Zillow Model Update: Boulder County, CO

Jasmine Siyu Wu & Zoe Yoo

10/19/2021

Contents

1. Introduction

[motivation & objectives]

```
knitr::opts_chunk$set(echo = TRUE,fig.width = 10, fig.height = 5)

#Loading Libraries
library(tidyverse)
library(sf)
library(spdep)
library(caret)
library(ckanr)
library(riem)
library(lubridate)
library(FNN)
library(grid)
library(gridExtra)
library(ggcorrplot)
library(kableExtra)
library(jtools)     # for regression model plots
library(ggstance)
library(osmdata)
library(knitr)
library(tidycensus)
library(scales)
library(stargazer)
library(ggplot2)
library(ggpubr)
library(xtable)


options(scipen=999)
options(tigris_class = "sf")
options(tigris_use_cache = TRUE)

# functions and data directory
root.dir = "https://github.com/zoenyoo/MUSA508_Final.git"
```

```r
source("https://raw.githubusercontent.com/urbanSpatial/Public-Policy-Analytics-Landing/master/functions


#Loading Styling Options
mapTheme <- function(base_size = 12) {
  theme(
    text = element_text( color = "black"),
    plot.title = element_text(size = 16,colour = "black"),
    plot.subtitle=element_text(face="italic"),
    plot.caption=element_text(hjust=0),
    axis.ticks = element_blank(),
    panel.background = element_blank(),axis.title = element_blank(),
    axis.text = element_blank(),
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_rect(colour = "black", fill=NA, size=2),
    strip.text.x = element_text(size = 14))
}

plotTheme <- function(base_size = 12) {
  theme(
    text = element_text( color = "black"),
    plot.title = element_text(size = 16,colour = "black"),
    plot.subtitle = element_text(face="italic"),
    plot.caption = element_text(hjust=0),
    axis.ticks = element_blank(),
    panel.background = element_blank(),
    panel.grid.major = element_line("grey80", size = 0.1),
    panel.grid.minor = element_blank(),
    panel.border = element_rect(colour = "black", fill=NA, size=2),
    strip.background = element_rect(fill = "grey80", color = "white"),
    strip.text = element_text(size=12),
    axis.title = element_text(size=12),
    axis.text = element_text(size=10),
    plot.background = element_blank(),
    legend.background = element_blank(),
    legend.title = element_text(colour = "black", face = "italic"),
    legend.text = element_text(colour = "black", face = "italic"),
    strip.text.x = element_text(size = 14)
  )
}

#Loading Quantile Break Functions
qBr <- function(df, variable, rnd) {
 if (missing(rnd)) {
    as.character(quantile(round(df[[variable]],0),
                 c(.01,.2,.4,.6,.8), na.rm=T))
 } else if (rnd == FALSE | rnd == F) {
    as.character(formatC(quantile(df[[variable]],
                 c(.01,.2,.4,.6,.8), na.rm=T), digits = 3))
 }
}
```

	Delay_minutes	Dalay_hours
mean	4.186744	0.06977907
median	2.283333	0.03805556
min	0	0
max	406	6.766667

```
q5 <- function(variable) {as.factor(ntile(variable, 5))}

#Loading Hexadecimal Color Palette

palette5 <- c("#324376", "#586ba4", "#f5dd90", "#ee964b", "#f95738")
palette4 <- c("#324376", "#586ba4", "#ee964b", "#f95738")
palette2 <- c("#324376", "#f95738")
```

2. Data Wrangling

2.1. Import Rail Delay Data

```
# January for training and 2-3 weeks in February for testing
rail_2020_01 <- read.csv(unz('Data/2020_01.csv.zip','2020_01.csv'), header = T)
rail_2020_02 <- read.csv(unz('Data/2020_02.csv.zip','2020_02.csv'), header = T)

rail <- rbind(rail_2020_01, rail_2020_02) %>%
  mutate(schedule60 = floor_date(ymd_hms(scheduled_time), unit = "hour"),
         actual60 = floor_date(ymd_hms(actual_time), unit = "hour"),
         week = week(schedule60),
         dotw = wday(schedule60, label=TRUE),
         year = year(schedule60),
         month = month(schedule60)) %>%
  drop_na(delay_minutes)
  #filter(week %in% c(14:18))


summary_statistics <-
  cbind(" " = list( "mean", "median", "min", "max"),
        "Delay_minutes" = list( "mean" = mean(rail$delay_minutes),
                                "median" = median(rail$delay_minutes),
                                "min" = min(rail$delay_minutes),
                                "max" = max(rail$delay_minutes)),
        "Dalay_hours" = list("mean" = mean(rail$delay_minutes/60),
                             "median" = median(rail$delay_minutes/60),
                             "min" = min(rail$delay_minutes/60),
                             "max" = max(rail$delay_minutes/60))) %>%
  as_data_frame()

summary_statistics %>%
  as_data_frame() %>%
  kable() %>%
  kable_styling()
```

```
ggplot(rail)+
  geom_histogram(aes(delay_minutes/60), binwidth = 0.5, fill = palette2[2], alpha=0.8)+
  #xlim(0, 100) +
  labs(title="NJ Transit and Amtrak rail delayed hours",
       subtitle = "New Jersey, Jan. - Feb., 2020",
       x="Hours",
       y="Frequency",
       caption="Figure 2.1")+
  plotTheme()
```

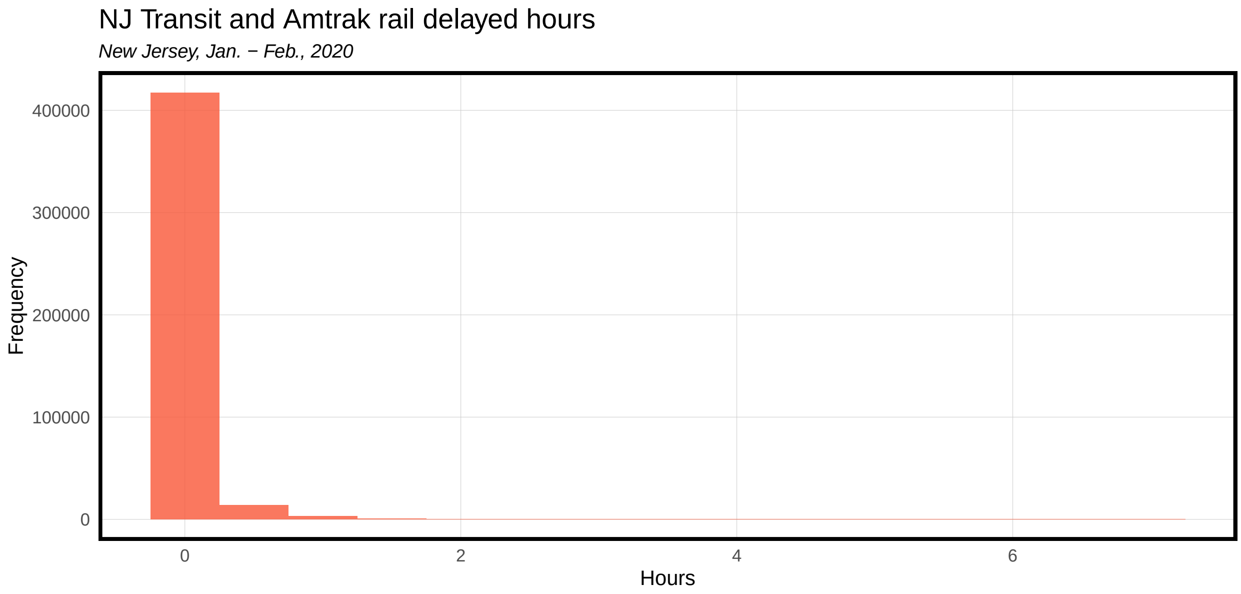

Figure 2.1

2.2. Import Weather Data

```
#https://mesonet.agron.iastate.edu/request/download.phtml?network=CA_ASOS
# EWR station is at Newark International Airport
weather.Data <-
  riem_measures(station = "EWR", date_start = "2020-01-01", date_end = "2020-03-01")


weather.Panel <-
  weather.Data %>%
    mutate_if(is.character, list(~replace(as.character(.), is.na(.), "0"))) %>%
    replace(is.na(.), 0) %>%
    mutate(interval60 = ymd_h(substr(valid, 1, 13))) %>%
    mutate(week = week(interval60),
           dotw = wday(interval60, label=TRUE)) %>%
    group_by(interval60) %>%
    summarize(Temperature = max(tmpf),
              Percipitation = sum(p01i),
              Wind_Speed = max(sknt)) %>%
    mutate(Temperature = ifelse(Temperature == 0, 42, Temperature))
```

```
grid.arrange(bottom="Figure 2.2 Weather Data, New Jersey, January - February, 2020",
  ggplot(weather.Panel, aes(interval60, Percipitation)) +
    geom_line(color = palette2[2]) +
    labs(title="Percipitation", x="Hour", y="Percipitation") +
    plotTheme(),
  ggplot(weather.Panel, aes(interval60, Wind_Speed)) +
    geom_line(color = palette2[2]) +
    labs(title="Wind Speed", x="Hour", y="Wind Speed") +
    plotTheme(),
  ggplot(weather.Panel, aes(interval60, Temperature)) +
    geom_line(color = palette2[2]) +
    labs(title="Temperature", x="Hour", y="Temperature") +
    plotTheme())
```

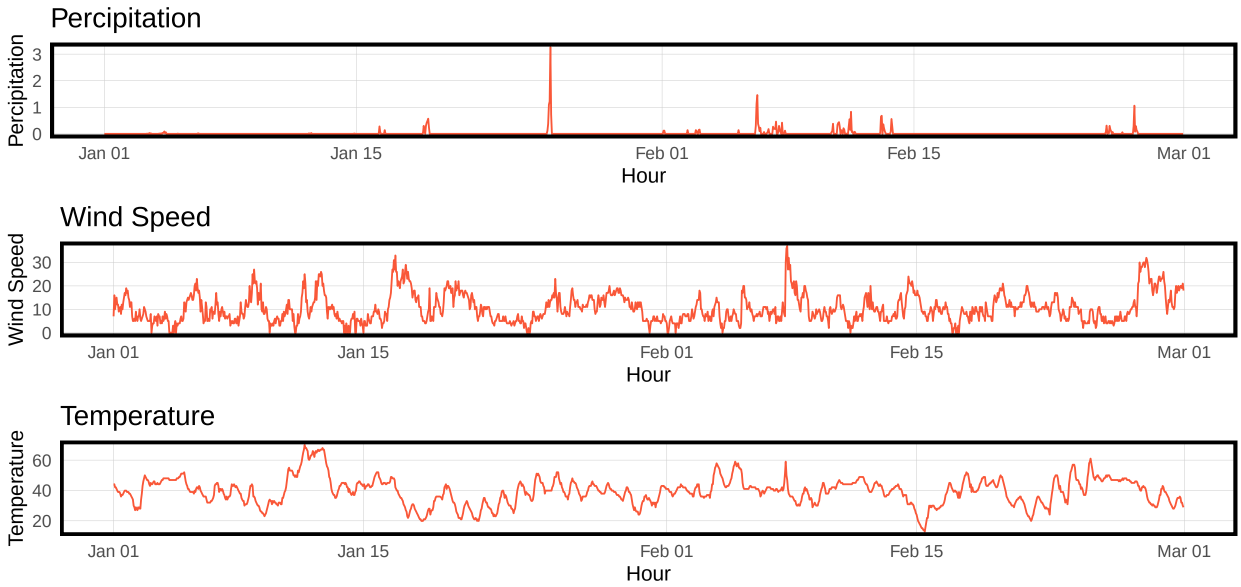

Figure 2.2 Weather Data, New Jersey, January – February, 2020

2.3. Import Station and Place Data

```
station.sf <- st_read("https://opendata.arcgis.com/datasets/4809dada94c542e0beff00600ee930f6_0.geojson"
  st_transform("EPSG:3424") %>% #NAD83 / New Jersey (ftUS)
  rename(STATION_NAME = STATION_ID)

station_list <- read.csv('Data/StationName_ID.csv', header = T)
station.sf <- left_join(station.sf, station_list, by=c("STATION_NAME" = "STATION_NAME"))


# mainly New Jersey yet a few in New York State and Penn
counties <- rbind(get_acs(geography = "county",
                        year = 2019,
                        state = 34,
                        variables = (TotalPop = 'B01001_001E'),
                        survey = "acs5",
                        output = "wide",
                        geometry = TRUE),
                get_acs(geography = "county",
                        year = 2019,
                        state = 42,
                        variables = (TotalPop = 'B01001_001E'),
                        survey = "acs5",
                        output = "wide",
                        geometry = TRUE),
                get_acs(geography = "county",
                        year = 2019,
                        state = 36,
                        variables = (TotalPop = 'B01001_001E'),
                        survey = "acs5",
                        output = "wide",
                        geometry = TRUE)) %>%
  st_transform(st_crs(station.sf))

intersect.counties <-  subset(counties, GEOID %in%
                              (st_intersection(counties, station.sf) %>%
                                 dplyr::select(GEOID) %>%
                                 st_drop_geometry() %>%
                                 unique())$GEOID)

rm(counties)

states <- rbind(get_acs(geography = "state",
                        year = 2019,
                        state = 34,
                        variables = (TotalPop = 'B01001_001E'),
                        survey = "acs5",
                        output = "wide",
                        geometry = TRUE),
                get_acs(geography = "state",
                        year = 2019,
                        state = 42,
                        variables = (TotalPop = 'B01001_001E'),
                        survey = "acs5",
                        output = "wide",
                        geometry = TRUE),
                get_acs(geography = "state",
                        year = 2019,
                        state = 36,
                        variables = (TotalPop = 'B01001_001E'),
                        survey = "acs5",
                        output = "wide",
                        geometry = TRUE)) %>%
  st_transform(st_crs(station.sf))
```

```
ggplot() +
  geom_sf(data=intersect.counties, color='grey', fill=NA) +
  geom_sf(data=station.sf, color=palette2[1], alpha=0.8, size=1) +
  labs(title="New Jersey Transit and Amtrak Stations",
       caption = "Figure 2.1") +
  mapTheme()
```

New Jersey Transit and Amtrak Stations

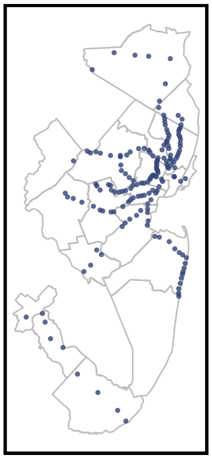

Figure 2.1