Argentina flight data analysis

mmuzzi

Data import and sanitization

- Data comes from Argentinian *Ministerio de Transporte*, *Empresa Argentina de Navegacion Civil*

```
wd <- "C:\\Users\\mmuzzi\\Desktop\\eana"
setwd(wd)

df_eana <- read.csv('eana1401-1802.csv',
                    stringsAsFactors=F, header=T, sep=';')
colnames(df_eana) <- c('date', 'time', 'flight_class', 'flight_type', 'arr_dep',
                       'orig', 'dest', 'carrier', 'plane', 'apc_code')

df_eana$date <- as.Date(df_eana$date, "%d/%m/%Y")
df_eana <- df_eana %>% filter(date >= "2017-01-01" & date < "2018-01-01")
```

Airports data

- Data from OurAirports (direct link to .csv file!)

```
df_airports <- read.csv('airports.csv',
                        stringsAsFactors=F,
                        header=T, sep=',')

df_airports$iata_code[df_airports$iata_code == 0 |
                        df_airports$iata_code == ''] = NA
df_airports$local_code[df_airports$local_code == 0 |
                         df_airports$local_code == ''] = NA
```

Analysis and manipulation

```
# dest

# Commercial flight from non-null carriers
df_eana <- df_eana %>%
  filter (!is.na(carrier) &
            (flight_class == 'Regular' | flight_class == 'No Regular')) %>%
  filter(flight_type == 'Cabotaje')

# Converting gps_code airport in EANA dataset to iata_code
# (not all airports have an IATA code)
# orig
df_eana$orig_iata <- df_airports$iata_code[match(df_eana$orig,
                                                 df_airports$gps_code)]
# df_eana$orig_local <- df_airports$local_code[match(df_eana$orig,
#                                                    df_airports$gps_code)]
```

```
# dest
df_eana$dest_iata <- df_airports$iata_code[match(df_eana$dest,
                                                 df_airports$gps_code)]
# df_eana$dest_local <- df_airports$local_code[match(df_eana$dest,
#                                                    df_airports$gps_code)]

# replacing blanks with NA
# "^$|^ $" either nothing "" or " "
df_eana$orig_iata <- sapply(df_eana$orig_iata,
                            function (x) gsub("^$|^ $", NA, x))
df_eana$dest_iata <- sapply(df_eana$dest_iata,
                            function (x) gsub("^$|^ $", NA, x))

df_eana_filter <-  df_eana %>%
  filter(!is.na(orig_iata) & !is.na(dest_iata))

head(df_eana_filter)
```

```
##         date  time flight_class flight_type    arr_dep orig dest
## 1 2017-01-01 00:08      Regular    Cabotaje   Despegue SACO SABE
## 2 2017-01-01 00:08      Regular    Cabotaje Aterrizaje SAWE SAEZ
## 3 2017-01-01 00:15      Regular    Cabotaje Aterrizaje SAWC SABE
## 4 2017-01-01 00:15      Regular    Cabotaje Aterrizaje SAWC SABE
## 5 2017-01-01 00:19      Regular    Cabotaje   Despegue SABE SAZN
## 6 2017-01-01 00:29      Regular    Cabotaje Aterrizaje SARI SABE
##                  carrier         plane apc_code orig_iata dest_iata
## 1 Aerolíneas Argentinas  BOEING B-737        D       COR       AEP
## 2 Aerolíneas Argentinas  BOEING B-737        C       RGA       EZE
## 3 Aerolíneas Argentinas  BOEING B-737        C       FTE       AEP
## 4 Aerolíneas Argentinas  BOEING B-737        D       FTE       AEP
## 5 Austral Líneas Aéreas EMBRAER E-190        C       AEP       NQN
## 6 Austral Líneas Aéreas EMBRAER E-190        C       IGR       AEP
```

```
# we can safely ignore those flights from airports with no IATA code: 0.7%
# > sum(df_eana_filter$percent[is.na(df_eana_filter$orig_iata) |
#                              is.na(df_eana_filter$dest_iata)])
# [1] 0.007217057

# Get airport pairs
df_pairs <- df_eana_filter[,c("orig_iata", "dest_iata")]
df_pairs$pair <- NA

# Paste them alphabetically
df_pairs$pair <- mapply(function (x, y) {
  pair <- c(x, y)
  return(paste(sort(pair), collapse='-'))},
  df_pairs$orig_iata, df_pairs$dest_iata)

# Get Pareto distribution
pair_count <- df_pairs %>%
  select(orig_iata, dest_iata, pair) %>%
  group_by(pair) %>%
  summarize(n=n()) %>%
  arrange(desc(n)) %>%
```

```
  ungroup() %>%
  mutate(cumsum = cumsum(n)/sum(n),
         percent = n/sum(n))
```

Distance analysis

```
haversine <- function (lat_from, lon_from, lat_to, lon_to, r=6371) {
 radians <- pi/180
 lat_to <- lat_to * radians
 lat_from <- lat_from * radians
 lon_to <- lon_to * radians
 lon_from <- lon_from * radians
 dLat <- (lat_to - lat_from)
 dLon <- (lon_to - lon_from)
 a <- (sin(dLat/2)^2) + (cos(lat_from) * cos(lat_to)) * (sin(dLon/2)^2)
 return(2 * atan2(sqrt(a), sqrt(1 - a)) * r)
}

# Airport pairs
pair_count <- pair_count %>% separate(pair, c("from", "to"), "-", remove=F)

# Get coordinates
pair_count$lat_from <- df_airports$latitude_deg[match(pair_count$from,
                                                      df_airports$iata_code)]
pair_count$lon_from <- df_airports$longitude_deg[match(pair_count$from,
                                                       df_airports$iata_code)]

pair_count$lat_to <- df_airports$latitude_deg[match(pair_count$to,
                                                    df_airports$iata_code)]
pair_count$lon_to <- df_airports$longitude_deg[match(pair_count$to,
                                                     df_airports$iata_code)]

# Calculate distance
pair_count$dist <- haversine(pair_count$lat_from, pair_count$lon_from,
                             pair_count$lat_to, pair_count$lon_to)

# Weighted average distance
dist_weighted <- pair_count %>%
  summarize(sum(percent*dist))

dist_weighted
```

```
## # A tibble: 1 x 1
##   `sum(percent * dist)`
##                   <dbl>
## 1                 1012.
```

```
write.csv(pair_count, file='pair_count.csv')
```

Further analysis

```
airports <- pair_count %>%
  select(from, n) %>%
  group_by(from) %>%
  summarize(n_group=sum(n)) %>%
  mutate(percent=n_group/sum(n_group)) %>%
  arrange(desc(n_group))

df_provinces <- read.csv('provinces.csv', stringsAsFactors=F,
                         sep=',', header=T, encoding='UTF-8')
colnames(df_provinces) <- c("code", "name", "type", "province")

airports$iso_region <- df_airports$iso_region[match(airports$from,
                                                    df_airports$iata_code)]

airports$province <- df_provinces$province[match(airports$iso_region,
                                                 df_provinces$code)]

airports$long <- df_airports$longitude_deg[match(airports$from,
                                                 df_airports$iata_code)]
airports$lat <- df_airports$latitude_deg[match(airports$from,
                                               df_airports$iata_code)]

airports %>% group_by(province) %>%
  summarize(n_province=sum(n_group)) %>%
  mutate(percent_province=n_province/sum(n_province),
         cumsum_province=cumsum(n_province)/sum(n_province)) %>%
  arrange(desc(n_province))
```

```
## # A tibble: 23 x 4
##    province              n_province percent_province cumsum_province
##    <chr>                      <int>           <dbl>           <dbl>
##  1 Buenos Aires Province     211935          0.829           0.829
##  2 Córdoba                   17219          0.0674          0.920
##  3 Río Negro                  6673          0.0261          0.980
##  4 Chubut                     5214          0.0204          0.853
##  5 Mendoza                    4412          0.0173          0.943
##  6 Santa Cruz                 3613          0.0141          0.995
##  7 Misiones                   1753          0.00686         0.950
##  8 Jujuy                      1177          0.00460         0.926
##  9 Neuquén                    1129          0.00442         0.954
## 10 Santa Fe                    769          0.00301         0.998
## # ... with 13 more rows
```

Plotting results

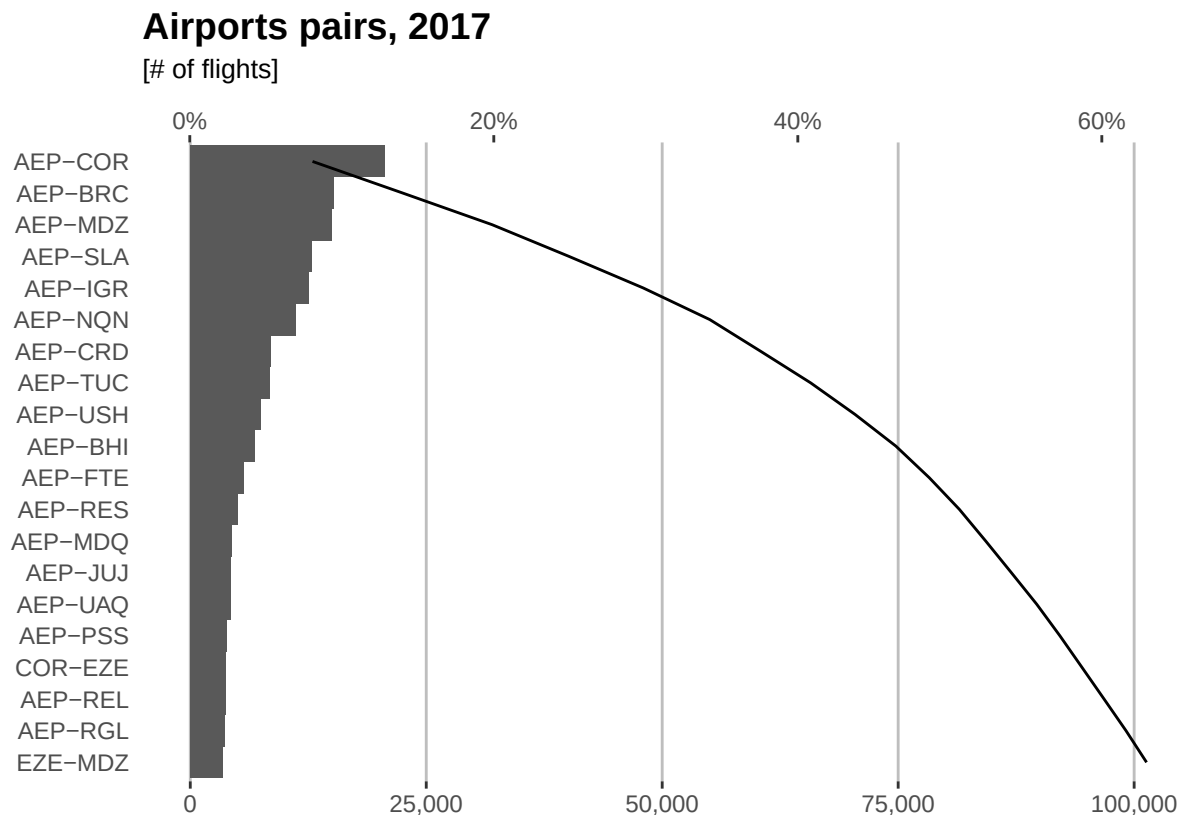
```
# Pareto
pair_count_plot <- pair_count %>% top_n(20, n)
pair_count_plot$pair <- factor(pair_count_plot$pair, levels=rev(pair_count_plot$pair))
N <- sum(pair_count_plot$n)

theme_custom <- theme(plot.title=element_text(hjust=0,
                                              margin=margin(b=0),
                                              size=14, face="bold"),
  plot.subtitle=element_text(hjust=0, size=10,
                             margin=margin(t=5, b=10)),
  panel.grid.minor=element_blank(),
  panel.grid.major=element_line(color='gray', size=.5),
  panel.grid.major.y=element_blank(),
  panel.background=element_blank(),
  axis.ticks.y=element_blank())

pareto <- ggplot(pair_count_plot) +
  geom_bar(aes(x=pair, y=n), width=1, stat='identity', colors='#44546a') +
  geom_line(aes(x=pair, y=cumsum*N, group=1)) +
  scale_x_discrete(breaks=pair_count$pair) +
  scale_y_continuous(labels=scales::comma,
                     sec.axis=sec_axis(~./N, labels=scales::percent)) +
  labs(x="", y="", title="Airports pairs, 2017", subtitle="[# of flights]") +
  theme_custom +
  coord_flip()
```

```
## Warning: Ignoring unknown parameters: colors
```

```
plot(pareto)
```



Maps

```
# Reading shapefiles
shp_country <- readOGR('./shp/country/pais.shp',
                       encoding='UTF-8', use_iconv=T)
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\mmuzzi\Desktop\eana\shp\country\pais.shp", layer: "pais"
## with 1 features
## It has 5 fields
```

```
shp_province <- readOGR('./shp/province/provincia.shp',
                       encoding='UTF-8', use_iconv=T)
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\mmuzzi\Desktop\eana\shp\province\provincia.shp", layer: "provincia"
## with 24 features
## It has 5 fields
```

```
# Transforming them in dataframes
df_country <- fortify(shp_country)
```

```
## Regions defined for each Polygons
```

```
df_province <- fortify(shp_province)
```

```
## Regions defined for each Polygons
# Filtering Antartica and stuff like that
new_df <- df_province %>% filter(lat > -56 & long < -50)

df_citypair <- pair_count %>% top_n(80, n)

# Base map with outline of provinces
map_base <- ggplot() +
  geom_polygon(data=new_df, aes(x=long, y=lat, group=group),
               fill=NA, color='grey') +
  coord_map() + theme_custom

# Transparency values
df_citypair <- df_citypair %>%
  mutate(alpha=n/max(n))

map_country <- map_base +
  geom_point(data=df_citypair, aes(x=lon_from, y=lat_from),
             size=1) +
  geom_segment(data=df_citypair, aes(x=lon_from, y=lat_from,
                                     xend=lon_to, yend=lat_to,
                                     color=from, alpha=alpha),
               size=2) +
  scale_alpha_continuous(guide='none') +
  guides(color=guide_legend(title="Airports")) +
  labs(title='Airports in Argentina',
       subtitle='Most active airports pairs by traffic') +
  theme(panel.grid.major=element_blank(),
        axis.title=element_blank(),
        axis.text=element_blank(),
        axis.ticks=element_blank(),
        legend.key=element_blank())

plot(map_country)
```

Airports in Argentina

Most active airports pairs by traffic