Exploring citizen-government interactions through analysis of twitter data

Mehdi Khan

December 2, 2017

Project Proposal:

Introduction: My interest in built environment was the reason I studied architecture. As an architect in my early career and now as a GIS/data professional in Planning departments in local government settings I regularly see the importance of data on the success or failure of design and planning decisions. Although the application of data science in business and finance industries etc. are huge, nevertheless, the wave of Bigdata and analytics have hit the field of urban planning too, which is conceptualized and defined by several terms, one of which is "Smart City". The concept of smart city could be explained as data driven city or urban development through the engagement of its four components - the government, the citizens, private businesses and academia. A data science project to measure the engagement and/or relationship between a local government entity and its citizens within the context of urban development or city operations is proposed here.

The problem statement: The project will examine if tweeter messages used by local governments and/or tweeter interactions between the governments and citizens can be used to track the level of involvement of citizens with their government (and vice versa) about urban planning or urban policy issues; and if these interactions can successfully be used to capture and visualize the frustrations or satisfactions of the citizens about various development/policy decisions.

Data source and scope of the project: Tweeter data that were sent by the governments and responses to those messages by the citizens (such as number of retweets, replies etc.) will be used as the primary data sources. Based on the availability of the data, the project will be limited to either one or more local governments or one or more agencies. Private or non-profit entities may be included based on the data availability, relevance and time.

Other consideration: Since urban developments and policies are tied to the use of land with specific boundaries, a spatial component or spatial analysis may be added to the project.

PROJECT DETAILS:

Area of interest and sample data: Howard County, Maryland a jurisdiction of around 300,000 people was selected as the area of interest for this project. Howard County government is active in social media and post messages about government events and news regularly. The diverse citizens with above average education and income were thought to be responsive and concerned about their governments' activities. Therefore, Howard County seemed to be a good candidate for the proposed study.

Although the proposal intended to only examine tweets related to urban planning and urban policy, because of the lack of enough data all tweets were considered.

Project restrictions: Twitter does not allow to access tweets that are more than two weeks old. In addition to that there are also restrictions on how many tweets will be returned by individual functions using twitter API.

Load libraries:

```
suppressWarnings(suppressMessages(library(twitteR)))
suppressWarnings(suppressMessages(library(RCurl)))
suppressWarnings(suppressMessages(library(RJSONIO)))
suppressWarnings(suppressMessages(library(stringr)))
suppressWarnings(suppressMessages(library(rtweet)))
suppressWarnings(suppressMessages(library(dismo)))
suppressWarnings(suppressMessages(library(maps)))
suppressWarnings(suppressMessages(library(ggplot2)))
suppressWarnings(suppressMessages(library(XML)))
suppressWarnings(suppressMessages(library(dplyr)))
suppressWarnings(suppressMessages(library(aws.s3)))
suppressWarnings(suppressMessages(library(aws.signature)))
suppressWarnings(suppressMessages(library(tm)))
suppressWarnings(suppressMessages(library(qdap)))
suppressWarnings(suppressMessages(library(SnowballC)))
suppressWarnings(suppressMessages(library(wordcloud)))
suppressWarnings(suppressMessages(library(topicmodels)))
suppressWarnings(suppressMessages(library(data.table)))
suppressWarnings(suppressMessages(library(tidytext)))
suppressWarnings(suppressMessages(library(RNewsflow)))
suppressWarnings(suppressMessages(library(portfolio)))
suppressWarnings(suppressMessages(library(jsonlite)))
suppressWarnings(suppressMessages(library(readr)))
```

Different libraries were used to access tweets that required authentication and access rights. The project also accessed to AWS to store and read data. All the API keys and tokens were saved as environmental variables that were retrieved when necessary.

Follwing codes were used in datacollection.Rmd but commented out here:

```
# api_key <- Sys.getenv('tweet_api_key') api_secret <-
# Sys.getenv('tweet_api_secret') token <-
# Sys.getenv('tweet_token') token_secret <-
# Sys.getenv('tweet_token_secret') #Create Twitter Connection
# setup_twitter_oauth(api_key, api_secret, token, token_secret)
# app <- Sys.getenv('tweet_app') consumer_key <-
# Sys.getenv('tweet_consumer_key') consumer_secret <-
# Sys.getenv('tweet_consumer_secret') twitter_token <-
# create_token( app = app, consumer_key = consumer_key,
# consumer_secret = consumer_secret)
```

Tweet Analysis of Howard County, Maryland

Using the function lookup_coords in the library 'rtweet' bounding box coordinates of Howard county was collected. The coordinates would be used to filter tweets to find county specific tweets only. Most frequently used twitter accounts by County government were collected from the Howard County website (https://www.howardcountymd.gov/)

Follwing codes were used in datacollection.Rmd but commented out here:

```
# HCcoord <- lookup_coords('Howard County, MD', 'country:US')
# HowardCounty_accounts <-
```

```r
# c('HoCoGov','HoCoGovExec','HCPDNews','HCDFRS','HC_JonWeinstein','HoCoBOEMaryland','JenTerrasa')
```

Government twitter accounts were then used to find the associated twitter users and their followers (i.e.the citizens who have interests in government tweets)

The first four statements were used in datacollection.Rmd but commented out here:

```r
# hcUsers <- lookupUsers(HowardCounty_accounts) HCfollowers <-
# lapply(hcUsers,function(x) { usr <- x; followersCount(usr) })
# HCfollowersDF <- as.data.frame(HCfollowers)
# write.csv(HCfollowersDF, file = 'HCfollowersDF.csv')

HCfollowersDF <- read.csv(file = "HCfollowersDF.csv", header = TRUE,
    sep = ",", stringsAsFactors = FALSE)

Gov_users <- colnames(HCfollowersDF)
Gov_users <- Gov_users[-1]
followers_count <- as.numeric(as.vector(HCfollowersDF[1, ]))
followers_count <- followers_count[-1]
HCfollowersDF <- data.frame(Gov_users = Gov_users, followers_count = followers_count)
Total_Follower <- sum(HCfollowersDF$followers_count)

HCfollowersDF
```

```
##         Gov_users followers_count
## 1         HoCoGov           12960
## 2     HoCoGovExec            3142
## 3        HCPDNews           99536
## 4          HCDFRS           14614
## 5 HC_JonWeinstein            1311
## 6 HoCoBOEMaryland             366
## 7      JenTerrasa            1351
```

```r
ggplot(HCfollowersDF, aes(x = Gov_users, y = followers_count, fill = Gov_users)) +
    geom_bar(stat = "identity") + theme(axis.text.x = element_blank(),
    plot.title = element_text(size = 12, color = "blue", hjust = 0.5)) +
    ggtitle("Number of tweet followers by county accounts")
```

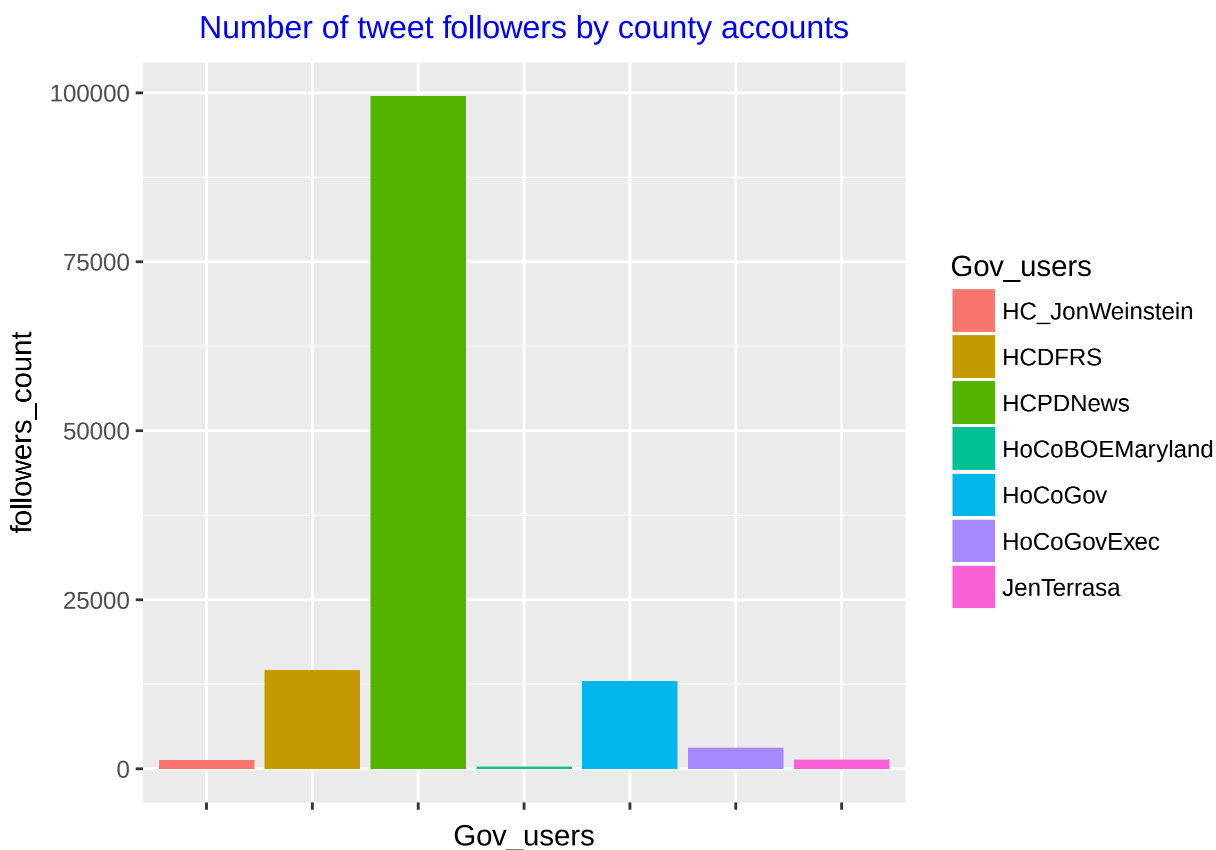


Figure 1.

While the total number (133,166) of Howard County followers are impressive compared to the County population (313,414), a closer look at the data shows that the Police Department (HCPDNews) is an outlier with 99,438 followers. So on the surface it might seem citizens pay close attention to their government while they are concerned about a specfic agency that deals with crime, safety and traffic control. Figure-1 shows the number of follwers following each county accounts.

Further Analysis

Functions were created to collect and evaluate citizens' tweets within the government. The first function "getGov_tweets" takes government accounts (government users) as its parameter and collect the recents tweets sent out by each of those government accounts. It returns all those tweets in a data frame.
the second function "FindHashtags" take the output of the "getGov_tweets" function as its parameter and check all the hashtags used by government accounts. It returns the most common hashtags used by the government. All the hashtags are stored in a character variable seperated by "OR" so that they can be used to search tweets as a query parameter.

Below two functions were used in datacollection.Rmd but also included here for reference:

```
getGov_tweets <- function(x) {
    gdf <- c()
    for (usr in x) {
        gvt <- userTimeline(x[1], n = 150)
        gvdf <- twListToDF(gvt)
        gdf <- rbind(gdf, gvdf)
```

```
    }
    return(gdf)

}


FindHashtags <- function(x) {
    all_hashtags <- str_extract_all(x$text, "#\\w+")
    DF <- as.data.frame(table(tolower(unlist(all_hashtags))))
    mostUsedHashTags <- as.character(DF[order(-DF$Freq)[1:4], 1])
    mostUsedHashTags <- mostUsedHashTags[!is.na(mostUsedHashTags)]
    mostUsed_HashTags <- paste(mostUsedHashTags, sep = "", collapse = " OR ")

    return(mostUsed_HashTags)
}
```

Tweets sent by Howard County, MD and its citizens:

search_tweets function of rtweet library was used to collect the citizen tweets. In order to select the tweets that were possibly generated as responds/reactions to government tweets, the most recent common hashtags used by the Howard County government and to control the citizen locations, the bounding box (coordinates of opposite corner points of the rectangle that contains the county polygon) of the County were used as query parameter. user_data function returned the users information of all the tweets. The citizens tweets were seperated from government tweets by comparing the users_id of the tweets.

Government tweets at a glance:

The first two statements were used in datacollection.Rmd but commented out here:

```
# HCgov_tweetDF <- getGov_tweets(hcUsers) write.csv(HCgov_tweetDF,
# file='HCgov_tweetDF.csv')

HCgov_tweetDF <- read.csv("HCgov_tweetDF.csv")
HC_retweet_count <- sum(HCgov_tweetDF$retweetCount)
HC_tweets_retwweted <- nrow(filter(HCgov_tweetDF, !HCgov_tweetDF$retweetCount ==
    0))

HC_favorite_count <- sum(HCgov_tweetDF$favoriteCount)
HC_tweets_favorited <- nrow(filter(HCgov_tweetDF, !HCgov_tweetDF$favoriteCount ==
    0))

total_count <- nrow(HCgov_tweetDF)

category <- c("total tweet", "retweet_count", "retweeted_tweet", "favorite_count",
    "favorited_tweet")
tweet_count <- c(total_count, HC_retweet_count, HC_tweets_retwweted,
    HC_favorite_count, HC_tweets_favorited)
id <- c(1:5)
likedTweetDF <- data.frame(id, category, tweet_count)

ggplot(likedTweetDF, aes(x = category, y = tweet_count, fill = category)) +
    geom_bar(stat = "identity") + geom_text(aes(label = tweet_count),
```

```
    vjust = 1.6, color = "white", position = position_dodge(0.9),
    size = 3.5) + scale_fill_brewer(palette = "Paired") + theme(axis.text.x = element_blank(),
    plot.title = element_text(size = 12, color = "blue", hjust = 0.5)) +
    ggtitle("Number of tweets that were retweeted or liked  \n and the counts of retweet and liked (favo
```

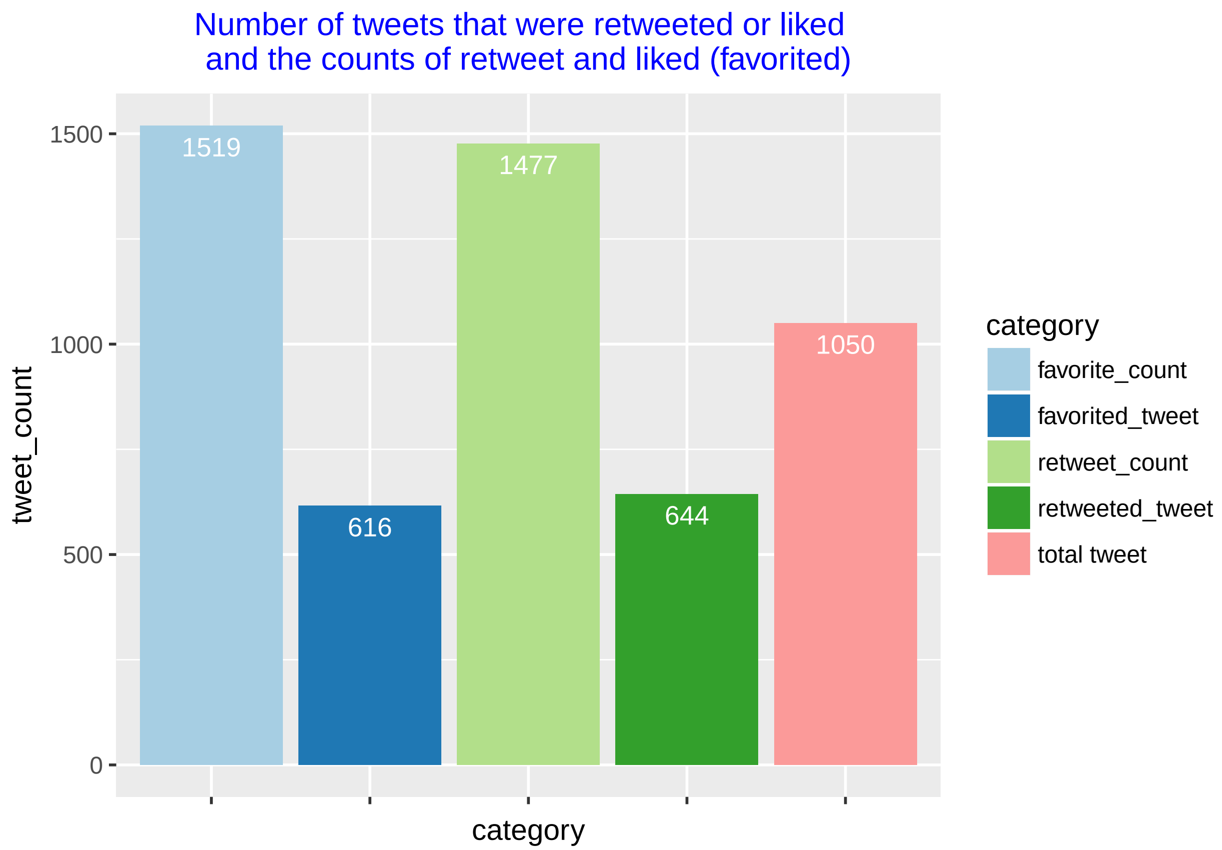


Figure 2

Above figure (Figure 2) shows a good number of government tweets were liked and retweeted. Out of 1050 original tweets, 644 and 616 were retweeted and favorited a total of 1477 and 1519 times respectively. The statisctics here suggest a good response to government tweets.

Frequency of government tweets:

```
HCgov_tweetDF$created <- as.Date.character(HCgov_tweetDF$created)
ggplot(HCgov_tweetDF, aes(x = created, fill = "red", col = "blue",
    alpha = 0.2)) + geom_histogram(position = "identity", bins = 20,
    show.legend = FALSE) + theme(plot.title = element_text(size = 12,
    color = "blue", hjust = 0.5)) + ggtitle("Frequency of government tweets")
```

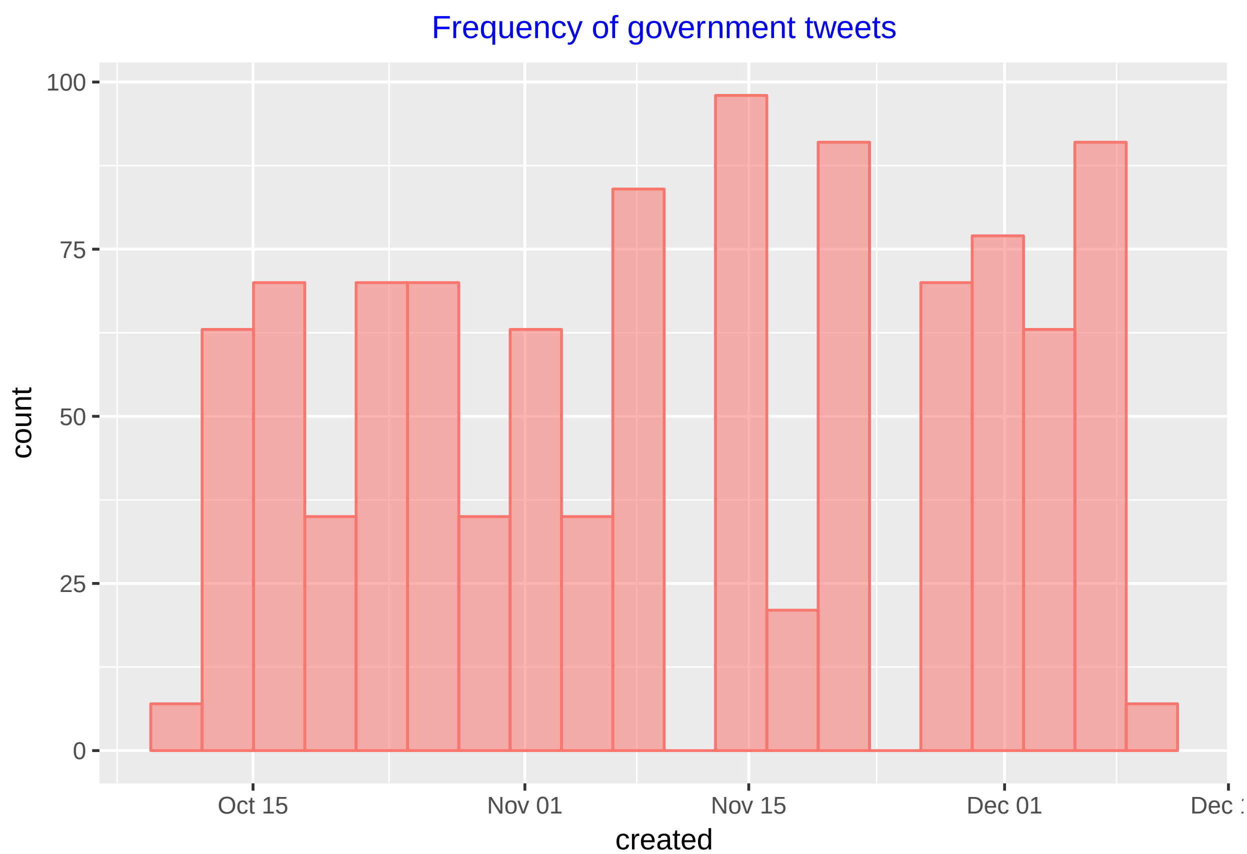


Figure 3.

Figure-3 shows that tweet frequencies i.e. tweets sent out by the county government every week are almost consistent.

collecting and evaluating citizen tweets:

Howrd county general tweets and citizen user-ids were collected using the below statements in datacollection.Rmd, which were commented out here:

```
hocogov_hashtags = FindHashtags(HCgov_tweetDF)
print(hocogov_hashtags)
```

```
## [1] "#hocomd OR #hocopolice OR #columbiamd OR #ellicottcitymd"
```

```
# HowardCounty_genTweets <- search_tweets( hocogov_hashtags,
# n=2000, token=twitter_token, type = 'mixed' )

HowardCounty_genTweets <- read.csv(file = "HCgen_tweetDF.csv", header = TRUE,
    sep = ",", stringsAsFactors = FALSE)


# HCgovUsersid <- sapply(hcUsers,function(x) x$id ) HCcitizens <-
# users_data(HowardCounty_genTweets) HCcitizens <-
# HCcitizens[!HCcitizens$location=='',] HCcitizens <-
# HCcitizens[!HCcitizens$user_id %in% HCgovUsersid,]
```

```
HCcitizens <- read.csv(file = "HCcitizens.csv", header = TRUE, sep = ",",
    stringsAsFactors = FALSE)
```

Connecting systems in real time:

The intention of the project was also to be able to share data with other systems, particularly with GIS so that various spatial analysis could be done with the tweet data. Two seperate cloud based systems were explored. Tweet data with location information were direcly stored to AWS (Amazon Web Service), which were consumed by ArcGIS online (an ESRI based cloud GIS) in order to analyze and visualuize data spatially in conjunction with other spatial data. Thus, all the changes could be updated and reflected across the systems real or near real time.

While it was possible to geocode data in ESRI platform, the geocode capability of 'dismo' library was experimented with 'geocode' function, which uses Google API. Note that the geocode operation here was limited due to the restrictions on free version of Google API.

The following geocode operations were done in datacollection.Rmd but commented out here:

```
# locations <- geocode(HCcitizens$location) locations <-
# na.omit(locations) locations <- filter(locations,
# !locations$longitude < -77.18711 & !locations$longitude >
# -76.69732) write.csv(locations, file='locate2.csv')
```

The mapping capabilities in R (ggplot2) was also experimented, which was found to be very limited (see the commented out code snippet that was found in 'https://gist.github.com/dsparks/4329876')

```
## MAPPING in R with(locations, plot(longitude, latitude)) worldMap
## <- map_data('county','maryland', wrap = TRUE)

# zp1 <- ggplot(worldMap) zp1 <- zp1 + geom_path(aes(x = long, y =
# lat, group = group), # Draw map colour = gray(2/3), lwd = 1/3)
# zp1 <- zp1 + geom_point(data = locations, # Add points
# indicating users aes(x = locations$longitude, y =
# locations$latitude), colour = 'RED', alpha = 1/2, size = 1) zp1
# <- zp1 + coord_equal() # Better projections are left for a
# future post zp1 <- zp1 + theme_minimal() # Drop background
# annotations print(zp1)
```

Exporting data into AWS (using 'aws.s3'library), the file can be accessed by the following link: https://s3.amazonaws.com/khdata/locate.csv The below HTML snippet can be used to view the map that was created based on locations data that was exported to AWS:

```
# <style>.embed-container {position: relative; padding-bottom:
# 75%; height: 0; max-width: 100%;} .embed-container iframe,
# .embed-container object, .embed-container iframe{position:
# absolute; top: 0; left: 0; width: 100%; height: 100%;}
# small{position: absolute; z-index: 40; bottom: 0; margin-bottom:
# -15px;}</style><div class='embed-container'><iframe width='400'
# height='300' frameborder='0' scrolling='no' marginheight='0'
# marginwidth='0' title='data607'
# src='//data607.maps.arcgis.com/apps/Embed/index.html?webmap=592b2fa442044589aacad05f7aafa313&amp;exte
```

the map can also be accessed by the below link: https://arcg.is/1ebujL

The following statements were used in datacollection.Rmd to stored data in AWS but commented out here:

```
b <- get_bucket("khdata")
# s3write_using(locations,FUN = write.csv, object = 'locate.csv',
# bucket = b )
```

geocoded data is also used to further filter the citizen users to make sure that the location of the users are in fact in and aroud Howard County and the tweets were originated by the Howard County residents and/or stake holders:

```
obj <- get_object(object = "locate.csv", bucket = b)  # getting data from AWS
locations <- read.csv(text = rawToChar(obj))

HCcitizens <- HCcitizens[HCcitizens$location %in% locations$originalPlace,
    ]


HowardCounty_citizensTweets <- HowardCounty_genTweets[HowardCounty_genTweets$user_id %in%
    HCcitizens$user_id, ]
```

Frequency of citizen tweets:

```
HowardCounty_citizensTweets$created_at <- as.Date(HowardCounty_citizensTweets$created_at)
ggplot(HowardCounty_citizensTweets, aes(x = created_at, fill = "green",
    col = "blue", alpha = 0.2)) + geom_histogram(position = "identity",
    bins = 20, show.legend = FALSE) + theme(plot.title = element_text(size = 12,
    color = "blue", hjust = 0.5)) + ggtitle("Frequency of citizen tweets")
```

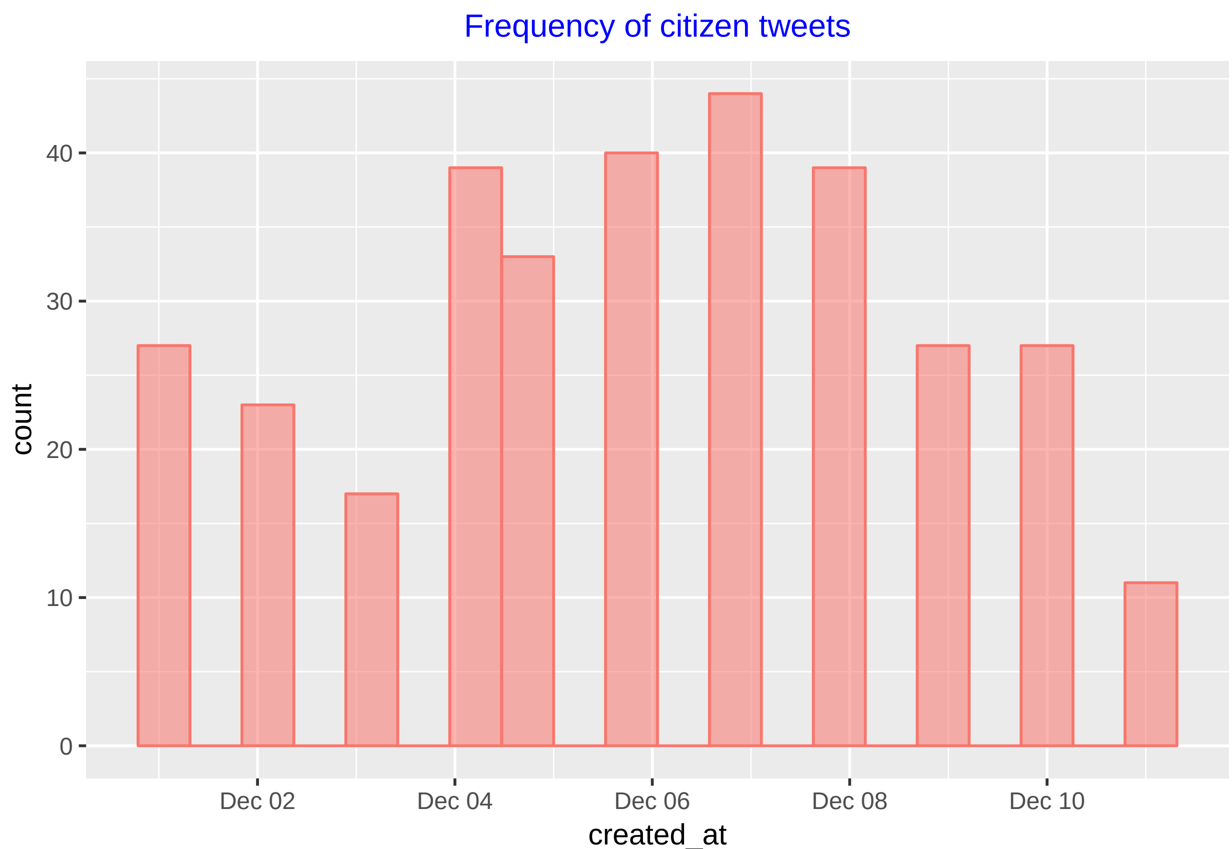


Figure 4.

Figure-4 shows that tweet frequencies of citizens varies a lot as oppose to the consistent nature government tweet frequencies.

Tweet Text mining

Texts were analyzed to see if similar terms are common in both government and citizens tweets. in other word texts were explored to examine if the concerns and interests of citizens match with what government wanted to talk about, or how much of the concerns of the both groups overlapped.

A function was created clean a Corpus that would be created with tweet texts:

```
cleanCorp <- function(corp) {


    corp <- tm_map(corp, str_replace_all, "<[^>]+>", "")
    corp <- tm_map(corp, str_replace_all, "@\\w+", "")
    corp <- tm_map(corp, str_replace_all, "#\\w+", "")
    corp <- tm_map(corp, str_replace_all, "http\\w+", "")
    corp <- tm_map(corp, content_transformer(removePunctuation))

    # since in tweet people tend to abbrebriate and symbolize texts
    # the following three functions were used from qdab library
    corp <- tm_map(corp, content_transformer(replace_abbreviation))
    corp <- tm_map(corp, content_transformer(replace_contraction))
    corp <- tm_map(corp, content_transformer(replace_symbol))
    corp <- tm_map(corp, removeNumbers)

    corp <- tm_map(corp, content_transformer(tolower))
    corp <- tm_map(corp, PlainTextDocument)
    corp <- tm_map(corp, stripWhitespace)
    corp <- tm_map(corp, str_replace_all, "^ ", "")
    corp <- tm_map(corp, str_replace_all, " $", "")

    # corp <- tm_map(corp, content_transformer(stemDocument))
    corp <- tm_map(corp, removeWords, stopwords("english"))
    return(corp)
}




HCgov_corpus <- Corpus(VectorSource(HCgov_tweetDF$text))
HCgov_corpus <- cleanCorp(HCgov_corpus)

HCcitizen_corpus <- Corpus(VectorSource(HowardCounty_citizensTweets$text))
HCcitizen_corpus <- cleanCorp(HCcitizen_corpus)


HCgov_corpus <- tm_map(HCgov_corpus, str_replace_all, "md|pm|am",
    "")
HCgov_corpus <- tm_map(HCgov_corpus, removeWords, stopwords("english"))
frequent_terms <- freq_terms(HCgov_corpus$content, 30)
plot(frequent_terms)
```

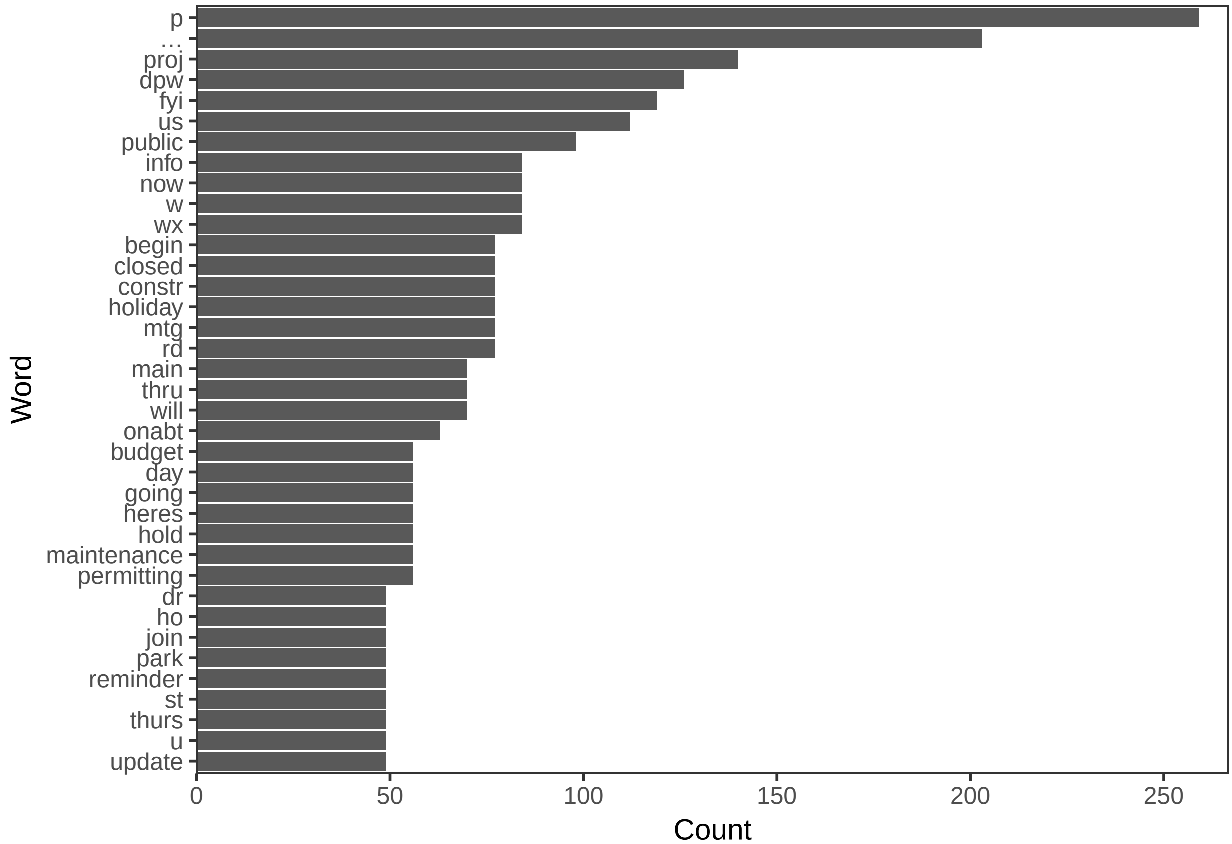


Figure 5.

```
HCcitizen_corpus <- tm_map(HCcitizen_corpus, str_replace_all, "md|pm|am",
    "")
HCcitizen_corpus <- tm_map(HCcitizen_corpus, removeWords, stopwords("english"))
citizen_frequent_terms <- freq_terms(HCcitizen_corpus$content, 30)
plot(citizen_frequent_terms)
```

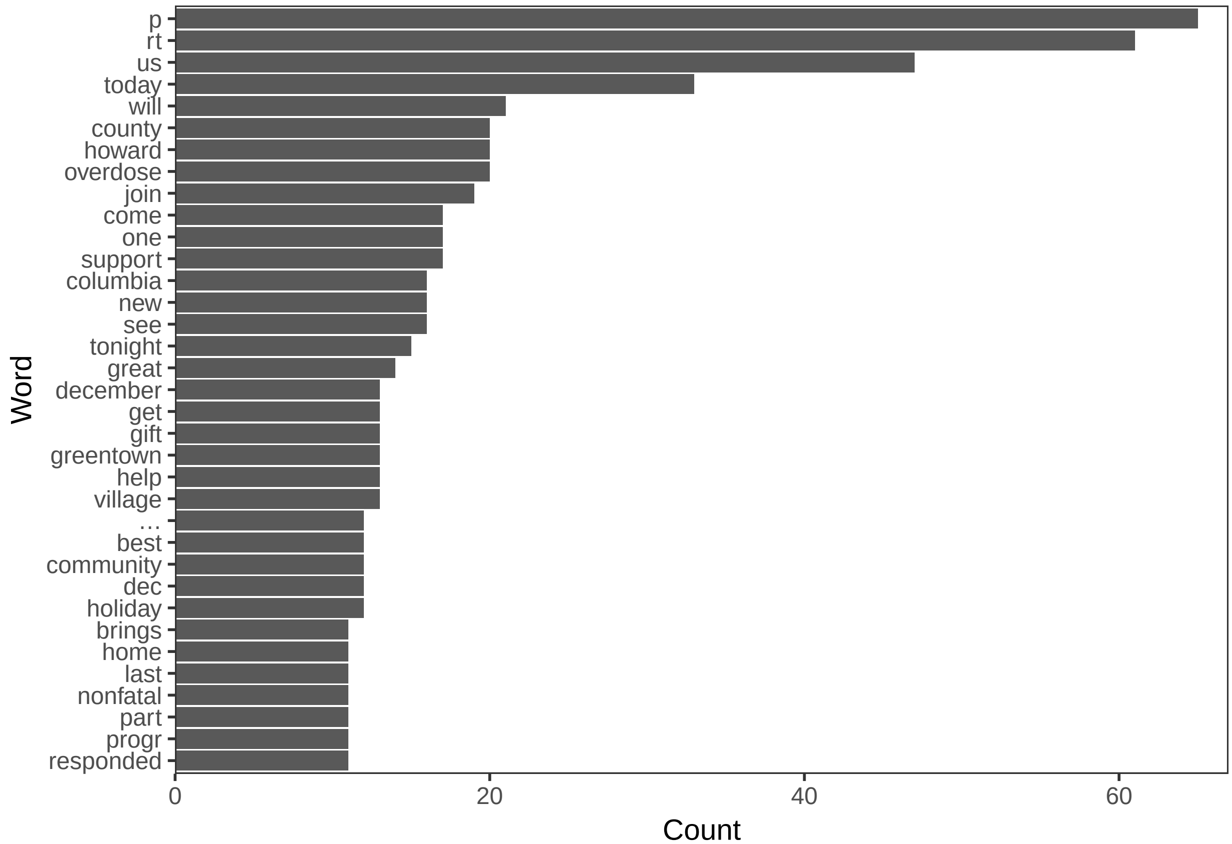


Figure 6.

Above two plots (Figure 5 and 6) show 30 most frequently used terms in tweets sent out by the government and the citizens. No significant match are seen between these two sets of words (terms), which suggest a very low overlapping of common discussions between citizens and governments.

Creating Term-document Matrix:

```
HCgov_tdm <- TermDocumentMatrix(HCgov_corpus)
HCcitizen_tdm <- TermDocumentMatrix(HCcitizen_corpus)



HCgov_tdm <- removeSparseTerms(HCgov_tdm, 0.99)
HCcitizen_tdm <- removeSparseTerms(HCcitizen_tdm, 0.99)

print(HCgov_tdm)
```

```
## <<TermDocumentMatrix (terms: 223, documents: 1050)>>
## Non-/sparse entries: 5600/228550
## Sparsity           : 98%
## Maximal term length: 12
## Weighting          : term frequency (tf)
```

```
print(HCcitizen_tdm)
```

```
## <<TermDocumentMatrix (terms: 191, documents: 327)>>
```

```
## Non-/sparse entries: 1291/61166
## Sparsity           : 98%
## Maximal term length: 14
## Weighting          : term frequency (tf)
```

Evaluation through Dendrograms:

Dendrograms were drawn for both government and citizens to see if they provide any interesting insights by creating clusteres based on word similarities.

```
drawDendogram <- function(x) {
    df <- as.data.frame(inspect(x))
    df_scale <- scale(df)
    d <- dist(df_scale, method = "euclidean")
    fit <- hclust(d, method = "ward.D2")
    return(fit)
}
```

Dendrogram for Howard County government tweets

```
HCgovDendo <- drawDendogram(HCgov_tdm)
```

```
## <<TermDocumentMatrix (terms: 223, documents: 1050)>>
## Non-/sparse entries: 5600/228550
## Sparsity           : 98%
## Maximal term length: 12
## Weighting          : term frequency (tf)
## Sample             :
##          Docs
## Terms      1015 112 115 265 415 565 715 76 865 89
##   begin       0   0   0   0   0   0   0  0   0  0
##   closed      0   0   0   0   0   0   0  0   0  0
##   constr      0   0   0   0   0   0   0  0   0  0
##   dpw         1   0   1   1   1   1   1  1   1  1
##   fyi         0   0   0   0   0   0   0  0   0  0
##   holiday     0   0   0   0   0   0   0  0   0  0
##   info        1   1   1   1   1   1   1  1   1  1
##   now         1   0   1   1   1   1   1  1   1  0
##   proj        0   0   0   0   0   0   0  0   0  0
##   public      0   1   0   0   0   0   0  0   0  0
```

```
plot(HCgovDendo)
rect.hclust(HCgovDendo, k = 4)
```

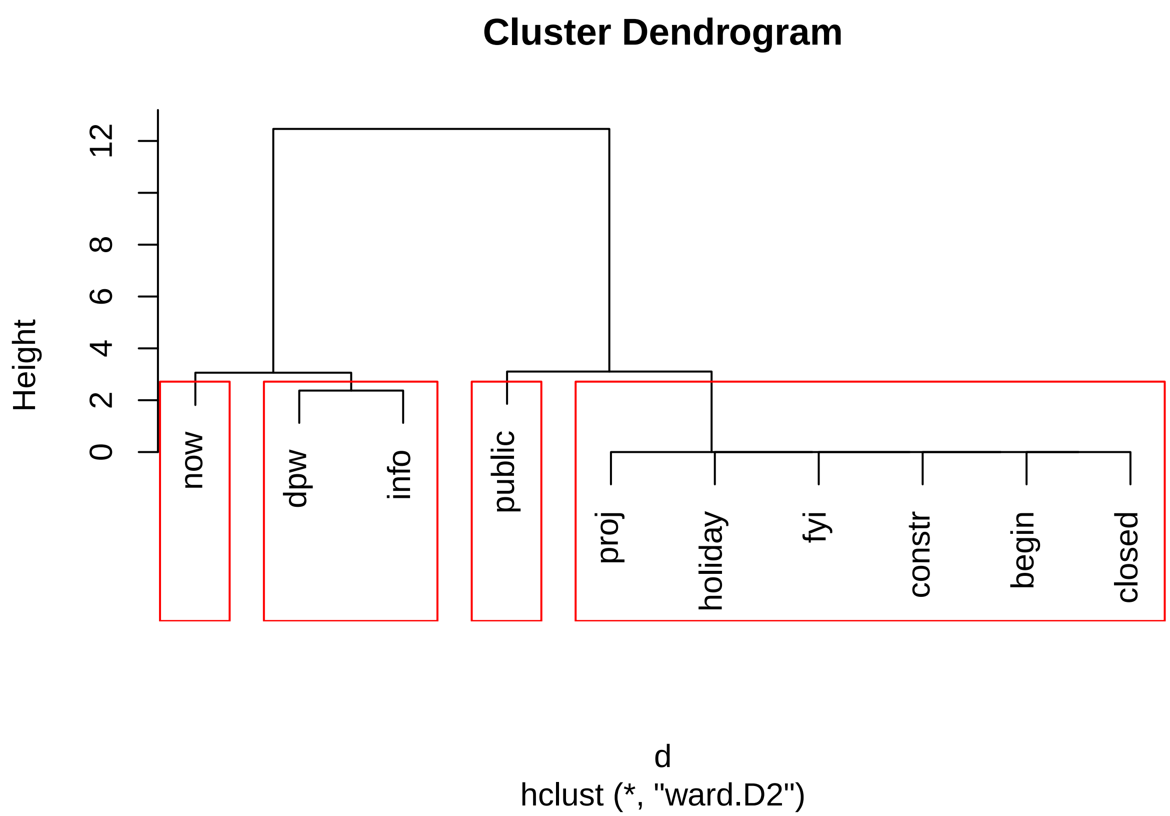


Figure 6.

Dendrogram for Howard County citizens tweets

```r
HCcitizenDendo <- drawDendogram(HCcitizen_tdm)
```

```
## <<TermDocumentMatrix (terms: 191, documents: 327)>>
## Non-/sparse entries: 1291/61166
## Sparsity           : 98%
## Maximal term length: 14
## Weighting          : term frequency (tf)
## Sample             :
##             Docs
## Terms        171 196 198 25 26 269 270 29 30 57
##   columbia     0   0   0  0  0   0   0  0  0  0
##   come         0   0   0  0  0   0   0  1  1  0
##   county       1   0   0  0  0   0   0  0  0  0
##   howard       0   0   0  0  0   0   0  0  0  0
##   join         1   0   0  1  1   1   1  1  1  0
##   one          0   0   1  0  0   1   1  0  0  0
##   overdose     0   0   0  0  0   0   0  0  0  0
##   support      1   1   0  0  0   0   0  0  0  1
##   today        0   0   1  1  1   1   1  1  1  0
##   will         0   1   0  1  1   0   0  1  1  1
```

```r
plot(HCcitizenDendo)
```

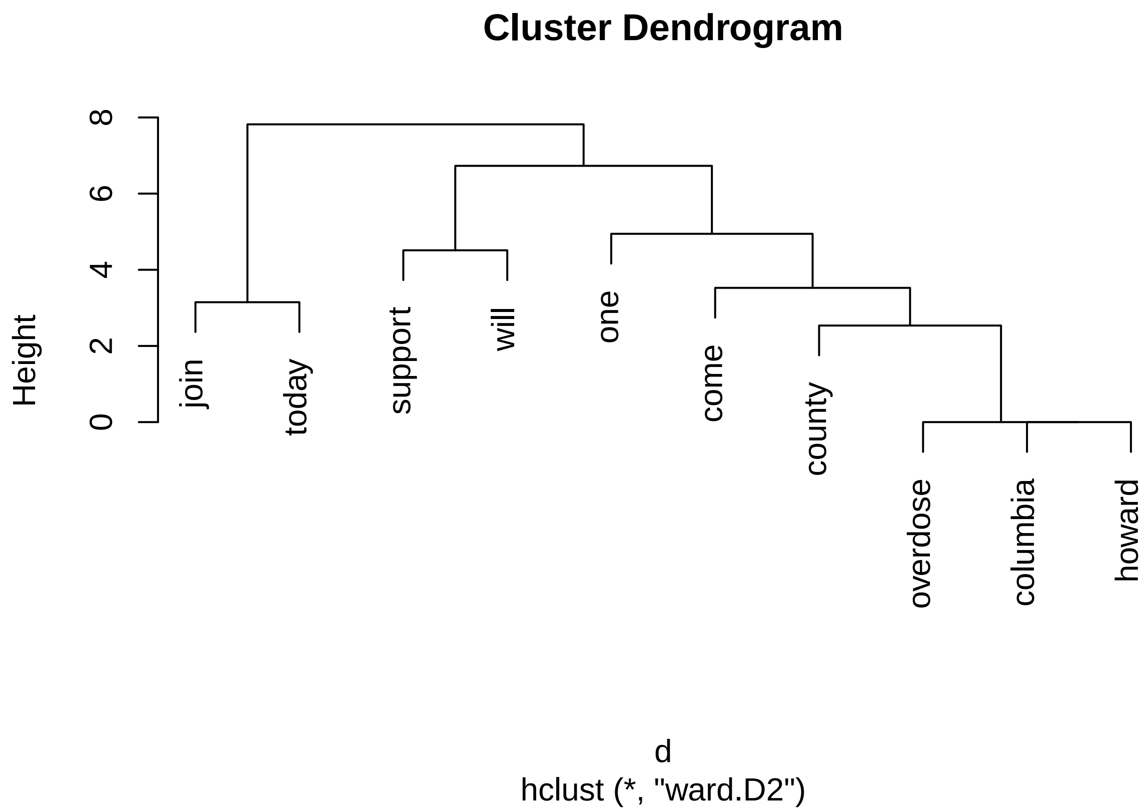


Figure 7.

The government dendogram (Figure 6) shows some association of the words that were used in their tweets. There were no association of words or no distinct clusters in the citizens tweets (Figure 7) suggesting no focused discussion on certain topics but many scattered interests.

Evaluation through Wordclouds:

In order to see the difference or commonality of interests or concerns of these two groups (government and citizens) two wordclouds were created. All the texts of each group were represented in two documents representing the government and the citizens in a common Corpus:

```
try.tolower = function(x) {
    y = NA
    try_error = tryCatch(tolower(x), error = function(e) e)
    if (!inherits(try_error, "error"))
        y = tolower(x)
    return(y)
}

HcgovText <- paste(unlist(HCgov_tweetDF$text), sep = " ", collapse = " ")
HccitizenText <- paste(unlist(HowardCounty_citizensTweets$text), sep = " ",
    collapse = " ")
HccitizenText <- sapply(HccitizenText, function(row) iconv(row, "latin1",
    "ASCII", sub = ""))

HcgovText <- sapply(HcgovText, try.tolower)
```

```r
HccitizenText <- sapply(HccitizenText, try.tolower)

# HcgovText <- paste(HcgovText, collapse=' ') HccitizenText <-
# paste(HccitizenText, collapse=' ') HccitizenText <-
# as.character(HccitizenText)
HCtexts <- c(HcgovText, HccitizenText)

HC_Corpus <- Corpus(VectorSource(HCtexts))
HC_Corpus <- cleanCorp(HC_Corpus)
HC_Corpus <- tm_map(HC_Corpus, removePunctuation)
HC_Corpus <- tm_map(HC_Corpus, content_transformer(stemDocument))

HC_tdm <- TermDocumentMatrix(HC_Corpus)
HC_tdm <- as.matrix(HC_tdm)
colnames(HC_tdm) = c("Government Tweets", "Citizent Tweets")
```

Comarison cloud:

```r
comparison.cloud(HC_tdm, colors = c("#00B2FF", "red"), title.size = 1,
    max.words = 200, scale = c(2.1, 0.49))
```

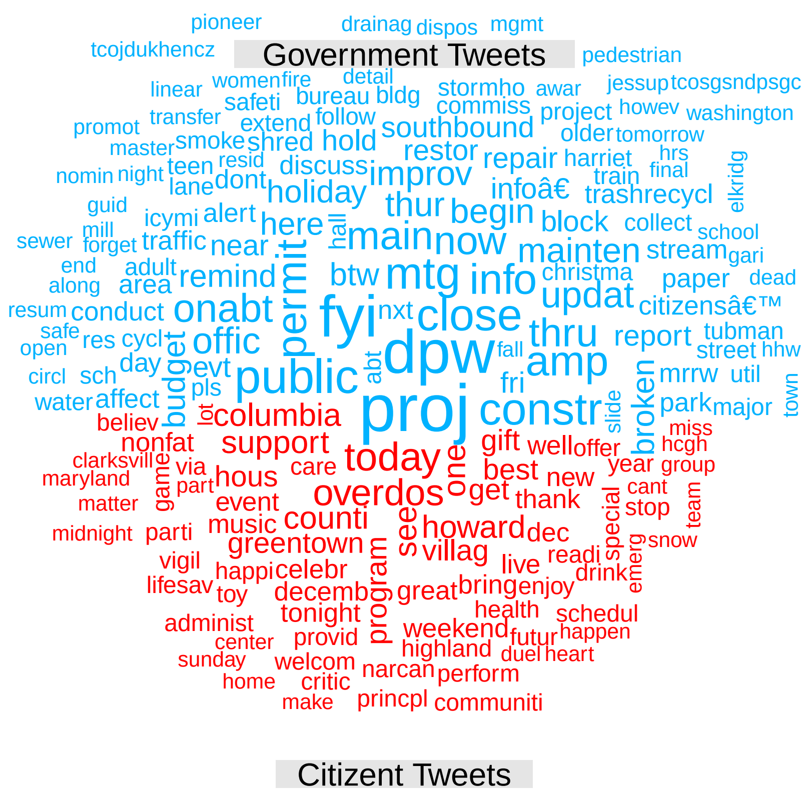


Figure 8.

The above cloud (Figure 8) suggests some relevant words concerning governmnet operations, such as project, DPW (public works), meeting, permit, construction, improvment, public, repair, maintenance etc. on the other hand citizen tweets seems very diverse and nothing really stands out i.e. no suggestion of any interaction between government and citizens, Commonality Cloud:

```r
commonality.cloud(HC_tdm, colors = brewer.pal(8, "Dark2"))
```

```
## Warning in wordcloud(rownames(term.matrix)[freq > 0], freq[freq > 0],
## min.freq = 0, : amp could not be fit on page. It will not be plotted.
```

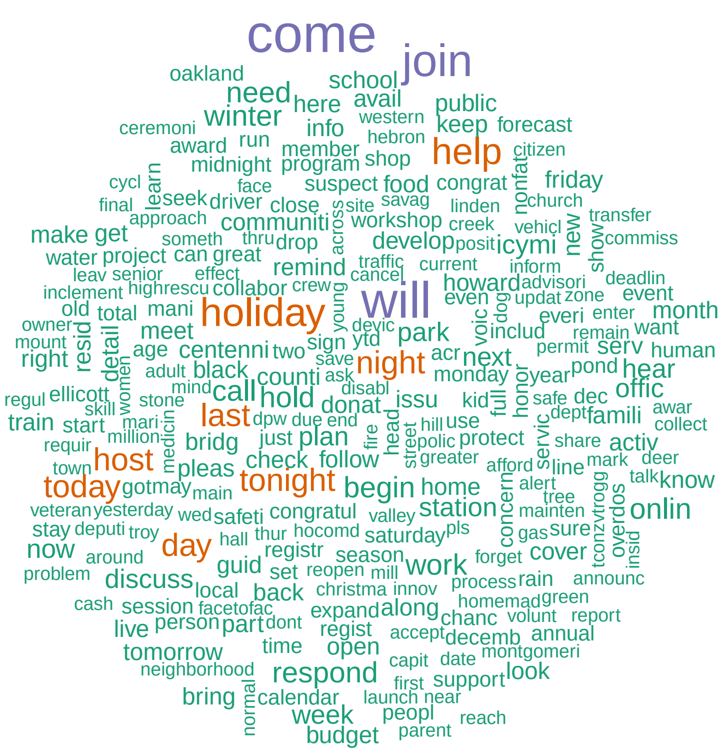


Figure 9.

The commonality cloud (Figure 9) again suggest disconnect between the two groups. The common words found in the cloud such as amp, join, holiday, tonight, work etc. are very general and does not seem to suggest any interaction between the government and the citizens.

Topic model comparison:

Both government and citizens tweet texts were grouped under five topics each, and the 10 most frequent terms related to each topics were plotted to examine if there were any similarities between the topics and terms that would suggest any interaction:

Goverment topics:

```r
HCgov_DTM <- as.DocumentTermMatrix(HCgov_tdm)

# HCgov_DTM_DS <- as.matrix(HCgov_DTM)
rowTotals <- apply(HCgov_DTM, 1, sum)  #Find the sum of words in each Document
HCgov_DTM <- HCgov_DTM[rowTotals > 0, ]
HCgov_DTM_DS <- as.matrix(HCgov_DTM)
ldamodel <- LDA(HCgov_DTM, k = 5, control = list(seed = 1500))
```

```
## topicwords <- terms(ldamodel,5) topicwords

gov_per_topic_per_word <- tidy(ldamodel, matrix = "beta")
head(gov_per_topic_per_word)
```

```
## # A tibble: 6 x 3
##   topic        term           beta
##   <int>       <chr>          <dbl>
## 1     1      budget 6.028790e-177
## 2     2      budget 1.049166e-177
## 3     3      budget  1.404210e-42
## 4     4      budget 1.501938e-172
## 5     5      budget  6.003521e-02
## 6     1 citizensâ<U+0080><U+0099> 1.195581e-176
```

```
gov_top_terms <- gov_per_topic_per_word %>% group_by(topic) %>% top_n(10,
    beta) %>% ungroup() %>% arrange(topic, -beta)

gov_top_terms %>% mutate(term = reorder(term, beta)) %>% ggplot(aes(term,
    beta, fill = factor(topic))) + geom_col(show.legend = FALSE) +
    facet_wrap(~topic, scales = "free") + coord_flip() + theme(plot.title = element_text(size = 11,
    color = "blue", hjust = 0.5)) + ggtitle("Most Frequent terms in government tweets \n catagorized und
```

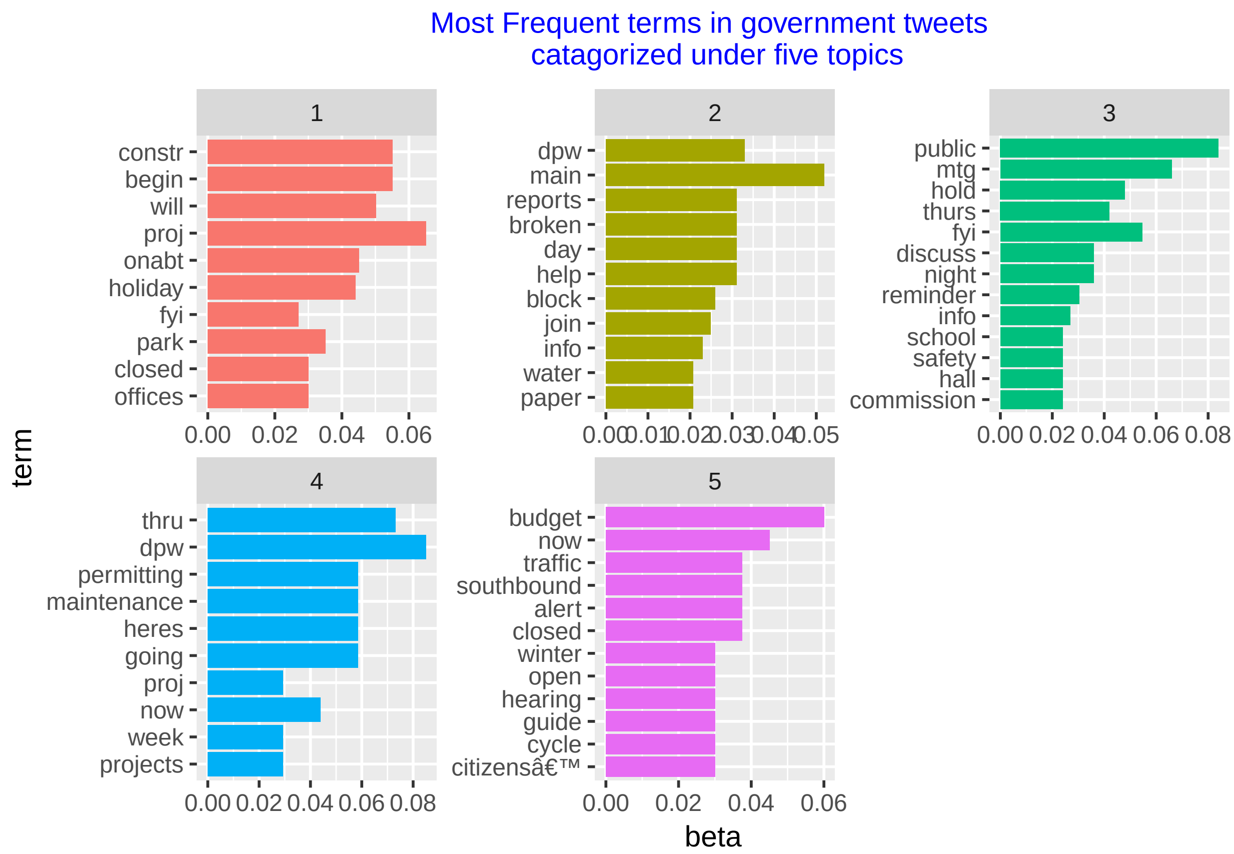


Figure 10.

Citizen topics:

```r
HCcitizen_DTM <- as.DocumentTermMatrix(HCcitizen_tdm)

# HCgov_DTM_DS <- as.matrix(HCgov_DTM)
rowTotal <- apply(HCcitizen_DTM, 1, sum)  #Find the sum of words in each Document
HCcitizen_DTM <- HCcitizen_DTM[rowTotal > 0, ]

ctznldamodel <- LDA(HCcitizen_DTM, k = 5, control = list(seed = 1500))
## topicwords <- terms(ldamodel,5) topicwords

ctzn_per_topic_per_word <- tidy(ctznldamodel, matrix = "beta")
head(ctzn_per_topic_per_word)
```

```
## # A tibble: 6 x 3
##   topic     term         beta
##   <int>    <chr>        <dbl>
## 1     1 columbia 1.011136e-10
## 2     2 columbia 1.826983e-04
## 3     3 columbia 5.346606e-02
## 4     4 columbia 2.048270e-21
## 5     5 columbia 1.519486e-23
## 6     1     soon 1.336023e-02
```

```r
ctzn_top_terms <- ctzn_per_topic_per_word %>% group_by(topic) %>%
    top_n(10, beta) %>% ungroup() %>% arrange(topic, -beta)

ctzn_top_terms %>% mutate(term = reorder(term, beta)) %>% ggplot(aes(term,
    beta, fill = factor(topic))) + geom_col(show.legend = FALSE) +
    facet_wrap(~topic, scales = "free") + coord_flip() + theme(plot.title = element_text(size = 11,
    color = "blue", hjust = 0.5)) + ggtitle("Most Frequent terms in citizen tweets \n catagorized under
```

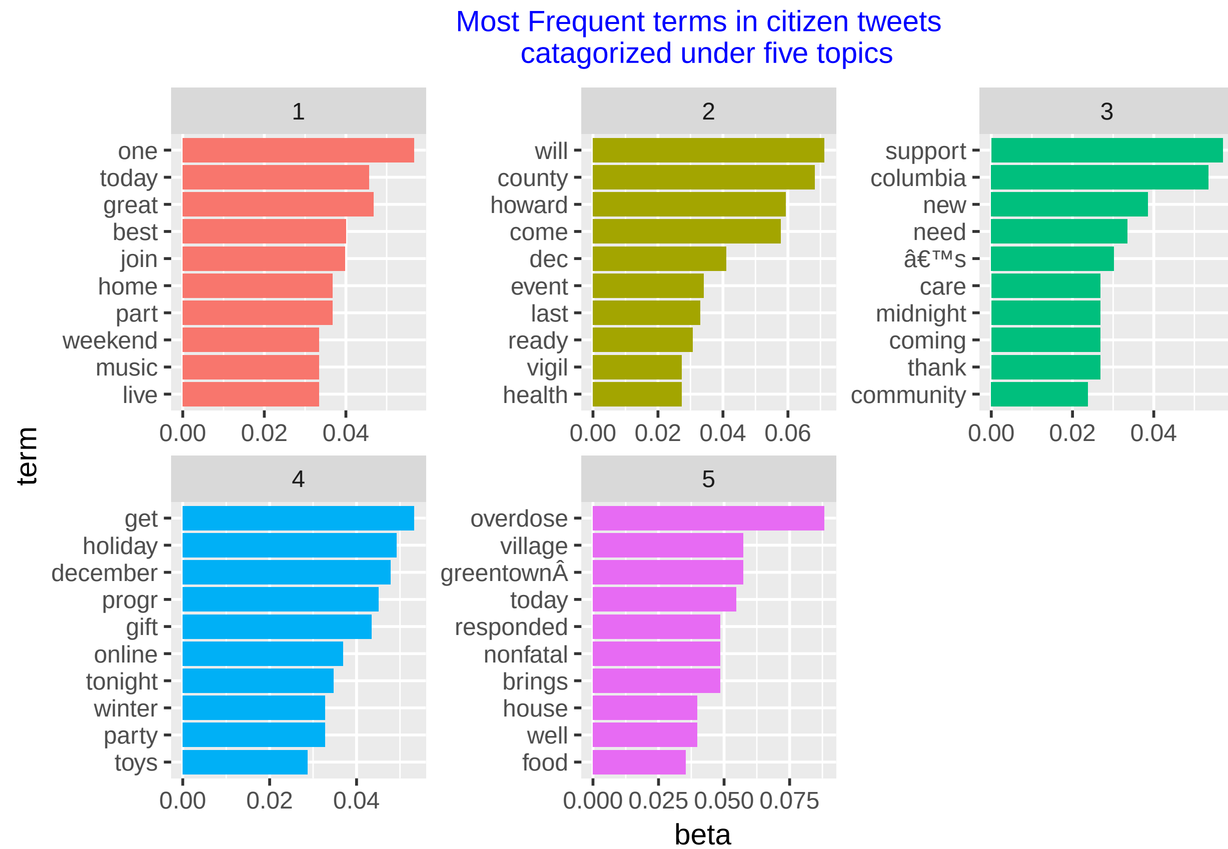


Figure 11.

All the above topics and related terms in government and citizen tweets (Figure 10 and 11) do not show any similarities re-affirm the suspicion that there is not enough citizens-government interactions through tweets.

Compare the document-term-matrix of government and citizen tweets:

```
govDTM <- DocumentTermMatrix(HCgov_corpus, control = list(weighting = weightTfIdf,
    stopwords = TRUE))
ctznDTM <- DocumentTermMatrix(HCcitizen_corpus, control = list(weighting = weightTfIdf,
    stopwords = TRUE))

doc_compare <- documents.compare(ctznDTM, govDTM, min.similarity = 0.45,
    n.topsim = NULL, return.zeros = FALSE)
```

```
## Warning in colnames(dtm) == colnames(dtm.y): longer object length is not a
## multiple of shorter object length
```

```
head(doc_compare)
```

```
##       x  y similarity
## 7   146  1  0.6628568
## 72   91  7  0.6625379
## 73   99  7  0.4957166
## 173 140 11  0.7022255
## 252 146 15  0.8417213
## 267 202 16  0.8663407
```

```r
similar <- nrow(doc_compare)
gov_documents <- nrow(govDTM)
citizen_documents <- nrow(ctznDTM)

doc_count <- c(citizen_documents, gov_documents, similar)

CompareDF <- data.frame(documents = c("citizen", "government", "similar"),
    doc_count = doc_count)

ggplot(CompareDF, aes(x = documents, y = doc_count, fill = documents)) +
    geom_bar(stat = "identity", color = "black") + scale_fill_manual(values = c("#999999",
    "#E69F00", "#56B4E9")) + geom_text(aes(label = doc_count), vjust = 1.6,
    color = "white", position = position_dodge(0.9), size = 3.5) +
    scale_fill_brewer(palette = "Paired") + theme(axis.text.x = element_blank(),
    plot.title = element_text(size = 11, color = "blue", hjust = 0.5)) +
    ggtitle("Number of government and citizen documents (tweets) \n and the number of documents with at
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill',
## which will replace the existing scale.
```

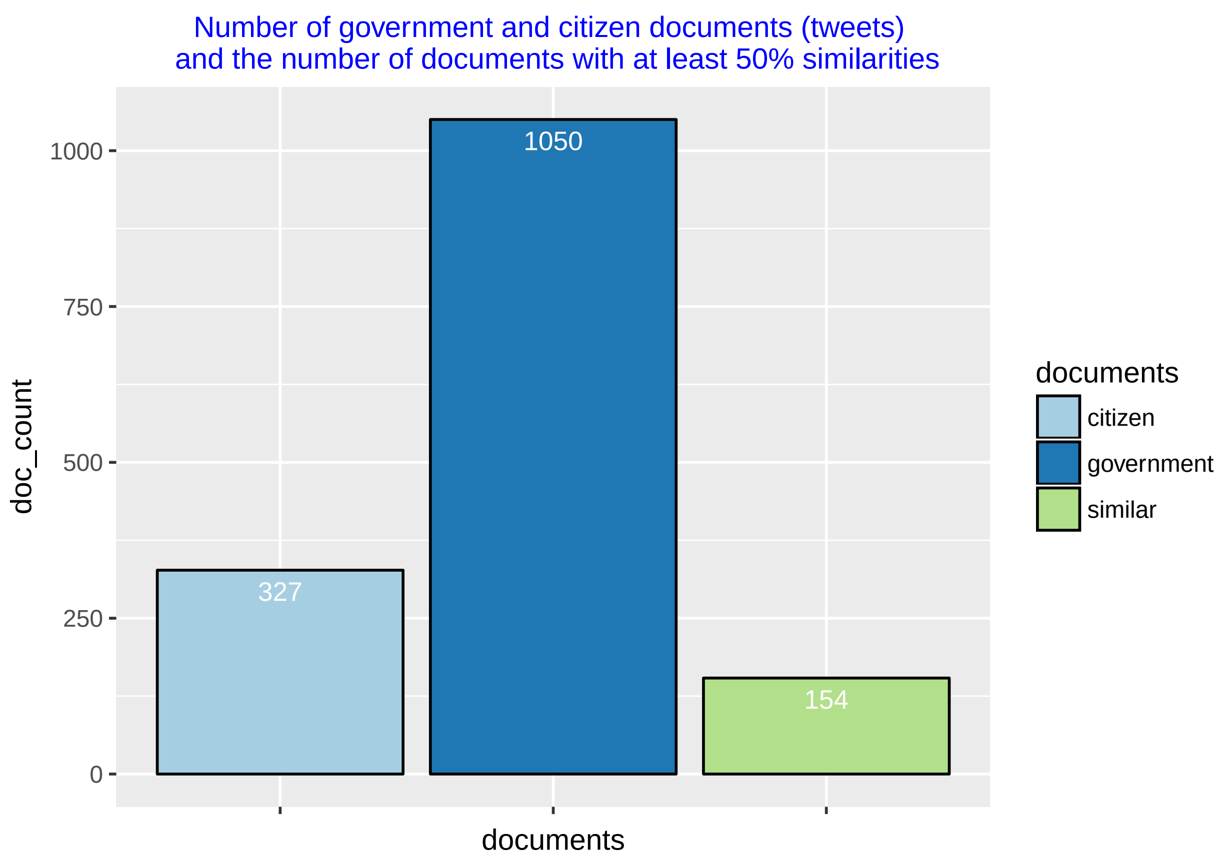


Figure 12.

While it seems significant to find 210 pair of documents have some significant similarities (see Figure 12) given the low number of citizen documents (only 351), the reason probably is that same citizen documents might have matched with multiple government documents. It could be the similarities among very general words as seen in the commonality cloud above. So the result of this document comparison process does not necessarily show any interaction between two the two groups.

Summary:

All the analysis above point to the fact that both the County government and the citizens need to take initiatives for effective communications. The number of tweets sent out the by the government and the number of totals followers they have are encouraging, which suggest both the willingness and environment are there to use social media such as tweeters for better communication between Howard County government and its citizens. The huge number of followers of Police Department means people, in general, are naturally drawn to stories or news that have quick and explicit impact on them such as a crime event or accidents. Therefore departments like Planning and zoning etc. that have significant influence on citizens' future livelihoods but are not immediately felt should be more proactive to connect to the citizens.

Reference:

1. http://rstudio-pubs-static.s3.amazonaws.com/256588_57b585da6c054349825cba46685d8464.html
2. http://tidytextmining.com/twitter.html#getting-the-data-and-distribution-of-tweets
3. https://heuristically.wordpress.com/2011/04/08/text-data-mining-twitter-r/
4. http://fredgibbs.net/tutorials/document-similarity-with-r.html
5. https://sites.google.com/site/miningtwitter/home
6. https://developer.twitter.com/en/docs/basics/getting-started
7. http://bogdanrau.com/blog/collecting-tweets-using-r-and-the-twitter-search-api/
8. https://davetang.org/muse/2013/04/06/using-the-r_twitter-package/
9. https://rstudio-pubs-static.s3.amazonaws.com/66739_c4422a1761bd4ee0b0bb8821d7780e12.html http://tidytextmining.com/topicmodeling.html

Race and ownership of residential properties

Mehdi Khan

October 8, 2017

Introduction:

This data was extracted from 2000 Census with information about race, rental and ownership of residences in Anne Arundel County, Maryland. The data was used by Anne Arundel County Community Development Services, Inc - a nonprofit agency in their housing research.

The data was downloaded as an excel file which was converted to a CSV file, which then needed to be made tidy to do analysis in R. The analysis was done to see if the data could provide any relationship in race and ownership of residential properties.

Load necessary libraries:

```
suppressMessages(suppressWarnings(library(stringr)))
suppressMessages(suppressWarnings(library(tidyr)))
suppressMessages(suppressWarnings(library(dplyr)))
suppressMessages(suppressWarnings(library(data.table)))
suppressMessages(suppressWarnings(library(ggplot2)))
```

load data:

```
houseDS <- read.csv("C:\\Temp\\race versus tenure of owner & rental housing units.csv",
    sep = ",", stringsAsFactors = FALSE, fill = TRUE)
# str(houseDS)
```

Since data is distributed under two categories of ownership (rental and owned) two datsets were created with only relevant rows and columns, all other rows and columns were avoided:

```
owned_houses <- houseDS[2:9, 1:2]
rental_houses <- houseDS[12:19, 1:2]
```

A funtion was created to do untidy operations on these two dataset, the function does the following: a. update with meaningful column names b. update race column with simplified race names c. add and populate new column with ownership information

```
untidyData <- function(dat) {
    colnames(dat) <- c("race", "population")
    dat$population <- str_replace(dat$population, ",", "")

    dat$population <- as.numeric(dat$population)
    for (i in (1:length(dat$race))) {
        if (str_detect(dat$race[i], "White")) {
            dat$race[i] <- "White"
        } else if (str_detect(dat$race[i], "Black")) {
            dat$race[i] <- "African American"
        } else if (str_detect(dat$race[i], "Indian")) {
            dat$race[i] <- "American Indian and Alaska Native"
```

```r
        } else if (str_detect(dat$race[i], "Asian")) {
            dat$race[i] <- "Asian"
        } else if (str_detect(dat$race[i], "Native")) {
            dat$race[i] <- "Native Hawaiian, Pacific Islander"
        } else if (str_detect(dat$race[i], "other race")) {
            dat$race[i] <- "Others"
        } else if (str_detect(dat$race[i], "Two or more")) {
            dat$race[i] <- " Two or more races"
        }
    }
    mutate(dat, ownership = "")
    if (str_detect(dat$race[1], "Owner") == TRUE) {
        dat$ownership <- "owner"
    } else if (str_detect(dat$race[1], "Renter")) {
        dat$ownership <- "renter"
    }

    dat <- dat[-1, ]

    ## dat$percentage=dat$population/sum(dat$population)

    return(dat)
}
owned_houses <- untidyData(owned_houses)
rental_houses <- untidyData(rental_houses)

HouseData <- rbind(owned_houses, rental_houses)

head(HouseData)
```

```
##                                 race population ownership
## 3                              White     119609     owner
## 4                   African American      11615     owner
## 5 American Indian and Alaska Native        350     owner
## 6                              Asian       1807     owner
## 7 Native Hawaiian, Pacific Islander         57     owner
## 8                             Others        337     owner
```

Some statistics:

```r
house_stat <- HouseData %>% group_by(ownership) %>% summarise(total_population = sum(population),
    max_population = max(population), race_max_occupancy = race[which.max(population)],
    min_population = min(population), race_min_occupancy = race[which.min(population)])

head(house_stat)
```

```
## # A tibble: 2 x 6
##   ownership total_population max_population race_max_occupancy
##       <chr>            <dbl>          <dbl>              <chr>
## 1     owner           134922         119609              White
## 2    renter            43748          31035              White
## # ... with 2 more variables: min_population <dbl>,
## #   race_min_occupancy <chr>
```

Figure 1:

```
ggplot(HouseData, aes(race, population)) + geom_bar(aes(fill = ownership),
    stat = "identity", position = "dodge") + labs(title = "Occupancy of residence by race and ownership
    y = "Population") + theme(axis.text.x = element_text(angle = 45,
    hjust = 1))
```

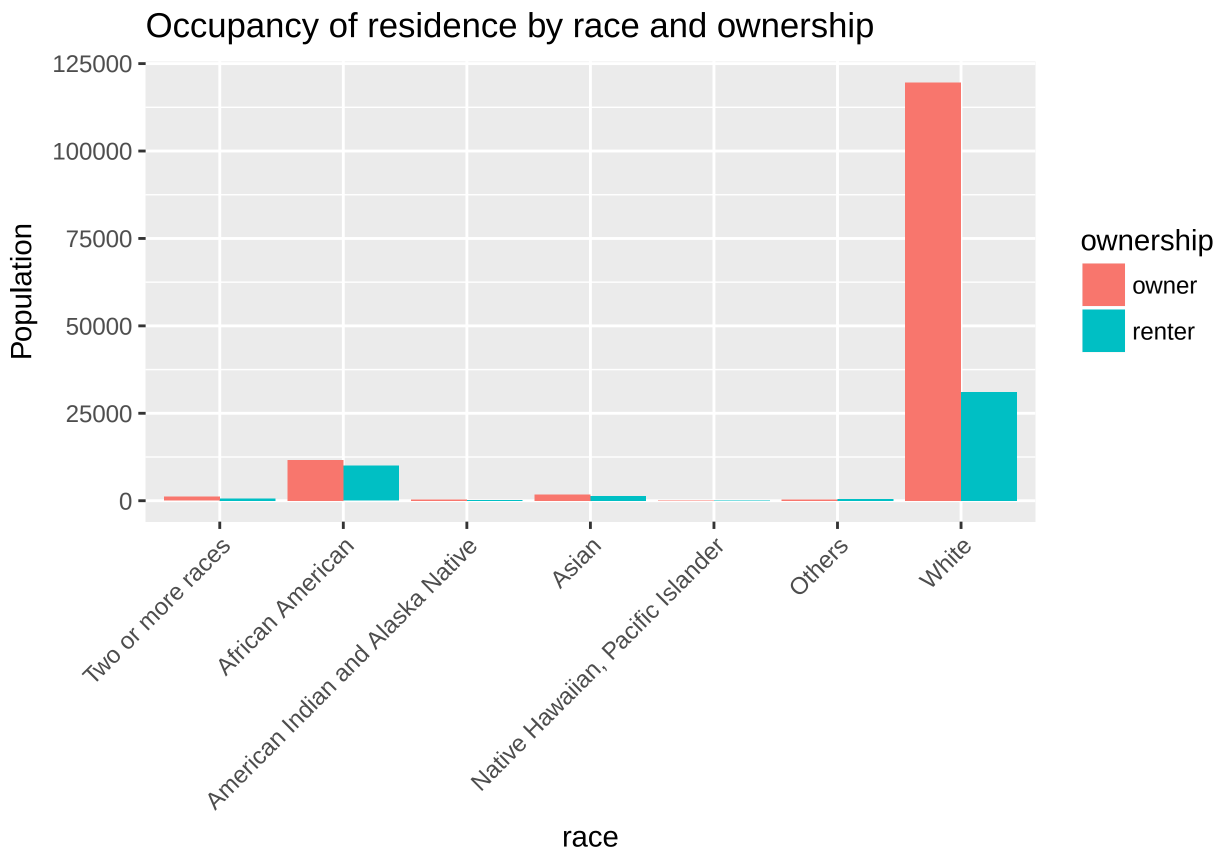


Figure 1 shows that the white population has bigger occupancy in both owned and rental properties.

spread function is used to untidy the data again:

```
HouseData2 <- spread(HouseData, 3, 2)
```

Figure 2:

```
ggplot(HouseData2, aes(race, owner/renter)) + geom_bar(aes(fill = race),
    stat = "identity", position = "dodge") + labs(title = "Occupancy of residence by race and ownership
    y = "owner renter ratio") + theme(axis.text.x = element_blank())
```

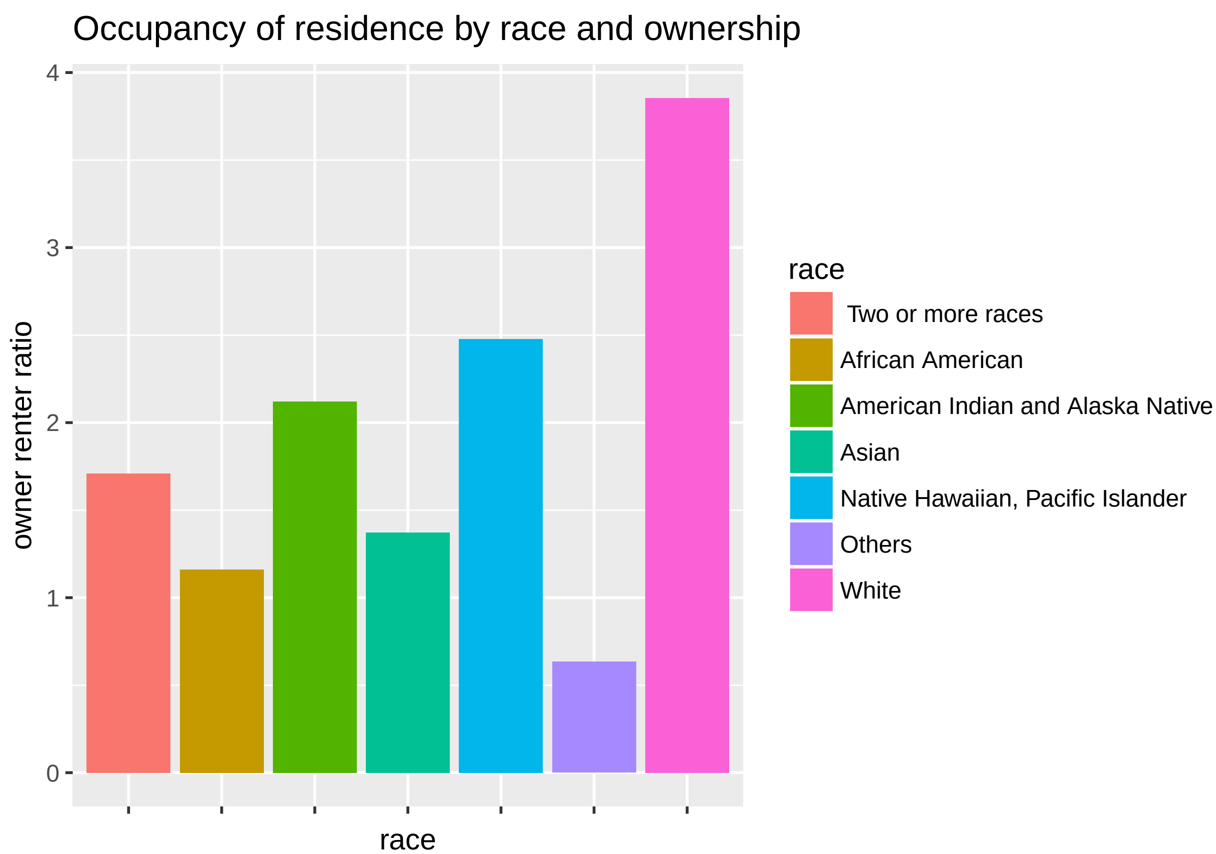


Figure 2 shows that the ratio of white population has more house ownership compared to other races.

Figure 3:

```
ggplot(HouseData2, aes(x = race, y = owner/sum(owner))) + geom_point(aes(color = race,
    size = owner)) + labs(title = "Ratio house ownership by race",
    y = "ownership ratio") + theme(axis.text.x = element_blank())
```

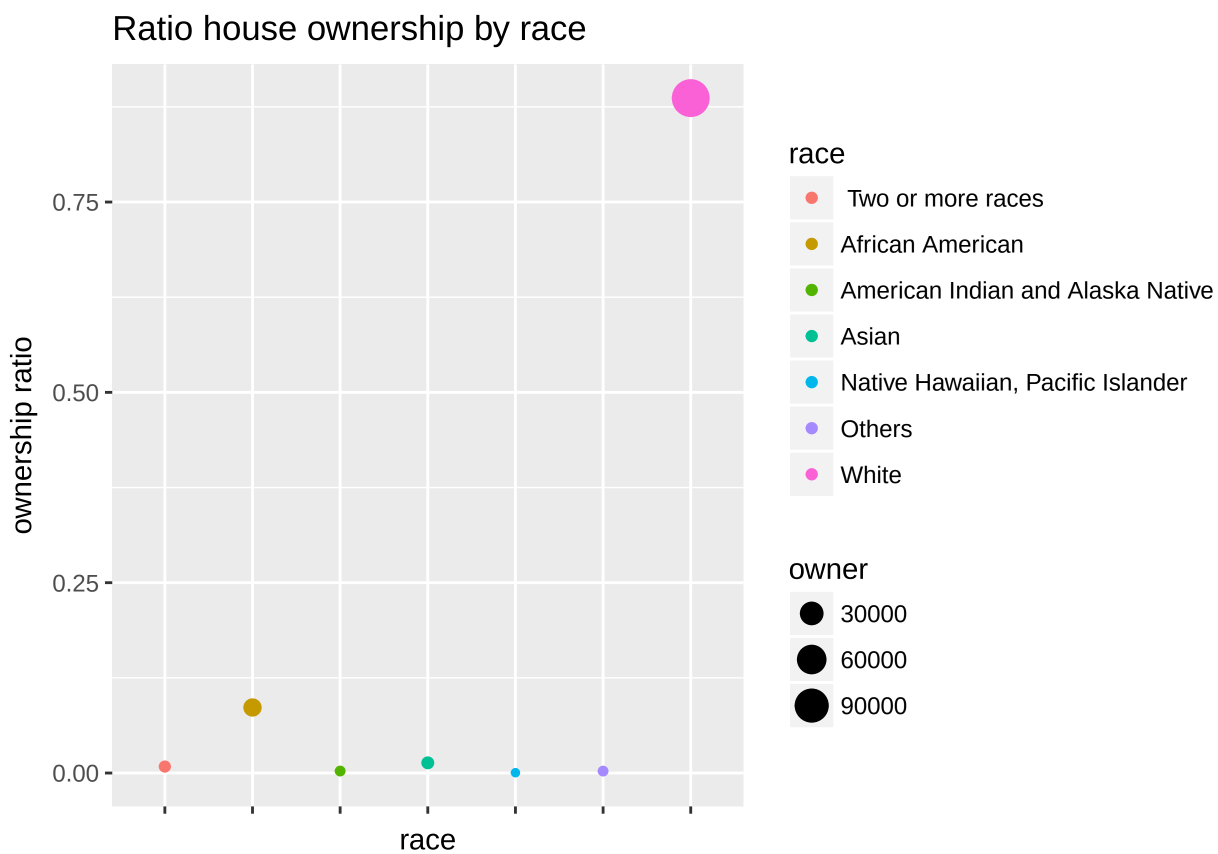


Figure 3 also suggest the white population has a bigger percentage of home ownership while african american population hold the distant second position.

Analysis of building permits and school data

Mehdi Khan

December 13, 2017

Introduction:

Building permit and school data from 2002 to 2017 have been collected to see if there is any significant relationship between the building permits approved by the county and the changes in the schools' percentage capacity (enrollment/capacity) and/or enrollment in schools. Only the completed building permits were considered here.

Research question:

Does the number of completed building permits have any impact on the percentage capacity and/or enrollment in the schools?

Result summary:

No significant relationship was found. Very weak correlations were found between building permits and other two variables (permit, percentage capacity).

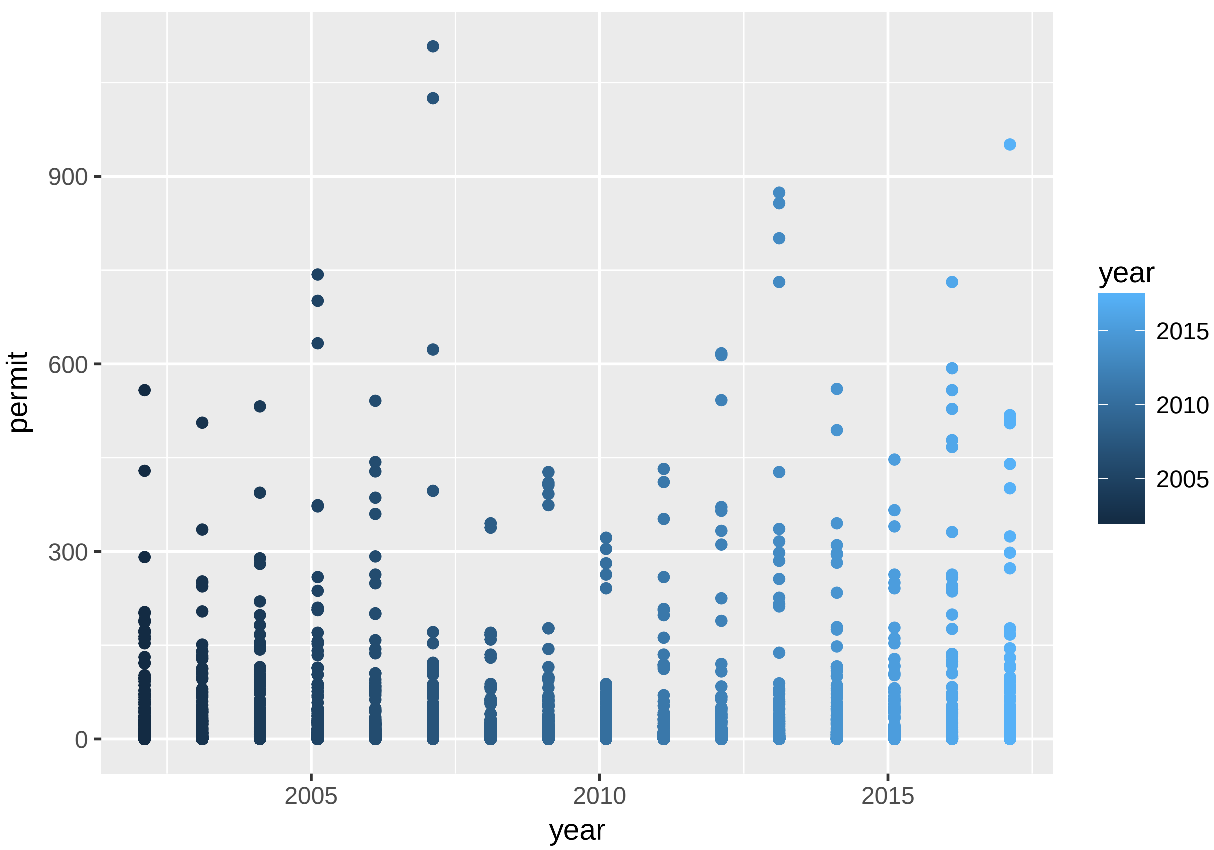
Analysis:

```r
school_info <- read.csv("newSchoolEnrollment.csv", sep = ",", stringsAsFactors = FALSE,
    header = FALSE)
fields <- as.character(as.vector(school_info[2, ]))
colnames(school_info) <- fields
schoolDS_new <- school_info[-c(1:2, 113), -c(20, 53, 54)]
schoolDS_new <- rename(schoolDS_new, school = "")
```

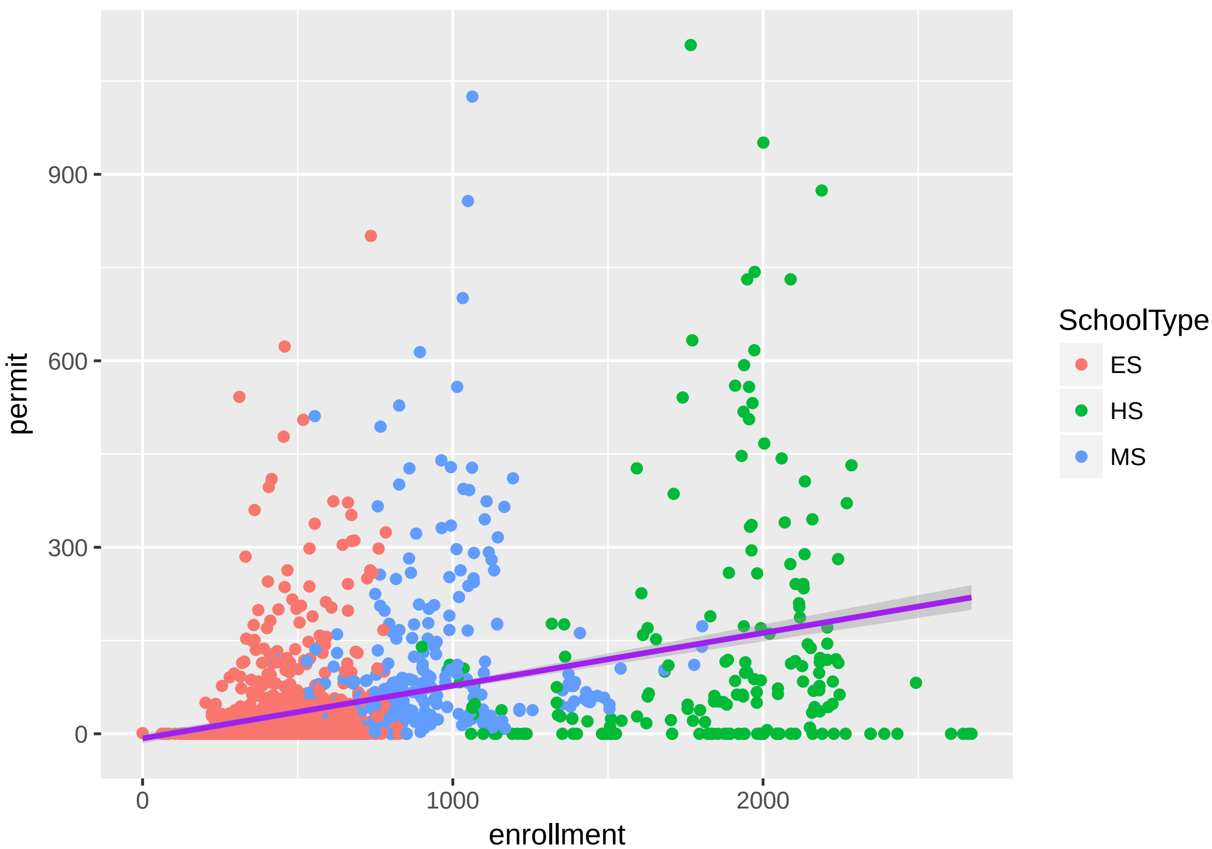
```r
school_new <- gather(schoolDS_new, key, value, -c(1:3))
year <- school_new$key


school_new[grepl("\\.", school_new$key) == FALSE, "key"] <- "enrollment"
school_new[grepl("\\.1", school_new$key) == TRUE, "key"] <- "capacity"
school_new[grepl("\\.2", school_new$key) == TRUE, "key"] <- "%utilization"
school_new[grepl("\\.3", school_new$key) == TRUE, "key"] <- "permit"

school_new <- cbind(school_new, year)

enrollDs <- school_new[school_new$key == "enrollment", -4]
capDs <- school_new[school_new$key == "capacity", -c(1, 2, 4, 6)]
utilDs <- school_new[school_new$key == "%utilization", -c(1, 2, 4,
    6)]
permitDs <- school_new[school_new$key == "permit", -c(1, 2, 4, 6)]

enrollDs <- rename(enrollDs, enrollment = "value")
```

```r
capDs <- rename(capDs, capacity = "value")
utilDs <- rename(utilDs, `%utilization` = "value")
permitDs <- rename(permitDs, permit = "value")

schoolDF <- cbind(enrollDs, capDs, utilDs, permitDs)
schoolDF <- schoolDF[, -c(6, 8, 10)]
year <- schoolDF$year
schoolDF <- schoolDF[, -5]
schoolDF <- cbind(year, schoolDF)

schoolDF$enrollment <- as.numeric(schoolDF$enrollment)
schoolDF$capacity <- as.numeric(schoolDF$capacity)
schoolDF$permit <- as.numeric(as.character(schoolDF$permit))
# schoolDF$`%utilization` <- as.numeric(schoolDF$`%utilization`)
schoolDF$year <- as.Date(schoolDF$year, "%Y")

schoolDF <- mutate(schoolDF, percent_capacity = schoolDF$enrollment/schoolDF$capacity)
schoolDF <- na.omit(schoolDF)
# schoolDF <- schoolDF[-schoolDF$percent_capacity==Inf,]
schoolDF <- subset(schoolDF, !schoolDF$percent_capacity == Inf)
```

```r
p1 <- ggplot(schoolDF, aes(x = year, y = permit, col = year)) + geom_point()
p1
```




```r
ggplot(schoolDF, aes(x = enrollment, y = permit, col = SchoolType)) +
    geom_point() + geom_smooth(method = lm, col = "purple")
```



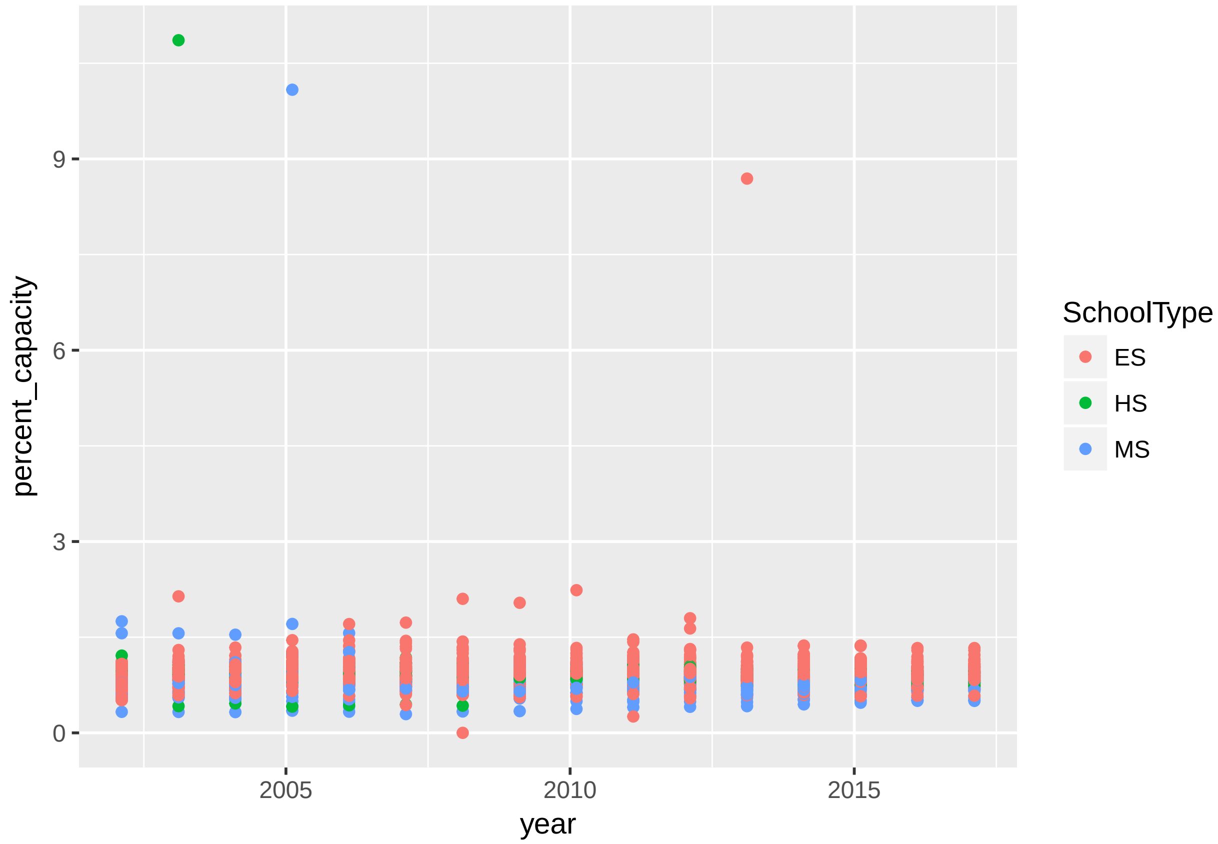

```
cor(schoolDF$permit, schoolDF$enrollment, method = c("pearson", "kendall",
    "spearman"))
```

```
## [1] 0.3790642
```

```
cor.test(schoolDF$permit, schoolDF$enrollment, method = c("pearson",
    "kendall", "spearman"))
```

```
##
##  Pearson's product-moment correlation
##
## data:  schoolDF$permit and schoolDF$enrollment
## t = 17.122, df = 1747, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3382018 0.4185000
## sample estimates:
##       cor
## 0.3790642
```

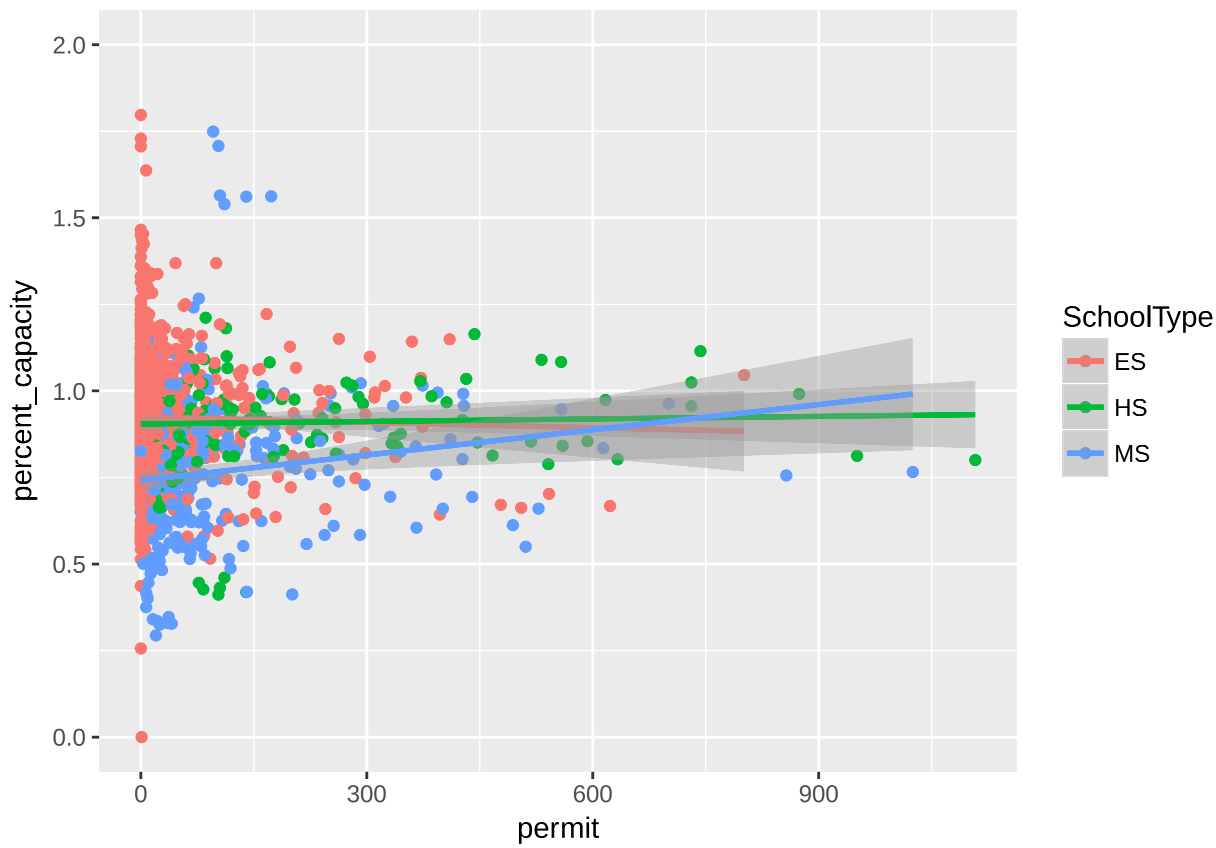
```
ggplot(schoolDF, aes(x = year, y = percent_capacity, col = SchoolType)) +
    geom_point()
```




```
ggplot(schoolDF, aes(x = permit, y = percent_capacity, col = SchoolType)) +
    geom_point() + ylim(0, 2) + geom_smooth(method = lm)
```

```
## Warning: Removed 7 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 7 rows containing missing values (geom_point).
```



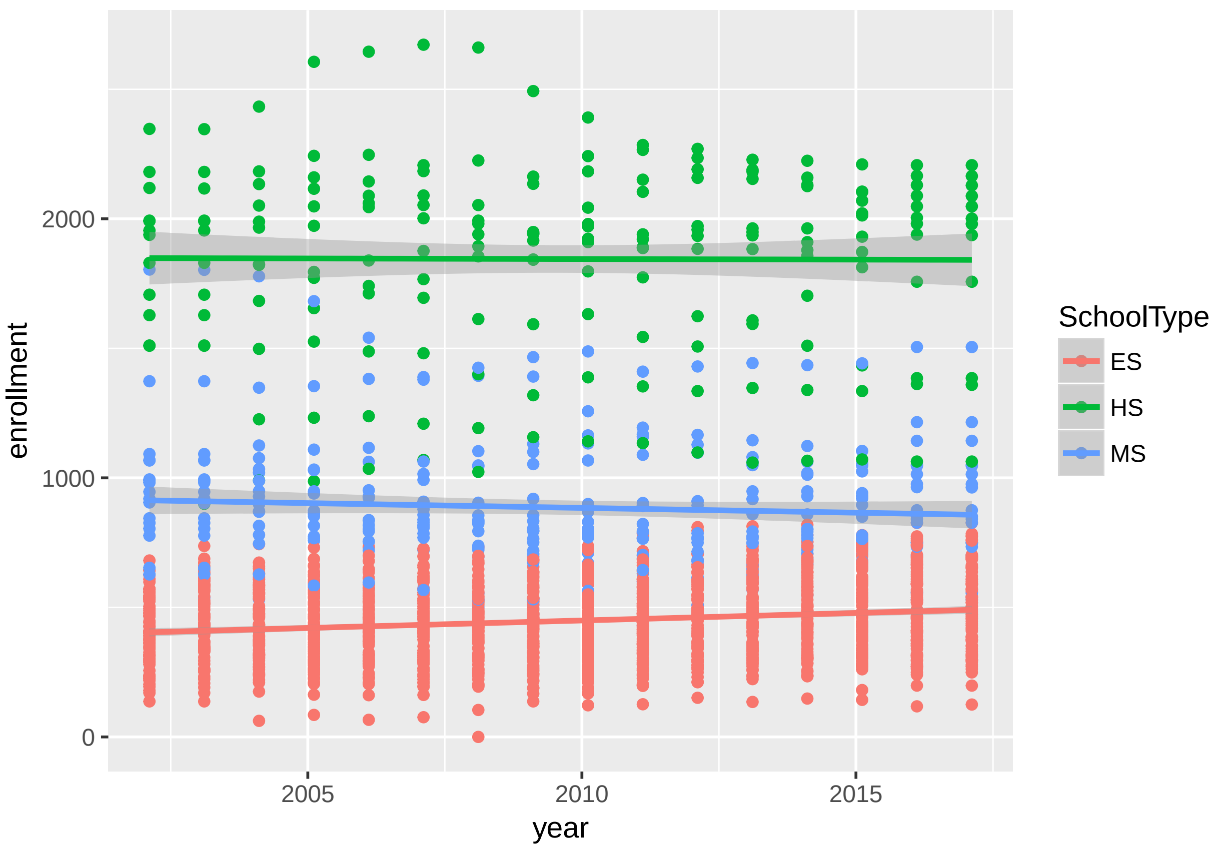

```
cor(schoolDF$permit, schoolDF$percent_capacity, method = c("pearson",
    "kendall", "spearman"))
```

```
## [1] 0.0520807
```

```
cor.test(schoolDF$permit, schoolDF$percent_capacity, method = c("pearson",
    "kendall", "spearman"))
```

```
##
##  Pearson's product-moment correlation
##
## data:  schoolDF$permit and schoolDF$percent_capacity
## t = 2.1798, df = 1747, p-value = 0.02941
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.005222081 0.098711106
## sample estimates:
##       cor
## 0.0520807
```

```
ggplot(schoolDF, aes(x = year, y = enrollment, col = SchoolType)) +
    geom_point() + geom_smooth(method = lm)
```




```
cor(schoolDF$permit, schoolDF$enrollment, method = c("pearson", "kendall",
    "spearman"))
```

```
## [1] 0.3790642
```

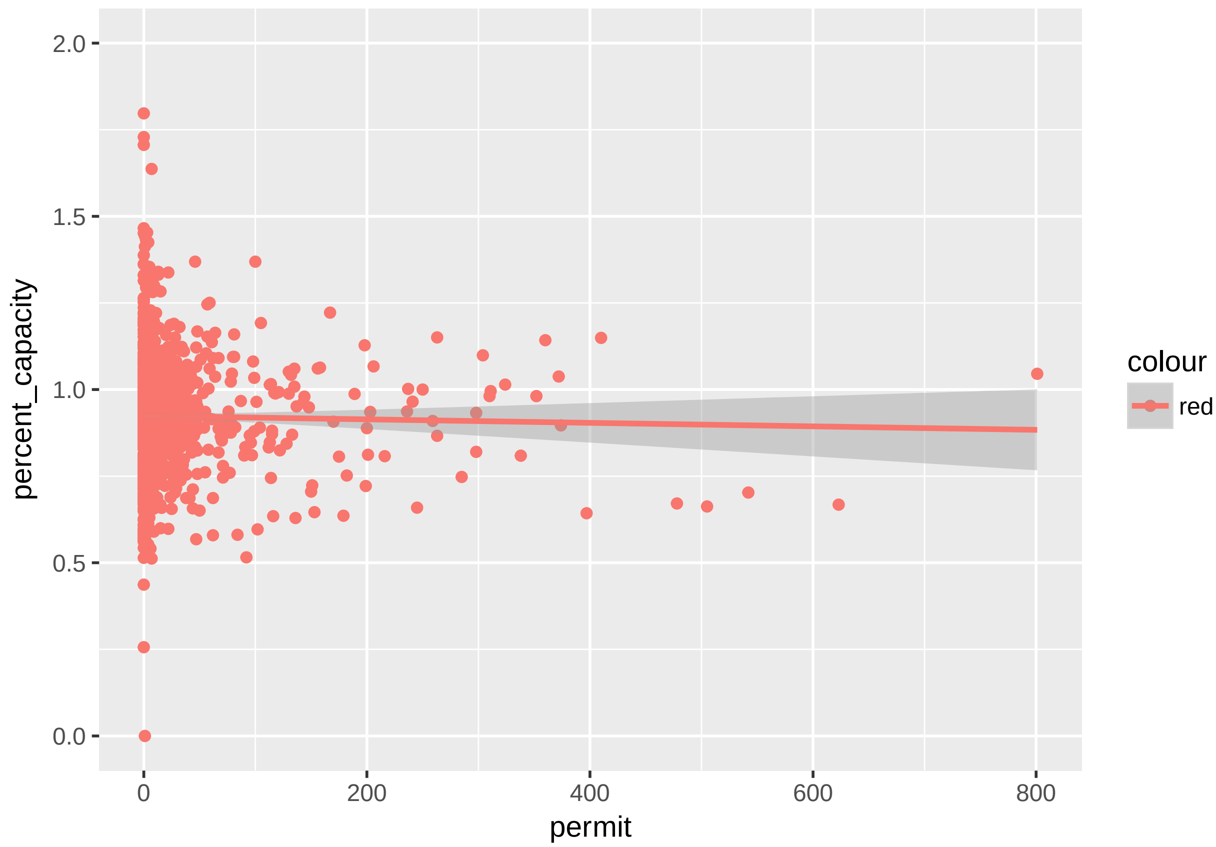
```
cor.test(schoolDF$permit, schoolDF$enrollment, method = c("pearson",
    "kendall", "spearman"))
```

```
##
##  Pearson's product-moment correlation
##
## data:  schoolDF$permit and schoolDF$enrollment
## t = 17.122, df = 1747, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3382018 0.4185000
## sample estimates:
##       cor
## 0.3790642
```

Elementary Schools

```
schoolDFES <- schoolDF[schoolDF$SchoolType == "ES", ]
ggplot(schoolDFES, aes(x = permit, y = percent_capacity, col = "red")) +
    geom_point() + ylim(0, 2) + geom_smooth(method = lm)
```

```
## Warning: Removed 5 rows containing non-finite values (stat_smooth).

## Warning: Removed 5 rows containing missing values (geom_point).
```




```r
cor(schoolDFES$permit, schoolDFES$percent_capacity, method = c("pearson",
    "kendall", "spearman"))
```

```
## [1] 0.04995587
```

```r
cor.test(schoolDFES$permit, schoolDFES$percent_capacity, method = c("pearson",
    "kendall", "spearman"))
```

```
##
##  Pearson's product-moment correlation
##
## data:  schoolDFES$permit and schoolDFES$percent_capacity
## t = 1.7691, df = 1251, p-value = 0.07712
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.005438614  0.105044692
## sample estimates:
##        cor
## 0.04995587
```

```r
# ggplot(schoolDFES,aes(x=enrollment, y=permit, col='red'))+
# geom_point()
```

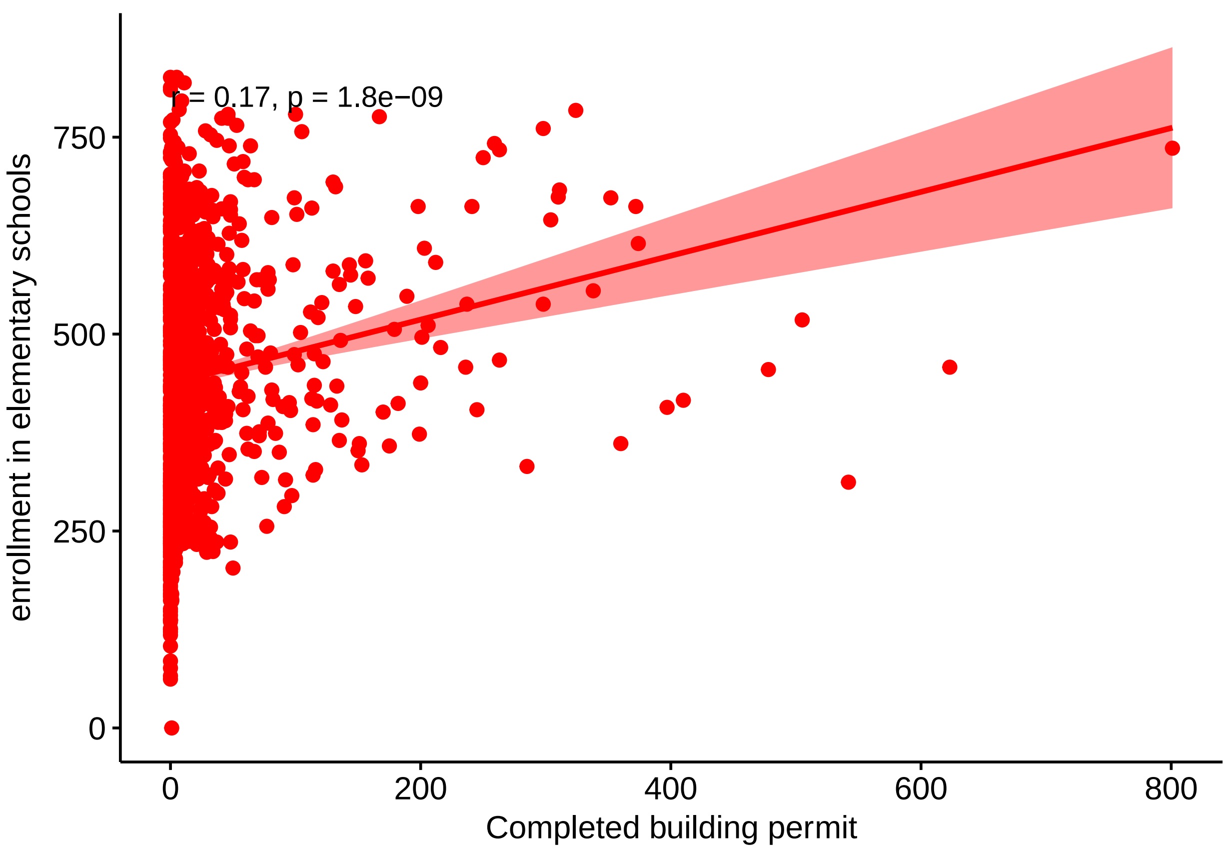
```r
cor(schoolDFES$permit, schoolDFES$enrollment, method = c("pearson",
    "kendall", "spearman"))
```

```
## [1] 0.1687993
```

```r
cor.test(schoolDFES$permit, schoolDFES$enrollment, method = c("pearson",
    "kendall", "spearman"))
```

```
##
##  Pearson's product-moment correlation
##
## data:  schoolDFES$permit and schoolDFES$enrollment
## t = 6.0573, df = 1251, p-value = 1.829e-09
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1144901 0.2221025
## sample estimates:
##       cor
## 0.1687993
```

```r
ggscatter(schoolDFES, x = "permit", y = "enrollment", add = "reg.line",
    conf.int = TRUE, cor.coef = TRUE, cor.method = "pearson", xlab = "Completed building permit",
    ylab = "enrollment in elementary schools", color = "red")
```

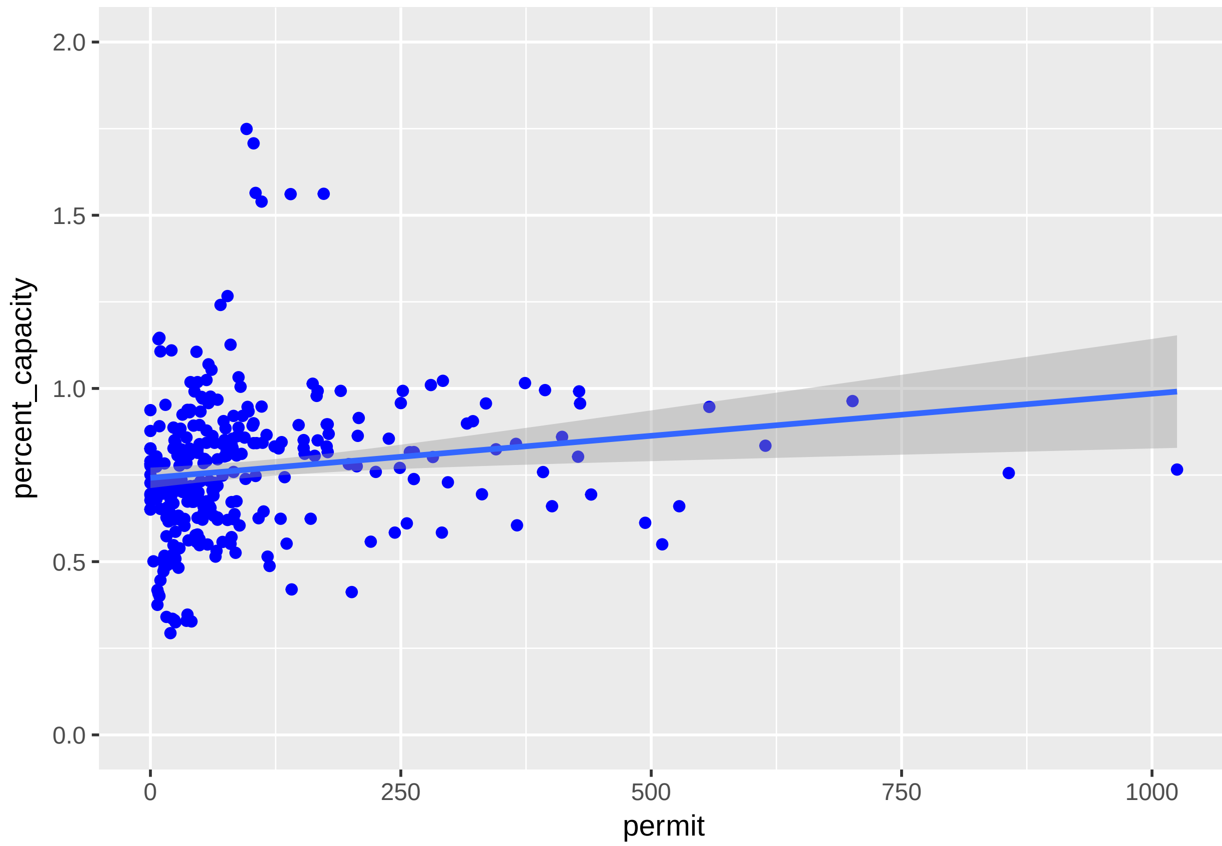


Middle Schools

```r
schoolDFMS <- schoolDF[schoolDF$SchoolType == "MS", ]
ggplot(schoolDFMS, aes(x = permit, y = percent_capacity)) + geom_point(color = "blue") +
    ylim(0, 2) + geom_smooth(method = lm)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



Correlation between enrollment and percentage capacity

```r
cor(schoolDFMS$permit, schoolDFMS$percent_capacity, method = c("pearson",
    "kendall", "spearman"))
```

```
## [1] 0.03625019
```

```r
cor.test(schoolDFMS$permit, schoolDFMS$percent_capacity, method = c("pearson",
    "kendall", "spearman"))
```

```
##
##  Pearson's product-moment correlation
##
## data:  schoolDFMS$permit and schoolDFMS$percent_capacity
## t = 0.63038, df = 302, p-value = 0.5289
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.07655429  0.14813840
```

```
## sample estimates:
##        cor
## 0.03625019
```

```r
# ggplot(schoolDFMS,aes(x=enrollment, y=permit, col='blue'))+
# geom_point(color='blue')
```

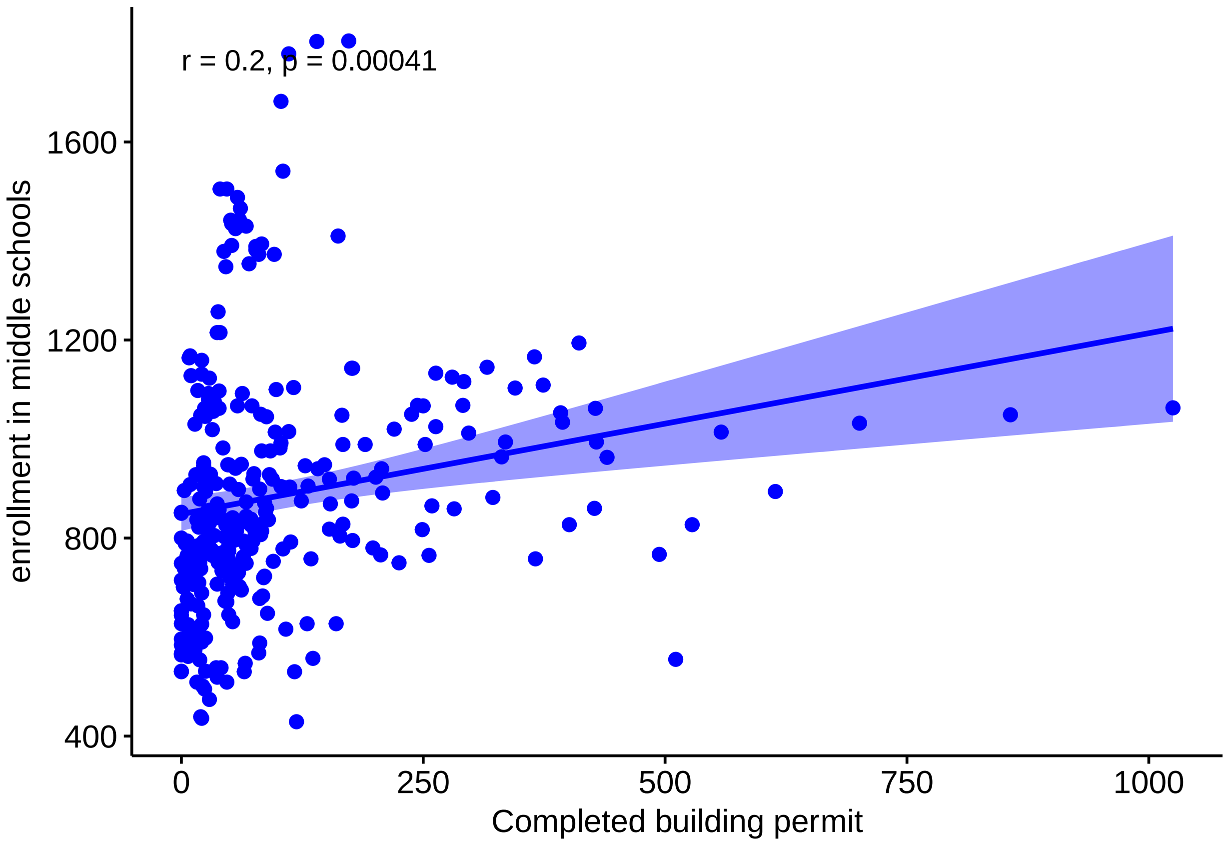
```r
cor(schoolDFMS$permit, schoolDFMS$enrollment, method = c("pearson",
    "kendall", "spearman"))
```

```
## [1] 0.2012391
```

```r
cor.test(schoolDFMS$permit, schoolDFMS$enrollment, method = c("pearson",
    "kendall", "spearman"))
```

```
##
##  Pearson's product-moment correlation
##
## data:  schoolDFMS$permit and schoolDFMS$enrollment
## t = 3.5702, df = 302, p-value = 0.0004148
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.09080236 0.30678640
## sample estimates:
##       cor
## 0.2012391
```

```r
ggscatter(schoolDFMS, x = "permit", y = "enrollment", add = "reg.line",
    conf.int = TRUE, cor.coef = TRUE, cor.method = "pearson", xlab = "Completed building permit",
    ylab = "enrollment in middle schools", color = "blue")
```

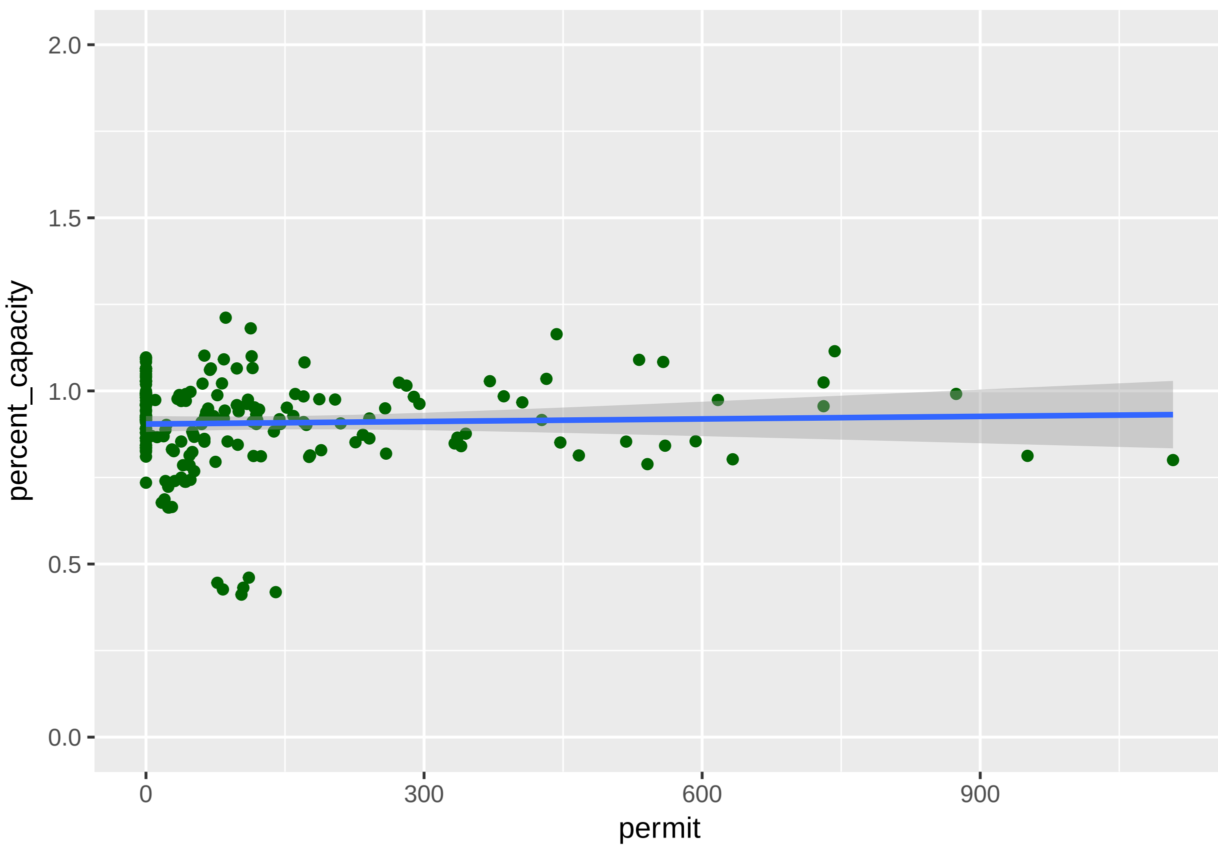


High Schools

```
schoolDFHS <- schoolDF[schoolDF$SchoolType == "HS", ]
ggplot(schoolDFHS, aes(x = permit, y = percent_capacity)) + geom_point(color = "darkgreen") +
    ylim(0, 2) + geom_smooth(method = lm)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



Correlation between enrollment and percentage capacity

```
cor(schoolDFHS$permit, schoolDFHS$percent_capacity, method = c("pearson",
    "kendall", "spearman"))
```

```
## [1] 0.1410996
```

```
cor.test(schoolDFHS$permit, schoolDFHS$percent_capacity, method = c("pearson",
    "kendall", "spearman"))
```

```
##
##  Pearson's product-moment correlation
##
## data:  schoolDFHS$permit and schoolDFHS$percent_capacity
## t = 1.9646, df = 190, p-value = 0.05092
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.0005190177  0.2771698446
## sample estimates:
##       cor
## 0.1410996
```

```
# ggplot(schoolDFHS,aes(x=enrollment, y=permit))+
# geom_point(color='darkgreen')
```

Correlation between enrollment and building permit

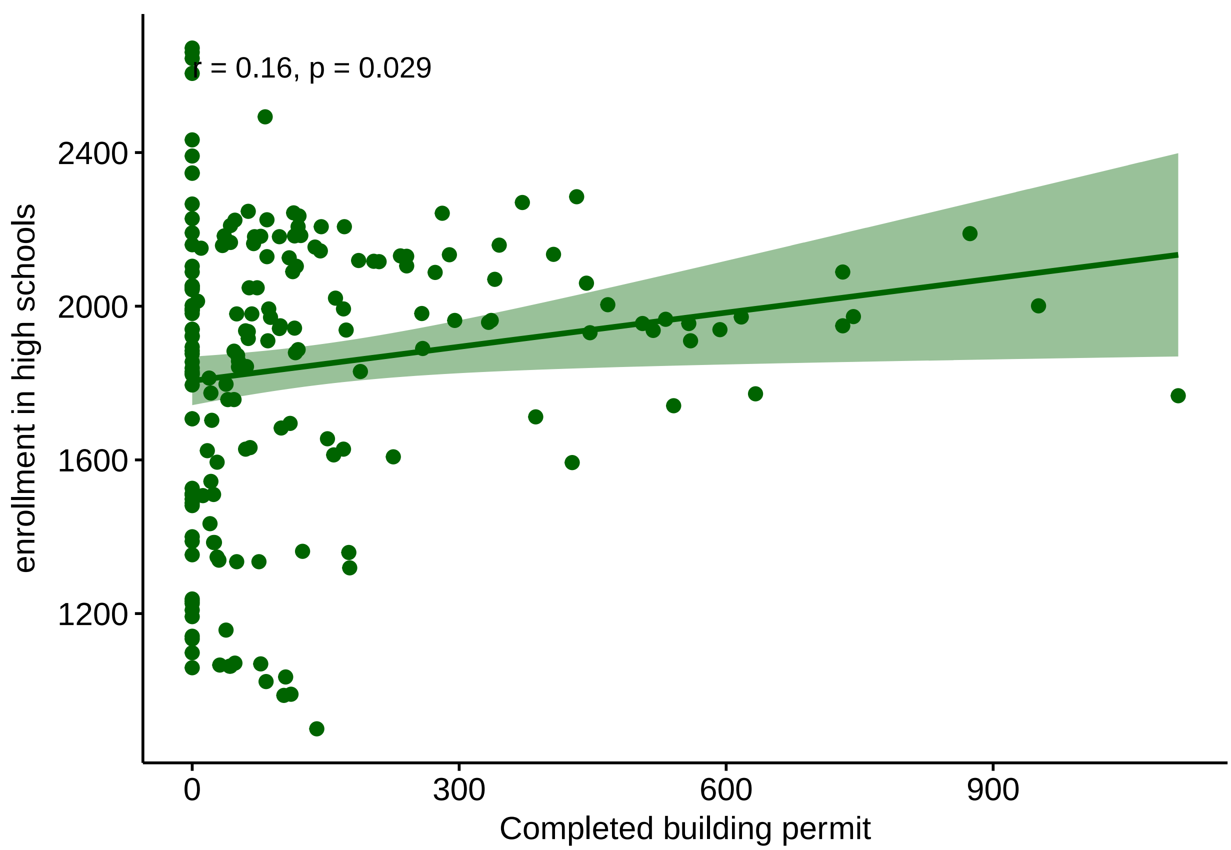
```
cor(schoolDFHS$permit, schoolDFHS$enrollment, method = c("pearson",
    "kendall", "spearman"))
```

```
## [1] 0.1574705
```

```
cor.test(schoolDFHS$permit, schoolDFHS$enrollment, method = c("pearson",
    "kendall", "spearman"))
```

```
##
##  Pearson's product-moment correlation
##
## data:  schoolDFHS$permit and schoolDFHS$enrollment
## t = 2.198, df = 190, p-value = 0.02916
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.01622401 0.29255509
## sample estimates:
##       cor
## 0.1574705
```

```
ggscatter(schoolDFHS, x = "permit", y = "enrollment", add = "reg.line",
    conf.int = TRUE, cor.coef = TRUE, cor.method = "pearson", xlab = "Completed building permit",
    ylab = "enrollment in high schools", color = "darkgreen")
```



Education and family structure

Mehdi Khan

December 13, 2017

load the libraries

```
suppressMessages(suppressWarnings(library(dplyr)))
suppressMessages(suppressWarnings(library(stringr)))

suppressMessages(suppressWarnings(library(psych)))
suppressMessages(suppressWarnings(library(ggplot2)))

suppressMessages(suppressWarnings(library(devtools)))
suppressMessages(suppressWarnings(library(stats)))
suppressMessages(suppressWarnings(library(tidyr)))
suppressMessages(suppressWarnings(library(ggpubr)))
```

Introduction

Data about Households, family structures and their characterstics in 50 US states published by US Census Bureau was examined in this project to see if the data provides any insight on the impact of family structures on education of people in a society.

Data collection

The data represents selected social chracterstics in the United States in the period of 5 years from 2011 to 2015 and compiled by American Community Survey,US Census Bureau. This publicly available data was downloaded in CSV format for this project.

Data Source

The data was published by US Census Bureau and posted in American Fact Finder website: https://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml?pid=ACS_15_5YR_DP02&src=pt

data prepration:

First the data was imported in R:

```
originDS <- read.csv("https://raw.githubusercontent.com/kmehdi2017/projectProp/master/ProjectProposal/A
    sep = ",", stringsAsFactors = FALSE)
```

The original data had a lot of variables, which are not relevant to this study, therefore a subset of the data was extracted. The following are the variables with their descriptions that were selected for the project:

GEO.display.label: Geography HC01_VC04: Estimate: Total households - Family households (families) HC01_VC06: Estimate: Total households - Family households (families) - Married-couple family HC01_VC08: Estimate: Total households - Family households (families) - Male householder, no wife present HC01_VC10: Estimate: Total households - Family households (families) - Female householder, no husband present HC01_VC76: Estimate: SCHOOL ENROLLMENT - Population 3 years and over enrolled in school HC01_VC91: Estimate: EDUCATIONAL ATTAINMENT - Population 25 years and over - Bachelor's degree

```
vars <- c("GEO.display.label", "HC01_VC04", "HC01_VC06", "HC01_VC08",
    "HC01_VC10", "HC01_VC76", "HC01_VC91")

familyEduDS <- originDS[-1, vars]

head(familyEduDS)
```

```
##   GEO.display.label HC01_VC04 HC01_VC06 HC01_VC08 HC01_VC10 HC01_VC76
## 2           Alabama   1238967    880942     78073    279952   1206014
## 3            Alaska    167562    124649     14733     28180    195151
## 4           Arizona   1581380   1142828    131803    306749   1754549
## 5          Arkansas    759924    558920     50484    150520    750024
## 6        California   8732734   6245351    759047   1728336  10579176
## 7          Colorado   1300972   1003324     91627    206021   1395787
##   HC01_VC91
## 2    478812
## 3     83201
## 4    753425
## 5    267741
## 6   5002596
## 7    847977
```

providing meaningful names to columns:

```
columnNames <- c("states", "total_family", "married_couple_family",
    "husband_only_family", "wife_only_family", "school_enrollment",
    "bachelor_degree")

colnames(familyEduDS) <- columnNames

head(familyEduDS)
```

```
##       states total_family married_couple_family husband_only_family
## 2    Alabama      1238967                880942               78073
## 3     Alaska       167562                124649               14733
## 4    Arizona      1581380               1142828              131803
## 5   Arkansas       759924                558920               50484
## 6 California      8732734               6245351              759047
## 7   Colorado      1300972               1003324               91627
##   wife_only_family school_enrollment bachelor_degree
## 2           279952           1206014          478812
## 3            28180            195151           83201
## 4           306749           1754549          753425
## 5           150520            750024          267741
## 6          1728336          10579176         5002596
## 7           206021           1395787          847977
```

Research question

Does the family structure of single parents or two parents families have any impact on the number of educated people in a society ?

case

Each case represents a state in the United States, there are 51 of them.

Type of study

This is an observational study

Response

The response variables are the estimates of school enrollment, and number of bachelor degree holders. Both of them are numerical.

Explanatory

The explanatory variables are the estimates of family types (i.e. single-parents family and two-parents family) and are numerical.

further data prepration

The data types of the fields were converted to numeric for calculation:

```
familyEduDS$total_family <- as.numeric(familyEduDS$total_family)
familyEduDS$married_couple_family <- as.numeric(familyEduDS$married_couple_family)
familyEduDS$husband_only_family <- as.numeric(familyEduDS$husband_only_family)
familyEduDS$wife_only_family <- as.numeric(familyEduDS$wife_only_family)
familyEduDS$school_enrollment <- as.numeric(familyEduDS$school_enrollment)
familyEduDS$bachelor_degree <- as.numeric(familyEduDS$bachelor_degree)
```

Five derived fields were created that describe: 1. the number of single parent families in each state 2. average school enrollment per family in each state 3. average bachelor degree holders per family in each state 4. ratio of two-parents families in each state 5. ratio of single-parents families in each state

```
familyEduDS <- mutate(familyEduDS, single_parent_family = husband_only_family +
    wife_only_family)
familyEduDS <- mutate(familyEduDS, avg_enrollment = round(school_enrollment/total_family,
    2))
familyEduDS <- mutate(familyEduDS, avg_bachelor = round(bachelor_degree/total_family,
    2))
familyEduDS <- mutate(familyEduDS, ratio_both_parents = round(married_couple_family/total_family,
    2))
familyEduDS <- mutate(familyEduDS, ratio_single_parents = round(single_parent_family/total_family,
    2))
```

Analysis approach:

correlation analysis was used to find if there is any correlation between the type of family structures and the number of school enrollment and the number of bachelor degree holders.

Hypothesis:

Null Hypothesis, Ho: family structures does not affect education i.e.There is no correlation between family structures and education, correlation coefficients = 0

Alternative Hypothesis, Ha: family structures does affect education There is correlation between family structures and education, correlation coefficients != 0

descriptive Analysis:

```
describe(familyEduDS$married_couple_family)
```

```
##    vars  n    mean      sd median  trimmed      mad   min     max   range
## X1    1 51 1107424 1181185 752359 880225.4 708491.5 65383 6245351 6179968
##    skew kurtosis       se
## X1 2.28     6.15 165398.9
```

```
describe(familyEduDS$single_parent_family)
```

```
##    vars  n     mean       sd median  trimmed      mad   min     max
## X1    1 51 407488.5 471125.8 294017 311463.8 283268.5 29874 2487383
##      range skew kurtosis      se
## X1 2457509 2.42     6.67 65970.8
```

```
describe(familyEduDS$avg_enrollment)
```

```
##    vars  n mean  sd median trimmed  mad  min  max range skew kurtosis   se
## X1    1 51 1.05 0.1   1.03    1.04 0.06 0.86 1.39  0.53 1.33     3.31 0.01
```

```
describe(familyEduDS$avg_bachelor)
```

```
##    vars  n mean  sd median trimmed  mad  min  max range skew kurtosis   se
## X1    1 51  0.5 0.1   0.49     0.5 0.07 0.32 0.89  0.57 1.02     3.13 0.01
```

```
describe(familyEduDS$ratio_both_parents)
```

```
##    vars  n mean   sd median trimmed  mad  min  max range  skew kurtosis
## X1    1 51 0.74 0.05   0.74    0.74 0.04 0.55 0.82  0.27 -1.38     4.23
##      se
## X1 0.01
```

```
describe(familyEduDS$ratio_single_parents)
```

```
##    vars  n mean   sd median trimmed  mad  min  max range skew kurtosis
## X1    1 51 0.26 0.05   0.26    0.26 0.04 0.18 0.45  0.27 1.38     4.23
##      se
## X1 0.01
```

```
summary(familyEduDS$avg_enrollment)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.860   0.995   1.030   1.046   1.080   1.390
```

```
summary(familyEduDS$avg_bachelor)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3200  0.4400  0.4900  0.5022  0.5650  0.8900
```

```r
summary(familyEduDS$ratio_both_parents)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5500  0.7200  0.7400  0.7408  0.7750  0.8200
```

```r
summary(familyEduDS$ratio_single_parents)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1800  0.2250  0.2600  0.2592  0.2800  0.4500
```

```r
IQR_enrollment <- 1.08 - 0.995
IQR_bachelor <- 0.565 - 0.44

IQR_single_parents <- 0.28 - 0.225
IQR_two_parents <- 0.775 - 0.72

IQR_enrollment
```

```
## [1] 0.085
```

```r
IQR_bachelor
```

```
## [1] 0.125
```

```r
IQR_single_parents
```

```
## [1] 0.055
```

```r
IQR_two_parents
```

```
## [1] 0.055
```

```r
ggplot(familyEduDS, aes(x = avg_enrollment, fill = "red", col = "blue",
    alpha = 0.2)) + geom_histogram(position = "identity", bins = 20,
    show.legend = FALSE, binwidth = 0.05) + theme(plot.title = element_text(size = 12,
    color = "blue", hjust = 0.5)) + ggtitle("distribution of average school enrollment per family ") +
    xlab("average school enrollment per family")
```

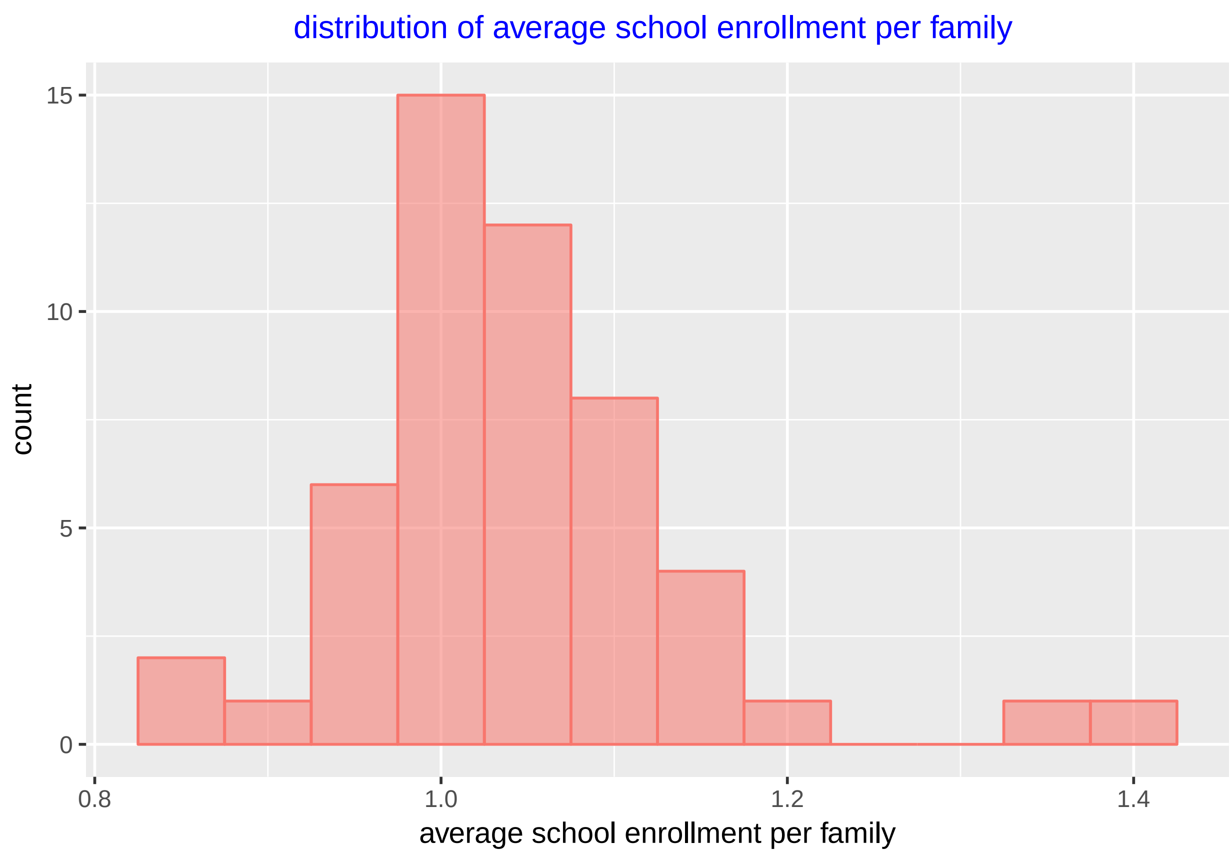


Figure 1.

```
ggplot(familyEduDS, aes(x = avg_bachelor, fill = "blue", col = "red",
    alpha = 0.2)) + geom_histogram(position = "identity", bins = 20,
    show.legend = FALSE) + theme(plot.title = element_text(size = 12,
    color = "blue", hjust = 0.5)) + ggtitle("distribution of average bachelor degree holder per family
    xlab("average bachelor degree holder per family")
```

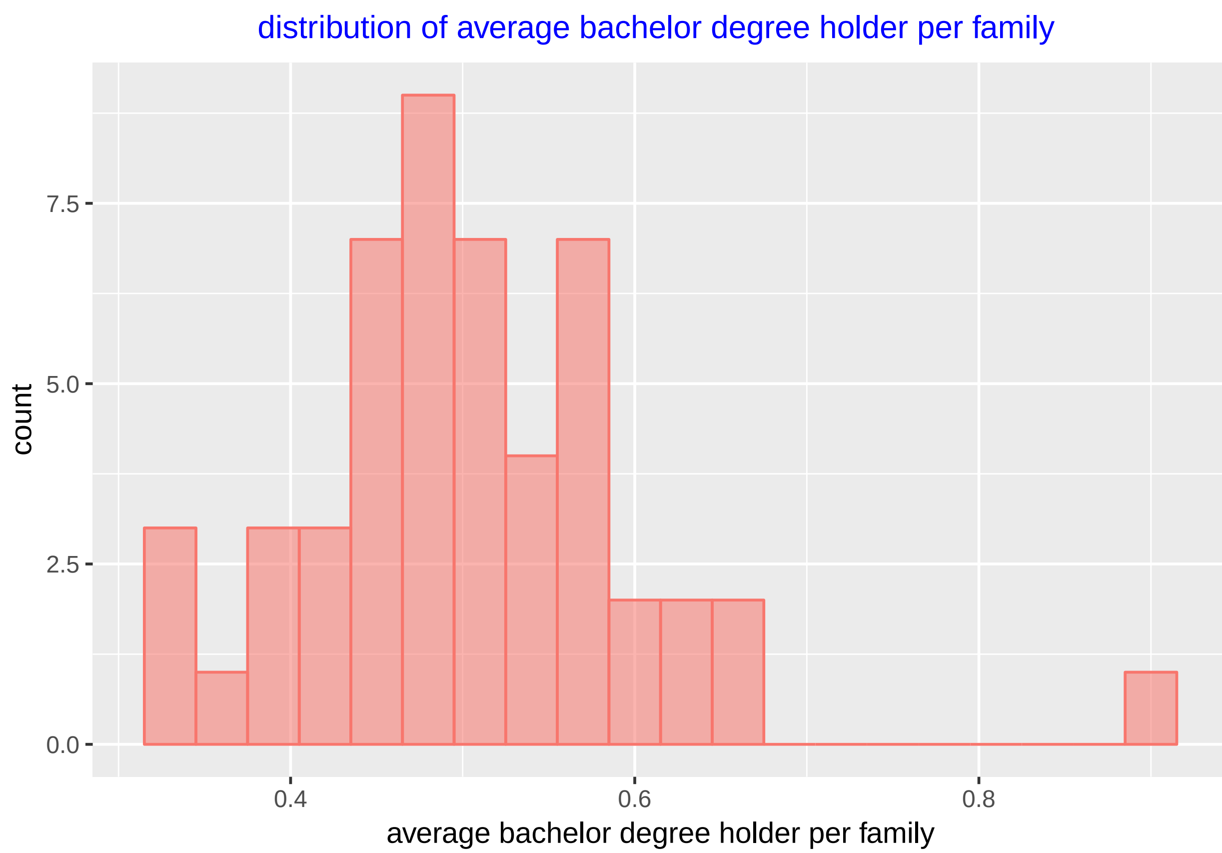


Figure 2.

```
ggplot(familyEduDS, aes(x = ratio_both_parents, fill = "blue", col = "red",
    alpha = 0.2)) + geom_histogram(position = "identity", bins = 20,
    show.legend = FALSE, binwidth = 0.01) + theme(plot.title = element_text(size = 12,
    color = "blue", hjust = 0.5)) + ggtitle("distribution of the ratios of families of both parents") +
    xlab("ratios of families of both parents")
```

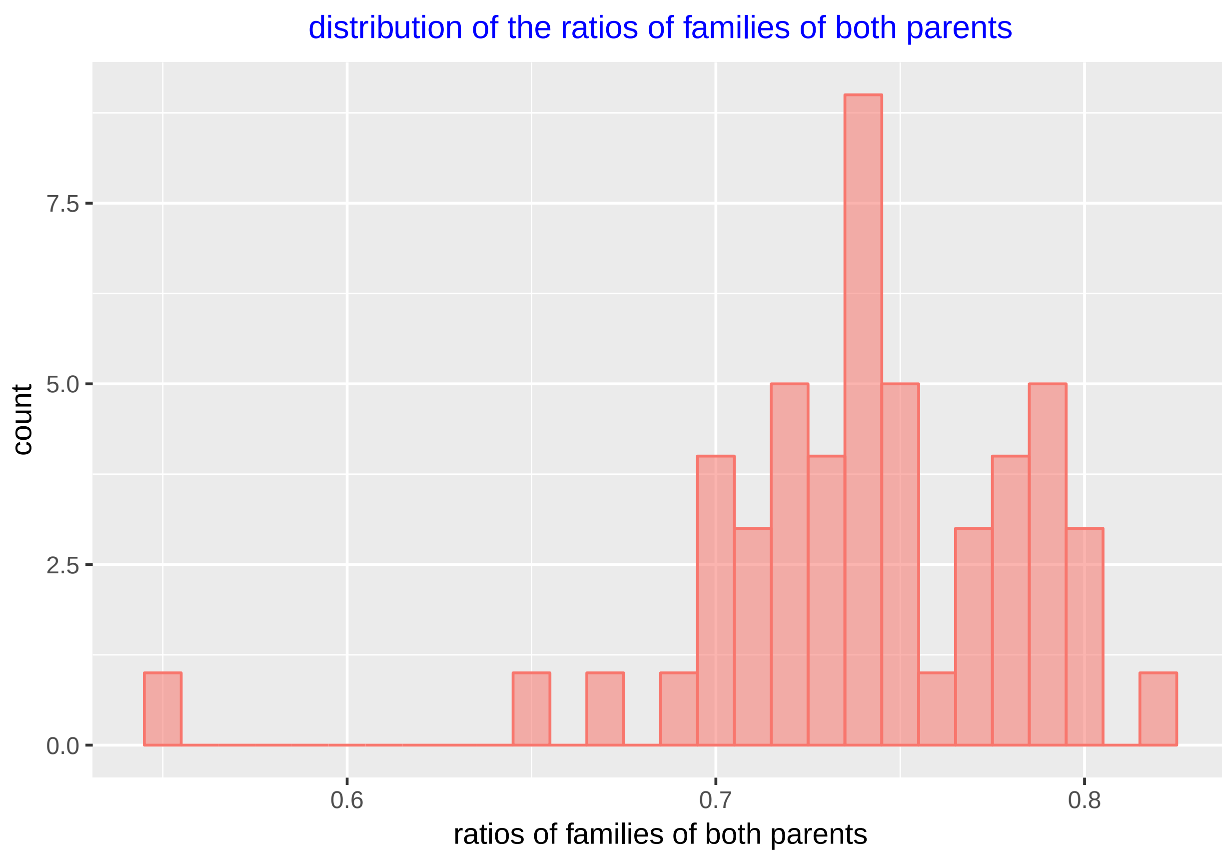


Figure 3.

```
ggplot(familyEduDS, aes(x = ratio_single_parents, alpha = 0.2)) +
    geom_histogram(position = "identity", bins = 20, show.legend = FALSE,
        binwidth = 0.01, col = "blue", fill = "blue") + theme(plot.title = element_text(size = 12,
    color = "blue", hjust = 0.5)) + ggtitle("distribution of the ratios of families of single parents")
    xlab("ratios of families of single parents")
```

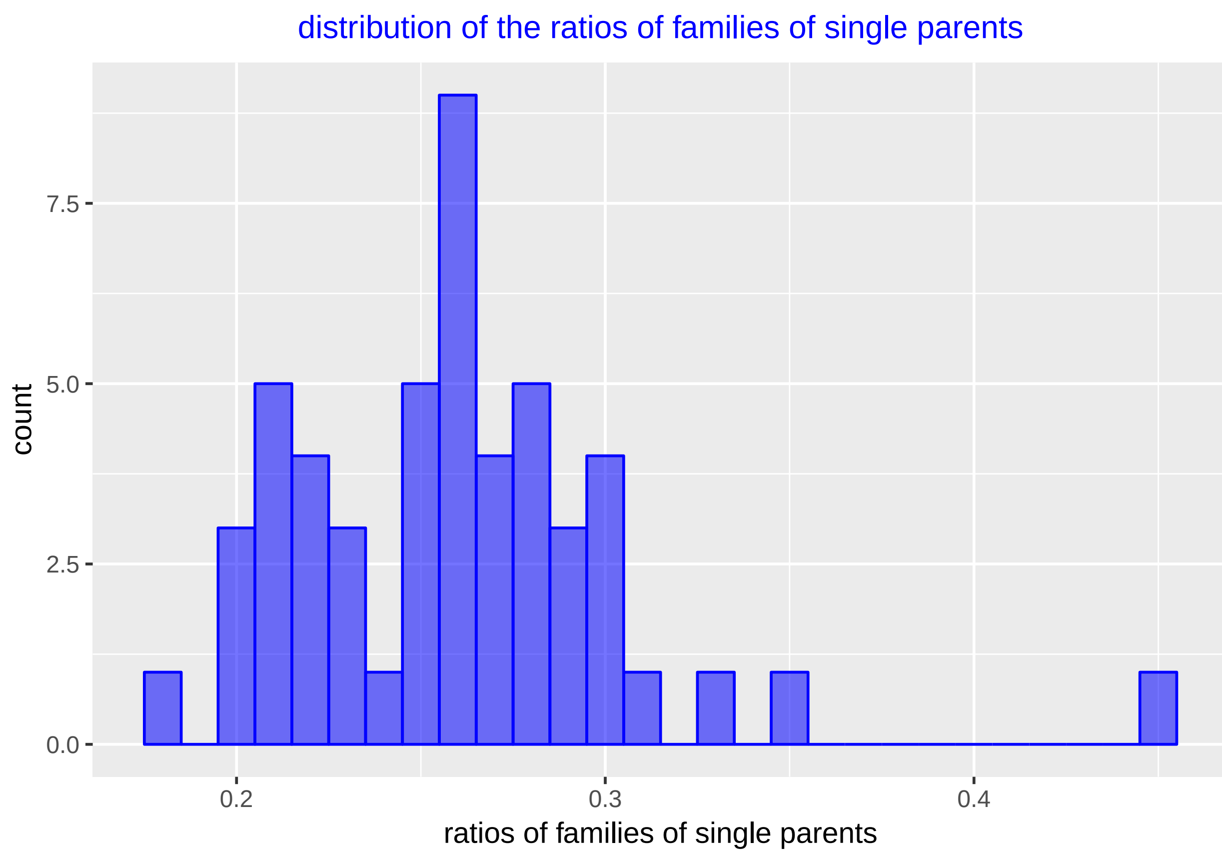


Figure 4.

```
ggplot(familyEduDS, aes(x = ratio_both_parents, y = avg_enrollment)) +
    geom_point(color = "red") + ggtitle("two-parents families vs school enrollment") +
    xlab("ratio of families with both parents") + ylab("average school enrollment per family") +
    geom_smooth(method = "auto", col = "red")
```

```
## `geom_smooth()` using method = 'loess'
```

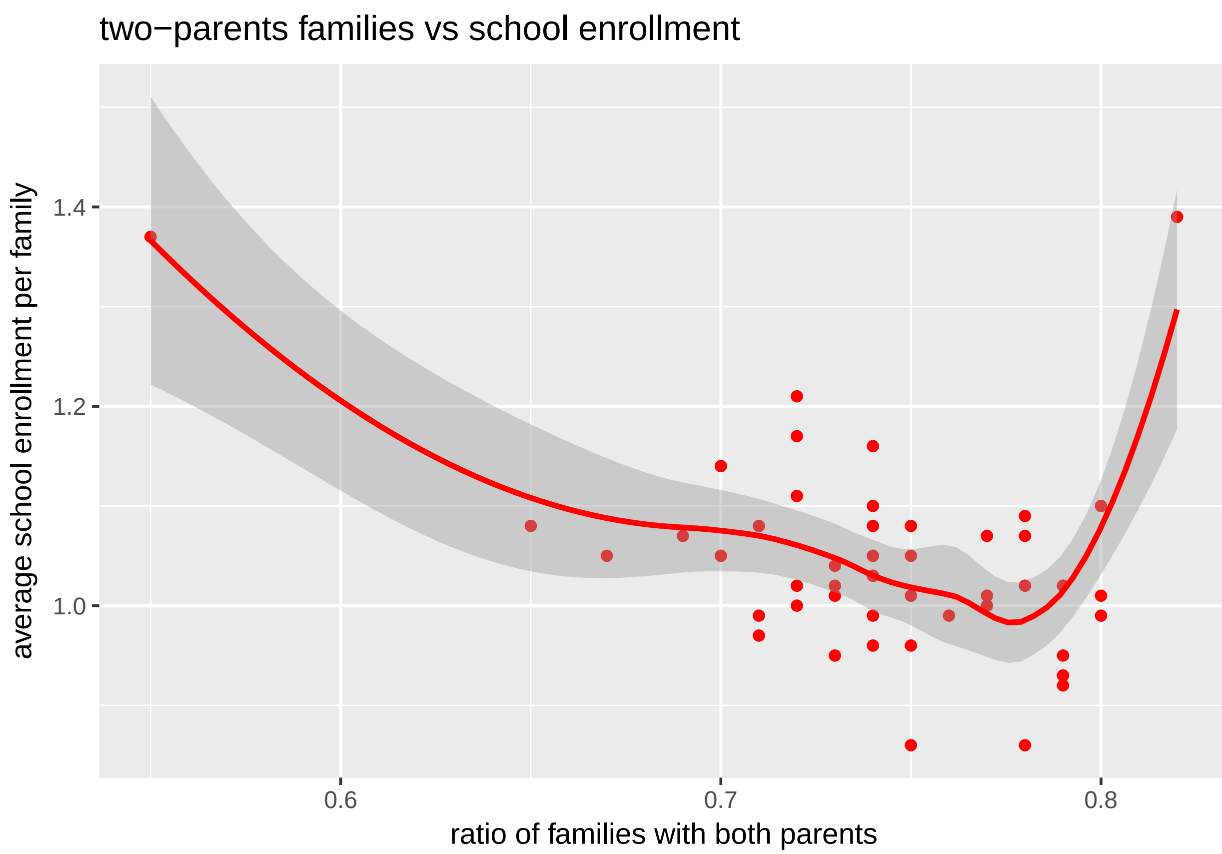


Figure 5.

The figure 5 shows a mostly linearity between school enrollment and two-parents families but two outliers on both ends heavily impact the relationship.

```
ggplot(familyEduDS, aes(x = ratio_both_parents, y = avg_bachelor)) +
    geom_point(color = "red") + ggtitle("two-parents families vs bachelor degree holders") +
    xlab("ratio of families with both parents") + ylab("average bachelor degree holders per family") +
    geom_smooth(method = "auto", col = "red")
```

```
## `geom_smooth()` using method = 'loess'
```

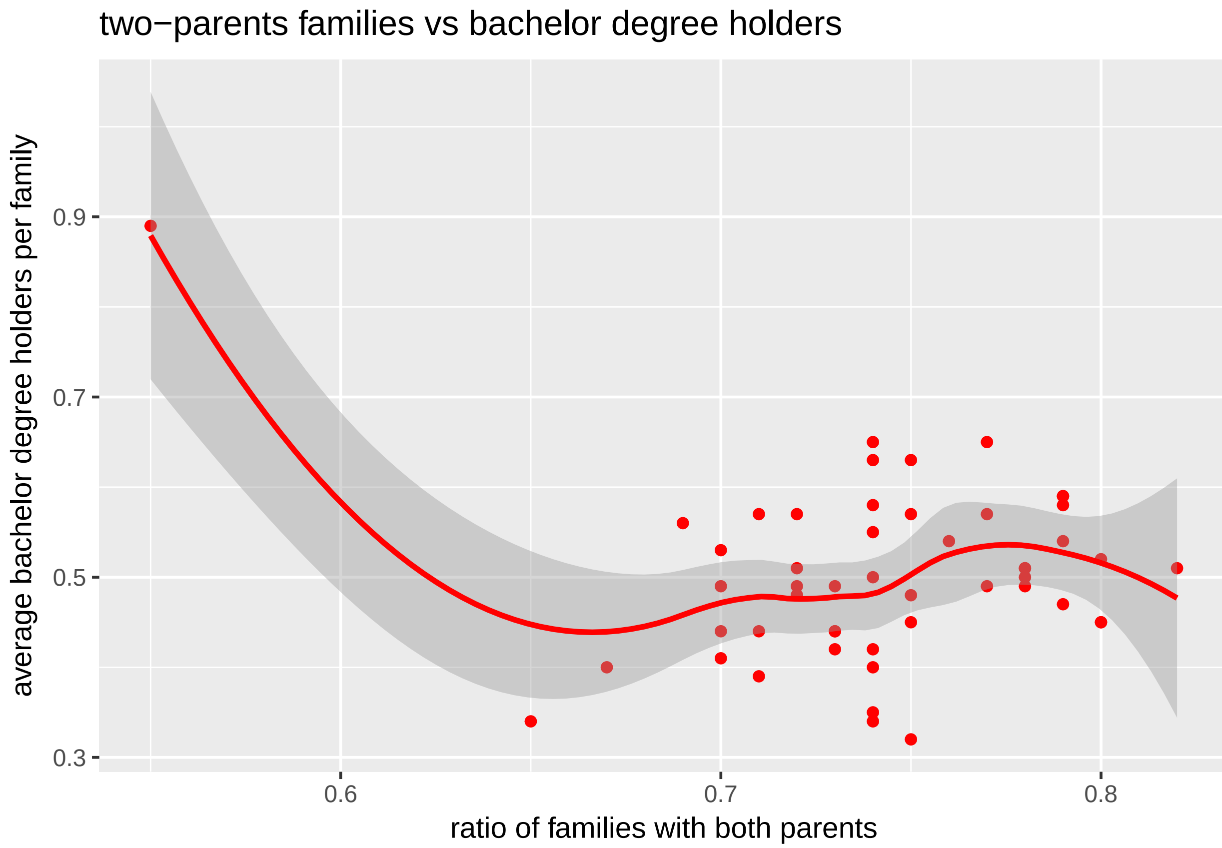


Figure 6.

The figure 6 also shows a mostly linearity between bachelor degree holders and two-parents families but one extreme outliers on one end heavily impact the relationship.

```
ggplot(familyEduDS, aes(x = ratio_single_parents, y = avg_enrollment)) +
    geom_point(color = "blue") + ggtitle("single parents families vs school enrollment") +
    xlab("ratio of families of single parents") + ylab("average school enrollment per family") +
    geom_smooth(method = "auto")
```

```
## `geom_smooth()` using method = 'loess'
```

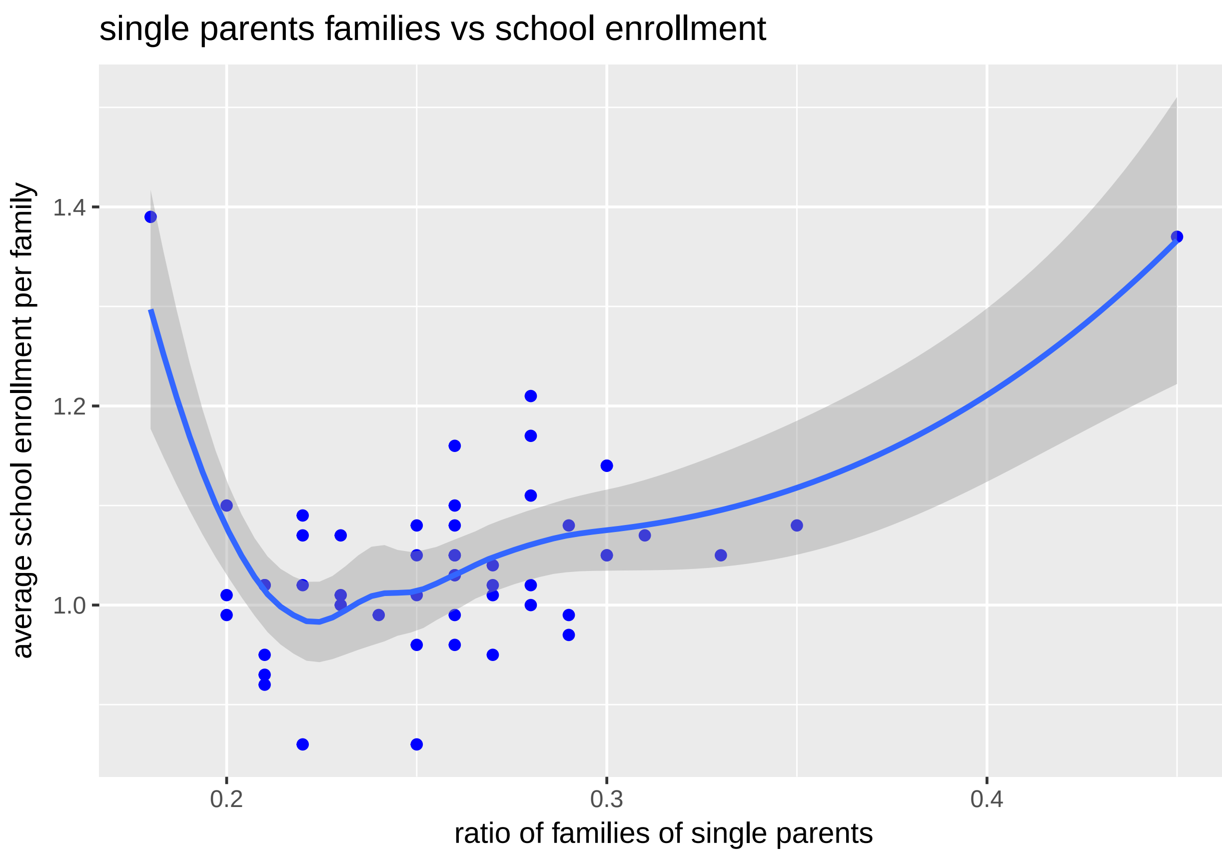


Figure 7.

The figure 7 also shows a mostly linearity between school enrollment and single parents families but two outliers on both ends heavily impact the relationship.

```
ggplot(familyEduDS, aes(x = ratio_single_parents, y = avg_bachelor)) +
    geom_point(color = "blue") + ggtitle("single parents families vs bachelor degree holders") +
    xlab("ratio of families of single parents") + ylab("average bachelor degree holders per family") +
    geom_smooth(method = "auto")
```

```
## `geom_smooth()` using method = 'loess'
```

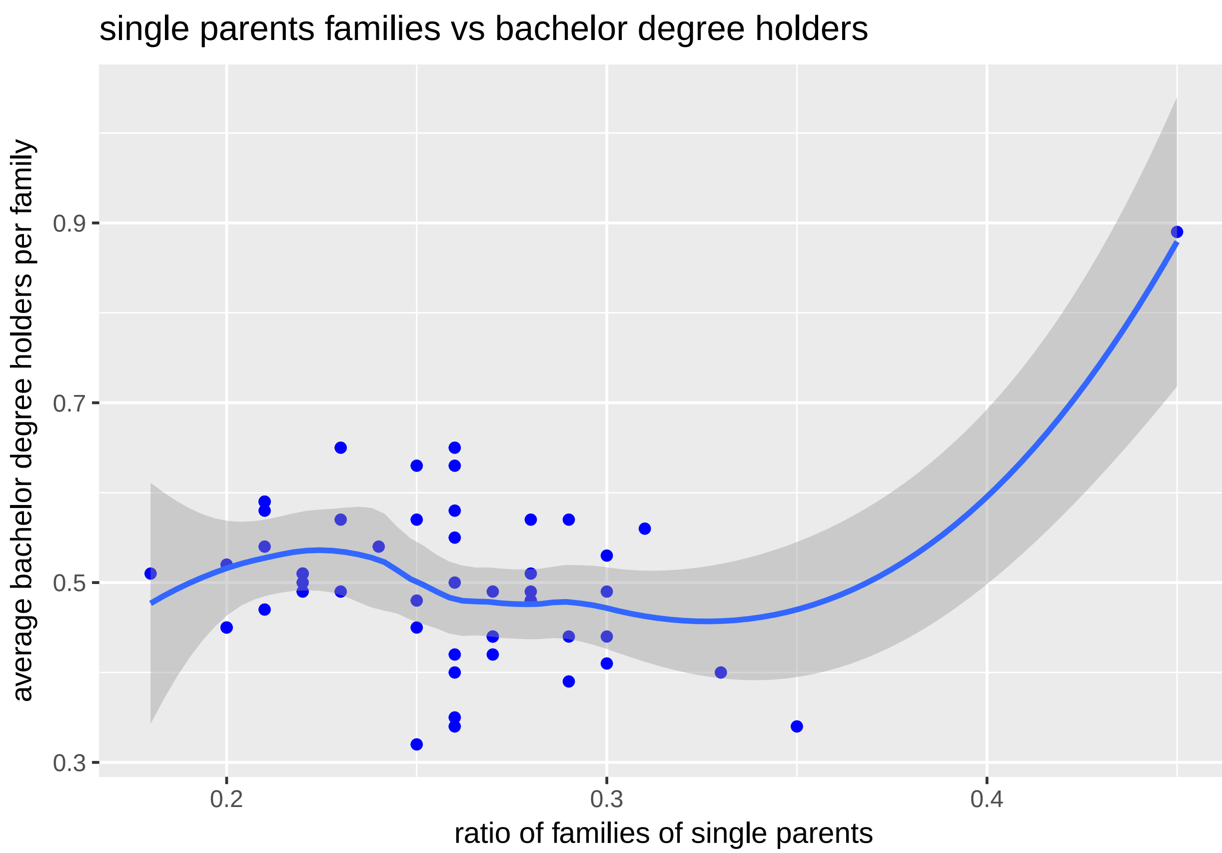


Figure 8.

The figure 8 shows a mostly linearity between bachelor degree holders and single parents family but one extreme outliers on one end heavily impact the relationship.

correlation analysis

Conditions check: All the histograms (from figure 1,2,3 and 4) shows the distributions of all the variables of interst are near normal with some skews which are the effect of outliers. All the variables are numerical.All the scatterplots (figure 5,6,7,8) show that the linearity condition is met for all the variables. Since the data represnts the whole population (50 states) the independence condition is not relevant.

So conditions are met except the presence of outliers.

cocorrelation tests without removing outliers:

Average school enrollment and two-parents families:

```
cor.test(familyEduDS$ratio_both_parents, familyEduDS$avg_enrollment,
    method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  familyEduDS$ratio_both_parents and familyEduDS$avg_enrollment
## t = -2.7312, df = 49, p-value = 0.008747
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.58088046 -0.09768515
## sample estimates:
##        cor
## -0.3634837
```

```r
ggscatter(familyEduDS, x = "ratio_both_parents", y = "avg_enrollment",
    add = "reg.line", conf.int = TRUE, cor.coef = TRUE, cor.method = "pearson",
    xlab = "ratio of two-parents families", ylab = "avg. enrollment in schools per family",
    color = "red")
```

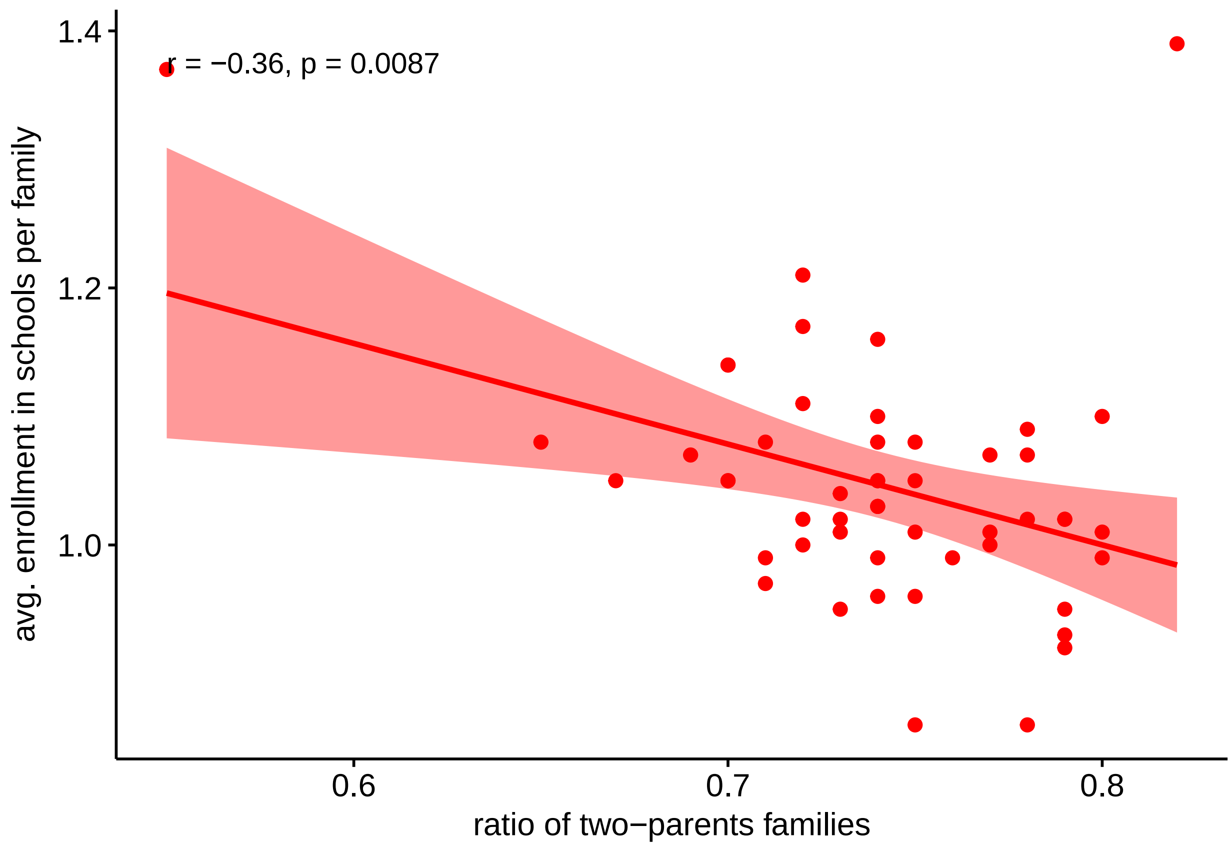


Figure 9.

The test and the figure 9 shows a negative correlation between school enrollment and both-parents families with correlation coefficients (r) of -0.36

Average school enrollment and single parents family:

```r
cor.test(familyEduDS$ratio_single_parents, familyEduDS$avg_enrollment,
    method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  familyEduDS$ratio_single_parents and familyEduDS$avg_enrollment
## t = 2.7312, df = 49, p-value = 0.008747
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.09768515 0.58088046
## sample estimates:
##       cor
## 0.3634837
```

```r
ggscatter(familyEduDS, x = "ratio_single_parents", y = "avg_enrollment",
    add = "reg.line", conf.int = TRUE, cor.coef = TRUE, cor.method = "pearson",
    xlab = "ratio of single-parents families", ylab = "avg. enrollment in schools per family",
    color = "blue")
```

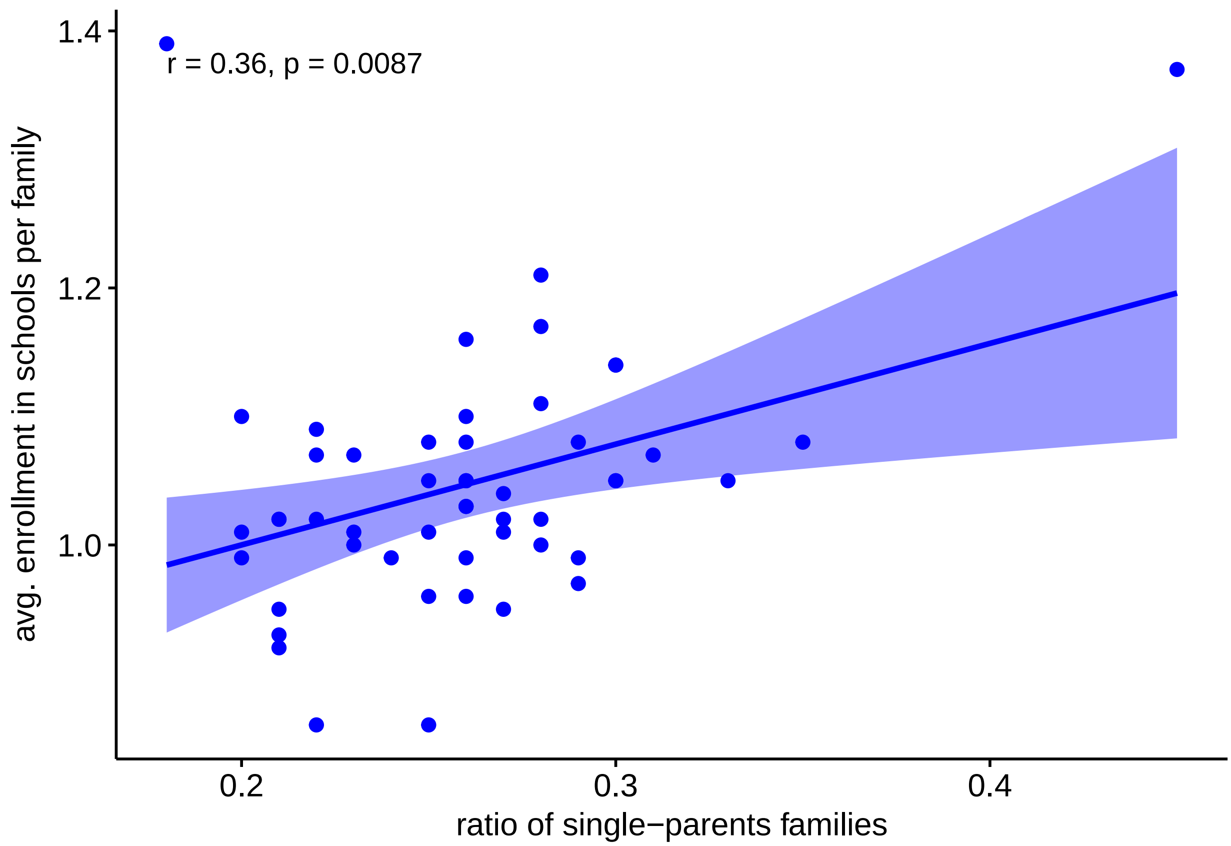


Figure 10.

The test and the figure 10 shows a positive correlation between school enrollment and single-parents families with correlation coefficients (r) of 0.36

Average bachelor degree holders and two-parents families

```r
cor.test(familyEduDS$ratio_both_parents, familyEduDS$avg_bachelor,
    method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  familyEduDS$ratio_both_parents and familyEduDS$avg_bachelor
## t = -0.94704, df = 49, p-value = 0.3483
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3950579  0.1469422
## sample estimates:
##        cor
## -0.1340707
```

```r
ggscatter(familyEduDS, x = "ratio_both_parents", y = "avg_bachelor",
    add = "reg.line", conf.int = TRUE, cor.coef = TRUE, cor.method = "pearson",
    xlab = "ratio of two-parents families", ylab = "avg. bachelor degree holders per family",
    color = "red")
```

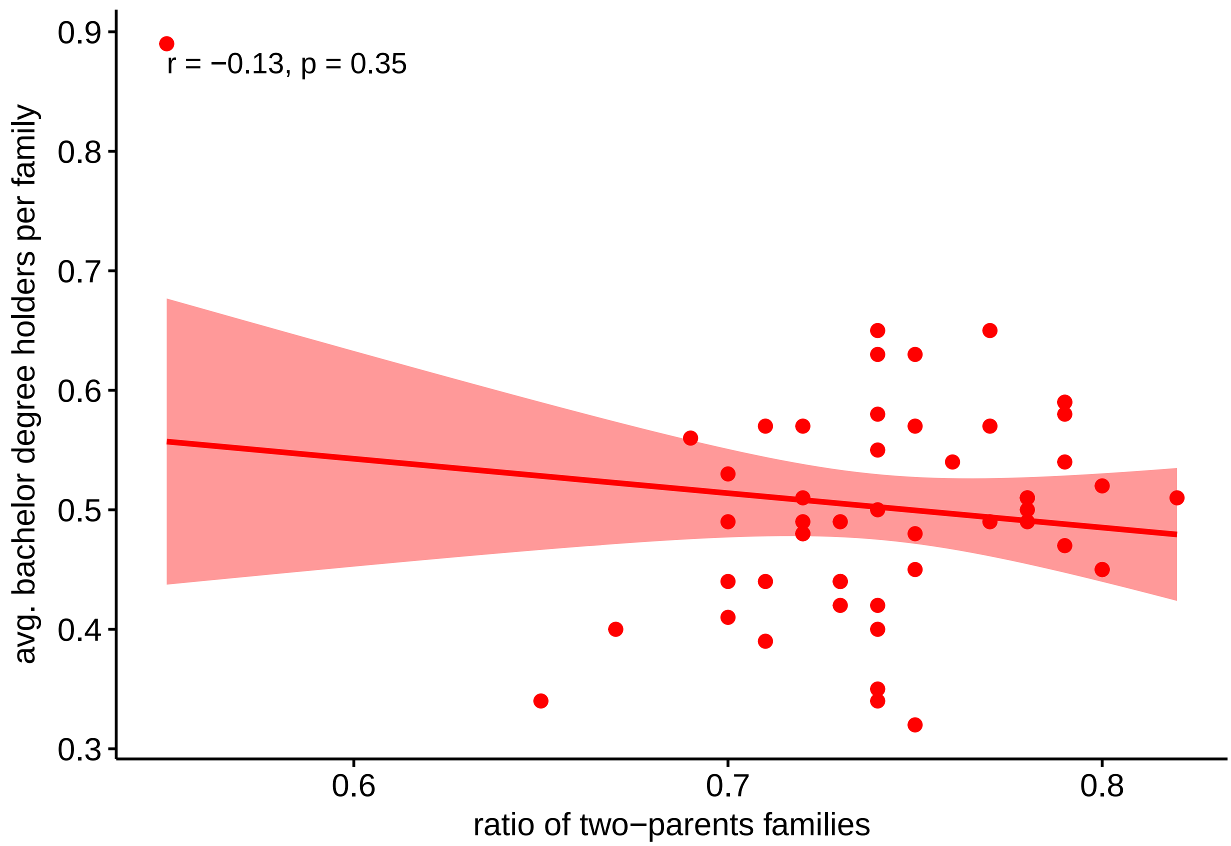


Figure 11.

The test and the figure 11 shows a negative correlation between bachelor degree holders and two-parents families with a correlation coefficients (r) of -0.13

Average bachelor degree holders and single parents families

```r
cor.test(familyEduDS$ratio_single_parents, familyEduDS$avg_bachelor,
    method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  familyEduDS$ratio_single_parents and familyEduDS$avg_bachelor
## t = 0.94704, df = 49, p-value = 0.3483
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1469422  0.3950579
## sample estimates:
##       cor
## 0.1340707
```

```r
ggscatter(familyEduDS, x = "ratio_single_parents", y = "avg_bachelor",
    add = "reg.line", conf.int = TRUE, cor.coef = TRUE, cor.method = "pearson",
    xlab = "ratio of single-parents families", ylab = "average bachelor degree holders per family",
    color = "blue")
```

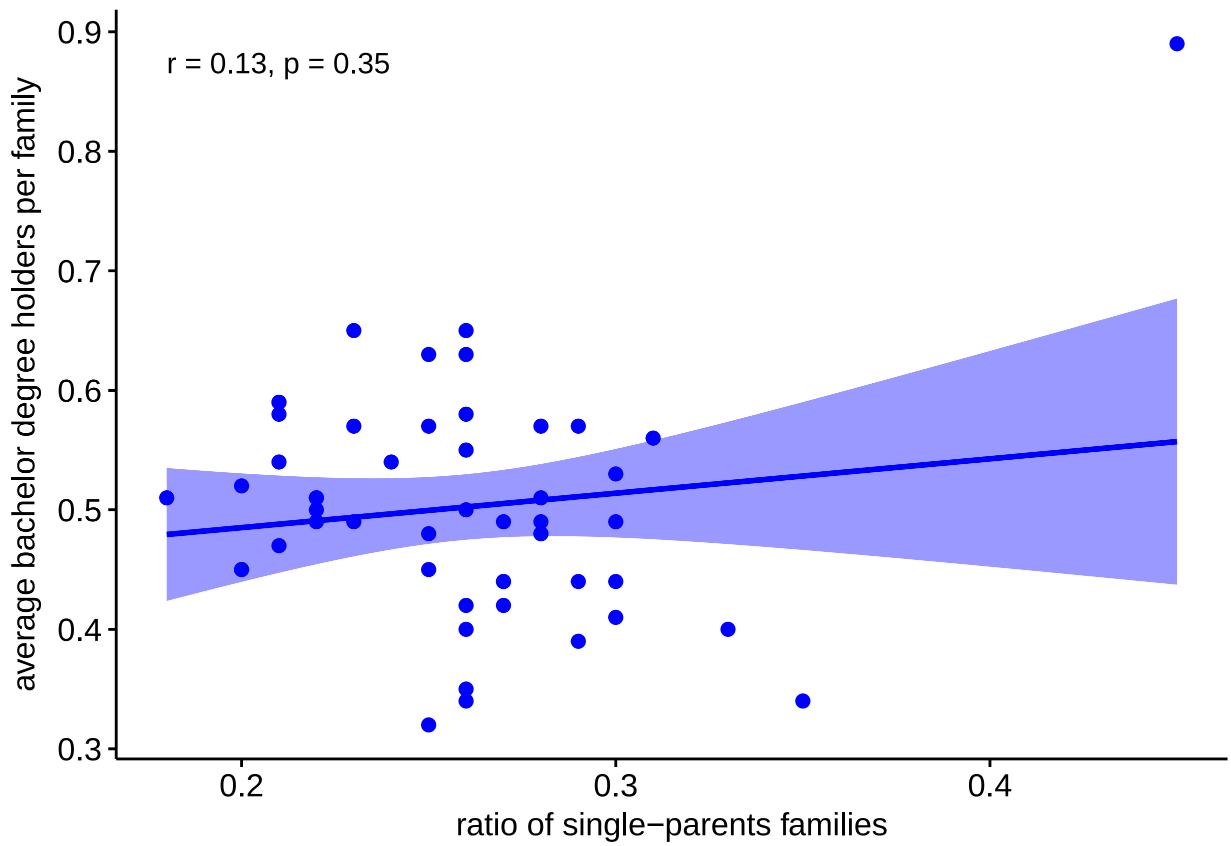


Figure 12.

The test and the figure 12 shows a positive correlation between average bachelor degree holders per family and single-parents families with correlation coefficients (r) of 0.13

So all the above tests show that there are a positive correaltions between single parent families with both school enrollment and average bachelor degree holders i.e. a community with single parent families would have more educated population while two parents families have negative correlations with both measures of education (i.e. school enrollment and number of bachelor degree holders)

cocorrelation tests without outliers:

Since not having outliers is a condition for correlation analysis, all the outliers were removed and the similar tests were done again on the revised data.

Finding outliers:

```r
familyEduDS[(familyEduDS$avg_enrollment < 0.995 - 1.5 * IQR_enrollment) |
    (familyEduDS$avg_enrollment > 1.08 + 1.5 * IQR_enrollment), ]
```

```
##                   states total_family married_couple_family
## 5           California      8732734               6245351
## 9  District of Columbia       118737                 65383
## 20                 Maine       347579                270147
## 45                  Utah       680007                554555
## 49         West Virginia       479803                361652
##    husband_only_family wife_only_family school_enrollment bachelor_degree
## 5               759047          1728336          10579176         5002596
## 9                10502            42852            162835          105880
## 20               24446            52986            299595          178375
## 45               38394            87058            942989          347460
## 49               33962            84189            410745          152377
##    single_parent_family avg_enrollment avg_bachelor ratio_both_parents
## 5               2487383           1.21         0.57               0.72
## 9                 53354           1.37         0.89               0.55
## 20                77432           0.86         0.51               0.78
## 45               125452           1.39         0.51               0.82
## 49               118151           0.86         0.32               0.75
##    ratio_single_parents
## 5                  0.28
## 9                  0.45
## 20                 0.22
## 45                 0.18
## 49                 0.25
```

```r
familyEduDS[(familyEduDS$avg_bachelor < 0.44 - 1.5 * IQR_bachelor) |
    (familyEduDS$avg_bachelor > 0.565 + 1.5 * IQR_bachelor), ]
```

```
##                  states total_family married_couple_family
## 9 District of Columbia       118737                 65383
##   husband_only_family wife_only_family school_enrollment bachelor_degree
## 9               10502            42852            162835          105880
##   single_parent_family avg_enrollment avg_bachelor ratio_both_parents
## 9                53354           1.37         0.89               0.55
##   ratio_single_parents
## 9                 0.45
```

```r
familyEduDS[(familyEduDS$ratio_both_parents < 0.72 - 1.5 * IQR_two_parents) |
    (familyEduDS$ratio_both_parents > 0.775 + 1.5 * IQR_two_parents),
    ]
```

```
##                  states total_family married_couple_family
## 9 District of Columbia       118737                 65383
##   husband_only_family wife_only_family school_enrollment bachelor_degree
## 9               10502            42852            162835          105880
##   single_parent_family avg_enrollment avg_bachelor ratio_both_parents
## 9                53354           1.37         0.89               0.55
##   ratio_single_parents
## 9                 0.45
```

```
familyEduDS[(familyEduDS$ratio_single_parents < 0.225 - 1.5 * IQR_single_parents) |
    (familyEduDS$ratio_single_parents > 0.28 + 1.5 * IQR_single_parents),
    ]
```

```
##                  states total_family married_couple_family
## 9 District of Columbia       118737                 65383
##   husband_only_family wife_only_family school_enrollment bachelor_degree
## 9               10502            42852            162835          105880
##   single_parent_family avg_enrollment avg_bachelor ratio_both_parents
## 9                53354           1.37         0.89               0.55
##   ratio_single_parents
## 9                 0.45
```

District of Columbia is the outlier in all the variables while California, Utah, Maine and West Virginia are outliers only in scholl enrollment variables.

Outliers were removed and two seperate datasets were created:

```
EduDS <- familyEduDS[-c(9), ]
EduDS_enroll <- familyEduDS[-c(5, 9, 20, 45, 49), ]
```

correlation test:

Average school enrollment and two-parents family

```
cor.test(EduDS_enroll$ratio_both_parents, EduDS_enroll$avg_enrollment,
    method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  EduDS_enroll$ratio_both_parents and EduDS_enroll$avg_enrollment
## t = -2.4519, df = 44, p-value = 0.01825
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.5787578 -0.0627270
## sample estimates:
##        cor
## -0.3467116
```

```
ggscatter(EduDS_enroll, x = "ratio_both_parents", y = "avg_enrollment",
    add = "reg.line", conf.int = TRUE, cor.coef = TRUE, cor.method = "pearson",
    xlab = "ratio of two-parents families", ylab = "avg. enrollment in schools per family",
    color = "red")
```

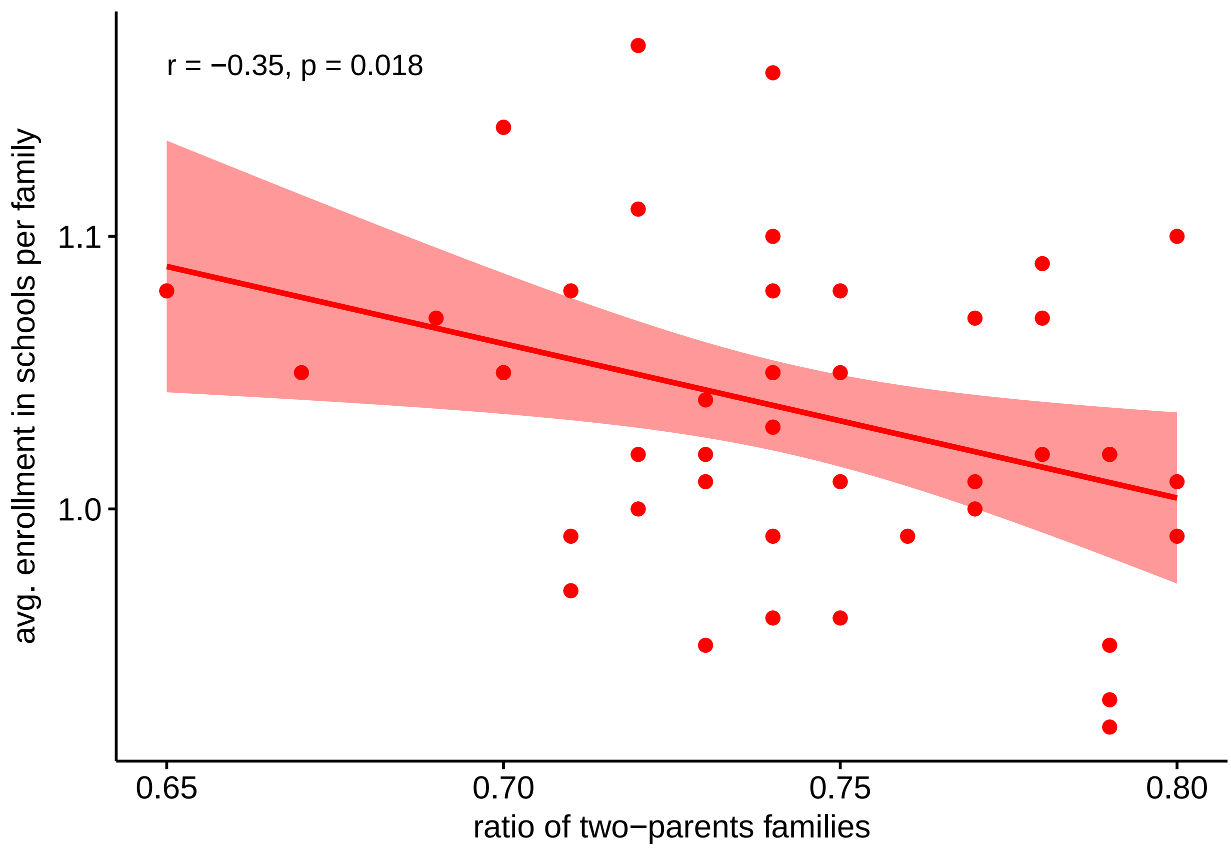


Figure 13.

The test and the figure 13 shows a negative correlation between school enrollment and two-parents families with correlation coefficients (r) of -0.35

Average school enrollment and single parents family

```
cor.test(EduDS_enroll$ratio_single_parents, EduDS_enroll$avg_enrollment,
    method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  EduDS_enroll$ratio_single_parents and EduDS_enroll$avg_enrollment
## t = 2.4519, df = 44, p-value = 0.01825
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.0627270 0.5787578
## sample estimates:
##       cor
## 0.3467116
```

```
ggscatter(EduDS_enroll, x = "ratio_single_parents", y = "avg_enrollment",
    add = "reg.line", conf.int = TRUE, cor.coef = TRUE, cor.method = "pearson",
    xlab = "single-parents families", ylab = "avg. enrollment in schools per family",
    color = "blue")
```

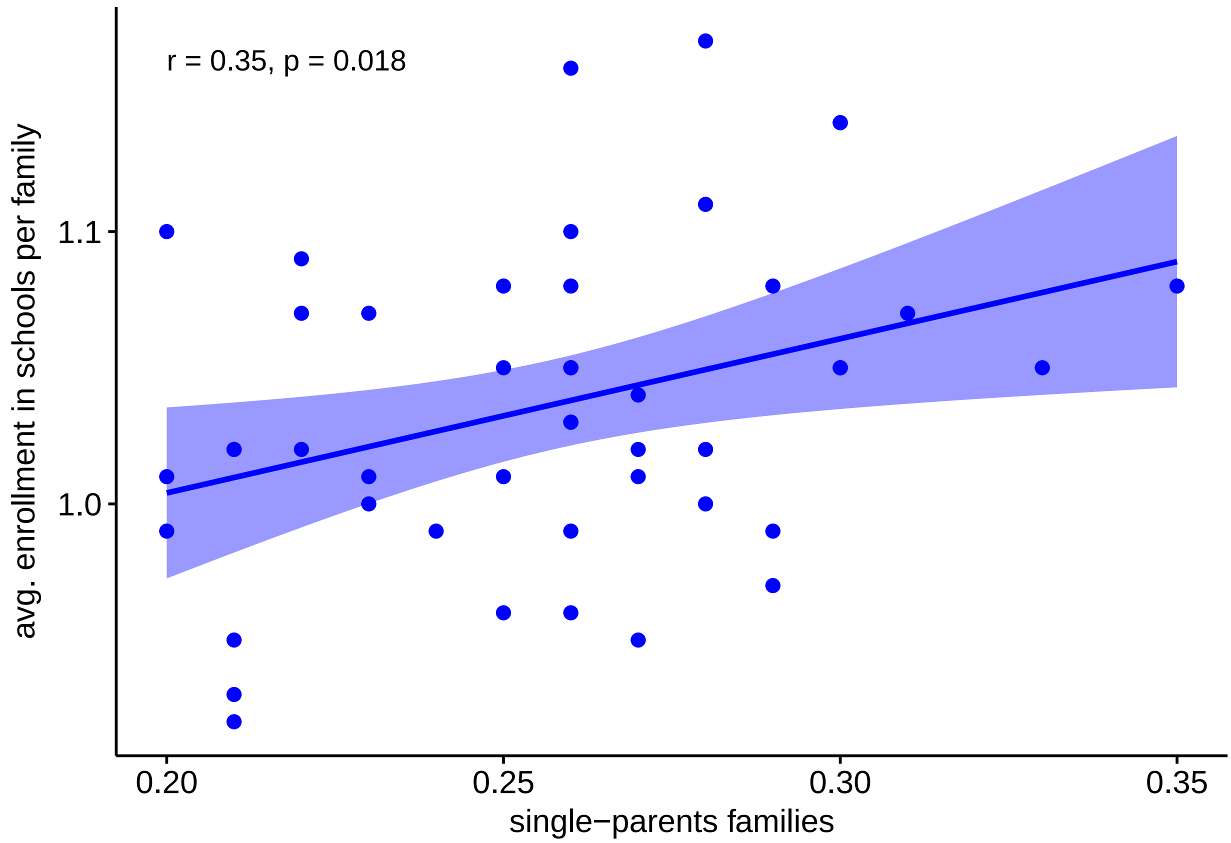


Figure 14.

The test and the figure 14 shows a positive correlation between school enrollment and single-parents families with correlation coefficients (r) of 0.35

Average bachelor degree holders and two-parents family

```
cor.test(EduDS$ratio_both_parents, EduDS$avg_bachelor, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  EduDS$ratio_both_parents and EduDS$avg_bachelor
## t = 2.3108, df = 48, p-value = 0.02518
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.04173024 0.54661051
## sample estimates:
##       cor
## 0.3164028
```

```
ggscatter(EduDS, x = "ratio_both_parents", y = "avg_bachelor", add = "reg.line",
    conf.int = TRUE, cor.coef = TRUE, cor.method = "pearson", xlab = "both-parents families",
    ylab = "avg. bachelor degree holders per family", color = "red")
```

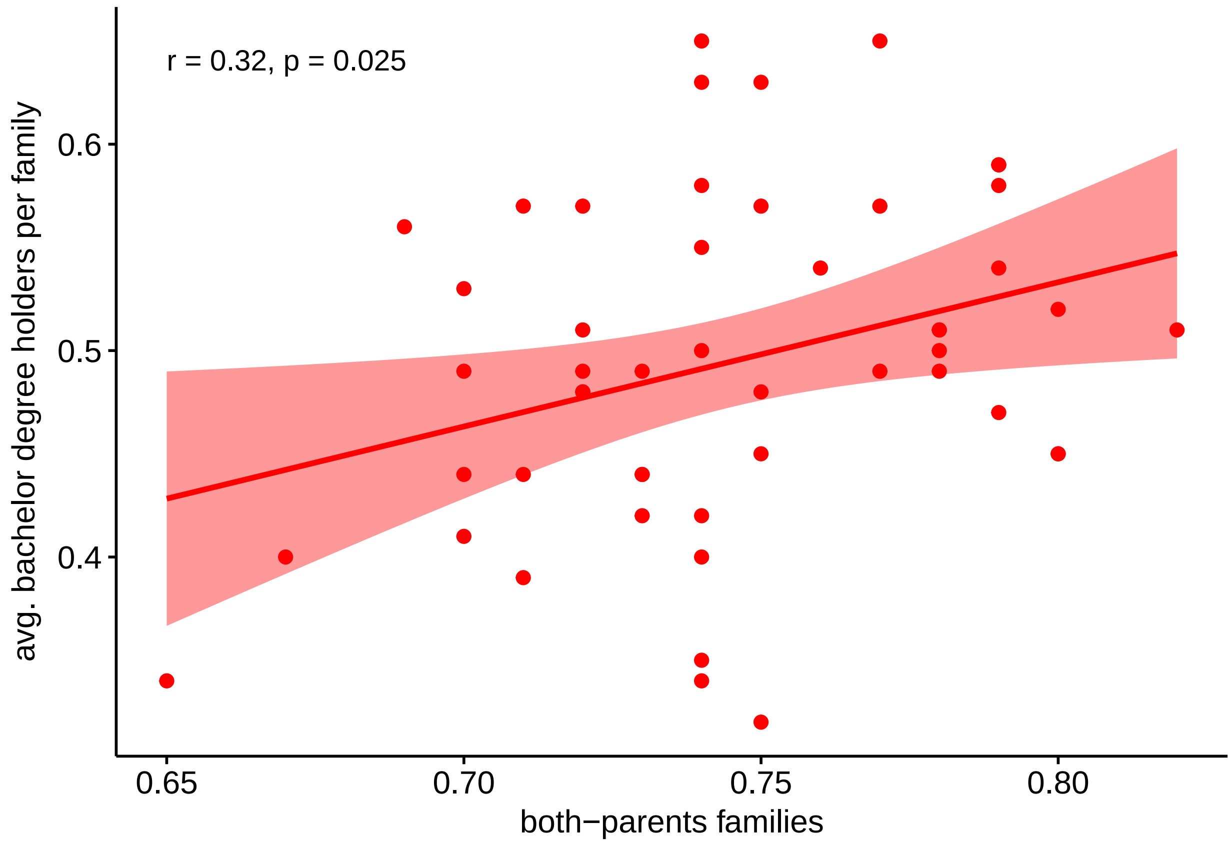


Figure 15.

The test and the figure 15 shows a positive correlation between bachelor degree holders and two-parents families with correlation coefficients (r) of 0.32

Average bachelor degree holders and single parents family

```
cor.test(EduDS$ratio_single_parents, EduDS$avg_bachelor, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  EduDS$ratio_single_parents and EduDS$avg_bachelor
## t = -2.3108, df = 48, p-value = 0.02518
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.54661051 -0.04173024
## sample estimates:
##        cor
## -0.3164028
```

```
ggscatter(EduDS, x = "ratio_single_parents", y = "avg_bachelor", add = "reg.line",
    conf.int = TRUE, cor.coef = TRUE, cor.method = "pearson", xlab = "single-parents families",
    ylab = "avg. bachelor degree holders per family", color = "blue")
```

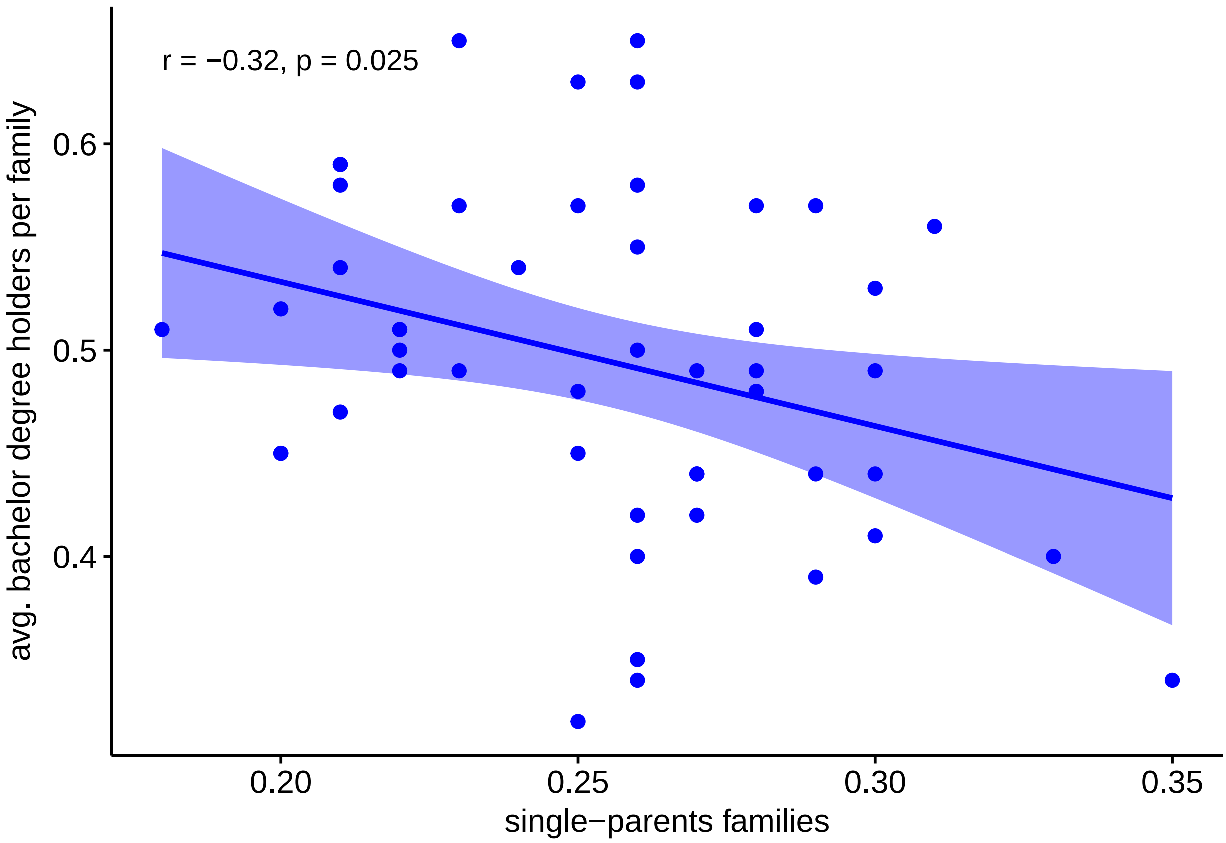


Figure 16.

The test and the figure 16 shows a negative correlation between average bachelor degree holders and single-parents families with correlation coefficients (r) of -0.32

CONCLUSION

Since the data represents the whole population (50 States) the statistical significance is meaningless here. There is no standard error and the p-value is irrelevant. Therefore the correlation coefficients found here represent the poulation correlation coefficients. So after removing the ouliers the result show that:

two-parents families have a positive correlation with population with graduate (bachelor) degree holders but have a negative coorelation with school enrollment. Single-parents familes have the same correlation but in the opposite directions.

Only 12.25% (0.35ˆ2) variablity in average scool enrollment and only 10.24% (0.32ˆ2) variablity in average bachelor degree holders can be explained by the family structure variables.

So there is an impact of family structures on the education of people and we reject the Null hypothesis.

Further analysis:

If the purpose of the analysis is to predict the affect of family structures on education in future cases , then the dataset may be considered as the smaple dataset from a population of an infinite cases of the future. Asuming the above, a regression analysis was done between average bachelor degree holders per family and the ratio of two-parents families

```r
fit_bachelor <- lm(avg_bachelor ~ ratio_both_parents, data = EduDS)
```

Histogram of residuals

```r
hist(fit_bachelor$residuals, col = "green")
```

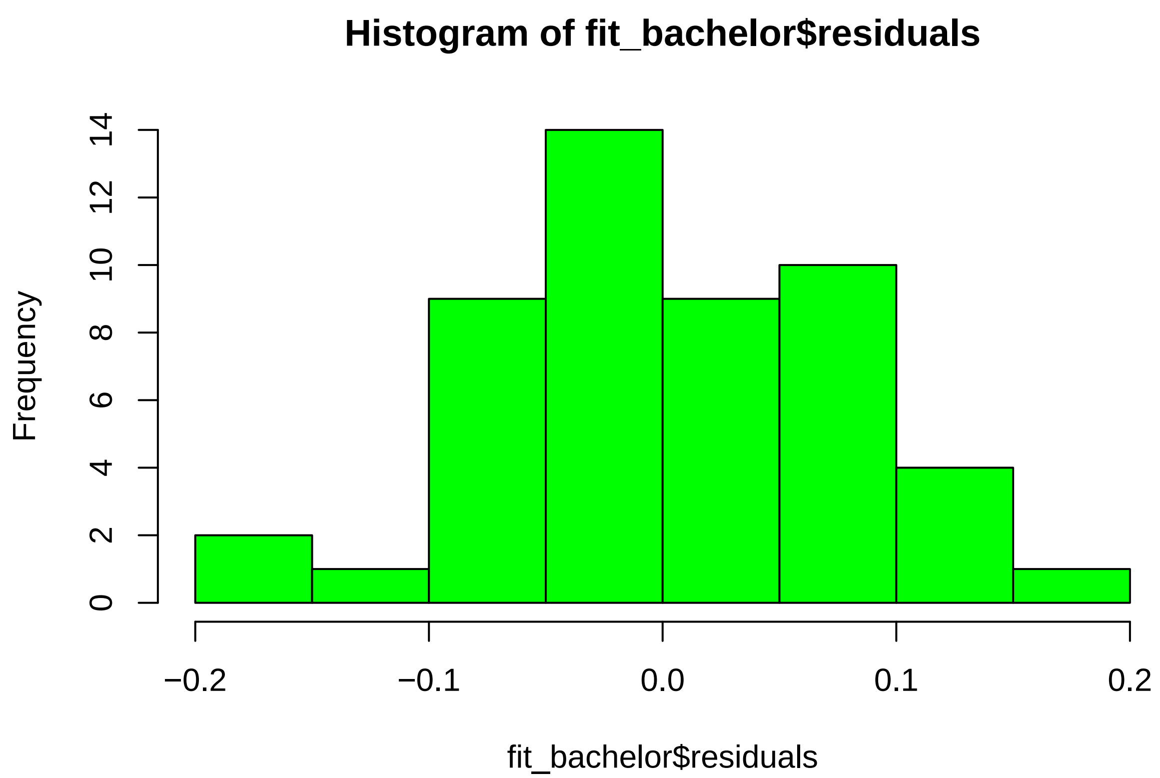


Figure 17.

```r
qqnorm(fit_bachelor$residuals)
qqline(fit_bachelor$residuals, col = "blue")
```

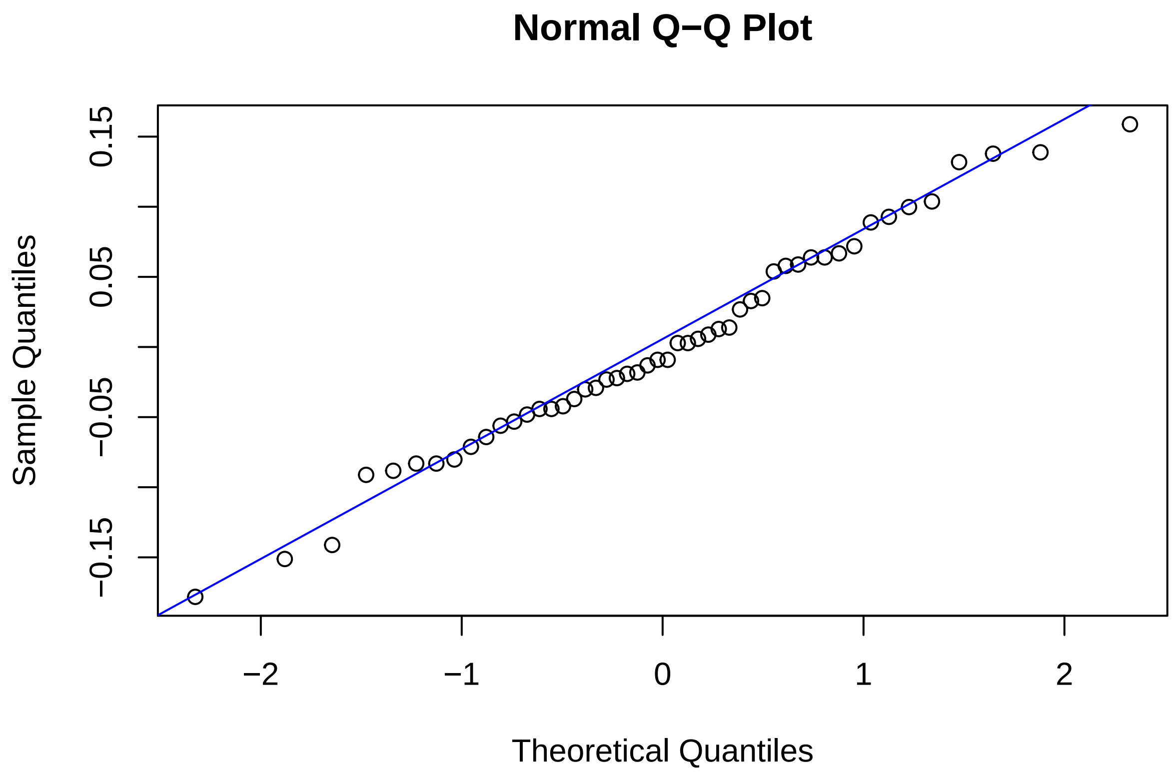


Figure 18.

```r
plot(fit_bachelor$residuals ~ EduDS$ratio_both_parents)
abline(h = 0, lt = 2, col = "red")
```

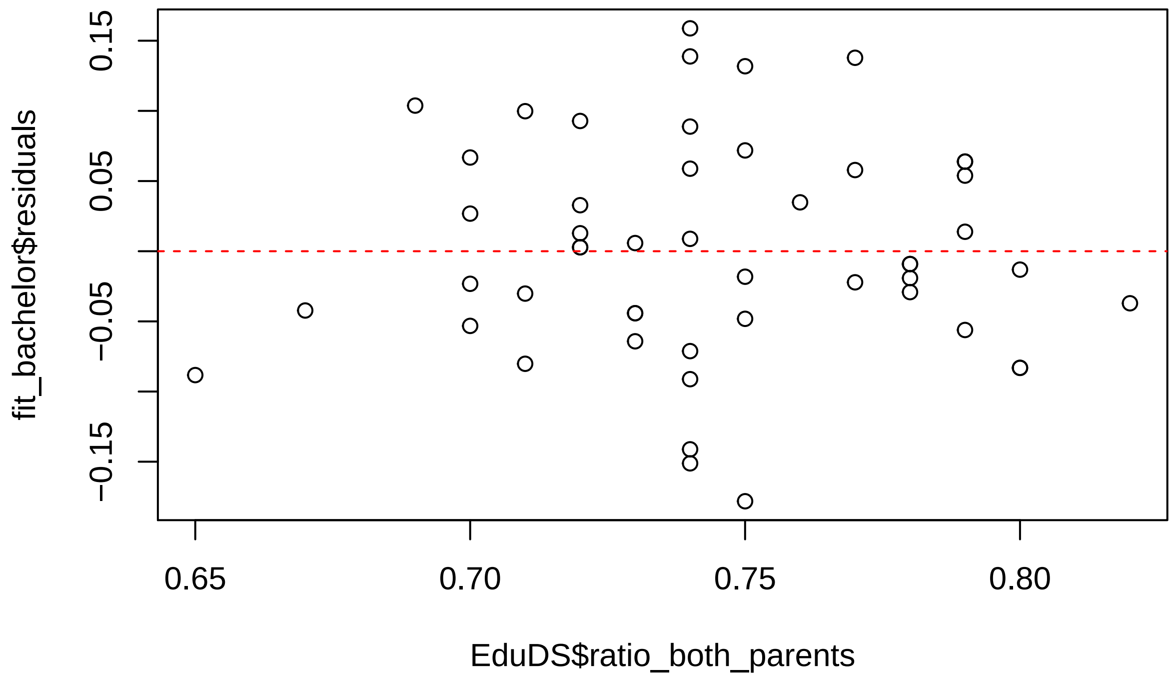


Figure 19.

Conditions check for simple regression analysis:

a. Linearity check: Figure 19 (scatterplot) shows a linear trend of the data. so linearity condtion is satisfied.

b. Nearly normal residuals: Both the histogram (Figure 17) and qqplot and qqline plots (Figure 18) show that the residuals are nearly normally distributed. So the condition is also met.

c. constant variability: The figure 19 also shows the residuals are scattered around the horizontal line almost at a constant variability, so this condition is also satisfied.

d. Independent observations: If we consider the datset as the sample from an infinite population of future cases we can assume that the sample size is less than 10% of the population, so independence is reasonable.

Therefor all the conditions of simple regression analysis are met.

```
summary(fit_bachelor)
```

```
##
## Call:
## lm(formula = avg_bachelor ~ ratio_both_parents, data = EduDS)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.178175 -0.047180 -0.009147  0.058573  0.158816
##
## Coefficients:
```

```
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.02612    0.22552  -0.116   0.9083
## ratio_both_parents  0.69906    0.30252   2.311   0.0252 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07751 on 48 degrees of freedom
## Multiple R-squared:  0.1001, Adjusted R-squared:  0.08136
## F-statistic:  5.34 on 1 and 48 DF,  p-value: 0.02518
```

The P Value is smaller than .05 and the Coefficients of both parents is greater than zero. So two-parent family structure does have an affect on the average number of bachelor degree holders per family. Since the R-squared value is 0.10, only 10% variablity in the average bachelor degree holders can be explained by the ratio of two-parents families.

So family structures have impacts on education, therefore null hypothesis is rejected.