BSDS 100 Case Study

Jacques Sham & Charles Siu

April 03, 2018

Part 1: Executive Summary

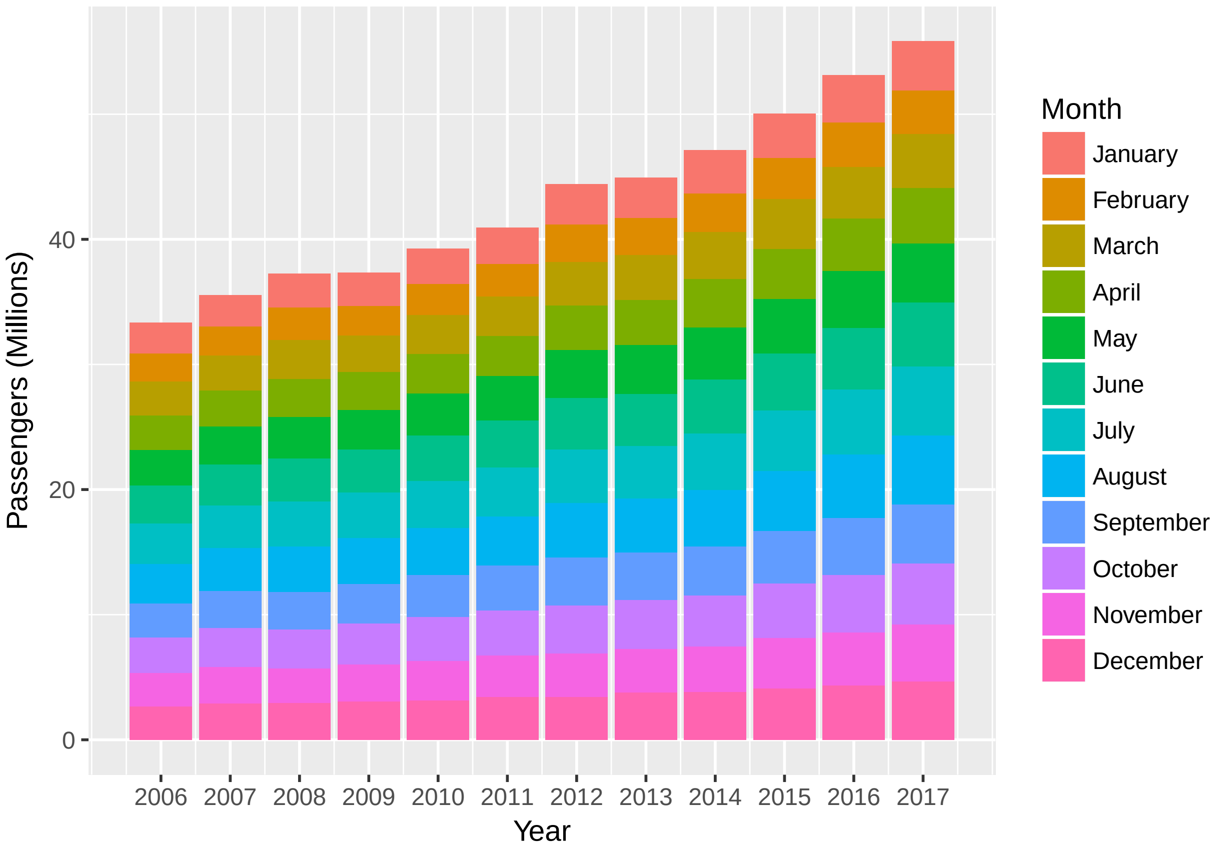
!!DOTO: No more than 250 words explaining, with extreme concision, your data and findings!!

Part 2: Introduction to the Data

First, let's clean the data.

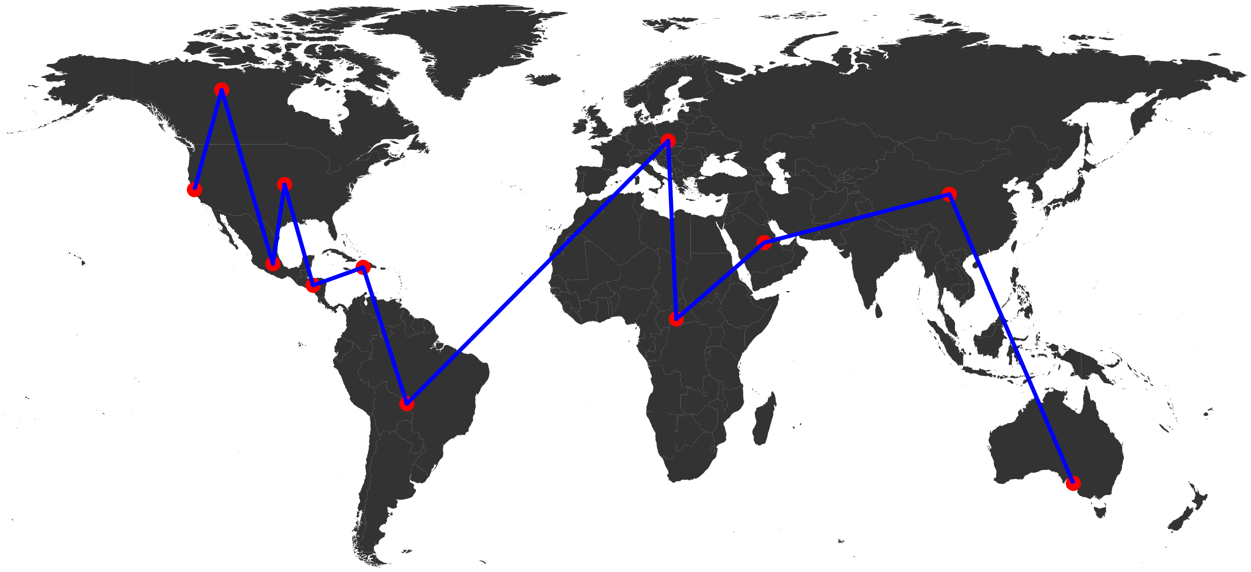
```
# Rename the column names
names(data) <- c("date", "operAirline", "operCode", "airline", "code", "isDomestic",
                 "region", "type", "category", "terminal", "area", "pax")

# Convert isDomestic to boolean
data$isDomestic %<>% recode("Domestic" = T, "International" = F)

# Reformat the dates into Date objects
data$date %<>%
  as.character() %>%
  as.yearmon("%Y%m") %>%
  as.Date()

# Get month and year values
data %<>%
  mutate(month = date %>% format("%m") %>% factor(labels = month.name),
         year = date %>% format("%Y") %>% as.numeric())

# Remove data from 2007 for easy comparison
data %<>% filter(year != 2005)

## operAirline = Operating Airline, the airline that really operating, some airline would own a child c
## operCode = Operating Airline Code
## airline = Published Airline, mother company airline
## code = Published Airline Code, code for variable "airline"
## isDomestic: If the flight is domestic,T; if international: F
## region: Geom region
## type: activity type; Deplaned means arrival, Enplaned means departure, Thru / Transit is other
## category: Airline price type; Low fare is Low cost carrier, else are others
## terminal: SFO terminal
## area: area within SFO terminal
## pax: passenger count of given row
```

```
data %>%
  ggplot() +
  geom_bar(aes(x = year, y = pax / 1000000, fill = month), stat = "identity") +
  scale_x_continuous(name = "Year", breaks = seq(2006, 2017, by = 1)) +
  scale_y_continuous(name = "Passengers (Millions)") +
  scale_fill_discrete(name = "Month")
```



```
world <- map_data("world")
cities <- world.cities

remove_axes <- theme(
  axis.text = element_blank(),
  axis.line = element_blank(),
  axis.ticks = element_blank(),
  panel.border = element_blank(),
  panel.grid = element_blank(),
  axis.title = element_blank()
  )

# Derive the cities and corresponding regions
cities %<>%
  filter(
    (name == "San Francisco" & country.etc == "USA") |
    (name == "Adelaide" & country.etc == "Australia") |
    (name %in% c("Topeka", "Yellowknife", "Tegucigalpa", "Port-au-Prince",
                   "Pontes e Lacerda", "Bambari", "Riyadh", "Mexico City",
                   "Xining", "Ostrava"))
    ) %>%
  mutate(region =
          ifelse(name == "San Francisco", "SF",
          ifelse(name == "Adelaide", "Australia / Oceania",
          ifelse(name == "Topeka", "US",
```

```r
           ifelse(name == "Yellowknife", "Canada",
           ifelse(name == "Tegucigalpa", "Central America",
           ifelse(name == "Port-au-Prince", "Caribbean",
           ifelse(name == "Pontes e Lacerda", "South America",
           ifelse(name == "Bambari", "Africa",
           ifelse(name == "Riyadh", "Middle East",
           ifelse(name == "Mexico City", "Mexico",
           ifelse(name == "Xining", "Asia",
           ifelse(name == "Ostrava", "Europe", NA))))))))))))
        )

myCurve <- curveGrob(0, 0, 1, 1, default.units = "npc",
               curvature = 0.2, angle = 90, ncp = 20, shape = 1,
               square = FALSE, squareShape = 1,
               inflect = FALSE, arrow = arrow(), open = TRUE,
               debug = FALSE, name = NULL, gp = gpar(), vp = NULL)

world %>%
  filter(region != "Antarctica") %>%
  ggplot() +
  geom_polygon(aes(x = long, y = lat, group = group)) +
  geom_point(data = cities, aes(x = long, y = lat), color = "red", size = 3) +
  coord_fixed(1.2) +
  theme_minimal() +
  geom_line(data = cities, aes(x = long, y = lat), color = "blue", size = 1) +
  remove_axes
```



```r
# annotation_custom(grob = myCurve, 0, 100, 0, 100)
```