_sexo(passo, od)
passo += 1

# -----
# ##Passo: "GRAU_INSTR"
od = passo_grau_instr(passo, od)
passo += 1

# -----
# ##Passo: "OCUP"
od = passo_ocup(passo, od)
passo += 1

# -----
# ##Passo: "SETOR_ATIV"
od = passo_setor_ativ(passo, od)

```

```

    passo += 1

    # ##Passo: "REN_IND"
    od = passo_ren_ind(passo, od, deflator)
    passo += 1

    # ##Passo: "CD_RENIND"
    od = passo_cd_renind(passo, od)
    passo += 1

    # -----
    # ##Passo: "F_VIAG"
    od = passo_f_viag(passo, od)
    passo += 1

    # -----
    # ##"FE_VIAG"
    od = passo_fe_viag(passo, od)
    passo += 1

    # -----
    # ##Passo: "ZONA_ESC"
    od = passo_zona_esc(passo, od)
    passo += 1

    # -----
    # ##Passo: "SUBZONA_ESC"
    od = passo_subzona_esc(passo, od)
    passo += 1

    # -----
    # ##Passo: "MUN_ESC"
    od = passo_mun_esc(passo, od)
    passo += 1

    # -----
    # ##Passo: "ESTUDA"
    # Este passo deve vir após os passos de localização da escola
    od = passo_estuda(passo, od)
    passo += 1

    # -----
    # ##Passo: "ZONA_TRAB1"
    od = passo_zona_trab1(passo, od)
    passo += 1

    # -----
    # ##Passo: "SUBZONA_TRAB1"
    od = passo_subzona_trab1(passo, od)
    passo += 1

    # -----
    # ##Passo: "MUN_TRAB1"
    od = passo_mun_trab1(passo, od)

```

```

    passo += 1

    # -----
    # ##Passo: "ZONA_TRAB2"
    od = passo_zona_trab2(passo, od)
    passo += 1

    # -----
    # ##Passo: "SUBZONA_TRAB2"
    od = passo_subzona_trab2(passo, od)
    passo += 1

    # -----
    # ##Passo: "MUN_TRAB2"
    od = passo_mun_trab2(passo, od)
    passo += 1

    # -----
    # ##Passo: "ZONA_ORIG"
    od = passo_zona_orig(passo, od)
    passo += 1

    # -----
    # ##Passo: "SUBZONA_ORIG"
    od = passo_subzona_orig(passo, od)
    passo += 1

    # -----
    # ##Passo: "MUN_ORIG"
    od = passo_mun_orig(passo, od)
    passo += 1

    # -----
    # ##Passo: "ZONA_DEST"
    od = passo_zona_dest(passo, od)
    passo += 1

    # -----
    # ##Passo: "SUBZONA_DEST"
    od = passo_subzona_dest(passo, od)
    passo += 1

    # -----
    # ##Passo: "MUN_DEST"
    od = passo_mun_dest(passo, od)
    passo += 1

    # -----
    # ##Passo: "SERV_PAS_ORIG"
    od = passo_serv_pas_orig(passo, od)
    passo += 1

    # -----
    # ##Passo: "SERV_PAS_DEST"

```

```

od = passo_serv_pas_dest(passo, od)
passo += 1

# -----
# ##Passo: "MOTIVO_ORIG"
od = passo_motivo_orig(passo, od)
passo += 1

# -----
# ##Passo: "MOTIVO_DEST"
od = passo_motivo_dest(passo, od)
passo += 1

# -----
# ##Passo: "MOD01"
od = passo_modo1(passo, od)
passo += 1

# -----
# ##Passo: "MOD02"
od = passo_modo2(passo, od)
passo += 1

# -----
# ##Passo: "MOD03"
od = passo_modo3(passo, od)
passo += 1

# -----
# ##Passo: "MOD04"
od = passo_modo4(passo, od)
passo += 1

# -----
# ##Passo: "MODO_PRIN"
od = passo_modo_prin(passo, od)
passo += 1

# -----
# ##Passo: "TIPO_VIAG"
od = passo_tipo_vitag(passo, od)
passo += 1

# -----
# ##"H_SAIDA"; "MIN_SAIDA"; "ANDA_ORIG"; "H_CHEG"; "MIN_CHEG";
# "ANDA_DEST"; "DIST_VIAG" e "DURACAO"
# Nada há que se fazer em relação aos dados das colunas acima mencionadas

# -----
# ##Passo: "TIPO_EST_AUTO"
od = passo_tipo_est_auto(passo, od)
passo += 1

# -----

```

```

# ##Passo: "VALOR_EST_AUTO"
od = passo_valor_est_auto(passo, od, deflator)
passo += 1

# -----
# ##Passo: Coordenadas
od = coordenadas(passo, od)
passo += 1

## O passo TOT_VIAG apenas é chamado após a geração dos IDs.

# -----
# ##Passo: Gerando NO's e ID's
od = gera_nos_e_ids(passo, od)
passo += 1

# -----
# ##Passo: "TOT_VIAG"
od = passo_tot_viag(passo, od)
passo += 1

# -----
# ##Passo: "CD_ENTRE"
od = passo_cd_entre(passo, od)
passo += 1

# -----
# ##Passo: Correções
od = passo_correcoes(passo, od)
passo += 1

log_tela.info('Salvando dataframe como arquivo CSV')
# -----
# ## Salvando o dataframe num arquivo local
od.to_csv('outputs/1997_od.csv', sep=';', decimal=',', index=False)

log_tela.info("Base gerada. Arquivo: outputs/1997_od.csv")

log_tela.info("Tempo total de execução: %s segundos" %
             (time.time() - start_time))

log_tela.info("Horário de finalização: %s" %
             (time.strftime("%H:%M", time.localtime(time.time()))))

log_tela.info("Terminou o main")

```

5 RUN the main() function....

```
In [ ]: if __name__ == "__main__":
    main()
```

rotina_final_2007

January 4, 2016

```
In [ ]: # Setup Inicial  
In [ ]: # coding: utf-8  
        import math  
        import logging  
        import time  
        import pandas as pd  
        import csv  
  
        # Faz os gráficos um pouco mais bonitos  
        pd.set_option('display.mpl_style', 'default')
```

1 Definindo Loggers

Define os loggers

Estes 'loggers' serão utilizados para salvar as saídas (outputs) em um arquivo de texto no diretório 'outputs'.

ref: <http://stackoverflow.com/questions/17035077/python-logging-to-multiple-log-files-from-different-classes>

ATENÇÃO: RODAR O BLOCO ABAIXO APENAS UMA VEZ, SE ESTE BLOCO FOR EXECUTADO MAIS DE UMA VEZ OS LOGS SERÃO DUPLICADOS.

```
In [ ]: log_formatter = logging.Formatter('%(asctime)s | %(levelname)s: %(message)s')  
  
log_output = logging.getLogger('log_output')  
FH_output = logging.FileHandler(  
    'outputs/2007_output.log', mode='w')  
FH_output.setFormatter(log_formatter)  
log_output.setLevel(logging.INFO)  
log_output.addHandler(FH_output)  
log_output.propagate = False  
  
# Este logger (log_tela) loga na tela e também  
# no arquivo de output, junto com o conteúdo do  
# logger log_output  
log_tela = logging.getLogger('log_tela')  
SH_tela = logging.StreamHandler()  
SH_tela.setFormatter(log_formatter)  
log_tela.addHandler(SH_tela)  
log_tela.addHandler(FH_output)  
log_tela.setLevel(logging.INFO)  
log_tela.propagate = False
```

2 Funções de 2007

2.1 Funções gerais assessórias

Função	Status
consulta_refext	OK
verifica_dummy	OK
verifica_range	OK
pre_processamento	OK
coordenadas	OK

```
In [ ]: log_tela.info('Definindo as funções gerais assessórias')
log_output.info('\n\n=====\\n')

def verifica_dummy(df, variavel):
    """
    Verifica se uma variável, dummy, contém algum valor diferente de 0 ou de 1.
    :param df: dataframe com os dados a serem verificados
    :param variavel: string com o nome da variável (coluna) que tem os dados
        que devem ser verificados.
    :return: Sem retorno, apenas salva as avaliações nos logs.
    Uso:
        verifica_dummy(dataframe, 'coluna a ser verificada')
    """
    contador_de_erros = 0
    log_tela.info('Verificando a variável Dummy: ' + variavel)

    df_erros = df[(df[variavel] != 0) & (df[variavel] != 1)]
    if len(df_erros[variavel].value_counts()) > 0:
        log_tela.warning(variavel + ": " +
                          str(len(df_erros[variavel].value_counts())) +
                          " erros encontrados:\\n" +
                          str(df_erros[variavel].value_counts()))
    else:
        log_tela.info(variavel + ": Nenhum erro encontrado.")

def verifica_range(df, variavel, valor_menor, valor_maior):
    """
    Verifica se uma variável, do tipo número inteiro, contém algum valor menor
    que "valor_menor" ou maior que "valor_maior".
    :param df: dataframe com os dados a serem verificados
    :param variavel: string com o nome da variável (coluna) que tem os dados
        que devem ser verificados
    :param valor_menor: Valor mínimo esperado na variável (int ou float)
    :param valor_maior: Valor máximo esperado na variável (int ou float)
    :return: Sem retorno, apenas salva as avaliações nos logs.
    Uso:
        verifica_range(dataframe, 'nome_variavel', valor_menor, valor_maior)
```

```

"""
log_tela.info('Verificando Range da variável: ' + variavel)
# Obs: Registros inválidos: None (equivalente a NA)
nulos = df[variavel].isnull().sum()
nulos = nulos if nulos else 0
log_output.info('\n\n' +
    ' - ' + 'Mínimo esperado: ' + str(valor_menor) + '\n' +
    ' - ' + 'Máximo esperado: ' + str(valor_maior) + '\n' +
    ' - ' + 'Total de registros: ' + str(len(df[variavel])) +
    '\n' +
    ' - ' + 'Registros nulos (NA): ' +
    str(df[variavel].isnull().sum()) + '\n'
)
df_erro = df[(df[variavel] < valor_menor) | (df[variavel] > valor_maior)]

if len(df_erro[variavel].value_counts()) > 0:

    result = df_erro[variavel].value_counts().sort_index()
    # Verificando limite inferior
    if result.first_valid_index() < valor_menor:
        log_tela.warning(
            variavel + ': ' + 'Valor inteiro mínimo encontrado: ' +
            str(result.first_valid_index()) + " - abaixo do esperado!")
    # Verificando limite superior
    if result.last_valid_index() > valor_maior:
        log_tela.warning(
            variavel + ': ' + 'Valor inteiro máximo encontrado: ' +
            str(result.last_valid_index()) + " - acima do esperado!")

    log_tela.warning(variavel + ': ' +
        str(len(df_erro[variavel].value_counts())) +
        ' valor(es) incorreto(s)' +
        'encontrado(s) nesta variável:\n' +
        str(df_erro[variavel].value_counts()))
else:
    log_tela.info(variavel + ": Nenhum erro encontrado.")

def pre_processamento(df):
"""
Realiza algumas ações prévias ao processamento dos dados,
removendo alguns registros e ajustando o dataframe.
"""

log_tela.info('Criando/Renomeando colunas no dataframe principal')

log_output.info('Renomeando coluna UCOD para UCOD_DOM')
df.rename(columns={'UCOD': 'UCOD_DOM'}, inplace=True)

log_output.info('Renomeando coluna ANDA_CHEG para ANDA_DEST')
df.rename(columns={'ANDA_CHEG': 'ANDA_DEST'}, inplace=True)

log_tela.info('Verificando se todas as variáveis esperadas' +

```

```

        'existem no dataframe.\n' +
        'Caso não exista alguma, ela é criada.')
variaveis = ['ANO', 'CD_ENTRE', 'DIA_SEM', 'UCOD_DOM', 'ZONA_DOM',
             'SUBZONA_DOM', 'MUN_DOM', 'CO_DOM_X', 'CO_DOM_Y', 'ID_DOM',
             'F_DOM', 'FE_DOM', 'NO_DOM', 'TIPO_DOM', 'TOT_FAM', 'ID_FAM',
             'F_FAM', 'FE_FAM', 'NO_FAM', 'COND_MORA', 'QT_AUTO', 'QT_BICI',
             'QT_MOTO', 'CD_RENFAM', 'REN_FAM', 'ID_PESS', 'F_PESS',
             'FE_PESS', 'NO_PESS', 'SIT_FAM', 'IDADE', 'SEXO', 'ESTUDA',
             'GRAU_INSTR', 'OCUP', 'SETOR_ATIV', 'CD_RENIND', 'REN_IND',
             'UCOD_ESC', 'ZONA_ESC', 'SUBZONA_ESC', 'MUN_ESC', 'CO_ESC_X',
             'CO_ESC_Y', 'UCOD_TRAB1', 'ZONA_TRAB1', 'SUBZONA_TRAB1',
             'MUN_TRAB1', 'CO_TRAB1_X', 'CO_TRAB1_Y', 'UCOD_TRAB2',
             'ZONA_TRAB2', 'SUBZONA_TRAB2', 'MUN_TRAB2', 'CO_TRAB2_X',
             'CO_TRAB2_Y', 'ID_VIAG', 'F_VIAG', 'FE_VIAG', 'NO_VIAG',
             'TOT_VIAG', 'UCOD_ORIG', 'ZONA_ORIG', 'SUBZONA_ORIG',
             'MUN_ORIG', 'CO_ORIG_X', 'CO_ORIG_Y', 'UCOD_DEST', 'ZONA_DEST',
             'SUBZONA_DEST', 'MUN_DEST', 'CO_DEST_X', 'CO_DEST_Y',
             'DIST_VIAG', 'SERV_PAS_ORIG', 'SERV_PAS_DEST', 'MOTIVO_ORIG',
             'MOTIVO_DEST', 'MODO1', 'MODO2', 'MODO3', 'MODO4', 'MODO_PRIN',
             'TIPO_VIAG', 'H_SAIDA', 'MIN_SAIDA', 'ANDA_ORIG', 'H_CHEG',
             'MIN_CHEG', 'ANDA_DEST', 'DURACAO', 'TIPO_EST_AUTO',
             'VALOR_EST_AUTO']

for variavel in variaveis:
    if variavel not in df.columns:
        # Se a variável não existe no dataframe então ela é criada
        # com valor padrão de NONE (NA)
        df[variavel] = None
    log_tela.info('Criando a variavel ' + variavel + ' no dataframe')

log_output.info('Reordenando as variáveis')
df = df[['ANO', 'CD_ENTRE', 'DIA_SEM', 'UCOD_DOM', 'ZONA_DOM',
          'SUBZONA_DOM', 'MUN_DOM', 'CO_DOM_X', 'CO_DOM_Y', 'ID_DOM',
          'F_DOM', 'FE_DOM', 'NO_DOM', 'TIPO_DOM', 'TOT_FAM', 'ID_FAM',
          'F_FAM', 'FE_FAM', 'NO_FAM', 'COND_MORA', 'QT_AUTO', 'QT_BICI',
          'QT_MOTO', 'CD_RENFAM', 'REN_FAM', 'ID_PESS', 'F_PESS',
          'FE_PESS', 'NO_PESS', 'SIT_FAM', 'IDADE', 'SEXO', 'ESTUDA',
          'GRAU_INSTR', 'OCUP', 'SETOR_ATIV', 'CD_RENIND', 'REN_IND',
          'UCOD_ESC', 'ZONA_ESC', 'SUBZONA_ESC', 'MUN_ESC', 'CO_ESC_X',
          'CO_ESC_Y', 'UCOD_TRAB1', 'ZONA_TRAB1', 'SUBZONA_TRAB1',
          'MUN_TRAB1', 'CO_TRAB1_X', 'CO_TRAB1_Y', 'UCOD_TRAB2',
          'ZONA_TRAB2', 'SUBZONA_TRAB2', 'MUN_TRAB2', 'CO_TRAB2_X',
          'CO_TRAB2_Y', 'ID_VIAG', 'F_VIAG', 'FE_VIAG', 'NO_VIAG',
          'TOT_VIAG', 'UCOD_ORIG', 'ZONA_ORIG', 'SUBZONA_ORIG',
          'MUN_ORIG', 'CO_ORIG_X', 'CO_ORIG_Y', 'UCOD_DEST', 'ZONA_DEST',
          'SUBZONA_DEST', 'MUN_DEST', 'CO_DEST_X', 'CO_DEST_Y',
          'DIST_VIAG', 'SERV_PAS_ORIG', 'SERV_PAS_DEST', 'MOTIVO_ORIG',
          'MOTIVO_DEST', 'MODO1', 'MODO2', 'MODO3', 'MODO4', 'MODO_PRIN',
          'TIPO_VIAG', 'H_SAIDA', 'MIN_SAIDA', 'ANDA_ORIG', 'H_CHEG',
          'MIN_CHEG', 'ANDA_DEST', 'DURACAO', 'TIPO_EST_AUTO',
          'VALOR_EST_AUTO']]

log_output.info('\n\n=====\\n')
log_output.info('\\n\\n=====\\n')

```

```
    return df

def coordenadas(passo, df):
    """
    Nada a fazer, as coordenadas já constam na base consolidada.

    :param passo: número do passo para o log
    :param df: Dafaframe a ser modificado (ex.: od2007)
    :return: retorna o dataframe
    """
    log_tela.info("### PASSO " + str(passo) + " - COORDENADAS")
    log_output.info('\n\n=====\\n')

    return df
```

2.2 Funções Gerais

Função/Variável	Status
passo_ano	OK
passo_dia_sem	OK
passo_ucods	OK

```
In [ ]: log_tela.info('Definindo as funções gerais')
log_output.info('\n'=====\\n')

def passo_ano(passo, df):
    """
    Preenche a coluna "ANO" com valor 4 em todas células
    Categorias:
    /valor/ano_correspondente/
    /-----/-----
    /1/1977/
    /2/1987/
    /3/1997/
    /4/2007/

    :param passo: Número do passo atual para registro/log
    :param df: dataframe a ser modificado
    :return: retorna o dataframe modificado
    """
    log_tela.info("### PASSO " + str(passo) + " - ANO")

    # Definindo valor '4' para todas as células da coluna ANO
    df["ANO"] = 4

    return df

def passo_dia_sem(passo, df):
    """
    Apenas verificar os valores existentes
    # #####Categorias:
    # Valor/Descrição
    # -----/-----
    # 2/Segunda-Feira
    # 3/Terça-Feira
    # 4/Quarta-Feira
    # 5/Quinta-Feira
    # 6/Sexta-Feira

    :param passo: Número do passo atual para registro/log
    :param df: dataframe a ser modificado
    :return: retorna o dataframe modificado
    """

```

```

log_tela.info("### PASSO " + str(passo) + " - DIA_SEM")

# Substituindo **0** por **None**
df.loc[df['DIA_SEM']==0,'DIA_SEM'] = None

# Verificando intervalo de valores - condições:
# "DIA_SEM >= 2" E "DIA_SEM <= 6"
verifica_range(df, 'DIA_SEM', 2, 6)
log_output.info('\n\n=====\\n')

return df

def passo_ucods(passo, df):
    """
    Itera sobre os registros do dataframe ucods,
    a cada iteração, utiliza o valor da coluna ZONA_REF
    para filtrar o dataframe df em cada um dos tipos de ZONA
    (DOM, ESC, TRAB1, TRAB2, ORIG, DEST) e substitui o valor
    da respectiva UCOD.

    :param passo: número do passo para o log
    :param df: Dataframe a ser modificado (ex.: od2007)
    :return: retorna o dataframe modificado com todas as UCODS aplicadas
    """
    log_tela.info("### PASSO " + str(passo) + " - UCODS")

    def ucod_aux(row):
        df.loc[df['ZONA_DOM']==row['ZONA_REF'], 'UCOD_DOM'] = row['UCOD_BUSCADA']
        df.loc[df['ZONA_ESC']==row['ZONA_REF'], 'UCOD_ESC'] = row['UCOD_BUSCADA']
        df.loc[df['ZONA_TRAB1']==row['ZONA_REF'], 'UCOD_TRAB1'] = row['UCOD_BUSCADA']
        df.loc[df['ZONA_TRAB2']==row['ZONA_REF'], 'UCOD_TRAB2'] = row['UCOD_BUSCADA']
        df.loc[df['ZONA_ORIG']==row['ZONA_REF'], 'UCOD_ORIG'] = row['UCOD_BUSCADA']
        df.loc[df['ZONA_DEST']==row['ZONA_REF'], 'UCOD_DEST'] = row['UCOD_BUSCADA']

    log_output.info('Lendo arquivo auxiliar UCOD')
    ucods = pd.read_csv('auxiliares/UCOD-2007.csv', sep=';', decimal=',')

    # Esta variável out não é utilizada para nada além de evitar um
    # monte de output que não será utilizado e que é gerado pelo método apply.
    out = ucods.apply(ucod_aux, axis=1)

    # Verificando intervalo de valores - condições:
    # "UCOD_XXX >= 1" E "UCOD_XXX <= 67"
    for tipo in ['DOM', 'ESC', 'TRAB1', 'TRAB2', 'ORIG', 'DEST']:
        verifica_range(df, 'UCOD_' + tipo, 1, 67)
    log_output.info('\n\n=====\\n')

    return df

```

2.3 Funções do Domicílio

Função/Variável	Status
passo_zona_dom	OK
passo_subzona_dom	OK
passo_mun_dom	OK
passo_f_dom	OK
passo_fe_dom	OK
passo_tipo_dom	OK

```
In [ ]: log_tela.info('Definindo as funções do domicílio')
log_output.info('\n\n=====\\n')

def passo_zona_dom(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 460

    [Teste: Checar se existe algum número < 1 ou > 460.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: passo
    :param df: dataframe
    :return: retorna o dataframe sem modificações
    """
    log_tela.info("### PASSO " + str(passo) + " - ZONA_DOM")

    # Substituindo valor 0 por None
    df.loc[df['ZONA_DOM']==0,'ZONA_DOM'] = None

    # Verificando intervalo de valores - condições:
    # "ZONA_DOM >= 1" E "ZONA_DOM <= 460"
    verifica_range(df, 'ZONA_DOM', 1, 460)
    log_output.info('\n\n=====\\n')

    return df

def passo_subzona_dom(passo, df):
    """
    A variável SUBZONA_DOM ficará vazia (NA) em 2007 pois este ano já contém
    os dados de coordenadas.

    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe sem modificações
    """
    log_tela.info("### PASSO " + str(passo) + " - SUBZONA_DOM")
```

```

df['SUBZONA_DOM'] = None

log_output.info('\n\n=====\\n')

return df

def passo_mun_dom(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias
    # > 1 a 39

    [Teste: Checar se existe algum número < 1 ou > 39.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe sem modificações
    """
    log_tela.info("### PASSO " + str(passo) + " - MUN_DOM")

    # Substituindo valor 0 por None
    df.loc[df['MUN_DOM']==0,'MUN_DOM'] = None

    # Verificando intervalo de valores - condições:
    # "MUN_DOM >= 1" E "MUN_DOM <= 39"
    verifica_range(df, 'MUN_DOM', 1, 39)
    log_output.info('\n\n=====\\n')

    return df

def passo_f_dom(passo, df):
    """
    Checar se existe algum erro na coluna "F_DOM"

    # #####Categorias
    # Valor/Descrição
    # ----/----
    # 0/Demais registros
    # 1/Primeiro Registro do Domicílio

    [Teste: Checar se existe algum número diferente de 0 ou 1.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe sem modificações
    """
    log_tela.info("### PASSO " + str(passo) + " - F_DOM")

    verifica_dummy(df, 'F_DOM')
    log_output.info('\n\n=====\\n')

```

```

    return df

def passo_fe_dom(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "FE_DOM"

    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe sem modificações
    """
    log_tela.info("### PASSO " + str(passo) + " - FE_DOM")
    log_output.info('\n\n=====\\n')

    return df

def passo_tipo_dom(passo, df):
    """
    * Substituir **0** por **None (NA)**
    * Substituir **3** por **2**
    * Substituir **2** por **0**

    # #####Categorias anteriores / novas
    # Valor / Descrição
    # ----/----
    # 1/Particular
    # 2/Coletivo
    # 3/Favela

    # #####Categorias novas
    # Valor / Descrição
    # ----/----
    # 0/Coletivo
    # 1/Particular

    [Teste: Checar se existe algum número < 0 ou > 1.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - TIPO_DOM")

    df.loc[df['TIPO_DOM']==0,'TIPO_DOM'] = None
    df.loc[df['TIPO_DOM']==3,'TIPO_DOM'] = 2
    df.loc[df['TIPO_DOM']==2,'TIPO_DOM'] = 0

    # Verificando intervalo de valores - condições:
    # "TIPO_DOM >= 0" E "TIPO_DOM <= 1"
    verifica_dummy(df, 'TIPO_DOM')
    log_output.info('\n\n=====\\n')

```

```
return df
```

2.4 Funções da Família

Função/Variável	Status
passo_tot_fam	OK
passo_f_fam	OK
passo_fe_fam	OK
passo_cond_mora	OK
passo_qt_auto	OK
passo_qt_bici	OK
passo_qt_moto	OK
passo_ren_fam	OK
passo_cd_renfam	OK

```
In [ ]: log_tela.info('Definindo as funções da família')
log_output.info('\n\n=====\\n')

def passo_tot_fam(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "TOT_FAM"
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe sem modificações
    """
    log_tela.info("### PASSO " + str(passo) + " - TOT_FAM")
    log_output.info('\\n\\n=====\\n')

    return df

def passo_f_fam(passo, df):
    """
    Checar se existe algum erro na coluna "F_FAM"

    # #####Categorias
    # Valor/Descrição
    # ----/----
    # 0/Demais registros
    # 1/Primeiro Registro da Família

    [Teste: Checar se existe algum número diferente de 0 ou 1.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe sem modificações
    """
    log_tela.info("### PASSO " + str(passo) + " - F_FAM")

    verifica_dummy(df, 'F_FAM')
```

```

log_output.info('\n\n=====\\n')
return df

def passo_fe_fam(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "FE_FAM"
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe sem modificações
    """
    log_tela.info("### PASSO " + str(passo) + " - FE_FAM")
    log_output.info('\n\n=====\\n')

    return df

def passo_cond_mora(passo, df):
    """
    Substituir valores da coluna "COND_MORA"

    * Substituir todos valores **5** por **0**
    * Substituir todos valores **0** por **None**
    * Substituir todos valores **4** por **3**

    ##### Categorias anteriores

    Valor/Descrição
    ----/----
    1/Alugada
    2/Própria
    3/Cedida
    4/Outros
    5/Não respondeu

    ##### Categorias novas

    Valor/Descrição
    ----/----
    1/Alugada
    2/Própria
    3/Outros

    [Teste: Checar se existe algum número < 1 ou > 3.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: passo
    :param df: dataframe
    :return: dataframe modificado
    """

    log_tela.info("### PASSO " + str(passo) + " - COND_MORA")

    # Substituindo valor 5 por 0
    df.loc[df['COND_MORA']==5,'COND_MORA'] = 0

```

```

# Substituindo valor 0 por None
df.loc[df['COND_MORA']==0,'COND_MORA'] = None
# Substituindo valor 4 por 3
df.loc[df['COND_MORA']==4,'COND_MORA'] = 3

# Verificando intervalo de valores - condições:
# "COND_MORA >= 1" E "COND_MORA <= 3"
verifica_range(df, 'COND_MORA', 1, 3)
log_output.info('\'\n\'=====\\n')

return df

def passo_qt_auto(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "QT_AUTO"
    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - QT_AUTO")
    log_output.info('\'\n\'=====\\n')

    return df

def passo_qt_bici(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "QT_AUTO"
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: dataframe modificado
    """
    log_tela.info('### PASSO ' + str(passo) + ' - QT_BICI')
    log_output.info('\'\n\'=====\\n')

    return df

def passo_qt_moto(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "QT_AUTO"
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: dataframe modificado
    """
    log_tela.info('### PASSO ' + str(passo) + ' - QT_MOTO')
    log_output.info('\'\n\'=====\\n')

    return df

def passo_ren_fam(passo, df, deflator):
    """

```

Corrigir a renda familiar de acordo com o deflator passado na função.

:param passo: Número do passo atual para registro/log

:param df: dataframe

:param deflator: Deflator utilizado para correção da renda.

:return: retorna o dataframe com as devidas modificações

"""

```
log_tela.info("### PASSO " + str(passo) + " - REN_FAM")

df['REN_FAM'] = df['REN_FAM'] * deflator

return df
```

def passo_cd_renfam(passo, df):

"""

Substituir valores da coluna "CD_RENFAM"

** Substituir todos valores **2** por **0***

** Substituir todos valores **3** por **2***

** Substituir todos valores **4** por **2***

Categorias anteriores

Valor/Descrição

----/----

1/Renda familiar declarada e maior que zero

2/Renda familiar declarada como zero

3/Renda atribuída pelo Critério Brasil

4/Renda atribuída pela Média da Zona

Categorias novas

Valor/Descrição

----/----

0/Renda Familiar Declarada como Zero

1/Renda Familiar Declarada e Maior que Zero

2/Renda Atribuída

[Teste: Checar se existe algum número < 0 ou > 2.

Se encontrar, retornar erro indicando em qual linha.]

:param passo: Número do passo atual para registro/log

:param df:

:return: retorna o dataframe corrigido

"""

```
log_tela.info("### PASSO " + str(passo) + " - CD_RENFAM")

# Substituindo valor 2 por 0
df.loc[df['CD_RENFAM']==2, 'CD_RENFAM'] = 0
# Substituindo valor 3 por 2
df.loc[df['CD_RENFAM']==3, 'CD_RENFAM'] = 2
# Substituindo valor 4 por 2
df.loc[df['CD_RENFAM']==4, 'CD_RENFAM'] = 2

# Dividindo a categoria '0', "Respondeu", em:
#   - 0 "Renda Familiar Declarada como Zero" e
#   - 1 "Renda Familiar Declarada e Maior que Zero"
```

```
df.loc[(df['CD_RENFAM'] == 0) &
       (df['REN_FAM'] != 0) &
       (df['REN_FAM'].notnull()), 'CD_RENFAM'] = 1

# Verificando intervalo de valores - condições:
    # "CD_RENFAM >= 0" E "CD_RENFAM <= 2"
verifica_range(df, 'CD_RENFAM', 0, 2)
log_output.info('\\n\\n=====\\n')

return df
```

2.5 Funções da pessoa

Função/Variável	Status
passo_f_pess	OK
passo_fe_pess	OK
passo_sit_fam	OK
passo_idade	OK
passo_sexo	OK
passo_grau_instr	OK
passo_ocup	OK
passo_setor_ativ	OK
passo_ren_ind	OK
passo_cd_renind	OK
passo_estuda	Verificar

Obs.: Se ESTUDA for ser alterado pela verificação do ZONA_ESC, no main() ele deverá ser chamado após ZONA_ESC. Atualmente ele já está localizado após o ZONA_ESC na chamada do main(). Após confirmar o que será feito com ele, se for necessário alterar sua localização.

```
In [ ]: log_tela.info('Definindo as funções da pessoa')
log_output.info('\n\n=====\\n')

def passo_f_pess(passo, df):
    """
    Checar se existe algum erro na coluna "F_PESS"

    # #####Categorias
    # Valor/Descrição
    # ----/----
    # 0/Demais registros
    # 1/Primeiro Registro da Pessoa

    [Teste: Checar se existe algum número diferente de 0 ou 1.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - F_PESS")

    verifica_dummy(df, 'F_PESS')
    log_output.info('\n\n=====\\n')

    return df

def passo_fe_pess(passo, df):
```

```

"""
Nada há que se fazer em relação aos dados das colunas "FE_PESS"
:param passo: Número do passo atual para registro/log
:param df:
:return:
"""
log_tela.info("### PASSO " + str(passo) + " - FE_PESS")
log_output.info('\n\n=====\\n')

return df

def passo_sit_fam(passo, df):
    """
    * Substituir todos valores **5** por **4**
    * Substituir todos valores **6** por **5**
    * Substituir todos valores **7** por **6**

    ##### Categorias anteriores
    Valor/Descrição
    ----/----
    1/Pessoa Responsável
    2/Cônjuge/Companheiro(a)
    3/Filho(a)/Enteado(a)
    4/Outro Parente
    5/Agregado
    6/Empregado Residente
    7/Parente do empregado

    ##### Categorias novas:
    Valor/Descrição
    ----/----
    1/ Pessoa Responsável
    2/ Cônjuge/Companheiro(a)
    3/ Filho(a)/Enteado(a)
    4/ Outro Parente / Agregado
    5/ Empregado Residente
    6/ Outros (visitante não residente / parente do empregado)

    [Teste: Checar se existe algum número < 1 ou > 6.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe modificado
    """
    log_tela.info("### PASSO " + str(passo) + " - SIT_FAM")

    # Substituindo valor 5 por 4
    df.loc[df['SIT_FAM']==5,'SIT_FAM'] = 4
    # Substituindo valor 6 por 5
    df.loc[df['SIT_FAM']==6,'SIT_FAM'] = 5
    # Substituindo valor 7 por 6
    df.loc[df['SIT_FAM']==7,'SIT_FAM'] = 6

```

```

# Verificando intervalo de valores - condições:
# "SIT_FAM >= 1" E "SIT_FAM <= 6"
verifica_range(df, 'SIT_FAM', 1, 6)
log_output.info('\'\n\'=====\'\n')

return df

def passo_idade(passo, df):
    """
    Nada há que se fazer em relação aos dados da coluna "IDADE"
    :param passo: Número do passo atual para registro/log
    :param df:
    :return:
    """
    log_tela.info("### PASSO " + str(passo) + " - IDADE")
    log_output.info('\'\n\'=====\'\n')

    return df

def passo_sexo(passo, df):
    """
    #####Categorias anteriores
    # Valor/Descrição
    # ----/----
    # 0/Não Respondeu (-> None)
    # 1/Masculino
    # 2/Feminino

    #####Categorias novas
    # Valor/Descrição
    # ----/----
    # 0/Masculino
    # 1/Feminino

    [Teste: Checar se existe algum número diferente de 0 ou 1.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe modificado
    """
    log_tela.info("### PASSO " + str(passo) + " - SEXO")

    # Substituindo valor 0 por None
    df.loc[df['SEXO']==0,'SEXO'] = None
    # Substituindo valor 1 por 0
    df.loc[df['SEXO']==1,'SEXO'] = 0
    # Substituindo valor 2 por 1
    df.loc[df['SEXO']==2,'SEXO'] = 1

    # Verificando intervalo de valores - condições:
    # "SEXO >= 0" E "SEXO <= 1"
    verifica_dummy(df, 'SEXO')

```

```

log_output.info('\n\n=====\\n')

return df

def passo_grau_instr(passo, df):
    """
    Substituir valores da coluna "GRAU_INSTR"

    * Substituir todos valores **2** por **1**
    * Substituir todos valores **3** por **2**
    * Substituir todos valores **4** por **3**
    * Substituir todos valores **5** por **4**
    * Substituir todos valores **0** por **None**

    ##### Categorias anteriores:
    Valor/Descrição
    ----/----
    1/Não-alfabetizado/Primario Incompleto
    2/Primario Completo/Ginasio Incompleto
    3/Ginásio Completo/Colegial Incompleto
    4/Colegial Completo/Superior Incompleto
    5/Superior Completo

    ##### Categorias novas
    Valor/Descrição
    ----/----
    1/Não-Alfabetizado/Fundamental Incompleto
    2/Fundamental Completo/Médio Incompleto
    3/Médio Completo/Superior Incompleto
    4/Superior completo

    [Teste: Checar se existe algum número < 1 ou > 4.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna dataframe modificado
    """
    log_tela.info("### PASSO " + str(passo) + " - GRAU_INSTR")

    # Substituindo valor 2 por 1
    df.loc[df['GRAU_INSTR']==2, 'GRAU_INSTR'] = 1
    # Substituindo valor 3 por 2
    df.loc[df['GRAU_INSTR']==3, 'GRAU_INSTR'] = 2
    # Substituindo valor 4 por 3
    df.loc[df['GRAU_INSTR']==4, 'GRAU_INSTR'] = 3
    # Substituindo valor 5 por 4
    df.loc[df['GRAU_INSTR']==5, 'GRAU_INSTR'] = 4
    # Substituindo valor 0 por None
    df.loc[df['GRAU_INSTR']==0, 'GRAU_INSTR'] = None

    # Verificando intervalo de valores - condições:
    # "GRAU_INSTR >= 1" E "GRAU_INSTR <= 4"
    verifica_range(df, 'GRAU_INSTR', 1, 4)

```

```

log_output.info('\'\n\n=====
\n')

return df

def passo_ocup(passo, df):
    """
    Substituir valores da coluna "OCUP"

    Substituir todos valores **2** por **1**
    Substituir todos valores **3** por **2**
    Substituir todos valores **4** por **3**
    Substituir todos valores **5** por **4**
    Substituir todos valores **6** por **5**
    Substituir todos valores **7** por **6**
    Substituir todos valores **8** por **7**
    Substituir todos valores **0** por **None**

    #####Categorias anteriores:
    Valor/Descrição
    ----/----
    1/Tem trabalho
    2/Faz bico
    3/Em licença médica
    4/Aposentado / pensionista
    5/Sem trabalho
    6/Nunca trabalhou
    7/Dona de casa
    8/Estudante

    #####Categorias novas
    Valor/Descrição
    ----/----
    1/Tem trabalho
    2/Em licença médica
    3/Aposentado / pensionista
    4/Desempregado
    5/Sem ocupação
    6/Dona de casa
    7/Estudante

[Teste: Checar se existe algum número < 1 ou > 7.
Se encontrar, retornar erro indicando em qual linha.]"
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna dataframe modificado
"""

log_tela.info("### PASSO " + str(passo) + " - OCUP")

# Substituindo valor 2 por 1
df.loc[df['OCUP']==2,'OCUP'] = 1
# Substituindo valor 3 por 2
df.loc[df['OCUP']==3,'OCUP'] = 2
# Substituindo valor 4 por 3
df.loc[df['OCUP']==4,'OCUP'] = 3

```

```

# Substituindo valor 5 por 4
df.loc[df['OCUP']==5,'OCUP'] = 4
# Substituindo valor 6 por 5
df.loc[df['OCUP']==6,'OCUP'] = 5
# Substituindo valor 7 por 6
df.loc[df['OCUP']==7,'OCUP'] = 6
# Substituindo valor 8 por 7
df.loc[df['OCUP']==8,'OCUP'] = 7
# Substituindo valor 0 por None
df.loc[df['OCUP']==0,'OCUP'] = None

# Verificando intervalo de valores - condições:
# "OCUP >= 1" E "OCUP <= 7"
verifica_range(df, 'OCUP', 1, 7)
log_output.info('\'\n\'=====\\n')

return df

def passo_setor_ativ(passo, df):
    """
    Substituir valores da coluna "SETOR_ATIV"

    Na coluna "SETOR_ATIV", linha i,
        ler o valor da linha i da coluna "SETOR_ATIV", daí,
        buscar o mesmo valor na coluna "COD" do arquivo setor_ativ-2007.csv.
    Ao achar, retornar o valor da mesma linha, só que da coluna "COD_UNIF"

    #####Categorias anteriores
    > ver arquivo .csv

    #####Categorias novas
    Valor/Descrição
    ----/----
    1/Agrícola
    2/Construção Civil
    3/Indústria
    4/Comércio
    5/Administração Pública
    6/Serviços de Transporte
    7/Doutros serviços
    8/Doutros
    9/Não se aplica

    [Teste: Checar se existe algum número < 1 ou > 9.
     Se encontrar, retornar erro indicando em qual linha.]"
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna dataframe modificado
    """
    log_tela.info("### PASSO " + str(passo) + " - SETOR_ATIV")

    log_output.info('Lendo arquivo de referência externa setor_ativ-2007.csv')
    df_setor = pd.read_csv('auxiliares/setor_ativ-2007.csv', sep=';', decimal=',')

```

```

def setor_aux(row):
    df.loc[df['SETOR_ATIV']==row['COD'], 'SETOR_ATIV'] = row['COD_UNIF']

# Esta variável out não é utilizada para nada além de evitar um
# monte de output que não será utilizado e que é gerado pelo método apply.
out = df_setor.apply(setor_aux, axis=1)

# Substituindo valor 0 por None
df.loc[df['SETOR_ATIV']==0, 'SETOR_ATIV'] = None

# Verificando intervalo de valores - condições:
    # "SETOR_ATIV >= 1" E "SETOR_ATIV <= 9"
verifica_range(df, 'SETOR_ATIV', 1, 9)
log_output.info('\n\n=====\\n')

return df


def passo_ren_ind(passo, df, deflator):
    """
    Corriga a renda individual de acordo com o deflator passado na função.
    :param passo: Número do passo atual para registro/log
    :param df: dataframe
    :param deflator: Deflator utilizado para correção da renda.
    :return: retorna o dataframe com as devidas modificações
    """
    log_tela.info("### PASSO " + str(passo) + " - REN_IND")

    df['REN_IND'] = df['REN_IND'] * deflator

    return df


def passo_cd_renind(passo, df):
    """
    Substituir valores da coluna "CD_RENIND"

    * Substituir todos valores **3** por None
    * Substituir todos valores **2** por **0**

    ##### Categorias anteriores
    Valor/Descrição
    ----/----
    1/Tem renda
    2/Não tem renda
    3/Não respondeu

    #####Categorias novas
    Valor/Descrição
    ----/-----
    0    /Não tem renda
    1    /Tem renda
    """

```

```

[Teste: Checar se existe algum número < 0 ou > 1.
Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe com as devidas modificações
"""
log_tela.info("### PASSO " + str(passo) + " - CD_RENIND")

df.loc[df['CD_RENIND']==3,'CD_RENIND'] = None
df.loc[df['CD_RENIND']==2,'CD_RENIND'] = 0

# Verificando intervalo de valores - condições:
# "CD_RENIND >= 0" E "CD_RENIND <= 1"
verifica_dummy(df, 'CD_RENIND')
log_output.info('\n\n=====\\n')

return df

def passo_estuda(passo, df):
"""
Se zona da escola for zero (0) então não estuda (0), senão, estuda (1)

#####
#      ATENÇÃO      #
# ESTA FUNÇÃO SÓ DEVE SER #
# EXECUTADA APÓS A GERAÇÃO#
# DE TODOS OS ID'S, POIS #
# HÁ UMA DESCONFIANÇA   #
# QUANTO À QUALIDADE    #
#      DESTE DADO       #
#####

* Substituir todos valores **1** por **0**
* Substituir todos valores **2** por **1**
* Substituir todos valores **3** por **1**
* Substituir todos valores **4** por **1**
* Substituir todos valores **5** por **1**
* Substituir todos valores **6** por **1**

#### Categorias anteriores
Valor/Descrição
----/----
1/Não
2/Creche/Pré-Escola
3/1º Grau/Fundamental
4/2º Grau/Médio
5/Superior/Universitário
6/Outros

#### Categorias novas
Valor/Descrição
----/----
0/Não estuda

```

1 / Estuda

```
[Teste: Checar se existe algum número diferente de 0 ou 1.  
Se encontrar, retornar erro indicando em qual linha.]  
:param passo: Número do passo atual para registro/log  
:param df:  
:return: retorna dataframe com campo ESCOLA resolvido  
"""  
log_tela.info("### PASSO " + str(passo) + " - ESTUDA")  
  
# Substitui 1 por 0  
df.loc[df['ESTUDA'] == 1, 'ESTUDA'] = 0  
# Substitui 2 por 1  
df.loc[df['ESTUDA'] == 2, 'ESTUDA'] = 1  
# Substitui 3 por 1  
df.loc[df['ESTUDA'] == 3, 'ESTUDA'] = 1  
# Substitui 4 por 1  
df.loc[df['ESTUDA'] == 4, 'ESTUDA'] = 1  
# Substitui 5 por 1  
df.loc[df['ESTUDA'] == 5, 'ESTUDA'] = 1  
# Substitui 6 por 1  
df.loc[df['ESTUDA'] == 6, 'ESTUDA'] = 1  
  
# Substituindo todos que declararam zona escola diferente de zero  
# com campo ESTUDA igual a 1.  
df.loc[(df['ZONA_ESC'] != 0)&  
       (df['ZONA_ESC'].notnull()), 'ESTUDA'] = 1  
  
verifica_dummy(df, 'ESTUDA')  
log_output.info('\n\n=====\\n')  
  
return df
```

2.6 Funções referentes da Viagem

Função/Variável	Status
passo_f_vitag	OK
passo_fe_vitag	OK
passo_zona_esc	OK
passo_subzona_esc	OK
passo_mun_esc	OK
passo_zona_trab1	OK
passo_subzona_trab1	OK
passo_mun_trab1	OK
passo_zona_trab2	OK
passo_subzona_trab2	OK
passo_mun_trab2	OK
passo_zona_orig	OK
passo_subzona_orig	OK
passo_mun_orig	OK
passo_zona_dest	OK
passo_subzona_dest	OK
passo_mun_dest	OK
passo_serv_pas_orig	OK
passo_serv_pas_dest	OK
passo_motivo_orig	OK
passo_motivo_dest	OK
passo_modo1	OK
passo_modo2	OK
passo_modo3	OK
passo_modo4	OK
passo_modo_prin	OK
passo_tipo_est_auto	OK
passo_valor_est_auto	OK
passo_tot_vitag	OK

Obs: O passo_tot_vitag só deve ser executado após a produção dos ID's e NO's.

```
In [ ]: log_tela.info('Definindo funções referentes às viagens')
log_output.info('\n'=====\\n')
```

```
def passo_f_vitag(passo, df):
```

```

"""
Checar se existe algum erro na coluna "F_VIAG"

# #####Categorias
# Valor/Descrição
# ----/----
# 0/Não fez viagem
# 1/Fez viagem

[Teste: Checar se existe algum número diferente de 0 ou 1.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe com F_VIAG modificado
"""

log_tela.info("### PASSO " + str(passo) + " - F_VIAG")

# Setando F_VIAG=1 para todos os casos
df['F_VIAG'] = 1

# Definindo F_VIAG=0 se DURACAO==0
df.loc[df['DURACAO']==0, 'F_VIAG'] = 0

verifica_dummy(df, 'F_VIAG')
log_output.info('\n\n=====\\n')

return df


def passo_fe_viag(passo, df):
"""
# Nada há que se fazer em relação aos dados da coluna "FE_VIAG"
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe sem modificações
"""

log_tela.info("### PASSO " + str(passo) + " - FE_VIAG")
log_output.info('\n\n=====\\n')

return df


def passo_zona_esc(passo, df):
"""
Checar se existe algum erro

# #####Categorias:
# > 1 a 460

[Teste: Checar se existe algum número < 1 ou > 460.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe sem modificações
"""

```

```

"""
log_tela.info("### PASSO " + str(passo) + " - ZONA_ESC")

# Substituindo valor 0 por None
df.loc[df['ZONA_ESC']==0,'ZONA_ESC'] = None

# Verificando intervalo de valores - condições:
# "ZONA_ESC >= 1" E "ZONA_ESC <= 460"
verifica_range(df, 'ZONA_ESC', 1, 460)
log_output.info('\n\n=====\\n')

return df
"""

def passo_subzona_esc(passo, df):
    """
    A variável SUBZONA_ESC ficará vazia (NA) em 2007 pois este ano já contém os dados de coordenadas.

    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe sem modificações
    """
    log_tela.info("### PASSO " + str(passo) + " - SUBZONA_ESC")

    df['SUBZONA_ESC'] = None

    log_output.info('\n\n=====\\n')

    return df

def passo_mun_esc(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias
    # > 1 a 39

    [Teste: Checar se existe algum número < 1 ou > 39.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe sem modificações
    """
    log_tela.info("### PASSO " + str(passo) + " - MUN_ESC")

    # Substituindo valor 0 por None
    df.loc[df['MUN_ESC']==0,'MUN_ESC'] = None

    # Verificando intervalo de valores - condições:
    # "MUN_ESC >= 1" E "MUN_ESC <= 39"
    verifica_range(df, 'MUN_ESC', 1, 39)
    log_output.info('\n\n=====\\n')

```

```

    return df

def passo_zona_trab1(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 460

    [Teste: Checar se existe algum número < 1 ou > 460.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe sem modificações
    """
    log_tela.info("### PASSO " + str(passo) + " - ZONA_TRAB1")

    # Substituindo valor 0 por None
    df.loc[df['ZONA_TRAB1']==0,'ZONA_TRAB1'] = None

    # Verificando intervalo de valores - condições:
    # "ZONA_TRAB1 >= 1" E "ZONA_TRAB1 <= 460"
    verifica_range(df, 'ZONA_TRAB1', 1, 460)
    log_output.info('\n\n=====\\n')

    return df

def passo_subzona_trab1(passo, df):
    """
    A variável SUBZONA_TRAB1 ficará vazia (NA) em 2007 pois este ano já contém
    os dados de coordenadas.

    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe sem modificações
    """
    log_tela.info("### PASSO " + str(passo) + " - SUBZONA_TRAB1")

    df['SUBZONA_TRAB1'] = None

    log_output.info('\n\n=====\\n')

    return df

def passo_mun_trab1(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias
    # > 1 a 39

```

```

[Teste: Checar se existe algum número < 1 ou > 39.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe sem modificações
"""

log_tela.info("### PASSO " + str(passo) + " - MUN_TRAB1")

# Substituindo valor 0 por None
df.loc[df['MUN_TRAB1']==0,'MUN_TRAB1'] = None

# Verificando intervalo de valores - condições:
# "MUN_TRAB1 >= 1" E "MUN_TRAB1 <= 39"
verifica_range(df, 'MUN_TRAB1', 1, 39)
log_output.info('\n\n=====\\n')

return df


def passo_zona_trab2(passo, df):
"""
Checar se existe algum erro

# #####Categorias:
# > 1 a 460

[Teste: Checar se existe algum número < 1 ou > 460.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe sem modificações
"""

log_tela.info("### PASSO " + str(passo) + " - ZONA_TRAB2")

# Substituindo valor 0 por None
df.loc[df['ZONA_TRAB2']==0,'ZONA_TRAB2'] = None

# Verificando intervalo de valores - condições:
# "ZONA_TRAB2 >= 1" E "ZONA_TRAB2 <= 460"
verifica_range(df, 'ZONA_TRAB2', 1, 460)
log_output.info('\n\n=====\\n')

return df


def passo_subzona_trab2(passo, df):
"""
A variável SUBZONA_TRAB2 ficará vazia (NA) em 2007 pois este ano já contém
os dados de coordenadas.

:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe sem modificações

```

```

"""
log_tela.info("### PASSO " + str(passo) + " - SUBZONA_TRAB2")
df['SUBZONA_TRAB2'] = None
log_output.info('\n\n=====\\n')
return df

def passo_mun_trab2(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias
    # > 1 a 39

    [Teste: Checar se existe algum número < 1 ou > 39.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe sem modificações
    """
    log_tela.info("### PASSO " + str(passo) + " - MUN_TRAB2")

    # Substituindo valor 0 por None
    df.loc[df['MUN_TRAB2']==0, 'MUN_TRAB2'] = None

    # Verificando intervalo de valores - condições:
    # "MUN_TRAB2 >= 1" E "MUN_TRAB2 <= 39"
    verifica_range(df, 'MUN_TRAB2', 1, 39)
    log_output.info('\n\n=====\\n')

    return df

def passo_zona_orig(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias:
    # > 1 a 460

    [Teste: Checar se existe algum número < 1 ou > 460.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe sem modificações
    """
    log_tela.info("### PASSO " + str(passo) + " - ZONA_ORIG")

    # Substituindo valor 0 por None
    df.loc[df['ZONA_ORIG']==0, 'ZONA_ORIG'] = None

    # Verificando intervalo de valores - condições:

```

```

# "ZONA_ORIG >= 1" E "ZONA_ORIG <= 460"
verifica_range(df, 'ZONA_ORIG', 1, 460)
log_output.info('\'\n\'=====\'\n')

return df

def passo_subzona_orig(passo, df):
    """
    A variável SUBZONA_ORIG ficará vazia (NA) em 2007 pois este ano já contém os dados de coordenadas.

    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe sem modificações
    """
    log_tela.info("### PASSO " + str(passo) + " - SUBZONA_ORIG")

    df['SUBZONA_ORIG'] = None

    log_output.info('\'\n\'=====\'\n')

    return df

def passo_mun_orig(passo, df):
    """
    Checar se existe algum erro

    # #####Categorias
    # > 1 a 39

    [Teste: Checar se existe algum número < 1 ou > 39.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: retorna o dataframe sem modificações
    """
    log_tela.info("### PASSO " + str(passo) + " - MUN_ORIG")

    # Substituindo valor 0 por None
    df.loc[df['MUN_ORIG']==0, 'MUN_ORIG'] = None

    # Verificando intervalo de valores - condições:
    # "MUN_ORIG >= 1" E "MUN_ORIG <= 39"
    verifica_range(df, 'MUN_ORIG', 1, 39)
    log_output.info('\'\n\'=====\'\n')

    return df

def passo_zona_dest(passo, df):
    """
    Checar se existe algum erro

```

```

# #####Categorias:
# > 1 a 460

[Teste: Checar se existe algum número < 1 ou > 460.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe sem modificações
"""
log_tela.info("### PASSO " + str(passo) + " - ZONA_DEST")

# Substituindo valor 0 por None
df.loc[df['ZONA_DEST']==0,'ZONA_DEST'] = None

# Verificando intervalo de valores - condições:
# "ZONA_DEST >= 1" E "ZONA_DEST <= 460"
verifica_range(df, 'ZONA_DEST', 1, 460)
log_output.info('\n\n=====\\n')

return df

def passo_subzona_dest(passo, df):
"""
A variável SUBZONA_DEST ficará vazia (NA) em 2007 pois este ano já contém
os dados de coordenadas.

:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe sem modificações
"""
log_tela.info("### PASSO " + str(passo) + " - SUBZONA_DESTM")

df['SUBZONA_DEST'] = None

log_output.info('\n\n=====\\n')

return df

def passo_mun_dest(passo, df):
"""
Checar se existe algum erro

# #####Categorias
# > 1 a 39

[Teste: Checar se existe algum número < 1 ou > 39.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe sem modificações
"""

```

```

log_tela.info("### PASSO " + str(passo) + " - MUN_DEST")

# Substituindo valor 0 por None
df.loc[df['MUN_DEST']==0,'MUN_DEST'] = None

# Verificando intervalo de valores - condições:
# "MUN_DEST >= 1" E "MUN_DEST <= 39"
verifica_range(df, 'MUN_DEST', 1, 39)
log_output.info('\n\n=====\\n')

return df

def passo_serv_pas_orig(passo, df):
    """
    Substituir **0** por None
    Substituir **2** por **0**

    #####Categorias antigas
    Valor/Descrição
    ----/----
    0/Não Respondido
    1/Sim
    2/Não

    #####Categorias novas
    Valor/Descrição
    ----/----
    0/Não
    1/Sim

:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe sem modificações
"""
log_tela.info("### PASSO " + str(passo) + " - SERV_PAS_ORIG")

df.loc[df['SERV_PAS_ORIG']==0,'SERV_PAS_ORIG'] = None
df.loc[df['SERV_PAS_ORIG']==2,'SERV_PAS_ORIG'] = 0

verifica_dummy(df, 'SERV_PAS_ORIG')
log_output.info('\n\n=====\\n')

return df

def passo_serv_pas_dest(passo, df):
    """
    Substituir **0** por None
    Substituir **2** por **0**

    #####Categorias antigas
    Valor/Descrição
    ----/----

```

```

0/Não Respondido
1/Sim
2/Não

####Categorias novas
Valor/Descrição
----/----
0/Não
1/Sim

:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe sem modificações
"""
log_tela.info("### PASSO " + str(passo) + " - SERV_PAS_DEST")

df.loc[df['SERV_PAS_DEST']==0,'SERV_PAS_DEST'] = None
df.loc[df['SERV_PAS_DEST']==2,'SERV_PAS_DEST'] = 0

verifica_dummy(df, 'SERV_PAS_DEST')
log_output.info('\n\n=====\\n')

return df

def passo_motivo_orig(passo, df):
"""

* Substituir todos valores **10** por **9**
* Substituir todos valores **0** por **None**

#### Categorias anteriores
Valor/Descrição
----/----
1/Trabalho/Indústria
2/Trabalho/Comércio
3/Trabalho/Serviços
4/Educação
5/Compras
6/Saúde
7/Lazer
8/Residência
9/Procurar Emprego
10/Assuntos Pessoais

#### Categorias novas
Valor/Descrição
----/----
1/Trabalho/Indústria
2/Trabalho/Comércio
3/Trabalho/Serviços
4/Educação
5/Compras
6/Saúde

```

```

7/Lazer
8/Residência
9/Doutros

[Teste: Checar se existe algum número < 1 ou > 9.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: dataframe modificado
"""
log_tela.info("### PASSO " + str(passo) + " - MOTIVO_ORIG")

# Substitui 10 por 9
df.loc[df['MOTIVO_ORIG'] == 10, 'MOTIVO_ORIG'] = 9
# Substitui 0 por None
df.loc[df['MOTIVO_ORIG'] == 0, 'MOTIVO_ORIG'] = None

# Verificando intervalo de valores - condições:
# "MOTIVO_ORIG >= 1" E "MOTIVO_ORIG <= 9"
verifica_range(df, 'MOTIVO_ORIG', 1, 9)
log_output.info('\n\n=====\\n')

return df

def passo_motivo_dest(passo, df):
"""

* Substituir todos valores **10** por **9**
* Substituir todos valores **0** por **None**

#### Categorias anteriores
Valor/Descrição
----/----
1/Trabalho/Indústria
2/Trabalho/Comércio
3/Trabalho/Serviços
4/Educação
5/Compras
6/Saúde
7/Lazer
8/Residência
9/Procurar Emprego
10/Assuntos Pessoais

#### Categorias novas
Valor/Descrição
----/----
1/Trabalho/Indústria
2/Trabalho/Comércio
3/Trabalho/Serviços
4/Educação
5/Compras
6/Saúde

```

```

7/Lazer
8/Residência
9/Doutros

[Teste: Checar se existe algum número < 1 ou > 9.
 Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: dataframe modificado
"""
log_tela.info("### PASSO " + str(passo) + " - MOTIVO_DEST")

# Substitui 10 por 9
df.loc[df['MOTIVO_DEST'] == 10, 'MOTIVO_DEST'] = 9
# Substitui 10 por 9
df.loc[df['MOTIVO_DEST'] == 0, 'MOTIVO_DEST'] = None

# Verificando intervalo de valores - condições:
# "MOTIVO_DEST >= 1" E "MOTIVO_DEST <= 9"
verifica_range(df, 'MOTIVO_DEST', 1, 9)
log_output.info('\n\n=====\\n')

return df

def passo_modo1(passo, df):
"""
Substituir valores da coluna "MOD01"

* Substituir todos valores **2** por **1**
* Substituir todos valores **3** por **1**
* Substituir todos valores **4** por **2**
* Substituir todos valores **5** por **2**
* Substituir todos valores **6** por **3**
* Substituir todos valores **7** por **4**
* Substituir todos valores **8** por **5**
* Substituir todos valores **9** por **6**
* Substituir todos valores **10** por **6**
* Substituir todos valores **11** por **6**
* Substituir todos valores **12** por **7**
* Substituir todos valores **13** por **8**
* Substituir todos valores **14** por **9**
* Substituir todos valores **15** por **10**
* Substituir todos valores **16** por **11**
* Substituir todos valores **17** por **12**
* Substituir todos valores **0** por **None**

#### Categorias anteriores
Valor/Descrição
----/----
1/Onibus Municipio S.Paulo
2/Onibus Outros Municipios
3/Onibus Metropolitano
4/Onibus Fretado

```

```

5/Escolar
6/Dirigindo Automóvel
7/Passageiro de Automóvel
8/Taxi
9/Microônibus/Van Município de S.Paulo
10/Microônibus/Van Outros Municípios
11/Microônibus/Van Metropolitano
12/Metrô
13/Trem
14/Moto
15/Bicicleta
16/A Pé
17/Outros

```

```

#### Categorias novas
Valor/Descrição
----/----
1/Ônibus
2/Ônibus Escolar / Empresa
3/Dirigindo Automóvel
4/Passageiro de Automóvel
5/Táxi
6/Lotação / Peruá / Van / Microônibus
7/Metrô
8/Trem
9/Moto
10/Bicicleta
11/A Pé
12/Outros

```

```

[Teste: Checar se existe algum número < 1 ou > 12.
Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe com as devidas modificações
"""
log_tela.info("### PASSO " + str(passo) + " - MODO1")

df.loc[df['MODO1']==2,'MODO1'] = 1
df.loc[df['MODO1']==3,'MODO1'] = 1
df.loc[df['MODO1']==4,'MODO1'] = 2
df.loc[df['MODO1']==5,'MODO1'] = 2
df.loc[df['MODO1']==6,'MODO1'] = 3
df.loc[df['MODO1']==7,'MODO1'] = 4
df.loc[df['MODO1']==8,'MODO1'] = 5
df.loc[df['MODO1']==9,'MODO1'] = 6
df.loc[df['MODO1']==10,'MODO1'] = 6
df.loc[df['MODO1']==11,'MODO1'] = 6
df.loc[df['MODO1']==12,'MODO1'] = 7
df.loc[df['MODO1']==13,'MODO1'] = 8
df.loc[df['MODO1']==14,'MODO1'] = 9
df.loc[df['MODO1']==15,'MODO1'] = 10
df.loc[df['MODO1']==16,'MODO1'] = 11
df.loc[df['MODO1']==17,'MODO1'] = 12

```

```

df.loc[df['MODO1']==0,'MODO1'] = None

# Verificando intervalo de valores - condições:
# "MODO1 >= 1" E "MODO1 <= 12"
verifica_range(df, 'MODO1', 1, 12)
log_output.info('\n\n=====\\n')

return df

def passo_modo2(passo, df):
    """
    Substituir valores da coluna "MODO2"

    * Substituir todos valores **2** por **1**
    * Substituir todos valores **3** por **1**
    * Substituir todos valores **4** por **2**
    * Substituir todos valores **5** por **2**
    * Substituir todos valores **6** por **3**
    * Substituir todos valores **7** por **4**
    * Substituir todos valores **8** por **5**
    * Substituir todos valores **9** por **6**
    * Substituir todos valores **10** por **6**
    * Substituir todos valores **11** por **6**
    * Substituir todos valores **12** por **7**
    * Substituir todos valores **13** por **8**
    * Substituir todos valores **14** por **9**
    * Substituir todos valores **15** por **10**
    * Substituir todos valores **16** por **11**
    * Substituir todos valores **17** por **12**
    * Substituir todos valores **0** por **None**

    ##### Categorias anteriores
    Valor/Descrição
    ----/----
    1/Onibus Municipio S.Paulo
    2/Onibus Outros Municipios
    3/Onibus Metropolitano
    4/Onibus Fretado
    5/Escolar
    6/Dirigindo Automóvel
    7/Passageiro de Automóvel
    8/Taxi
    9/Microônibus/Van Município de S.Paulo
    10/Microônibus/Van Outros Municípios
    11/Microônibus/Van Metropolitano
    12/Metrô
    13/Trem
    14/Moto
    15/Bicicleta
    16/A Pé
    17/Outros

    ##### Categorias novas

```

Valor/Descrição

----/----

1/*Ônibus*
 2/*Ônibus Escolar / Empresa*
 3/*Dirigindo Automóvel*
 4/*Passageiro de Automóvel*
 5/*Táxi*
 6/*Lotação / Peruá / Van / Microônibus*
 7/*Metrô*
 8/*Trem*
 9/*Moto*
 10/*Bicicleta*
 11/*A Pé*
 12/*Outros*

[*Teste: Checar se existe algum número < 1 ou > 12.*
Se encontrar, retornar erro indicando em qual linha.]

```
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe com as devidas modificações
"""

log_tela.info("### PASSO " + str(passo) + " - MODO1")

df.loc[df['MODO2']==2,'MODO2'] = 1
df.loc[df['MODO2']==3,'MODO2'] = 1
df.loc[df['MODO2']==4,'MODO2'] = 2
df.loc[df['MODO2']==5,'MODO2'] = 2
df.loc[df['MODO2']==6,'MODO2'] = 3
df.loc[df['MODO2']==7,'MODO2'] = 4
df.loc[df['MODO2']==8,'MODO2'] = 5
df.loc[df['MODO2']==9,'MODO2'] = 6
df.loc[df['MODO2']==10,'MODO2'] = 6
df.loc[df['MODO2']==11,'MODO2'] = 6
df.loc[df['MODO2']==12,'MODO2'] = 7
df.loc[df['MODO2']==13,'MODO2'] = 8
df.loc[df['MODO2']==14,'MODO2'] = 9
df.loc[df['MODO2']==15,'MODO2'] = 10
df.loc[df['MODO2']==16,'MODO2'] = 11
df.loc[df['MODO2']==17,'MODO2'] = 12
df.loc[df['MODO2']==0,'MODO2'] = None

# Verificando intervalo de valores - condições:
# "MODO2 >= 1" E "MODO2 <= 12"
verifica_range(df, 'MODO2', 1, 12)
log_output.info('\n\n=====\\n')

return df

def passo_modo3(passo, df):
"""
Substituir valores da coluna "MODO3"

* Substituir todos valores **2** por **1**

```

* Substituir todos valores **3** por **1**
* Substituir todos valores **4** por **2**
* Substituir todos valores **5** por **2**
* Substituir todos valores **6** por **3**
* Substituir todos valores **7** por **4**
* Substituir todos valores **8** por **5**
* Substituir todos valores **9** por **6**
* Substituir todos valores **10** por **6**
* Substituir todos valores **11** por **6**
* Substituir todos valores **12** por **7**
* Substituir todos valores **13** por **8**
* Substituir todos valores **14** por **9**
* Substituir todos valores **15** por **10**
* Substituir todos valores **16** por **11**
* Substituir todos valores **17** por **12**
* Substituir todos valores **0** por **None**

Categorias anteriores

Valor/Descrição

----/----

- 1/Ônibus Município S.Paulo
- 2/Ônibus Outros Municípios
- 3/Ônibus Metropolitano
- 4/Ônibus Fretado
- 5/Escolar
- 6/Dirigindo Automóvel
- 7/Passageiro de Automóvel
- 8/Taxi
- 9/Microônibus/Van Município de S.Paulo
- 10/Microônibus/Van Outros Municípios
- 11/Microônibus/Van Metropolitano
- 12/Metrô
- 13/Trem
- 14/Moto
- 15/Bicicleta
- 16/A Pé
- 17/Outros

Categorias novas

Valor/Descrição

----/----

- 1/Ônibus
- 2/Ônibus Escolar / Empresa
- 3/Dirigindo Automóvel
- 4/Passageiro de Automóvel
- 5/Táxi
- 6/Lotação / Peruá / Van / Microônibus
- 7/Metrô
- 8/Trem
- 9/Moto
- 10/Bicicleta
- 11/A Pé
- 12/Outros

```

[Teste: Checar se existe algum número < 1 ou > 12.
Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe com as devidas modificações
"""

log_tela.info("### PASSO " + str(passo) + " - MODO1")

df.loc[df['MODO3']==2,'MODO3'] = 1
df.loc[df['MODO3']==3,'MODO3'] = 1
df.loc[df['MODO3']==4,'MODO3'] = 2
df.loc[df['MODO3']==5,'MODO3'] = 2
df.loc[df['MODO3']==6,'MODO3'] = 3
df.loc[df['MODO3']==7,'MODO3'] = 4
df.loc[df['MODO3']==8,'MODO3'] = 5
df.loc[df['MODO3']==9,'MODO3'] = 6
df.loc[df['MODO3']==10,'MODO3'] = 6
df.loc[df['MODO3']==11,'MODO3'] = 6
df.loc[df['MODO3']==12,'MODO3'] = 7
df.loc[df['MODO3']==13,'MODO3'] = 8
df.loc[df['MODO3']==14,'MODO3'] = 9
df.loc[df['MODO3']==15,'MODO3'] = 10
df.loc[df['MODO3']==16,'MODO3'] = 11
df.loc[df['MODO3']==17,'MODO3'] = 12
df.loc[df['MODO3']==0,'MODO3'] = None

# Verificando intervalo de valores - condições:
# "MODO3 >= 1" E "MODO3 <= 12"
verifica_range(df, 'MODO3', 1, 12)
log_output.info('\'\n\'=====\\n')

return df

def passo_modo4(passo, df):
    """
    Substituir valores da coluna "MODO4"

    * Substituir todos valores **2** por **1**
    * Substituir todos valores **3** por **1**
    * Substituir todos valores **4** por **2**
    * Substituir todos valores **5** por **2**
    * Substituir todos valores **6** por **3**
    * Substituir todos valores **7** por **4**
    * Substituir todos valores **8** por **5**
    * Substituir todos valores **9** por **6**
    * Substituir todos valores **10** por **6**
    * Substituir todos valores **11** por **6**
    * Substituir todos valores **12** por **7**
    * Substituir todos valores **13** por **8**
    * Substituir todos valores **14** por **9**
    * Substituir todos valores **15** por **10**
    * Substituir todos valores **16** por **11**
    * Substituir todos valores **17** por **12**

```

* Substituir todos valores **0** por **None**

```
#### Categorias anteriores
Valor/Descrição
----/----
1/Ônibus Município S.Paulo
2/Ônibus Outros Municípios
3/Ônibus Metropolitano
4/Ônibus Fretado
5/Escolar
6/Dirigindo Automóvel
7/Passageiro de Automóvel
8/Taxi
9/Microônibus/Van Município de S.Paulo
10/Microônibus/Van Outros Municípios
11/Microônibus/Van Metropolitano
12/Metrô
13/Trem
14/Moto
15/Bicicleta
16/A Pé
17/Outros
```

```
#### Categorias novas
Valor/Descrição
----/----
1/Ônibus
2/Ônibus Escolar / Empresa
3/Dirigindo Automóvel
4/Passageiro de Automóvel
5/Táxi
6/Lotação / Peruá / Van / Microônibus
7/Metrô
8/Trem
9/Moto
10/Bicicleta
11/A Pé
12/Outros
```

[Teste: Checar se existe algum número < 1 ou > 12.

Se encontrar, retornar erro indicando em qual linha.]

```
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe com as devidas modificações
"""
log_tela.info("### PASSO " + str(passo) + " - MODO1")
```

```
df.loc[df['MODO4']==2,'MODO4'] = 1
df.loc[df['MODO4']==3,'MODO4'] = 1
df.loc[df['MODO4']==4,'MODO4'] = 2
df.loc[df['MODO4']==5,'MODO4'] = 2
df.loc[df['MODO4']==6,'MODO4'] = 3
df.loc[df['MODO4']==7,'MODO4'] = 4
df.loc[df['MODO4']==8,'MODO4'] = 5
```

```

df.loc[df['MODO4']==9,'MODO4'] = 6
df.loc[df['MODO4']==10,'MODO4'] = 6
df.loc[df['MODO4']==11,'MODO4'] = 6
df.loc[df['MODO4']==12,'MODO4'] = 7
df.loc[df['MODO4']==13,'MODO4'] = 8
df.loc[df['MODO4']==14,'MODO4'] = 9
df.loc[df['MODO4']==15,'MODO4'] = 10
df.loc[df['MODO4']==16,'MODO4'] = 11
df.loc[df['MODO4']==17,'MODO4'] = 12
df.loc[df['MODO4']==0,'MODO4'] = None

# Verificando intervalo de valores - condições:
    # "MODO4 >= 1" E "MODO4 <= 12"
verifica_range(df, 'MODO4', 1, 12)
log_output.info(' \n\n===== \n')

return df

def passo_modo_prin(passo, df):
    """
    Substituir valores da coluna "MODO_PRIN"

    * Substituir todos valores **2** por **1**
    * Substituir todos valores **3** por **1**
    * Substituir todos valores **4** por **2**
    * Substituir todos valores **5** por **2**
    * Substituir todos valores **6** por **3**
    * Substituir todos valores **7** por **4**
    * Substituir todos valores **8** por **5**
    * Substituir todos valores **9** por **6**
    * Substituir todos valores **10** por **6**
    * Substituir todos valores **11** por **6**
    * Substituir todos valores **12** por **7**
    * Substituir todos valores **13** por **8**
    * Substituir todos valores **14** por **9**
    * Substituir todos valores **15** por **10**
    * Substituir todos valores **16** por **11**
    * Substituir todos valores **17** por **12**
    * Substituir todos valores **0** por **None**

    ##### Categorias anteriores
    Valor/Descrição
    ----/----
    1/Onibus Municipio S.Paulo
    2/Onibus Outros Municipios
    3/Onibus Metropolitano
    4/Onibus Fretado
    5/Escolar
    6/Dirigindo Automóvel
    7/Passageiro de Automóvel
    8/Taxi
    9/Microônibus/Van Município de S.Paulo
    10/Microônibus/Van Outros Municípios

```

```

11/Microônibus/Van Metropolitano
12/Metrô
13/Trem
14/Moto
15/Bicicleta
16/A Pé
17/Outros

##### Categorias novas
Valor/Descrição
----/----
1/Ônibus
2/Ônibus Escolar / Empresa
3/Dirigindo Automóvel
4/Passageiro de Automóvel
5/Táxi
6/Lotação / Peruá / Van / Microônibus
7/Metrô
8/Trem
9/Moto
10/Bicicleta
11/A Pé
12/Outros

[Teste: Checar se existe algum número < 1 ou > 12.
Se encontrar, retornar erro indicando em qual linha.]
```

:param passo: Número do passo atual para registro/log

:param df:

:return: retorna o dataframe com as devidas modificações

"""

```

log_tela.info("### PASSO " + str(passo) + " - MODO_PRIN")

df.loc[df['MODO_PRIN']==2, 'MODO_PRIN'] = 1
df.loc[df['MODO_PRIN']==3, 'MODO_PRIN'] = 1
df.loc[df['MODO_PRIN']==4, 'MODO_PRIN'] = 2
df.loc[df['MODO_PRIN']==5, 'MODO_PRIN'] = 2
df.loc[df['MODO_PRIN']==6, 'MODO_PRIN'] = 3
df.loc[df['MODO_PRIN']==7, 'MODO_PRIN'] = 4
df.loc[df['MODO_PRIN']==8, 'MODO_PRIN'] = 5
df.loc[df['MODO_PRIN']==9, 'MODO_PRIN'] = 6
df.loc[df['MODO_PRIN']==10, 'MODO_PRIN'] = 6
df.loc[df['MODO_PRIN']==11, 'MODO_PRIN'] = 6
df.loc[df['MODO_PRIN']==12, 'MODO_PRIN'] = 7
df.loc[df['MODO_PRIN']==13, 'MODO_PRIN'] = 8
df.loc[df['MODO_PRIN']==14, 'MODO_PRIN'] = 9
df.loc[df['MODO_PRIN']==15, 'MODO_PRIN'] = 10
df.loc[df['MODO_PRIN']==16, 'MODO_PRIN'] = 11
df.loc[df['MODO_PRIN']==17, 'MODO_PRIN'] = 12
df.loc[df['MODO_PRIN']==0, 'MODO_PRIN'] = None

# Verificando intervalo de valores - condições:
# "MODO_PRIN >= 1" E "MODO_PRIN <= 12"
verifica_range(df, 'MODO_PRIN', 1, 12)
log_output.info('\'\n\'=====\\n')
```

```

    return df

def passo_tipo_vitag(passo, df):
    """
    * Substituir os valores **0** por **None**

    # #####Categorias novas
    # Valor/Descrição
    # ----/----
    # 1/Coletivo
    # 2/Individual
    # 3/A pé

    [Teste: Checar se existe algum número < 1 ou > 3.
     Se encontrar, retornar erro indicando em qual linha.]
    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno

    :param passo: Número do passo atual para registro/log
    :param df:
    :return: sem retorno
    """
    log_tela.info("### PASSO " + str(passo) + " - TIPO_VIAG")

    # Substituindo valor 0 por None
    df.loc[df['TIPO_VIAG']==0,'TIPO_VIAG'] = None

    # Verificando intervalo de valores - condições:
    # "MODO_PRIN >= 1" E "MODO_PRIN <= 3"
    verifica_range(df, 'TIPO_VIAG', 1, 3)

    log_output.info('\n\n=====\\n')
    return df


def passo_tipo_est_auto(passo, df):
    """
    * Substituir todos valores **0** por None (NA)
    * Substituir todos valores **8** por None (NA)
    * Substituir todos valores **1** por **0**
    * Substituir todos valores **3** por **1**
    * Substituir todos valores **4** por **1**
    * Substituir todos valores **6** por **1**
    * Substituir todos valores **7** por **1**
    * Substituir todos valores **5** por **2**

    ##### Categorias anteriores
    Valor/Descrição
    ----/----
    0/Não Respondeu
    1/Não Estacionou
    2/Zona Azul/Zona Marrom

```

```

3/Estacionamento Patrocinado
4/Estacionamento Próprio
5/Meio-Fio
6/Avulso
7/Mensal
8/Não Respondeu

##### Categorias novas
Valor/Descrição
----/----
0/Não Estacionou
1/Estacionamento Particular (Avulso / Mensal / Próprio / Patrocinado)
2/Estacionamento Público (meio fio / zona azul / zona marrom / parquímetro)

[Teste: Checar se existe algum número < 0 ou > 5.
Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna dataframe modificado
"""

log_tela.info("### PASSO " + str(passo) + " - TIPO_EST_AUTO")

# Substituindo valor 0 por None
df.loc[df['TIPO_EST_AUTO']==0,'TIPO_EST_AUTO'] = None
# Substituindo valor 8 por None
df.loc[df['TIPO_EST_AUTO']==8,'TIPO_EST_AUTO'] = None
# Substituindo valor 1 por 0
df.loc[df['TIPO_EST_AUTO']==1,'TIPO_EST_AUTO'] = 0
# Substituindo valor 3 por 1
df.loc[df['TIPO_EST_AUTO']==3,'TIPO_EST_AUTO'] = 1
# Substituindo valor 4 por 1
df.loc[df['TIPO_EST_AUTO']==4,'TIPO_EST_AUTO'] = 1
# Substituindo valor 6 por 1
df.loc[df['TIPO_EST_AUTO']==6,'TIPO_EST_AUTO'] = 1
# Substituindo valor 7 por 1
df.loc[df['TIPO_EST_AUTO']==7,'TIPO_EST_AUTO'] = 1
# Substituindo valor 5 por 2
df.loc[df['TIPO_EST_AUTO']==5,'TIPO_EST_AUTO'] = 2

verifica_range(df, 'TIPO_EST_AUTO', 0, 2)
log_output.info('\n\n=====\\n')

return df

def passo_valor_est_auto(passo, df, deflator):
"""
Corrigir o valor da variável com o deflator passado.

:param passo: Número do passo atual para registro/log
:param df: dataframe a ser modificado
:return: dataframe com o VALOR_EST_AUTO corrigido pelo deflator
"""

```

```

log_tela.info("### PASSO " + str(passo) + " - VALOR_EST_AUTO")

df['VALOR_EST_AUTO'] = df['VALOR_EST_AUTO'] * deflator

log_output.info('\n\n=====\\n')

return df

def passo_tot_viag(passo, df):
    """
    #####
    # ATENÇÃO #
    # ESTA FUNÇÃO SÓ DEVE SER #
    # EXECUTADA APÓS A GERAÇÃO#
    # DE TODOS OS ID'S, POIS #
    # HÁ UMA DESCONFIANÇA #
    # QUANTO À QUALIDADE #
    # DESTE DADO #
    #####
    Calcula e confere o campo TOT_VIAG,
    baseado no maior valor de NO_VIAG para cada pessoa
:param passo: Número do passo atual para registro/log
:param df:
:return: retorna o dataframe com o "TOT_VIAG" corrigido
"""

log_tela.info("### PASSO " + str(passo) + " - TOT_VIAG")
log_tela.warning('\n\n#####\\n' +
                 'Este passo DEVE ser executado apenas após a \\n' +
                 'execução dos passos que geram os novos IDs\\n' +
                 '#####\\n' )

# 'Calculando' o máximo de viagens para cada indivíduo
# Agrupa por ID_PESS e encontra o máximo para NO_VIAG.
# O resultado é um objeto do tipo "Series" cujo index é
# ID_PESS e a variável é NO_VIAG. Em seguida transforma
# esse objeto num DataFrame e depois renomeia a coluna
# NO_VIAG para MAX_VIAG.
df_max = pd.DataFrame(
    df.groupby('ID_PESS')['NO_VIAG'].max()).rename(
        columns={'NO_VIAG': 'MAX_VIAG'})

# Criando um novo dataframe auxiliar apenas com ID_PESS, apenas para ficar mais leve

# O passo seguinte é atribuir o MAX_VIAG à coluna TOT_VIAG
# do dataframe df, linkando por ID_PESS. Isso é feito usando o
# método 'merge' da biblioteca pandas.
df['TOT_VIAG'] = pd.merge(df, df_max, how='left', left_on='ID_PESS',
                           right_index=True)['MAX_VIAG']

log_output.info('TOT_VIAG:\\n\\n' +
                '    Situação final dos dados: \\n' +
                str(df['TOT_VIAG'].describe()) + '\\n' +
                '    TOT_VIAG: Situação final dos dados: \\n' +
                str(df['TOT_VIAG'].value_counts()))

```

```

# Agora uma função que irá verificar se para todo "ID_PESS" o "TOT_VIAG"
# é igual ao 'NO_VIAG' máximo.
def verifica_no_viag_tot_viag(row):
    if row['NO_VIAG'] != row['TOT_VIAG']:
        log_output.warning('TOT_VIAG: Erro encontrado na linha '
                           + str(row) + ':\n'
                           + ' => ' + row
                           )
df.loc[:, ['ID_PESS', 'NO_VIAG', 'TOT_VIAG']]\
    .groupby('ID_PESS')\
    .agg({'NO_VIAG': 'max', 'ID_PESS': 'max', 'TOT_VIAG': 'max'})\
    .apply(verifica_no_viag_tot_viag, axis=1)
log_output.info('\n\n=====\\n')

return df

```

2.7 Funções que geram os “NO”s e os “ID”s

Função/Variável	Status
gera_nos_e_ids	Verificar NO VIAG

```
In [ ]: log_tela.info('Definindo funções que geram os "NO"s e os "ID"s')
log_output.info('\n\n=====\\n')
```

```
def gera_nos_e_ids(passo, df):
    """
    Esta função gera todos os IDs e todos os NOs das variáveis:
    DOM (domicílio), FAM (família), PESS (pessoa) e VIAG (viagem)
    A ordem de geração é:
    NO_DOM, ID_DOM, NO_FAM, ID_FAM, NO_PESS, ID_PESS, NO_VIAG, ID_VIAG
    Esta ordem é fixa pois cada uma dessas variáveis depende da geração
    da variável anterior.

    Os NOs são gerados como se fossem subíndices.
    No caso dos domicílios, eles são contabilizados dentro de cada zona.
    Assim, cada novo domicílio que aparece na listagem recebe um número
    que representa sua posição na sequência de domicílios dentro da zona
    na qual ele está contido.
    As famílias seguem o mesmo raciocínio, com relação ao domicílio, as
    pessoas com relação às famílias e as viagens com relação às pessoas.

    Deve-se apenas tomar cuidado com as viagens, pois existem pessoas que
    não realizaram viagens. Neste caso, estes registros devem contabilizar
    o NO_VIAG como zero, e não como 1. Para contemplar estes casos, vamos
    utilizar a variável F_VIAG, que representa quando há ou não viagem
    naquele registro (F_VIAG==1 tem viagem, F_VIAG==0 não tem viagem).
    Assim, o que precisamos fazer para calcular o NO_VIAG é agrupar os
    registros por pessoa (ID_PESS) e, dentro de cada agrupamento, somar
    o valor de F_VIAG registro a registro (linha por linha)
    cumulativamente. No final do processo, atribui-se zero a NO_VIAG
    quando F_VIAG for igual a zero.
    """

def gera_id_dom(row):
    """
    Gera o ID_DOM baseado no 'ANO', na 'ZONA_DOM' e no 'NO_DOM'
    O argumento passado é a "linha".
    Uso:
        df['ID_DOM'] = df.apply(gera_ID_DOM, axis=1)
    Retorna: ID_DOM da respectiva linha
    """
    # Formatando o ano para string
    ano = str(row['ANO'])

    # Formatando a zona para ficar como string com 3 caracteres
    zona = str('%03d' % row['ZONA_DOM'])
```

```

# Formatando no_dom para ficar como string com 4 caracteres
no_dom = str('%04d' % row['NO_DOM'])

# Retornando o número inteiro correspondente à string
# concatenada de ano + zona + no_dom
return ano + zona + no_dom


def gera_id_fam(row):
    """
    Gera o ID_FAM baseado no 'ID_DOM' e no 'NO_FAM'
    O argumento passado é a "linha".
    Uso:
        df['ID_FAM'] = df.apply(gera_ID_FAM, axis=1)
    Retorna: ID_FAM da respectiva linha
    """
    # Formatando id_dom como string
    id_dom = str(row['ID_DOM'])

    # Formatando no_fam como string para ficar com 2 caracteres
    no_fam = str('%02d' % row['NO_FAM'])

    # Retornando o número inteiro correspondente à string
    # concatenada de ID_DOM com NO_FAM
    return id_dom + no_fam


def gera_id_pess(row):
    """
    Gera o ID_PESS baseado no 'ID_FAM' e no 'NO_PESS'
    O argumento passado é a "linha".
    Uso:
        df['ID_PESS'] = df.apply(gera_ID_PESS, axis=1)
    Retorna: ID_PESS da respectiva linha
    """
    # Formatando id_fam como string
    id_fam = str(row['ID_FAM'])

    # Formatando no_pess como string para ficar com 2 caracteres
    no_pess = str('%02d' % row['NO_PESS'])

    # Retornando o número inteiro correspondente à string
    # concatenada de ID_FAM com NO_PESS
    return id_fam + no_pess


def gera_id_viag(row):
    """
    Gera o ID_VIAG baseado no 'ID_PESS' e no 'NO_VIAG'
    O argumento passado é a "linha".
    Uso:
        df['ID_VIAG'] = df.apply(gera_ID_VIAG, axis=1)
    Retorna ID_VIAG da respectiva linha
    """
    # Formatando id_pess como string
    id_pess = str(row['ID_PESS'])

```

```

# Formatando no_viag como string para ficar com 2 caracteres
no_viag = str('%02d' % row['NO_VIAG'])

# Retornando o número inteiro correspondente à string
# concatenada de ID_PESS com NO_VIAG
return id_pess + no_viag

#gera NO_DOM
log_tela.info("Gerando NO_DOM")
df['NO_DOM'] = df.groupby('ZONA_DOM', sort=False)['F_DOM'].cumsum()
log_tela.info("NO_DOM gerado")
log_tela.info("Gerando ID_DOM")
#gera ID_DOM
df['ID_DOM'] = df.apply(gera_id_dom, axis=1)
log_tela.info("ID_DOM gerado")

#gera NO_FAM
log_tela.info("Gerando NO_FAM")
df['NO_FAM'] = df.groupby('ID_DOM', sort=False)['F_FAM'].cumsum()
log_tela.info("NO_FAM gerado")
log_tela.info("Gerando ID_FAM")
#gera ID_FAM
df['ID_FAM'] = df.apply(gera_id_fam, axis=1)
log_tela.info("ID_FAM gerado")

#gera NO_PESS
log_tela.info("Gerando NO_PESS")
df['NO_PESS'] = df.groupby('ID_FAM', sort=False)['F_PESS'].cumsum()
log_tela.info("NO_PESS gerado")
log_tela.info("Gerando ID_PESS")
#gera ID_PESS
df['ID_PESS'] = df.apply(gera_id_pess, axis=1)
log_tela.info("ID_PESS gerado")

#gera NO_VIAG
log_tela.info("Gerando NO_VIAG")
#df['NO_VIAG'] = 1
df['NO_VIAG'] = df.groupby('ID_PESS', sort=False)['F_VIAG'].cumsum()
# Verificando se FE_VIAG == 0 e zerando NO_VIAG nesse caso
df.loc[df['FE_VIAG'] == 0, 'NO_VIAG'] = 0
log_tela.info("NO_VIAG gerado")
log_tela.info("Gerando ID_VIAG")
#gera ID_VIAG
df['ID_VIAG'] = df.apply(gera_id_viag, axis=1)
log_tela.info("ID_VIAG gerado")

return df

```

2.8 Função relativa à entrevista

Função/Variável	Status
passo_cd_entre	OK

Obs: o passo_cd_entre deve ser executado após o passo_tot_viag.

```
In [ ]: log_tela.info('Definindo função relativa à entrevista')
log_output.info('\n'=====\\n')

def passo_cd_entre(passo, df):
    """
    #####
    #      ATENÇÃO      #
    # ESTA FUNÇÃO SÓ DEVE SER #
    # EXECUTADA APÓS A GERAÇÃO#
    #      DO TOT_VIAG     #
    #####
    ----
    Substituir valores da coluna "CD_ENTRE"
    Todas viagens são consideradas "completas", segundo informações do Metrô

    * sem viagem: se TOT_VIAG == 0
    * com viagem: se TOT_VIAG != 0

    #####Categorias anteriores
    | Valor | Descrição |
    | ----- | ----- |
    | 5 | Completa sem viagem |
    | 6 | Completa com viagem |

    #####Categorias novas
    | Valor | Descrição |
    | ----- | ----- |
    | 0 | Completa sem viagem |
    | 1 | Completa com viagem |

    [Teste: Checar se existe algum número diferente de 0 ou 1.
     Se encontrar, retornar erro indicando em qual linha.]
:param passo: Número do passo atual para registro/log
:param df:
:return:
"""
log_tela.info("### PASSO " + str(passo) + " - CD_ENTRE")
log_tela.warning('\n'#####\\n' +
                 'Este passo DEVE ser executado apenas após a \\n' +
                 'execução do passo que geram o TOT_VIAG.\\n' +
                 '#####\\n' )

# Definindo 'CD_ENTRE' baseado no valor de 'TOT_VIAG'
df.loc[df['TOT_VIAG'] == 0, 'CD_ENTRE'] = 0
```

```
df.loc[df['TOT_VIAG'] != 0, 'CD_ENTRE'] = 1

verifica_dummy(df, 'CD_ENTRE')
log_output.info('\n\n=====\\n')

return df

In [ ]: def passo_correcoes(passo, df):
    log_tela.info("### PASSO " + str(passo) + " - Correções")
    log_tela.info("Nenhuma correção a ser efetuada.")

    return df
```

3 Definindo a função principal (main)

Esta função vai ler os arquivos (csv) necessários e, então, irá efetuar cada passo, chamando as respectivas funções que foram definidas acima. Ao final, esta função irá salvar o resultado das transformações realizadas num arquivo csv.

```
In [ ]: def main():
    start_time = time.time()

    log_tela.info("Horário de início da execução: %s" %
                  time.strftime("%H:%M", time.localtime(start_time)))

    # ----
    log_tela.info('Lendo arquivos CSV')

    log_output.info('Lendo CSV da base original da OD.')
    od = pd.read_csv('bases/OD_2007.csv', sep=';', decimal=',')

    deflator = 1
    log_output.info('Deflator a ser utilizado para correção monetária: ' +
                    str(deflator))

    # Filtrando dataframe para pegar apenas uma amostra
    #od = od[:1000].copy()

    log_output.info('\n\n=====\\n')
    log_tela.info('Pré-Processamento.')
    od = pre_processamento(od)

    log_tela.info('Imprimindo a lista de variáveis do dataframe da OD.')
    log_tela.debug(od.columns.tolist())

    #Contador de 'PASSO'
    passo = 1

    # ----
    # ##Passo: "ANO"
    od = passo_ano(passo, od)
    passo += 1

    # ----
    # ##Passo: "DIA_SEM"
    od = passo_dia_sem(passo, od)
    passo += 1

    # ----
    # ##Passo: "UCODs"
    od = passo_ucods(passo, od)
    passo += 1

    # ----
    # ##Passo: "ZONA_DOM"
    od = passo_zona_dom(passo, od)
    passo += 1
```

```

# -----
# ##Passo: "SUBZONA_DOM"
od = passo_subzona_dom(passo, od)
passo += 1

# -----
# ##Passo: "MUN_DOM"
od = passo_mun_dom(passo, od)
passo += 1

# -----
# ##Passo: "F_DOM"
od = passo_f_dom(passo, od)
passo += 1

# -----
# ##Passo: "FE_DOM"
od = passo_fe_dom(passo, od)
passo += 1

# -----
# ##Passo: "TIPO_DOM"
od = passo_tipo_dom(passo, od)
passo += 1

# -----
# ##Passo: "TOT_FAM"
od = passo_tot_fam(passo, od)
passo += 1

# -----
# ##Passo: "F_FAM"
od = passo_f_fam(passo, od)
passo += 1

# -----
# ##Passo: "FE_FAM"
od = passo_fe_fam(passo, od)
passo += 1

# -----
# ##Passo: "COND_MORA"
od = passo_cond_mora(passo, od)
passo += 1

# -----
# ##Passo: "QT_AUTO"
od = passo_qt_auto(passo, od)
passo += 1

# -----
# ##Passo: "QT_BICI"
od = passo_qt_bici(passo, od)
passo += 1

```

```

# -----
# ##Passo: "QT_MOTO"
od = passo_qt_moto(passo, od)
passo += 1

# -----
# ##PASSO: "REN_FAM"
od = passo_ren_fam(passo, od, deflator)
passo += 1

# -----
# ##Passo: "CD_RENFAM"
od = passo_cd_renfam(passo, od)
passo += 1

# -----
# ##Passo: "F_PESS"
od = passo_f_pess(passo, od)
passo += 1

# -----
# ##Passo: "FE_PESS"
od = passo_fe_pess(passo, od)
passo += 1

# -----
# ##Passo: "SIT_FAM"
od = passo_sit_fam(passo, od)
passo += 1

# -----
# ##Passo: "IDADE"
od = passo_idade(passo, od)
passo += 1

# -----
# ##Passo: "SEXO"
od = passo_sexo(passo, od)
passo += 1

# -----
# ##Passo: "GRAU_INSTR"
od = passo_grau_instr(passo, od)
passo += 1

# -----
# ##Passo: "OCUP"
od = passo_ocup(passo, od)
passo += 1

# -----
# ##Passo: "SETOR_ATIV"
od = passo_setor_ativ(passo, od)

```

```

    passo += 1

    # ##Passo: "REN_IND"
    od = passo_ren_ind(passo, od, deflator)
    passo += 1

    # ##Passo: "CD_RENIND"
    od = passo_cd_renind(passo, od)
    passo += 1

    # -----
    # ##Passo: "F_VIAG"
    od = passo_f_viag(passo, od)
    passo += 1

    # -----
    # ##"FE_VIAG"
    od = passo_fe_viag(passo, od)
    passo += 1

    # -----
    # ##Passo: "ZONA_ESC"
    od = passo_zona_esc(passo, od)
    passo += 1

    # -----
    # ##Passo: "SUBZONA_ESC"
    od = passo_subzona_esc(passo, od)
    passo += 1

    # -----
    # ##Passo: "MUN_ESC"
    od = passo_mun_esc(passo, od)
    passo += 1

    # -----
    # ##Passo: "ESTUDA"
    # Este passo deve vir após os passos de localização da escola
    od = passo_estuda(passo, od)
    passo += 1

    # -----
    # ##Passo: "ZONA_TRAB1"
    od = passo_zona_trab1(passo, od)
    passo += 1

    # -----
    # ##Passo: "SUBZONA_TRAB1"
    od = passo_subzona_trab1(passo, od)
    passo += 1

    # -----
    # ##Passo: "MUN_TRAB1"
    od = passo_mun_trab1(passo, od)

```

```

    passo += 1

    # -----
    # ##Passo: "ZONA_TRAB2"
    od = passo_zona_trab2(passo, od)
    passo += 1

    # -----
    # ##Passo: "SUBZONA_TRAB2"
    od = passo_subzona_trab2(passo, od)
    passo += 1

    # -----
    # ##Passo: "MUN_TRAB2"
    od = passo_mun_trab2(passo, od)
    passo += 1

    # -----
    # ##Passo: "ZONA_ORIG"
    od = passo_zona_orig(passo, od)
    passo += 1

    # -----
    # ##Passo: "SUBZONA_ORIG"
    od = passo_subzona_orig(passo, od)
    passo += 1

    # -----
    # ##Passo: "MUN_ORIG"
    od = passo_mun_orig(passo, od)
    passo += 1

    # -----
    # ##Passo: "ZONA_DEST"
    od = passo_zona_dest(passo, od)
    passo += 1

    # -----
    # ##Passo: "SUBZONA_DEST"
    od = passo_subzona_dest(passo, od)
    passo += 1

    # -----
    # ##Passo: "MUN_DEST"
    od = passo_mun_dest(passo, od)
    passo += 1

    # -----
    # ##Passo: "SERV_PAS_ORIG"
    od = passo_serv_pas_orig(passo, od)
    passo += 1

    # -----
    # ##Passo: "SERV_PAS_DEST"

```

```

od = passo_serv_pas_dest(passo, od)
passo += 1

# -----
# ##Passo: "MOTIVO_ORIG"
od = passo_motivo_orig(passo, od)
passo += 1

# -----
# ##Passo: "MOTIVO_DEST"
od = passo_motivo_dest(passo, od)
passo += 1

# -----
# ##Passo: "MOD01"
od = passo_modo1(passo, od)
passo += 1

# -----
# ##Passo: "MOD02"
od = passo_modo2(passo, od)
passo += 1

# -----
# ##Passo: "MOD03"
od = passo_modo3(passo, od)
passo += 1

# -----
# ##Passo: "MOD04"
od = passo_modo4(passo, od)
passo += 1

# -----
# ##Passo: "MODO_PRIN"
od = passo_modo_prin(passo, od)
passo += 1

# -----
# ##Passo: "TIPO_VIAG"
od = passo_tipo_vitag(passo, od)
passo += 1

# -----
# ##"H_SAIDA"; "MIN_SAIDA"; "ANDA_ORIG"; "H_CHEG"; "MIN_CHEG";
# "ANDA_DEST"; "DIST_VIAG" e "DURACAO"
# Nada há que se fazer em relação aos dados das colunas acima mencionadas

# -----
# ##Passo: "TIPO_EST_AUTO"
od = passo_tipo_est_auto(passo, od)
passo += 1

# -----

```

```

# ##Passo: "VALOR_EST_AUTO"
od = passo_valor_est_auto(passo, od, deflator)
passo += 1

# -----
# ##Passo: Coordenadas
od = coordenadas(passo, od)
passo += 1

# -----
# ##Passo: Gerando NO's e ID's
od = gera_nos_e_ids(passo, od)
passo += 1

# -----
# ##Passo: "TOT_VIAG"
od = passo_tot_viag(passo, od)
passo += 1

# -----
# ##Passo: "CD_ENTRE"
od = passo_cd_entre(passo, od)
passo += 1

# -----
# ##Passo: Correções
od = passo_correcoes(passo, od)
passo += 1

log_tela.info('Salvando dataframe como arquivo CSV')
# -----
# ## Salvando o dataframe num arquivo local
od.to_csv('outputs/2007_od.csv', sep=';', decimal=',', index=False)

log_tela.info("Base gerada. Arquivo: outputs/2007_od.csv")

log_tela.info("Tempo total de execução: %s segundos" %
             (time.time() - start_time))

log_tela.info("Horário de finalização: %s" %
             (time.strftime("%H:%M", time.localtime(time.time()))))

log_tela.info("Terminou o main")

```

4 RUN the main() function...

```
In [ ]: if __name__ == "__main__":
    main()
```