M2 Homework Assignment

Joshua Conte

September 24, 2017

1 M2 Homework Assignment

R studio was configured with the following parameters before beginning the project:

```
# clears the console in RStudio
cat("\014")
```

```
# clears environment
rm(list = ls())

# Set working directory
setwd("C:/R/DA5030/module_02")

# Load required packages
require(ggplot2)
```

1.1 load the file M01_quasi_twitter.csv.

I opened the data using read.csv2 and changed the data from integer to numeric so I could analyze it with R.

Below is my R code:

```
# Some csv files are really big and take a while to open.  This command checks to
# see if it is already opened, if it is, it does not open it again.
# I also omitted the first column
if (!exists("df1")) {
  df1 <-
    read.csv2(
      'M01_quasi_twitter.csv',
      sep = ",",
      stringsAsFactors = FALSE,
      row.names = NULL,
      header = TRUE
    )
}

# Check to make sure the data is in numeric form for analysis:
sapply(df1, class)
```

```
##             screen_name        created_at_month          created_at_day
##             "character"               "integer"               "integer"
##         created_at_year                 country                location
##               "integer"             "character"             "character"
##           friends_count         followers_count          statuses_count
##               "integer"               "integer"               "integer"
##        favourites_count        favourited_count                 dob_day
##               "integer"               "integer"               "integer"
##                dob_year               dob_month                  gender
##               "integer"               "integer"             "character"
## mobile_favourites_count mobile_favourited_count               education
##               "integer"               "integer"               "integer"
##              experience                     age                    race
##               "integer"               "integer"             "character"
##                    wage         retweeted_count           retweet_count
##             "character"               "integer"               "integer"
```

```
##                  height
##               "integer"
```

```
# change int to numeric
df1[2:4] <- sapply(df1[2:4], as.numeric)
df1[7:14] <- sapply(df1[7:14], as.numeric)
df1[16:20] <- sapply(df1[16:20], as.numeric)
df1[22:25] <- sapply(df1[22:25], as.numeric)

# Confirm changes with info
str(df1)
```

```
## 'data.frame':    21916 obs. of  25 variables:
##  $ screen_name            : chr  "CNN" "osbrFe" "WSJ" "ninc" ...
##  $ created_at_month       : num  2 11 4 3 4 2 7 5 1 1 ...
##  $ created_at_day         : num  9 21 1 24 23 9 15 23 23 13 ...
##  $ created_at_year        : num  2007 2009 2007 2007 2009 ...
##  $ country                : chr  "USA" "India" "India" "USA" ...
##  $ location               : chr  "Miami Florida" "Mumbai" "Bangalore" "North Carolina" ...
##  $ friends_count          : num  1087 5210 1015 338 641 ...
##  $ followers_count        : num  22187643 6692814 6257020 3433218 2929559 ...
##  $ statuses_count         : num  60246 93910 118465 78082 93892 ...
##  $ favourites_count       : num  1122 3825 1143 0 226 ...
##  $ favourited_count       : num  105005 40487 87968 25943 32589 ...
##  $ dob_day                : num  29 24 4 22 9 1 2 6 15 26 ...
##  $ dob_year               : num  1999 1991 1997 1998 1963 ...
##  $ dob_month              : num  4 10 3 8 11 1 11 10 2 9 ...
##  $ gender                 : chr  "female" "female" "male" "male" ...
##  $ mobile_favourites_count: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ mobile_favourited_count: num  0 5032191 0 0 0 ...
##  $ education              : num  8 15 9 9 13 15 14 10 11 12 ...
##  $ experience             : num  0 0 0 44 24 21 31 0 27 20 ...
##  $ age                    : num  29 0 32 40 45 14 27 31 34 40 ...
##  $ race                   : chr  "white" "white" "white" "white" ...
##  $ wage                   : num  16.3 17.9 15.7 7 17.9 ...
##  $ retweeted_count        : num  1 1 2 0 1 2 1 2 0 0 ...
##  $ retweet_count          : num  30 6 65 8 7 64 13 14 15 10 ...
##  $ height                 : num  156 162 168 180 162 158 160 178 156 173 ...
```

1.2 Plot Two Linear Models

To find two significant linear models, I made a new data frame with all of the numeric data and used pairs to analyze all of the data. This produces scatter plots of all of the data. This process took a really long time, so I commented out the graphing part and attached the image below:

```
df2 <-
  data.frame(
    df1$created_at_month,
    df1$created_at_day,
    df1$created_at_year,
    df1$friends_count,
    df1$followers_count,
    df1$statuses_count,
    df1$favourites_count,
```

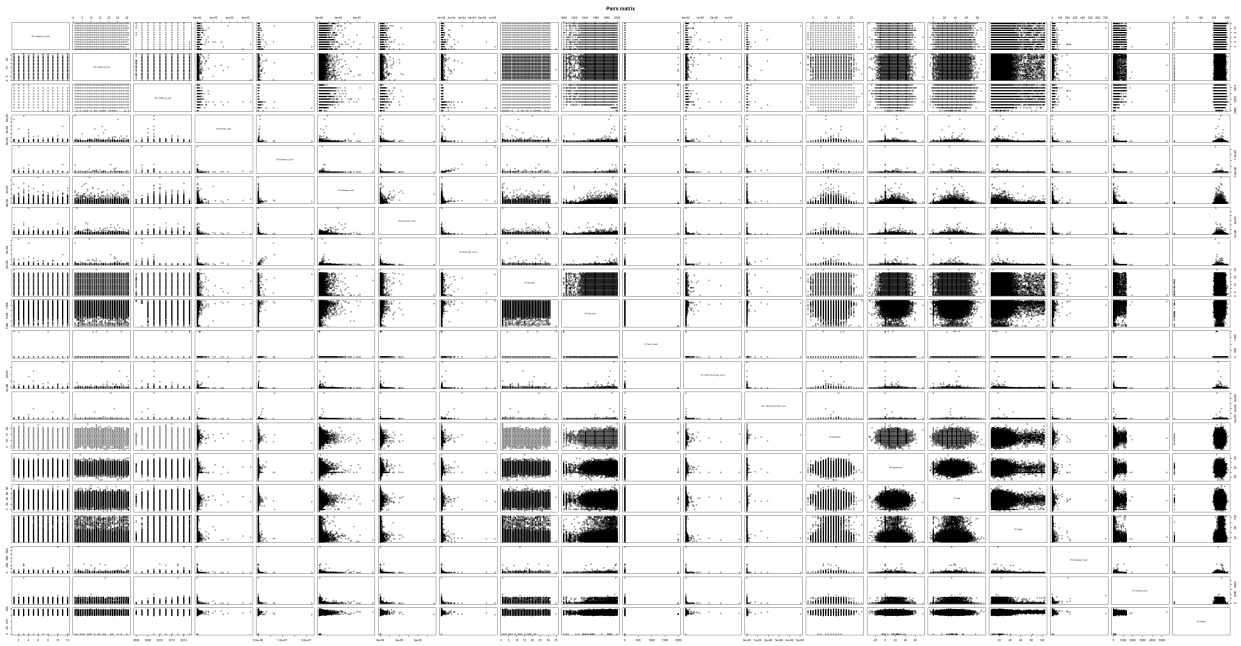


Figure 1: Pairs Image

```
    df1$favourited_count,
    df1$dob_day,
    df1$dob_year,
    df1$dob_month,
    df1$mobile_favourites_count,
    df1$mobile_favourited_count,
    df1$education,
    df1$experience,
    df1$age,
    df1$wage,
    df1$retweeted_count,
    df1$retweet_count,
    df1$height
  )
#pairs(df2, labels = colnames(df2), main = "Pairs matrix", pch = 21)
```

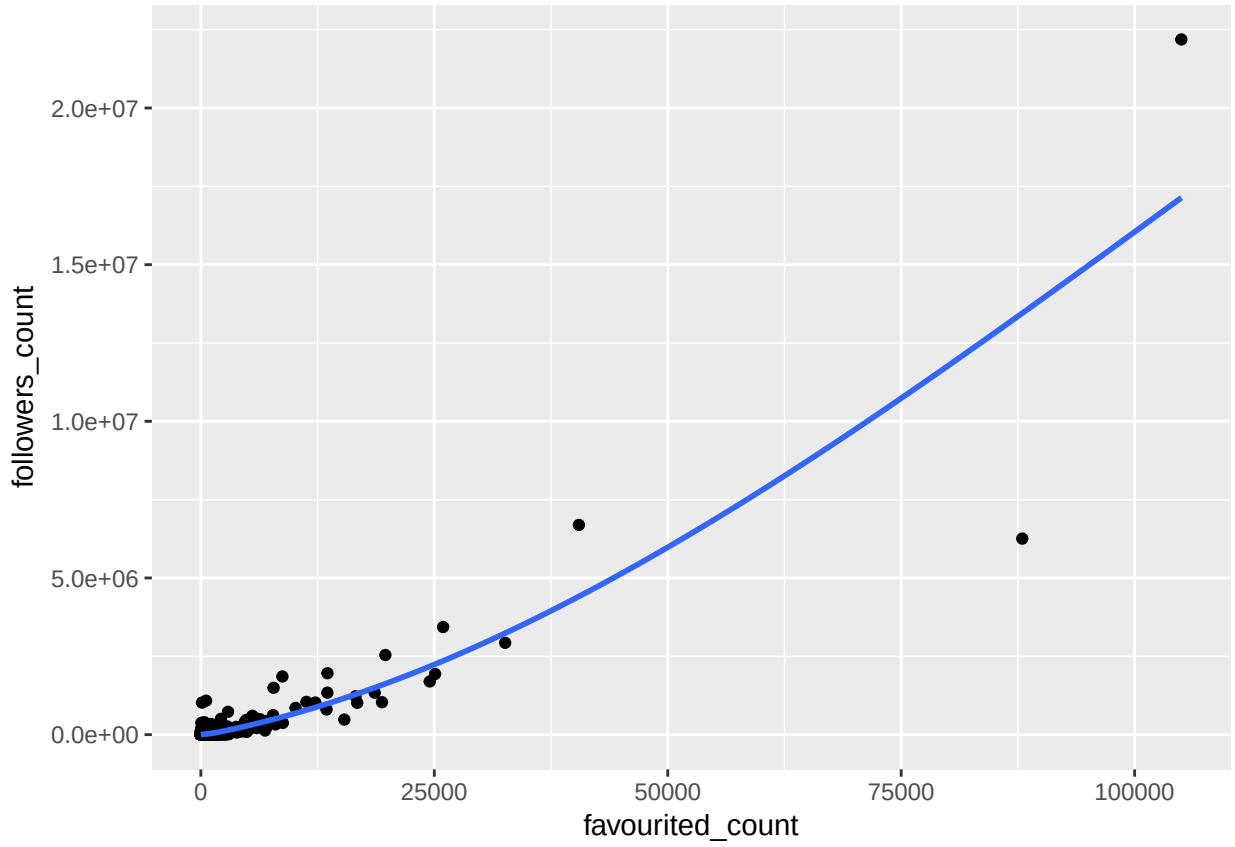
That, gives us plenty to look at, doesn't it?

1.2.1 Model 1

Model 1 looks at the relationship between favorited count and followers count, to see if the followers count matters in the favorited count, I plotted the graph below:

```
p <-
  qplot(favourited_count,
        followers_count,
        data = df1,
        position = "jitter")
p + stat_smooth()
```

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



It looks like there is some correlation between the two so I can fit the linear model.

1.2.1.1 Fit the Linear Model

lm is used to fit linear models. It can be used to carry out regression, single stratum analysis of variance and analysis of covariance.

To begin I need a response variable (favourited_count) and the predictor (followers_count) and then I need to save this model for further analysis:

```
# First I am going to make a data frame with this info
dfFavFol <-
  subset(df1, select = c(favourited_count, followers_count))

m_favourited_followers <-
  lm(favourited_count ~ followers_count, data = dfFavFol)
m_favourited_followers
```

```
##
## Call:
## lm(formula = favourited_count ~ followers_count, data = dfFavFol)
##
## Coefficients:
##     (Intercept)  followers_count
##       57.777272         0.005882
```

Which translates in to the equations below:

$$favoritedcount = 57.777272 + 0.005882x_{followerscount} + \varepsilon$$

1.2.1.2 P-value, t-statistic and standard error:

Below is the analysis of the linear model:

```
summary(m_favourited_followers)
```

```
##
## Call:
## lm(formula = favourited_count ~ followers_count, data = dfFavFol)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -25553    -57    -52    -31  51109
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     5.778e+01  3.539e+00   16.33   <2e-16 ***
## followers_count 5.882e-03  2.072e-05  283.87   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 523.6 on 21914 degrees of freedom
## Multiple R-squared:  0.7862, Adjusted R-squared:  0.7862
## F-statistic: 8.058e+04 on 1 and 21914 DF,  p-value: < 2.2e-16
```

This returns significant information that the t value is significant, there is not much standard error and the small p value, the probability is very significant. This tells me that followers count matters in the favorited count.

Next I move onto an anova.

1.2.1.3 Anova

Analysis of variance (ANOVA) is a collection of statistical models used to analyze the differences among group means and their associated procedures (such as "variation" among and between groups), developed by statistician and evolutionary biologist Ronald Fisher.

An F-test is any statistical test in which the test statistic has an F-distribution under the null hypothesis. Like a t-statistic, or a p-value it provides an estimate of whether one should accept or reject the null hypothesis. The F-test is sensitive to non-normality (as is a t-statistic) but is appropriate under the assumptions of normality and homoscedasticity.

```
anova(m_favourited_followers)
```

```
## Analysis of Variance Table
##
## Response: favourited_count
##                     Df     Sum Sq    Mean Sq F value    Pr(>F)
## followers_count      1 2.2093e+10 2.2093e+10   80582 < 2.2e-16 ***
## Residuals        21914 6.0082e+09 2.7417e+05
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From this output, the f value is high and the p-value is less than 0.05 (which tells me the F value is significant), therefore I can reject the null hypothesis of no differences between the data (similar to the above results).

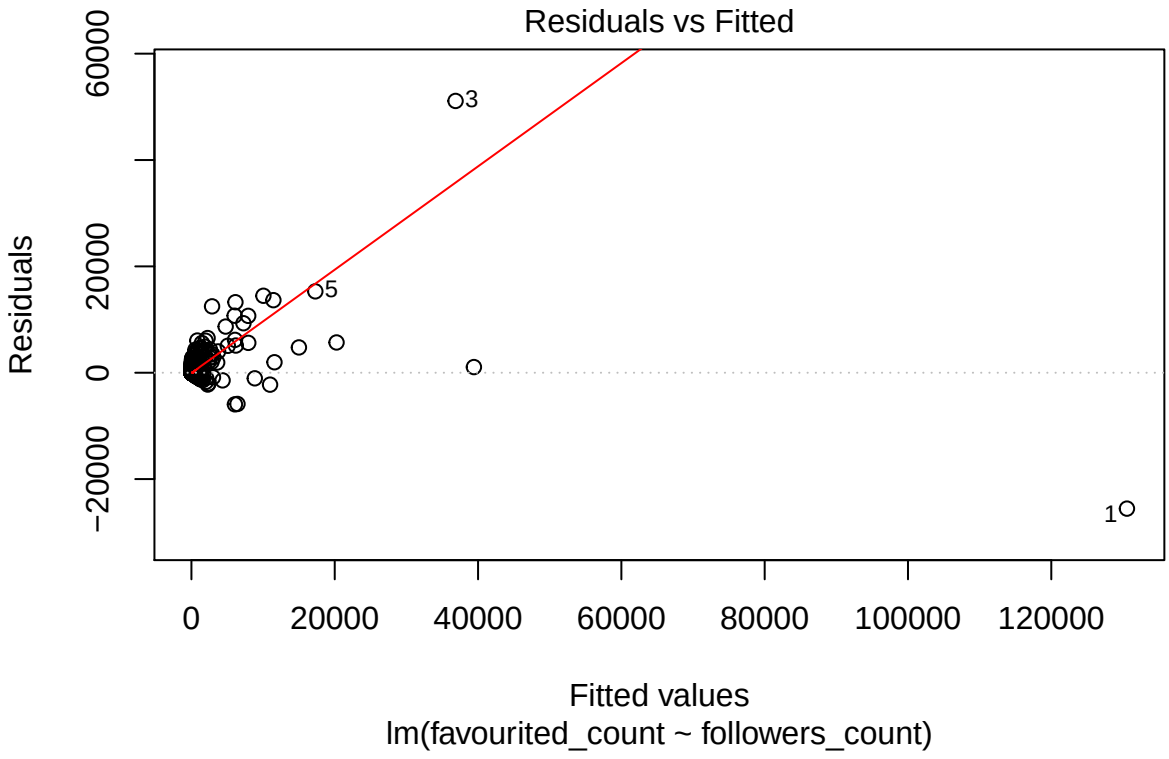
1.2.1.4 Generating the residual plots

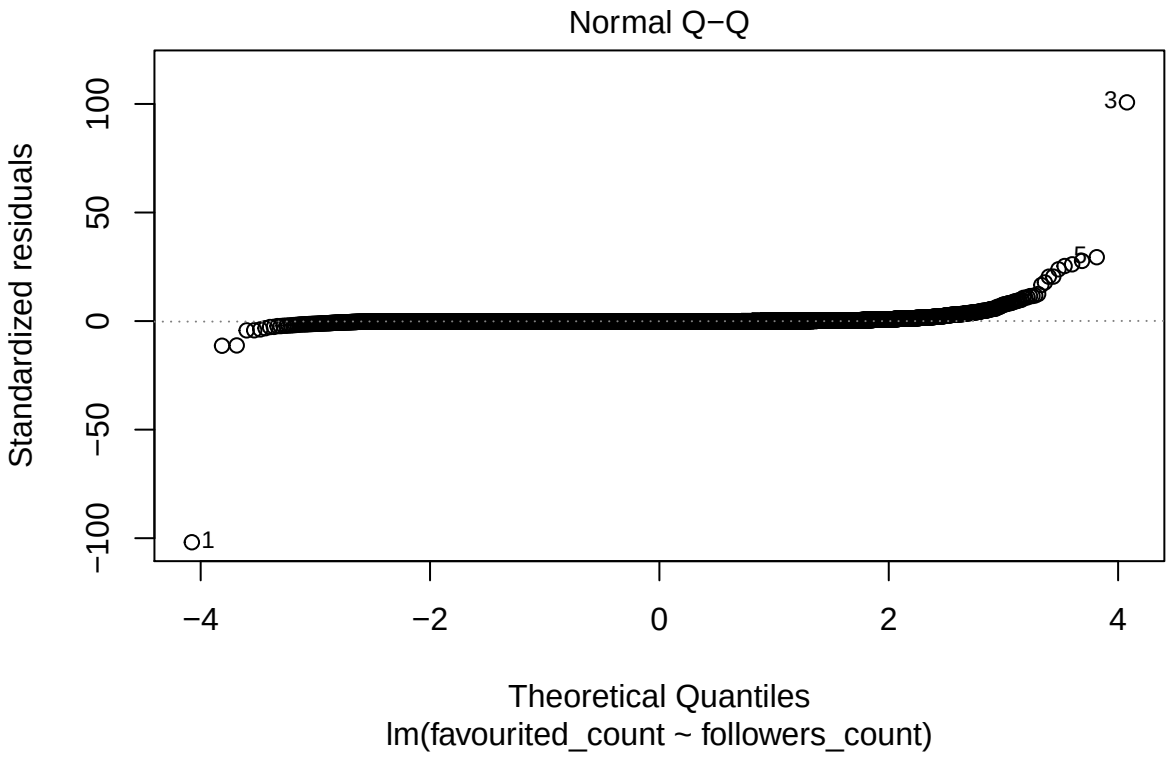
The error term has the following assumptions:

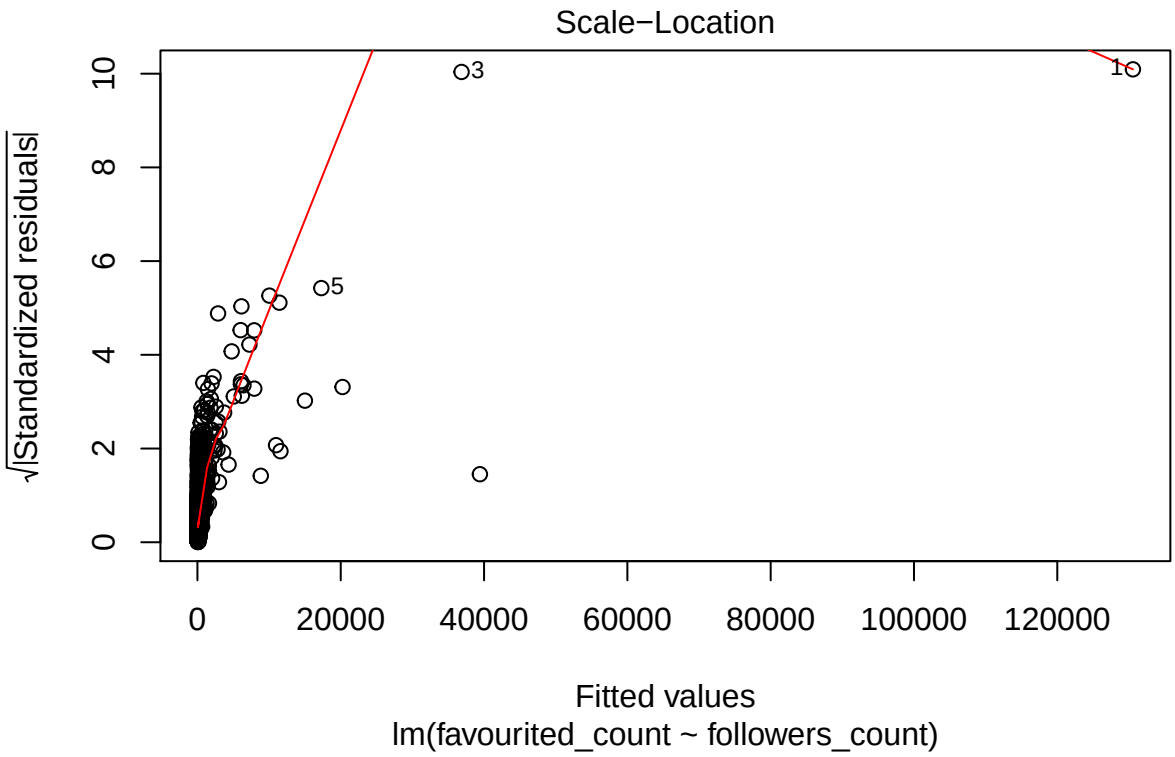
- Have mean zero; otherwise the forecasts will be systematically biased.
- Statistical independence of the errors (in particular, no correlation between consecutive errors in the case of time series data).
- Homoscedasticity (constant variance) of the errors.
- Normality of the error distribution.

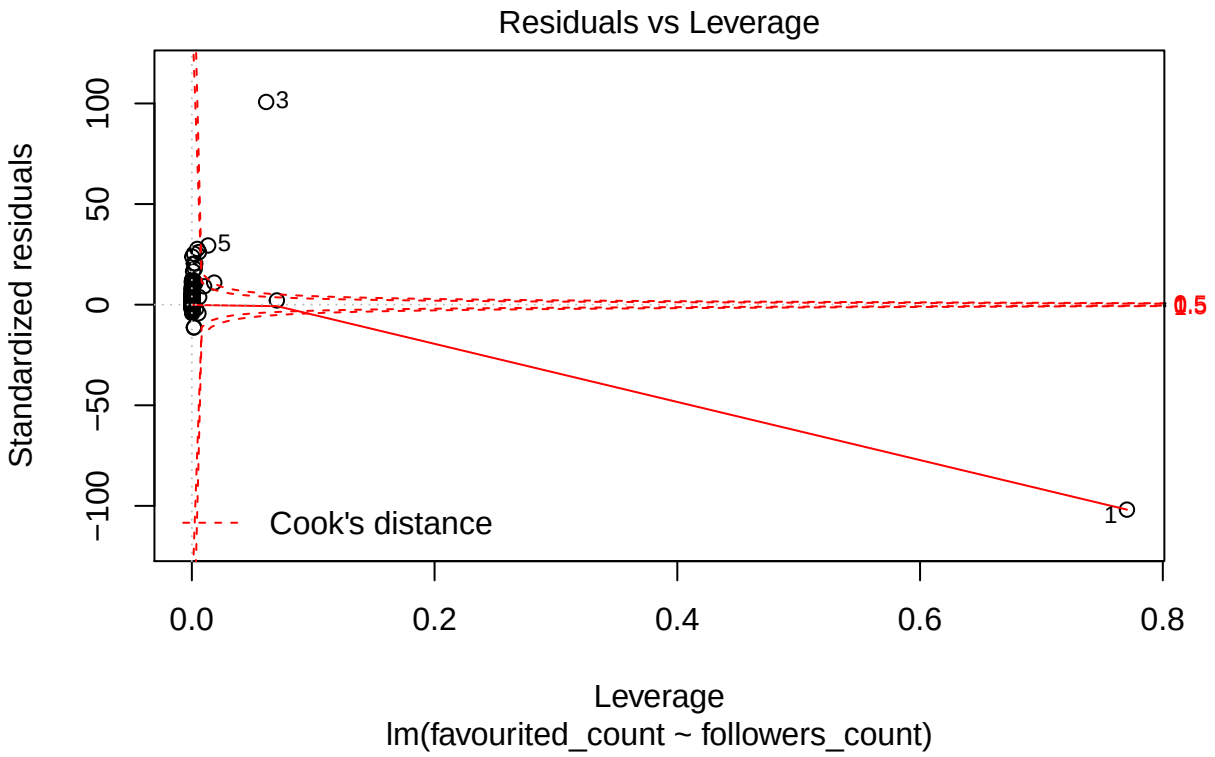
This function will plot the graphs: Residuals vs Fitted, Residuals vs Normal, Standardized Residuals vs Fitted Values, Residuals vs Leverage:

```
plot(m_favourited_followers)
```




Normal Q−Q
Standardized residuals
100
50
0
−50
−100
3
5
1
−4
−2
0
2
4
Theoretical Quantiles
lm(favourited_count ~ followers_count)


Scale−Location
√|Standardized residuals|
0
2
4
6
8
10
3
1
5
0
20000
40000
60000
80000
100000
120000
Fitted values
lm(favourited_count ~ followers_count)



Residuals vs Fitted: The residuals and the fitted values should be uncorrelated in a homoscedastic linear model with normally distributed errors. There should not be a dependency between the residuals and the fitted values. It's hard to tell with this plot, but there is some indication that there is some dependency.

Residuals vs Normal: This is a Q-Q plot to check if the residuals are normal. The ends of the plot do not look so good for this data.

Standardized Residuals vs Fitted Values: standardized residuals means every residual plot you look at with any model is on the same standardized y-axis. This makes it easier to compare many residual plots. This process is also called studentizing (after William Sealey Gosset, who wrote under the pseudonym Student).

Residuals vs Leverage: We use leverage to check for outliers. To understand leverage, recognize that simple linear regression fits a line that will pass through the center of your data. High-leverage points are those observations, if any, made at extreme or outlying values of the independent variables such that the lack of neighboring observations means that the fitted regression model will pass close to that particular observation. There are some points that have a large amount of leverage and probably should be removed.

I tried removing some outliers but this did not improve the results much.

```
library(data.table)
outlierReplace = function(dataframe, cols, rows, newValue = NA) {
    if (any(rows)) {
        set(dataframe, rows, cols, newValue)
    }
}

outlierReplace(dfFavFol, "favourited_count", which(dfFavFol$favourited_count >
    50000), NA)
```

```
outlierReplace(dfFavFol, "favourited_count", which(dfFavFol$favourited_count <
    500), NA)

outlierReplace(dfFavFol, "followers_count", which(dfFavFol$followers_count >
    500000), NA)

outlierReplace(dfFavFol, "followers_count", which(dfFavFol$followers_count <
    500), NA)
```

```
# First I am going to make a data frame with this info
p <-
  qplot(favourited_count,
        followers_count,
        data = dfFavFol,
        position = "jitter")
```

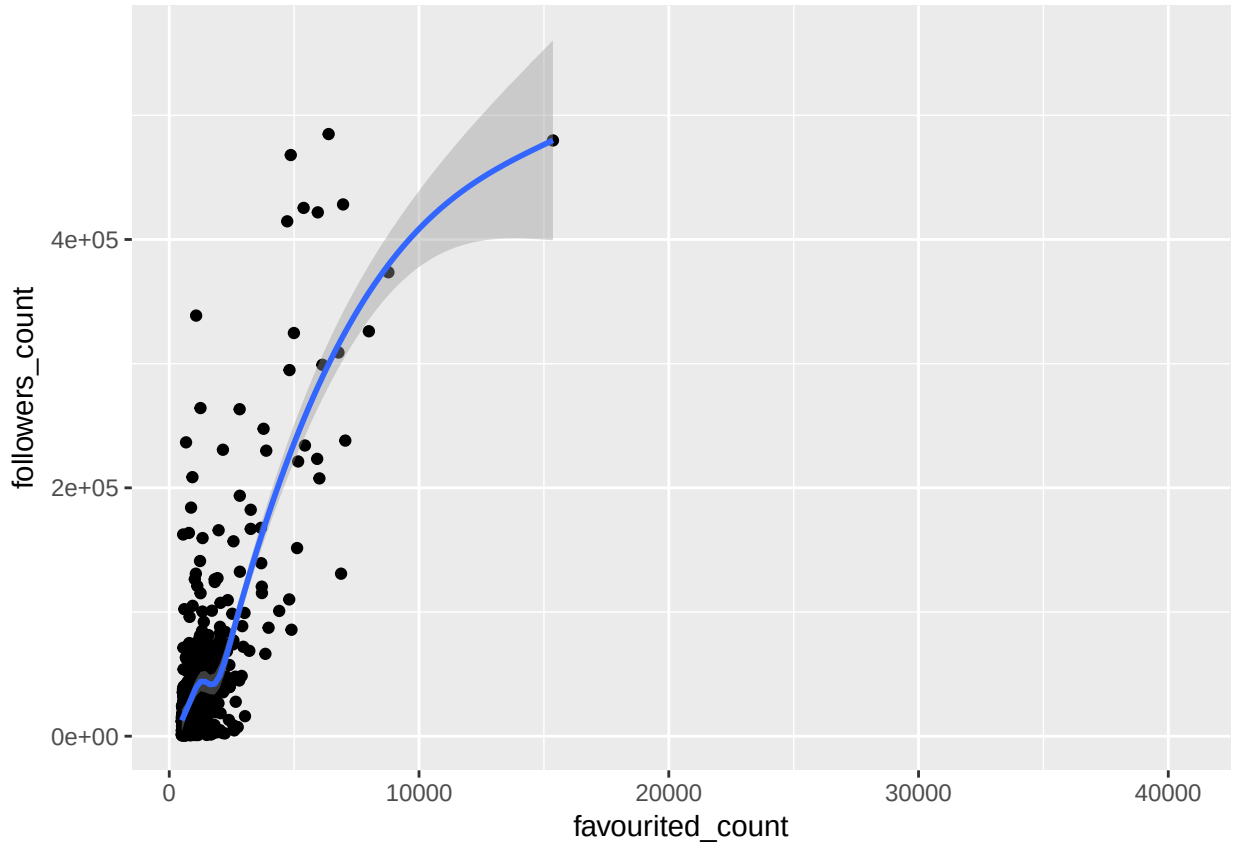
```
## Warning: `position` is deprecated
```

```
p + stat_smooth()
```

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

```
## Warning: Removed 21362 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 21362 rows containing missing values (geom_point).
```




```
m_favourited_followers <-
  lm(favourited_count ~ followers_count, data = dfFavFol)
```

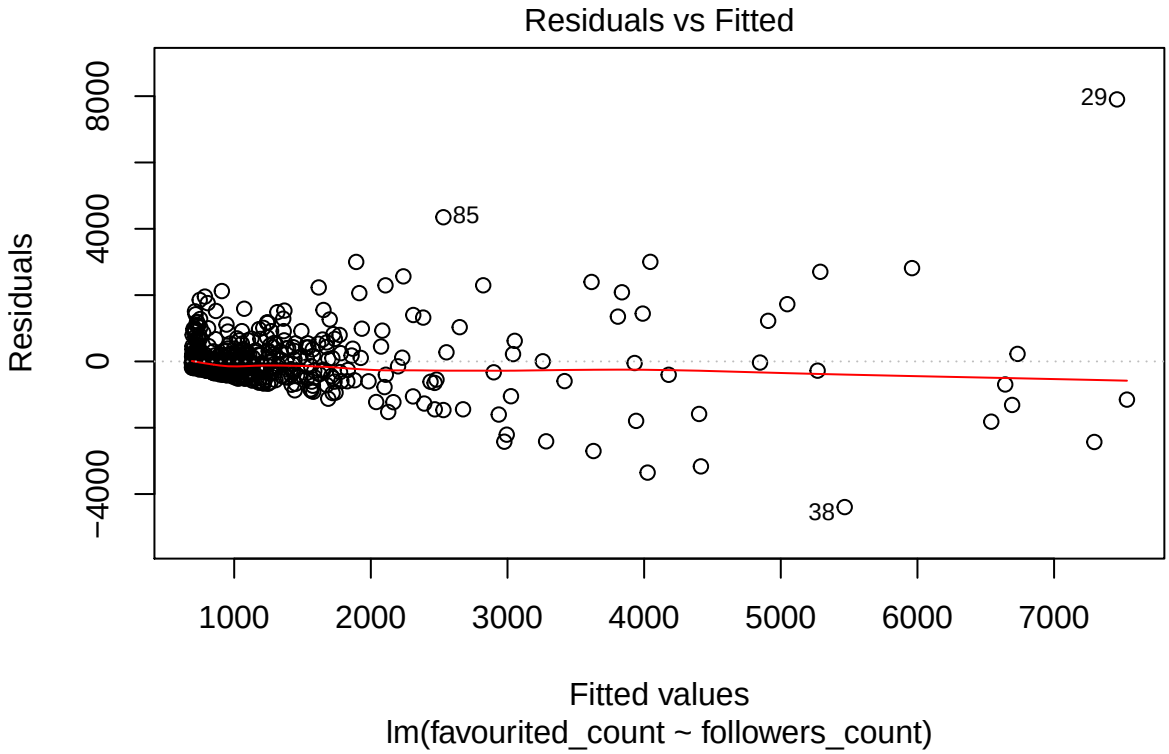
```
m_favourited_followers
```

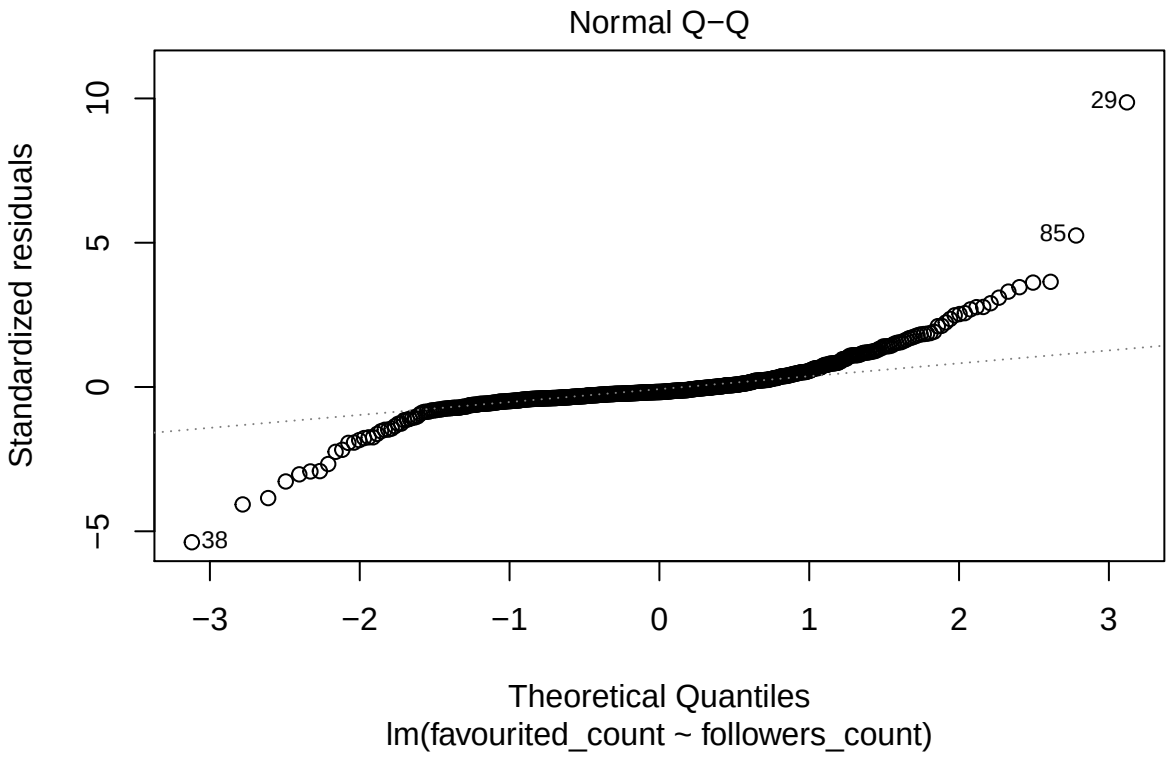
```
##
## Call:
## lm(formula = favourited_count ~ followers_count, data = dfFavFol)
##
## Coefficients:
##     (Intercept)  followers_count
##       681.95197          0.01413
```

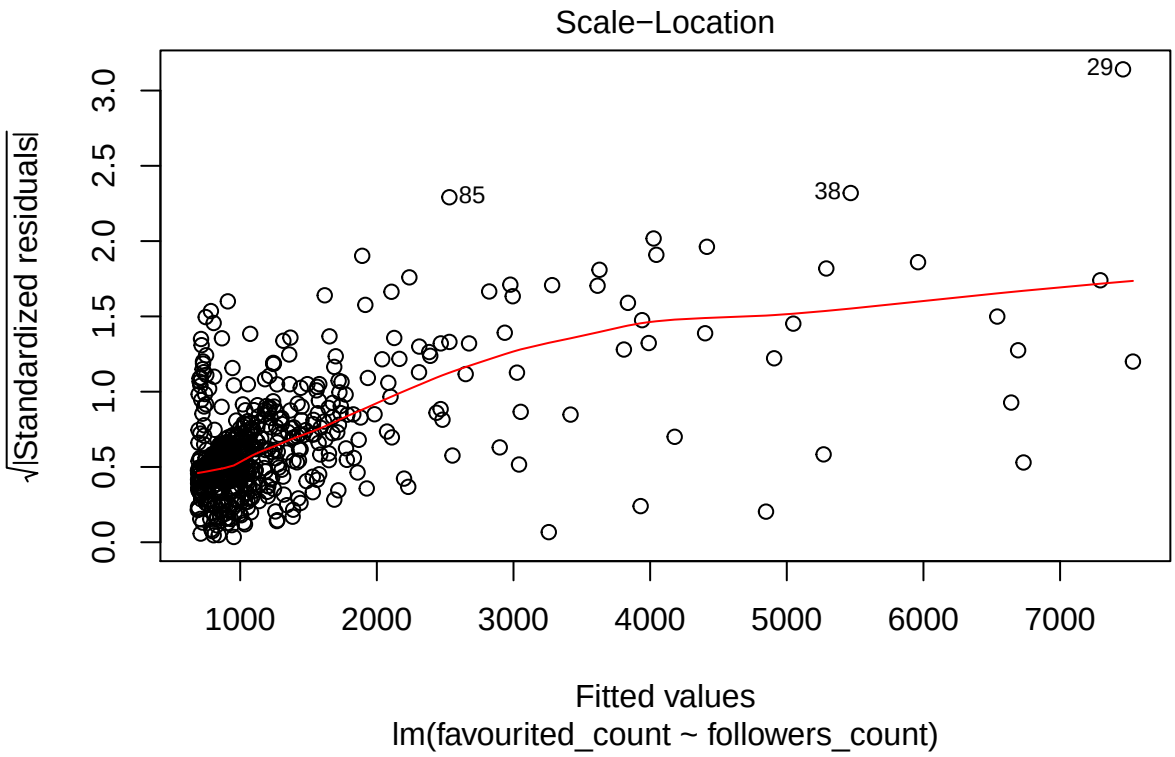
```
anova(m_favourited_followers)
```

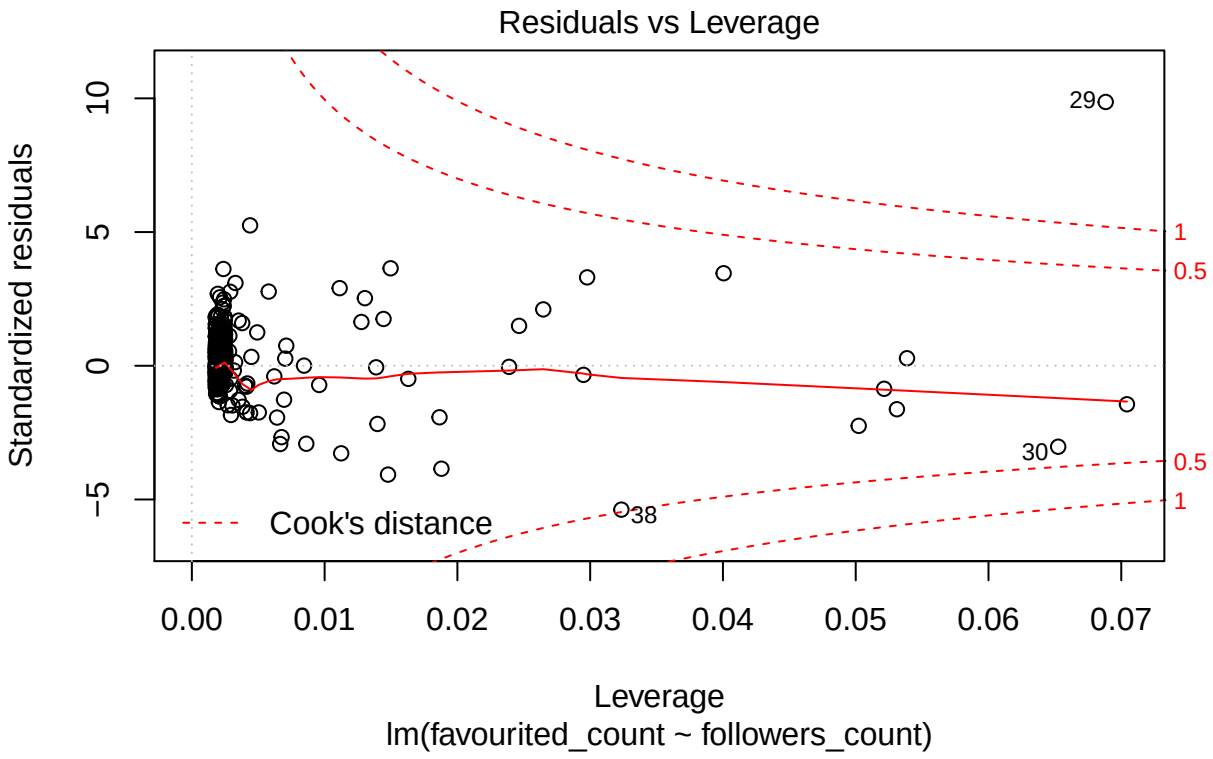
```
## Analysis of Variance Table
##
## Response: favourited_count
##                   Df    Sum Sq   Mean Sq F value    Pr(>F)
## followers_count    1 561265774 561265774  814.59 < 2.2e-16 ***
## Residuals        552 380336623    689016
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(m_favourited_followers)
```




Normal Q−Q
Standardized residuals
10
5
0
−5
29
85
38
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(favourited_count ~ followers_count)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
29
85
38
1000
2000
3000
4000
5000
6000
7000
Fitted values
lm(favourited_count ~ followers_count)



1.2.1.5 Questions:

1. Is the relationship significant?

The t value and F value are significant, there is not much standard error and the small p value (less than 0.05) tells us the probability is very significant. Therefor we can conclude that we can reject the null hypothesis and that followers count matters in the favorited count.

2. Are any model assumptions violated?

Yes, it looks like all of the error term assumptions are violated. The forecasts look to be systematically biased, it looks like there is correlation between consecutive errors in the case of time series data, there is not constant variance, and the error distribution is not normal.

3. Does the model make sense? Why or why not?

No, the t, F and p values look good and should reject the null hypothesis, but when I look at the plots, the data looks like it violates all of the assumptions. I think the data needs to be 'cleaned' and re-run.

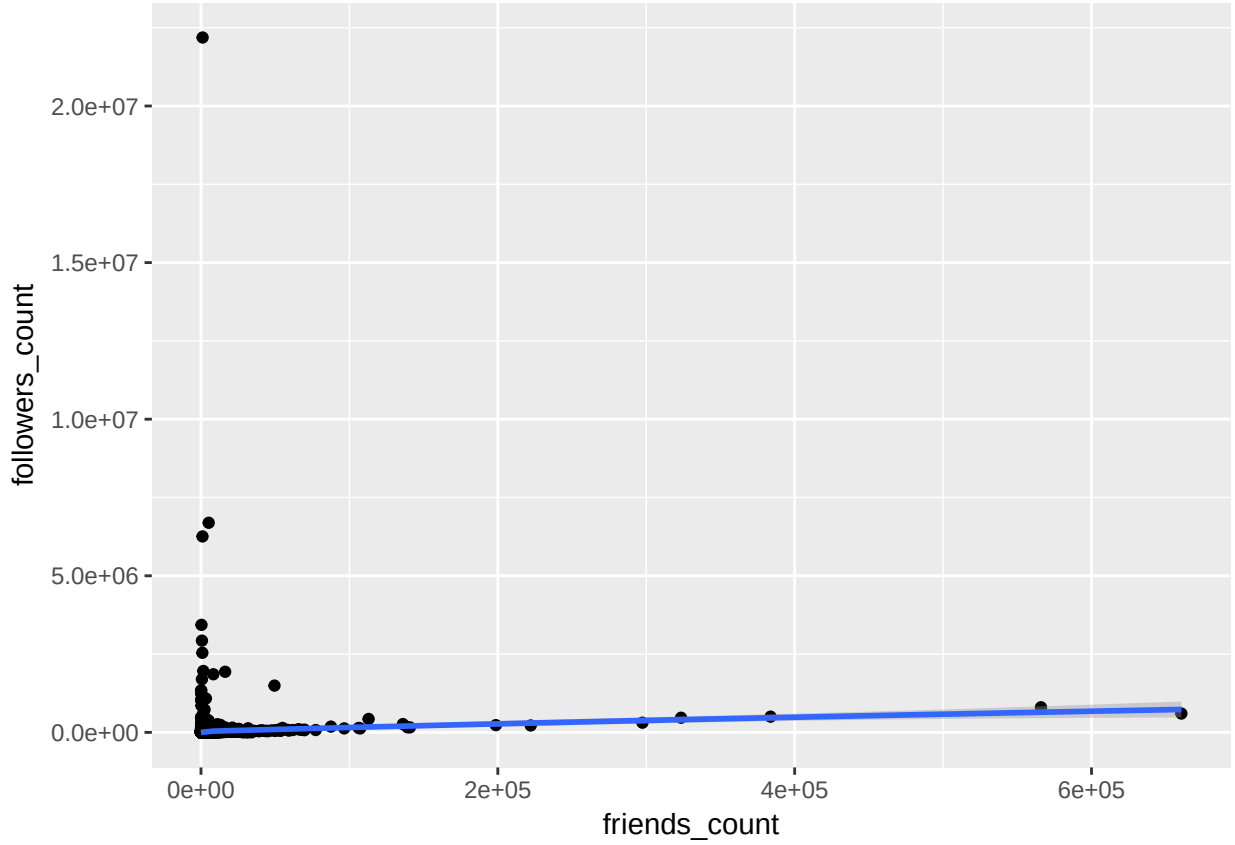
1.2.2 Model 2

Model 2 looks at the relationship between friends count and followers count to see if the followers count matters in the friends count, I plotted the graph below:

```
p2 <-
  qplot(friends_count,
        followers_count,
        data = df1,
```

```
        position = "jitter")
p2 + stat_smooth()
```

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



It looks like there is some correlation between the two so I can fit the linear model.

1.2.2.1 Fit Linear Model

I applied the same approach as I did for model 1:

```
m_friend_followers <- lm(friends_count ~ followers_count, data = df1)
m_friend_followers
```

```
##
## Call:
## lm(formula = friends_count ~ followers_count, data = df1)
##
## Coefficients:
##     (Intercept)  followers_count
##       1.040e+03        3.021e-03
```

Which translates in to the equations below:

$$favouritedcount = 1040 + 0.003x_{followerscount} + \varepsilon$$

1.2.2.2 P-value, t-statistic and standard error:

Below is the analysis of the linear model:

```
summary(m_friend_followers)
```

```
##
## Call:
## lm(formula = friends_count ~ followers_count, data = df1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -66985   -921   -721   -203 657692
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.040e+03  5.481e+01  18.980   <2e-16 ***
## followers_count 3.021e-03  3.209e-04   9.415   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8109 on 21914 degrees of freedom
## Multiple R-squared:  0.004029,   Adjusted R-squared:  0.003984
## F-statistic: 88.65 on 1 and 21914 DF,  p-value: < 2.2e-16
```

The t value is significant, there is not much standard error and the small p value, the probability is very significant. This tells me that followers count matters in the friends count.

Next I move onto an anova.

1.2.2.3 Anova

Analysis of variance (ANOVA) is a collection of statistical models used to analyze the differences among group means and their associated procedures (such as "variation" among and between groups), developed by statistician and evolutionary biologist Ronald Fisher.

An F-test is any statistical test in which the test statistic has an F-distribution under the null hypothesis. Like a t-statistic, or a p-value it provides an estimate of whether one should accept or reject the null hypothesis. The F-test is sensitive to non-normality (as is a t-statistic) but is appropriate under the assumptions of normality and homoscedasticity.

```
anova(m_friend_followers)
```

```
## Analysis of Variance Table
##
## Response: friends_count
##                    Df     Sum Sq    Mean Sq F value    Pr(>F)
## followers_count     1 5.8291e+09 5829102339  88.651 < 2.2e-16 ***
## Residuals       21914 1.4409e+12   65753516
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From this output, the f value is high and the p-value is less than 0.05 (which tells me the F value is significant), therefore I can reject the null hypothesis of no differences between the data (similar to the above results).

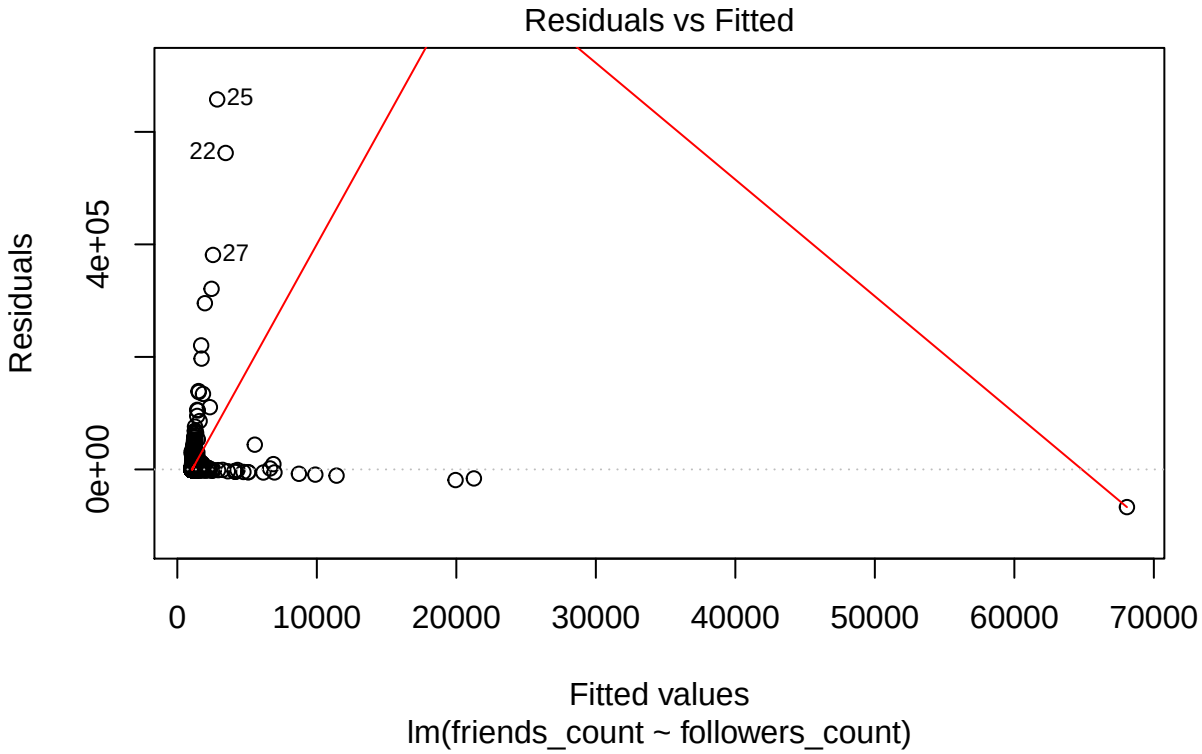
1.2.2.4 Generating the residual plots

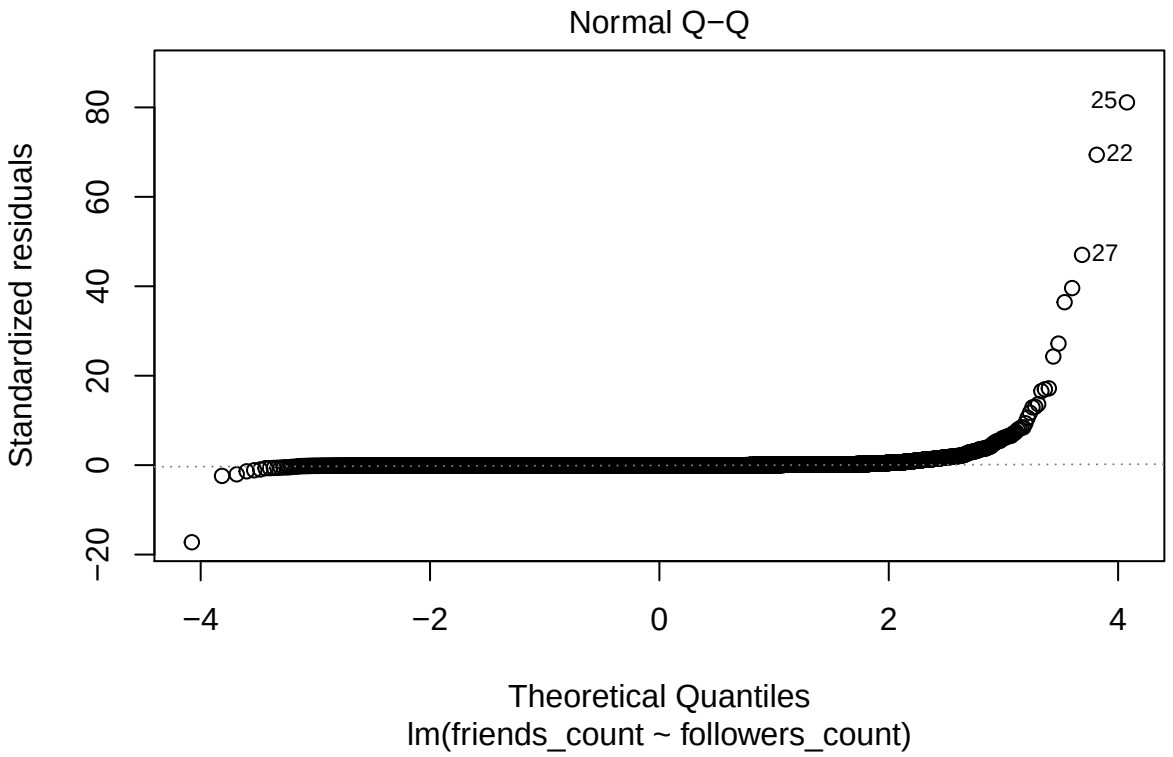
The error term has the following assumptions:

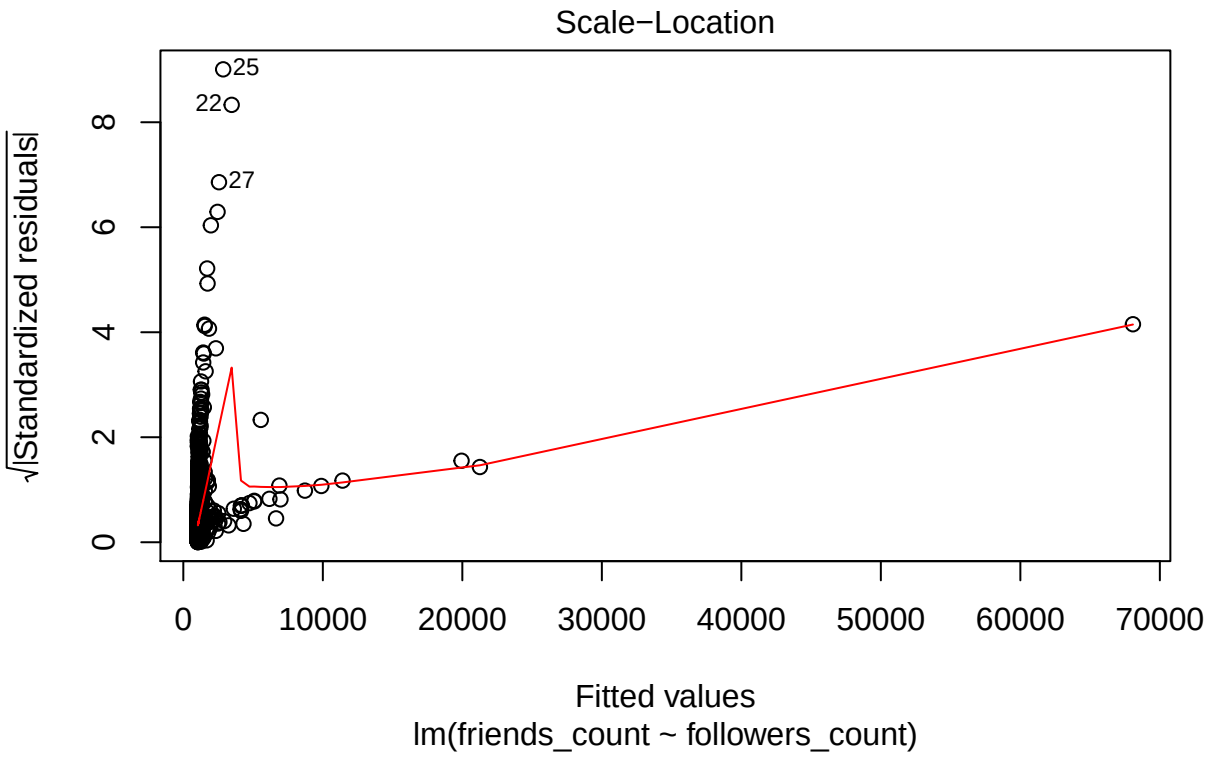
- Have mean zero; otherwise the forecasts will be systematically biased.
- Statistical independence of the errors (in particular, no correlation between consecutive errors in the case of time series data).
- Homoscedasticity (constant variance) of the errors.
- Normality of the error distribution.

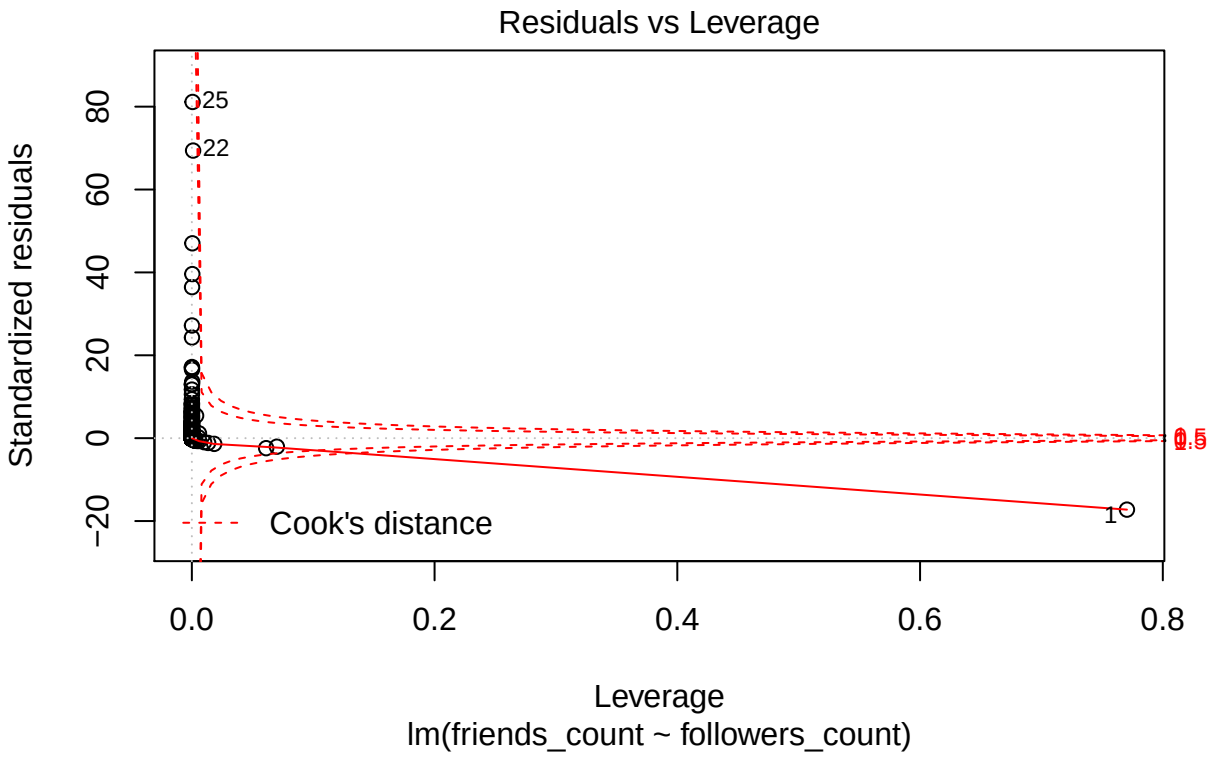
This function will plot the graphs: Residuals vs Fitted, Residuals vs Normal, Standardized Residuals vs Fitted Values, Residuals vs Leverage:

```
plot(m_friend_followers)
```




Normal Q−Q
Standardized residuals
−20
0
20
40
60
80
25
22
27
−4
−2
0
2
4
Theoretical Quantiles
lm(friends_count ~ followers_count)


Scale−Location
25
22
27
√|Standardized residuals|
0
2
4
6
8
0
10000
20000
30000
40000
50000
60000
70000
Fitted values
lm(friends_count ~ followers_count)



Residuals vs Fitted: The residuals and the fitted values should be uncorrelated in a homoscedastic linear model with normally distributed errors. There should not be a dependency between the residuals and the fitted values. It's hard to tell with this plot, but there is some indication that there is some dependency.

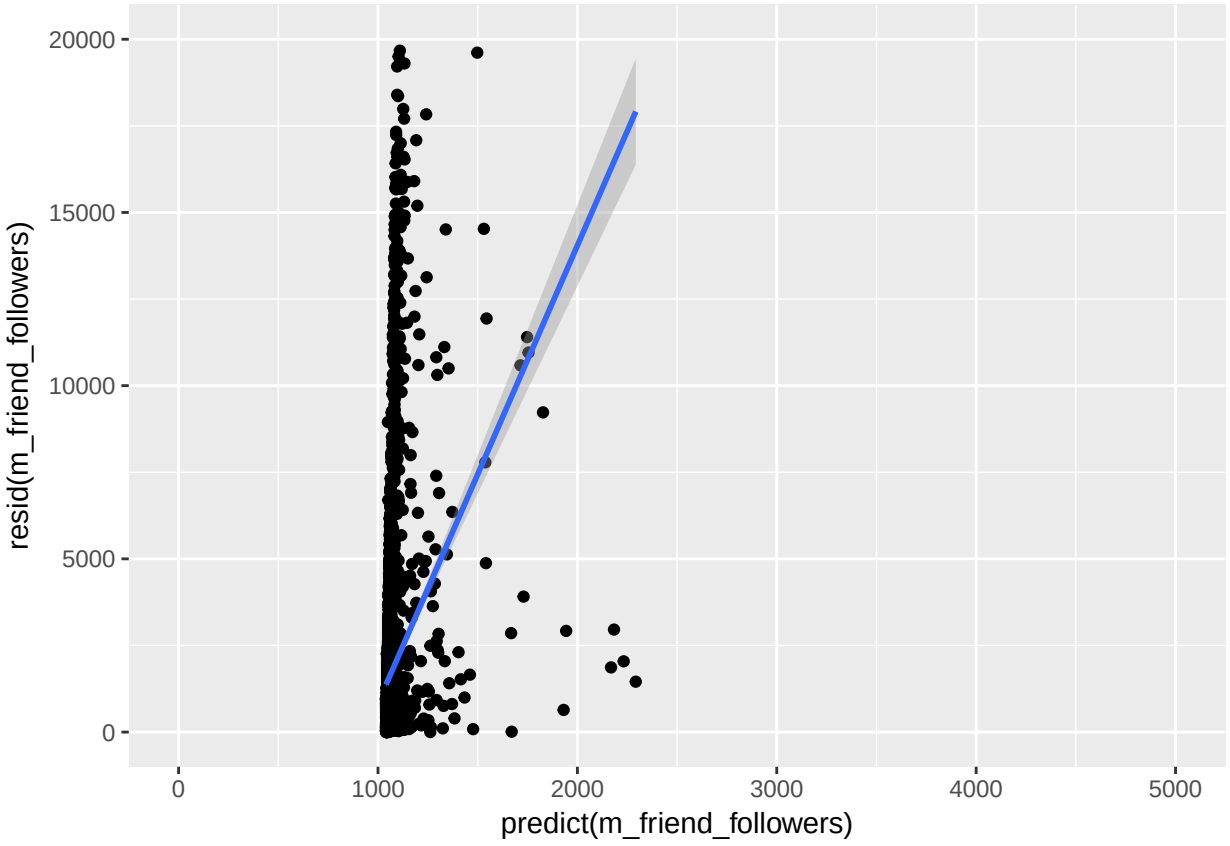
Residuals vs Normal: This is a Q-Q plot to check if the residuals are normal. The ends of the plot do not look so good for this data.

Standardized Residuals vs Fitted Values: standardized residuals means every residual plot you look at with any model is on the same standardized y-axis. This makes it easier to compare many residual plots. This process is also called studentizing (after William Sealey Gosset, who wrote under the pseudonym Student).

Residuals vs Leverage: We use leverage to check for outliers. To understand leverage, recognize that simple linear regression fits a line that will pass through the center of your data. High-leverage points are those observations, if any, made at extreme or outlying values of the independent variables such that the lack of neighboring observations means that the fitted regression model will pass close to that particular observation. There are some points that have a large amount of leverage and probably should be removed.

This is a more detailed version of the Residuals vs Fitted plot. This does not look uncorrelated.

```
qplot(predict(m_friend_followers), resid(m_friend_followers)) +
  geom_smooth(method = lm) + xlim(0, 5000) + ylim(0, 20000)
```



1.2.2.5 Questions:

1. Is the relationship significant? The t value is significant, there is not much standard error and the small p value, the probability is very significant. This tells us that followers count matters in the friends count.

2. Are any model assumptions violated? Yes, it looks like all of the error term assumptions are violated. The forecasts look to be systematically biased, it looks like there is correlation between consecutive errors in the case of time series data, there is not constant variance, and the error distribution is not normal.

3. Does the model make sense? Why or why not? No, the t, F and p values look good and should reject the null hypothesis, but when I look at the plots, the data looks like it violates all of the assumptions. I think the data needs to be 'cleaned' and re-run.

1.3 Multivariate Linear Regression

A multivariate relation between wage & height, race, age, education, and experience:

```
# First I am going to make a data frame with this info
dfWage <-
  subset(df1, select = c(wage, height, race, age, education, experience))

# This is the model
m_wage_height_race_age_education_experience <-
  lm(wage ~ height + race + age + education + experience, data = dfWage)
```

```r
# This is the result
m_wage_height_race_age_education_experience
```

```
##
## Call:
## lm(formula = wage ~ height + race + age + education + experience,
##     data = dfWage)
##
## Coefficients:
##          (Intercept)                height             raceasian
##           -12.003381              0.193717              0.911786
##         racehispanic            raceindian            racelatino
##              1.730412              1.400118              1.057495
##            racemixed   racenative american  racepacific islander
##              0.092136              0.513804              0.392584
##          racepersian             racewhite                   age
##              1.325729              0.865941              0.004442
##            education            experience
##              0.060723             -0.004366
```

```r
# This is the summary
summary(m_wage_height_race_age_education_experience)
```

```
##
## Call:
## lm(formula = wage ~ height + race + age + education + experience,
##     data = dfWage)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.917  -9.269  -2.435   5.367  91.811
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -12.003381   1.994039  -6.020 1.78e-09 ***
## height                  0.193717   0.009193  21.073  < 2e-16 ***
## raceasian               0.911786   1.157688   0.788    0.431
## racehispanic            1.730412   1.308928   1.322    0.186
## raceindian              1.400118   1.552721   0.902    0.367
## racelatino              1.057495   1.144607   0.924    0.356
## racemixed               0.092136   1.473952   0.063    0.950
## racenative american     0.513804   1.392160   0.369    0.712
## racepacific islander    0.392584   1.371930   0.286    0.775
## racepersian             1.325729   1.296183   1.023    0.306
## racewhite               0.865941   1.064457   0.814    0.416
## age                     0.004442   0.007656   0.580    0.562
## education               0.060723   0.037019   1.640    0.101
## experience             -0.004366   0.007638  -0.572    0.568
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.46 on 21902 degrees of freedom
## Multiple R-squared:  0.02041,    Adjusted R-squared:  0.01983
## F-statistic:  35.1 on 13 and 21902 DF,  p-value: < 2.2e-16
```

```
# This is the anova test
anova(m_wage_height_race_age_education_experience)
```

```
## Analysis of Variance Table
##
## Response: wage
##               Df  Sum Sq Mean Sq  F value Pr(>F)
## height         1   93912   93912 449.1906 <2e-16 ***
## race           9     777      86   0.4127 0.9292
## age            1      69      69   0.3317 0.5647
## education      1     565     565   2.7033 0.1002
## experience     1      68      68   0.3267 0.5676
## Residuals  21902 4579052     209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From this output, the p-value is greater than 0.05 (with an exception to height), not very small, this indicates weak evidence against the null hypothesis, so I fail to reject the null hypothesis and that there is no significant difference between specified populations.

1.3.1 Multi-colinearity

Multicollinearity (also collinearity) is a phenomenon in which two or more predictor variables in a multiple regression model are highly correlated, meaning that one can be linearly predicted from the others with a non-trivial degree of accuracy.

To understand multicollinearity, I can run a correlation between the variables:

```
# multicollinearity for wage and height
cor(dfWage$wage, dfWage$height)
```

```
## [1] 0.1417412
```

```
# multicollinearity for wage and race
cor(dfWage$wage, as.numeric(dfWage$race))
```

```
## Warning in is.data.frame(y): NAs introduced by coercion
```

```
## [1] NA
```

```
# multicollinearity for wage and age
cor(dfWage$wage, dfWage$age)
```

```
## [1] 0.0007564446
```

```
# multicollinearity for wage and education
cor(dfWage$wage, dfWage$education)
```

```
## [1] 0.01057567
```

```
# #multicollinearity for wage and experience
cor(dfWage$wage, dfWage$experience)
```

```
## [1] -0.00863662
```

This shows that there really is not much correlation between the variables, even wage and height appears small.

1.3.2 Stepwise regression

Stepwise regression means to iteratively select the best predictor (that improves the model the most), then the next best until we have no predictors that improves the model or use all of the predictors. This is also called forward stepwise selection.

```
beg<-lm(wage ~ height, data = dfWage)
end<-lm(wage ~ ., data = dfWage)
empty<-lm(wage ~ 1, data = dfWage)
bounds<-list(upper=end,lower=empty)
stepwise_reg<-step(beg,bounds, direction = "forward")
```

```
## Start:  AIC=117087
## wage ~ height
##
##              Df Sum of Sq     RSS    AIC
## + education   1    554.33 4579977 117086
## <none>                    4580532 117087
## + experience  1     77.56 4580454 117089
## + age         1     68.23 4580463 117089
## + race        9    776.63 4579755 117101
##
## Step:  AIC=117086.4
## wage ~ height + education
##
##              Df Sum of Sq     RSS    AIC
## <none>                    4579977 117086
## + experience  1     74.79 4579903 117088
## + age         1     67.17 4579910 117088
## + race        9    788.64 4579189 117101
```

```
stepwise_reg
```

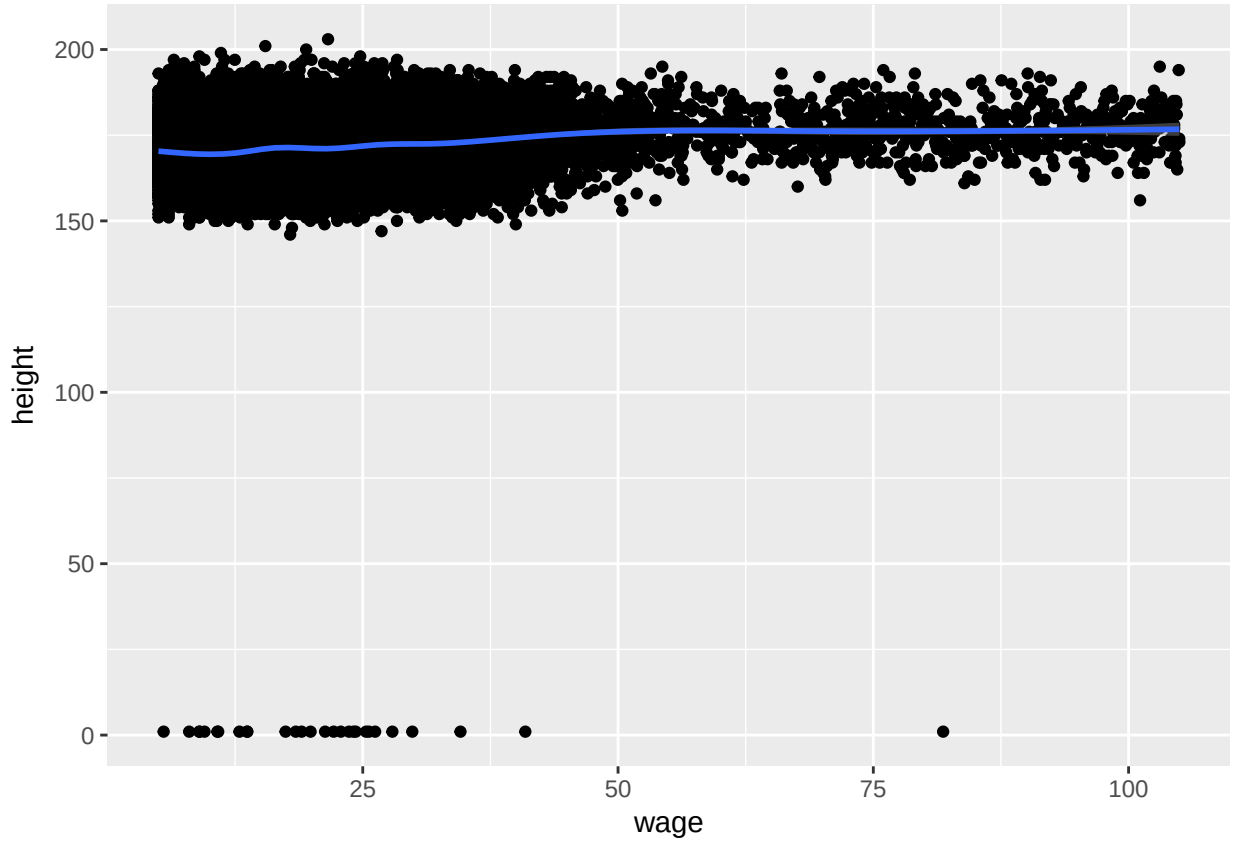
```
##
## Call:
## lm(formula = wage ~ height + education, data = dfWage)
##
## Coefficients:
## (Intercept)       height    education
##   -11.10646      0.19429      0.06027
```

The AIC is too high for this information to be significant. This is what I expected from the previous results.

Since the only significant data is wage and height, I am making a model of that:

```
p3 <-
  qplot(wage,
        height,
        data = dfWage,
        position = "jitter")
p3 + stat_smooth()
```

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



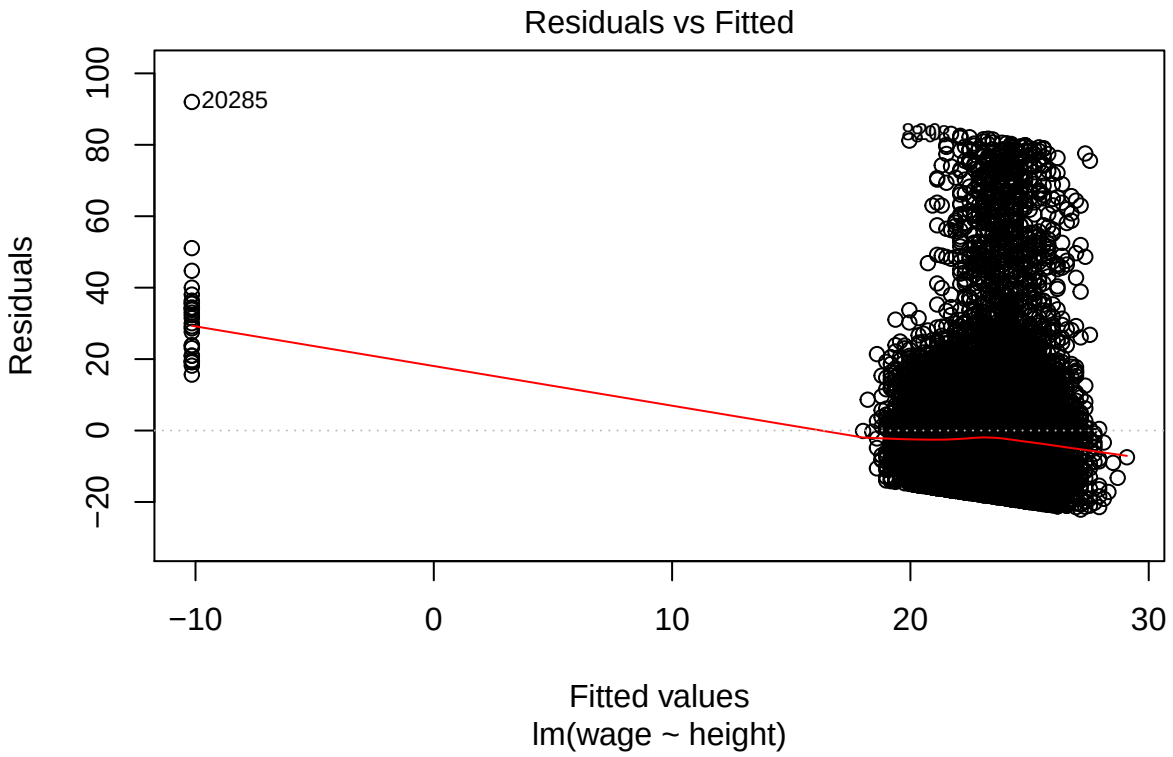
It looks like there is some correlation between the two so I can fit the linear model.

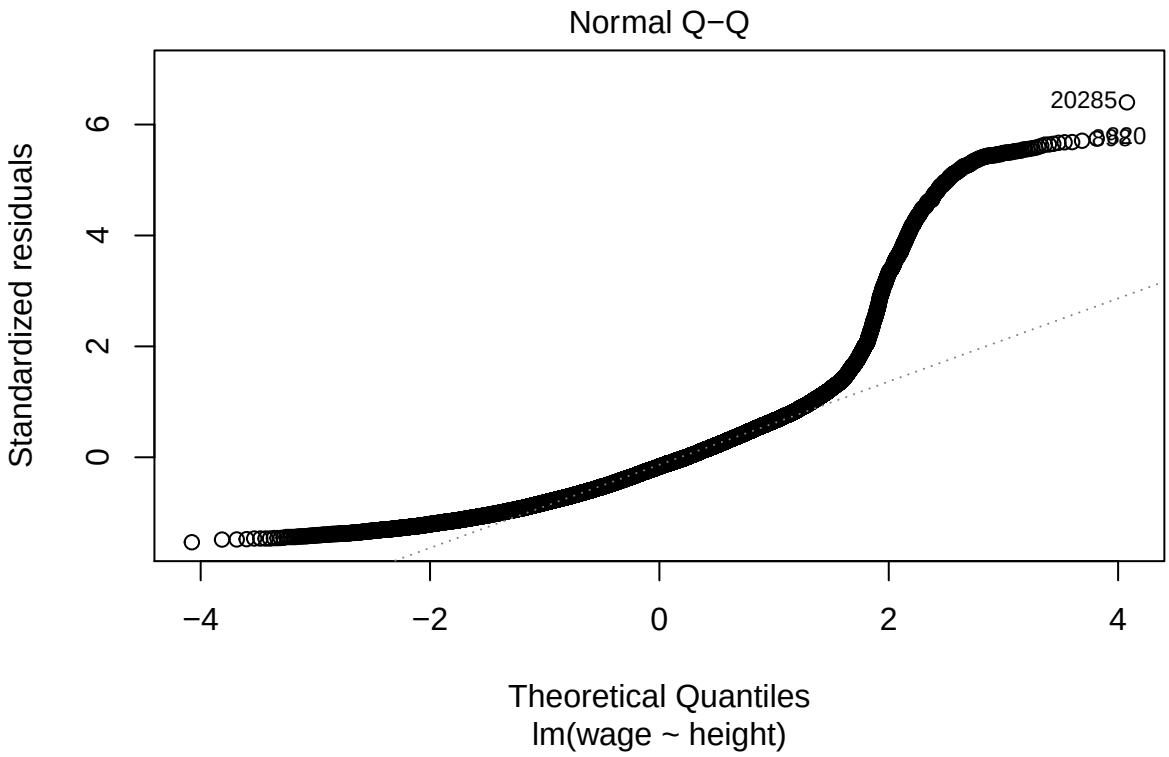
I applied the same approach as I did for model 1 and 2:

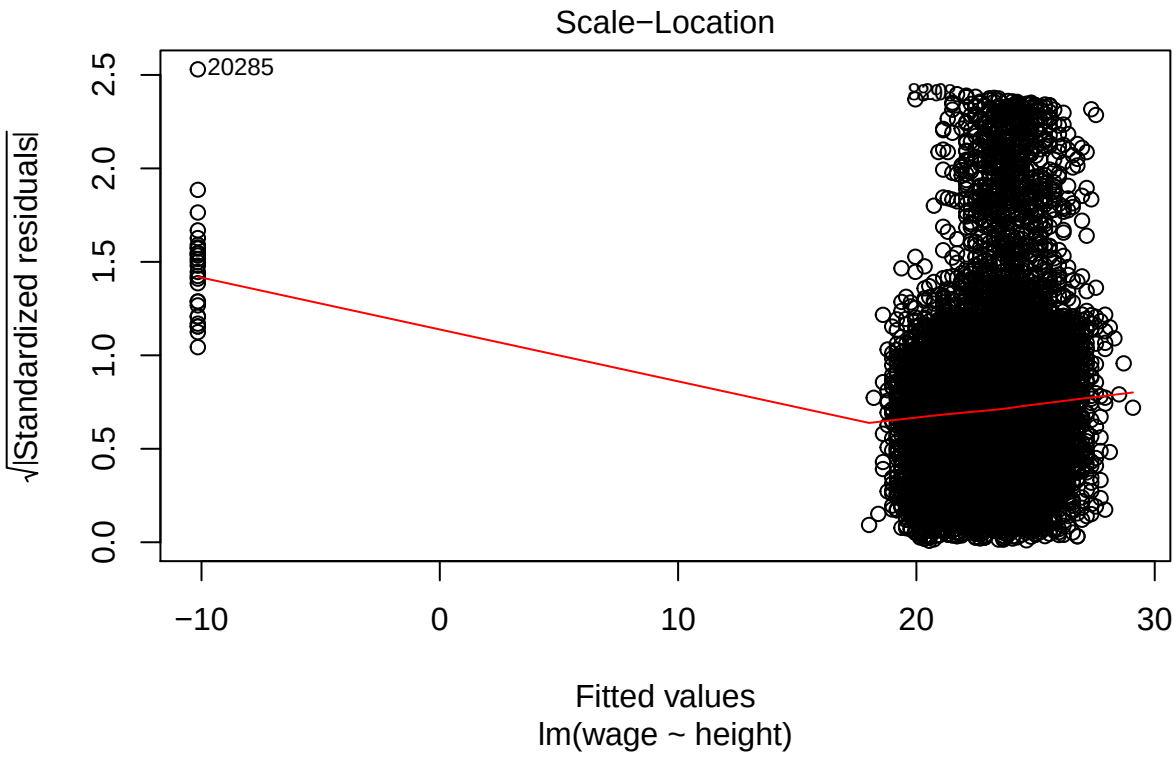
```
m_wage_height <- lm(wage ~ height, data = dfWage)
m_wage_height
```

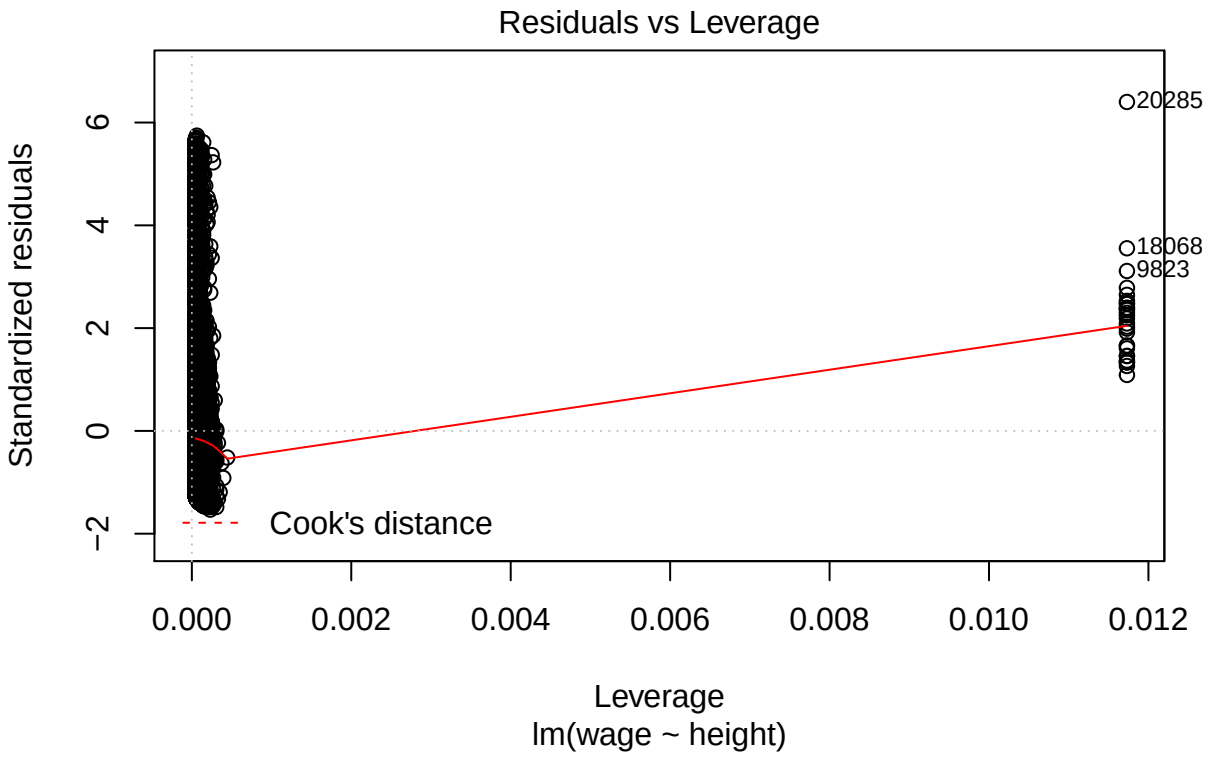
```
##
## Call:
## lm(formula = wage ~ height, data = dfWage)
##
## Coefficients:
## (Intercept)       height
##    -10.3476       0.1943
```

```
plot(m_wage_height)
```


Residuals vs Fitted
20285
Residuals
100
80
60
40
20
0
−20
−10
0
10
20
30
Fitted values
lm(wage ~ height)


Normal Q−Q
Standardized residuals
0
2
4
6
20285
−4
−2
0
2
4
Theoretical Quantiles
lm(wage ~ height)


Scale−Location
20285
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
−10
0
10
20
30
Fitted values
lm(wage ~ height)



The plots do not look that great and the data should be cleaned a little. This data violates all of the assumptions for the residual plots.

1.3.3 Questions:

1. Is the relationship significant? Height and wage is the only significant relationship, the t value is significant, there is not much standard error and the small p value, the probability is very significant. This tells us that height matters with regards to wage. Everything else is insignificant.

2. Is there any multi-colinearity in multivariate models? Not really, there was no significant correlation between the variables, height was the most significant and that was only 0.14.

3. In in multivariate models are predictor variables independent of all the other predictor variables? It looks like they are all independent of each other, with an exception to height.

4. In in multivariate models rank the most significant predictor variables and exclude insignificant one from the model. Height is the only significant predictor.

5. Does the model make sense? Why or why not? Not really, the data shows that wage and height are significant, common sense would tell us that this isn't true. In the next section I show that it is based on gender not height.

1.4 A significant logistic linear model

Logistic regression, or logit regression, or is a regression model where the outcome variable is categorical. Often this is used when the variable is binary (e.g. yes/no, survived/dead, pass/fail, etc.)

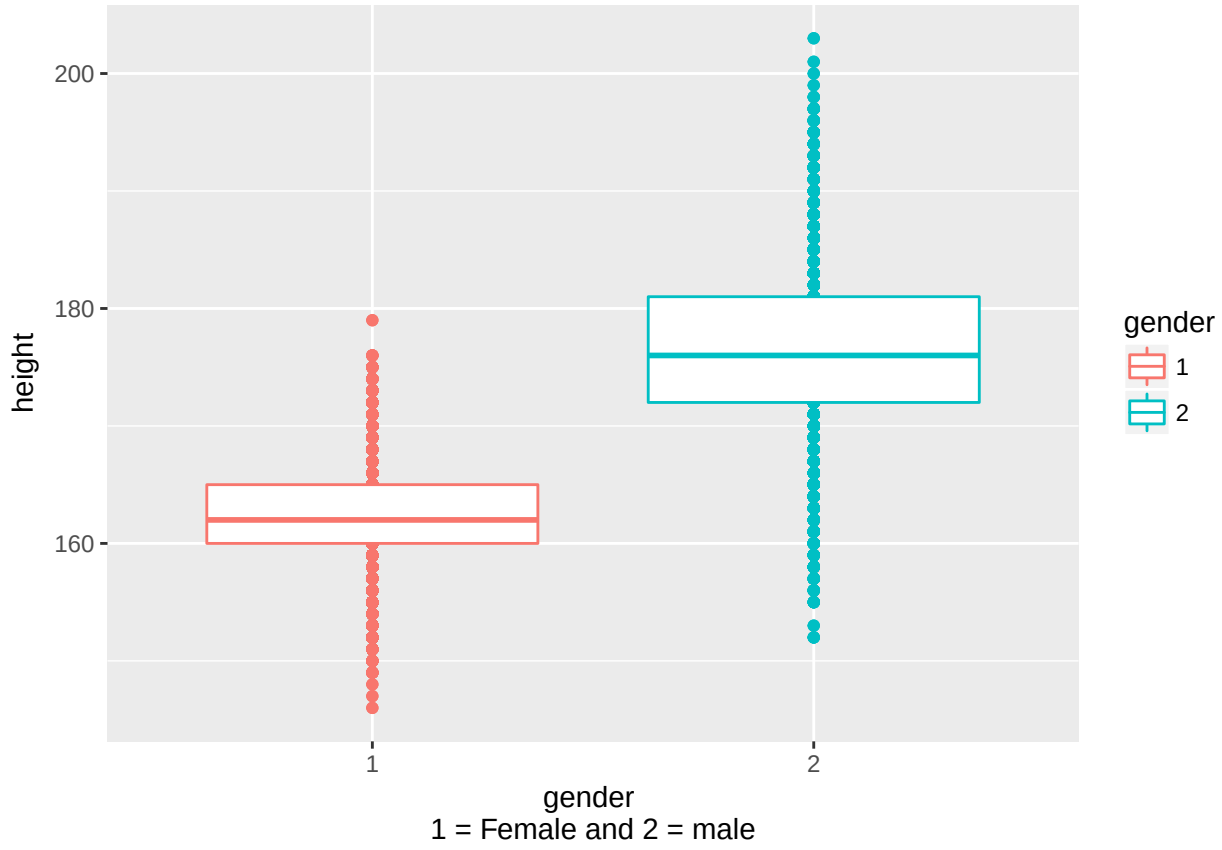
I am going to compare height and gender to see if I can use height to predict gender:

```
# Put data into data frame
dfGenderH <- subset(df1, select = c(gender, height))

# Remove na
dfGenderH <- na.omit(dfGenderH)

# Change male and female to 1 and 2 to evaluate.
dfGenderH$gender <-
  factor(dfGenderH$gender,
         levels = c("female", "male"),
         labels = c(1, 2))

# Plot the info
qplot(gender, height, data = dfGenderH, color = gender) + geom_boxplot() +
  xlab("gender \n1 = Female and 2 = male")
```



This shows that there is a significant difference between men and women and that I can model a significant difference between the two.

I begin by setting up a model with the gender info and the height:

```
m_gender_logistic <- lm(as.numeric(gender) ~ height, data = dfGenderH)
m_gender_logistic
```

```
##
## Call:
## lm(formula = as.numeric(gender) ~ height, data = dfGenderH)
```

```
##
## Coefficients:
## (Intercept)       height
##    -5.33007      0.04073
```

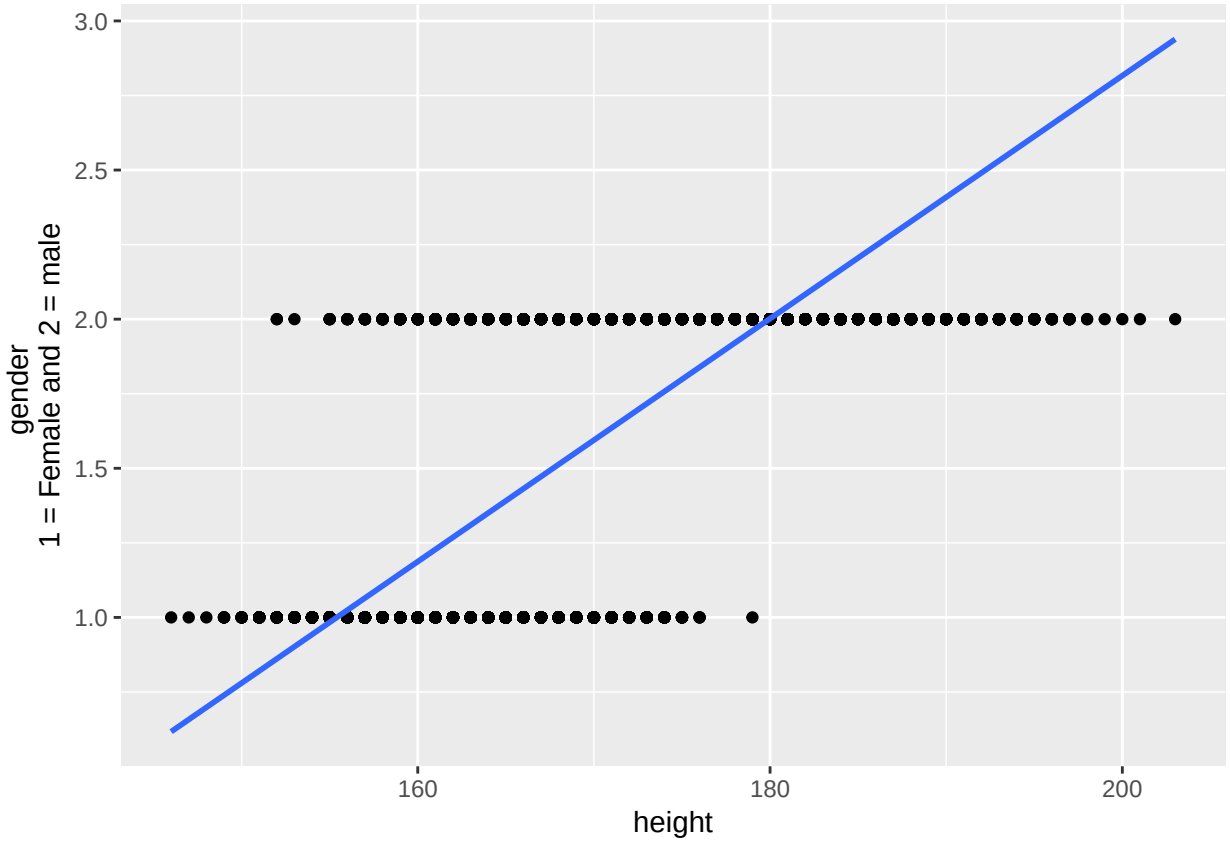
```
summary(m_gender_logistic)
```

```
##
## Call:
## lm(formula = as.numeric(gender) ~ height, data = dfGenderH)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.96104 -0.22785 -0.00177  0.24263  1.13874
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.3300696  0.0411321  -129.6   <2e-16 ***
## height       0.0407324  0.0002392   170.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3094 on 21886 degrees of freedom
## Multiple R-squared:  0.5699, Adjusted R-squared:  0.5699
## F-statistic: 2.9e+04 on 1 and 21886 DF,  p-value: < 2.2e-16
```

```
anova(m_gender_logistic)
```

```
## Analysis of Variance Table
##
## Response: as.numeric(gender)
##              Df Sum Sq Mean Sq F value    Pr(>F)
## height        1 2776.4  2776.4   29002 < 2.2e-16 ***
## Residuals 21886 2095.2     0.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
qplot(height, as.numeric(gender), data = dfGenderH) +
  stat_smooth(method = "lm",
              formula = y ~ x,
              se = FALSE) + ylab("gender \n1 = Female and 2 = male")
```



The results are as expected that the data is different and we can use height to predict gender. The data shows a high t value and a high F value with a very low p value, this means that I can reject the null hypothesis and that there is no significant difference between specified populations.

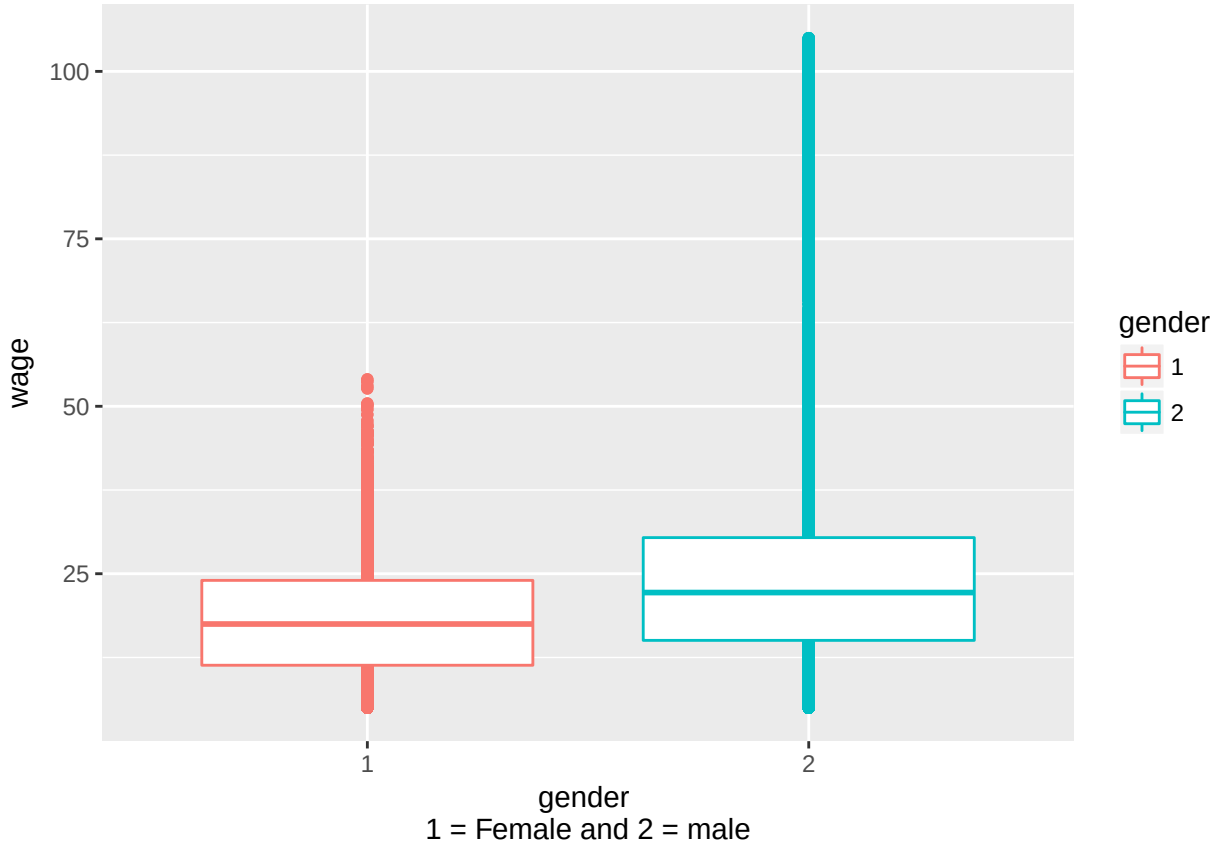
I think it is strange that there is a correlation between height and wage from the previous section and I think it is actually a correlation between gender and wage. The analysis above showed that men are typically taller than women and the analysis below shows the pay gap based on gender. I think this is the reason there is a correlation between height and wage.

```
# Put data into data frame
dfWageGender <- subset(df1, select = c(gender, wage))

# Remove na
dfWageGender <- na.omit(dfWageGender)

# Change male and female to 1 and 2 to evaluate.
dfWageGender$gender <-
  factor(dfWageGender$gender,
         levels = c("female", "male"),
         labels = c(1, 2))

# Plot the info
qplot(gender, wage, data = dfWageGender, color = gender) + geom_boxplot() +
  xlab("gender \n1 = Female and 2 = male")
```




```
m_gender_wage <- lm(as.numeric(gender) ~ wage, data = dfWageGender)
m_gender_wage
```

```
##
## Call:
## lm(formula = as.numeric(gender) ~ wage, data = dfWageGender)
##
## Coefficients:
## (Intercept)         wage
##    1.500141     0.007203
```

```
summary(m_gender_wage)
```

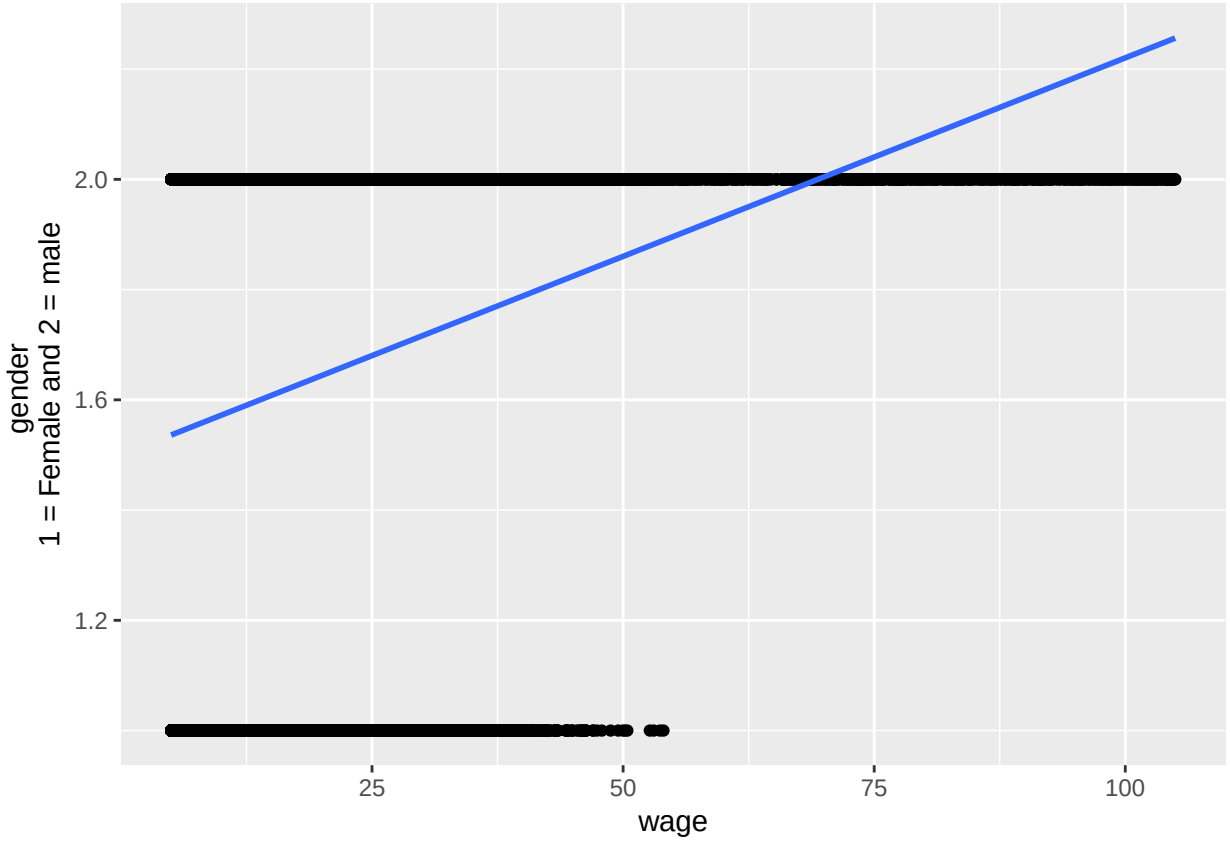
```
##
## Call:
## lm(formula = as.numeric(gender) ~ wage, data = dfWageGender)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8892 -0.5820  0.2806  0.3647  0.4638
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.5001412  0.0057947  258.88   <2e-16 ***
## wage        0.0072025  0.0002129   33.84   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.4599 on 21886 degrees of freedom
## Multiple R-squared:  0.04971,    Adjusted R-squared:  0.04967
## F-statistic:  1145 on 1 and 21886 DF,  p-value: < 2.2e-16
```

```
anova(m_gender_wage)
```

```
## Analysis of Variance Table
##
## Response: as.numeric(gender)
##              Df Sum Sq Mean Sq F value    Pr(>F)
## wage          1  242.2 242.193    1145 < 2.2e-16 ***
## Residuals 21886 4629.4   0.212
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
qplot(wage, as.numeric(gender), data = dfWageGender) +
  stat_smooth(method = "lm",
              formula = y ~ x,
              se = FALSE) + ylab("gender \n1 = Female and 2 = male")
```



1.4.1 Questions:

1. Is the relationship significant?

The t value is significant, there is not much standard error and the small p value, the probability is very significant. This tells us that there is correlation between gender and height.

2. Does the model make sense? Why or why not?

Yes, this data makes sense. The data supports that we can reject the null hypothesis and it supports the notion that we can predict gender from height.