Forecasting of Server Usage for the Steam Gaming Platform and Distribution Service

Project Report

Simon Fraser University
STAT 485: Applied Time Series Analysis
Gary Parker
December 08, 2022

Team 5
Reid Gendron – 301311069
Sean Yang – 301364025
Yiding Zhi – 301403905

Executive Summary

Online gaming platform Steam, is one of the most popular computer platforms for players to connect online with one another and download the latest digital releases of games. SteamDB, an unaffiliated third-party website, collects Steam related data and was created to provide insights into the official Steam database. The company has collected the records of Steam's player counts, including the number of in-game active users, from January 2013 to October 2022. Using these records, we provide a useful forecast into the number of active users, for use by the Steam management team to gain a better understanding of the necessary equipment required to handle their expected server usage for the coming terms.

Some assumptions made in the process of this project include the general assumption that SteamDB has collected this data accurately and that it is well maintained. Additionally, we need to assume that the data collected by SteamDB is truly representative of the official Steam database.

The time series data in this report is used to analyze the pattern in the daily user count for the Steam platform. We could conclude that there is a strong seasonality in daily active users on Steam, and the period time is seven days. We created a seasonal ARIMA model that could forecast an estimate of the number of future active players.

1 Introduction

There are roughly 3.2 billion gamers worldwide, easily earning gaming the reputation of being the single most popular hobby around the world. This accounts for approximately 40% of the entire population. As this may be hard to believe, gaming has become a very mainstream form of entertainment and is constantly growing in accessibility. With a very diverse set of devices to play on, ranging from consoles, to computers, to smartphones, the industry has become bigger than the music and movie industries in earnings combined. The most popular gaming device is the PC, with a substantial 1.7 billion gamers. Steam is the leading online game distribution platform for PC, raking in massive business from the commission of game purchases, rentals, in-game add-ons, downloadable content, licensing fees and more. As the lead platform in the industry, there is massive server usage with maintenance required to hold its top spot in this sector.

The data we will be working with has been collected by SteamDB, an unaffiliated third-party, collecting Steam data by using a combination of data mining, web scraping, and APIs made publicly available from Steam itself. This data can give us a very useful insight into the true server usage statistics of the official Steam database. The data set consists of the daily user counts from 2004 to 2022, and began recording the number of active users in 2017.

2 Objectives

Using this data, we plan to analyze the number of gamers using the Steam platform to gain a deeper understanding of the correlation and response that is happening between concurrent daily player counts. Using this understanding, we will attempt to fit the best model from what we have learned throughout the course and apply it to the series using the R statistical software. Once a sufficient model is fit, we use it for forecasting in order to provide Steam with a rough estimate of their server usage for the following 6 months to a year. Additionally, a good understanding of the type of model fit can help us to study extreme events and use it to determine probable shocks during this time.

3 Analysis

3.1 Preprocessing

We read the data into R and stored the data into a variable named *steam.raw* using the function *read.csv* below,

```
> steam.raw <- read.csv("chart.csv", header = True)
```

setting the header parameter to *TRUE* in order to use the header line of the csv file as column names. Upon examining the data, we found that there are four recorded variables: DateTime, Users, In.Game, and Flags. We also found that the variable Flags is completely missing its values, and In.Game is missing roughly 73% of its values.

```
> colMeans(is.na(steam.raw))
```

```
  DateTime      Users    In.Game      Flags
0.00000000 0.06275304 0.73279352 1.00000000
```

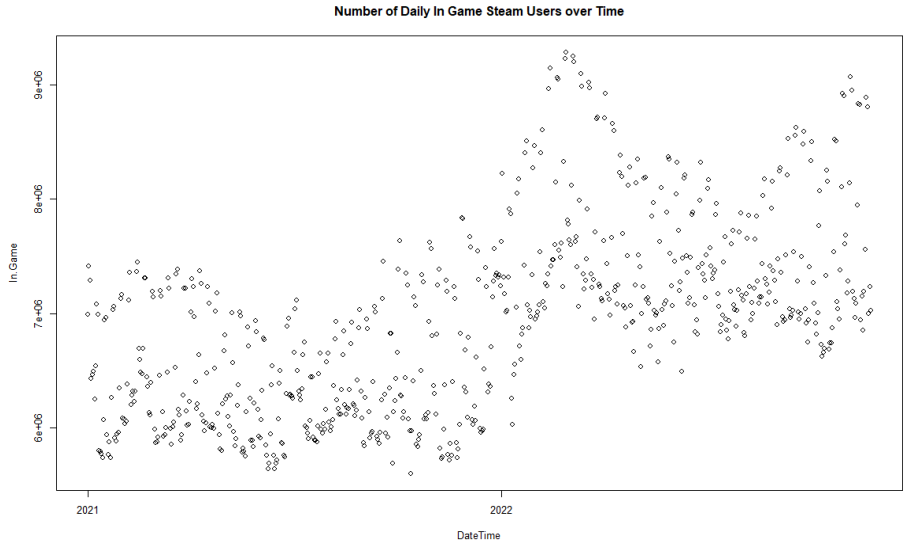
Considering the missing observations for Flag, we choose to omit this variable. The high percentage of missing values in In.Game is not alarming as the majority of these values are from before 2017. However, even if we were to use data from 2017 and on, there are a few gaps left in this time series. Records from March 21, 2017 to March 22, 2017, February 21, 2018 to February 25, 2018 and February 18, 2019 to February 19, 2019 all have missing observations. These unreported values may be the results of technical errors in the data collection process. We thought about replacing the NAs with zeros, as there may have been a server outage for these dates. However, this will greatly alter the original distribution of the dataset by reducing the mean E(Y). We then thought about imputing the NAs with E(Y), but this may also change the distribution of the original data. Thus, we only keep records from 2020 and on such that we are left with a continuous time series.

3.2 Assumptions

The cleaned time series keeps the data recent, and relevant to the server capabilities of today's technology, however, in using this data we need to make some assumptions about its integrity. We assume that the Steam data provided by SteamDB has been collected accurately, and that it is well maintained. We also need to assume that the data is truly representative of the official Steam database in order to make any conclusions about Steam's server usage. Our forecast will be a rough approximation because of this, however, if Steam were truly interested in our findings, we could use our analysis to fit a model on the true data, given their consent. Finally, we decide to further exclude data from before 2020 in our analysis of the cleaned data, to reduce the effects of covid-19 on the number of in-game Steam players.

3.3 Trends

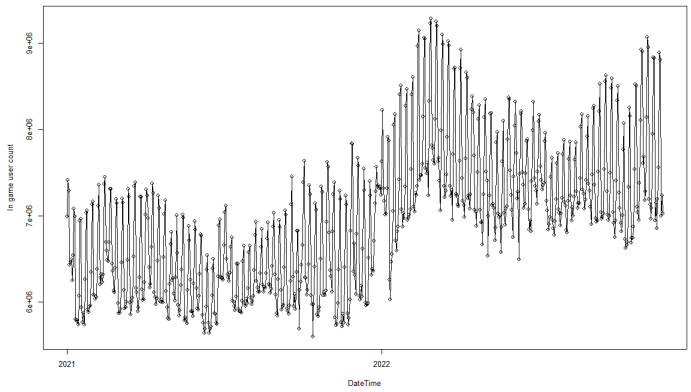
The visual trend of the data is a gradual positive increase of In.Game players over time. By the look of the series, we can make the conjecture that the series may be of the moving average model type, given its shape.



There also appears to be some annual seasonality happening with the number of in-game players, however, we will touch on this later. We now begin our model fitting using the Box-Jenkins approach.

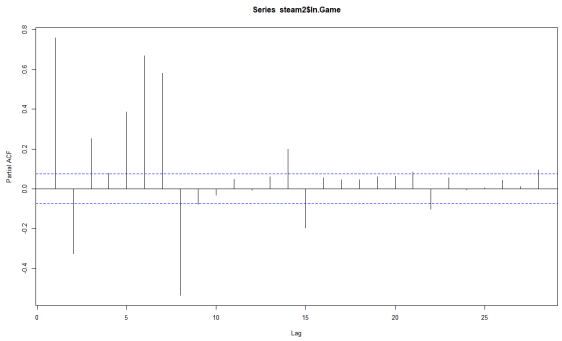
3.4 Box-Jenkins Method

Upon our first look at a visualization of the data, we now attempt to fit a model to the time series of in-game Steam players using the Box-Jenkins method. A line graph of the series can be seen below.

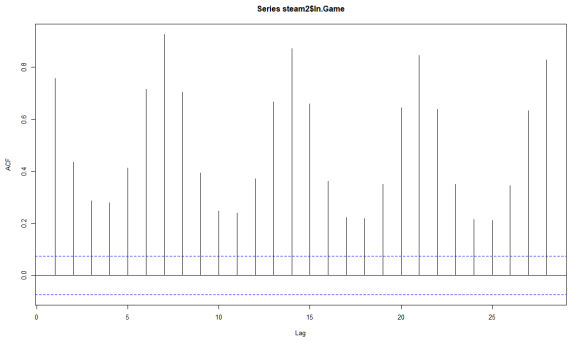


The data shows an upwards trend from 2021 to 2022. But there are clearly two "layers" of the data, meaning that the maximum values and minimum values have similar trends if the data is viewed "separately", but continuously, there is a clear seasonal trend.

Examination of the ACF, PACF, and EACF of the data:



PACF(1.1)



ACF(1.1)

From the ACF(1.1), we can see that the trend still exists, as the correlation reaches the max at around 0.8 for every lag7, and for lag2, lag3, lag4, and lag5, there is a small negative correlation, meaning that the observation at lag n is largely positively determined by the observation at lag n-7, and somewhat negatively determined by observations at lag2, lag3, lag4, and lag5.

From the PACF(1.1), we can see that the PACF reduced to 0 after lag8, if we exclude the two observations in lag14 and lag15.

From the EACF, we can see it does not make much sense.

Since the pure stationary MA and AR models can be identified by studying their ACF and PACF, as the table of Box-Jenkins Identification Tables:

By looking at the ACF for the series, we can see that the ACF does not decay exponentially, nor decays to zero after certain times, but instead there is a clear trend in the ACF for every 7 observations. For the PACF, we can see that the PACF dies out slowly exponentially, oscillating

exponentially of a damped sine wave. Thus we conclude that the original series is neither a pure AR(p), MA(q), nor a ARMA(p,q).

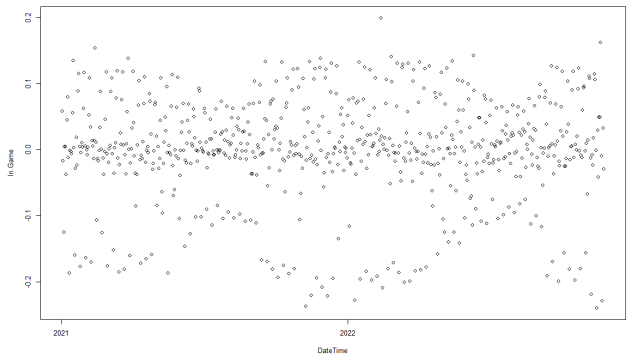
	AR(p)	MA(q)	ARMA(p,q)
ACF	Exponential Decay	0 for k > q	Exponential Decay
PACF	0 for k > p	Exponential Decay	Exponential Decay

Box-Jenkins Identification of MA or AR models Table

Box-Jenkins Approach, Transform In.Game

The next step is to stationary transform the series. We can use a log change approach for a stationary transformation. We named the transformed set as steam.log.
In.Game.log <- log(steam2$In.Game) - lag(log(steam2$In.Game))

First, we plot the transformed data, and its ACF, PACF.

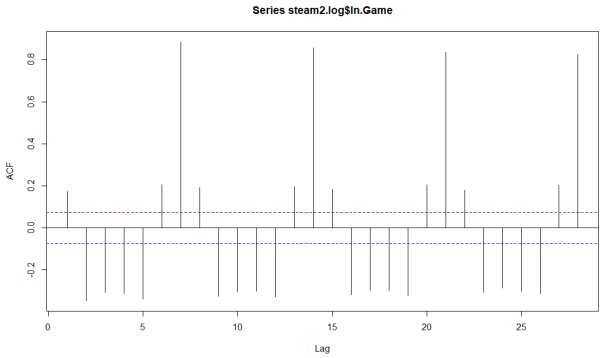


From the transformed plot, we can see that there is still a layer of the different values, most observations lay in the middle of the plot, where the transformed in.game = 0.0, where there are two layers at around transformed in.game = +/- 0.2.

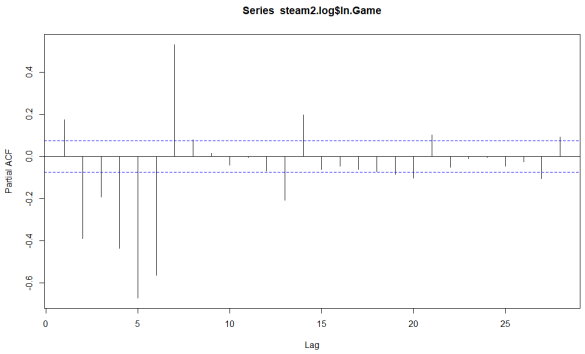
We need to examine the ACF and PACF.

From the ACF(2.1), we can see that the trend still exist, as the correlation reaches the max at around 0.8 for every lag7, and for lag2, lag3, lag4, and lag5, there is a small negative correlation, meaning that the observation at lag n is largely positively determined by the observation at lag n-7, and somewhat negatively determined by observations at lag2, lag3, lag4, and lag5.

The PACF(2.1) shows that there is indeed a diminishing return. We can see that the PACF reduced to 0 after lag8, if we exclude the two observations in lag13 and lag14.



ACF(2.1)



PACF(2.1)

Using the Box-Jenkins method, we can conclude that the series is neither a pure AR(p) or MA(q), nor a ARMA(p,q) model, by referring to the Box-Jenkins identification of MA or AR models table.

Dynamics Method (See Appendix for the Output Plot)

Using the dynamics method, start with assuming the series is an ARMA(p,q), consider all special cases for theta or phi equals 0. Given the following outputs from the dynamic method, we could see that when trying to fit AR(1), phi_1 close to when but the log likelihood is a small number, while we could continue to examine the details in Dynamics Method, we have our conclusions of the series is neither pure MA(q), AR(p), nor ARMA(p,q) from the previous Box-Jenkins method. We could save time and, rather, consider the clear seasonality of the series instead.

3.5 Seasonality

There is also a possibility that a multiplicative seasonal ARMA model may be better for our data. One hypothesis we could make is that there is a seasonal trend per seven days. After the first difference. The general upward trend has now disappeared.

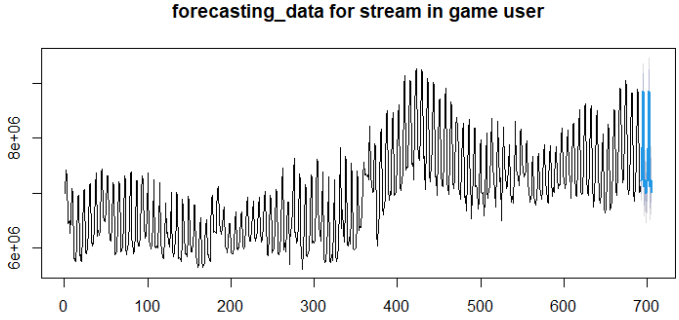
There is a clear seasonal trend based on the ACF.

```
m1.steam <- arima(steam2$In.Game,order=c(2,1,2),
      seasonal=list(order=c(0,1,1),period=7))
```

```
Call:
arima(x = steam2$In.Game, order = c(2, 1, 2), seasonal = list(order = c(0, 1,
    1), period = 7))

Coefficients:
          ar1     ar2      ma1      ma2     sma1
      -0.1502  0.6649  -0.1182  -0.8817  -0.6250
s.e.     NaN     NaN      NaN      NaN   0.0335

sigma^2 estimated as 3.679e+10:  log likelihood = -9295.87,  aic = 18601.74
```

Checking the histogram for standard error residuals, QQ plot for normality, and the ACF.

Forecasting



Based on the seasonal ARIMA model we get, we could do a forecasting of the number of active players in the future. Here, we predict the next two weeks of player counts. Compared to the last observation of 7,023,597 players, we expect the player capacity for the next two weeks to be an estimated 9,487,428 in-game active players.

4 Concluding Remarks

In the end, we decided on the seasonal ARIMA model. Since the observations are on a daily basis, it's clear that we can see a weekly pattern from the ACF. Thus, we chose an ARIMA(2,1,2) with seasonal (0,1,1) and a period of 7 units of time. This model can help the Steam company to predict possible player counts during the holidays so that they can be prepared to handle the sudden shock of in-game active players.

References

- https://www.statista.com/statistics/308330/number-stream-users/
- https://steamdb.info/app/753/graphs/
- https://steamdb.info/faq/
- https://www.statista.com/statistics/748044/number-video-gamers-world/
- https://dataprot.net/statistics/gamer-demographics/
- https://businessmodelanalyst.com/steam-business-model/?v=3e8d115eb4b3
- https://stackoverflow.com/questions/29522841/the-curious-case-of-arima-modelling-using-r
- http://jduras.github.io/files/teaching/eco5316/lec05slides.pdf

Appendix

Reading the Data into R

```
> library(dplyr)
> steam.raw <- read.csv("chart.csv", header = TRUE)
> glimpse(steam.raw) # needs dplyr
Rows: 494
Columns: 4
$ DateTime <chr> "2004-01-01 00:00:00", "2004-01-15 00:00:00", "2004-01-29 00:00:00", "2004-02-12 …
$ Users    <int> 84998, NA, NA, NA, NA, NA, 93635, 97480, NA, NA, NA, 117802, 114501, 111825, 1120…
$ In.Game  <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N…
$ Flags    <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N…
```

Preprocessing

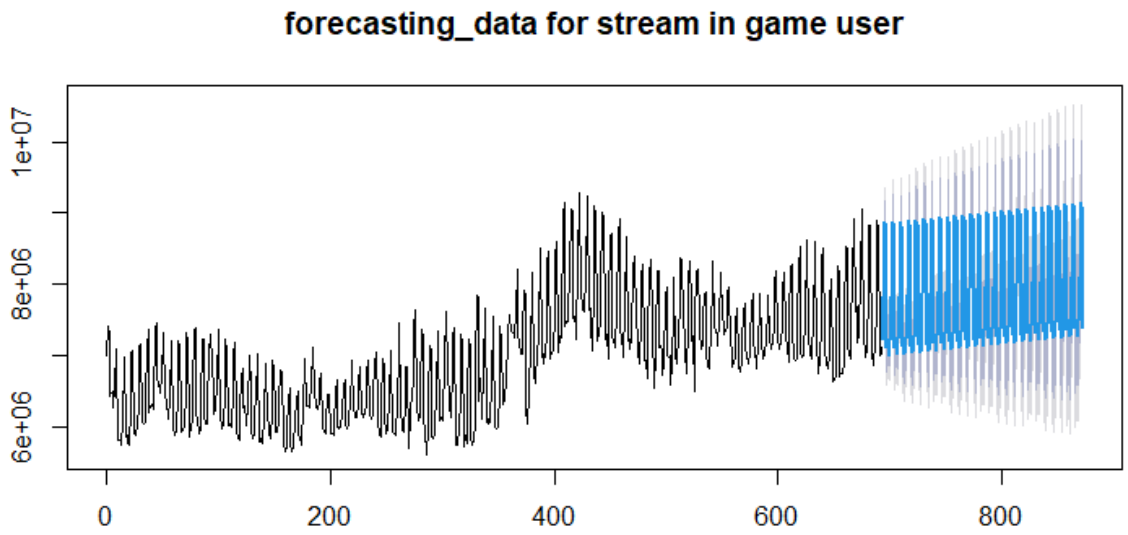
```
# Get the ratio of missing values for each variable
> colMeans(is.na(steam))
   DateTime      Users    In.Game      Flags
0.00000000 0.06275304 0.73279352 1.00000000


# Remove unnecessary variables
> steam.raw$Flags <- NULL

# Format DateTime observations to remove unreported Time
> steam.raw$DateTime <- as.Date(steam.raw$DateTime)

# Trim all records before 2020-01-01, store the trimmed data into steam2
n <- nrow(steam.raw)
idx <- match(as.Date("2020-01-01"), as.Date(steam.raw[,"DateTime"]))
steam2 <- steam.raw[idx:n,]

# Make sure the cleaned time series is continuous.
colMeans(is.na(steam2))
```

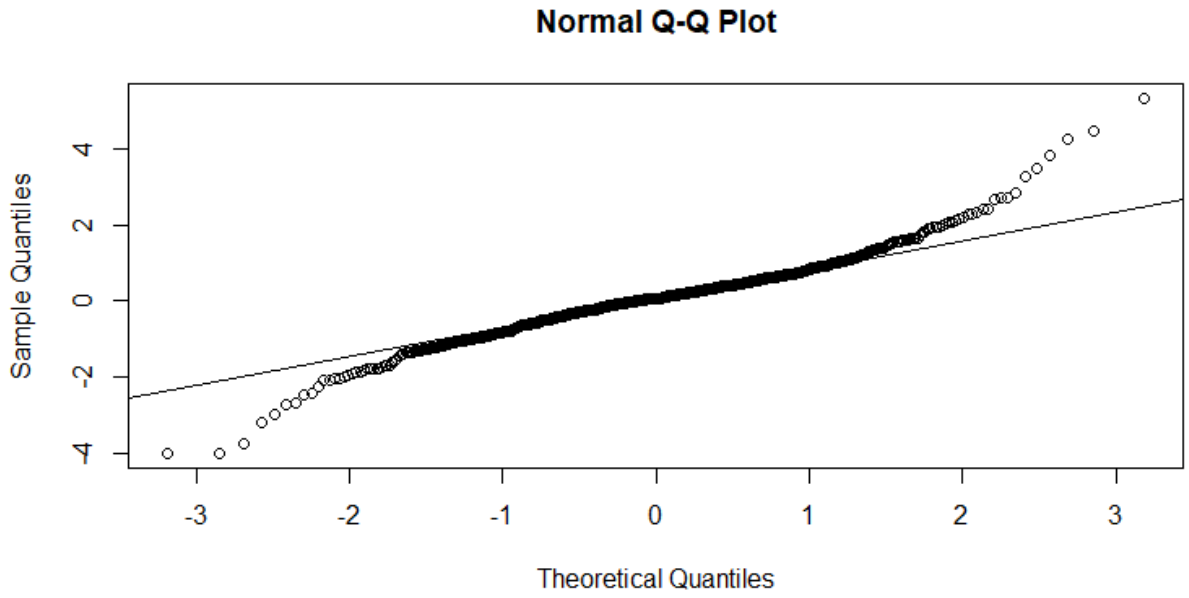
```
plot(steam2$In.Game, type = 'l',ylab='First Difference of User Count',xlab='Time')

plot(diff(steam2$In.Game), type = 'l',ylab='First Difference of User Count',xlab='Time')

acf(as.vector(diff(steam2$In.Game)),lag.max=100)

plot(diff(diff(steam2$In.Game),lag=7),type = 'l',xlab='Time',
     ylab='First and Seasonal Difference of User Count')


m1.steam <- arima(steam2$In.Game,order=c(2,1,2),seasonal=list(order=c(0,1,1),period=7))
m1.steam

plot(window(rstandard(m1.steam)),ylab='Standardized Residuals m1',type='l')

acf(as.vector(window(rstandard(m1.steam))),
    lag.max=200,ci.type='ma')

hist(window(rstandard(m1.steam)),
     xlab='Standardized Residuals')


qqnorm(window(rstandard(m1.steam)))
qqline(window(rstandard(m1.steam)))
```

Forecasting

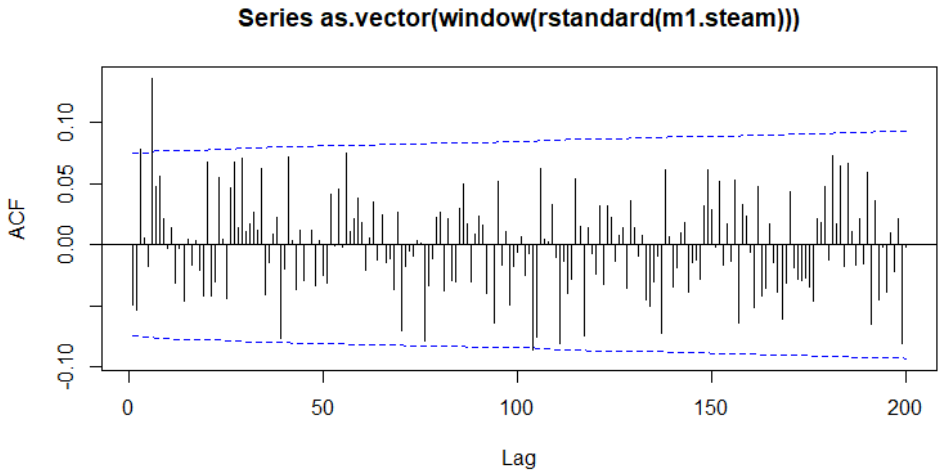
```
forecast_data <- forecast(m1.steam, 14)
print(forecast_data)
plot(forecast_data, main = "forecasting_data for stream in game user")
```

```
    Point Forecast   Lo 80   Hi 80   Lo 95   Hi 95
693        7220602 6973694 7467511 6842988 7598217
694        7826792 7520655 8132929 7358596 8294988
695        8868822 8533155 9204490 8355464 9382181
696        8822752 8472408 9173097 8286947 9358558
697        7109705 6750967 7468444 6561063 7658348
698        7232819 6869765 7595873 6677576 7788062
699        6999642 6633981 7365304 6440411 7558873
700        7213265 6827203 7599328 6622834 7803697
701        7817619 7420549 8214689 7210353 8424885
702        8870974 8467897 9274052 8254521 9487428
703        8821982 8415533 9228432 8200371 9443594
704        7116904 6708571 7525237 6492412 7741395
705        7236878 6827461 7646295 6610729 7863027
706        7009470 6599438 7419503 6382379 7636561
```

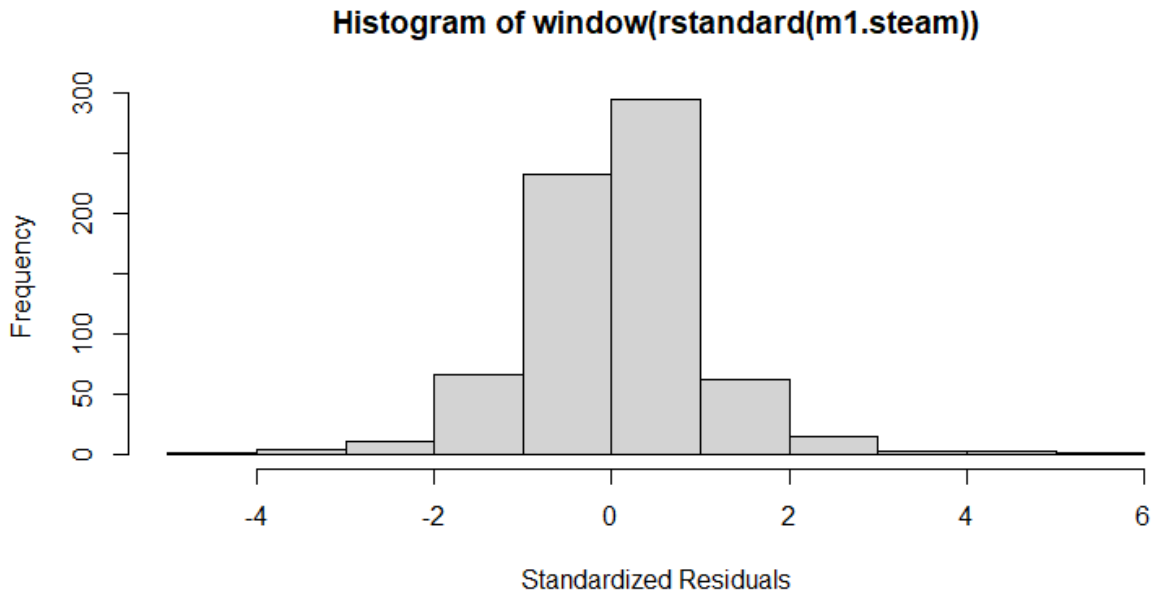




QQ plot for the seasonal model



Acf for the seasonal model, clear that no data lay out side +- 2 sd.



Standardized Residuals for the seasonal model.

Another forecasting example for the next 180 time units (daily).

EACF for the original series:

```
> eacf(steam2$In.Game)
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x x x x x x x x x x x  x  x  x
1 x x x x o x x x x x x  o  x  x
2 x x o o o x x x x o o  o  x  x
3 x x x o x o x x x x o  x  o  x
4 x x o o x o x x x o o  x  o  x
5 x x x o o x x x x x o  o  x  x
6 x x x o x x x x x x o  x  x  x
7 x x x x x o x o o x x  x  x  x
```

Dynamics Method Output.
Order (c(1,0,0), c(2,0,1), c(3,0,2), c(4,0,3))

```
> for (i in seq(1, length(order_list), by = 3)) {
+   my_order = c(order_list[i],order_list[i+1],order_list[i+2])
+
+   print(arima(steam2$In.Game, order = my_order))
+
+   }

Call:
arima(x = steam2$In.Game, order = my_order)

Coefficients:
        ar1   intercept
      0.7568  6937638.42
s.e.  0.0247    81803.57

sigma^2 estimated as 2.761e+11:  log likelihood = -10097.39,  aic = 20198.78

Call:
arima(x = steam2$In.Game, order = my_order)

Coefficients:
        ar1     ar2     ma1   intercept
      0.2983  0.2309  0.7890  6938130.20
s.e.  0.0730  0.0678  0.0556    69572.22

sigma^2 estimated as 2.328e+11:  log likelihood = -10038.61,  aic = 20085.22

Call:
arima(x = steam2$In.Game, order = my_order)

Coefficients:
         ar1      ar2     ar3     ma1     ma2   intercept
      0.3285  -0.5483  0.6893  0.7261  0.8872  6933692.01
s.e.  0.0316   0.0331  0.0402  0.0386  0.0219    82361.62

sigma^2 estimated as 1.964e+11:  log likelihood = -9980.63,  aic = 19973.26

Call:
arima(x = steam2$In.Game, order = my_order)

Coefficients:
         ar1      ar2      ar3     ar4      ma1      ma2     ma3  intercept
      1.3760  -0.3735  -0.6839  0.6455  -0.6534  -0.5751  0.7443  6984849.0
s.e.  0.0457   0.0912   0.0895  0.0435   0.0366   0.0419  0.0271   183175.2

sigma^2 estimated as 1.279e+11:  log likelihood = -9833.13,  aic = 19682.27
>
```

Dependencies:
STAT485-685_Rcode.R
BCA_functions_source_file.R - Used for data cleaning (variable.summary())