

R Analysis of Park Spending

Gabriel Toscano

Contents

| | |
|---|---|
| About the project | 2 |
| Tracking trends over time | 2 |
| Looking at top 10 largest and smallest spenders per resident | 3 |
| Let's see which cities manage the most with the least amount of money in 2020 | 5 |
| Testing more ambitious visualizations using K-clustering methods | 9 |

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v forcats   1.0.0    v stringr   1.5.0
## v lubridate  1.9.2    v tibble   3.2.1
## v purrr      1.0.1    v tidyr    1.3.0
## v readr      2.1.4
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(stringr)
parkdata <- read.csv('park_data.csv')
```

About the project

After a successful internship in Pawnee, IN you have a new job in the city manager's office in Indianapolis. The citizens have approved a bond earmarked for parks spending (it's a lot of money, don't worry about dollars).

It falls to your office to spend the new bond money. The city's strategic plan clearly sets out a vision for making Indianapolis one of the premiere cities in the country for park access and amenities.

The new City Manager has no experience with parks and has asked you to give her "the lay of the land". You have pulled this dataset and want to provide interesting and relevant insights about parks in other major metropolitan areas.

This project is a data analysis simulation for park spending policy using U.S. Park data from 2012 to 2020. In summary this project:

- Identifies nationwide spending patterns over time
- Points out outliers
- Describes and visualizes efficiency in terms of spend per resident

```
parkdata <- parkdata %>%  
  # Changing spend from string to numeric  
  mutate(spend_per_resident_data = as.numeric(str_remove_all(spend_per_resident_data, "\\$|,"))) %>%  
  # Changing pct city to numeric  
  mutate(park_pct_city_data = as.numeric(str_remove(park_pct_city_data, "%"))) %>%  
  mutate(pct_near_park_data = as.numeric(str_remove(pct_near_park_data, "%"))) %>%  
  # making a new variable to track ration of spend to percentage  
  mutate(ratio_spend_to_park_pct = spend_per_resident_data / pct_near_park_data)  
  
# Define a named vector for replacements  
city_replacements <- c(  
  "Arlington, Virginia" = "Arlington",  
  "Washington, D.C." = "Washington, D.C."  
)  
  
# Replace city names in your data frame  
parkdata <- parkdata %>%  
  mutate(city = str_replace_all(city, city_replacements))
```

Tracking trends over time

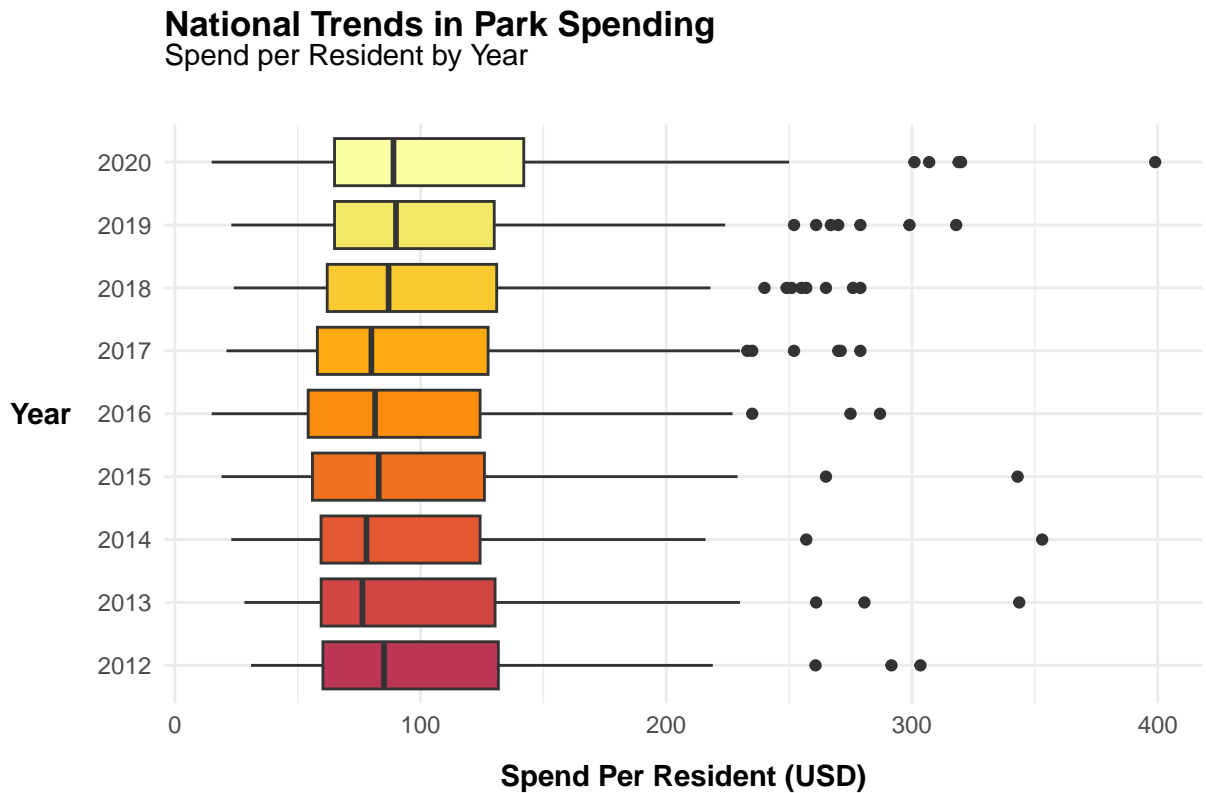
Looking at how spending changes by year

```
ggplot(parkdata, aes(x = factor(year), y = spend_per_resident_data, fill = factor(year))) +  
  geom_boxplot() +  
  scale_fill_viridis_d(option = "B", begin = 0.5, end = 1) +  
  labs(title = "National Trends in Park Spending",  
       subtitle = "Spend per Resident by Year",  
       x = "Year",  
       y = "Spend Per Resident (USD)") +  
  theme_minimal() +  
  
#Styles for text and labels  
theme(legend.position = 'none',
```

```

plot.title=element_text(face="bold", margin=margin(t=10)),
plot.subtitle = element_text(margin = margin(b=20)),
axis.title.y = element_text(face="bold", angle = 0, hjust = 0, vjust=0.5, margin=margin(l=10, r=10, t=0, b=0)),
axis.title.x = element_text(face="bold", margin=margin(t=10, b=10))
) +
coord_flip()

```



Looking at top 10 largest and smallest spenders per resident

```

year_i <- 2017

top_10_spend_2020 <- parkdata %>%
  dplyr::select(city, year, spend_per_resident_data) %>%
  filter(year == year_i) %>%
  arrange(desc(spend_per_resident_data)) %>%
  slice(1:10)

bottom_10_spend_2020 <- parkdata %>%
  dplyr::select(city, year, spend_per_resident_data) %>%
  filter(year == year_i) %>%
  arrange(spend_per_resident_data) %>%
  slice(1:10)

```

```

# indiana_df <- parkdata %>%
#   select(city, year, spend_per_resident_data) %>%
#   filter(city=="Indianapolis", year==year_i)

combine_top_and_bottom <- bind_rows(top_10_spend_2020, bottom_10_spend_2020)
# combine_top_and_bottom <- bind_rows(combine_top_and_bottom, indiana_df)

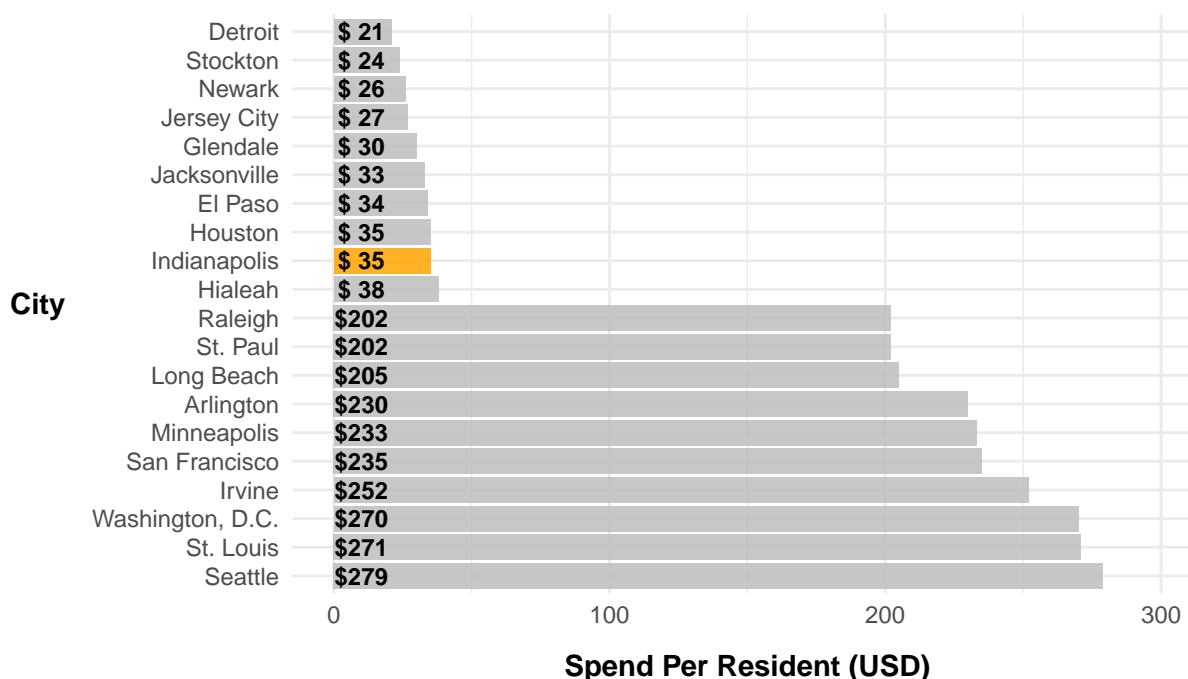
# top_10_spend_2020$city <- factor(top_10_spend_2020$city)
# Calculate a position for the labels
# max_spend <- max(top_10_spend_2020$spend_per_resident_data)
max_spend <- max(combine_top_and_bottom$spend_per_resident_data)

ggplot(combine_top_and_bottom, aes(x = fct_rev(fct_reorder(city, spend_per_resident_data)),
                                   y = spend_per_resident_data,
                                   fill = ifelse(city == "Indianapolis", "Indianapolis", "Other"))) +
  geom_col(alpha = 0.85) + # Use the fill mapping from aes()
  scale_fill_manual(values = c("Indianapolis" = "orange", "Other" = "gray")) + # Set colors
  labs(title = "Cities with the Least and Most Park Spending in 2017",
        subtitle = "Spend per Resident by City",
        x = "City",
        y = "Spend Per Resident (USD)") +
  theme_minimal() +
  ylim(0, 300) +
  theme(legend.position = 'none',
        plot.title = element_text(face = "bold", margin = margin(t = 10)),
        plot.subtitle = element_text(margin = margin(b = 20)),
        axis.title.y = element_text(face = "bold", angle = 0, hjust = 0, vjust = 0.5, margin = margin(10, 0, 0, 0)),
        axis.title.x = element_text(face = "bold", margin = margin(t = 10, b = 10))) +
  coord_flip() +
  geom_text(aes(y = max_spend - 269, # Fixed y position for all labels
                label = paste0("$", format(spend_per_resident_data, big.mark = ",")),
                color = "black", # Change text color as needed
                size = 3,
                fontface = "bold"))

```

Cities with the Least and Most Park Spending in 2017

Spend per Resident by City



Let's see which cities manage the most with the least amount of money in 2020

```

spend_to_pct_city_ratio <- parkdata %>%
  select(city, ratio_spend_to_park_pct, park_pct_city_data, spend_per_resident_data, year) %>%
  filter(year == 2020)

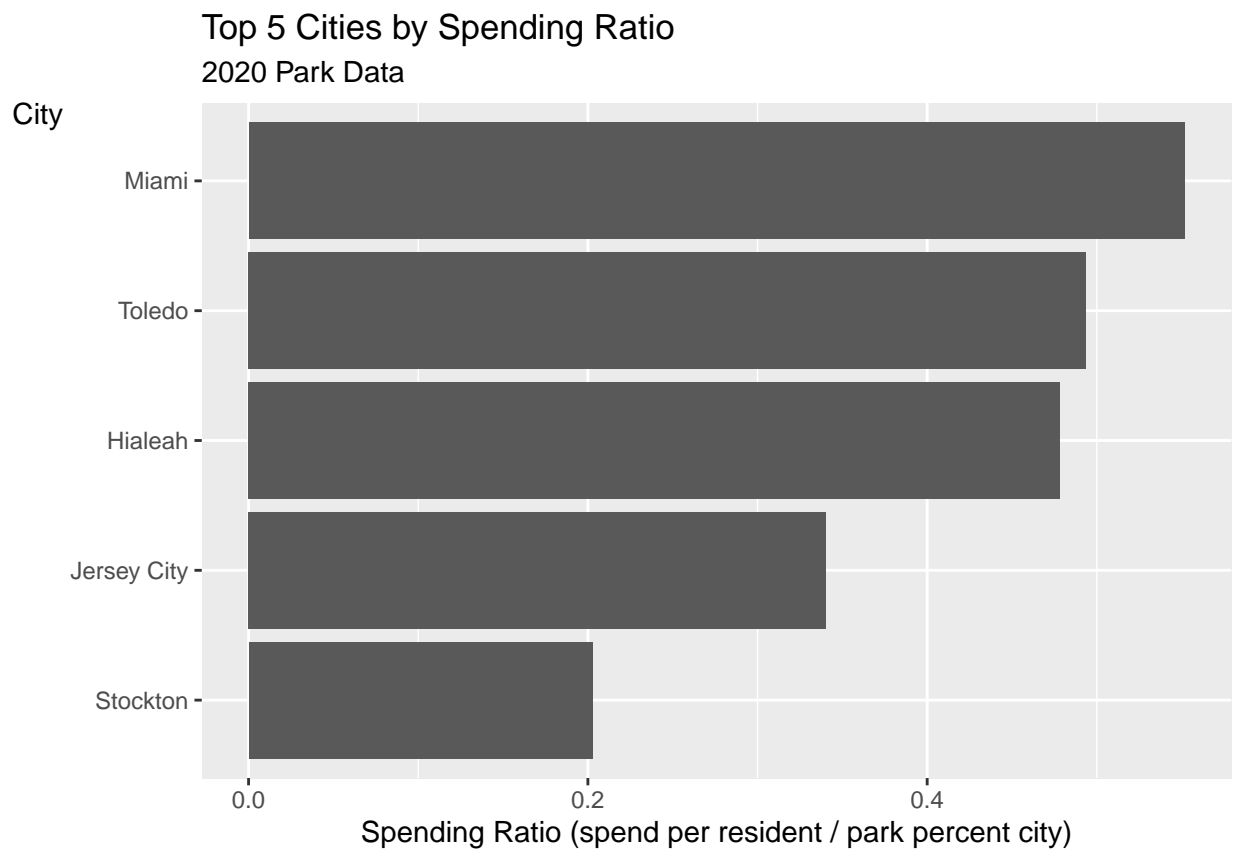
#spending the most amount of money per percent of city
doing_the_least <- spend_to_pct_city_ratio %>%
  arrange(-ratio_spend_to_park_pct) %>%
  slice(1:5)

#spending the least amount of money per percent of city
doing_the_most <- spend_to_pct_city_ratio %>%
  arrange(ratio_spend_to_park_pct) %>%
  slice(1:5)

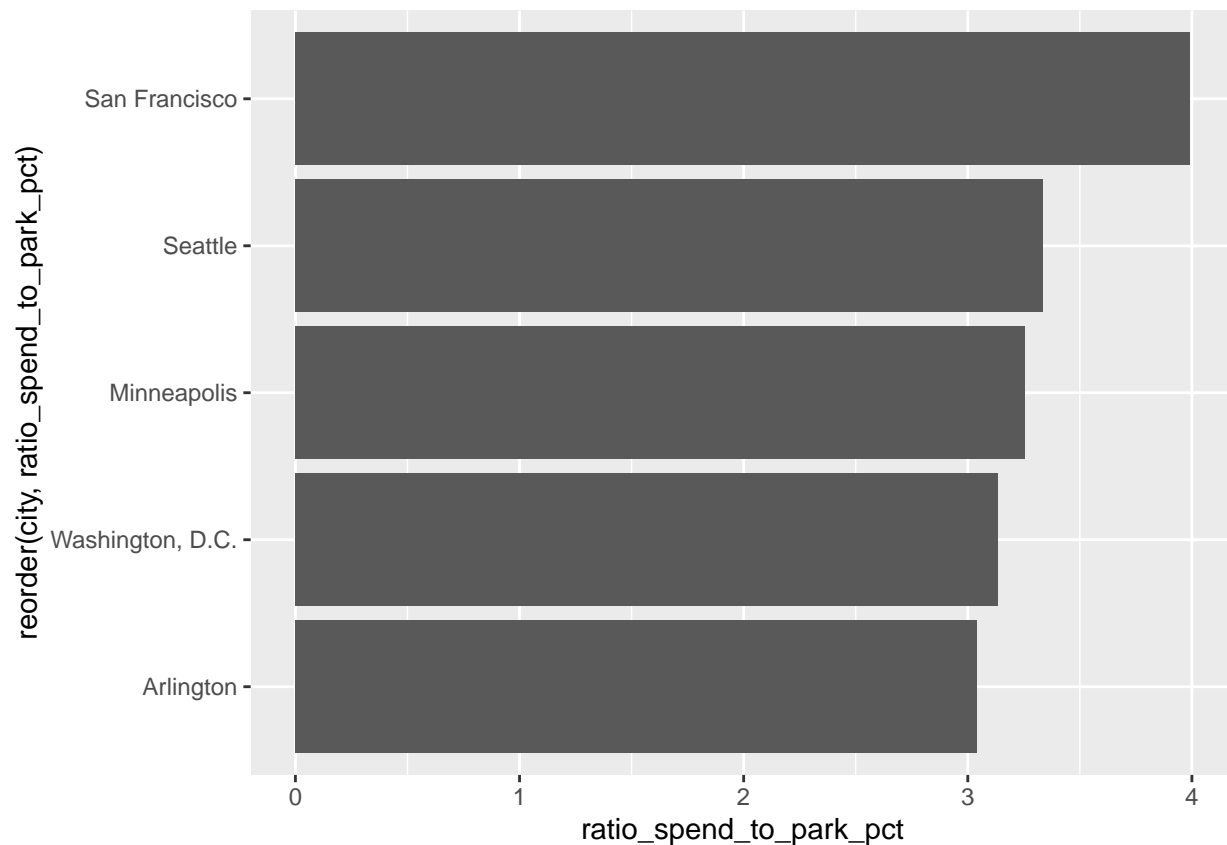
doing_the_most %>%
  ggplot(aes(x=reorder(city, ratio_spend_to_park_pct), y=ratio_spend_to_park_pct)) +
  geom_bar(stat='identity') +
  coord_flip() +
  labs(
    title="Top 5 Cities by Spending Ratio",
    subtitle="2020 Park Data",
    y = "Spending Ratio (spend per resident / park percent city)",
  )

```

```
x = "City"
) + theme(axis.title.y = element_text(angle = 0, hjust = 0, vjust=1))
```



```
doing_the_least %>%
  ggplot(aes(x=reorder(city, ratio_spend_to_park_pct), y=ratio_spend_to_park_pct)) +
  geom_bar(stat='identity') +
  coord_flip()
```



```

doing_the_most$color <- "top"
doing_the_least$color <- "bottom"

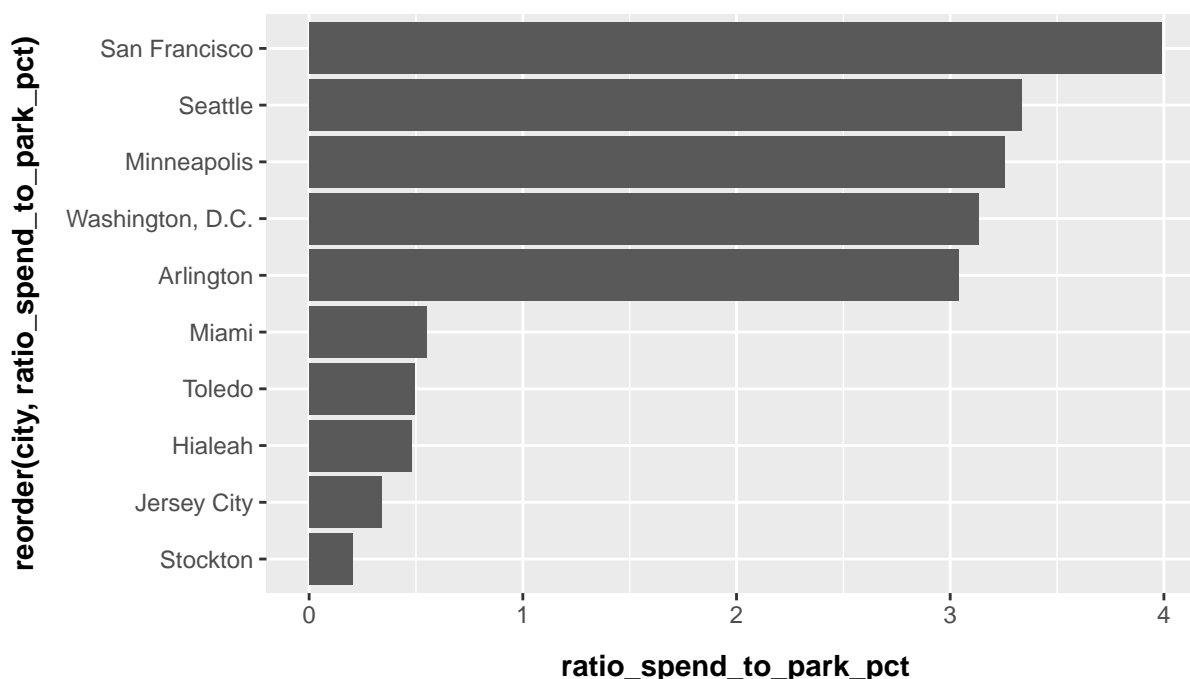
most_and_least_ratio <-
  bind_rows(ddoing_the_least, doing_the_most)

most_and_least_ratio %>%
  ggplot(aes(x=reorder(city, ratio_spend_to_park_pct), y=ratio_spend_to_park_pct)) +
  geom_bar(stat='identity') +
  coord_flip() +
  labs(
    title="Top 5 Most and Least Efficient Cities",
    subtitle="Turning money into proximity to parks"
  ) +
  theme(
    plot.title = element_text(face = "bold", margin = margin(t = 10)),
    plot.subtitle = element_text(margin = margin(b = 20)),
    axis.title.y = element_text(face = "bold", margin = margin(l = 10, r = 10)),
    axis.title.x = element_text(face = "bold", margin = margin(t = 10, b = 10)),
    legend.title = element_blank() # Optional: Remove legend title
  ) +
  scale_color_manual(values = c("top" = "darkgreen", "bottom" = "red")) # Set colors for the points

```

Top 5 Most and Least Efficient Cities

Turning money into proximity to parks



```
national_avg_2017 <- parkdata %>%
  filter(year == 2017, !is.na(spend_per_resident_data) ) %>% # Filter for the year 2017
  summarise(across(
    c(
      med_park_size_data,
      park_pct_city_data,
      pct_near_park_data,
      spend_per_resident_data,
      basketball_data,
      dogpark_data,
      playground_data,
      rec_sr_data,
      restroom_data,
      splashground_data
    ),
    ~ mean(.x, na.rm = TRUE) # Calculate mean for each variable
  )) %>%
  # select(where(~ !any(is.na(.)))) %>% # Select columns without any NA values
  mutate(city = "National Average")
```

```
parkdata$city_color <- ifelse(parkdata$city == "Indianapolis", "Indianapolis", "Other")

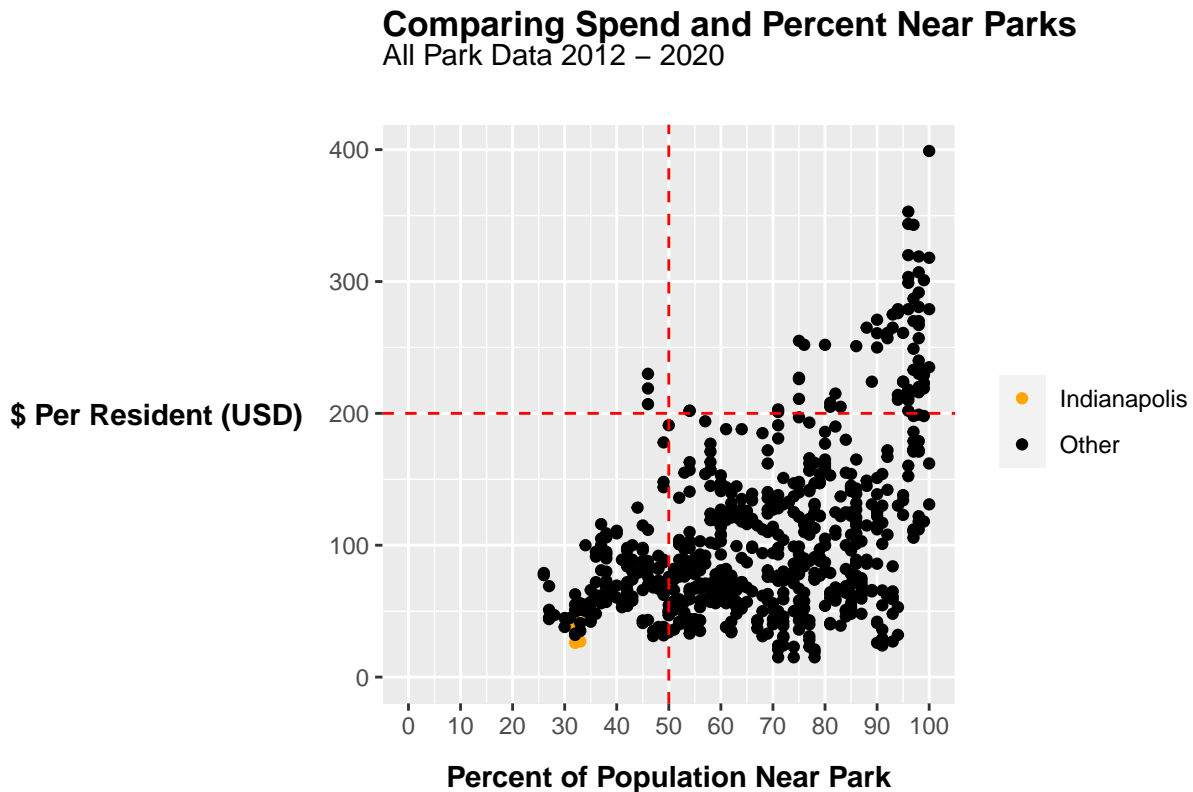
ggplot(parkdata, aes(x = pct_near_park_data, y = spend_per_resident_data, color = city_color)) +
  geom_point() +
  labs(
```



```

title = "Comparing Spend and Percent Near Parks",
subtitle = "All Park Data 2012 - 2020",
y = "$ Per Resident (USD)",
x = "Percent of Population Near Park"
) +
scale_x_continuous(breaks = seq(0, 100, by = 10), limits = c(0, 100)) + # Set x-axis labels every 10
scale_y_continuous(limits = c(0, max(parkdata$spend_per_resident_data, na.rm = TRUE))) + # Set y-axis
theme(
  plot.title = element_text(face = "bold", margin = margin(t = 10)),
  plot.subtitle = element_text(margin = margin(b = 20)),
  axis.title.y = element_text(face = "bold", margin = margin(l = 10, r = 10), angle=0, vjust=0.5),
  axis.title.x = element_text(face = "bold", margin = margin(t = 10, b = 10)),
  legend.title = element_blank() # Optional: Remove legend title
) +
geom_hline(yintercept = 200, linetype = "dashed", color = "red") + # Add horizontal line at y = 200
geom_vline(xintercept = 50, linetype = "dashed", color = "red") + # Add vertical line at x = 50
scale_color_manual(values = c("Indianapolis" = "orange", "Other" = "black")) # Set colors for the po

```



```

ggsave(file="spend_and_pct_near_parks.png", width=10, height=4, dpi=300)

```

Testing more ambitious visualizations using K-clustering methods

```
library(useful)
```

```
## Warning: package 'useful' was built under R version 4.3.1
```

```
# Select relevant columns and ensure they are numeric
```

```
parkTrainingSet <- parkdata %>%  
  select(city, pct_near_park_data, spend_per_resident_data, park_pct_city_data) %>%  
  mutate(across(c(pct_near_park_data, spend_per_resident_data), as.numeric))
```

```
# Remove rows with NAs
```

```
parkTrain <- na.omit(parkTrainingSet)
```

```
# Check the structure to ensure the numeric data
```

```
str(parkTrain)
```

```
## 'data.frame': 713 obs. of 4 variables:
```

```
## $ city : chr "Charlotte" "Charlotte" "Charlotte" "Charlotte" ...
```

```
## $ pct_near_park_data : num 26 28 26 27 27 37 42 43 27 36 ...
```

```
## $ spend_per_resident_data: num 78.9 47 77.3 69 51 ...
```

```
## $ park_pct_city_data : num 4.1 4 5.01 5.1 5.6 11 11 11 4 6 ...
```

```
# Set seed for reproducibility
```

```
set.seed(278613)
```

```
# Perform k-means clustering using only the numeric columns
```

```
park_model <- kmeans(x = parkTrain[, c("pct_near_park_data", "spend_per_resident_data", "park_pct_city_data")])
```

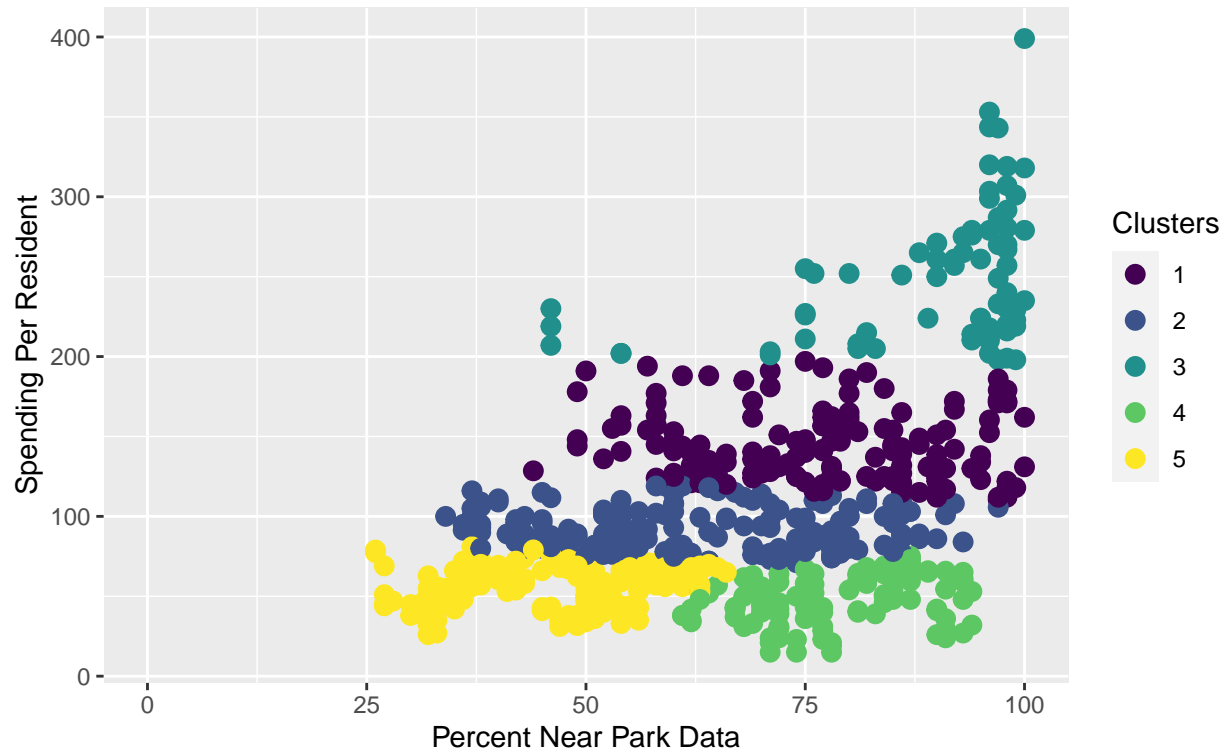
```
parkTrain$cluster <- park_model$cluster
```

```
# Create the ggplot
```

```
ggplot(parkTrain, aes(x = pct_near_park_data, y = spend_per_resident_data, color = factor(cluster))) +  
  geom_point(size = 3) + # Use points with a specified size  
  labs(  
    title = "K-means Clustering of Cities",  
    subtitle = "Grouping by Similarity",  
    x = "Percent Near Park Data",  
    y = "Spending Per Resident",  
    color = "Clusters" # Legend title  
  ) +  
  scale_color_viridis_d() + # Use viridis color scale for clusters  
  xlim(0, 100)
```

K-means Clustering of Cities

Grouping by Similarity



```
theme_minimal() + # Use a minimal theme for a clean look
theme(
  plot.title = element_text(face = "bold", margin = margin(t = 10)),
  plot.subtitle = element_text(margin = margin(b = 20)),
  axis.title.y = element_text(face = "bold", angle = 0, hjust = 0, vjust = 0.5, margin = margin(l = 10, r = 10)),
  axis.title.x = element_text(face = "bold", margin = margin(t = 10, b = 10)),
  legend.title = element_blank() # Optional: Remove legend title
)
```

```
## List of 97
## $ line :List of 6
## ..$ colour : chr "black"
## ..$ linewidth : num 0.5
## ..$ linetype : num 1
## ..$ lineend : chr "butt"
## ..$ arrow : logi FALSE
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_line" "element"
## $ rect :List of 5
## ..$ fill : chr "white"
## ..$ colour : chr "black"
## ..$ linewidth : num 0.5
## ..$ linetype : num 1
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_rect" "element"
## $ text :List of 11
```

```

## ..$ family      : chr ""
## ..$ face        : chr "plain"
## ..$ colour      : chr "black"
## ..$ size        : num 11
## ..$ hjust       : num 0.5
## ..$ vjust       : num 0.5
## ..$ angle       : num 0
## ..$ lineheight  : num 0.9
## ..$ margin      : 'margin' num [1:4] 0points 0points 0points 0points
## .. ..- attr(*, "unit")= int 8
## ..$ debug       : logi FALSE
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ title         : NULL
## $ aspect.ratio  : NULL
## $ axis.title     : NULL
## $ axis.title.x   :List of 11
## ..$ family      : NULL
## ..$ face        : chr "bold"
## ..$ colour      : NULL
## ..$ size        : NULL
## ..$ hjust       : NULL
## ..$ vjust       : num 1
## ..$ angle       : NULL
## ..$ lineheight  : NULL
## ..$ margin      : 'margin' num [1:4] 10points 0points 10points 0points
## .. ..- attr(*, "unit")= int 8
## ..$ debug       : NULL
## ..$ inherit.blank: logi FALSE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.title.x.top :List of 11
## ..$ family      : NULL
## ..$ face        : NULL
## ..$ colour      : NULL
## ..$ size        : NULL
## ..$ hjust       : NULL
## ..$ vjust       : num 0
## ..$ angle       : NULL
## ..$ lineheight  : NULL
## ..$ margin      : 'margin' num [1:4] 0points 0points 2.75points 0points
## .. ..- attr(*, "unit")= int 8
## ..$ debug       : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.title.x.bottom : NULL
## $ axis.title.y       :List of 11
## ..$ family      : NULL
## ..$ face        : chr "bold"
## ..$ colour      : NULL
## ..$ size        : NULL
## ..$ hjust       : num 0
## ..$ vjust       : num 0.5
## ..$ angle       : num 0
## ..$ lineheight  : NULL

```

```

## ..$ margin      : 'margin' num [1:4] 0points 10points 0points 10points
## .. ..- attr(*, "unit")= int 8
## ..$ debug       : NULL
## ..$ inherit.blank: logi FALSE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.title.y.left      : NULL
## $ axis.title.y.right     :List of 11
## ..$ family         : NULL
## ..$ face           : NULL
## ..$ colour         : NULL
## ..$ size           : NULL
## ..$ hjust          : NULL
## ..$ vjust          : num 0
## ..$ angle          : num -90
## ..$ lineheight     : NULL
## ..$ margin        : 'margin' num [1:4] 0points 0points 0points 2.75points
## .. ..- attr(*, "unit")= int 8
## ..$ debug         : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text          :List of 11
## ..$ family         : NULL
## ..$ face           : NULL
## ..$ colour         : chr "grey30"
## ..$ size           : 'rel' num 0.8
## ..$ hjust          : NULL
## ..$ vjust          : NULL
## ..$ angle          : NULL
## ..$ lineheight     : NULL
## ..$ margin        : NULL
## ..$ debug         : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.x        :List of 11
## ..$ family         : NULL
## ..$ face           : NULL
## ..$ colour         : NULL
## ..$ size           : NULL
## ..$ hjust          : NULL
## ..$ vjust          : num 1
## ..$ angle          : NULL
## ..$ lineheight     : NULL
## ..$ margin        : 'margin' num [1:4] 2.2points 0points 0points 0points
## .. ..- attr(*, "unit")= int 8
## ..$ debug         : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.x.top    :List of 11
## ..$ family         : NULL
## ..$ face           : NULL
## ..$ colour         : NULL
## ..$ size           : NULL
## ..$ hjust          : NULL
## ..$ vjust          : num 0

```

```

## ..$ angle          : NULL
## ..$ lineheight     : NULL
## ..$ margin         : 'margin' num [1:4] 0points 0points 2.2points 0points
## ..- attr(*, "unit")= int 8
## ..$ debug          : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.x.bottom : NULL
## $ axis.text.y         :List of 11
## ..$ family          : NULL
## ..$ face            : NULL
## ..$ colour          : NULL
## ..$ size            : NULL
## ..$ hjust           : num 1
## ..$ vjust           : NULL
## ..$ angle           : NULL
## ..$ lineheight      : NULL
## ..$ margin          : 'margin' num [1:4] 0points 2.2points 0points 0points
## ..- attr(*, "unit")= int 8
## ..$ debug           : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.y.left   : NULL
## $ axis.text.y.right  :List of 11
## ..$ family          : NULL
## ..$ face            : NULL
## ..$ colour          : NULL
## ..$ size            : NULL
## ..$ hjust           : num 0
## ..$ vjust           : NULL
## ..$ angle           : NULL
## ..$ lineheight      : NULL
## ..$ margin          : 'margin' num [1:4] 0points 0points 0points 2.2points
## ..- attr(*, "unit")= int 8
## ..$ debug           : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.ticks         : list()
## ..- attr(*, "class")= chr [1:2] "element_blank" "element"
## $ axis.ticks.x       : NULL
## $ axis.ticks.x.top   : NULL
## $ axis.ticks.x.bottom : NULL
## $ axis.ticks.y       : NULL
## $ axis.ticks.y.left  : NULL
## $ axis.ticks.y.right : NULL
## $ axis.ticks.length  : 'simpleUnit' num 2.75points
## ..- attr(*, "unit")= int 8
## $ axis.ticks.length.x : NULL
## $ axis.ticks.length.x.top : NULL
## $ axis.ticks.length.x.bottom: NULL
## $ axis.ticks.length.y : NULL
## $ axis.ticks.length.y.left : NULL
## $ axis.ticks.length.y.right : NULL
## $ axis.line          : list()

```

```

##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##   $ axis.line.x           : NULL
##   $ axis.line.x.top       : NULL
##   $ axis.line.x.bottom    : NULL
##   $ axis.line.y           : NULL
##   $ axis.line.y.left      : NULL
##   $ axis.line.y.right     : NULL
##   $ legend.background     : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##   $ legend.margin         : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.5points
##   ..- attr(*, "unit")= int 8
##   $ legend.spacing        : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##   $ legend.spacing.x      : NULL
##   $ legend.spacing.y      : NULL
##   $ legend.key            : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##   $ legend.key.size       : 'simpleUnit' num 1.2lines
##   ..- attr(*, "unit")= int 3
##   $ legend.key.height     : NULL
##   $ legend.key.width      : NULL
##   $ legend.text           :List of 11
##   ..$ family              : NULL
##   ..$ face                : NULL
##   ..$ colour              : NULL
##   ..$ size                : 'rel' num 0.8
##   ..$ hjust               : NULL
##   ..$ vjust               : NULL
##   ..$ angle               : NULL
##   ..$ lineheight          : NULL
##   ..$ margin              : NULL
##   ..$ debug               : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##   $ legend.text.align     : NULL
##   $ legend.title          : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##   $ legend.title.align    : NULL
##   $ legend.position       : chr "right"
##   $ legend.direction      : NULL
##   $ legend.justification  : chr "center"
##   $ legend.box            : NULL
##   $ legend.box.just       : NULL
##   $ legend.box.margin     : 'margin' num [1:4] 0cm 0cm 0cm 0cm
##   ..- attr(*, "unit")= int 1
##   $ legend.box.background : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##   $ legend.box.spacing    : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##   $ panel.background      : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##   $ panel.border          : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##   $ panel.spacing         : 'simpleUnit' num 5.5points

```

```

##   .- attr(*, "unit")= int 8
##   $ panel.spacing.x      : NULL
##   $ panel.spacing.y      : NULL
##   $ panel.grid           :List of 6
##   ..$ colour            : chr "grey92"
##   ..$ linewidth         : NULL
##   ..$ linetype          : NULL
##   ..$ lineend           : NULL
##   ..$ arrow             : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   .- attr(*, "class")= chr [1:2] "element_line" "element"
##   $ panel.grid.major     : NULL
##   $ panel.grid.minor     :List of 6
##   ..$ colour            : NULL
##   ..$ linewidth         : 'rel' num 0.5
##   ..$ linetype          : NULL
##   ..$ lineend           : NULL
##   ..$ arrow             : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   .- attr(*, "class")= chr [1:2] "element_line" "element"
##   $ panel.grid.major.x   : NULL
##   $ panel.grid.major.y   : NULL
##   $ panel.grid.minor.x   : NULL
##   $ panel.grid.minor.y   : NULL
##   $ panel.ontop          : logi FALSE
##   $ plot.background      : list()
##   .- attr(*, "class")= chr [1:2] "element_blank" "element"
##   $ plot.title           :List of 11
##   ..$ family            : NULL
##   ..$ face              : chr "bold"
##   ..$ colour            : NULL
##   ..$ size              : 'rel' num 1.2
##   ..$ hjust             : num 0
##   ..$ vjust             : num 1
##   ..$ angle             : NULL
##   ..$ lineheight        : NULL
##   ..$ margin            : 'margin' num [1:4] 10points 0points 0points 0points
##   .. .- attr(*, "unit")= int 8
##   ..$ debug             : NULL
##   ..$ inherit.blank: logi FALSE
##   .- attr(*, "class")= chr [1:2] "element_text" "element"
##   $ plot.title.position  : chr "panel"
##   $ plot.subtitle        :List of 11
##   ..$ family            : NULL
##   ..$ face              : NULL
##   ..$ colour            : NULL
##   ..$ size              : NULL
##   ..$ hjust             : num 0
##   ..$ vjust             : num 1
##   ..$ angle             : NULL
##   ..$ lineheight        : NULL
##   ..$ margin            : 'margin' num [1:4] 0points 0points 20points 0points
##   .. .- attr(*, "unit")= int 8
##   ..$ debug             : NULL

```



```

## ..$ inherit.blank: logi FALSE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ plot.caption          :List of 11
## ..$ family             : NULL
## ..$ face               : NULL
## ..$ colour             : NULL
## ..$ size               : 'rel' num 0.8
## ..$ hjust              : num 1
## ..$ vjust              : num 1
## ..$ angle              : NULL
## ..$ lineheight         : NULL
## ..$ margin             : 'margin' num [1:4] 5.5points 0points 0points 0points
## ..- attr(*, "unit")= int 8
## ..$ debug              : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ plot.caption.position : chr "panel"
## $ plot.tag              :List of 11
## ..$ family             : NULL
## ..$ face               : NULL
## ..$ colour             : NULL
## ..$ size               : 'rel' num 1.2
## ..$ hjust              : num 0.5
## ..$ vjust              : num 0.5
## ..$ angle              : NULL
## ..$ lineheight         : NULL
## ..$ margin             : NULL
## ..$ debug              : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ plot.tag.position     : chr "topleft"
## $ plot.margin           : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.5points
## ..- attr(*, "unit")= int 8
## $ strip.background      : list()
## ..- attr(*, "class")= chr [1:2] "element_blank" "element"
## $ strip.background.x    : NULL
## $ strip.background.y    : NULL
## $ strip.clip            : chr "inherit"
## $ strip.placement       : chr "inside"
## $ strip.text            :List of 11
## ..$ family             : NULL
## ..$ face               : NULL
## ..$ colour             : chr "grey10"
## ..$ size               : 'rel' num 0.8
## ..$ hjust              : NULL
## ..$ vjust              : NULL
## ..$ angle              : NULL
## ..$ lineheight         : NULL
## ..$ margin             : 'margin' num [1:4] 4.4points 4.4points 4.4points 4.4points
## ..- attr(*, "unit")= int 8
## ..$ debug              : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ strip.text.x          : NULL

```

```
## $ strip.text.x.bottom      : NULL
## $ strip.text.x.top        : NULL
## $ strip.text.y            :List of 11
## ..$ family                : NULL
## ..$ face                  : NULL
## ..$ colour                : NULL
## ..$ size                  : NULL
## ..$ hjust                 : NULL
## ..$ vjust                 : NULL
## ..$ angle                 : num -90
## ..$ lineheight           : NULL
## ..$ margin               : NULL
## ..$ debug                : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ strip.text.y.left       :List of 11
## ..$ family                : NULL
## ..$ face                  : NULL
## ..$ colour                : NULL
## ..$ size                  : NULL
## ..$ hjust                 : NULL
## ..$ vjust                 : NULL
## ..$ angle                 : num 90
## ..$ lineheight           : NULL
## ..$ margin               : NULL
## ..$ debug                : NULL
## ..$ inherit.blank: logi TRUE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ strip.text.y.right      : NULL
## $ strip.switch.pad.grid    : 'simpleUnit' num 2.75points
## ..- attr(*, "unit")= int 8
## $ strip.switch.pad.wrap    : 'simpleUnit' num 2.75points
## ..- attr(*, "unit")= int 8
## - attr(*, "class")= chr [1:2] "theme" "gg"
## - attr(*, "complete")= logi TRUE
## - attr(*, "validate")= logi TRUE
```

```
# Find the cluster number for Indianapolis
indianapolis_cluster <- parkTrain %>%
  filter(city == "Indianapolis") %>%
  pull(cluster)

# Get the names of cities in the same cluster as Indianapolis
cities_in_same_cluster <- parkTrain %>%
  filter(cluster == indianapolis_cluster) %>%
  select(city) %>%
  distinct(city)
```

```
## Warning: There was 1 warning in 'filter()'.
## i In argument: 'cluster == indianapolis_cluster'.
## Caused by warning in 'cluster == indianapolis_cluster':
## ! longer object length is not a multiple of shorter object length
```

```

# Get the names of cities in cluster 4 (low spend, high coverage)
cities_in_cluster_4 <- parkTrain %>%
  filter(cluster == 4) %>%
  select(city) %>%
  distinct(city)

# who changed the most in spending

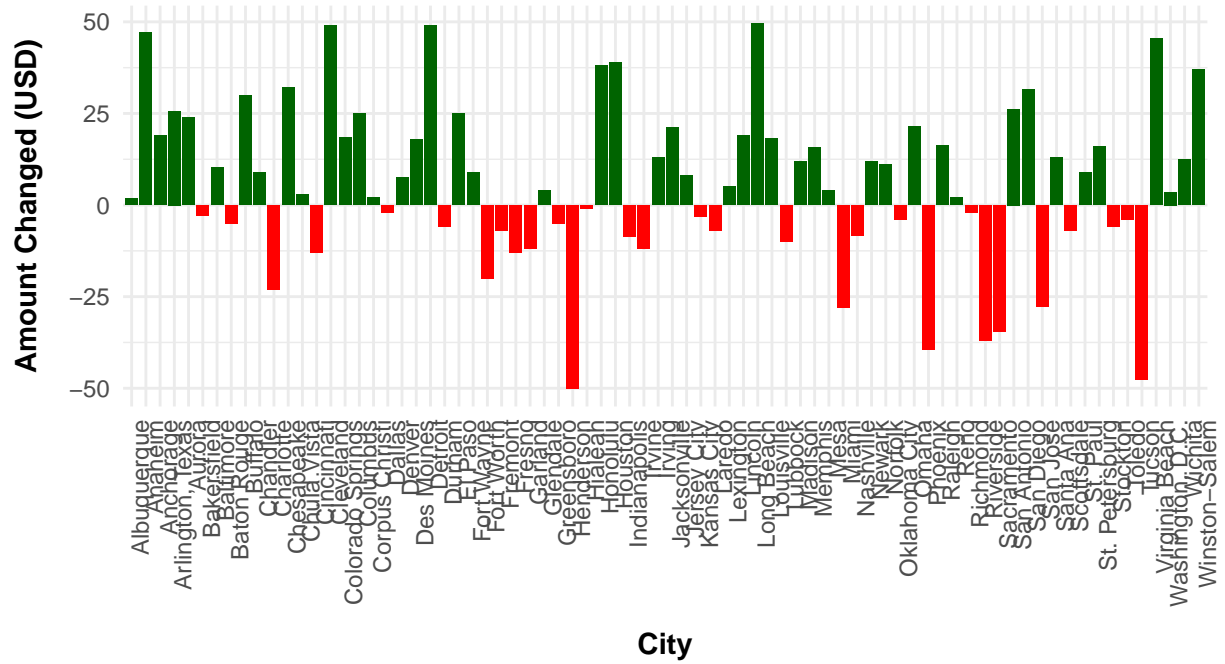
change_in_spend <- parkdata %>%
  group_by(city) %>%
  summarize(
    change = spend_per_resident_data[year == max(year)] - spend_per_resident_data[year == min(year)]
  ) %>%
  filter(change >= -50 & change <= 50 ) %>% #removing outliers
  filter(change != 0) #removing non changing cities

ggplot(change_in_spend, aes(x = city, y = change, fill = change > 0)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("TRUE" = "darkgreen", "FALSE" = "red")) +
  labs(title = "Change in Spending Per Resident by City",
       subtitle = "First Year vs Most Recent Year Spending",
       x = "City",
       y = "Amount Changed (USD)") +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", margin = margin(t = 10)),
    plot.subtitle = element_text(margin = margin(b = 20, t=5)),
    axis.title.y = element_text(face = "bold", margin = margin(l = 10, r = 10)),
    axis.title.x = element_text(face = "bold", margin = margin(t = 10, b = 10)),
    legend.position = "none", # Remove the legend
    axis.text.x = element_text(angle = 90, hjust = 1) # Angle x-axis labels to 90 degrees
  )

```

Change in Spending Per Resident by City

First Year vs Most Recent Year Spending



```
ggsave(file="changes_in_spending.png", width=15, height=4, dpi=300)
```