Macy's

Revenue Prediction

Group 14

Seng Fong Chang
Yiyang Duanmu
Ziyi Gao
Derek He
Yuxuan Yuan

macy's inc

macy's bloomingdales bluemercury

Table of Contents ★

Executive Summary

Competition in the department store industry has been more intense in recent years. Macy's, Inc. has been regarded as a leading expert in fashion by countless customers over the years. With one of the leading e-commerce platforms in the country, powered by macys.com and a mobile app, along with a countrywide network of shops, Macy's provides the most convenient and seamless shopping experience possible, delivering excellent deals in a variety of categories including clothing, home goods, cosmetics, and accessories. (Macy's About) Macy's gives customers even more options to shop and own their style by offering discounted merchandise at Macy's Backstage and a carefully chosen range of goods in a more compact store format under the name Market by Macy's. Macy's gives awe-inspiring experiences to millions of people every year via its Macy's Thanksgiving Day Parade® and Macy's 4th of July Fireworks® events. (Macy's About) Additionally, the department store assists consumers in commemorating all types of significant life events.

Our dataset consists of information about Macy's annual and quarterly revenue histories, along with their growth rates from 2008 through 2022. The revenue is the first line item on an income statement, and from that number, all other expenditures and expenses are deducted to arrive at the net income.

This project aims to determine how the pandemic has affected Macy's financial performance and to provide ideas for the department store's ongoing reaction to the pandemic. It is clear that Macy's revenue has been on a declining trend throughout the epidemic; this investigation aims to determine whether or not the pandemic is responsible for this revenue trend. We attempted a seasonal naïve model, moving average model, exponential smoothing, ARIMA models, and regression models. The regression models were of the utmost importance to us since we wanted to understand the impact that external variables had on Macy's revenue. Suppose a model that is driven entirely by the data and takes its inputs only from that data is successful. In that case, it will demonstrate that external influences are not the most significant predictors. The absence of a forecasting capacity is fascinating in and of itself. Revenue at Macy's is considerably impacted when external economic variables such as unemployment and GDP growth rates are taken into consideration. We used a time series linear regression model to examine these two factors impact on Macy's revenue. It is possible to observe the effects of the pandemic in both of these cases (the unemployment rate and the GDP growth rate), which indicates that the epidemic has significantly influenced Macy's Revenue.

Introduction and Motivation

Macy's is an American chain of high-end department stores founded in 1858 by Rowland Hussey. Macy's provides services through the Internet under two brand names: Macy’s and Bloomingdale’s which sells a wide range of merchandise, including men’s, women’s, and children’s apparel and accessories, cosmetics, home furnishings, and other consumer goods. The company had 130,000 employees and earned annual revenue of $24.8 billion as of 2017.

In 2020, COVID-19 devastated the global economy, especially the industry with close contact such as hotels, sports, performing arts, restaurants, clothing stores, and airlines. COVID-19 also forced physical shops to close and consumers to stay at home, therefore, companies such as Uber eats, DoorDash, and Amazon grew well during the pandemic. Specifically, Amazon was in a prime spot to capitalize. Amazon reported a near 200-percent rise in profits, accelerated by much of North America’s swift shift to exclusively online shopping. Amazon’s revenue was US$96.1 billion, up 37% from 2019, with profits rising to a jaw-dropping US$6.3 billion. The pandemic hasn’t only increased the company’s profits but also its expansion. Amazon expanded its fulfillment infrastructure by 50% in 2020, adding more than 250,000 employees in the process. For the first time in the company’s history, Amazon now employs more than one million workers around the world.

In this project, we wanted to find out how the pandemic has affected Macy's financial performance by forecasting Macy’s quarterly revenue. We also wanted to provide ideas for the department store's ongoing reaction to the future pandemic by forecasting Amazon’s quarterly revenue data and making comparisons to Macy’s forecast.

Data Description

To make forecasts for both Macy’s revenue and Amazon’s revenue, we used Macy’s revenue dataset and Amazon revenue dataset. Since we also incorporate external factors such as GDP Growth Rate and Unemployment Rate for our forecasts, we also used the historical GDP Growth Rate dataset and Unemployment Rate dataset.

[Macy’s Revenue]

- **Source:** Macrotrends (The Premier Research Platform for Long Term Investors)
- **Frequency:** Each record represents the quarterly revenue of Macy’s ranging from 2008 Q4 to 2022 Q2.

[Amazon’s Revenue]

- **Source:** Macrotrends (The Premier Research Platform for Long Term Investors)

- **Frequency:** Each record represents the quarterly revenue of Amazon ranging from 2009 Q1 to 2022 Q2.

[Unemployment Rate]

- **Source:** U.S. Bureau of Labor Statistics
- **Frequency:** Each record represents the monthly unemployment rate ranging from November 2008 to July 2022. We can then calculate the quarterly unemployment rate by averaging the months.

[GDP Growth Rate]

- **Source:** U.S. Bureau of Economic Analysis
- **Frequency:** Each record represents the quarterly GDP growth rate with adjusted inflation ranging from 2008 Q4 to 2022 Q2.

Revenue Analysis

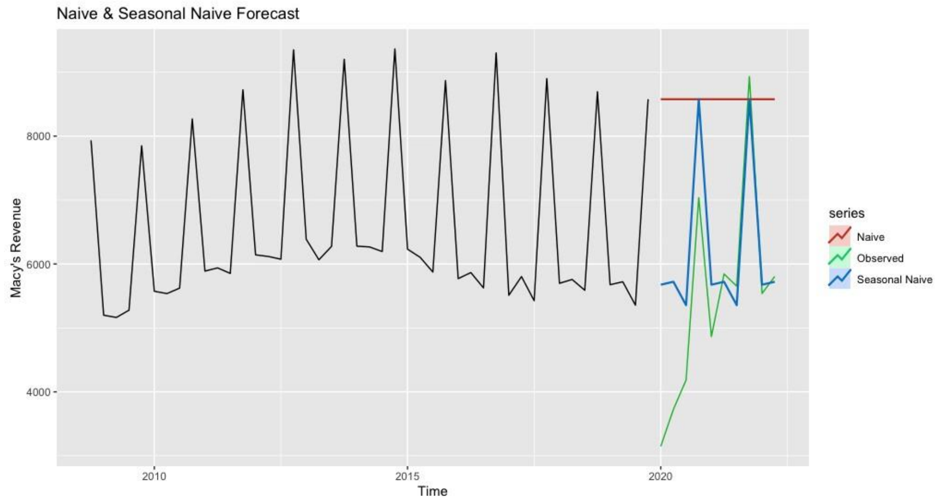
Our project mainly did two parts of analysis. We first try to better forecast Macy's revenue by deploying different models or incorporating external factors such as GDP Growth Rate and Unemployment Rate. Models we chose include Naive and Seasonal Naive Forecast, Moving Average, Exponential Smoothing, ARIMA, and Linear Regression with external factors. Secondly, we build forecast models on Amazon's revenue by deploying ARIMA models and Linear Regression with external factors. After the analysis, we want to find the difference between the model forecast accuracy between the two companies and provide recommendations to Macy's to better prepare for unexpected pandemics like COVID-19 in the future.

For the modeling, we decided to split our Macy's and Amazon's Revenue datasets into training and testing dataset. For the testing dataset, we left the last 10 months, from 2020Q1 to 2022Q2, as the testing dataset, and the previous quarters, from 2008 Q4 to 2019 Q4 as the training dataset. For the external factors, we also chose time frames that equal to the one for the training dataset.

Forecasting Macy's Revenue

Naive and Seasonal Naive Forecast

To analyze Macy's Revenue from the 2008 third quarter to the 2022 second quarter, our team first performed a Naive forecast and a seasonal naive forecast. As the Naive or seasonal naive forecast model only uses the last period observation or the last identical season as the next period forecast without actually predicting, it is better to use it for comparison with more complex models. However, comparing the accuracy of the naive forecast and seasonal naive forecast model, it is more obvious that the seasonal naive forecast model has better accuracy as it has lower MAPE.

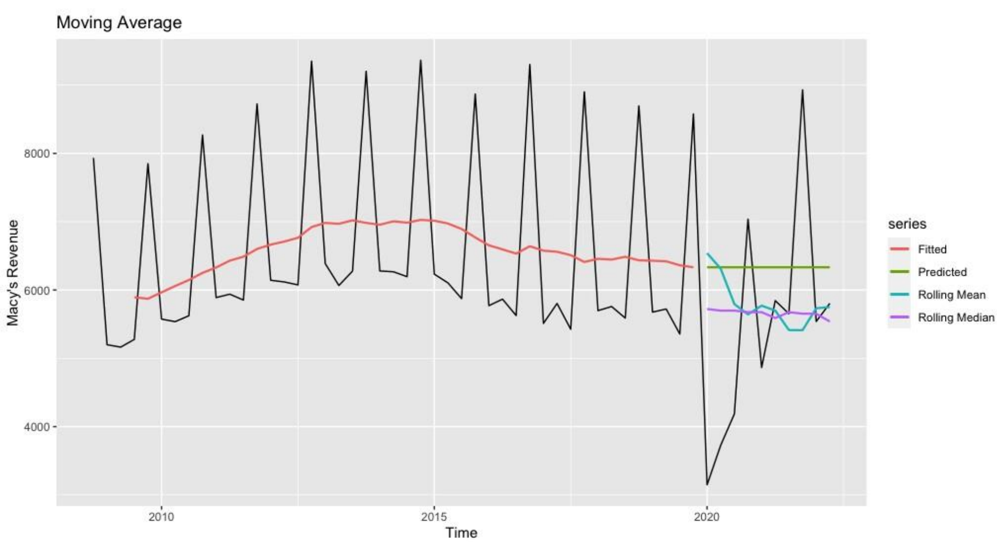


Naive Forecast	RMSE	MAPE
Training set	2142.21	22.47
Test set	3485.68	71.08

Seasonal Naive Forecast	RMSE	MAPE
Training set	279.06	3.60
Test set	1225.75	21.55

Moving Average

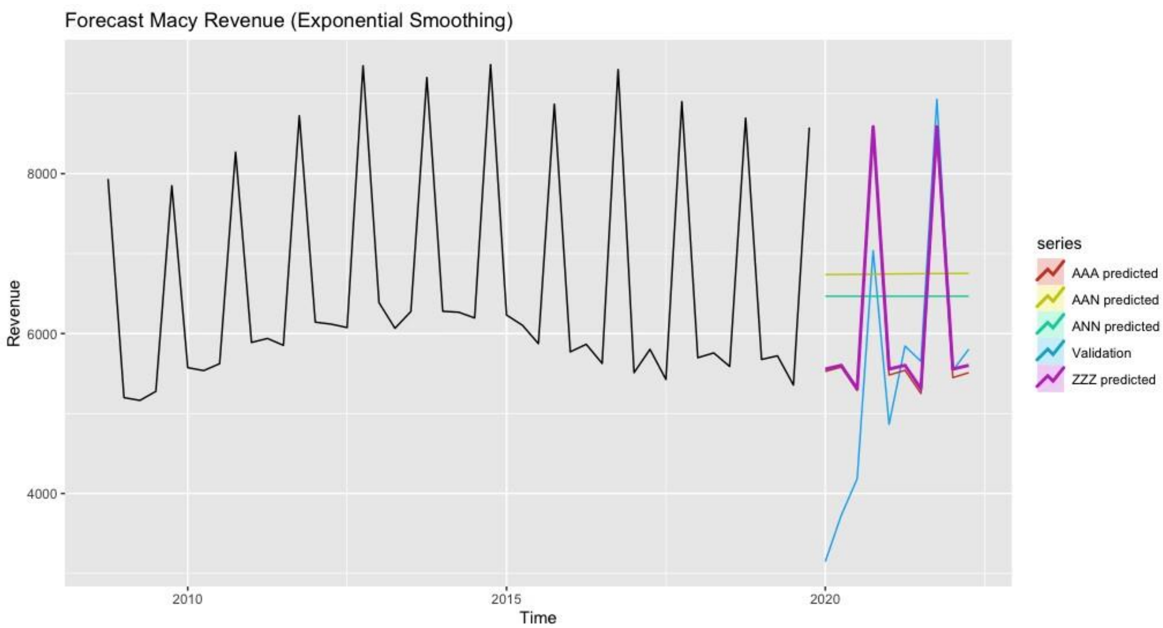
Then we performed the Moving Average smoothing method with trailing right and centered. After applying rollmean, rollmax, rollmedian with trailing right and checking the centered moving average, we identified that moving mean and moving median are more reasonable in presenting the observations in the graph, so we used a one-step ahead rolling forecast of these two methods and compared the accuracy of these two. Finally, in the moving average method, we concluded that using roll median is more accurate than using roll mean because it generated fewer errors. However, as they still have high MAPE, we want to apply more forecast methods to find the most accurate model of all.



Test Set	RMSE	MAPE
Roll Mean	1898.38	30.45
Roll Median	1618.55	25.49

Exponential Smoothing

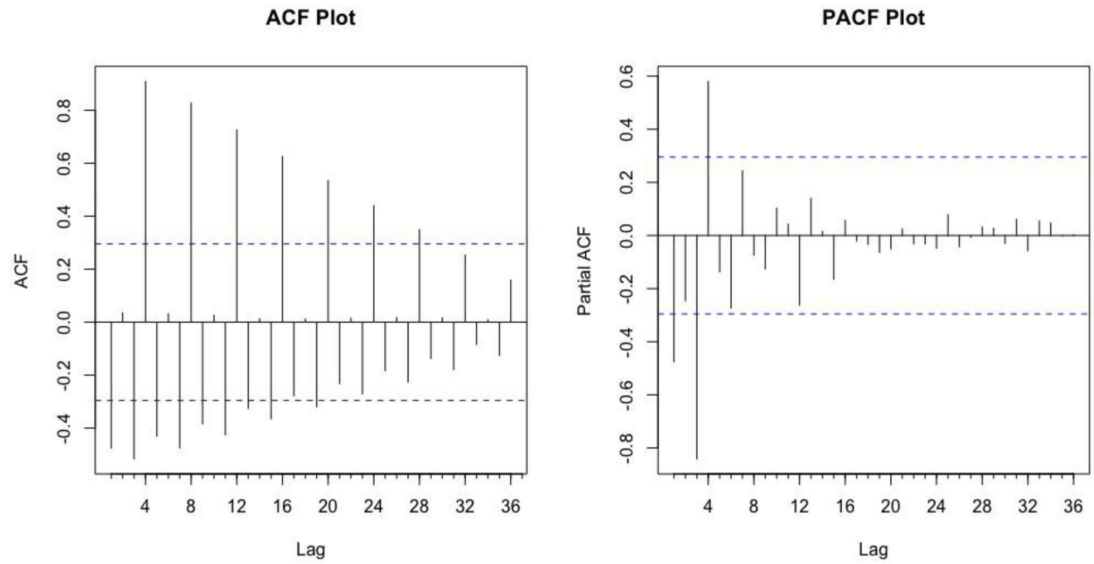
Apply the Holt-Winter's exponential smoothing (with additive trend and seasonality). First, we plot the observed data during the training set, and the forecasted revenue during the validation periods. After comparing all models, we believe model ZZZ(MNA) performs the best. It has the lowest RMSE in both the training and test sets.



ZZZ Model (MNA)	RMSE	MAPE
Training set	211.30	2.46
Test set	1174.49	20.77

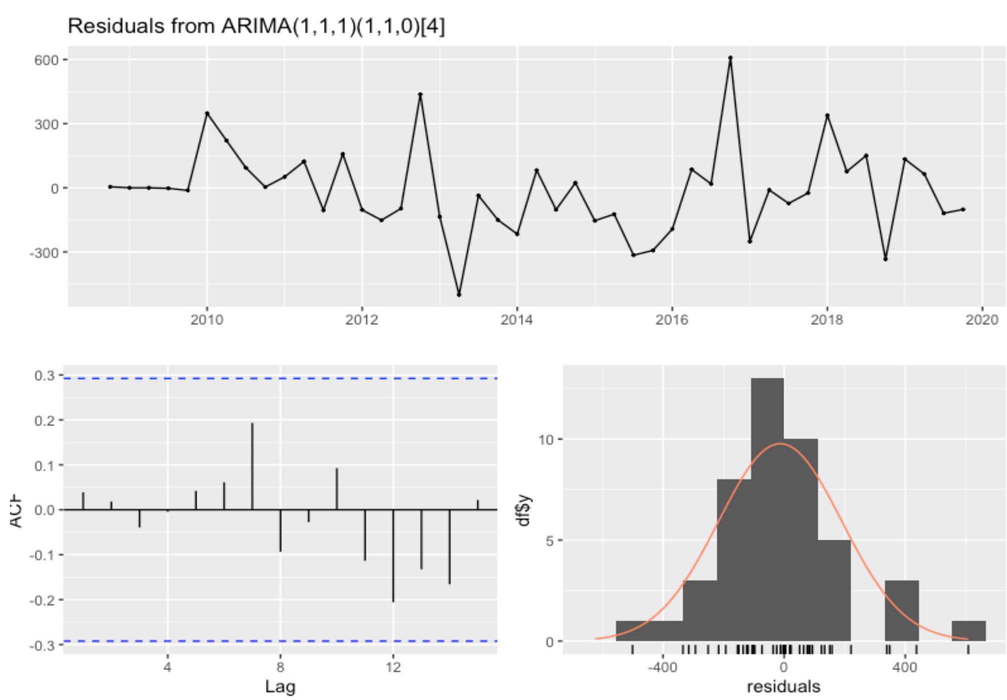
Seasonal ARIMA

ARIMA model is using the combination of Autoregression and Moving Average approaches and makes forecasts. To choose the model parameters, we observed the training dataset after detrending or deseasonality or removing both trends and seasonality. From the plots, we found that detrend provides a more obvious pattern, with tailing off ACF and cutting of PACF. What's more, because there is a seasonality pattern shown in the ACF, we decided to incorporate a seasonal component in our ARIMA model. After, we tried several combinations of parameters and decided on the best one with the lowest test MAPE.



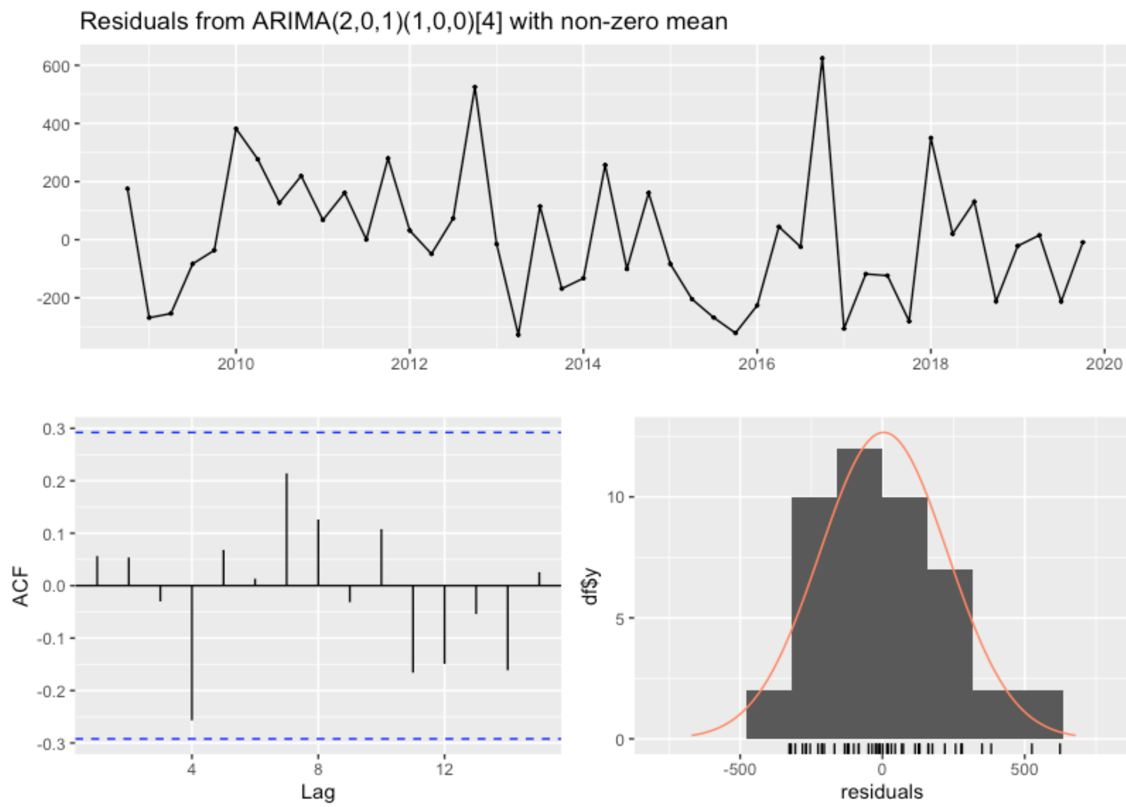
We first tried ARIMA(1,1,1)(1,1,0)[4] according to the ACF and PACF plots. Because PACF is showing significant value when lag = 1, and it is also showing a little tailing off pattern. Therefore, we chose (1,1,1) for the first part. For the seasonality part, we choose to apply both autoregressive and a constant trend into it. The model fitted successfully, and the residuals met the assumptions of the ARIMA model.

	RMSE	**MAPE**
Training set	201.89	2.23
Test set	1170.49	21.71



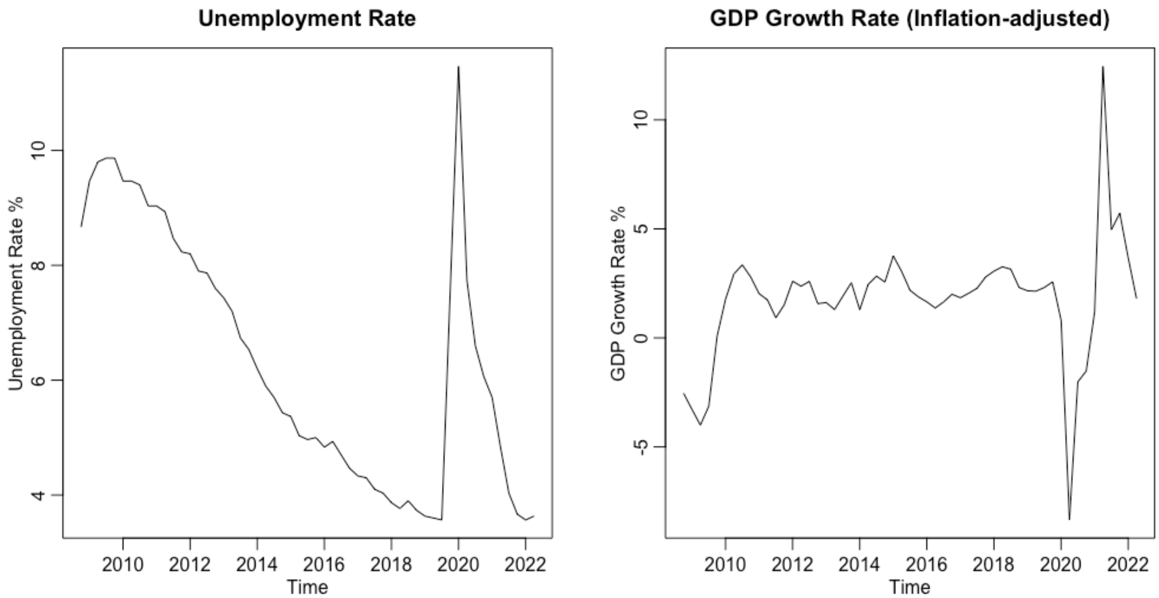
After, we also tried several combinations of parameters and found ARIMA(2,0,1)(1,0,0)[4] is performing well with the lowest MAPE value. We think this is caused by the instability of the ARIMA model itself, but since it is leading the lowest MAPE value, we decided to pick this model as the final ARIMA model we chose. The residuals also met the assumptions of the ARIMA model.

	RMSE	MAPE
Training set	222.37	2.64
Test set	1182.42	20.91



External Factors Analysis – Linear Regression & ARIMAX

Two main external factors that we think can reflect the impact of COVID-19 are GDP Growth Rate and Unemployment Rate. According to the time series plots of these two factors, we can see that there is a significant fluctuation at the beginning of the COVID-19 pandemic. Therefore, we decided to study the impact of the Unemployment Rate and GDP Growth Rate (Inflation-adjusted) on Macy's Revenue.



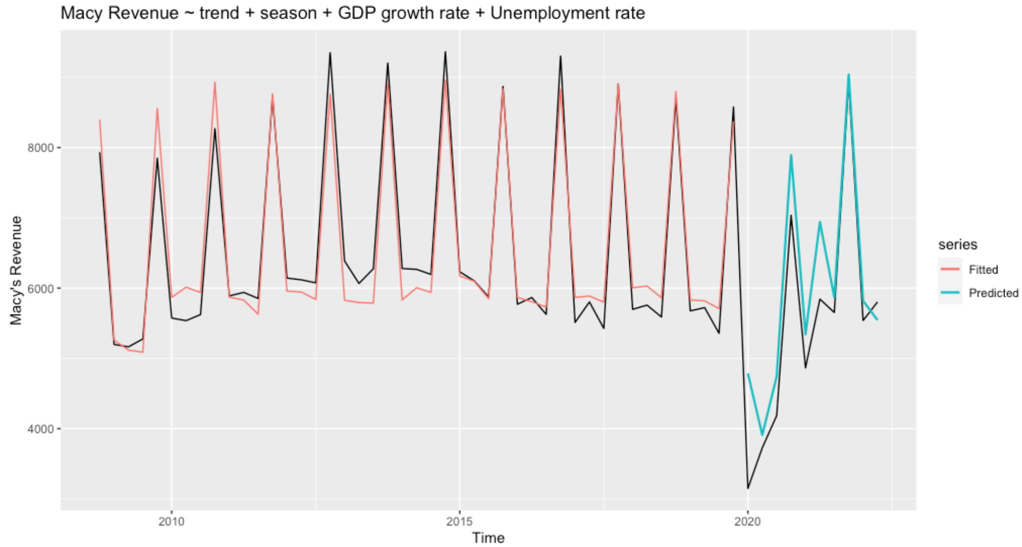
After creating a seasonally adjusted series of Macy's Revenue by rollmean, we calculated the correlation coefficient values between these two variables and Macy's Revenue and found that GDP Growth Rate (Inflation-adjusted) does have correlation with the revenue and Unemployment Rate may have a weak correlation or even have no correlation with the revenue.

COR(Unemployment Rate, Revenue)	COR(GDP Growth Rate, Revenue)
- 0.187	0.283

Then, we established several time series linear regression models which consider the impact of unemployment rate, GDP growth rate, trend, and seasonality of the revenue after splitting the data set into two parts: training set and test set.

Model	Variables	R Square	MAPE
LM1	trend, season, gdpg	0.9443	Training: 3.88 Test: 19.90
LM2	trend, season, ur	0.9275	Training: 4.81 Test: 19.75
LM3	**trend, season, gdpg, ur**	**0.9478**	**Training: 3.78 Test: 12.56**

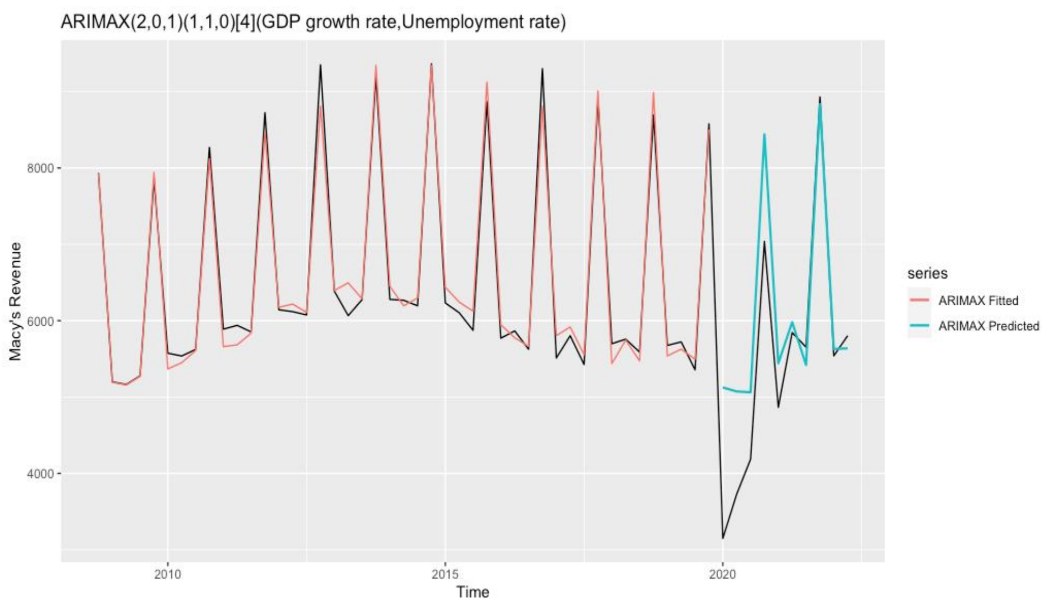
According to the MAPE and R Square, Model LM3 which considers the impact of both Unemployment Rate and GDP Growth Rate performs best. Although Unemployment rate has a low correlation coefficient value and its P value is not significant, we still consider it as an essential factor, because it does help improve the accuracy of forecasting (MAPE: from 19.90 to 12.56) and the Adjusted R Square of Model 3 (from 0.9371 to 0.9395).



	RMSE	MAPE
Training set	309.61	3.78
Test set	734.80	12.56

Then, we also tried seasonal ARIMAX models to study the impact of GDP Growth Rate and Unemployment Rate on Macy's Revenue.

ARIMAX(2,0,1)(1,1,0)[4] Model	External Variables	MAPE
AXM1	gdpg	Training: 2.19 Test: 18.52
AXM2	ur	Training: 2.19 Test: 18.65
AXM3	**gdpg, ur**	**Training: 2.12 Test: 16.38**



	RMSE	MAPE
Training set	190.93	2.12
Test set	944.05	16.38

Summary

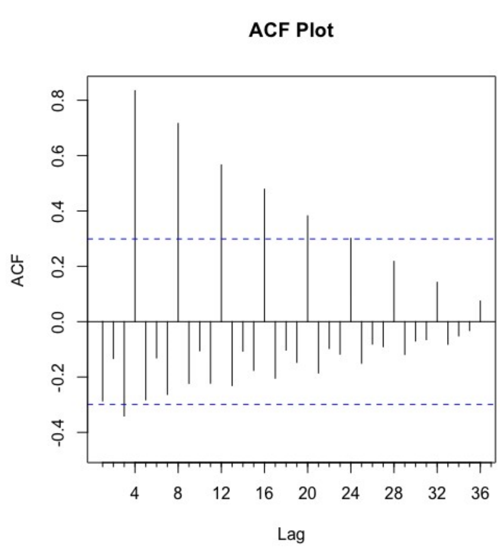
Based on the performances of above models, we can find the best model to forecast Macy's Revenue is the time series linear regression model LM3 which considers the impact of both Unemployment Rate and GDP Growth Rate. The MAPE of the test set of LM3 is 12.56 which is the lowest one. In addition, we can find LM3 does a really good job in predicting Macy's Revenue during the early period of COVID-19 and this reflects that the two external factors (GDP Growth Rate and Unemployment Rate) do have a strong impact on Macy's Revenue especially in the early period of COVID-19. In other words, COVID-19 does have a strong negative impact on Macy's Revenue which cannot be ignored when predicting Macy's future revenue.

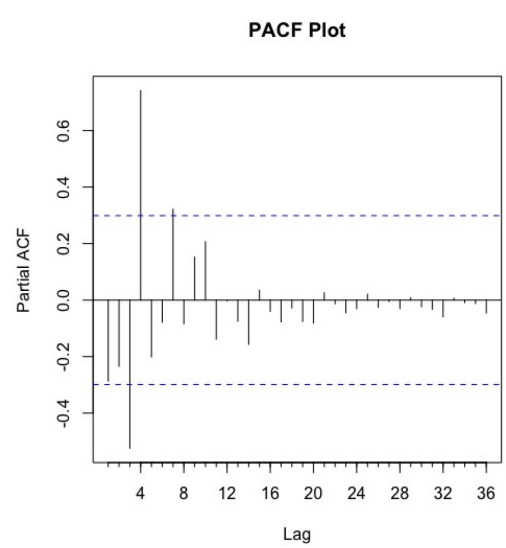
Competitor Analysis - Amazon

For comparison, we chose Amazon to investigate if COVID-19 also impacted its revenue severely and if we can still make a good forecast on its revenue under the COVID-19 pandemic. If we can make a good prediction, it means Amazon's revenue was not affected by the pandemic and Macy's can learn from Amazon's operation. What's more, we also want to investigate Amazon's revenue with external factors including GDP Growth Rate (Inflation-adjusted) and the unemployment rate. The model we chose for Amazon is the seasonal ARIMA model, linear regression models and seasonal ARIMAX models.

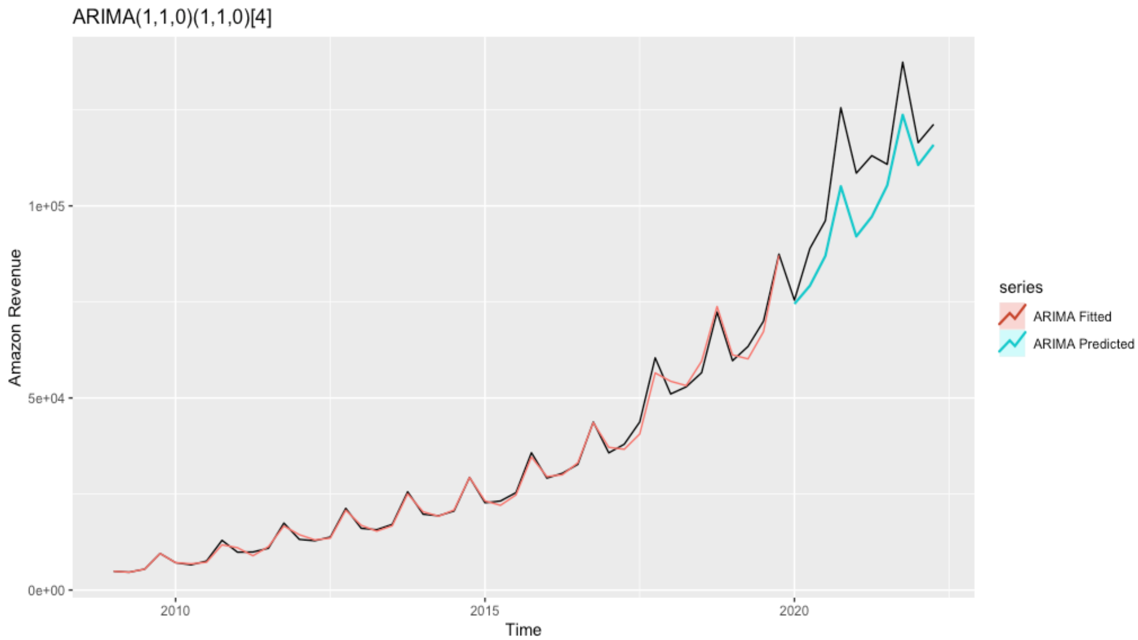
Seasonal ARIMA

Similar to the procedure we performed on Macy's revenue dataset, we detrended the dataset and observed the ACF and PACF of Amazon's revenue. It also has a tailing-off pattern for ACF and a cutting-off pattern for PACF.

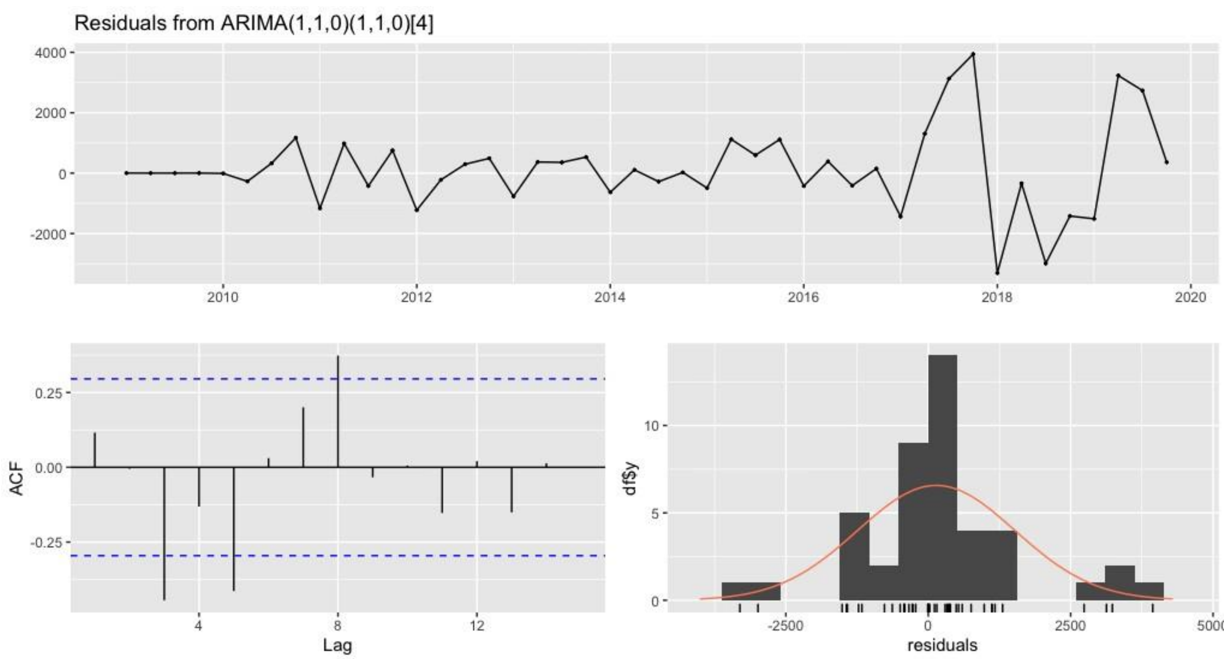




According to these two plots, we decided to incorporate AR in the model. After trying several combinations of parameters, we chose ARIMA(1,1,0)(1,1,0)[4] as our best model for Amazon. Although the residuals still have some correlation with their value when lag equals 3 or 5, we do not see a seasonality effect that we need to remove.



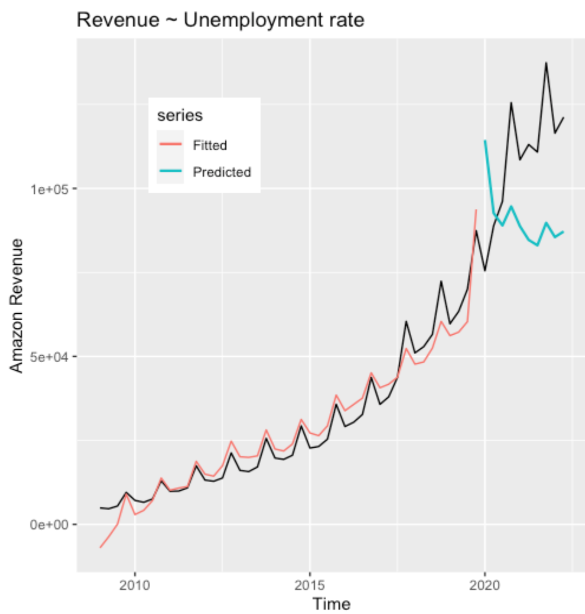
	RMSE	MAPE
Training set	1374.98	3.27
Test set	11842.37	9.16

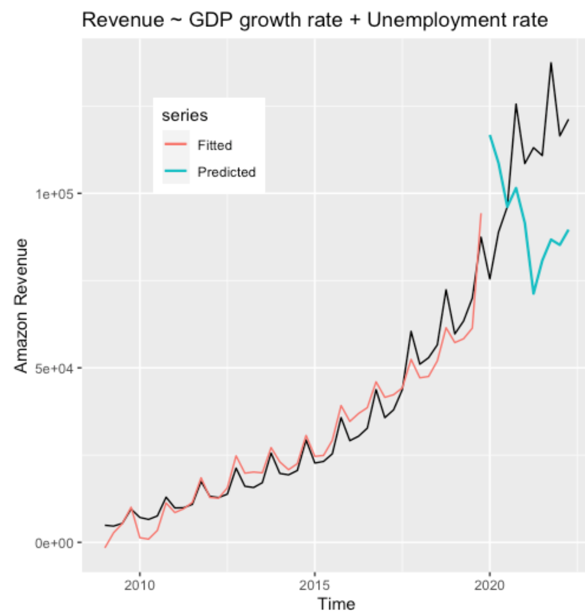


External Factors Analysis – Linear Regression & ARIMAX

We also analyzed the impact of GDP Growth Rate and Unemployment Rate on Amazon's Revenue and established several linear regression models and the seasonal ARIMAX models.

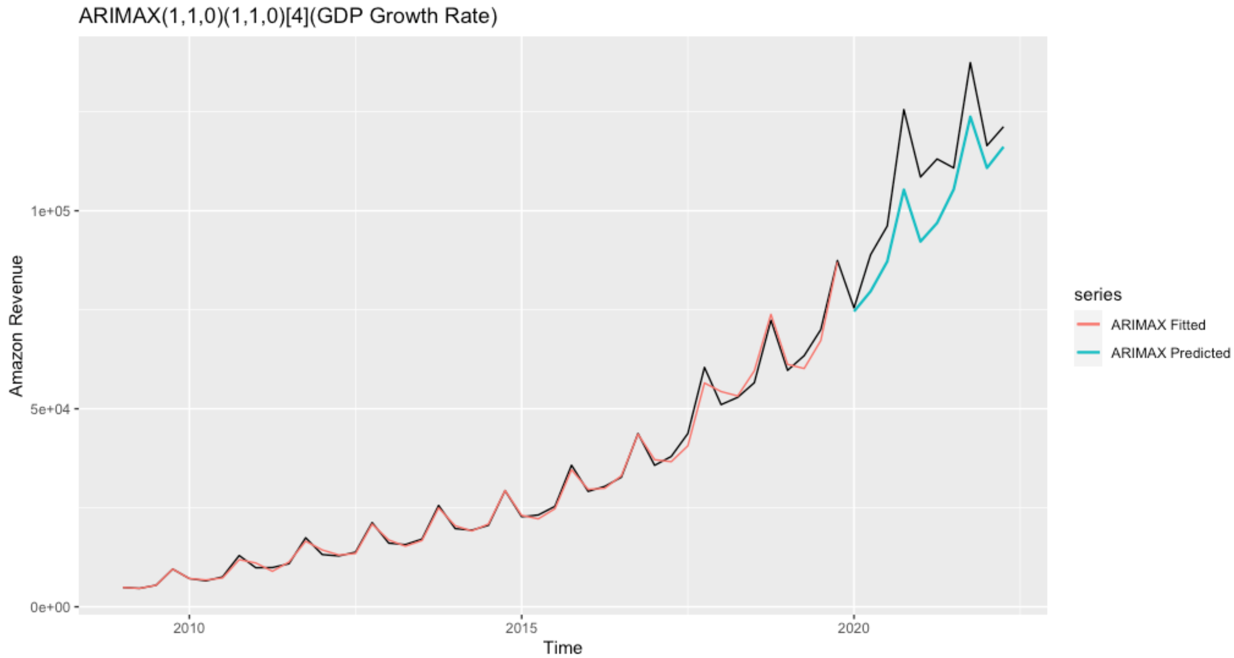
Model	Variables	R Square	MAPE
ALM1	trend, season, gdpg	0.9075	Training: 28.62 Test: 33.15
ALM2	trend, season, ur	0.9485	Training: 24.59 Test: 24.56
ALM3	trend, season, gdpg, ur	0.9575	Training: 18.39 Test: 26.56





According to the performance of linear regression models, we found these two external factors will mislead the linear prediction of Amazon's Revenue especially in the period of COVID-19.

ARIMAX(1,1,0)(1,1,0)[4] Model	External Variables	MAPE
AAXM1	**gdpg**	**Training: 3.25 Test: 9.02**
AAXM2	ur	Training: 3.25 Test: 10.30
AAXM3	gdpg, ur	Training: 3.2 Test: 10.51



According to the performance of ARIMAX models, we found the best ARIMAX model is AAXM1 which performed slightly better than the common ARIMA model and other ARIMAX models all performed worse. This means GDP Growth Rate and Unemployment Rate may have a weak impact or even no impact on Amazon's Revenue.

Summary

Based on the performances of each model of predicting Amazon's Revenue, we can find the best model is AAXM1 which is a seasonal ARIMAX model which considers the impact of GDP Growth Rate. However, the performance of AAXM1 is slightly better than the common seasonal ARIMA model and we can find that the predicted lines of these two models are almost the same. This means that we can nearly ignore the impact of external factors when predicting Amazon's Revenue. In other words, Amazon's revenue is barely impacted by COVID-19. Therefore, we can obtain business insights and provide recommendations by comparing the difference between Macy's and Amazon.

Conclusion & Recommendations

According to the analysis above, we mainly studied the impact of COVID-19 on Macy's revenue and Amazon's revenue via focusing on the revenue itself and the impact of inflation-adjusted GDP Growth Rate and Unemployment Rate. We think COVID-19 does have strong negative impact on Macy's revenue while Amazon's revenue is barely impacted by COVID-19. Therefore, we can then provide recommendations based on the analysis.

As our revenue source mainly comes from sales, our recommendation for Macy's will mostly focus on sales. According to our revenue analysis, Macy's competitor, Amazon, didn't lose much revenue and had a constantly increasing trend during the pandemic. It calls us to think more strategically about how Macy’s should make improvements on its operating strategy to meet any potential crisis in the future.

As our analysis indicates, Macy’s should and is now transferring to more digital led than focusing on off-line shopping. For the fourth quarter of 2021, Macy’s online revenue occupies 58% of all revenue, so it should pay more attention to online retail now as the pandemic has changed people’s shopping habits more from off-line to online.

Our team would recommend Macy’s to improve the online shopping experience for customers to shop online in a simple way just like they did for offline shopping. It can achieve this by adjusting the website interface to shorten the user's shopping path, simplifying the price to make it more clear for customers, offering a more personalized shopping experience for customers, and putting more budget on online promotion to attract more customers to encourage sales online.

Appendix

DSO522 Final Project RMD

Group 14

Import packages and data

```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(ggplot2)
library(astsa)
```

```
##
## Attaching package: 'astsa'

## The following object is masked from 'package:forecast':
##
##     gas
```

```
library(patchwork)

macy <- read.csv("~/Downloads/MacyRevenue.csv")
ama <- read.csv("~/Downloads/AmazonRevenue.csv")
ur <- read.csv("~/Downloads/QuarterlyUR.csv")
gdpg <- read.csv("~/Downloads/GDPGrowthRate.csv")

macy.ts <- ts(macy$Revenue, start = c(2008,4), end = c(2022,2), frequency = 4)
ama.ts <- ts(ama$Revenue, start = c(2009,1), end = c(2022,2), frequency = 4)
ur.ts <- ts(ur$UR, start = c(2008,4), end = c(2022,2), frequency = 4)
gdpg.ts <- ts(gdpg$GDP.Growth.Rate, start = c(2008,4), end = c(2022,2), frequency = 4)
```

Split data set

```
n <- length(macy.ts)
stepsAhead <- 10
nTrain <- n-stepsAhead
```

```r
train.m.ts <- window(macy.ts, end = c(2019,4), frequency = 4)
valid.m.ts <- window(macy.ts, start = c(2020,1), frequency = 4)
```

Navie Forecast

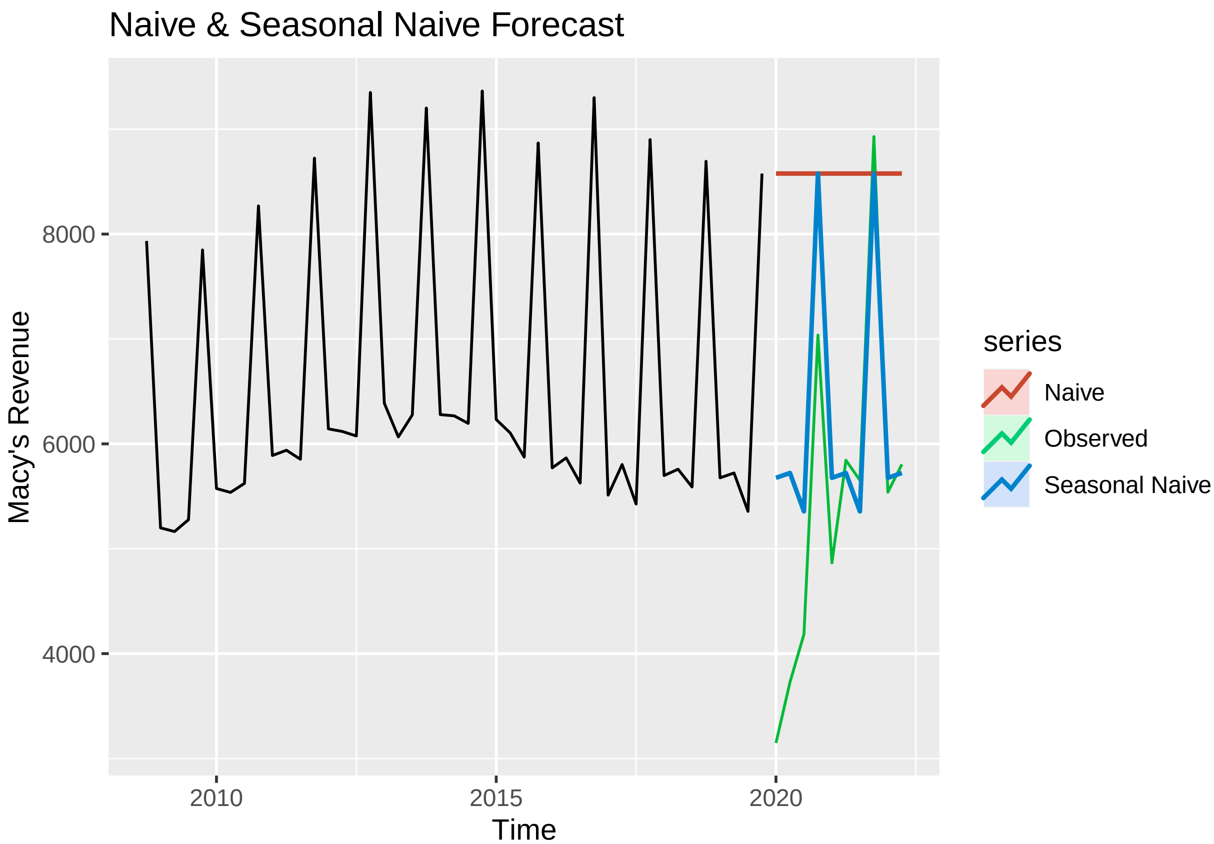
```r
naive <- naive(train.m.ts,h = 10)
snaive <- snaive(train.m.ts,h = 10)
accuracy(naive, valid.m.ts)
```

```
##                       ME     RMSE      MAE        MPE     MAPE      MASE
## Training set    14.61364 2142.210 1570.932  -4.278627 22.47360  6.560885
## Test set     -3103.80000 3485.676 3174.400 -70.288338 71.07893 13.257655
##                    ACF1 Theil's U
## Training set -0.4751925        NA
## Test set      0.1611863  2.124442
```

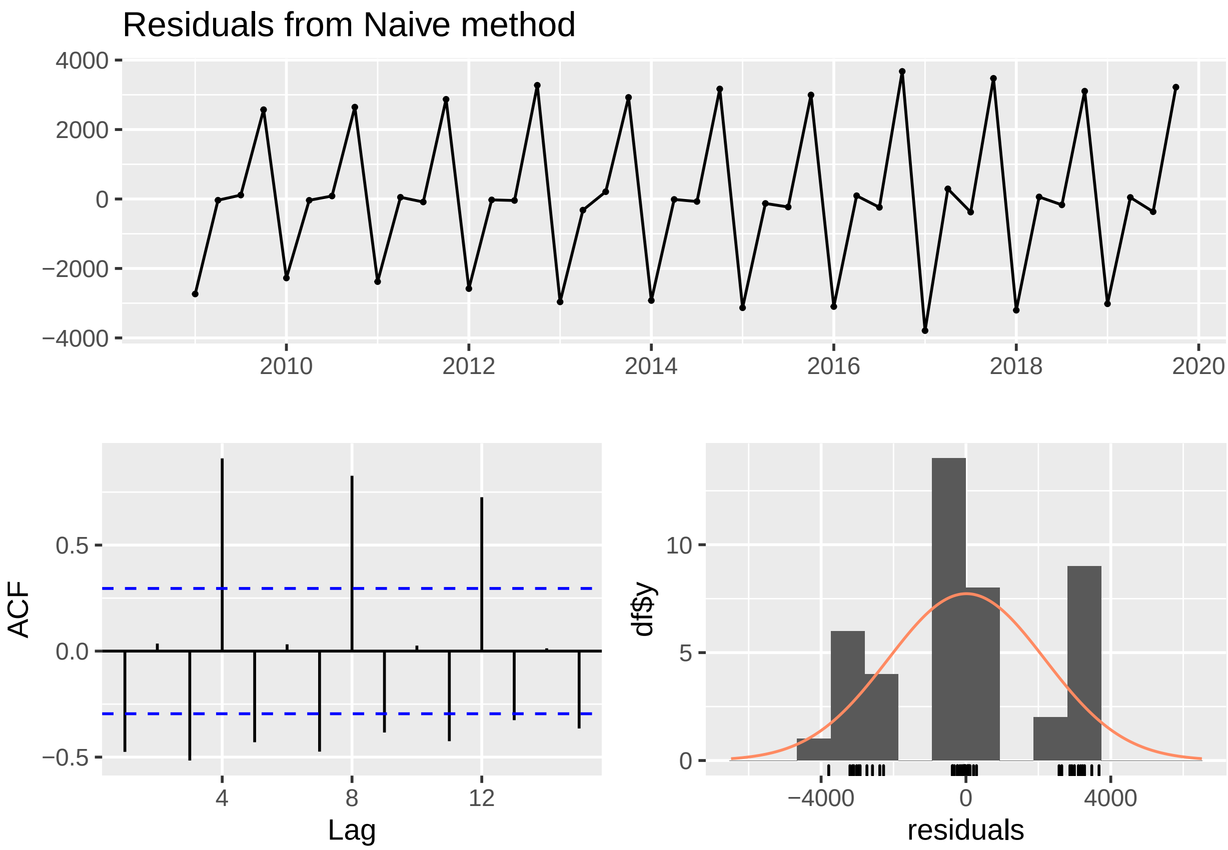
```r
accuracy(snaive, valid.m.ts)
```

```
##                     ME      RMSE     MAE         MPE      MAPE     MASE
## Training set   42.85366  279.0601 239.439   0.6047739  3.598779 1.000000
## Test set     -732.80000 1225.7474 903.600 -19.0134690 21.554918 3.773821
##                   ACF1 Theil's U
## Training set 0.5040912        NA
## Test set     0.6525172 0.7613611
```

```r
autoplot(train.m.ts, ylab="Macy's Revenue", main = "Naive & Seasonal Naive Forecast")+
  autolayer(valid.m.ts, series = "Observed")+
  autolayer(naive, series = "Naive", PI=F,lwd = 0.8)+
  autolayer(snaive, series = "Seasonal Naive",PI=F,lwd = 0.8)
```

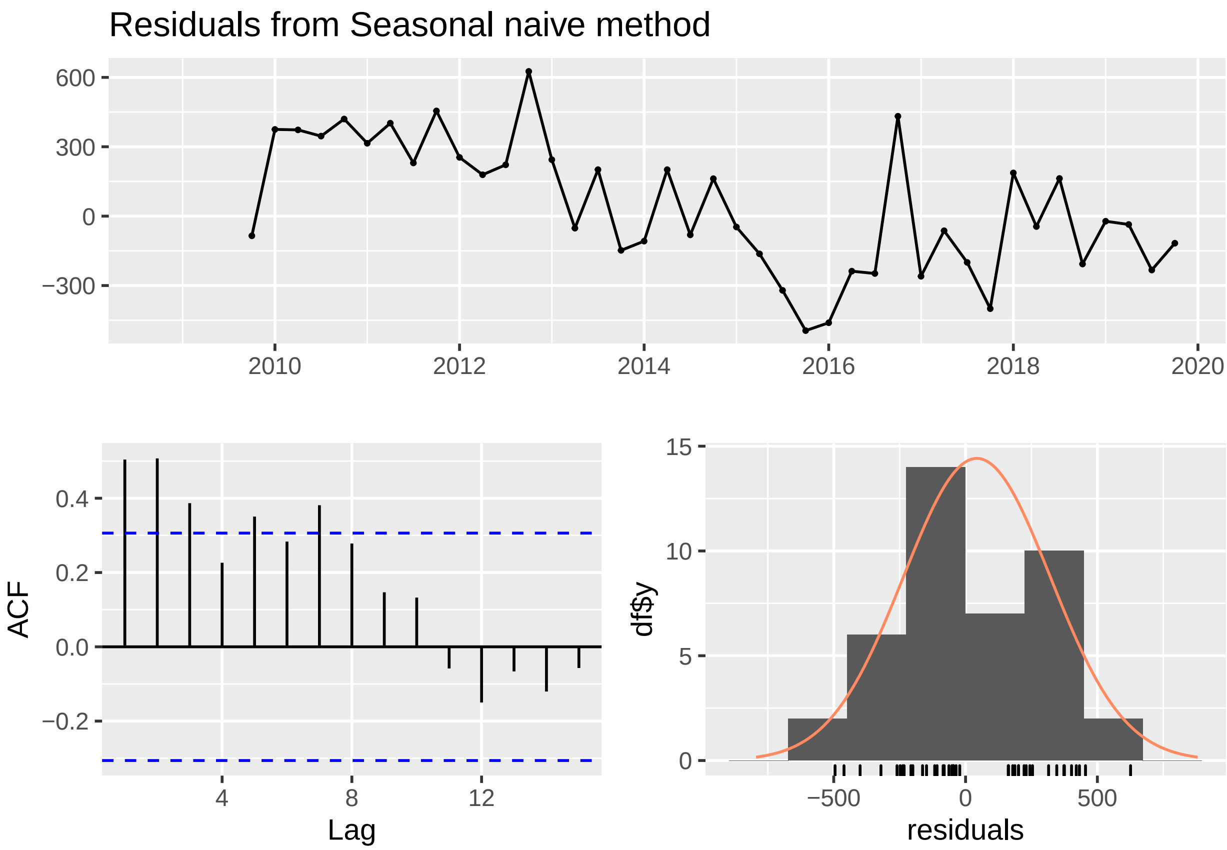



```
checkresiduals(naive)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Naive method
## Q* = 126.02, df = 8, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 8
```

```
checkresiduals(snaive)
```

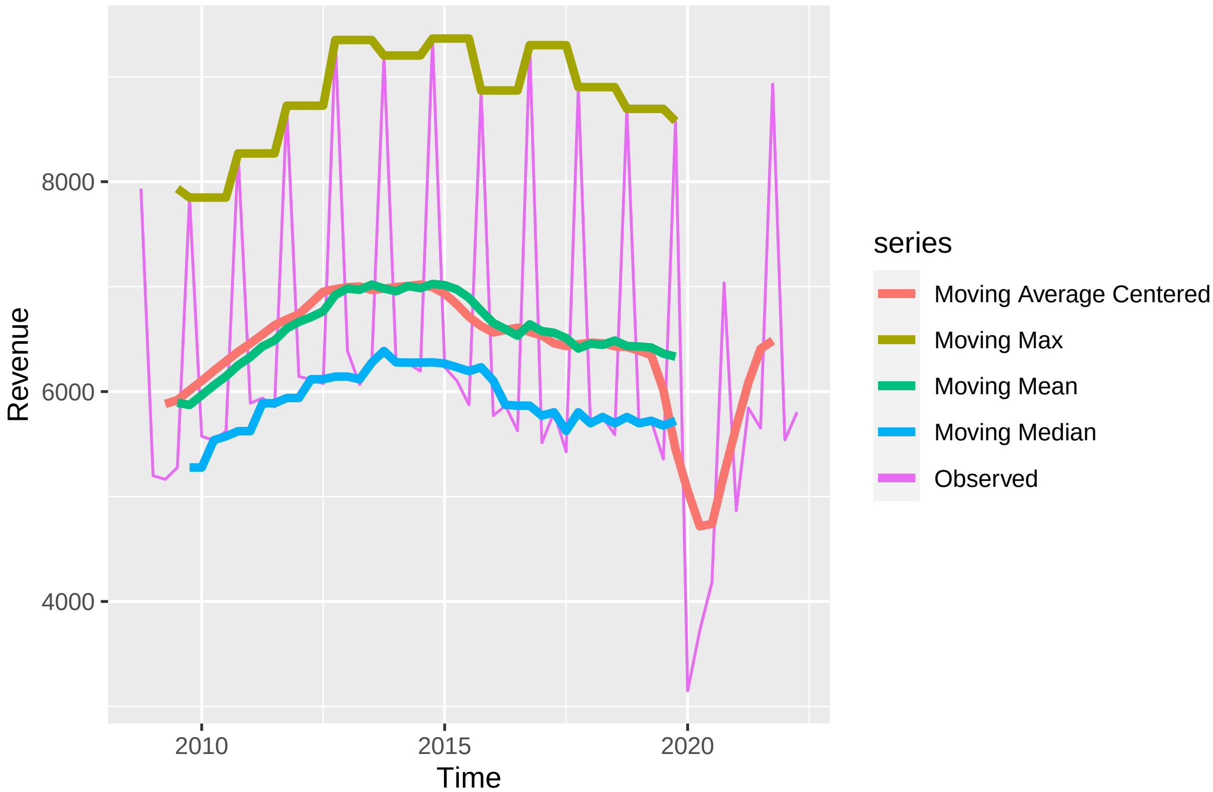



```
##
##  Ljung-Box test
##
## data:  Residuals from Seasonal naive method
## Q* = 53.945, df = 8, p-value = 7.074e-09
##
## Model df: 0.   Total lags used: 8
```

Moving average smoothing: centered and trailing

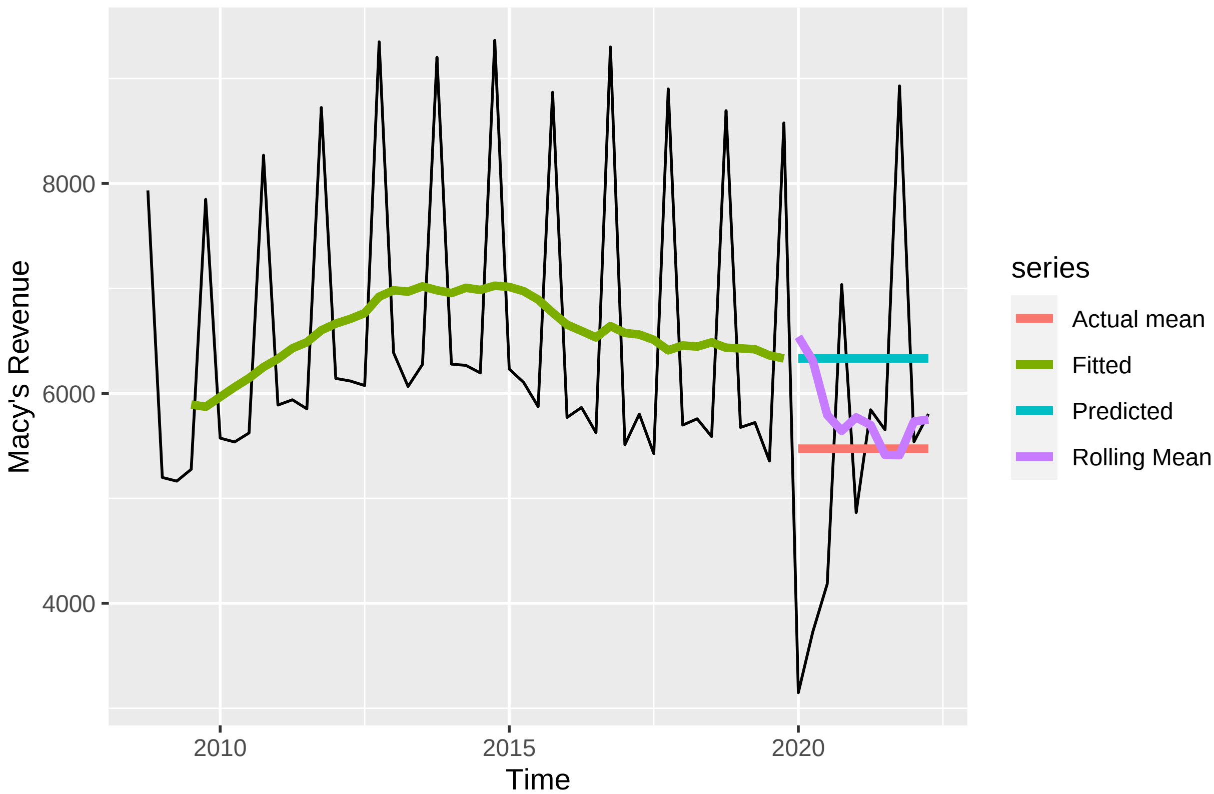
```
# center, roll mean
ma.centered <- ma(macy.ts, order = 4, centre = T) # centered
w <- 4

ma.trailing <- rollmean(train.m.ts, k = w, align = "right") # mean
ma.trailing.max<- rollmax(train.m.ts, k = w, align = "right") # max
ma.trailing.median <- rollmedian(train.m.ts, k = 5, align = "right") # median
ma.trailing.sum<- rollsum(train.m.ts, k = w, aligh="right") # sum

autoplot(macy.ts, ylab="Revenue", series = "Observed")+
  autolayer(ma.centered, series = "Moving Average Centered", lwd=1.5)+
  autolayer(ma.trailing, series = "Moving Mean", lwd=1.5)+
  autolayer(ma.trailing.median, series = "Moving Median", lwd=1.5)+
  autolayer(ma.trailing.max, series = "Moving Max", lwd=1.5)
```

```
## Warning: Removed 4 row(s) containing missing values (geom_path).
```



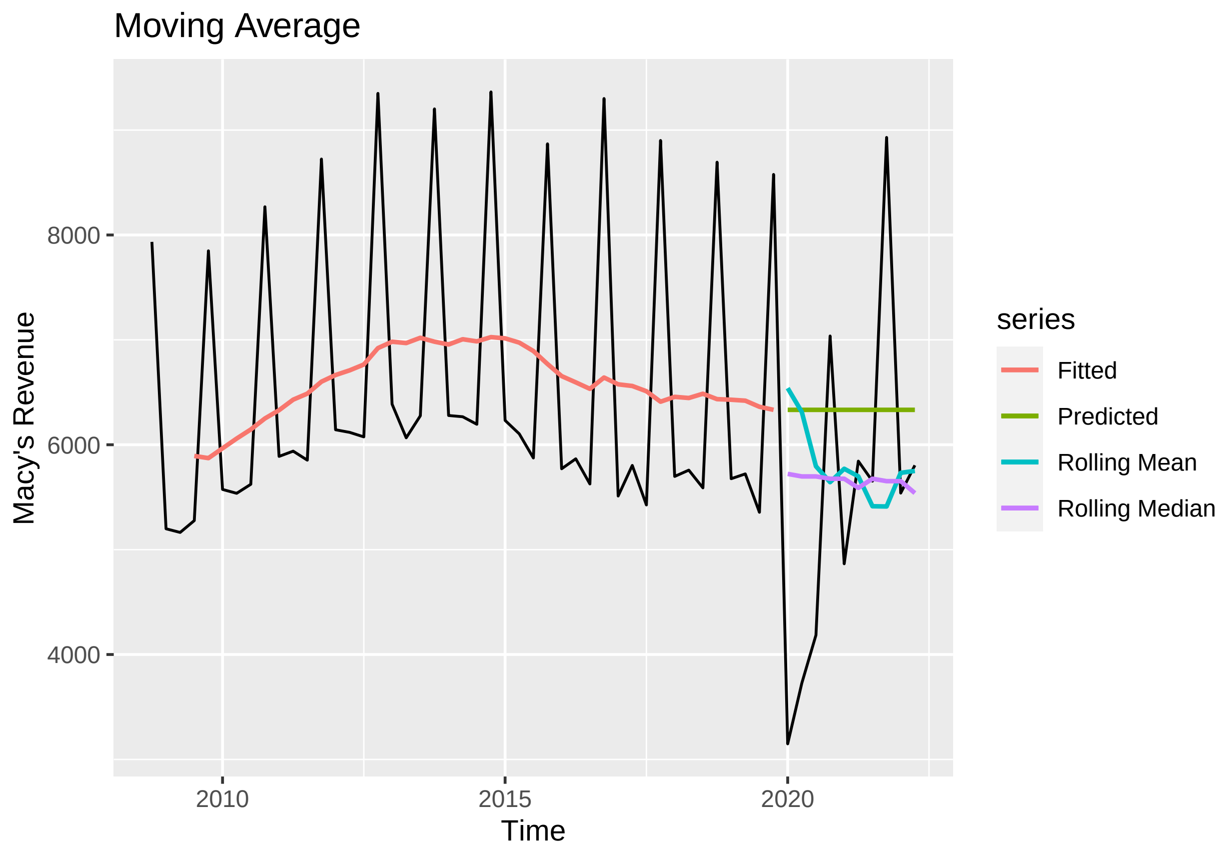

```
# One Step Ahead Rolling Forecast(rollmean)

last.ma <- tail(ma.trailing, 1)

ma.trailing.pred <- ts(rep(last.ma, stepsAhead),
                       start = c(2020,1),
                       end = c(2022,2),frequency =4)
# actual
valid.mean = ts(rep(mean(valid.m.ts), stepsAhead),
                start = c(2020, 1),
                end = c(2022, 2),
                frequency = 4)

# looping
ma.trailing.pred.rolling.mean <- rep(NA, stepsAhead)

start.year = 2008
w <- 10
for(i in 1:stepsAhead){
  #i = 1
  nTrain <- n - stepsAhead + (i-1)
  train.ts <- window(macy.ts,start=c(2008,4),end=c(start.year, nTrain+3))
  ma.trailing.mean = rollmean(train.ts, k = w, align = "right")
  # our long term forecast
  last.ma.mean <- tail(ma.trailing.mean, 1)
  ma.trailing.pred.rolling.mean[i] <- last.ma.mean
}

ma.roll.ts.mean <- ts(ma.trailing.pred.rolling.mean,
```

```
                 start = c(2020, 1), frequency = 4)

autoplot(macy.ts, ylab="Macy's Revenue")+
  autolayer(ma.trailing, series = "Fitted", lwd = 1.5)+
  autolayer(ma.trailing.pred, series = "Predicted", lwd=1.5)+
  autolayer(valid.mean, series="Actual mean", lwd=1.5)+
  autolayer(ma.roll.ts.mean, series ="Rolling Mean",lwd=1.5)
```




```
# One Step Ahead Rolling Forecast(Rollmedian)

nValid = 10
ma.trailing.pred.rolling.median <- rep(NA, stepsAhead)

for(i in 1:nValid){
  nTrain <- n - nValid + (i-1)
  train.ts<-window(macy.ts,start=c(2008,4),end=c(start.year, nTrain+3))
  ma.trailing.roll.median <- rollmedian(train.ts, k = 11, align = "right")
  # k=9/11 is same accuracy
  last.ma.median = tail(ma.trailing.roll.median, 1)
  ma.trailing.pred.rolling.median[i]=last.ma.median
}

ma.roll.median.ts <- ts(ma.trailing.pred.rolling.median,
                 start = c(2020,1), frequency = 4)

autoplot(macy.ts, ylab="Macy's Revenue", main = "Moving Average")+
  autolayer(ma.trailing, series = "Fitted", lwd = 0.8)+
  autolayer(ma.trailing.pred, series = "Predicted", lwd=0.8)+
```

```
autolayer(ma.roll.ts.mean, series ="Rolling Mean",lwd=0.8)+
autolayer(ma.roll.median.ts, series ="Rolling Median",lwd=0.8)
```




```
# Accuracy

accuracy(ma.roll.ts.mean,valid.m.ts)
```

```
##                ME     RMSE     MAE       MPE     MAPE      ACF1 Theil's U
## Test set -333.49 1898.384 1403.59 -17.08191 30.45191 0.2681699  1.103292
```

```
accuracy(ma.roll.median.ts,valid.m.ts)
```

```
##               ME    RMSE    MAE       MPE    MAPE      ACF1 Theil's U
## Test set -184.8 1618.55 1216.4 -12.49722 25.4905 0.1644417 0.9476044
```

Exponential Smoothing

```
aan <- ets(train.m.ts, model = 'AAN')
ann <- ets(train.m.ts, model = 'ANN')
aaa <- ets(train.m.ts, model = 'AAA')
zzz <- ets(train.m.ts, model='ZZZ') # mna <- ets(train.m.ts, model = 'MNA')

aan.predict <- forecast.ets(aan, h=10, level=0)
ann.predict <- forecast.ets(ann, h=10, level=0)
aaa.predict <- forecast.ets(aaa, h=10, level=0)
zzz.predict <- forecast.ets(zzz, h=10, level=0)
```

```
accuracy(aan.predict, valid.m.ts)
```

```
##                       ME     RMSE      MAE        MPE     MAPE     MASE
## Training set    40.22932 1348.597 1135.079  -2.974025 16.31517 4.740575
## Test set     -1272.53320 2031.146 1767.215 -33.907995 39.62435 7.380646
##                    ACF1 Theil's U
## Training set -0.2742338        NA
## Test set      0.1588736  1.235405
```

```
accuracy(ann.predict, valid.m.ts)
```

```
##                     ME     RMSE      MAE        MPE     MAPE     MASE
## Training set  214.2984 1380.315 1059.777  -0.311048 14.75147 4.426082
## Test set     -993.8566 1871.928 1600.434 -28.397378 35.53334 6.684098
##                    ACF1 Theil's U
## Training set -0.2604176        NA
## Test set      0.1611863  1.132855
```

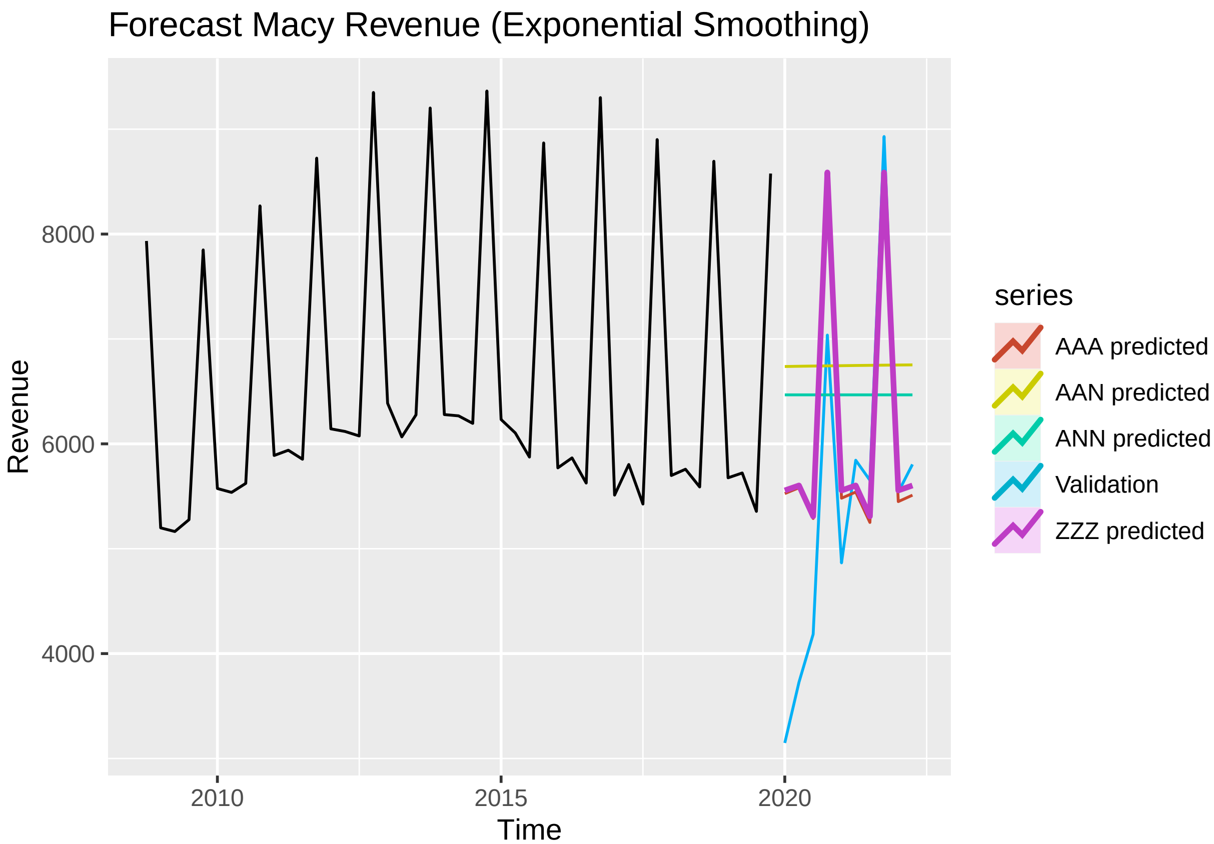
```
accuracy(aaa.predict, valid.m.ts)
```

```
##                     ME      RMSE      MAE         MPE      MAPE      MASE
## Training set  -26.74688  205.0471 156.6993  -0.5588188  2.394355 0.6544436
## Test set     -597.92401 1162.6762 896.8352 -16.2412617 20.947010 3.7455681
##                     ACF1 Theil's U
## Training set -0.07146444        NA
## Test set      0.65834604 0.7224527
```

```
accuracy(zzz.predict, valid.m.ts) # Best "MNA"
```

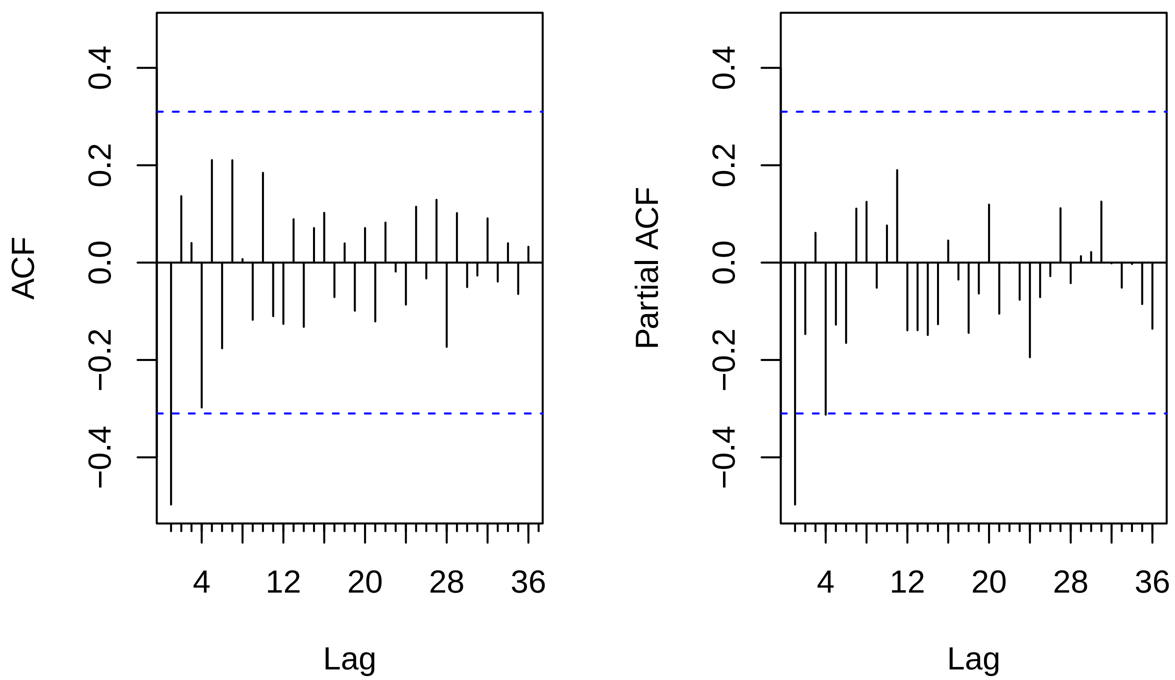
```
##                     ME      RMSE      MAE         MPE      MAPE      MASE
## Training set   12.89599  211.3024 163.3312   0.1347046  2.462124 0.6821409
## Test set     -653.63257 1174.4884 879.3759 -17.2684335 20.770735 3.6726509
##                     ACF1 Theil's U
## Training set -0.05591542        NA
## Test set      0.65177689 0.7301469
```

```
autoplot(train.m.ts,  ylab = 'Revenue',xlab = 'Time', main = 'Forecast Macy Revenue (Exponential Smoothi
  autolayer(valid.m.ts, series = 'Validation')+
  autolayer(aan.predict, series='AAN predicted')+
  autolayer(ann.predict, series='ANN predicted')+
  autolayer(aaa.predict, series='AAA predicted')+
  autolayer(zzz.predict, series='ZZZ predicted',lwd=1)
```

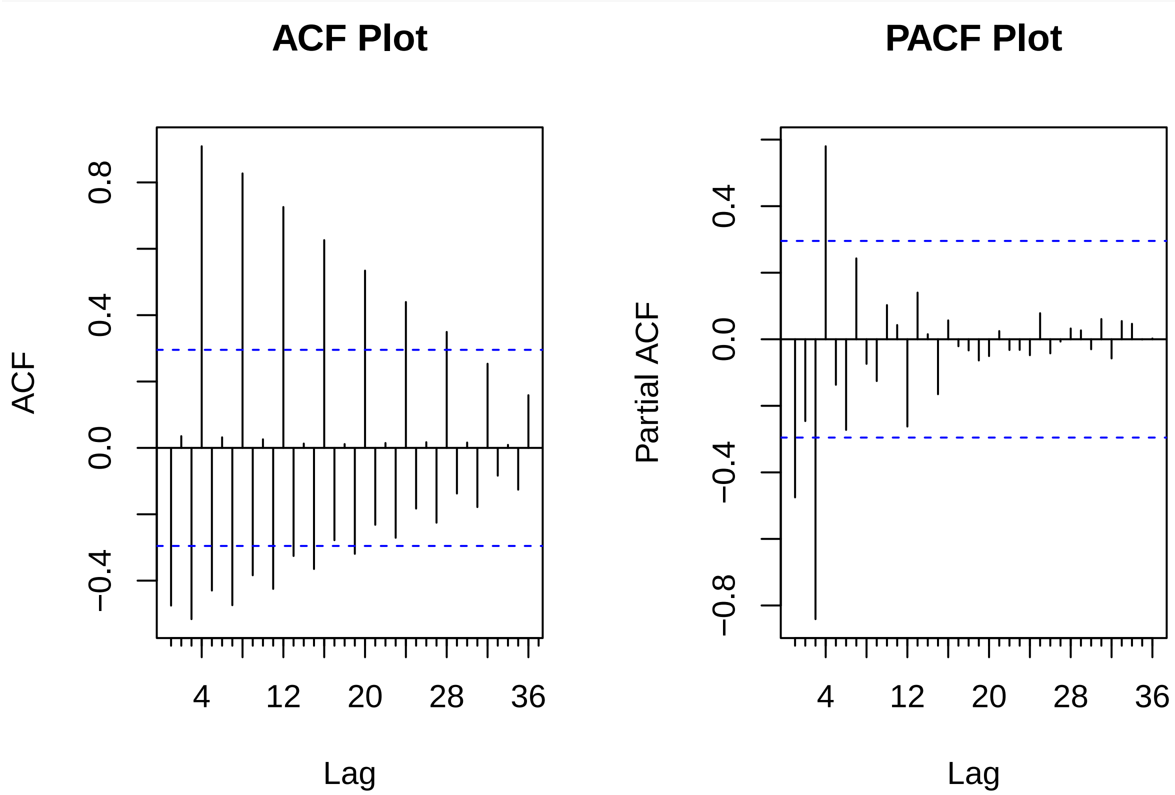


Arima

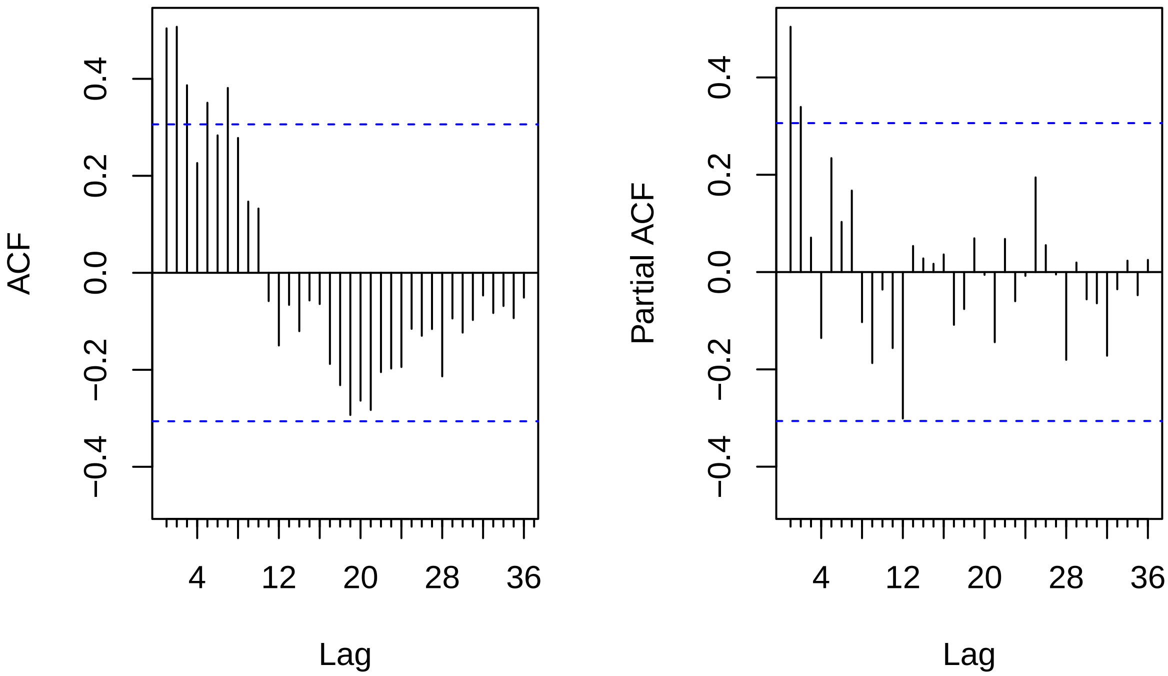
```
# Try Detrend and Deseasonality

#(1) Detrend and Deseasonality
ddtrain<-diff(diff(train.m.ts,1),4)
par(mfrow=c(1,2))
Acf(ddtrain,36,main="")
Pacf(ddtrain,36, main="")
```




```
par(mfrow=c(1,1))

#(2) Detrend
dttrain<-diff(train.m.ts,1)
par(mfrow=c(1,2))
Acf(dttrain,36,main="ACF Plot")
Pacf(dttrain,36, main="PACF Plot")
```



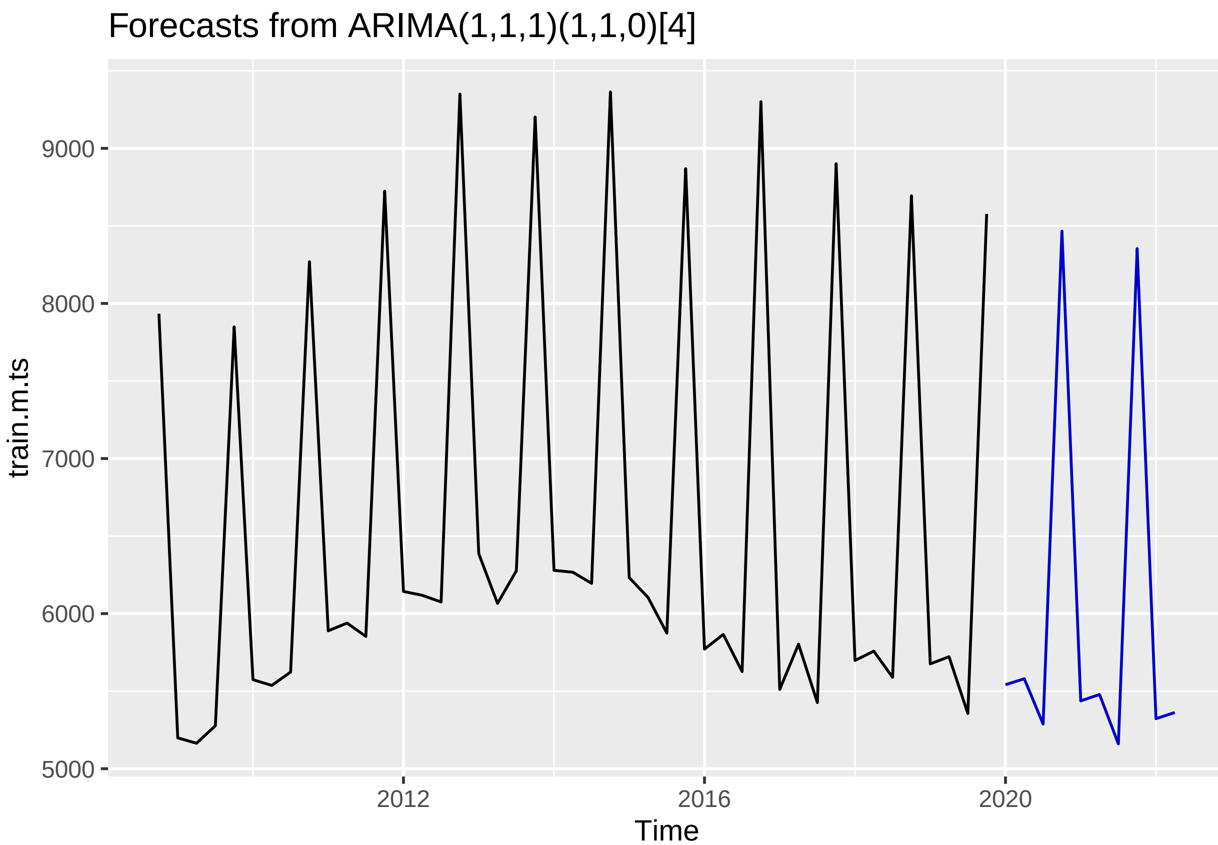
```r
par(mfrow=c(1,1))

#(3) Deseasonality
dstrain<-diff(train.m.ts,4)
par(mfrow=c(1,2))
Acf(dstrain,36,main="")
Pacf(dstrain,36, main="")
```



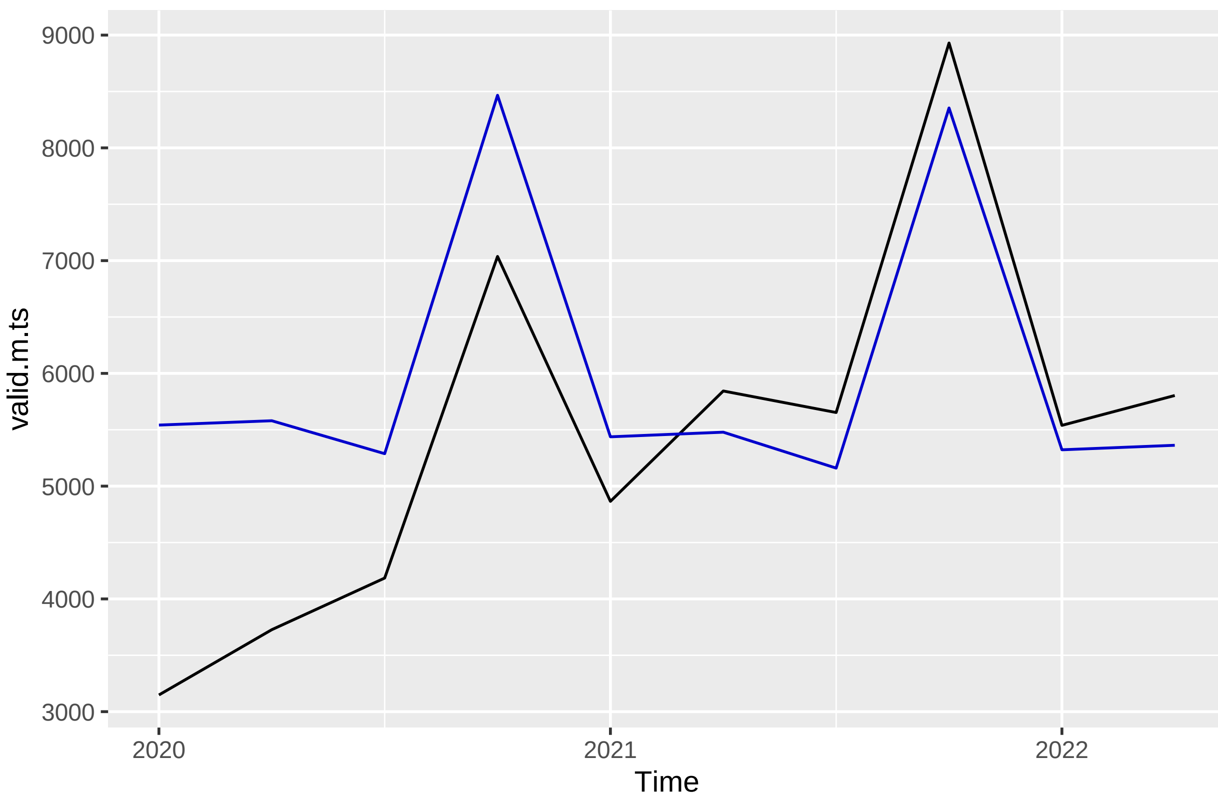

```r
par(mfrow=c(1,1))

# ACF tailing off and PACF cutting off. Therefore, choose AR model. Look at PACF, choose 1.

# model 1
am1 <- Arima(train.m.ts, order = c(1,1,1), seasonal = list(order = c(1,1,0), period = 4))
summary(am1)
```

```
## Series: train.m.ts
## ARIMA(1,1,1)(1,1,0)[4]
##
## Coefficients:
##           ar1      ma1     sar1
##       -0.2597  -0.4413  -0.3591
## s.e.   0.2562   0.2409   0.1488
##
## sigma^2 = 49573:  log likelihood = -271.96
## AIC=551.92   AICc=553.06   BIC=558.68
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -12.98012 201.8919 147.2646 -0.2398685 2.229155 0.6150399
##                    ACF1
## Training set 0.03882234
```

```r
am1.predict <- forecast(am1, h = 10, level = 0)
autoplot(am1.predict)
```



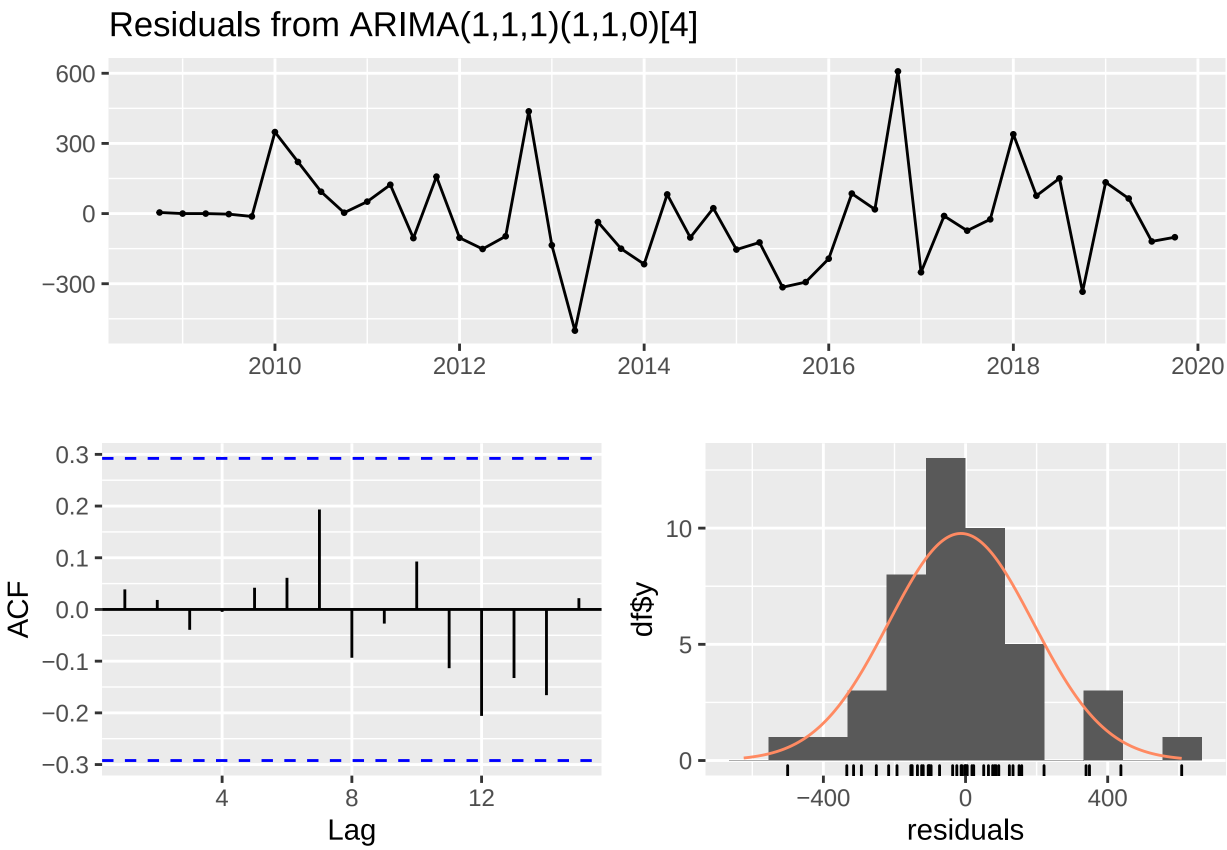

```r
autoplot(valid.m.ts)+
  autolayer(am1.predict)
```




```
accuracy(am1.predict, valid.m.ts)
```

```
##                      ME      RMSE      MAE         MPE      MAPE      MASE
## Training set  -12.98012  201.8919 147.2646  -0.2398685  2.229155 0.6150399
## Test set     -525.86151 1170.4972 944.4166 -15.1242937 21.713115 3.9442887
##                    ACF1 Theil's U
## Training set 0.03882234        NA
## Test set     0.67065449 0.7201248
```

```
checkresiduals(am1)
```

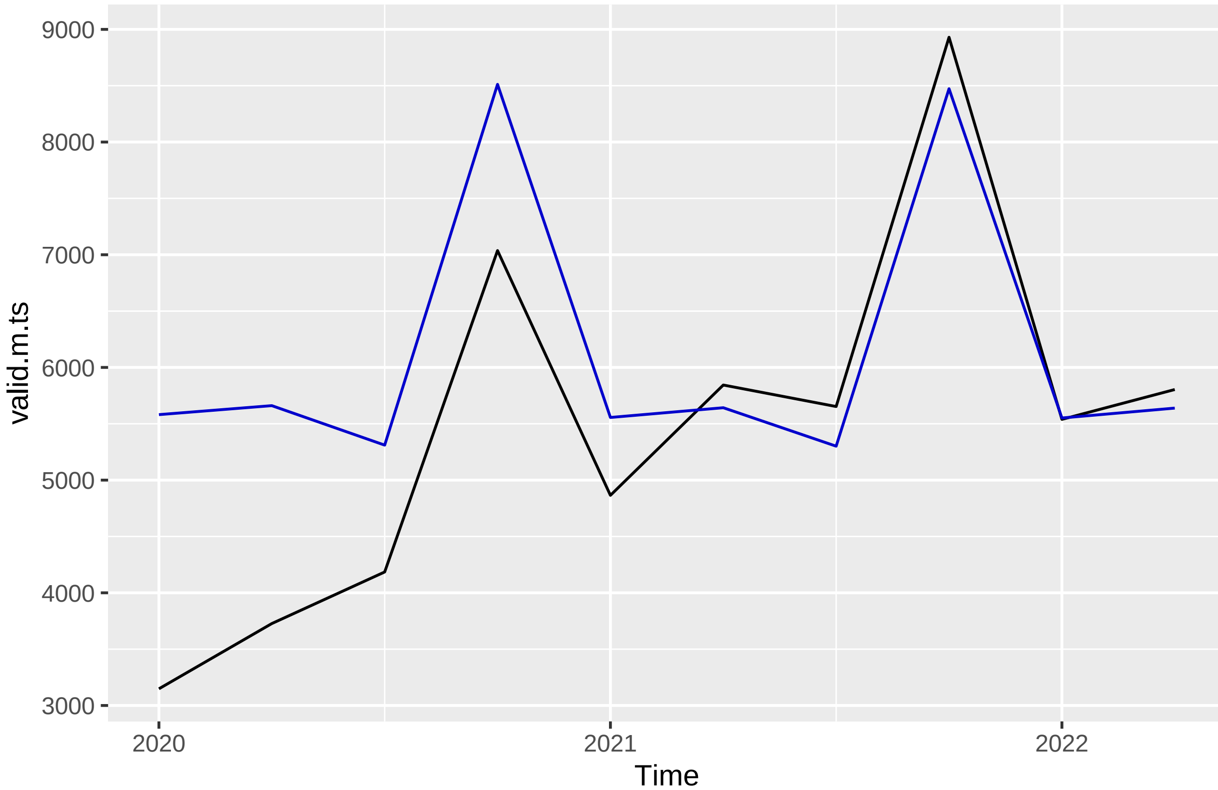



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)(1,1,0)[4]
## Q* = 3.0452, df = 5, p-value = 0.693
##
## Model df: 3.   Total lags used: 8
```

```
# model 2
am2 <- Arima(train.m.ts, order = c(2,0,1), seasonal = list(order = c(1,0,0), period = 4))
summary(am2)
```

```
## Series: train.m.ts
## ARIMA(2,0,1)(1,0,0)[4] with non-zero mean
##
## Coefficients:
##          ar1     ar2      ma1    sar1      mean
##       0.4142  0.2640  -0.1313  0.9896  6067.246
## s.e.  0.3951  0.2256   0.4250  0.0074  2027.586
##
## sigma^2 = 55630:  log likelihood = -315.43
## AIC=642.86   AICc=645.08   BIC=653.7
##
## Training set error measures:
##                    ME     RMSE      MAE         MPE     MAPE      MASE
## Training set 4.243583 222.3703 175.2347 -0.08122995 2.642154 0.7318551
##                    ACF1
## Training set 0.0565269
```

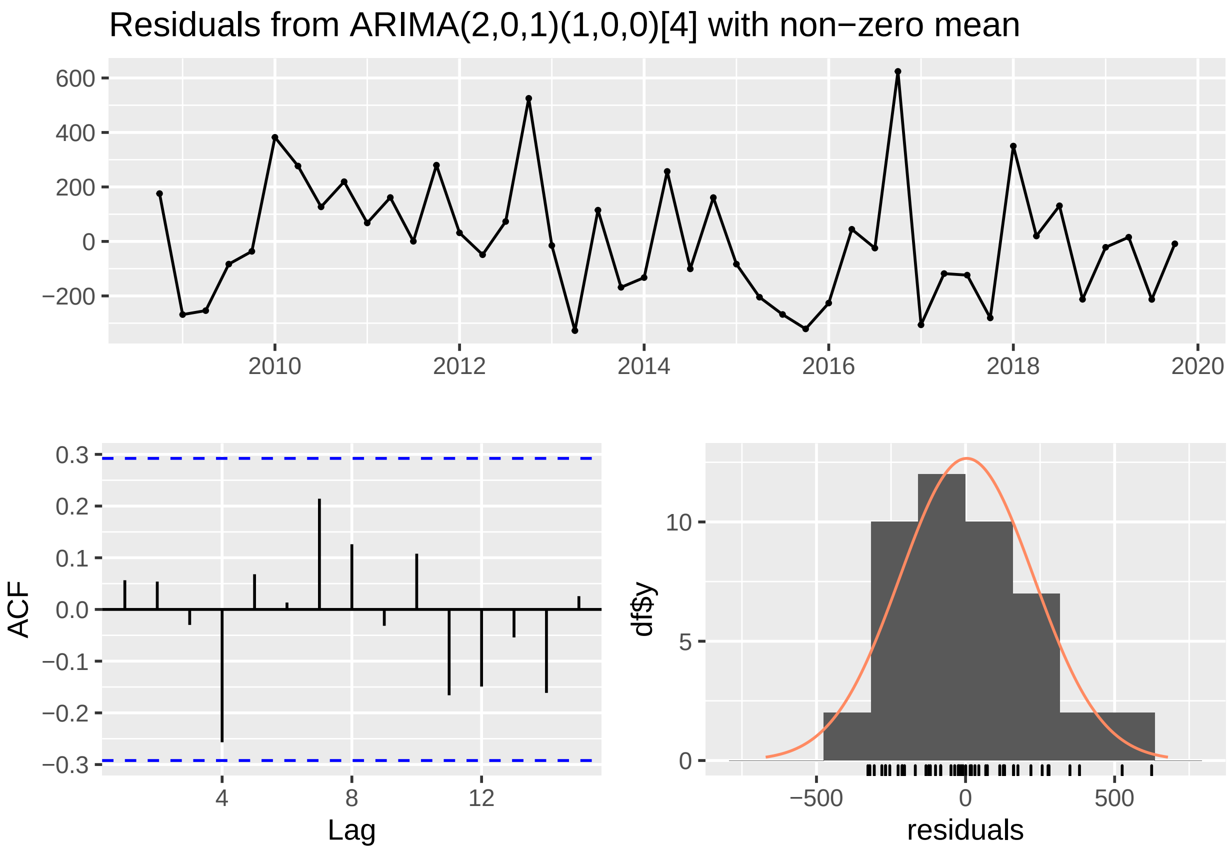
```r
am2.predict <- forecast(am2, h = 10, level = 0)
autoplot(valid.m.ts)+
  autolayer(am2.predict)
```




```r
accuracy(am2.predict, valid.m.ts)
```

```
##                      ME      RMSE      MAE          MPE      MAPE      MASE
## Training set    4.243583  222.3703 175.2347  -0.08122995  2.642154 0.7318551
## Test set     -649.648559 1182.4290 884.6324 -17.38464085 20.910085 3.6946040
##                   ACF1 Theil's U
## Training set 0.0565269        NA
## Test set     0.6636728  0.736998
```

```r
checkresiduals(am2)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,1)(1,0,0)[4] with non-zero mean
## Q* = 7.4626, df = 4, p-value = 0.1134
##
## Model df: 4.   Total lags used: 8
```

External Factors: Unemployment Rate, GDP Growth Rate (Inflation-adjusted)

Correlation Coefficient Value

```
sa <- rollmean(macy.ts, k = 4, align = "right")
newmacy <- ts.intersect(SA_M = sa, UR = ur.ts, GDPG = gdpg.ts)
cor(x = newmacy[,2], y = newmacy[,1]) # UR
```

```
## [1] -0.186684
```

```
cor(x = newmacy[,3], y = newmacy[,1]) # GDPG
```

```
## [1] 0.2826156
```

Split Data Set

```r
train.g.ts <- window(gdpg.ts, end = c(2019,4), frequency = 4)
valid.g.ts <- window(gdpg.ts, start = c(2020,1), frequency = 4)

train.u.ts <- window(ur.ts, end = c(2019,4), frequency = 4)
valid.u.ts <- window(ur.ts, start = c(2020,1), frequency = 4)
```

Time Series Linear Regression Models

```r
lm1 <- tslm(train.m.ts ~ trend + season + train.g.ts)
summary(lm1) # R square: 0.9443
```

```
##
## Call:
## tslm(formula = train.m.ts ~ trend + season + train.g.ts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -749.91 -248.17    7.54  191.71  591.46
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5697.72627  135.25038  42.127  < 2e-16 ***
## trend         -3.20261    4.68826  -0.683 0.498574
## season2        0.02641  146.52968   0.000 0.999857
## season3     -120.29466  146.67501  -0.820 0.417118
## season4     2906.56711  143.46201  20.260  < 2e-16 ***
## train.g.ts   132.92118   35.83934   3.709 0.000647 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 343.5 on 39 degrees of freedom
## Multiple R-squared:  0.9443, Adjusted R-squared:  0.9371
## F-statistic: 132.2 on 5 and 39 DF,  p-value: < 2.2e-16
```

```r
lm1.pred <- forecast(lm1, data.frame(train.g.ts = valid.g.ts))
accuracy(lm1.pred, valid.m.ts)
```

```
##                     ME      RMSE      MAE         MPE      MAPE     MASE
## Training set    0.0000  319.7667 257.0717  -0.2112999  3.877998 1.073642
## Test set     -869.1513 1104.0041 877.7779 -19.7487288 19.897359 3.665977
##                    ACF1 Theil's U
## Training set 0.54986131        NA
## Test set     0.08227184  0.517018
```

```r
autoplot(macy.ts, main = "lmcay ~ trend + season + gdp growth rate")+
  autolayer(lm1$fitted.values)+
  autolayer(lm1.pred$mean)
```

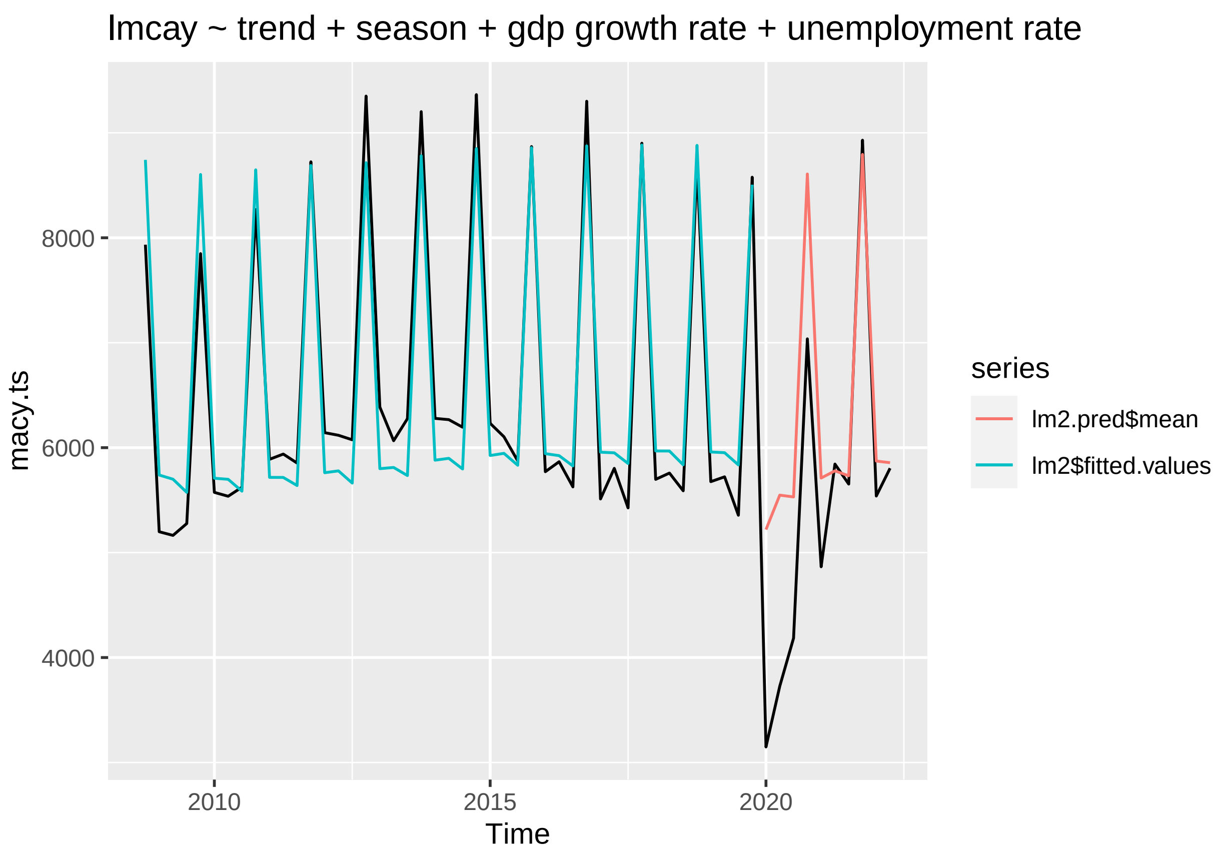

```r
lm2 <- tslm(train.m.ts ~ trend + season + train.u.ts)
summary(lm2) # R square: 0.9275
```

```
##
## Call:
## tslm(formula = train.m.ts ~ trend + season + train.u.ts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -808.34 -247.35   16.76  339.07  634.39
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6610.998    743.042   8.897 6.28e-11 ***
## trend         -7.701     12.123  -0.635    0.529
## season2       -1.886    167.243  -0.011    0.991
## season3     -113.797    167.449  -0.680    0.501
## season4     2923.057    165.145  17.700  < 2e-16 ***
## train.u.ts   -90.464     73.388  -1.233    0.225
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 391.9 on 39 degrees of freedom
## Multiple R-squared:  0.9275, Adjusted R-squared:  0.9182
## F-statistic: 99.72 on 5 and 39 DF,  p-value: < 2.2e-16
```

```r
lm2.pred <- forecast(lm2, data.frame(train.u.ts = valid.u.ts))
accuracy(lm2.pred, valid.m.ts)
```

```
##                         ME      RMSE      MAE         MPE      MAPE     MASE
## Training set -6.064686e-14  364.8665 309.4725  -0.3014489  4.808405 1.292490
## Test set     -7.918207e+02 1128.7490 831.9589 -19.2193916 19.745608 3.474617
##                   ACF1 Theil's U
## Training set 0.6280085        NA
## Test set     0.6721077 0.7413527
```

```
autoplot(macy.ts, main = "lmcay ~ trend + season + gdp growth rate + unemployment rate")+
  autolayer(lm2$fitted.values)+
  autolayer(lm2.pred$mean)
```




```
lm3 <- tslm(train.m.ts ~ trend + season + train.g.ts + train.u.ts)
summary(lm3) # R square: 0.9478
```

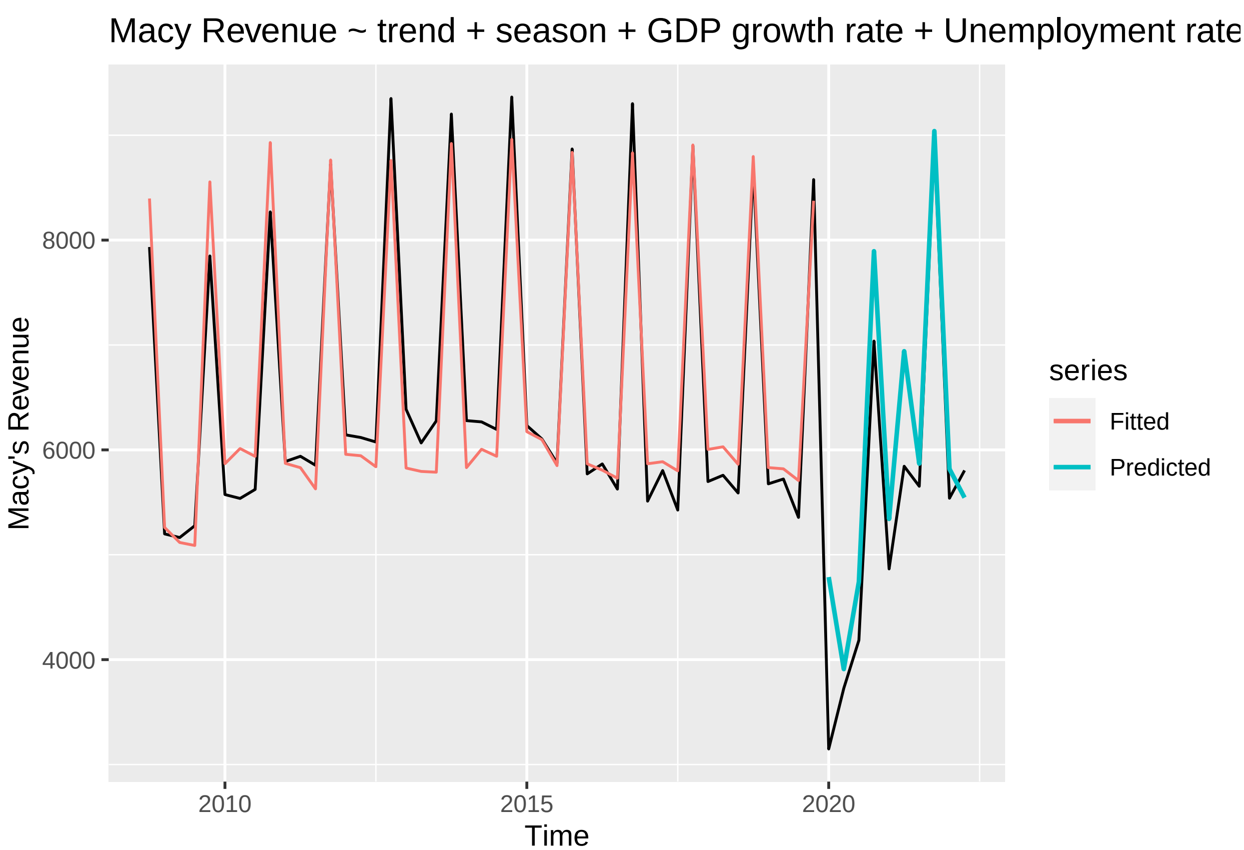
```
##
## Call:
## tslm(formula = train.m.ts ~ trend + season + train.g.ts + train.u.ts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -706.11 -271.19   17.31  224.95  590.92
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6693.088    639.107  10.473 9.32e-13 ***
## trend        -18.784     10.813  -1.737 0.090465 .
## season2        6.437    143.785   0.045 0.964526
## season3     -112.914    143.946  -0.784 0.437658
```

```
## season4      2936.997    142.012  20.681  < 2e-16 ***
## train.g.ts    135.244     35.185   3.844 0.000447 ***
## train.u.ts  -100.528     63.142  -1.592 0.119647
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 336.9 on 38 degrees of freedom
## Multiple R-squared:  0.9478, Adjusted R-squared:  0.9395
## F-statistic: 114.9 on 6 and 38 DF,  p-value: < 2.2e-16
```

```
lm3.pred <- forecast(lm3, data.frame(train.g.ts = valid.g.ts, train.u.ts = valid.u.ts))
accuracy(lm3.pred,valid.m.ts) # Best, Lowest MAPE: 12.56
```

```
##                          ME     RMSE      MAE         MPE      MAPE     MASE
## Training set  4.043093e-14 309.6071 248.6431  -0.1979147  3.779011 1.038440
## Test set     -5.154680e+02 734.8039 567.3489 -11.6646776 12.558560 2.369492
##                     ACF1 Theil's U
## Training set  0.46906549        NA
## Test set     -0.06642433 0.3361403
```

```
autoplot(macy.ts, ylab = "Macy's Revenue",main = "Macy Revenue ~ trend + season + GDP growth rate + Uner
  autolayer(lm3.pred$mean, series = "Predicted", lwd = 0.8)+
  autolayer(lm3$fitted.values, series = "Fitted")
```




```
# checkreservoirs
```

ArimaX

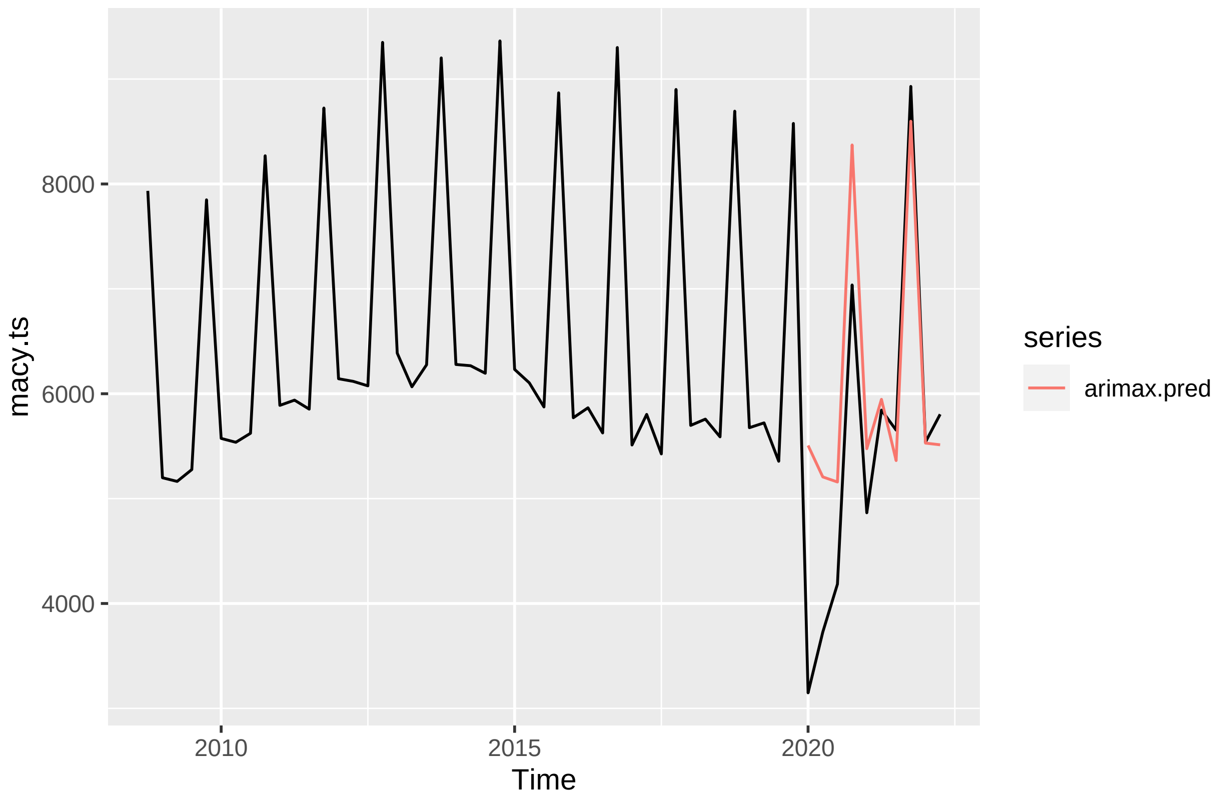
```
axm1 <- Arima(train.m.ts,order = c(2,0,1),seasonal = list(order = c(1,1,0),period = 4),xreg = train.g.ts
summary(axm1)
```

```
## Series: train.m.ts
## Regression with ARIMA(2,0,1)(1,1,0)[4] errors
##
## Coefficients:
##          ar1     ar2      ma1     sar1     xreg
##       0.5294  0.3274  -0.2701  -0.3294  37.1803
## s.e.  0.3044  0.2314   0.3191   0.1535  22.8644
##
## sigma^2 = 47258:  log likelihood = -276.64
## AIC=565.29   AICc=567.76   BIC=575.57
##
## Training set error measures:
##                    ME     RMSE     MAE         MPE     MAPE      MASE
## Training set 1.589917 194.4377 148.151 -0.03663296 2.194893 0.6187419
##                    ACF1
## Training set 0.02312432
```

```
arimax.pred <- forecast(axm1, h =10, xreg = valid.g.ts)$mean
accuracy(arimax.pred, valid.m.ts)
```

```
##                 ME    RMSE      MAE       MPE     MAPE      ACF1 Theil's U
## Test set -593.7967 1055.82 778.3749 -15.71486 18.52307 0.5946314 0.5999315
```

```
autoplot(macy.ts)+
  autolayer(arimax.pred)
```



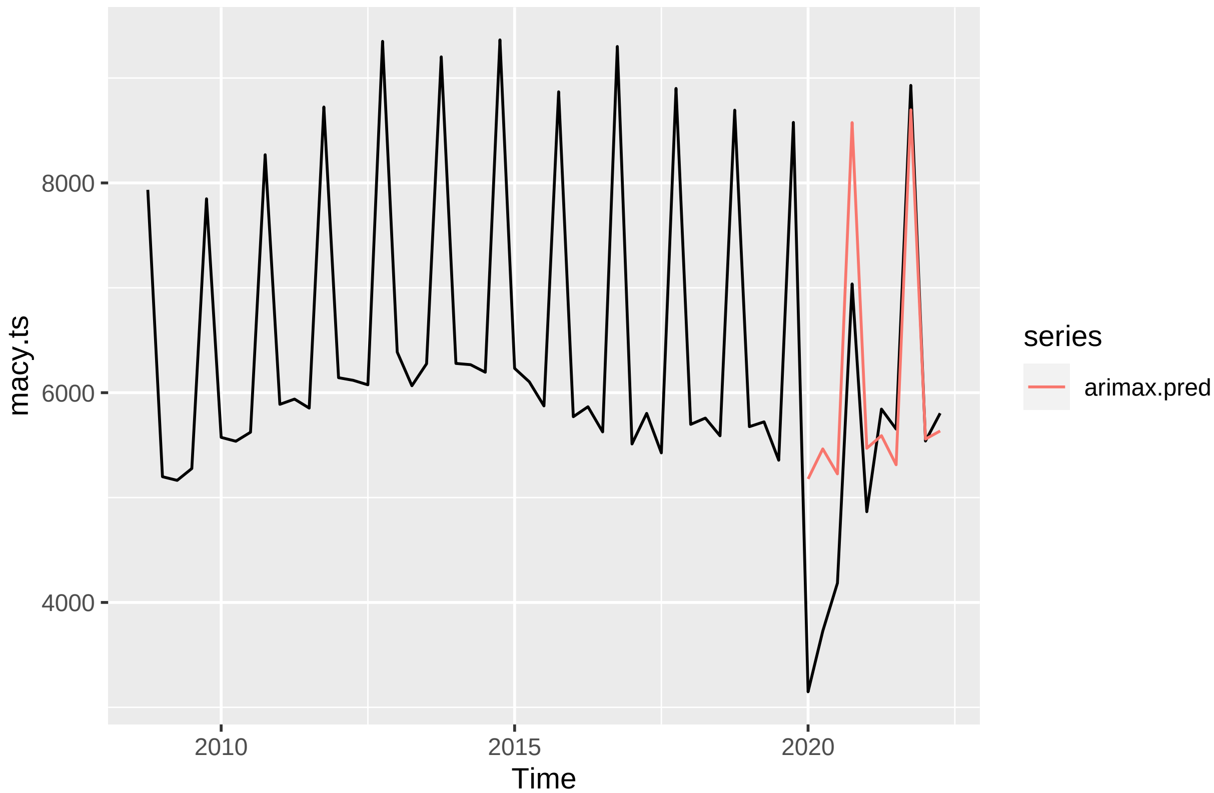
```r
axm2 <- Arima(train.m.ts,order = c(2,0,1),seasonal = list(order = c(1,1,0),period = 4),xreg = train.u.ts
summary(axm2)
```

```
## Series: train.m.ts
## Regression with ARIMA(2,0,1)(1,1,0)[4] errors
##
## Coefficients:
##          ar1     ar2      ma1     sar1      xreg
##       0.6034  0.2959  -0.3495  -0.3595  -53.9509
## s.e.  0.2845  0.2335   0.2855   0.1520   46.1780
##
## sigma^2 = 48590:  log likelihood = -277.31
## AIC=566.62   AICc=569.09   BIC=576.9
##
## Training set error measures:
##                     ME     RMSE      MAE         MPE     MAPE      MASE
## Training set -1.669957 197.1607 144.8919 -0.09282955 2.187435 0.6051306
##                    ACF1
## Training set 0.02853124
```

```r
arimax.pred <- forecast(axm2, h =10, xreg = valid.u.ts)$mean
accuracy(arimax.pred, valid.m.ts)
```

```
##                 ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U
## Test set -597.4419 1059.062 796.4766 -15.47201 18.64596 0.6611507 0.6864707
```

```r
autoplot(macy.ts)+
  autolayer(arimax.pred)
```



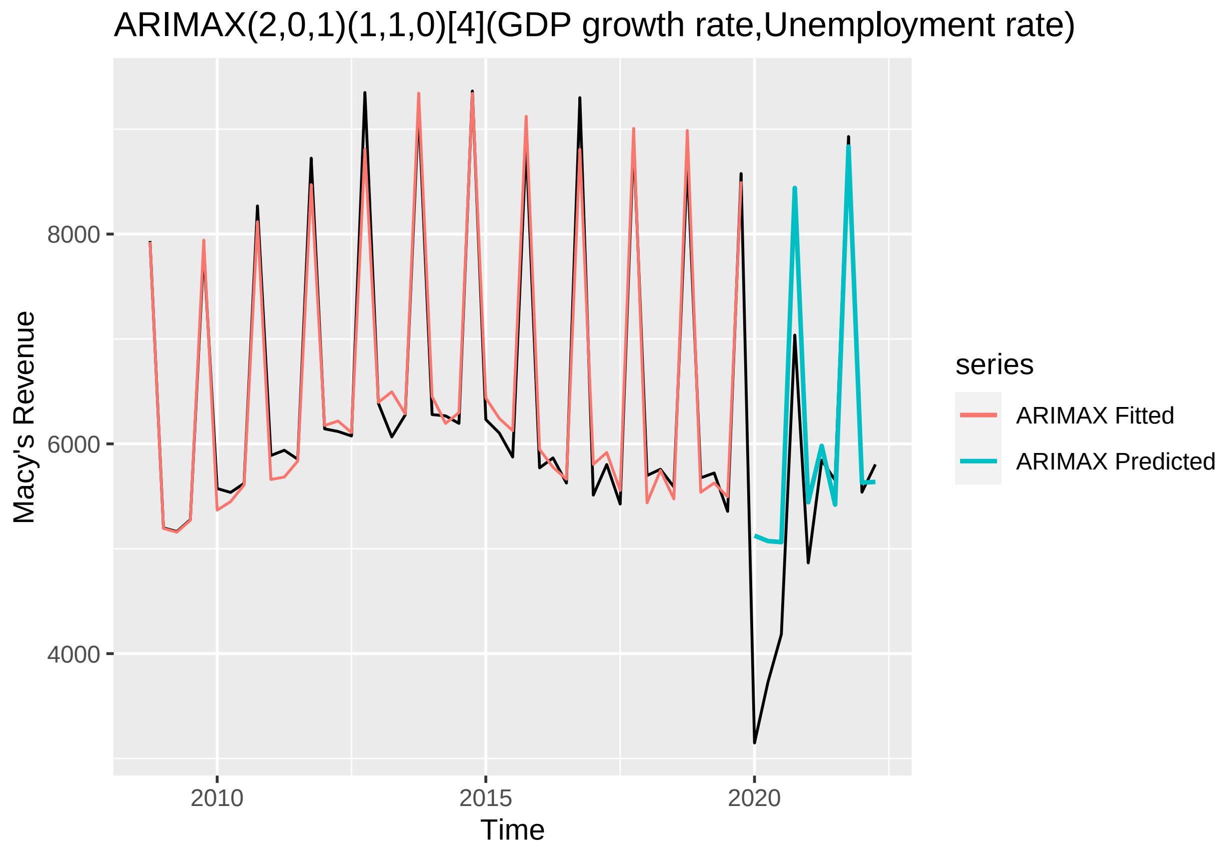
```r
axm3 <- Arima(train.m.ts,order = c(2,0,1),seasonal = list(order = c(1,1,0),period = 4),xreg = cbind(tra
summary(axm3)
```

```
## Series: train.m.ts
## Regression with ARIMA(2,0,1)(1,1,0)[4] errors
##
## Coefficients:
##          ar1     ar2      ma1     sar1  train.u.ts  train.g.ts
##       0.5291  0.3337  -0.2927  -0.3535    -54.0999     36.9437
## s.e.  0.2842  0.2198   0.2936   0.1517     44.5126     22.2864
##
## sigma^2 = 46868:  log likelihood = -275.92
## AIC=565.84   AICc=569.24   BIC=577.84
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE    MAPE      MASE
## Training set -2.447965 190.9262 142.5642 -0.1222242 2.12156 0.5954094
##                    ACF1
## Training set 0.01870448
```

```r
arimax.pred <- forecast(axm3, h =10, xreg = cbind(train.u.ts = valid.u.ts,train.g.ts = valid.g.ts))$mea
accuracy(arimax.pred, valid.m.ts)
```

```
##                 ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U
## Test set -591.5511 944.0488 690.4446 -14.76579 16.37605 0.5825612 0.5646098
```

```r
autoplot(macy.ts,ylab = "Macy's Revenue", main = "ARIMAX(2,0,1)(1,1,0)[4](GDP growth rate,Unemployment r
  autolayer(axm3$fitted, series = "ARIMAX Fitted" )+
  autolayer(arimax.pred, series = "ARIMAX Predicted", lwd = 0.8 )
```



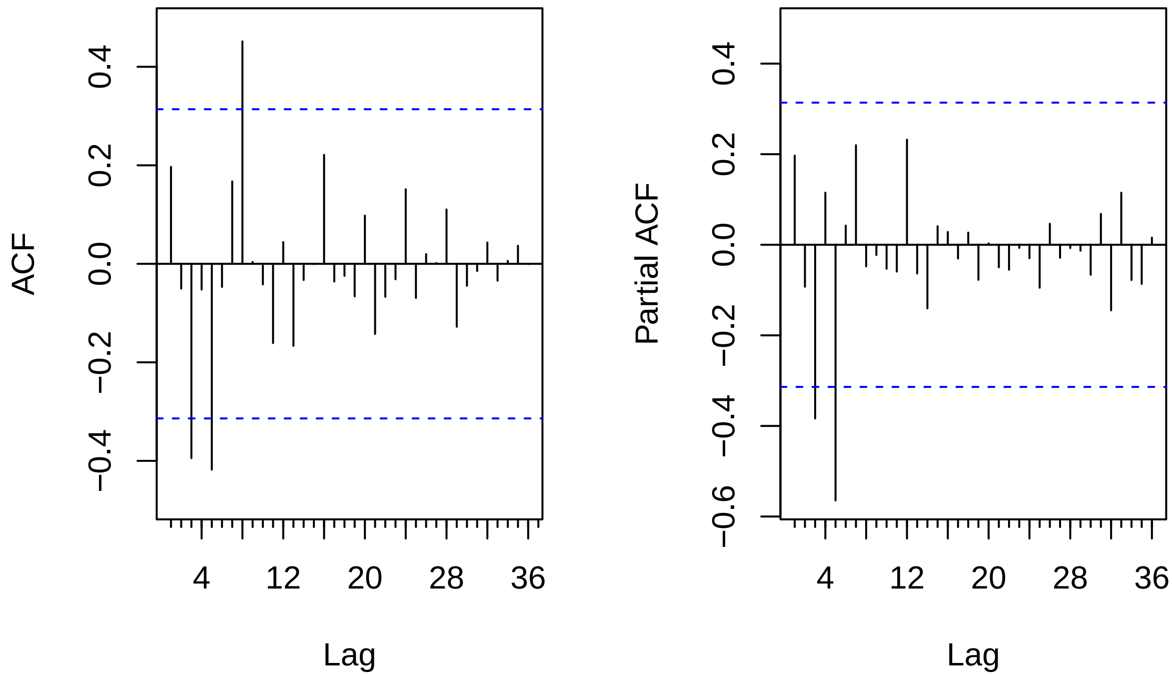
Competitor Analysis: Amazon

Split Data Set

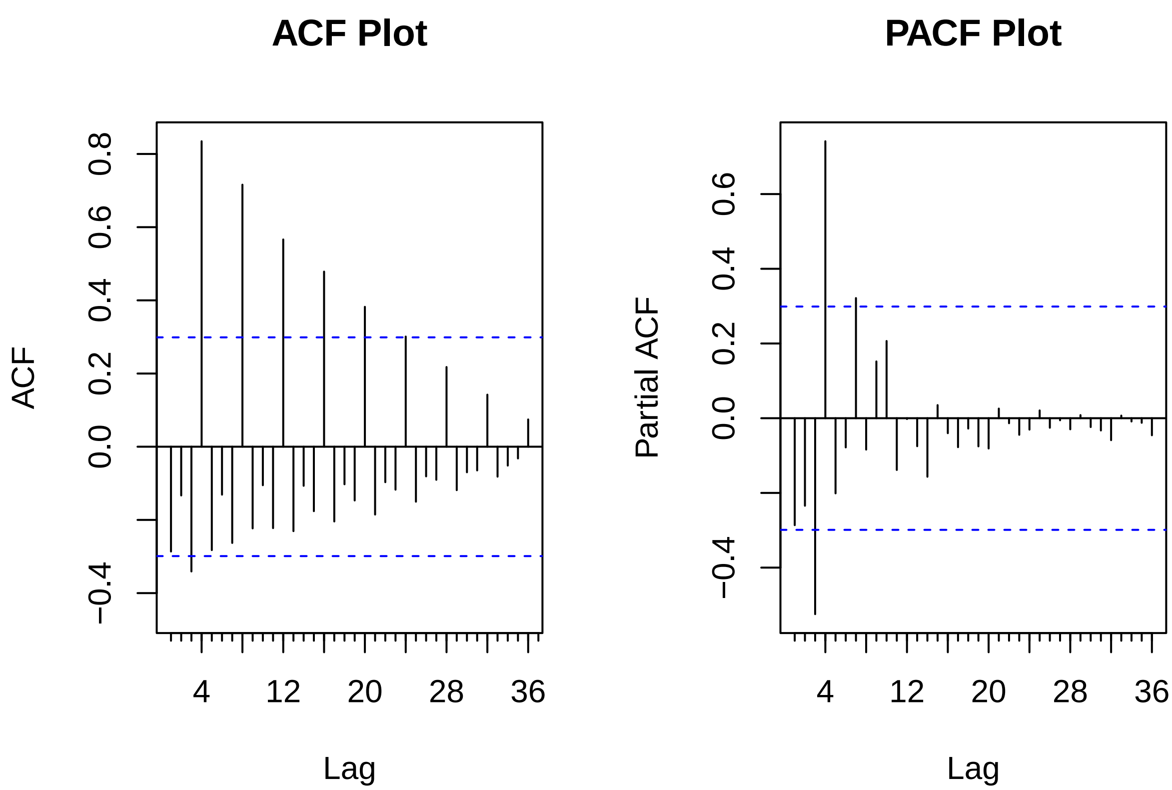
```
gdpg.ts <- window(gdpg.ts,start = c(2009,1), frequency = 4)
ur.ts <- window(ur.ts, start =c(2009,1), frequency = 4)

train.a.ts <- window(ama.ts, end = c(2019,4), frequency = 4)
valid.a.ts <- window(ama.ts, start = c(2020,1), frequency = 4)

train.g.ts <- window(gdpg.ts, end = c(2019,4), frequency = 4)
valid.g.ts <- window(gdpg.ts, start = c(2020,1), frequency = 4)

train.u.ts <- window(ur.ts, end = c(2019,4), frequency = 4)
valid.u.ts <- window(ur.ts, start = c(2020,1), frequency = 4)
```

Arima

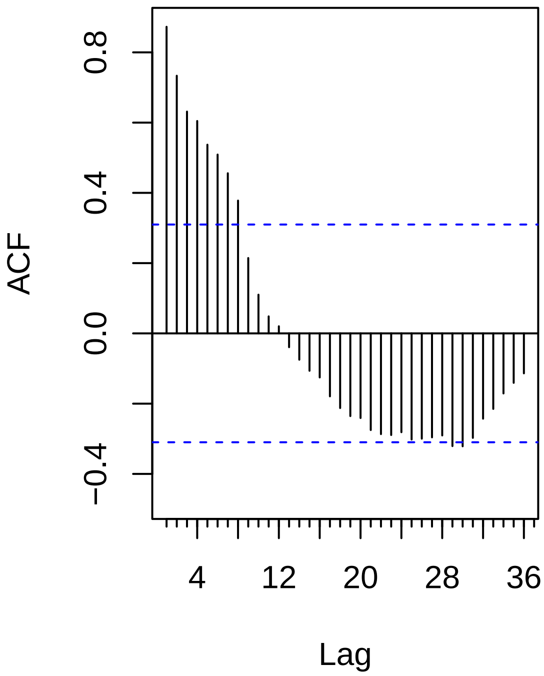
```
#(1) Detrend and Deseasonality
ddtrain<-diff(diff(train.a.ts,1),4)
par(mfrow=c(1,2))
Acf(ddtrain,36,main="")
Pacf(ddtrain,36, main="")
```

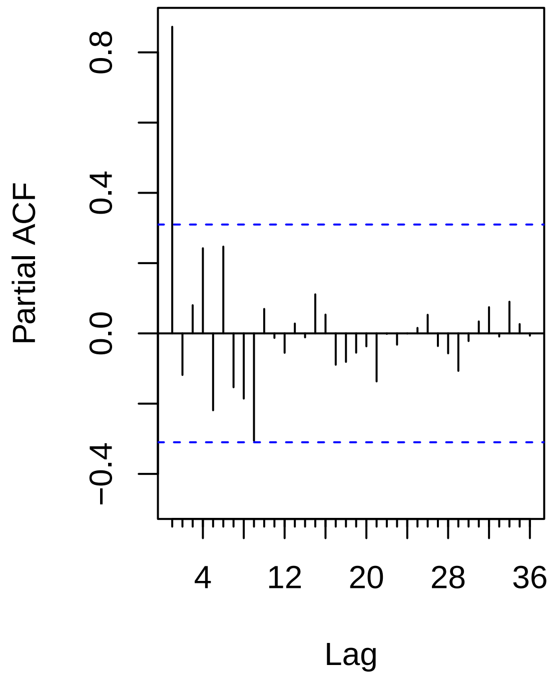



```r
par(mfrow=c(1,1))

#(2) Detrend
dttrain<-diff(train.a.ts,1)
par(mfrow=c(1,2))
Acf(dttrain,36,main="ACF Plot")
Pacf(dttrain,36, main="PACF Plot")
```

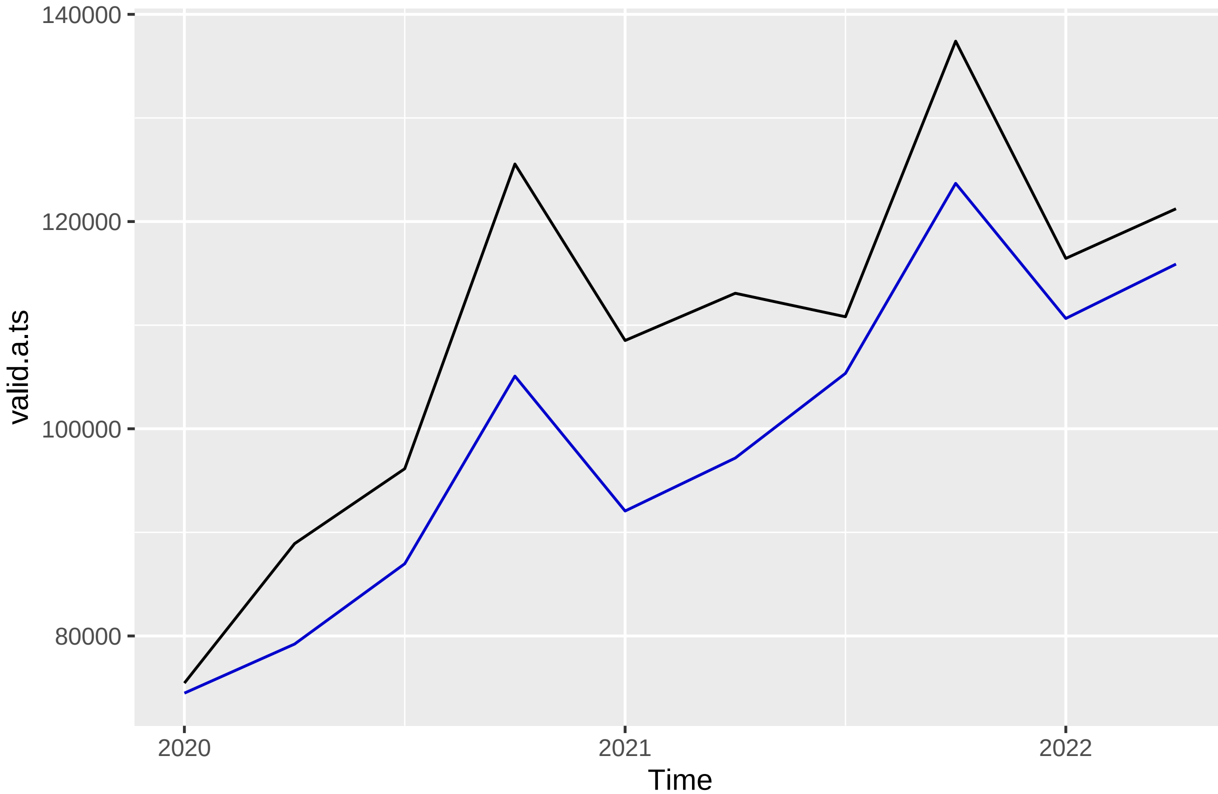


```
par(mfrow=c(1,1))

#(3) Deseasonality
dstrain<-diff(train.a.ts,4)
par(mfrow=c(1,2))
Acf(dstrain,36,main="")
Pacf(dstrain,36, main="")
```





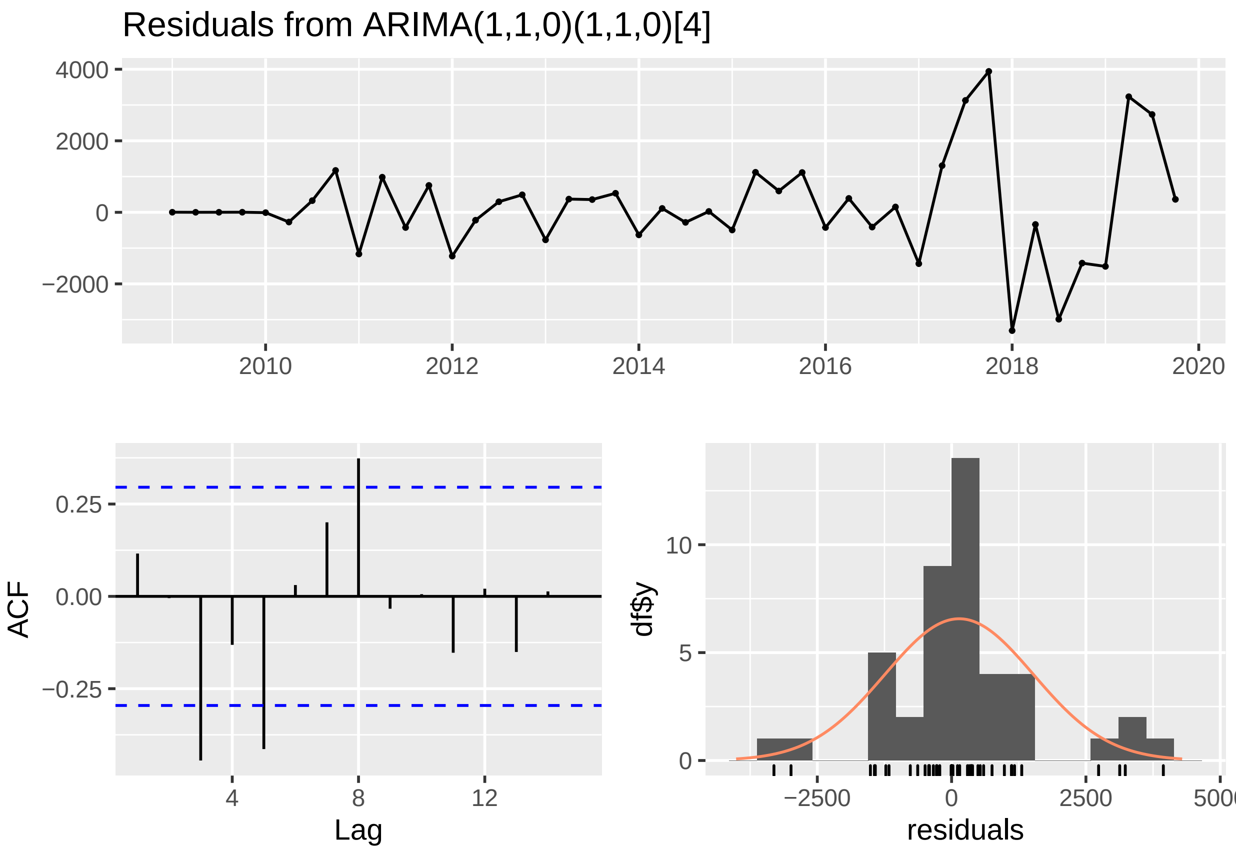

```
par(mfrow=c(1,1))

aam <- Arima(train.a.ts, order = c(1,1,0), seasonal = list(order=c(1,1,0),period=4))
aam.predict <- forecast(aam, h = 10,level = 0)
autoplot(valid.a.ts)+
  autolayer(aam.predict)
```




```
accuracy(aam.predict,valid.a.ts)
```

```
##                     ME      RMSE        MAE       MPE     MAPE      MASE
## Training set   140.1002  1374.978   927.8731 0.2649221 3.271664 0.1449728
## Test set     10300.2152 11842.374 10300.2152 9.1571551 9.157155 1.6093269
##                   ACF1 Theil's U
## Training set 0.1158612        NA
## Test set     0.1616983 0.7511982
```

```
checkresiduals(aam)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0)(1,1,0)[4]
## Q* = 30.222, df = 6, p-value = 3.567e-05
##
## Model df: 2.   Total lags used: 8
```

External Factors: Unemployment Rate, GDP Growth Rate (Inflation-adjusted)

```
# Correlation Coefficient Value

sa.ama <- rollmean(ama.ts, k = 4, align = "right")
newama <- ts.intersect(SA_M = sa.ama, UR = ur.ts, GDPG = gdpg.ts)

cor(x = newama[,2], y = newama[,1]) # -0.4943394
```

```
## [1] -0.4943394
```

```
cor(x = newama[,3], y = newama[,1]) # 0.1152857
```

```
## [1] 0.1152857
```

```
# Linear Regression

alm1 <- tslm(train.a.ts ~ trend + season + train.g.ts)
alm1.pred <- forecast(alm1, data.frame(train.g.ts = valid.g.ts))
accuracy(alm1.pred,valid.a.ts)
```

```
##                        ME     RMSE       MAE       MPE     MAPE      MASE
## Training set 6.414667e-13  6347.34  5430.597  7.814899 28.61952 0.8484877
## Test set     3.793965e+04 41587.45 37939.650 33.146097 33.14610 5.9277694
##                   ACF1 Theil's U
## Training set 0.7822810        NA
## Test set     0.5268681  2.433226
```

```
autoplot(ama.ts, main = "Amazon Revenue ~ trend + season + GDP growth rate")+
  autolayer(alm1$fitted.values)+
  autolayer(alm1.pred$mean)
```


```
alm2 <- tslm(train.a.ts ~ trend + season + train.u.ts)
summary(alm2) # 0.9485
```

```
##
## Call:
## tslm(formula = train.a.ts ~ trend + season + train.u.ts)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -6339  -3546  -1441   3386  11965
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -67873.3     9997.8  -6.789 4.75e-08 ***
## trend         2478.9      168.6  14.707  < 2e-16 ***
## season2      -1221.0     2175.4  -0.561  0.57790
## season3       -376.5     2178.5  -0.173  0.86370
```

```
## season4       6102.1     2224.8   2.743  0.00924 **
## train.u.ts    6169.1      994.5   6.203 3.00e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5097 on 38 degrees of freedom
## Multiple R-squared:  0.9485, Adjusted R-squared:  0.9417
## F-statistic: 139.9 on 5 and 38 DF,  p-value: < 2.2e-16
```

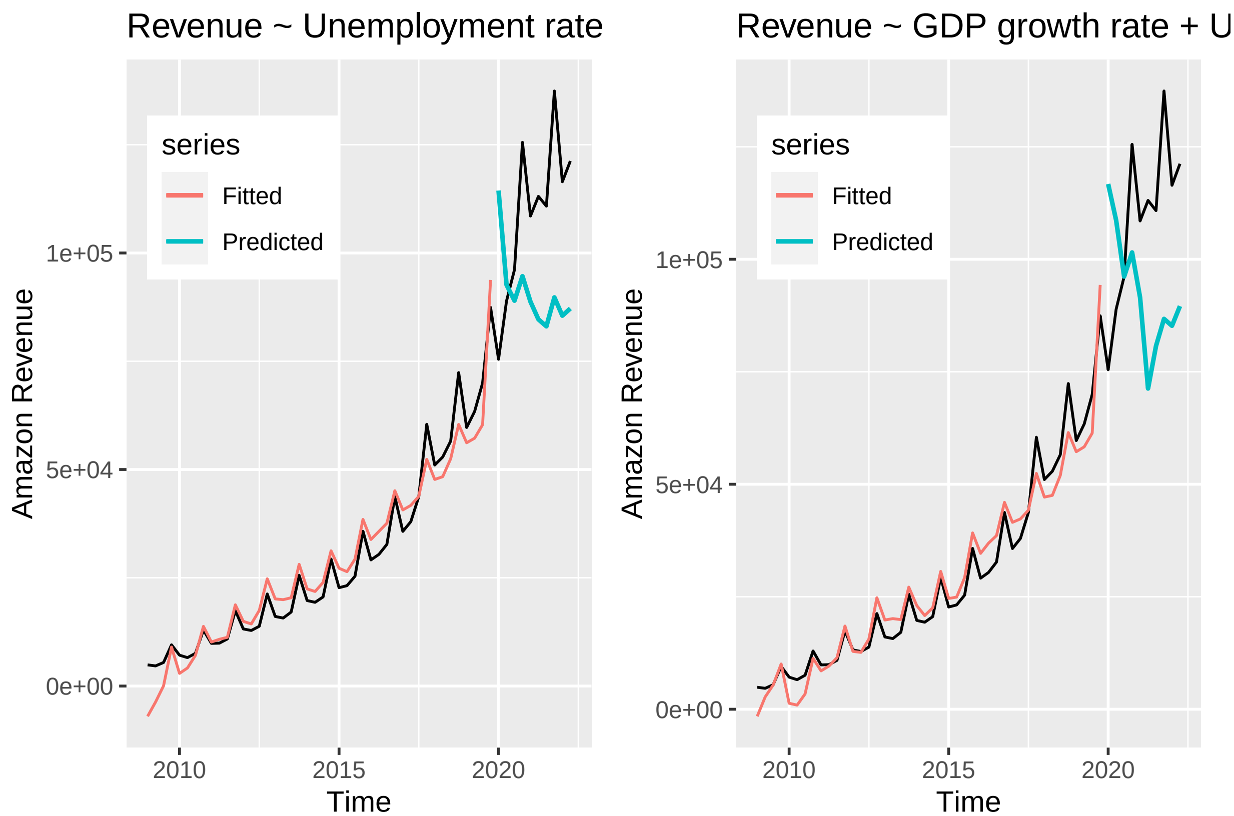
```
alm2.pred <- forecast(alm2, data.frame(train.u.ts = valid.u.ts))
accuracy(alm2.pred,valid.a.ts)
```

```
##                        ME      RMSE       MAE       MPE     MAPE      MASE
## Training set 5.992984e-13  4737.186  3842.981  7.759249 24.50189 0.6004352
## Test set     1.839443e+04 29823.556 26934.076 13.394480 24.56269 4.2082357
##                   ACF1 Theil's U
## Training set 0.5880114        NA
## Test set     0.4243564  1.579829
```

```
almp2 <- autoplot(ama.ts, ylab="Amazon Revenue", main = "Revenue ~ Unemployment rate")+
  autolayer(alm2$fitted.values, series = "Fitted")+
  autolayer(alm2.pred$mean, series = "Predicted", lwd = 0.8)+ theme(legend.position = c(0.25, 0.8))

alm3 <- tslm(train.a.ts ~ trend + season + train.g.ts + train.u.ts)
alm3.pred <- forecast(alm3, data.frame(train.u.ts = valid.u.ts, train.g.ts = valid.g.ts))
accuracy(alm3.pred,valid.a.ts)
```

```
##                        ME      RMSE       MAE       MPE     MAPE     MASE
## Training set 2.582379e-13  4300.563  3490.776  4.738362 18.39303 0.545406
## Test set     1.653752e+04 31864.929 28744.985 11.174460 26.56076 4.491176
##                   ACF1 Theil's U
## Training set 0.5694260        NA
## Test set     0.5547249  1.728586
```

```
almp3 <- autoplot(ama.ts, ylab="Amazon Revenue", main = "Revenue ~ GDP growth rate + Unemployment rate")
  autolayer(alm3$fitted.values, series = "Fitted")+
  autolayer(alm3.pred$mean, series = "Predicted", lwd = 0.8)+ theme(legend.position = c(0.25, 0.8))

almp2 + almp3
```



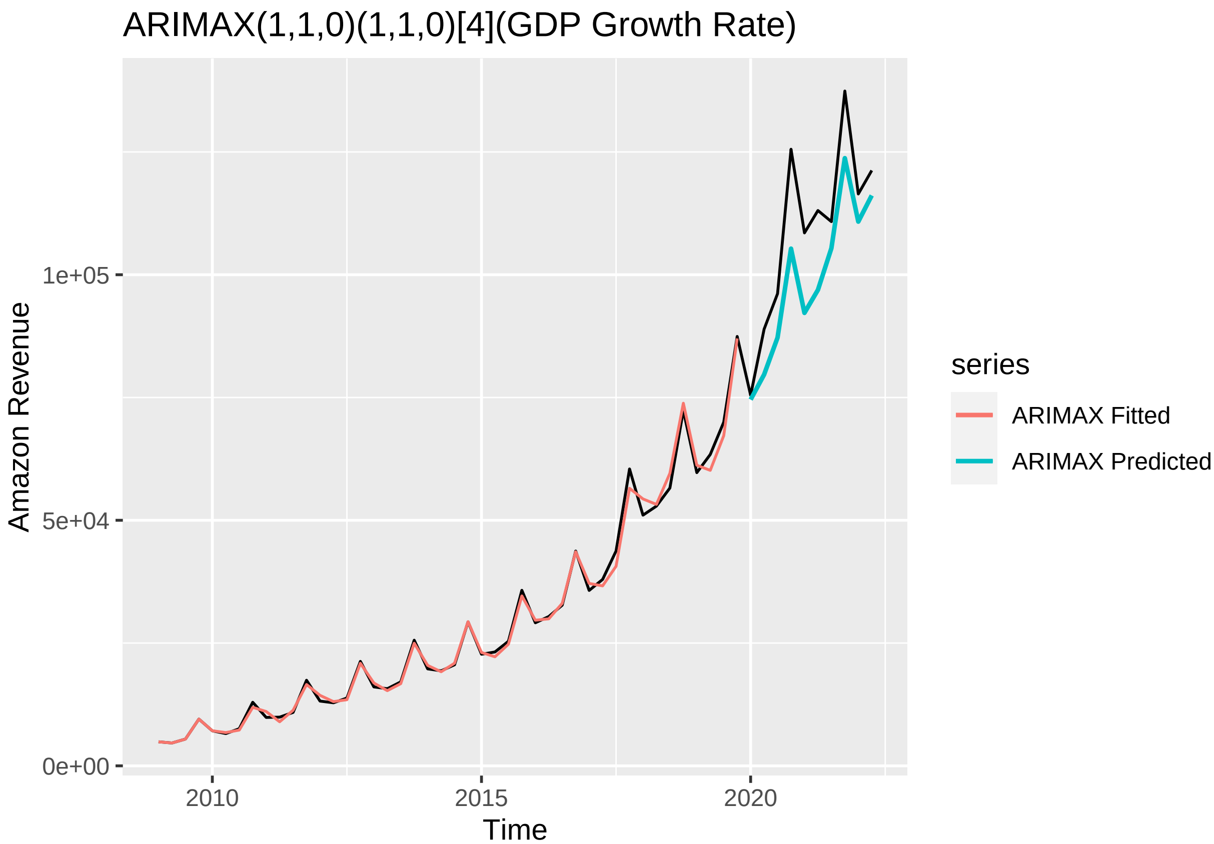

```r
# ArimaX

aaxm1 <- Arima(train.a.ts,order = c(1,1,0),seasonal = list(order = c(1,1,0),period = 4),xreg = train.g.
summary(aaxm1)
```

```
## Series: train.a.ts
## Regression with ARIMA(1,1,0)(1,1,0)[4] errors
##
## Coefficients:
##          ar1    sar1      xreg
##       0.4445  0.3481  -38.3525
## s.e.  0.1749  0.2033  148.8733
##
## sigma^2 = 2306796:  log likelihood = -339.86
## AIC=687.72   AICc=688.9   BIC=694.38
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
## Training set 138.8012 1373.819 930.0582 0.2490308 3.246154 0.1453142 0.1171472
```

```r
arimax.pred <- forecast(aaxm1, h =10, xreg = valid.g.ts)$mean
accuracy(arimax.pred, valid.a.ts)
```

```
##               ME     RMSE     MAE     MPE    MAPE      ACF1 Theil's U
## Test set 10158.7 11733.82 10158.7 9.01778 9.01778 0.1681976 0.7423479
```

```r
autoplot(ama.ts, ylab = "Amazon Revenue", main = "ARIMAX(1,1,0)(1,1,0)[4](GDP Growth Rate)")+
  autolayer(aaxm1$fitted, series = "ARIMAX Fitted")+
  autolayer(arimax.pred, series = "ARIMAX Predicted", lwd = 0.8)
```



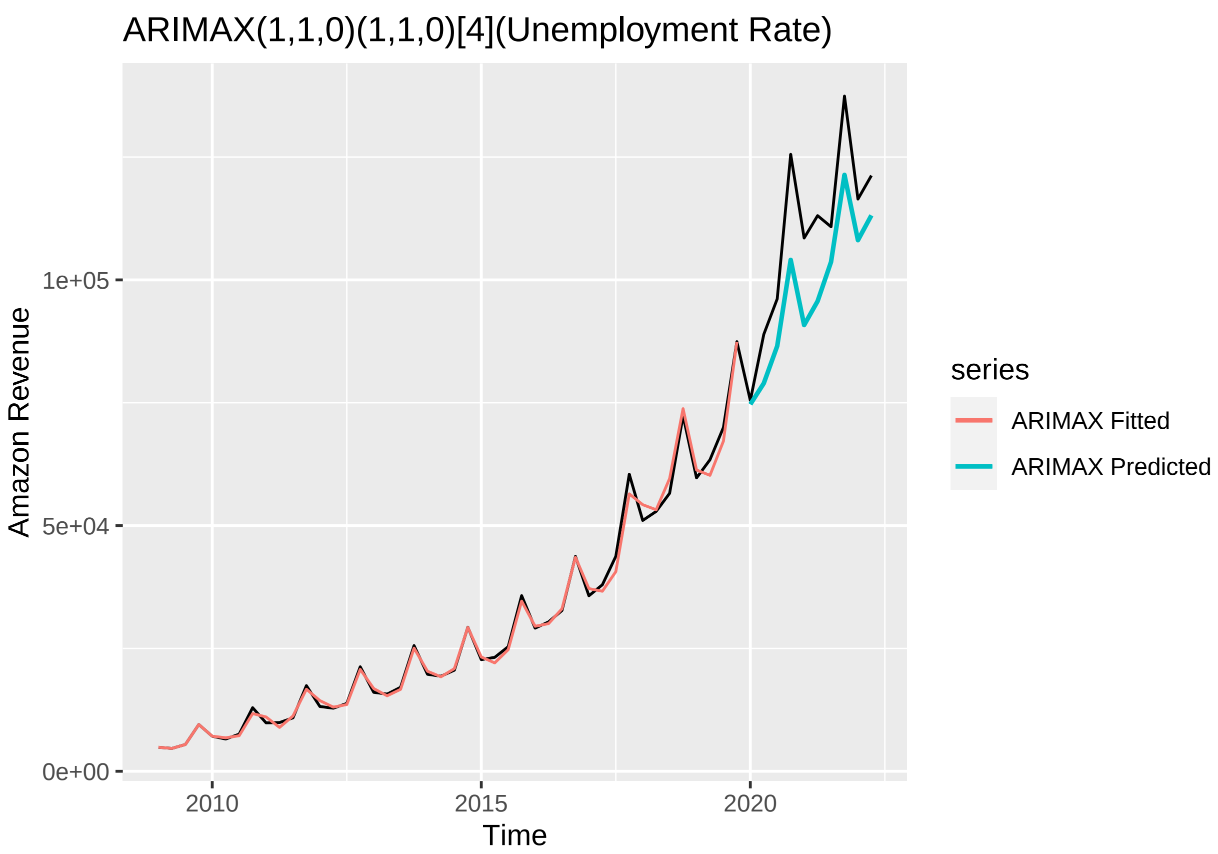

```
aaxm2 <- Arima(train.a.ts,order = c(1,1,0),seasonal = list(order = c(1,1,0),period = 4),xreg = train.u.t
summary(aaxm2)
```

```
## Series: train.a.ts
## Regression with ARIMA(1,1,0)(1,1,0)[4] errors
##
## Coefficients:
##          ar1    sar1      xreg
##       0.4246  0.3393   89.8109
## s.e.  0.1910  0.2110  334.1054
##
## sigma^2 = 2309611:  log likelihood = -339.86
## AIC=687.72   AICc=688.89   BIC=694.37
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
## Training set 135.8579 1374.657 921.0484 0.2708652 3.251021 0.1439065 0.1079263
```

```
arimax.pred <- forecast(aaxm2, h =10, xreg = valid.u.ts)$mean
accuracy(arimax.pred, valid.a.ts)
```

```
##                ME     RMSE      MAE      MPE     MAPE     ACF1 Theil's U
## Test set 11656.32 13089.63 11656.32 10.30057 10.30057 0.134893 0.8197492
```

```
autoplot(ama.ts, ylab = "Amazon Revenue", main = "ARIMAX(1,1,0)(1,1,0)[4](Unemployment Rate)")+
  autolayer(aaxm2$fitted, series = "ARIMAX Fitted")+
  autolayer(arimax.pred, series = "ARIMAX Predicted", lwd = 0.8)
```



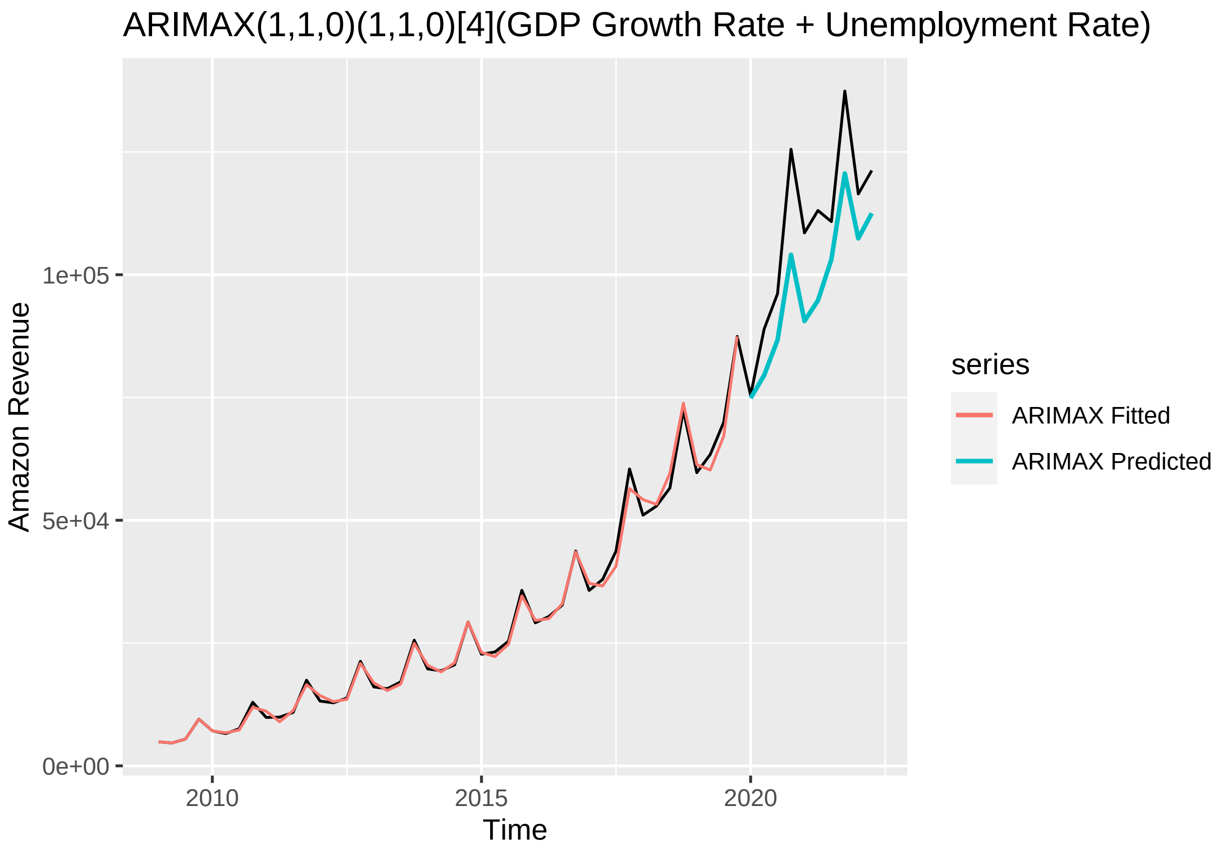

```
aaxm3 <- Arima(train.a.ts,order = c(1,1,0),seasonal = list(order = c(1,1,0),period = 4),xreg = cbind(tra
summary(aaxm3)
```

```
## Series: train.a.ts
## Regression with ARIMA(1,1,0)(1,1,0)[4] errors
##
## Coefficients:
##          ar1    sar1  train.u.ts  train.g.ts
##       0.4219  0.3353    122.9772    -54.5236
## s.e.  0.1902  0.2115    346.0827    157.2726
##
## sigma^2 = 2369200:  log likelihood = -339.8
## AIC=689.6   AICc=691.41   BIC=697.91
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
## Training set 132.0378 1372.804 920.5755 0.2495544 3.201961 0.1438326 0.1078655
```

```
arimax.pred <- forecast(aaxm3, h =10, xreg = cbind(train.u.ts = valid.u.ts,train.g.ts = valid.g.ts))$mea
accuracy(arimax.pred, valid.a.ts)
```

```
##                ME     RMSE      MAE     MPE    MAPE      ACF1 Theil's U
## Test set 11942.77 13403.86 11942.77 10.5137 10.5137 0.1468974 0.8331745
```

```
autoplot(ama.ts, ylab = "Amazon Revenue", main = "ARIMAX(1,1,0)(1,1,0)[4](GDP Growth Rate + Unemploymen
  autolayer(aaxm3$fitted, series = "ARIMAX Fitted")+
  autolayer(arimax.pred, series = "ARIMAX Predicted", lwd = 0.8)
```


ARIMAX(1,1,0)(1,1,0)[4](GDP Growth Rate + Unemployment Rate)
Amazon Revenue
1e+05
5e+04
0e+00
2010
2015
2020
Time
series
ARIMAX Fitted
ARIMAX Predicted