

EDA (Exploratory Data Analysis) - Modelo Gravitacional

Análise de dados exploratória e construção de um modelo inicial.

Lucas Coelho

27/09/2020

Sumário

1	Introdução	4
2	Análise de dados exploratória	4
2.1	Leitura dos dados	5
2.2	Tratamento dos dados	5
2.3	Gráficos	7
3	Construção do modelo gravitacional	8
4	Conclusão	9

Lista de Figuras

1 Introdução

Este relatório descreve a condução de uma análise exploratória dos dados da matriz OD aérea bem como a construção de um modelo gravitacional que retrata o fluxo de passageiros.

2 Análise de dados exploratória

O primeiro passo para a construção de um modelo é a condução de uma análise de dados exploratória a fim de se situar e conhecer melhor os dados. O objetivo aqui é visualizar as rotas aéreas dentro do estado de São Paulo bem como a construção de um modelo gravitacional para representar o fluxo de passageiros. Apenas as rotas do estado de São Paulo foram selecionadas visando a diminuição do custo computacional nesta fase de entendimento dos conceitos.

Boas referências:

- Dr Ds Idiots Guide to Spatial Interaction Modelling for Dummies - Part 1:
 - The Unconstrained (Total Constrained) Model by Adam Dennett available here https://rpubs.com/adam_dennett/257231
- Dr Ds Idiots Guide to Spatial Interaction Modelling for Dummies - Part 2:
 - Constrained Models by Adam Dennett available here https://rpubs.com/adam_dennett/259068

O primeiro passo é a importação das bibliotecas a serem utilizadas:

```
library(geobr)
library(arrow)
library(ggplot2)
library(dplyr)
library(sf)
library(stplanr)
library(tidyr)
```

2.1 Leitura dos dados

Em seguida, podemos fazer a importação dos dados. Os *data frames* de **matriz_od_aerea** e **idades** foram previamente processada e salva em um arquivo .parquet para possibilitar uma leitura mais eficiente. O dataframe **idades_geobr** consiste nos dados extraídos da biblioteca geobr, e contém os dados de mapeamento dos municípios.

```
matriz_od_aerea <- read_parquet("./bd_tele_processed/matriz_od_aerea.gz.parquet") # m
idades <- read_parquet("./bd_tele_processed/idades.gz.parquet") # idades processa
idades_geobr <- readRDS("./extra_data/idades_geobr.rds") # Dados das idades Brasi
```

2.2 Tratamento dos dados

Primeiramente, os dados são filtrados a fim de exibir apenas as rotas interiores ao estado de São Paulo.

```
# Filtrando os dados para apenas o estado de SP
matriz_od_aerea <- matriz_od_aerea %>% filter(UFUTPorigemunicipio=='SP',UFUTPdestinou
idades <- idades %>% filter(uf=="SP")
idades_geobr <- idades_geobr %>% filter(abbrev_state=="SP")
```

Em seguida, os dados são consolidados. O código de cidades utilizado nos dados de entrada é convertido no código de cidades do IBGE e presente no *data frame* de geometrias.

```
# Unificando os dataframes de cidades, visando adicionar a geometria para o mapa
idades$codigo <- as.numeric(as.character(idades$codigo))
idades_geobr$code_muni <- as.numeric(as.character(idades_geobr$code_muni))
idades_joined <-left_join(idades, idades_geobr, by = c("codigo" = "code_muni"))

matriz_od_aerea$code_origem <- idades_joined$codigo[match(matriz_od_aerea$idmunicipi
matriz_od_aerea$code_destino <- idades_joined$codigo[match(matriz_od_aerea$idmunicipi
```

Posteriormente, os dados são agrupados para pares únicos de origem-destino e quantidade total de viagens é obtida.

```
# calculo das viagens totais por origem-destino
matriz_od_aerea_summarized <- matriz_od_aerea %>%
  group_by(code_origem,code_destino) %>%
  summarize(
    viagensTotais = sum(quantidadeviagem)
  ) %>%
  drop_na() #drop NA a ser corrigido. Por algum motivo, algumas viagens de MG estavam
```

Para fins de simplificação e redução de custo computacional, apenas os trechos com mais de 150 viajantes são selecionados:

```
# Filtrando para apenas as maiores viagens (mais de 150)
matriz_od_aerea_summarized <- matriz_od_aerea_summarized %>%
  filter(viagensTotais >= 150)
```

A malha de viagens é criada com a função `od2line`. Ela retorna um *data frame* que possui uma coluna com a geometria da rota, para o mapa.

```
# Criação da rede de viagens com a função od2line
travel_network <- od2line(flow = matriz_od_aerea_summarized, zones = cidades_geobr)

## Creating centroids representing desire line start and end points.

head(travel_network)

## Simple feature collection with 6 features and 3 fields
## geometry type:  LINESTRING
## dimension:      XY
## bbox:           xmin: -51.05679 ymin: -23.65003 xmax: -46.64815 ymax: -20.79729
## geographic CRS: SIRGAS 2000
##   code_origem code_destino viagensTotais geometry
```

## 1	3500105	3550308	245 LINESTRING (-51.05679 -21.5...
## 2	3501608	3549805	160 LINESTRING (-47.28912 -22.7...
## 3	3502804	3509502	5133 LINESTRING (-50.57457 -21.1...
## 4	3502804	3549805	206 LINESTRING (-50.57457 -21.1...
## 5	3502804	3550308	3504 LINESTRING (-50.57457 -21.1...
## 6	3502804	3552403	237 LINESTRING (-50.57457 -21.1...

2.3 Gráficos

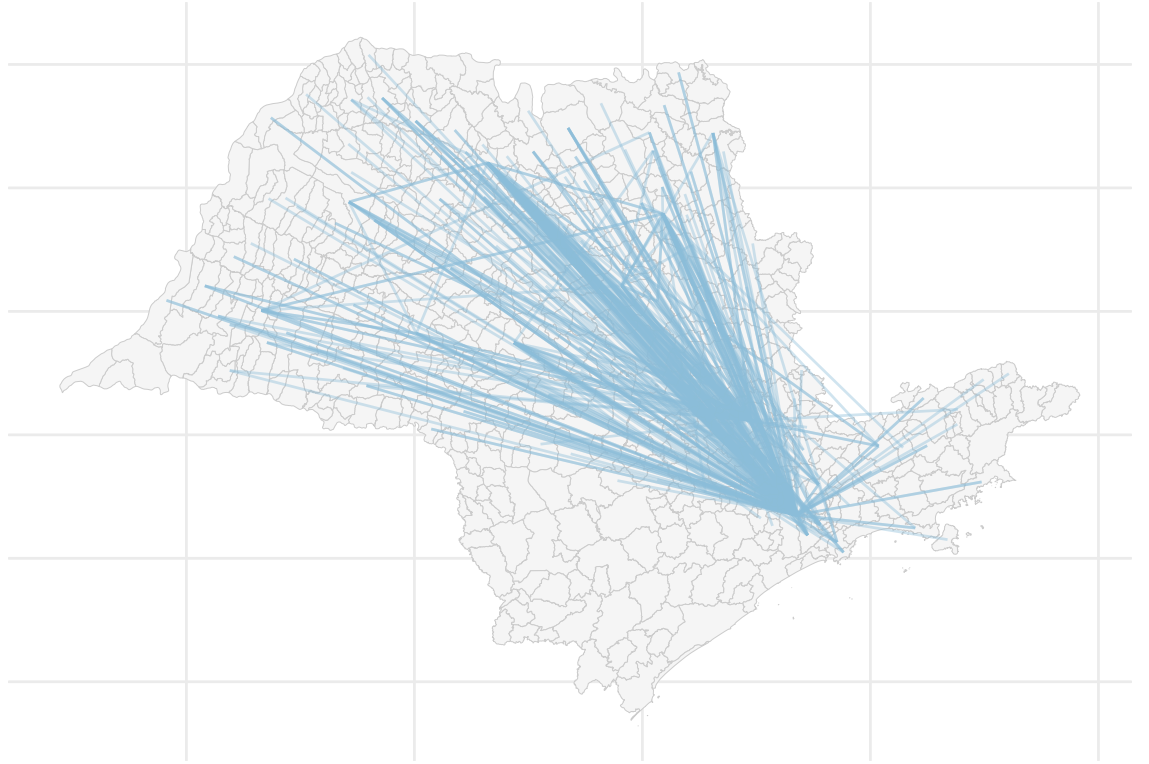
Por fim, as viagens aéreas dentro do estado de São Paulo são traçadas no mapa do estado. A função `read_municipality` da biblioteca **geobr** é utilizada para obter os limites do mapa de São Paulo.

```
all_muni_sp <- read_municipality(code_muni = "SP", year= 2019)
```

```
no_axis <- theme(axis.title=element_blank(),
                 axis.text=element_blank(),
                 axis.ticks=element_blank())

ggplot() +
  geom_sf(data=all_muni_sp, fill="#F5F5F5", color="#CCCCCC", size=.15, show.legend =
  geom_sf(data = travel_network$geometry, color=alpha("#8BBD9", 0.4)) +
  labs(subtitle="Viagens no estado de São Paulo", size=8) +
  theme_minimal() +
  no_axis
```

Viagens no estado de São Paulo



3 Construção do modelo gravitacional

O passo seguinte é a construção de um modelo gravitacional que estime as viagens entre duas localidades.

A equação clássica do modelo gravitacional é dada por:

$$T_{ij} = kV_i^\mu W_j^\alpha d_{ij}^{-\beta}$$

onde: T_{ij} é o fluxo entre a origem e o destino; V_i é a “emissibilidade” da origem - uma medida que indica o quanto as pessoas de lá saem. O que leva a ir; W_j é a “atratividade” do destino - uma medida que indica o quanto as pessoas chegam. O que leva a vir; d_{ij} é a resistência ao fluxo - como a distância; k é a constante de proporcionalidade; μ , α e β são os parâmetros do modelo.

Cenário de exemplo: viagens entre os bairros de uma cidade. - Exemplo de V_i :

população. Pode-se argumentar que, quanto mais gente em um bairro, maior o fluxo de saída. - Exemplo de W_j : postos de trabalho. Pode-se argumentar que, quanto mais postos de trabalho em um bairro, mais pessoas fariam o traslado diariamente em direção ao bairro.

4 Conclusão

Após a elaboração do estudo, incluir o texto com as conclusões e aprendizados iniciais.