STAT-231 Final Project: Technical Report

Martin Glusker, Nicole Frontero, Oliver Baldwin Edwards

Due: 12/18/18

Abstract:

This project seeks to better understand the relationship between GDP per capita and transit in metropolitan areas in the United States. While the original intention was to create a quality of life score or statistic utilizing multiple variables to see how a city's transit may predict that score, doing so is beyond the scope of this project (see introduction for an elaboration of the motivation behind altering project scope). We decided to pivot instead to explore how well transit statistics can predict GDP per capita in U.S. metropolitan areas. We created 31 regression tree models, and our best model suggests that the percentage of workers commuting by public transport and the amount of vehicle hours per capita (percent_commuting_msa and per_capita_vrh, respectively) are best at predicting GDP per capita (based on the variables and U.S. areas present in our data set).

Introduction:

The research question at hand is if the quality of a U.S. metropolitan area's transit system can predict the GDP per capita of that area. This question is interesting because GDP is a valued metric in terms of a city's economic well-being. Looking for a relationship between GDP and a city's transit system thus has the potential to provide insight on how much a transit system matters in terms of GDP.

It's important to note that the question stated above was not the initial question that we sought to address. We initially wanted to analyze the relationship between a city's transit system and the quality of life in that city. However, this question presented various difficulties. For example, creating a model that predicted different combinations of explanatory transit variables based on multiple quality of life response variables would prove to be beyond the scope of our capabilities. An alternative option for combining multiple quality of life variables for a given city in a model would be to utilize only one quality of life response variable. However, this would necessitate a reframing of the question at hand, as choosing one variable to represent the quality of life at large would be misleading (and any variables that might serve as a reasonable proxy—such as Human Development Index—). To resolve these problems, we decided to alter the research question to ask the question, "How does a city's transit system predict its GDP per capita?" We decided upon GDP per capita as a response variable as it is well-regarded as a broad metric of the health of a city's economy and avoids the problem of the population of a city skewing the GDP.

In order to answer this question, one large data frame was created that combined quality of life (including GDP per capita) and transit system information by metropolitan statistical areas. This data frame can be accessed via the R package, "gluskr" created during the course of this project. Each variable in the data frame is explained in the documentation for the gluskr data frame (which is named `transit_qol_df`).

Upon creating multiple visualizations and models, the best model suggests that the percentage of workers commuting by public transport and the number of vehicle hours per capita (percent_commuting_msa and per_capita_vrh, respectively) together are the best way to predict the GDP per capita (based on the variables and U.S. areas present in our data set).

Data:

The data included in the final data frame, transit_qol_df, comes from the U.S. Census Bureau, the Bureau of Economic Analysis, and the Federal Transit Administration's National Transit Database. This final data frame has 42 variables and has data from 2007-2017. For specifics on each variable, consult the transit_qol_df documentation within the gluskr package or the readme for the project.

We first compiled our quality of life data frame by combining all relevant quality of life statistics for each metropolitan statistical area. Once this was complete, we did the same with all of our transit data. We then joined the quality of life and transit data frames by a geographical ID to create our aggregate data frame. Lastly, we mutated this aggregate data frame to create 13 new transit related metrics based on existing variables in our data frame, such as the cost per trip (which was calculated by doing the total amount of transit expenses divided by the number of passenger trips).

It's important to note that we only used metropolitan statistical areas in the United States with a population of above one million in our modeling. This is an attempt to limit the error created by the differences between metropolitan statistical areas and urbanized areas. Our census data is grouped by metropolitan statistical area, but our transit data is by urbanized area. The relationship file between urbanized areas and metropolitan statistical area connects the two when they overlap in any way, so for example the urbanized area of "New York-Newark, NY-NJ-CT" relates to both the metropolitan area of "New York-Northern New Jersey-Long Island, NY-NJ-PA," but also the metro area of "Trenton-Ewing, NJ." This leads to problem when using transit variables with census variables. In the example above, the metric of trips per capita will be heavily skewed because the population of Trenton is significantly smaller than the population served by the transit system of the "New York-Newark, NY-NJ-CT" urbanized area leading to a skewed and inaccurate statistic. Filtering out cities below one million reduces this source of error because it reduces the number of metro areas that overlap or are in close proximity.

Quality of Life Data

Read in GDP Data

This creates a data frame with gdp for each metropolitan and micropolitan statistical area from the U.S. census from years 2001-2017.

```
gdp_df <- read_xls(
  "../final-data/data-sources/quality-of-life-data/gdp/gdp-industrytotal.xls",
                    sheet = 1, range = 'A6:S390', col_names=TRUE) %>%
  gather(key = "Year", value = "gdp", -c("GeoFips", "GeoName")) %>%
  mutate("GEO.id2" = as.integer(GeoFips), "Year" = as.double(Year)) %>%
  select(GEO.id2, GeoName, Year, gdp)
```

Read in Latitude Longitude Data

This creates a data frame with latitudinal and longitudinal coordinates for each metropolitan and micropolitan statistical area.

```
# coordinates for metropolitan areas
metropolitan_coordinates <- read_xlsx(
  "../final-data/data-sources/quality-of-life-data/latitude-longitude/StatisticalAreasLatLong.xlsx",
  sheet = 1, range = "A2:Q376", col_names = TRUE)

# coordinates for micropolitan areas
micropolitan_coordinates <- read_xlsx(
  "../final-data/data-sources/quality-of-life-data/latitude-longitude/StatisticalAreasLatLong.xlsx",
```

```
  sheet = 2, range = "A2:Q583", col_names = TRUE)

# combine metropolitan and micropolitan areas
lat_long_df <- rbind(metropolitan_coordinates, micropolitan_coordinates) %>%
  mutate("GEO.id2" = GEOID, "GEO.display-label" = NAME) %>%
  arrange(NAME) %>%
  select(GEO.id2, `GEO.display-label`, CENTLAT, CENTLON, INTPTLAT, INTPTLON)
```

Read in Population Data

This code chunk reads in the population estimates from the U.S. Census Bureau for each year between 2007 and 2017 and joins each year's data into one data frame.

```
# function to read in population DFs efficiently
read_population_df <- function(file_name, year) {

  df <- read_csv(file_name)

  colnames(df)[4] <- "PopEstimate"
  colnames(df)[5] <- "MOE"
  df <- df %>%
    mutate("Year" = year) %>%
    select(-MOE)

  return(df)
}

# read in population data for years 2007 through 2017
pop_2007 <- suppressMessages(read_population_df(
  '../final-data/data-sources/quality-of-life-data/population/2007/ACS_07_1YR_B01003.csv',
  2007))
pop_2008 <- suppressMessages(read_population_df(
  '../final-data/data-sources/quality-of-life-data/population/2008/ACS_08_1YR_B01003.csv',
  2008))
pop_2009 <- suppressMessages(read_population_df(
  '../final-data/data-sources/quality-of-life-data/population/2009/ACS_09_1YR_B01003.csv',
  2009))
pop_2010 <- suppressMessages(read_population_df(
  '../final-data/data-sources/quality-of-life-data/population/2010/ACS_10_1YR_B01003.csv',
  2010))
pop_2011 <- suppressMessages(read_population_df(
  '../final-data/data-sources/quality-of-life-data/population/2011/ACS_11_1YR_B01003.csv',
  2011))
pop_2012 <- suppressMessages(read_population_df(
  '../final-data/data-sources/quality-of-life-data/population/2012/ACS_12_1YR_B01003.csv',
  2012))
pop_2013 <- suppressMessages(read_population_df(
  '../final-data/data-sources/quality-of-life-data/population/2013/ACS_13_1YR_B01003.csv',
  2013))
pop_2014 <- suppressMessages(read_population_df(
  '../final-data/data-sources/quality-of-life-data/population/2014/ACS_14_1YR_B01003.csv',
  2014))
pop_2015 <- suppressMessages(read_population_df(
```

```
  '../final-data/data-sources/quality-of-life-data/population/2015/ACS_15_1YR_B01003.csv',
  2015))
pop_2016 <- suppressMessages(read_population_df(
  '../final-data/data-sources/quality-of-life-data/population/2016/ACS_16_1YR_B01003.csv',
  2016))
pop_2017 <- suppressMessages(read_population_df(
  '../final-data/data-sources/quality-of-life-data/population/2017/ACS_17_1YR_B01003.csv',
  2017))


# join all years into one dataframe
suppressMessages(
population_df <- full_join(pop_2007, pop_2008) %>%
  full_join(pop_2009) %>%
  full_join(pop_2010) %>%
  full_join(pop_2011) %>%
  full_join(pop_2012) %>%
  full_join(pop_2013) %>%
  full_join(pop_2014) %>%
  full_join(pop_2015) %>%
  full_join(pop_2016) %>%
  full_join(pop_2017))
```

Read in Census Quality of Life Data

This data frame consists of quality of life related variables obtained from the U.S. Census Bureau from 2007-2017. The Census Bureau changed the names and descriptions of their variables after 2009 which thus meant two different functions were needed to accurately read in the desired variables. There were also variables introduced in 2010 that we wanted to include in our data frame (`percent_no_insurance_msa` and `percent_below_poverty_level`). Because they didn't exist until 2010 they are recorded as NAs from 2007-2009.

```
# function to clean up data frames from 2007-2009
# (the census changed variable names after 2009)
clean_data_2007_2009 <- function(file_name, year) {
  df <- read_csv(file_name) %>%
    mutate('Median household income (dollars)' = HC01_EST_VC69,
           'Percent workers commuting by public transportation' = HC02_EST_VC23,
           'Percent population unemployed' = HC02_EST_VC06,
           'Year' = year) %>%
     select(Year, GEO.id, GEO.id2, `GEO.display-label`,
           `Median household income (dollars)`,
           `Percent workers commuting by public transportation`,
           `Percent population unemployed`)
  return(df)
}

# function to clean up data frames from 2010-2017
clean_data_2010_2017 <- function(file_name, year) {
  df <- read_csv(file_name) %>%
    mutate('Median household income (dollars)' = HC01_VC85,
           'Percent workers commuting by public transportation' = HC03_VC31,
           'Percent population unemployed' = HC03_VC08,
```

```
            'Percent no health insurance coverage' = HC03_VC134,

             # for all people 18 years and older:
            'Percent income below poverty level' = as.numeric(HC03_VC171),
            'Year' = year ) %>%
     select(Year, GEO.id, GEO.id2, `GEO.display-label`,
            `Median household income (dollars)`,
            `Percent workers commuting by public transportation`,
            `Percent population unemployed`,
            `Percent no health insurance coverage`,
            `Percent income below poverty level`)

  return(df)
}

# read in census data for  years 2007 through 2017
census_data_2007 <- suppressMessages(clean_data_2007_2009(
  "../final-data/data-sources/quality-of-life-data/census-data/2007/ACS_07_1YR_DP3.csv",
  2007))
census_data_2008 <- suppressMessages(clean_data_2007_2009(
  "../final-data/data-sources/quality-of-life-data/census-data/2008/ACS_08_1YR_DP3.csv",
  2008))
census_data_2009 <- suppressMessages(clean_data_2007_2009(
  "../final-data/data-sources/quality-of-life-data/census-data/2009/ACS_09_1YR_DP3.csv",
  2009))
census_data_2010 <- suppressMessages(clean_data_2010_2017(
  "../final-data/data-sources/quality-of-life-data/census-data/2010/ACS_10_1YR_DP03.csv",
  2010))
census_data_2011 <- suppressMessages(clean_data_2010_2017(
  "../final-data/data-sources/quality-of-life-data/census-data/2011/ACS_11_1YR_DP03.csv",
  2011))
census_data_2012 <- suppressMessages(clean_data_2010_2017(
  "../final-data/data-sources/quality-of-life-data/census-data/2012/ACS_12_1YR_DP03.csv",
  2012))
census_data_2013 <- suppressMessages(clean_data_2010_2017(
  "../final-data/data-sources/quality-of-life-data/census-data/2013/ACS_13_1YR_DP03.csv",
  2013))
census_data_2014 <- suppressMessages(clean_data_2010_2017(
  "../final-data/data-sources/quality-of-life-data/census-data/2014/ACS_14_1YR_DP03.csv",
  2014))
census_data_2015 <- suppressMessages(clean_data_2010_2017(
  "../final-data/data-sources/quality-of-life-data/census-data/2015/ACS_15_1YR_DP03.csv",
  2015))
census_data_2016 <- suppressMessages(clean_data_2010_2017(
  "../final-data/data-sources/quality-of-life-data/census-data/2016/ACS_16_1YR_DP03.csv",
  2016))
census_data_2017 <- suppressMessages(clean_data_2010_2017(
  "../final-data/data-sources/quality-of-life-data/census-data/2017/ACS_17_1YR_DP03.csv",
  2017))

# join all years into one dataframe
census_qol_df <- suppressMessages(full_join(census_data_2007, census_data_2008) %>%
  full_join(census_data_2009) %>%
```

```
  full_join(census_data_2010) %>%
  full_join(census_data_2011) %>%
  full_join(census_data_2012) %>%
  full_join(census_data_2013) %>%
  full_join(census_data_2014) %>%
  full_join(census_data_2015) %>%
  full_join(census_data_2016) %>%
  full_join(census_data_2017) )
```

Join All Quality of Life Dataframes

This joins all of the quality of life related data into one data frame.

```
# join all quality of life dataframes
qol_df <- left_join(population_df, census_qol_df,
  by = c("Year", "GEO.id", "GEO.id2", "GEO.display-label")) %>%
  left_join(gdp_df, by = c("GEO.id2", "Year")) %>%
  left_join(lat_long_df, by = c("GEO.id2", "GEO.display-label")) %>%
  select(-c(GEO.id, GeoName))
```

Transit Data

Functions to Clean up FTA Dataframes

These are three functions used to wrangle the transit data from the FTA. `clean_modes()` takes in a data frame with variables `Modes` and groups categories them into a new variable called `modes_clean`. `clean_fta_df()` is a function to clean FTA data frames. `clean_funding_df()` is a function to clean the FTA's transit agency funding data.

```
#function to clean the modes of transit
clean_modes <- function(df) {
  rail_abbr <- c('HR', 'LR', 'SR', 'CR', 'IP', 'MG', 'YR', 'AR', 'CC')
  bus_abbr <- c('MB', 'TB', 'CB', 'RB', 'PB', 'JT')
  other_abbr <- c('VP', 'DR', 'DT', 'TR', 'FB', 'OT', 'OR')

  mutate(df, modes_clean =
      case_when(
        Modes %in% rail_abbr ~ "Rail",
        Modes %in% bus_abbr ~ "Bus",
        Modes %in% other_abbr ~ "Other",
        TRUE ~ "OTHER"))  %>%
      filter(modes_clean != "OTHER")

}

# function to read in and clean up FTA dataframes
clean_fta_df <- function(sheet_number, col_name) {

  df <- read_xlsx(
    "../final-data/data-sources/FTA-data/FTA-dataframes/TS2.1TimeSeriesOpExpSvcModeTOS_2.xlsx",
                  sheet=sheet_number, range="A1:AR5659", col_names=TRUE)
```

```
  df <- df %>%
    filter(`Reporter Type` == "Full Reporter") %>%
     gather(key = "Year",
          value = Name_of_Var,
           -`Last Report Year`,
           -`NTD ID`,
           -`Legacy NTD ID`,
           -`Agency Name`,
           -`Agency Status`,
           -`Reporter Type`,
           -City,
           -State,
           -`Census Year`,
           -`UZA Name`,
           -UZA,
           -`UZA Area SQ Miles`,
           -`UZA Population`,
           -`2017 Status`,
          -Mode,
          -Service,
          -`Mode Status`)  %>%
    mutate(Modes = Mode) %>%
    clean_modes() %>%
      select(-Mode) %>%
    filter(Year >= 2007) %>%
    filter(UZA !=0)

    df[,col_name] <- df[,'Name_of_Var']
    df[,'Name_of_Var'] <- NULL

    return(df)
}

# function to clean up FTA funding dataframes
clean_funding_df <- function(sheet_number, col_name) {
  df <- read_xlsx(
    "../final-data/data-sources/FTA-data/FTA-dataframes/TS1.1TimeSeriesOpCapFundingSummary_4 (2).xlsx",
                  sheet=sheet_number, range="A1:AO2943", col_names=TRUE)

  df <- df %>%
    filter(`Reporter Type` == "Full Reporter") %>%
      select(-`Reporter Type`) %>%
      gather(key = "Year",
           value = Name_of_Var,
           -`Last Report Year`,
           -`NTD ID`,
            -`Legacy NTD ID`,
            -`Agency Name`,
           -`Agency Status`,
           -City,
           -State,
           -`Census Year`,
           -`Primary UZA Name`,
```

```
            -UZA,
            -`UZA Area SQ Miles`,
            -`UZA Population`,
            -`2017 Status`)

    df[,col_name] <- df[,'Name_of_Var']
    df[,'Name_of_Var'] <- NULL

    return(df)
}
```

Main Transit Variables

Creates and tidies data frame for each transit variable (UPT, VRM, VRH, DRM, PMT, expenses, and fares).

```
expenses_fta_df <- clean_fta_df(3, "total_expenses") %>%
  group_by(UZA, Year, modes_clean) %>%
  mutate(total_expenses = sum(total_expenses, na.rm = TRUE)) %>%
  select(UZA, `UZA Name`, Year, modes_clean, total_expenses) %>%
  unique()

fares_fta_df <- clean_fta_df(8, "total_fares") %>%
  group_by(UZA, Year, modes_clean) %>%
  mutate(total_fares = sum(total_fares, na.rm = TRUE)) %>%
  select(UZA, `UZA Name`, Year, modes_clean, total_fares) %>%
  unique()

drm_fta_df <- clean_fta_df(9, "drm") %>%
  group_by(UZA, Year, modes_clean) %>%
  mutate(drm = sum(drm, na.rm = TRUE)) %>%
  select(UZA, `UZA Name`, Year, modes_clean, drm) %>%
  unique()

vrm_fta_df <- clean_fta_df(11, "vrm") %>%
  group_by(UZA, Year, modes_clean) %>%
  mutate(vrm = sum(vrm, na.rm = TRUE)) %>%
  select(UZA, `UZA Name`, Year, modes_clean, vrm) %>%
  unique()

vrh_fta_df <- clean_fta_df(12, "vrh") %>%
  group_by(UZA, Year, modes_clean) %>%
  mutate(vrh = sum(vrh, na.rm = TRUE)) %>%
  select(UZA, `UZA Name`, Year, modes_clean, vrh) %>%
  unique()

upt_fta_df <- clean_fta_df(13, "upt") %>%
  group_by(UZA, Year, modes_clean) %>%
  mutate(upt = sum(upt, na.rm = TRUE)) %>%
  select(UZA, `UZA Name`, Year, modes_clean, upt) %>%
  unique()

pmt_fta_df <- clean_fta_df(14, "pmt") %>%
  group_by(UZA, Year, modes_clean) %>%
```

```
  mutate(pmt = sum(pmt, na.rm = TRUE)) %>%
  select(UZA, `UZA Name`, Year, modes_clean, pmt) %>%
  unique()
```

Join FTA Variable Dataframes

Joins above transit data frames into one data frame.

```
# create file with master info
master_fta_df <- clean_fta_df(3, "total_expenses") %>%
  select(UZA, `UZA Area SQ Miles`, `UZA Population`)

# join all FTA Variable dataframes
primary_variable_fta_df <-
  inner_join(master_fta_df, expenses_fta_df, by= c("UZA")) %>%
  inner_join(fares_fta_df,    by= c("UZA", "UZA Name", "Year", "modes_clean")) %>%
  inner_join(drm_fta_df,      by= c("UZA", "UZA Name", "Year", "modes_clean")) %>%
  inner_join(vrh_fta_df,      by= c("UZA", "UZA Name", "Year", "modes_clean")) %>%
  inner_join(vrm_fta_df,      by= c("UZA", "UZA Name", "Year", "modes_clean")) %>%
  inner_join(pmt_fta_df,      by= c("UZA", "UZA Name", "Year", "modes_clean")) %>%
  inner_join(upt_fta_df,      by= c("UZA", "UZA Name", "Year", "modes_clean")) %>%
  unique()
```

Read in Transit Stations Dataframe

This chunk reads in the transit stations data frame from the FTA and cleans it. It then joins the master data frame to the transit stations data frame.

```
# Read in Master FTA Dataframe to use
# as joining key from transit agency to UZA
transit_stations_master_fta_df <- read_xlsx(
  "../final-data/data-sources/FTA-data/FTA-dataframes/FTA_September_2018.xlsx",
  sheet=2, range="A1:Y2129", col_names=TRUE) %>%
  filter(`Reporter Type` == "Full Reporter") %>%
  select(UZA, `5 digit NTD ID`)

# read in transit stations dataframe
transit_stations_df <- read_xlsx(
  "../final-data/data-sources/FTA-data/FTA-dataframes/Transit Stations_0.xlsx",
  sheet=1, range="A1:L1100", col_names=TRUE) %>%
  filter(`Reporter Type` == "Full Reporter") %>%
  mutate(Modes = Mode) %>%
  clean_modes() %>%
  select(-Mode) %>%
  select(`NTD ID`, modes_clean, `Total Stations`)


# Creates final transit stations df by IDing with UZA ids
# and joining those ids onto the stations data via Agnecy ID
# NOTE: this potentially double counts transit stations
# that are used by multiple transit networks
final_transit_stations_df <- left_join(transit_stations_master_fta_df,
  transit_stations_df, by = c("5 digit NTD ID" = "NTD ID" )) %>%
```

```
  group_by(UZA, modes_clean) %>%
  mutate(Total_Stations_2017 = sum(`Total Stations`, na.rm = TRUE)) %>%
  select(UZA, modes_clean, Total_Stations_2017) %>%
  filter(!is.na(modes_clean)) %>%
  unique()
```

Read in Transit Funding Data

Creates and tidies data frame for each transit funding variable (total funding, state funding, local funding, other funding).

```
# read in Total Funding data frame
Total_Funding_df <- clean_funding_df(3, "Total_Funding") %>%
  group_by(UZA, Year) %>%
  mutate(Total_Funding = sum(Total_Funding, na.rm = TRUE)) %>%
  select(UZA, Year, Total_Funding) %>%
  unique()

# read in Federal Funding data frame
Federal_Funding_df <- clean_funding_df(4, "Federal_Funding") %>%
  group_by(UZA, Year) %>%
  mutate(Federal_Funding = sum(Federal_Funding, na.rm = TRUE)) %>%
  select(UZA, Year, Federal_Funding) %>%
  unique()

# read in State Funding data frame
State_Funding_df <- clean_funding_df(5, "State_Funding") %>%
  group_by(UZA, Year) %>%
  mutate(State_Funding = sum(State_Funding, na.rm = TRUE)) %>%
  select(UZA, Year, State_Funding) %>%
  unique()

# read in Local Funding data frame
Local_Funding_df <- clean_funding_df(6, "Local_Funding") %>%
  group_by(UZA, Year) %>%
  mutate(Local_Funding = sum(Local_Funding, na.rm = TRUE)) %>%
  select(UZA, Year, Local_Funding) %>%
  unique()

# read in Other Funding data frame (aka Fares and other assorted revenue)
Other_Funding_df <- clean_funding_df(7, "Other_Funding") %>%
  group_by(UZA, Year) %>%
  mutate(Other_Funding = sum(Other_Funding, na.rm = TRUE)) %>%
  select(UZA, Year, Other_Funding) %>%
  unique()
```

Join FTA Funding Data frames

Joins the five above funding data frames into one comprehensive funding data frame.

```
# join all Funding dataframes
fta_funding_df <- full_join(Total_Funding_df,
                            Federal_Funding_df,
```

```
                              by = c("UZA", "Year")) %>%
  full_join(State_Funding_df, by = c("UZA", "Year")) %>%
  full_join(Local_Funding_df, by = c("UZA", "Year")) %>%
  full_join(Other_Funding_df, by = c("UZA", "Year"))
```

Add Identifiers to FTA Data frame Through Relationship File and Join FTA data frames

This chunk reads in a file that contains the relationship between urbanized areas (used in our transit data) and metropolitan statistical areas (used in our quality of life data). It then tidies our final transit data frame.

```
fta_df <- left_join(primary_variable_fta_df,
  final_transit_stations_df, by = c("UZA", "modes_clean"))%>%
  left_join(fta_funding_df, by=c("UZA", "Year")) %>%
  unique()

# get census key dataframe
suppressMessages(
census_UZA_MSA_key <- read_csv(
  "../final-data/data-sources/census-FTA-joining-key/UZA_MSA_join.txt") %>%
  mutate("GEO.id2" = CBSA) %>%
  arrange(desc(UAPOP)) %>%
  select(GEO.id2, UA, UANAME, UAPOP))

# join FTA dataframes here
final_fta_df <- left_join(census_UZA_MSA_key, fta_df,
                    by = c("UAPOP" = "UZA Population")) %>%
  mutate(Year = as.numeric(Year))%>%
  filter(Year > 2006, Year < 2018) %>%
  filter(!is.na(`UZA Name`)) %>%
  filter(UANAME != "Not in a 2010 urban area" ) %>%
  filter(UAPOP>100000) %>%
  select(-UANAME, -UZA)
```

Create Final Data Frame

Join Quality of Life and Transit Data

This chunk joins our quality of life and transit data frames by geographic ID and year. It also renames each of the columns for clarity.

```
# join final quality of life and fta datasets
# and filter out non-urban areas and metro areas below 1 million population
final_df <- left_join(qol_df, final_fta_df, by=c("GEO.id2", "Year"))%>%
  filter(UA != 99999) %>%
  mutate(gdp = gdp*1000000) %>%
  select(-CENTLON, -CENTLAT, -UAPOP) %>%
  filter(PopEstimate>=1000000)

# figure out which columns we want and rename columns accordingly
# clarify variables names
colnames(final_df)[1:29] <- c("msa_id",
                              "msa_name",
```

```
                              "pop_estimate_msa",
                              "year",
                              "median_household_income_msa",
                              "percent_commuting_msa",
                              "percent_unemployed_msa",
                              "percent_no_insurance_msa",
                              "percent_below_poverty_level",
                              "gdp_msa",
                              "intptlat",
                              "intptlon",
                              "ua_census_id",
                              "ua_sq_miles_2010",
                              "ua_fta_name",
                              "transit_modes",
                              "total_transit_expenses",
                              "total_fares",
                              "directional_route_miles",
                              "vehicle_hours",
                              "vehicle_miles",
                              "passenger_miles",
                              "passenger_trips",
                              "total_stations_2017",
                              "total_funding",
                              "federal_funding",
                              "state_funding",
                              "local_funding",
                              "other_funding")
```

Adding Per Capita Variables

This chunk adds new aggregate transit variables based on existing transit variables and adds them to our final data frame.

```
#Per Capita Transit Variables
final_df <- final_df %>%
  mutate(per_capita_gdp = gdp_msa / pop_estimate_msa,
         pmt_per_vrm = passenger_miles / vehicle_miles,
         pmt_per_vrh = passenger_miles / vehicle_hours,
         upt_per_vrh = passenger_trips / vehicle_hours,
         per_capita_vrm = vehicle_miles / pop_estimate_msa,
         per_capita_vrh = vehicle_hours / pop_estimate_msa,
         per_capita_pmt = passenger_miles / pop_estimate_msa,
         per_capita_upt = passenger_trips / pop_estimate_msa,
         recovery_ratio = total_fares / total_transit_expenses,
         fares_per_upt = total_fares / passenger_trips,
         cost_per_hour = total_transit_expenses / vehicle_hours,
         cost_per_trip = total_transit_expenses / passenger_trips,
         cost_per_pmt = total_transit_expenses / passenger_miles)
```

Save Final Data Frame

This converts our final data frame into a CSV as well as saving it as an RDS. This set of data is what the `transit_qol_df` in `gluskr` uses. Lines are commented out so as to not create duplicate data frames.

```
# write.csv(final_df, "FINAL_DATA.csv", row.names=F)

# saveRDS(final_df, file = "final_data.RDA")
```

Visualizations and Modeling

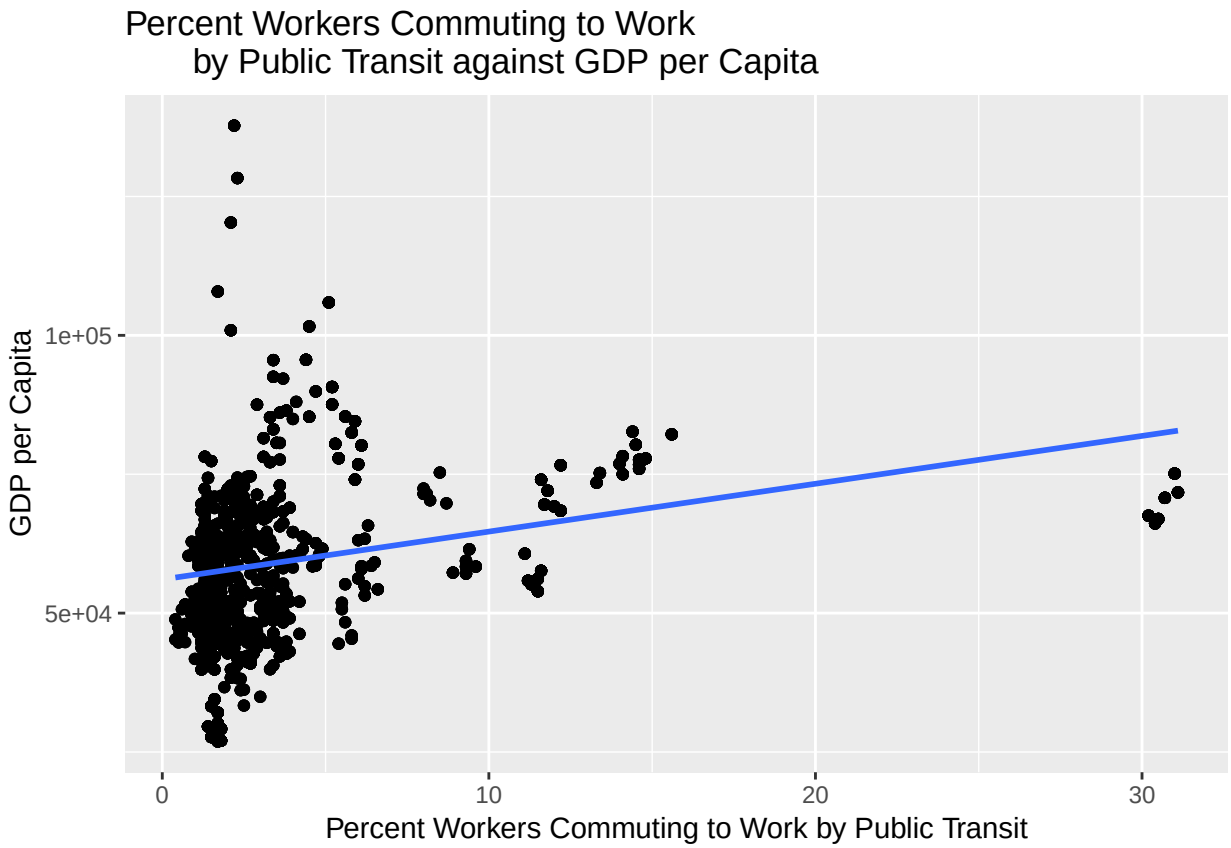
Initial ggplots

These are two examples of ggplots that we created. We created 20 ggplots with different transit variables as explanatory variables. These two ggplots, in which `percent_commuting_msa` and `per_capita_pmt` were utilized, had the highest R2 outputs.

```
## linear models with per_capita_gdp as response variable and different transit
## variables as explanatory variables

# explanatory variable: percent_commuting_msa
model_percent_commuting_msa <- lm(formula = per_capita_gdp ~ percent_commuting_msa,
                                  data = final_df)

ggplot(final_df, aes(x = percent_commuting_msa, y = per_capita_gdp)) +
  geom_point() +
  geom_smooth(method = 'lm', se = F) +
  labs(x = "Percent Workers Commuting to Work by Public Transit",
y = "GDP per Capita",
title = "Percent Workers Commuting to Work
       by Public Transit against GDP per Capita")
```




```
summary(model_percent_commuting_msa)$adj.r.squared
```

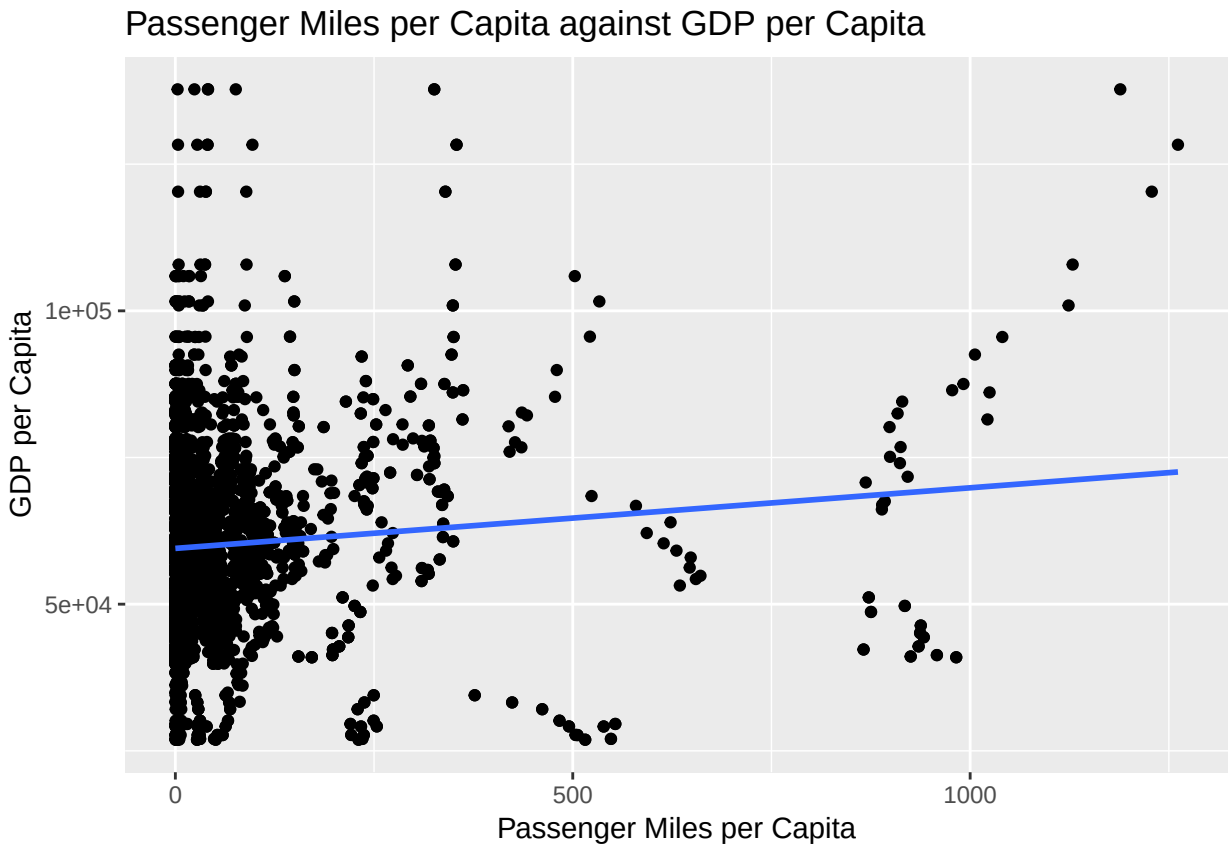
```
## [1] 0.09358545
```

```
summary(model_percent_commuting_msa)$r.squared
```

```
## [1] 0.0938561
```

```
# explanatory variable: per_capita_pmt
model_per_capita_pmt <- lm(formula = per_capita_gdp ~ per_capita_pmt,
                           data = final_df)

ggplot(final_df, aes(x = per_capita_pmt, y = per_capita_gdp)) +
  geom_point() +
  geom_smooth(method = 'lm', se = F) +
  labs(x = "Passenger Miles per Capita", y = "GDP per Capita",
title = "Passenger Miles per Capita against GDP per Capita")
```




```
summary(model_per_capita_pmt)$adj.r.squared
```

```
## [1] 0.007075703
```

```
summary(model_per_capita_pmt)$r.squared
```

```
## [1] 0.007372187
```

Modeling

This function take in up to five variables and uses them as explanatory variables in a regression tree with per capita GDP as the response variable. To create the regression tree we first selected out all columns in the final data frame that weren't being used in this model. We then split up our data 80-20 into training and testing subsets. We then were able to get the resulting MAE, MSE, and R2 from our regression tree. This function returns a data frame with the resulting MAE, MSE, and R2 of the regression tree model for the given explanatory variables.

```
# function to make a regression tree model with response variable per_capita_gdp
gdp_model <- function(explanatory_variables_eq, input1, input2=NULL, input3=NULL,
                      input4=NULL, input5=NULL) {

  set.seed(2)

  select1 <- substitute(input1)
  select2 <- substitute(input2)
  select3 <- substitute(input3)
  select4 <- substitute(input4)
  select5 <- substitute(input5)
```

```
  if(is.null(input2)){
    cleaned_data <- final_df %>%
    select(per_capita_gdp, select1)
  }else if(is.null(input3)){
    cleaned_data <- final_df %>%
    select(per_capita_gdp, select1, select2)
  }else if(is.null(input4)){
    cleaned_data <- final_df %>%
    select(per_capita_gdp, select1, select2, select3)
  }else if(is.null(input5)){
    cleaned_data <- final_df %>%
    select(per_capita_gdp, select1, select2, select3, select4)
  }else{
    cleaned_data <- final_df %>%
    select(per_capita_gdp, select1, select2, select3, select4, select5)}

  # only take rows with all data present
  final_cleaned_data <- cleaned_data[complete.cases(cleaned_data),]

  n <- nrow(final_cleaned_data)

  # select approx 80% of the row numbers between 1 and n
  train_id <- sample(1:n, size=round(n*0.8))

  # the data we'll train the model on
  train_data <- final_cleaned_data[train_id,]

  # the data we'll test the model on
  test_data <- final_cleaned_data[-train_id,]

  # final_train_data <- train_data %>% drop_na("per_capita_gdp")

  regression_formula <- paste0("per_capita_gdp ~ ", explanatory_variables_eq)

  regression_tree <- rpart(regression_formula, data=train_data)
  predictions <- as.vector(predict(regression_tree, test_data))
  final_test_data <- test_data %>% mutate(prediction = predictions)
  final_test_data$truth <- final_test_data[,"per_capita_gdp"]
  results <- final_test_data %>%
    summarise("Explanatory Variables" = explanatory_variables_eq,
              #Percent_Rows = round(n/3416, 2),
              MSE = formatC( (1/n() * sum((truth - predictions)^2, na.rm = T)),
                             format = "e", digits = 2),
              MAE  = round(1/n() * sum(abs(truth - predictions), na.rm = T), 0),
              R2   = round(cor(truth, predictions),3))

  return(results)
}
```

Begin Modeling

Because our ggplots (of transit variables vs. per capita GDP) had very weak R2 values, we did further exploratory modeling using the above function to determine the best transit variables to predict per capita GDP.

```
# use above function on each transit variable as an explanatory variable
a <- gdp_model("percent_commuting_msa", "percent_commuting_msa")
b <- gdp_model("total_transit_expenses", "total_transit_expenses")
c <- gdp_model("directional_route_miles", "directional_route_miles")
d <- gdp_model("vehicle_hours", "vehicle_hours")
e <- gdp_model("passenger_miles", "passenger_miles")
f <- gdp_model("passenger_trips", "passenger_trips")
g <- gdp_model("total_funding", "total_funding")
h <- gdp_model("pmt_per_vrm", "pmt_per_vrm")
i <- gdp_model("pmt_per_vrh", "pmt_per_vrh")
j <- gdp_model("upt_per_vrh", "upt_per_vrh")
k <- gdp_model("per_capita_vrm", "per_capita_vrm")
l <- gdp_model("per_capita_vrh", "per_capita_vrh")
m <- gdp_model("per_capita_pmt", "per_capita_pmt")
n <- gdp_model("per_capita_upt", "per_capita_upt")
o <- gdp_model("recovery_ratio", "recovery_ratio")
p <- gdp_model("fares_per_upt", "fares_per_upt")
q <- gdp_model("cost_per_hour", "cost_per_hour")
r <- gdp_model("cost_per_trip", "cost_per_trip")
s <- gdp_model("cost_per_pmt", "cost_per_pmt")

model_results <- rbind(a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s)

kable(arrange(model_results, desc(as.numeric(R2))),
      caption = "Exploratory Models")
```

Table 1: Exploratory Models

Explanatory Variables	MSE	MAE	R2
percent_commuting_msa	1.91e+08	9982	0.562
total_funding	2.22e+08	11056	0.452
passenger_trips	2.24e+08	11662	0.447
vehicle_hours	2.37e+08	12000	0.389
per_capita_vrm	2.39e+08	11960	0.386
directional_route_miles	2.38e+08	12049	0.385
per_capita_upt	2.44e+08	11983	0.357
per_capita_vrh	2.48e+08	11980	0.335
per_capita_pmt	2.51e+08	12091	0.318
upt_per_vrh	2.46e+08	12517	0.301
total_transit_expenses	2.55e+08	12633	0.294
passenger_miles	2.59e+08	12575	0.268
cost_per_hour	2.33e+08	12332	0.209
pmt_per_vrm	2.60e+08	12687	0.204
fares_per_upt	2.59e+08	12675	0.200
cost_per_trip	2.34e+08	12410	0.198
pmt_per_vrh	2.63e+08	12780	0.172
recovery_ratio	2.91e+08	13004	0.164
cost_per_pmt	2.44e+08	12708	0.081

Refined Modeling

Based on our above modeling we determined that we should do further modeling on `percent_commuting_msa`, `per_capita_vrm`, `per_capita_upt`, `per_capita_vrh` and `per_capita_pmt`. In this code chunk we use our `gdp_model` function with every possible combination of these five explanatory variables and return the results in a data frame with each model's resulting MAE, MSE, and R2.s

```
# get model results

# one explanatory variable
model0 <- gdp_model("per_capita_vrm", "per_capita_vrm")
model1 <- gdp_model("percent_commuting_msa", "percent_commuting_msa")
model2 <- gdp_model("per_capita_pmt","per_capita_pmt")
model3 <- gdp_model("per_capita_vrh", "per_capita_vrh")
model4 <- gdp_model("per_capita_upt", "per_capita_upt")

# two explanatory variables
model5 <- gdp_model("per_capita_vrm + percent_commuting_msa",
                    "per_capita_vrm", "percent_commuting_msa")
model6 <- gdp_model("per_capita_vrm + per_capita_pmt",
                    "per_capita_vrm", "per_capita_pmt")
model7 <- gdp_model("per_capita_vrm + per_capita_vrh",
                    "per_capita_vrm", "per_capita_vrh")
model8 <- gdp_model("per_capita_vrm + per_capita_upt",
                    "per_capita_vrm", "per_capita_upt")
model9 <- gdp_model("percent_commuting_msa + per_capita_pmt",
                    "percent_commuting_msa", "per_capita_pmt")
model10 <- gdp_model("percent_commuting_msa + per_capita_vrh",
                    "percent_commuting_msa", "per_capita_vrh")
model11 <- gdp_model("percent_commuting_msa + per_capita_upt",
                    "percent_commuting_msa", "per_capita_upt")
model12 <- gdp_model("per_capita_upt + per_capita_vrh",
                    "per_capita_upt", "per_capita_vrh")
model13 <- gdp_model("per_capita_upt + per_capita_pmt",
                    "per_capita_upt", "per_capita_pmt")
model14 <- gdp_model("per_capita_vrh + per_capita_pmt",
                    "per_capita_vrh", "per_capita_pmt")

# three explanatory variables
model15 <- gdp_model("per_capita_vrm + per_capita_pmt + percent_commuting_msa",
                     "per_capita_vrm", "per_capita_pmt", "percent_commuting_msa")
model16 <- gdp_model("per_capita_vrm + per_capita_pmt + per_capita_vrh",
                     "per_capita_vrm","per_capita_pmt", "per_capita_vrh")
model17 <- gdp_model("per_capita_vrm + per_capita_pmt + per_capita_upt",
                     "per_capita_vrm","per_capita_pmt", "per_capita_upt")
model18 <- gdp_model("per_capita_vrm + percent_commuting_msa + per_capita_vrh",
                     "per_capita_vrm", "percent_commuting_msa", "per_capita_vrh")
model19 <- gdp_model("per_capita_vrm + percent_commuting_msa + per_capita_upt",
                     "per_capita_vrm", "percent_commuting_msa", "per_capita_upt")
model20 <- gdp_model("per_capita_vrm + per_capita_upt + per_capita_vrh",
                     "per_capita_vrm", "per_capita_upt", "per_capita_vrh")
model21 <- gdp_model("percent_commuting_msa + per_capita_upt + per_capita_vrh",
                     "percent_commuting_msa", "per_capita_upt", "per_capita_vrh")
model22 <- gdp_model("percent_commuting_msa + per_capita_upt + per_capita_pmt",
```

```
                      "percent_commuting_msa", "per_capita_upt", "per_capita_pmt")
model23 <- gdp_model("percent_commuting_msa + per_capita_vrh + per_capita_pmt",
                      "percent_commuting_msa", "per_capita_vrh", "per_capita_pmt")
model24 <- gdp_model("per_capita_vrh + per_capita_upt + per_capita_pmt",
                      "per_capita_vrh", "per_capita_upt", "per_capita_pmt")

# four explanatory variables
model25 <- gdp_model(
  "per_capita_vrm + percent_commuting_msa + per_capita_pmt + per_capita_vrh",
  "per_capita_vrm", "percent_commuting_msa",
  "per_capita_pmt","per_capita_vrh")
model26 <- gdp_model(
  "per_capita_upt + percent_commuting_msa + per_capita_pmt + per_capita_vrh",
  "per_capita_upt", "percent_commuting_msa",
  "per_capita_pmt","per_capita_vrh")
model27 <- gdp_model(
  "per_capita_upt + per_capita_vrm + per_capita_pmt + per_capita_vrh",
  "per_capita_upt", "per_capita_vrm",
  "per_capita_pmt","per_capita_vrh")
model28 <- gdp_model(
  "per_capita_upt + per_capita_vrm + percent_commuting_msa + per_capita_vrh",
  "per_capita_upt", "per_capita_vrm",
  "percent_commuting_msa","per_capita_vrh")
model29 <-  gdp_model(
  "per_capita_upt + per_capita_vrm + percent_commuting_msa + per_capita_pmt",
  "per_capita_upt", "per_capita_vrm",
  "percent_commuting_msa","per_capita_pmt")

# all five explanatory variables
model30 <-  gdp_model(
  "per_capita_vrm + percent_commuting_msa + per_capita_pmt + per_capita_vrh
  + per_capita_upt",
  "per_capita_vrm", "percent_commuting_msa",
  "per_capita_pmt","per_capita_vrh","per_capita_upt")

final_model <- rbind(model0, model1, model2, model3, model4, model5,
                      model6, model7, model8, model9, model10,
                      model11, model12, model13, model14, model15,
                      model16, model17, model18, model19, model20,
                      model21, model22, model23, model24, model25,
                      model26, model27, model28, model29, model30)

kable(head(arrange(final_model, desc(as.numeric(R2))),
           caption="Final Models, Arranged by MAE"), n=6)
```

Explanatory Variables	MSE	MAE	R2
percent_commuting_msa + per_capita_vrh	1.70e+08	9517	0.624
percent_commuting_msa + per_capita_vrh + per_capita_pmt	1.70e+08	9517	0.624
per_capita_vrm + percent_commuting_msa + per_capita_vrh	1.71e+08	9480	0.622
per_capita_vrm + percent_commuting_msa + per_capita_pmt + per_capita_vrh	1.71e+08	9480	0.622
per_capita_upt + per_capita_vrm + percent_commuting_msa + per_capita_vrh	1.71e+08	9480	0.622
per_capita_vrm + percent_commuting_msa + per_capita_pmt + per_capita_vrh + per_capita_upt	1.71e+08	9480	0.622

Results:

We utilized the gluskr package and its comprehensive data frame (transit_qol_df) to perform initial visualizations (using ggplot). In each of the initial visualizations, the response variable was per capita GDP, while the explanatory variable was a different transit-related variable. When a linear model was added to each of these ggplots, every adjusted r2 was below 0.10. Given such low adjusted r2 values for these linear models, when we moved on to modeling, they decided to model all transit variables (individually) against per capita GDP in the regression tree.

We wrote a function that took in a combination of explanatory variables and created a regression tree with per capita GDP as the response variable. Each of the 20 transit variables was first put into the function individually. The five transit variables that yielded the highest r2 were then chosen to be utilized in the next stage of modeling. These were vehicle miles per capita, passenger miles per capita, vehicles hours per capita, and the percent of workers commuting by public transport. It's important to note that these five variables selected were the variables that yielded the highest r2 and that wouldn't be directly related to the size of a transit system, which is, in turn, would be related to population. We decided to not include the variables: total transit funding, number of passenger trips, directional route miles, and vehicle hours, despite their high r2 values, because of their direct relation to the size of a transit system.

In the second stage of modeling, the five chosen transit variables (per_capita_vrm, percent_commuting_msa, per_capita_pmt, per_capita_vrh) were then passed through the same function (with per capita GDP as the response variable) to create regression trees. These five variables were passed into the function in each possible combination for a total of 31 different models. The function allows for any combination of up to five variables to be passed through at one time.

The top six models had very similar (and very high) MSE and MAE values and each contained percent_commuting_msa and per_capita_vrh as variables. Consequently, we chose the model with just explanatory variables percent_commuting_msa and per_capita_vrh because it is the simplest model, and also because it has the lowest MSE and r2, with MAE being the second lowest of the models. This model had an r2 of 0.624, MAE of 9517, and an MSE of 1.70 x 108. Despite MAE and MSE at first appearing to be quite large, given the scale of the response variable (per capita GDP) in use, they are relatively good. This means that any prediction for per capita GDP (which ranges from around $27,000 to $138,000) based on this best model has a mean average error of $9,517. While this is a fairly large number, it is still on the right scale of per capita GDP and means that any prediction this model gives is still in the ballpark of the true value for per capita GDP.

Diagnostics:

The results suggest that the explanatory variables percent_commuting_msa and per_capita_vrh when utilized in a regression tree model can do a relatively good job at predicting the GDP per capita of a metropolitan statistical area. The metrics of MAE, MSE and r2 were what we used for assessing our model. When compared to the other 30 models that we made, the chosen model performed the best. For more description on how we chose our best model, see the results section.

It is worthwhile to discuss implications of what this model suggests about the relationship between transit and GDP per capita. Our model suggests that investing in a city's transit system such that the system sees increases in vehicle hours per capita (per_capita_vrh) may be beneficial for a city's GDP. Furthermore, promotion of the number of workers that commute by public transit (percent_commuting_msa) may achieve a similar goal.

Conclusion:

The question that we sought to address was how the transit system of a metropolitan statistical area could predict GDP per capita. Specifically, we sought to answer which variables pertaining to transit could predict GDP per capita and how well. From our exploration, we learned that the best predictors of a statistical area's GDP per capita are percent_commuting_msa and per_capita_vrh. This finding of correlation intuitively makes sense to us, because it makes sense that a high number of vehicle hours per person and a higher percent of people commuting via public transit would lead to increased gdp per capita.

There are several limitations/sources of error to this current exploration of the relationship between transit variables and GDP per capita for U.S. cities. For example, as mentioned above the overlap between the urbanized areas and the metropolitan statistical areas is a source of error.
Additionally, we did not calculate a transit score that could serve as single explanatory variable. This transit score would have incorporated various transit-related variables and would have weighed them accordingly. We did not pursue this avenue for data analysis because they lacked the transit knowledge necessary to create such a score as they are not transit experts.

There are multiple directions for future research available following this study. Working with the data utilized for this study is made accessible through the creation of the gluskr package.

One suggestion for future research would involve resolving the discrepancy between urbanized areas and metropolitan statistical areas. Another suggestion includes finding human development index (HDI) data that can be joined to the final data frame so as to explore the relationship between transit and HDI instead of transit and GDP per capita. Finally, it would be valuable to create a transit score for data analysis purposes.

Notes:

We created a Shiny app that allows a user to click on any of the cities incorporated into our analyses and to see an output that displays the population, GDP per capita, vehicle hours per capita, and percent workers commuting by public transit for that city. The cities available to click on can be filtered by size of city. Note that all of these statistics are averaged over a 10 year span from 2007-2017 and were informed by our best model (described in sections above). The Shiny app can be found here: https://jonah.shinyapps.io/test/.