Final project STA 141A 2021 Spring

Group members: Aria Hamidi, Elizabeth Steinbach, Jonathan Jiang, Jinghong Chen, Daniel Momeni

Email: esteinbach@ucdavis.edu, jsjjiang@ucdavis.edu, jghchen@ucdavis.edu, dmomeni@ucdavis.edu, hamidi@ucdavis.edu

Contribution:

Elizabeth: Planning project proposal data sets, questions, and methodology. Linear regression analysis. Written analysis. Jonathan: Planning project proposal data sets, questions, and methodology. ANOVA coding on the final project. Linear regression analysis. Daniel: Planning project proposal data sets, questions, and methodology ANOVA coding on the final project. Written analysis Jinghong: Planning project proposal data sets, questions, and methodology. ANOVA coding on the final project. Linear regression analysis. Aria: ANOVA coding on the final project. Linear regression analysis. Written analysis.

source:

r package tidytuesday

https://steamdb.info/sales/

Introduction and Data Set

This project analyzes video game data from the games.csv file built into tidyverse. We are fundementally interested in how video game playership changes over time.

The variables in this package include the name of the videogame ("gamename"), the year the measure was taken ("year"), the month the measure was taken ("month"), the average number of players for a given game ("avg"). The gain or loss/negative gain in average between each monthly number of players for a given game ("Gain"). These are the variables we will primarily focus on.

The highest number of players at the same time for a given game ("peak"), and the share of the average in the maximum value as a percentage for a given game ("Avg_peak_perc").

Based on the given values in the dataset, we plan to predict the number of gamers in the future and also compare any changes in the number of gamers during any specified time frame.

The games.csv file does give good quality macrodata, but we are also interested in more specific data that is not included in this dataset. Therefore, we decided to include additional data on "Total War: WARHAMMER II" in order to run tests that will provide more targeted insights.

Questions of interest and Methodology

1. Do the total number of gamers change seasonally? We believe there could be changes due to weather patterns and school breaks. We will be using ANOVA to explore the data and run a hypothesis test to determine if there is a significant difference in averages between the seasons.

2. Have the total number of gamers changed between years? We believe there could be changes because of the increasing popularity and quality of video games. We will be using ANOVA to explore the data and run a hypothesis test to determine if there is a significant difference in averages between the years.

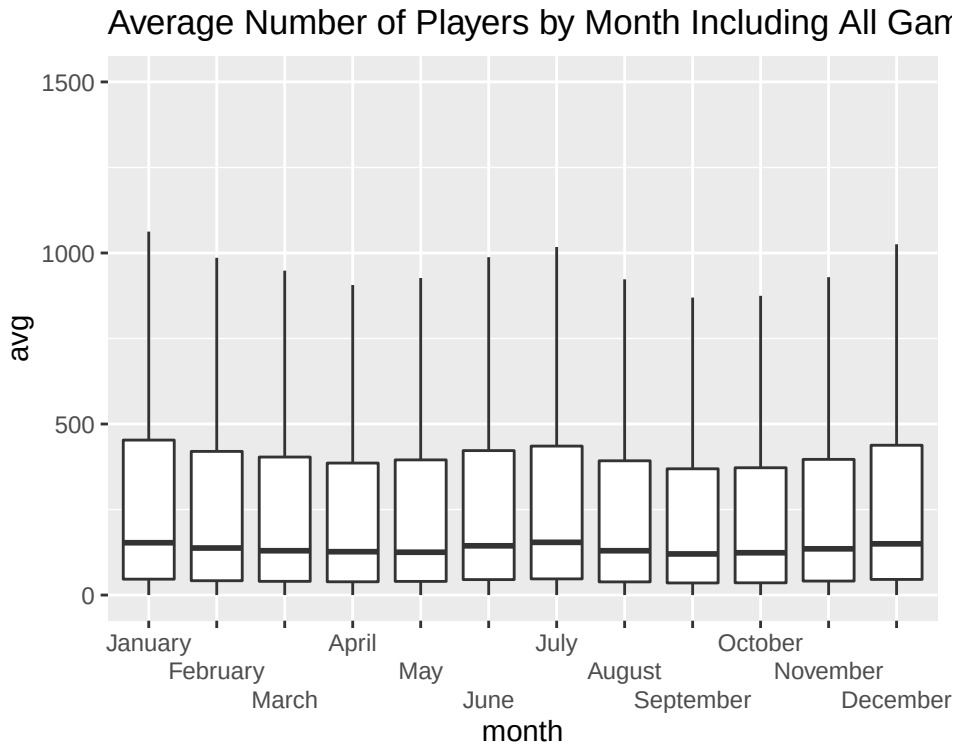
3. Did the total number of gamers meaningfully change before and after the pandemic? We believe that there will be a meaningful change, since quarantine lock-downs limited people's abilities to exercise outdoors or socialize in person, and increased unemployment meant people had more free time.
We will be using ANOVA to explore the data and run a hypothesis test to determine if there is a significant difference in averages before and after pandemic lockdowns, controlling for seasonal differences.

4. How many people will be playing the video game "Total War: WARHAMMER II" in April? We will using a linear regression model to make a prediction.

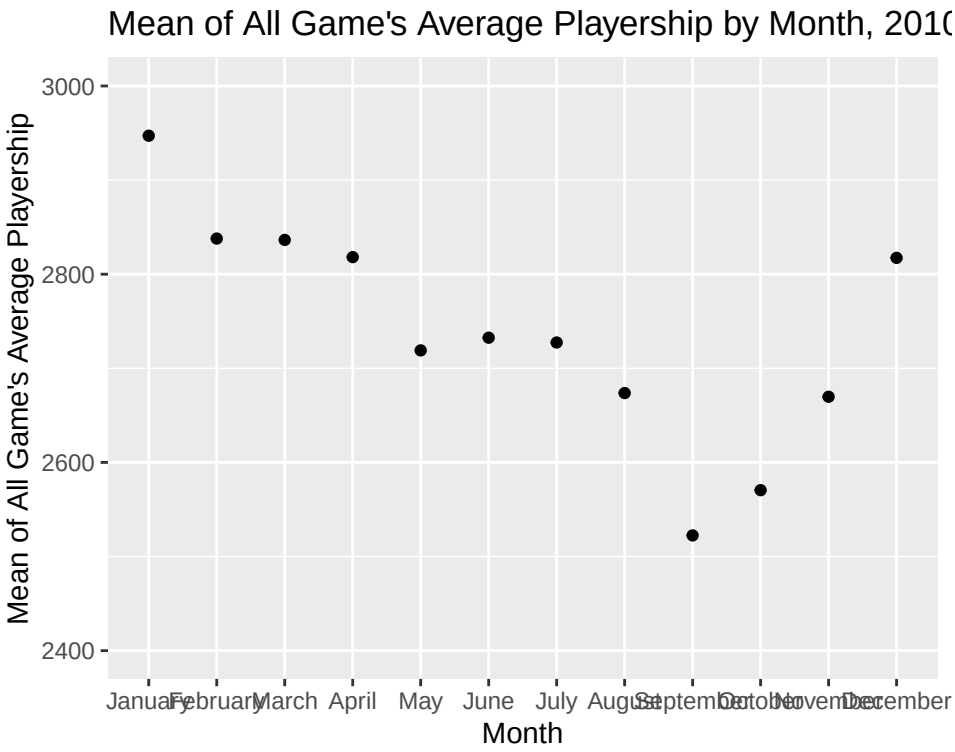
Visualization

In our basic initial visualizations of the fully unmanipulated data, there does seem to be a pattern between months and across years. There also appears to be a difference in playership before and after the start of the Covid-19 pandemic lockdowns in March of 2020.

MONTHLY AVERAGES:



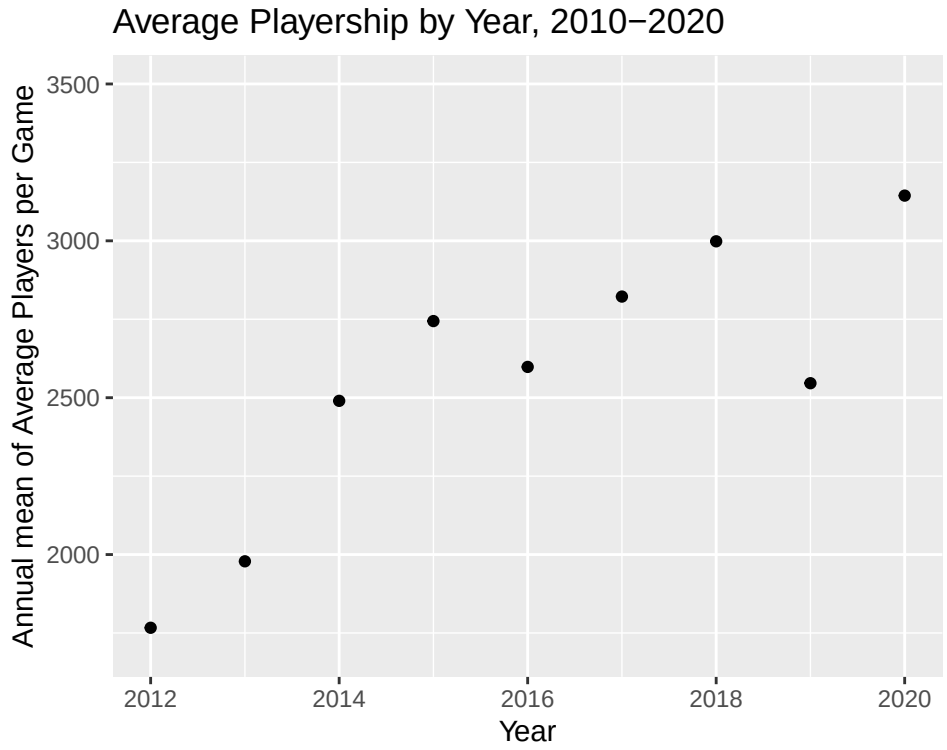
In the plot above, we can suggest the average number of players by different months including all games from 2010 to 2021. In order to have a better observation here, we removed the outliers in each month by using boxplots. If we want to have further observations, there could be seen some slight differences between the quantiles of each month's boxplots.



MONTHLY AVERAGES: In our visualization called "Average Playership by Month, 2010-2020," the Y axis represents the mean number of players in each game in a given month, averaged across all games. The X axis represents each month on record. We chose not to include the year 2021 because this could mean a change in the mean number of players in January through April that cannot be applied for May through November.

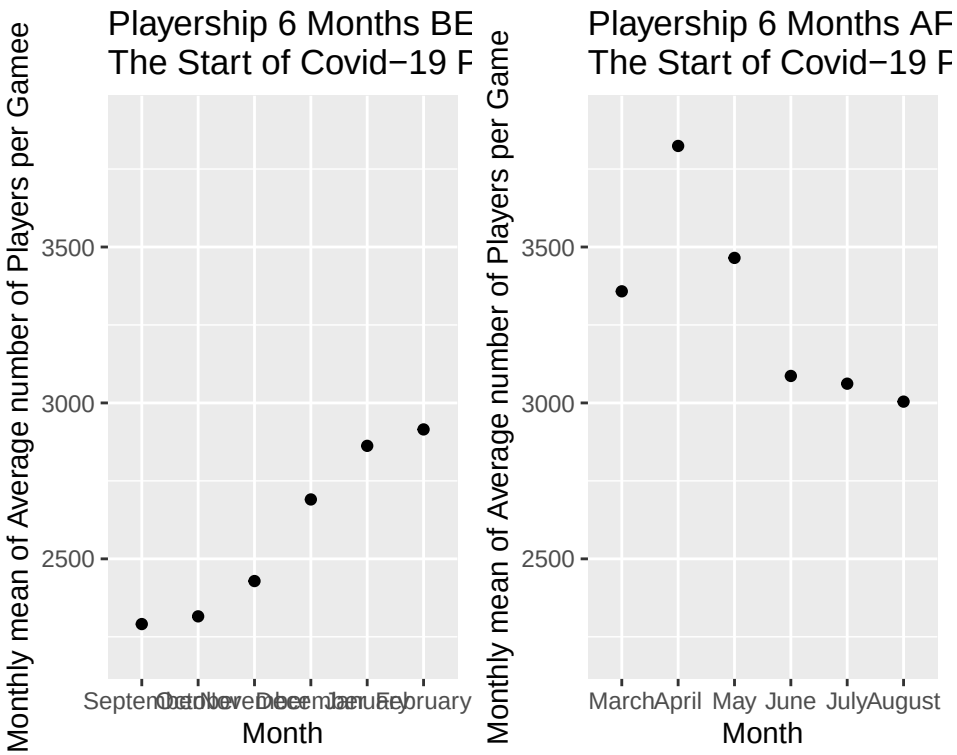
On average, the number of players decreases from January through September. This is followed by an increase between October and January. January has the highest number of players and September the lowest.

YEARLY AVERAGES:



YEARLY AVERAGES: In our visualization called "Average Playership by Year, 2010-2020," the Y axis represents the mean number of players in each game in a given year, averaged across all games. The X axis represents the year. Here we chose not to include 2021 data because the data for this year is less than half complete since it is only June. On average, the number of players increased in 8 out of the 10 years sampled. 2012 is the lowest average on record, and 2020 is the highest. Between these two years, playership nearly doubled. So far this is supportive of our second alternative hypothesis.

PRE/POST PANDEMIC:



PRE/POST PANDEMIC: There is a notable difference in number of players in the 6 months before the covid-19 pandemic and the 6 months after. Every value before the pandemic is lower than every value recorded after the start of the pandemic. The fact that January 2020 had a lower number of players than September 2020 is particularly notable, since in the previous 10 years, January had some of the highest number of players and September the lowest.

Anova

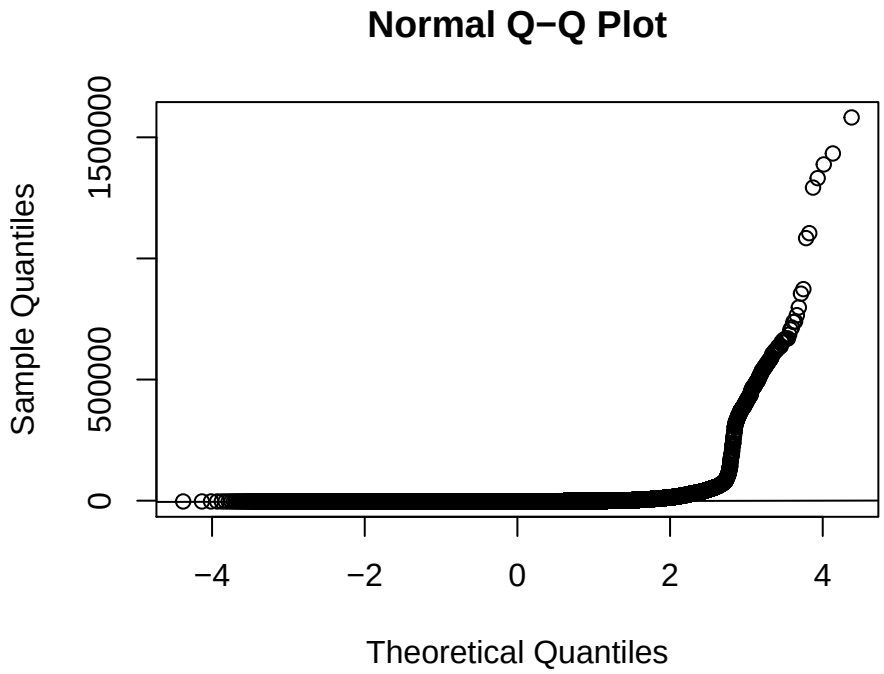
```
##
##  Downloading file 1 of 1: ‘games.csv‘
```

When looking at the amount of video game users gained per month on average, January and December come out on top. January had an average gain of 131 and December with 212. This further leads us to believe that indeed the holidays play a role in the spike of video game usage since video games are common gifts. The summer months also saw games which leads us to believe that even though less people play per average in the summer these months still experience a gain in playership since some people especially students do have more free time. The year 2021 actually had a negative amount of new players which leads us to believe that Covid-19 prevented many people from joining the gaming market. The lack of financial stability from the pandemic may have played a huge role here.

When it comes down to the actual average number of players per month it is quite spread out fairly. The summer months do quite well in this metric along with the months of January and December. When it comes down to the average number of players per year we can see that it generally has been increasing for the past decade but during the pandemic the number has shot up. The gaming market seems to be thriving and the pandemic seems to have made the industry better off.

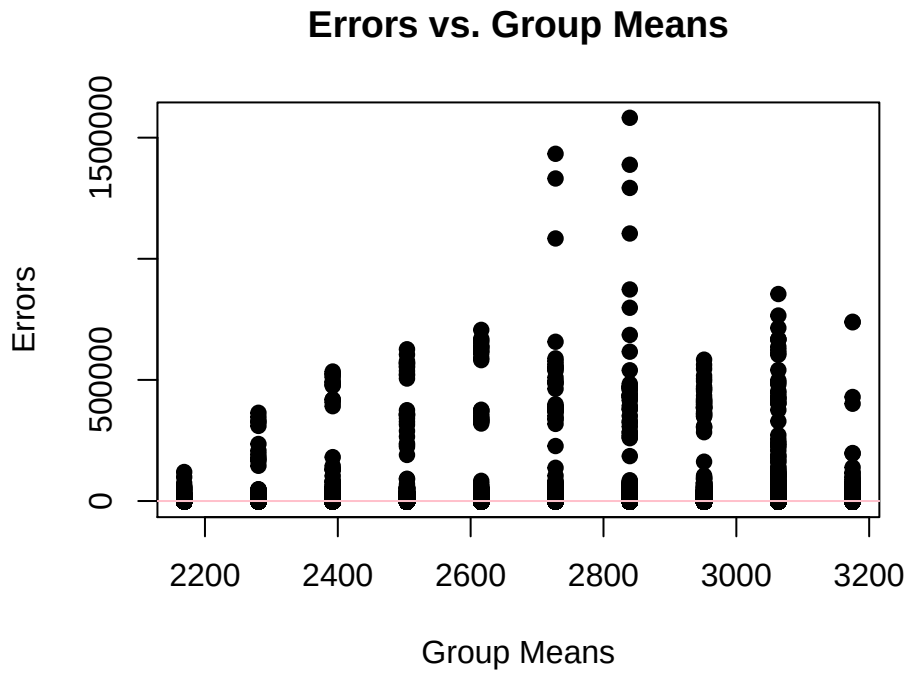
Plot Intepretation:

From this Errors vs Group Means graph, it looks as though we have some have outliers. The outlier on the bottom left and the outlier on the top middle skew the y-axis scaling; however, the variance looks to be fairly equal throughout the plot.



Plot Interpretation:

From the Normal Q-Q Plot, it looks like the data skews heavily to the right side. The trend here is predictable. Popular steam games will obviously have more players per month than other smaller games. Some steam games have a large community;however, there may be factors that discourage new players.



Plot Interpretation:

From this Errors vs Group Means plot, we can see the variances in the plot are extremely varied. There appears to be alot of outliers.

Pairwise CIs:

CI: $\mu_{oct} - \mu_{jul}$

NUll hypothesis: $\mu_{oct} - \mu_{jul} = 0$

Alternative hypothesis: $\mu_{oct} - \mu_{jul}$ does not equal 0

The estimated difference in average monthly players is -1033, 720. Due to the presence of 0 in the confidence interval, we can conclude that there is no significant difference the average monthly players between July and October. This is significant since it disproves the theory that summer has more players due to summer break.

CI: $\mu_{apr} - \mu_{jun}$

NUll hypothesis: $\mu_{apr} - \mu_{jun} = 0$

Alternative hypothesis: $\mu_{apr} - \mu_{jun}$ does not equal 0

The estimated difference in average monthly players is -824.3, 995,5. Due to the presence of 0 in the confidence interval, we can conclude that there is no significant difference the average monthly players between April and June. This is significant since it further disproves the theory that summer has more players due to summer break.

CI: $\mu_{aug} - \mu_{jul}$

NUll hypothesis: $\mu_{aug} - \mu_{jul} = 0$

Alternative hypothesis: $\mu_{aug} - \mu_{feb}$ does not equal 0

The estimated difference in average monthly players is -1122, 605. Due to the presence of 0 in the confidence interval, we can conclude that there is no significant difference the average monthly players between August and February. This is significant since it disproves the theory that summer has more players due to summer break.

Regression Analysis

Because our data set contains not much information besides time and amount of player, it is tough for us to create a linear regression model. So, I will focus on one game call Total War: WARHAMMER II and add more information I found online as predicted variables to the data.

```
## # A tibble: 1 x 7
##   gamename                 year month       avg  gain  peak avg_peak_perc
##   <chr>                   <dbl> <fct>     <dbl> <dbl> <dbl> <chr>
## 1 Total War: WARHAMMER II  2021 February 28418. 3795. 48027 59.1701%
```

```
##                       ï..Name   Month Year
## 1 Blood for the Blood God II October 2017
```

```
##          ï..Date Final.price Year Month
## 1 6/12/2017 0:00       59.99 2017  June
```

```
##   Year Month Average.monthly.viewers
## 1 2017     8                      38
```

Data frame warhammer extracts the original dataset by filtering only the game Total War: WARHAMMER II. Next, data frame dlc contains the Downloadable content(new package) releasing month and names. Next, the data frame price contains the price record of this game. And last, the data frame viewer contains the average twitch viewer per month for this game.

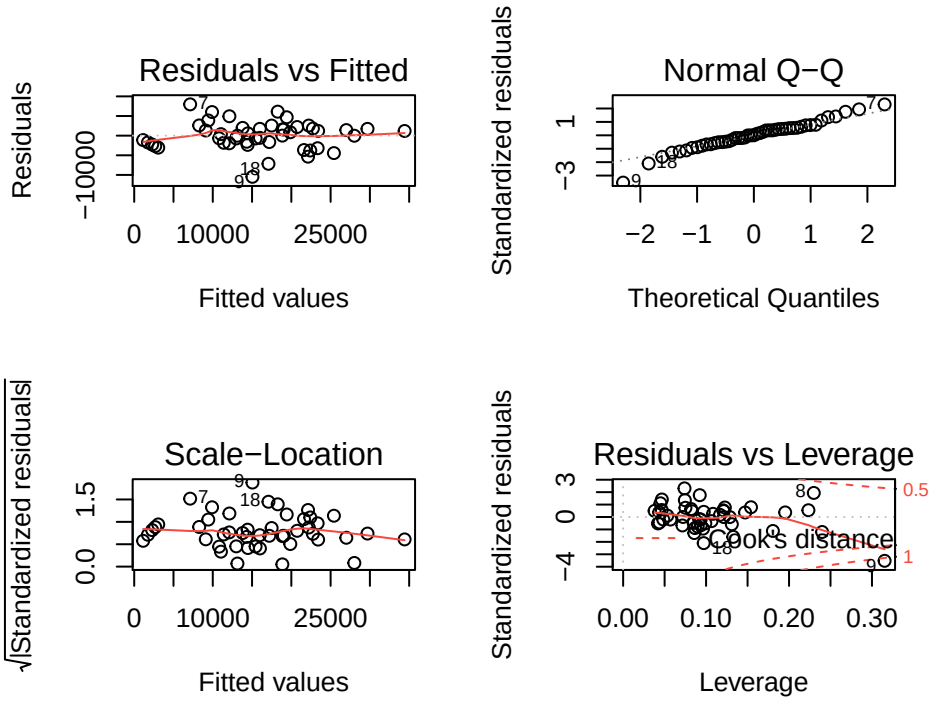
```
##   year.month  avg update min.price months.after.release Average.monthly.viewers
## 1   2017 Apr 7.45      0     59.99                    0                       0
```

After some data wrangling, our final data warhammer_new has six columns. Variable year.month is the time index per month. Variable avg is the average player per month. The variable update is a categorical variable that tells if this month has an update (1) or not(0). Variable min.price is the lowest price for the game each month. Variable months.after.release tell us how long the game has been released. And, variable Average.monthly.viewers tells us the average twitch viewers per month.

We are going to use variables update, min.price, months.after.release, and Average.monthly.viewers to predict avg.

```
##
## Call:
## lm(formula = avg ~ . - year.month, data = warhammer_new)
##
## Residuals:
```

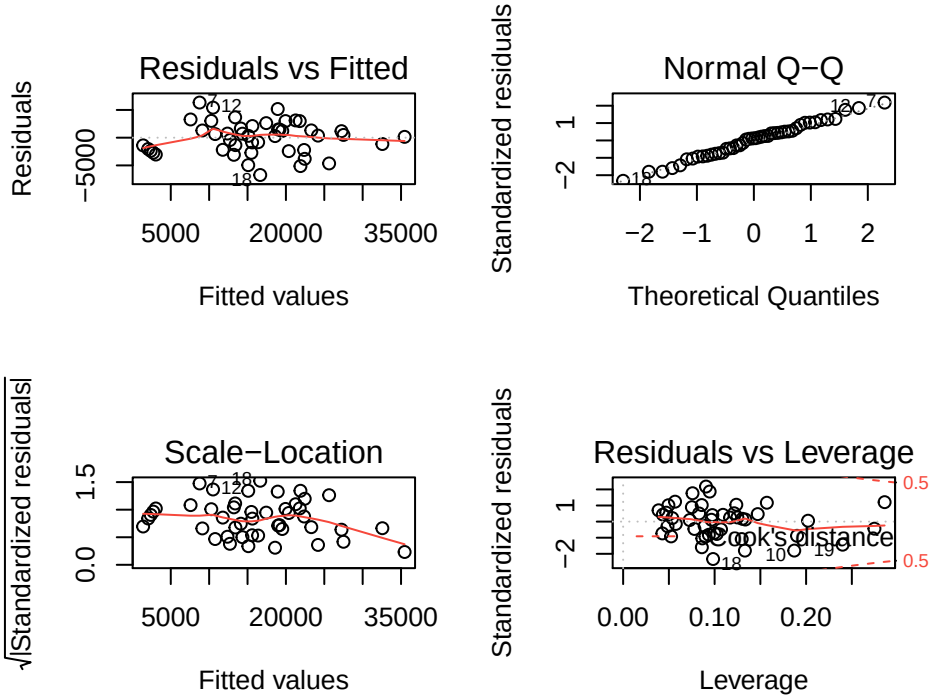
```
##      Min       1Q   Median       3Q      Max
## -10481.4  -1885.2      8.8   1770.9   7966.4
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              4684.883   2308.342   2.030   0.0488 *
## update                   4637.127   1334.011   3.476   0.0012 **
## min.price                 -59.161     32.558  -1.817   0.0763 .
## months.after.release      384.489     45.537   8.443 1.36e-10 ***
## Average.monthly.viewers     5.993      1.280   4.680 2.98e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3592 on 42 degrees of freedom
## Multiple R-squared:  0.8289, Adjusted R-squared:  0.8126
## F-statistic: 50.88 on 4 and 42 DF,  p-value: 1.45e-15
```

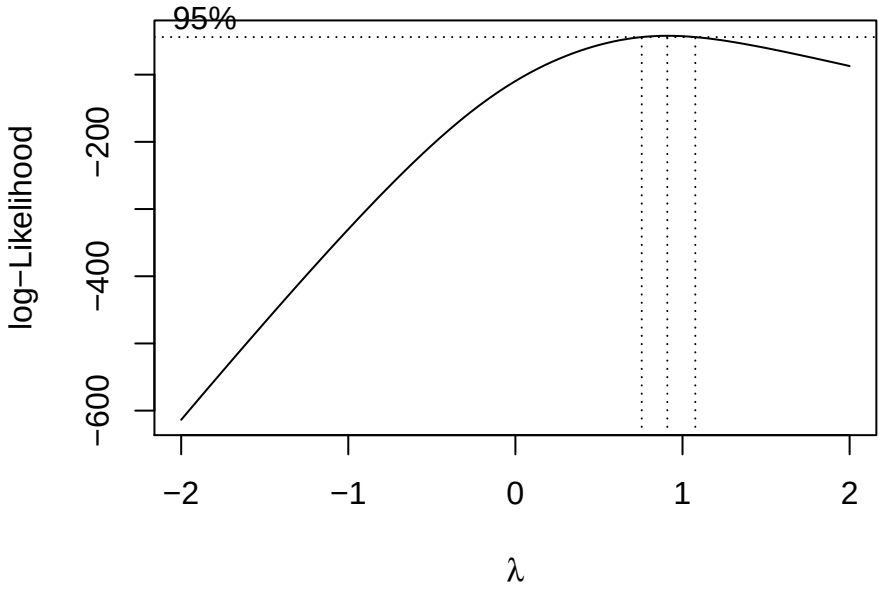


After applying the linear regression, we can see the r square of the model is 0.8289. which indicated the model is a good fit. By observing the residuals plot, we can say the variance of the model is pretty consistent except for data 9. By observing the normal QQ plot, all the points are fitted well to the line except for data 9; our model is normal. And by observing the residuals vs. leverage plot, again, only data 9 exceed Cook's distance. Data 9 is clearly an outlier. We should remove it from our model.

```
##
## Call:
## lm(formula = avg ~ . - year.month, data = warhammer_new[-9, ])
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -6731   -2211     350    1611    6333
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               4250.591   1963.060   2.165  0.03624 *
## update                    3232.699   1182.278   2.734  0.00919 **
## min.price                  -47.547     27.790  -1.711  0.09465 .
## months.after.release       330.572     40.793   8.104 4.78e-10 ***
## Average.monthly.viewers      8.753      1.274   6.868 2.53e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3051 on 41 degrees of freedom
## Multiple R-squared:  0.8744, Adjusted R-squared:  0.8621
## F-statistic: 71.34 on 4 and 41 DF,  p-value: < 2.2e-16
```





After removing the outlier, our model looks perfect with an r-squared of 0.8744. And, by observing the box cox plot, 1 is within 95% CI. for the maximum log-likelihood. That shows a transformation is unnecessary.

We want to further improve our model by adding a potential two-way interaction variable. We will run a step-wise both-way selection to find a better model.

The best model considering lowest AIC is lm(formula = avg ~ Average.monthly.viewers + months.after.release + update + min.price + Average.monthly.viewers:min.price, data = warhammer_new)

The best model considering lowest BIC is lm(formula = avg ~ Average.monthly.viewers + months.after.release + update, data = warhammer_new)

We will use leave-one-out cross-validation to see which model is more accurate.

```
##              Mean Squared Error
## first model            10447407
## second model           10750149
```

We can see the first model lm(formula = avg ~ Average.monthly.viewers + months.after.release + update + min.price + Average.monthly.viewers:min.price, data = warhammer_new) has a better accuracy.

```
##       avg update min.price months.after.release Average.monthly.viewers
## 1 23160.5      0     59.99                    48                      620
```

```
##        fit      lwr      upr
## 1 22598.77 20077.24 25120.29
```

I use real-life data to predict the average number of players in April 2021. The prediction by using our data is 22728.4 players with a 95% C.I of (20308.17,25148.63). The actual average number of players for April 2021 is 23,160.5. Thus, our prediction is pretty accurate.

Appendix

```
knitr::opts_chunk$set(
    error = FALSE,
    message = FALSE,
    warning = FALSE,
    echo = FALSE, # hide all R codes!!
    fig.width=5, fig.height=4,#set figure size
    fig.align='center',#center plot
    options(knitr.kable.NA = ''), #do not print NA in knitr table
    tidy = FALSE #add line breaks in R codes
)
rm(list=ls())
tuesdata <- tidytuesdayR::tt_load('2021-03-16')

games <-
  readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2021/2021-0
games
par(mfrow=c(2,2))
library(ggplot2)
games$month = factor(games$month, levels = month.name)
p1 <- ggplot(games , aes(month, avg)) +
  geom_boxplot(outlier.shape=NA) +
  scale_x_discrete(guide = guide_axis(n.dodge=3)) +
  labs(title="Average Number of Players by Month Including All Games, 2010-2021",
       xlab = "Month", ylab = "Average Number of Players Including All Games")+
      ylim(0, 1500)
p1
avg.month.mean = aggregate(avg ~ month, data = games[2514:83631, 1:7], mean)

ggplot(avg.month.mean, aes(avg.month.mean$month, avg.month.mean$avg)) +
    geom_point()+
    labs(title="Mean of All Game's Average Playership by Month, 2010-2020",
         x = "Month", y = "Mean of All Game's Average Playership")+
    ylim(2400,3000)
avg.year.mean = aggregate(avg ~ year, data = games, mean)
avg.year.mean <- avg.year.mean[1:9, 1:2]

ggplot(avg.year.mean, aes(avg.year.mean$year, avg.year.mean$avg)) +
  geom_point()+
  labs(title='Average Playership by Year, 2010-2020', x='Year',
```

```
       y='Annual mean of Average Players per Game')+
  ylim(1700,3500)

#6 months before pandemic (September 2019[21617] (cutoff between august2019 and september2019) - Februa
#6 months after start of pandemic (March 2020[14658] (Cuttoff between February and March) - August 2020

afterpan <- aggregate(avg ~ month, data = games[14658:7471, 1:7], mean)
afterpan$month = factor(afterpan$month, levels = month.name)

beforepan <-aggregate(avg ~ month, data = games[21617:14659, 1:7], mean)
month.name2 <- c("September", "October", "November", "December",
                 "January", "February")
beforepan$month = factor(beforepan$month, levels = month.name2)

plot1 <- ggplot(beforepan, aes(beforepan$month, beforepan$avg)) +
  geom_point()+
  labs(title='Playership 6 Months BEFORE \nThe Start of Covid-19 Pandemic',
       x='Month', y='Monthly mean of Average number of Players per Gamee')+
  ylim(2200, 3900)

plot2 <- ggplot(afterpan, aes(afterpan$month, afterpan$avg)) +
  geom_point()+
  labs(title='Playership 6 Months AFTER \nThe Start of Covid-19 Pandemic',
       x='Month', y='Monthly mean of Average number of Players per Game')+
  ylim(2200, 3900)


require(gridExtra)
grid.arrange(plot1, plot2, ncol=2)
library(tidytuesdayR)
library(tidyverse)
tt <- tt_load("2021-03-16")
data = as.data.frame(tt$games)
head(data)
gain.year.mean = aggregate(gain ~ year, data = data, mean)
gain.year.sd = aggregate(gain ~ year, data = data, sd)
gain.year.mean
avg.month.mean = aggregate(avg ~ month, data = data, mean)
avg.month.sd = aggregate(avg ~ month, data = data, sd)
avg.month.mean
avg.year.mean = aggregate(avg ~ year, data = data, mean)
avg.year.sd = aggregate(avg ~ year, data = data, sd)
avg.year.mean
gainmonth.model = lm(avg ~ year, data = data)
data$ei2 = gainmonth.model$residuals
qqnorm(gainmonth.model$residuals)
qqline(gainmonth.model$residuals)
plot(gainmonth.model$fitted.values, gainmonth.model$residuals, main =
       "Errors vs. Group Means",xlab = "Group Means",ylab = "Errors",pch = 19)
abline(h = 0,col = "pink")
find.means.1 = function(data,fun.name = mean){
  a = length(unique(data[,2]))
  b = length(unique(data[,3]))
```

```
  means.A = by(data[,4], data[,2], fun.name)
  means.B = by(data[,4],data[,3],fun.name)
  means.AB = by(data[,4],list(data[,2],data[,3]),fun.name)
  MAB = matrix(means.AB,nrow = b, ncol = a, byrow = TRUE)
  colnames(MAB) = names(means.A)
  rownames(MAB) = names(means.B)
  MA = as.numeric(means.A)
  names(MA) = names(means.A)
  MB = as.numeric(means.B)
  names(MB) = names(means.B)
  results = list(A = MA, B = MB, AB = MAB)
  return(results)
}
the.means.1 =find.means.1(data,mean)
the.sizes.1=find.means.1(data,length)
the.sds.1=find.means.1(data,sd)
overall.treatment.mean = the.means.1$AB
data.means = by(data$avg, data$month, mean)
data.nis = by(data$avg, data$month, length)
MSE.data = anova(lm(data$avg ~ data$month, data = data))[2,3]
give.me.CI = function(ybar,ni,ci,MSE,multiplier){
  if(sum(ci) != 0 & sum(ci !=0 ) != 1){
    return("Error - you did not input a valid contrast")
  } else if(length(ci) != length(ni)){
    return("Error - not enough contrasts given")
  }
  else{
    estimate = sum(ybar*ci)
    SE = sqrt(MSE*sum(ci^2/ni))
    CI = estimate + c(-1,1)*multiplier*SE
    result = c(estimate,CI)
    names(result) = c("Estimate","Lower Bound","Upper Bound")
    return(result)
  }
}
ci1 = c(0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 1, 0 )
t.a.1 = qt(1-.05/2, sum(data.nis) - length(data.nis))
CI = give.me.CI(data.means, data.nis, ci1, MSE.data, t.a.1)

ci2 = c(1, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0 )
CI2 = give.me.CI(data.means, data.nis, ci2, MSE.data, t.a.1)

ci3 = c(0, 1, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0 )
CI3 = give.me.CI(data.means, data.nis, ci3, MSE.data, t.a.1)

X2 = data$avg
X3 = data$gain
X4 = data$year

library(tidyverse)
library(tidytuesdayR)
library(dplyr)
library(astsa)
```

```
library(zoo)
library(tseries)
library(forecast)
library(MASS)
library(leaps)
library(boot)
warhammer <- games %>% filter(gamename == "Total War: WARHAMMER II")
dlc <- read.csv("Warhammer_II_DLC.csv")
price <- read.csv("price-history-for-Warhammer-II.csv")
viewer <- read.csv("Average_monthly_viewers.csv")
head(warhammer,1)
head(dlc,1)
head(price,1)
head(viewer,1)
warhammer <- warhammer %>% mutate(month.num = match(warhammer$month,month.name))
warhammer <- warhammer %>%
  mutate(year.month = paste(year,month.abb[month.num],sep = " "))
dlc <- dlc %>% mutate(month.num = match(dlc$Month,month.name))
dlc <- dlc %>% mutate(year.month = paste(Year,month.abb[month.num],sep = " "))
dlc <- dlc %>% group_by(year.month) %>% slice(1)
dlc <- dlc %>% mutate(update = 1)
warhammer_new <- merge(warhammer[,c(4,9)],dlc[,5:6],by = "year.month", all.x = TRUE)
warhammer_new[is.na(warhammer_new)] <- 0

price <- price %>% mutate(month.num = match(price$Month,month.name))
price <- price %>% mutate(year.month = paste(Year,month.abb[month.num],sep = " "))
price <- price %>%
  group_by(year.month) %>%
  summarise(min.price = min(Final.price, na.rm = T),)

warhammer_new <- warhammer_new[order(warhammer_new$year.month), ]
warhammer_new <- merge(warhammer_new,price,by = "year.month", all.x = TRUE)
warhammer_new[is.na(warhammer_new)] <- 59.99
warhammer_new <- warhammer_new %>%
  mutate(months.after.release = cbind(0:(nrow(warhammer_new)-1)))

viewer <- viewer %>% mutate(year.month = paste(Year,month.abb[Month],sep = " "))
warhammer_new <- merge(warhammer_new,viewer[,3:4],by = "year.month", all.x = TRUE)
warhammer_new[is.na(warhammer_new)] <- 0

head(warhammer_new,1)
par(mfrow= c(2,2))
warhammer_new <- as.data.frame(warhammer_new)
lm.data.fit <- lm(avg ~ . - year.month,data = warhammer_new)
summary(lm.data.fit)
plot(lm.data.fit)
par(mfrow= c(2,2))
lm.data.fit <- lm(avg ~ . - year.month,data = warhammer_new[-9,])
summary(lm.data.fit)
plot(lm.data.fit)
par(mfrow= c(1,1))
boxcox(lm.data.fit)
warhammer_new <- as.data.frame(warhammer_new[-9,])
```

```
warhammer_new$months.after.release <- as.numeric(warhammer_new$months.after.release)
model0 = lm(avg~1, data = warhammer_new)
modelF = lm(avg ~ (. - year.month)^2,data = warhammer_new)
step(model0, scope=list(lower=model0, upper=modelF), direction="both")
model0 = lm(avg~1, data = warhammer_new)
modelF = lm(avg ~ (. - year.month)^2, data = warhammer_new)
step(model0, scope=list(lower=model0, upper=modelF),
     direction="both",k = log(nrow(warhammer_new)))
n = nrow(warhammer_new)
model1 <- glm(formula = avg ~ Average.monthly.viewers + months.after.release
              + update + min.price + Average.monthly.viewers:min.price,
              data = warhammer_new)
model2 <- glm(formula = avg ~ Average.monthly.viewers + months.after.release
              + update, data = warhammer_new)

error <- rbind(cv.glm(warhammer_new,model1,K = n)$delta[1],
               cv.glm(warhammer_new,model2,K = n)$delta[1])
rownames(error) <- c("first model","second model")
colnames(error) <- "Mean Squared Error"
error
model <- lm(formula = avg ~ Average.monthly.viewers + months.after.release
            + update + min.price + Average.monthly.viewers:min.price,
            data = warhammer_new)
new_data = data.frame(avg = 23160.5, update = 0, min.price = 59.99,
                      months.after.release = 48, Average.monthly.viewers = 620)
new_data
predict(model,newdata = new_data,interval = "confidence",level=0.95)
```