TIME SERIES ANALYSIS PROJECT

DSC - 425

FORTUNE 500

ARTE MOALIN

LOUIS NEWMAN

HARNAIN KAUR SARDARNI

DEPAUL UNIVERSITY

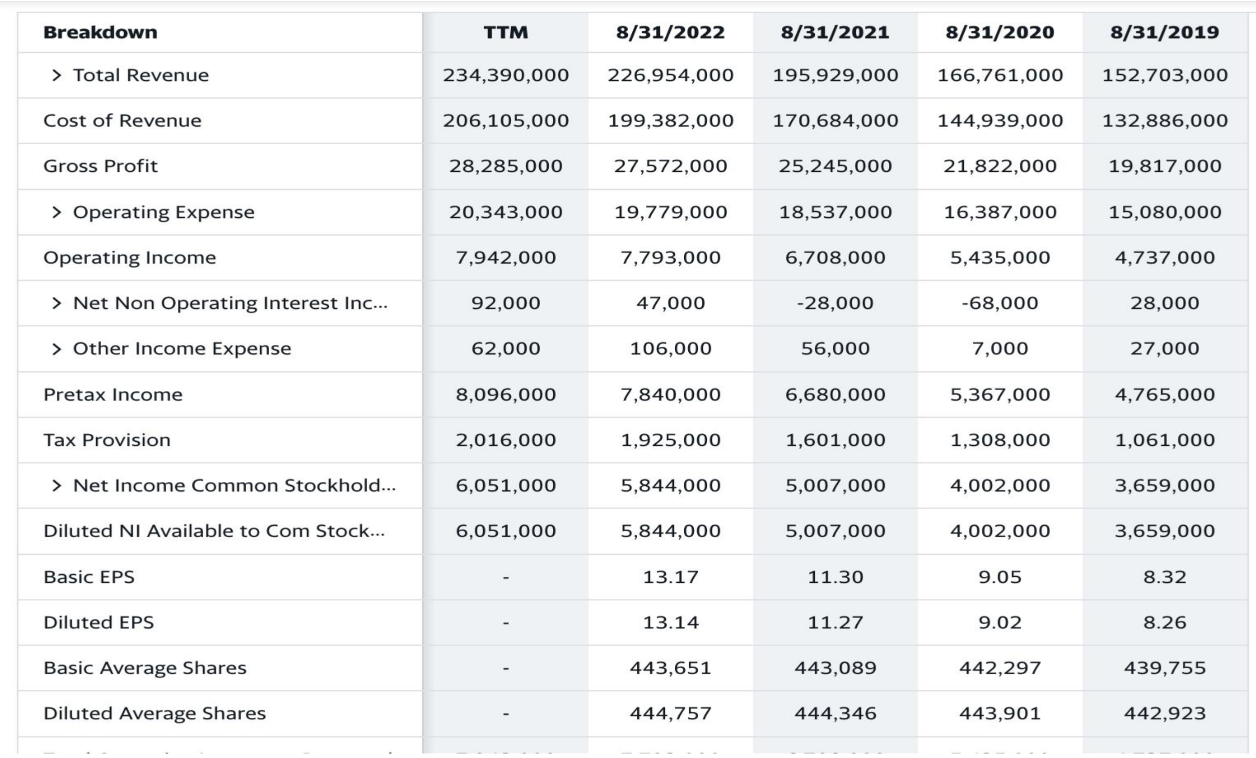
INTRODUCTION:

The analysis of the financial data plays an important role in understanding the performance and trends of the companies. Time series analysis plays a vital role in understanding the company's performance by identifying trends and patterns, forecasting future outcomes, exploring the relationships between variables and even evaluating past performances. Performing time series gives us insights for achieving success in business and decision making as well.

This project involves analyzing the financial performance of top 12 "Fortune 500" companies i.e., Walmart, Amazon, Apple, CVS, United Health Group, ExxonMobil Corp., BirkShire Heatherway, Alphabet Inc., McKesson, AmerisourceBergen, Microsoft and Costco. Each companies dataset's contains financial information for the company, such as total revenue, pretax income, and basic earnings per share etc.. By analyzing the growth rates of the financial metrics, such as Total Revenue, Pretax Income, Basic EPS, Operating Expenses, Operating Income, Cost of Revenue & Gross Profit. We can gain insights into the companies financial growth and it's stability.

OBJECTIVE:

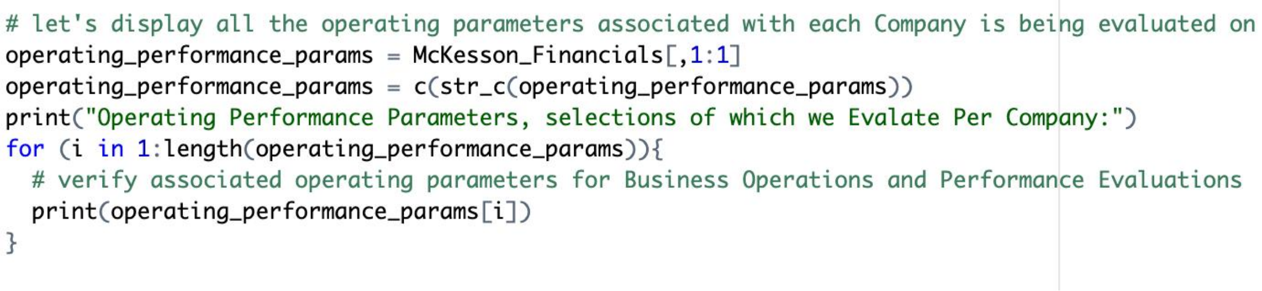
The goal of this project is to perform time series analysis and forecasting on the financial data for the companies Walmart, Amazon, Apple, CVS, United Health Group, ExxonMobil Corp., BirkShire Heatherway, Alphabet Inc., McKesson, AmerisourceBergen, Microsoft and Costco. The datasets were retrieved form yahoo finance's income statements for each company. The analysis aims to gain insights into the companies total revenue and other financial metrics trends and patterns, to generate forecasts for future periods using time series models. And look into the CEO's performances and how it affects the company's growth.



Breakdown	TTM	8/31/2022	8/31/2021	8/31/2020	8/31/2019
> Total Revenue	234,390,000	226,954,000	195,929,000	166,761,000	152,703,000
Cost of Revenue	206,105,000	199,382,000	170,684,000	144,939,000	132,886,000
Gross Profit	28,285,000	27,572,000	25,245,000	21,822,000	19,817,000
> Operating Expense	20,343,000	19,779,000	18,537,000	16,387,000	15,080,000
Operating Income	7,942,000	7,793,000	6,708,000	5,435,000	4,737,000
> Net Non Operating Interest Inc...	92,000	47,000	-28,000	-68,000	28,000
> Other Income Expense	62,000	106,000	56,000	7,000	27,000
Pretax Income	8,096,000	7,840,000	6,680,000	5,367,000	4,765,000
Tax Provision	2,016,000	1,925,000	1,601,000	1,308,000	1,061,000
> Net Income Common Stockhold...	6,051,000	5,844,000	5,007,000	4,002,000	3,659,000
Diluted NI Available to Com Stock...	6,051,000	5,844,000	5,007,000	4,002,000	3,659,000
Basic EPS	-	13.17	11.30	9.05	8.32
Diluted EPS	-	13.14	11.27	9.02	8.26
Basic Average Shares	-	443,651	443,089	442,297	439,755
Diluted Average Shares	-	444,757	444,346	443,901	442,923

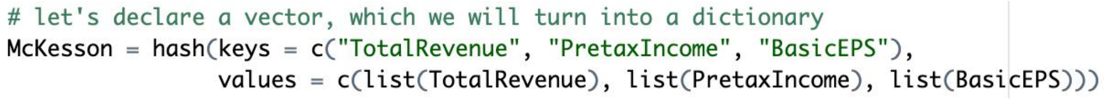
NON-TECHNICAL SUMMARY:

The project analysis involves importing libraries such as “ggplot2”, “parsedate”, “stringr”, “hash”, “tidyquant”, “lmtest” and “backtest” for data manipulation and visualizations. The data preprocessing stage involves extracting operating performance parameters, and populating list’s for Total Revenue, Pretax Income, Basic EPS, Operating Expenses, Operating Income, Cost of Revenue & Gross Profit. We also created Hash tables to store the financial indicators.



```r
# let's display all the operating parameters associated with each Company is being evaluated on
operating_performance_params = McKesson_Financials[,1:1]
operating_performance_params = c(str_c(operating_performance_params))
print("Operating Performance Parameters, selections of which we Evalate Per Company:")
for (i in 1:length(operating_performance_params)){
  # verify associated operating parameters for Business Operations and Performance Evaluations
  print(operating_performance_params[i])
}
```

We also created Hash tables to store the financial indicators. We then assigned the values from the hash table to respective variables for each company.



```r
# let's declare a vector, which we will turn into a dictionary
McKesson = hash(keys = c("TotalRevenue", "PretaxIncome", "BasicEPS"),
                values = c(list(TotalRevenue), list(PretaxIncome), list(BasicEPS)))
```

We then adjusted the data record years by adjusting the time series format for all the companies. This is very crucial to get accurate analysis for the financial data. When performing the exploratory analysis is gave us insights into the distribution and trends of the financial metrics for each company. We performed functions such as company records to reverse the order of the revenue, pretax income, basic EPS and other financial metrics to get the financial record years. We even updated the function the values in the “record_df” by removing commas, and converting the values to numeric.

For model fitting we performed plotting the time series data in order to look into it’s trends, and then performed manual models such as “SES” and “Holt’s linear Trend method” to fit the data and even the “Auto arima” models were fitted using AIC and BIC to select the best model to forecast. Then the residuals of the models were computed to check for randomness in the series, and also ljung - box test was also performed on the residuals to check their independence.

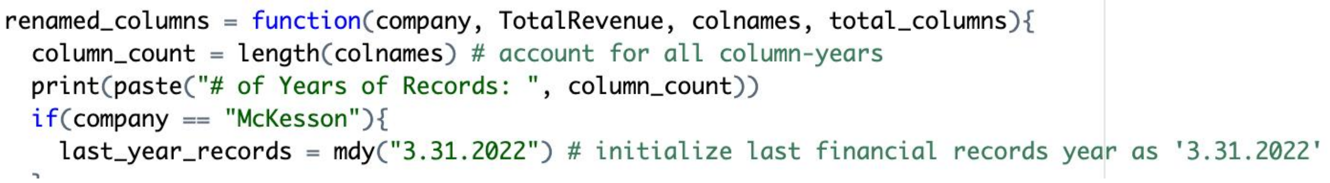
After performing the analysis, When looking at the Amerisource Bergen company, we saw that the best fit model for the companies Total Revenue time series is that “auto.arima” model with “AIC”. The model is ARIMA(0,2,1). The AIC is a statistical measure which is used to compare and choose between different models. A lower AIC value indicates a better fit. Therefore, the AIC was chosen as the best fit model here.

TECHNICAL SUMMARY:

For the analysis, we analyzed the financial performance of top 12 “Fortune 500” companies i.e., Walmart, Amazon, Apple, CVS, United Health Group, ExxonMobil Corp., BirkShire Heatherway, Alphabet Inc., McKesson, AmerisourceBergen, Microsoft and Costco. The following were performed in order to get to understand the companies even better.

EXPLORATORY ANALYSIS:

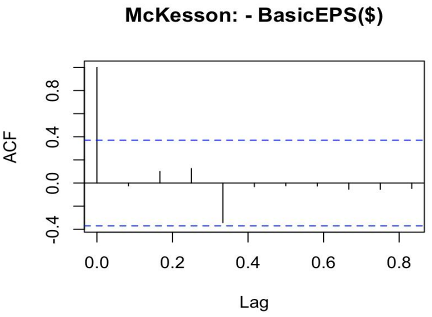
For the initial exploration for the financial data, we first extracted each companies financial indicators like “AmerisourceBergen_TotalRevenue”, “AmerisourceBergen_PretaxIncome”, “AmerisourceBergen_BasicEPS” similarly with the other companies to store in the values for later forecasting. Then we looked into adjusting and renaming the columns by converting them into actual dates.



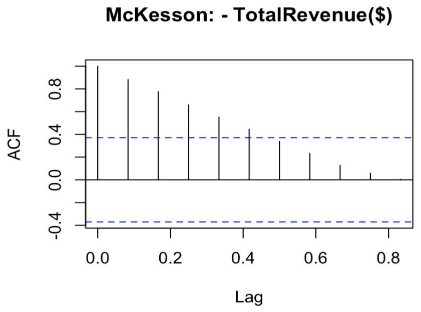
```
renamed_columns = function(company, TotalRevenue, colnames, total_columns){
  column_count = length(colnames) # account for all column-years
  print(paste("# of Years of Records: ", column_count))
  if(company == "McKesson"){
    last_year_records = mdy("3.31.2022") # initialize last financial records year as '3.31.2022'
```

Next we created a function called “company_stats” to calculate the basic statistics like mean and median etc.. for the financial metrics of the companies. Then created an other function “company_records” to combine the transformed financial record year and all the financial metrics into one data frame. In this we worked with conversions of data types, updating the column names and handling the missing values.

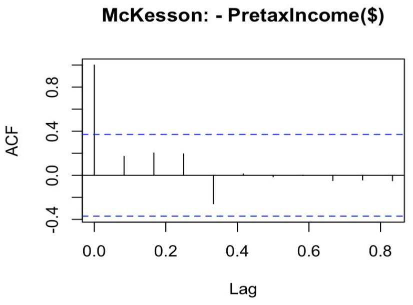
Then we looked into time series analysis. We created a function called “summary_TSAnalysis2” in which we added features like plotting the “auto correlation functions” and “partial correlation function” for the financial metrics. We also ran the “ARIMA” model to fit the data and to display the coefficients and significance of the model parameters. Lets look into some plots we retrieved.



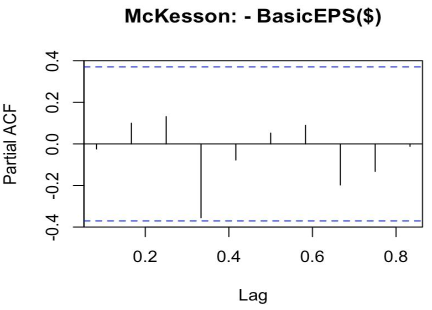
McKesson: - BasicEPS($)



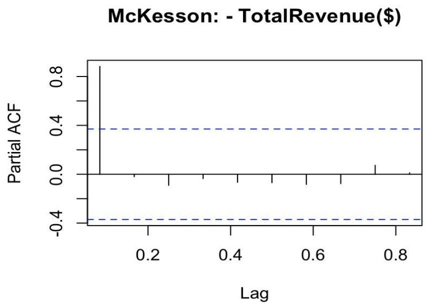
McKesson: - TotalRevenue($)



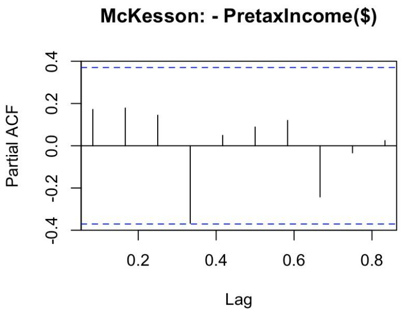
McKesson: - PretaxIncome($)



McKesson: - BasicEPS($)

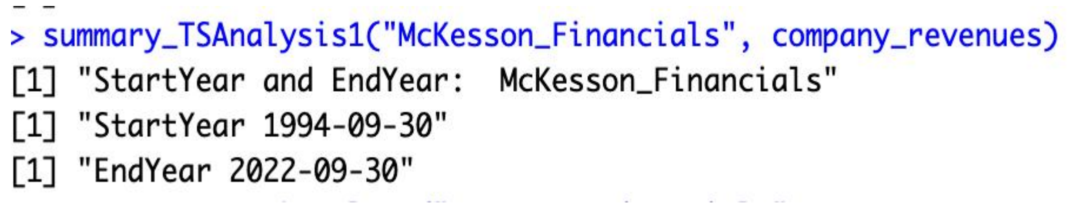


McKesson: - TotalRevenue($)



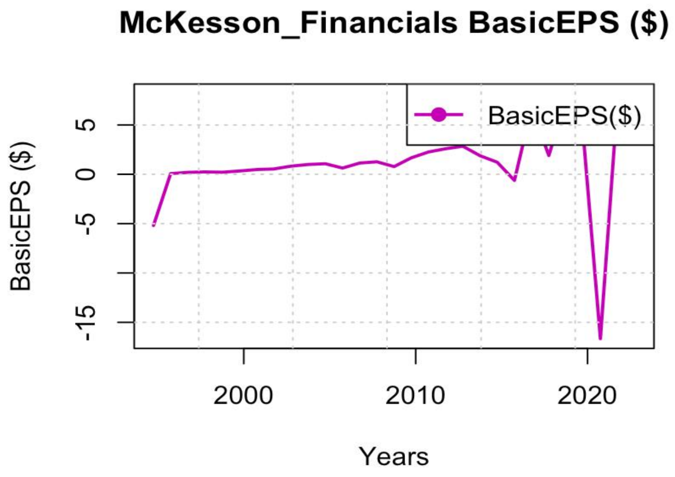
McKesson: - PretaxIncome($)

Then we created a function "summary_TSAnalysis1" to calculate the start and end years if the data records and perform time series on the financial indicators.



```
> summary_TSAnalysis1("McKesson_Financials", company_revenues)
[1] "StartYear and EndYear:  McKesson_Financials"
[1] "StartYear 1994-09-30"
[1] "EndYear 2022-09-30"
```

Then we plotted the time series for each financial indicators for the companies.



Then we looked into the correlations of the financial metrics by creating function "company_corr_plots" then created function "correlation_business_performance" to check for correlations between the tracked indicators by handling missing values, converting strings to numeric format and calling the company corr plots functions based on the companies.
When looking at the Microsoft company's correlation analysis of total revenue and pretax income, it exhibited a strong positive correlation, the same with total revenue and basic EPS. There is also a positive correlation between pretax income and basic EPS, suggesting us that the higher pretax income will correspond to the higher basic EPS.

After all the exploration, we finally retrieved the final financial record's with which we will be working ahead. Lets take a look at the companies correlations:

A) Mckesson's correlation :
Total revenue and Pretax income is "0.189" i.e, weak positive correlation.
Total revenue and Basic EPS is "0.010" i.e., negligible.
Pretax Income and Basic EPS is "0.953" i.e, strong positive correlation.

B) Microsoft:
Total revenue and pretax income is "0.967" i.e, strong positive correlation.
Total revenue and basic EPS is "0.9471" i.e, strong positive correlation.
Pretax income and basic EPS is "0.978" i.e, strong positive correlation.

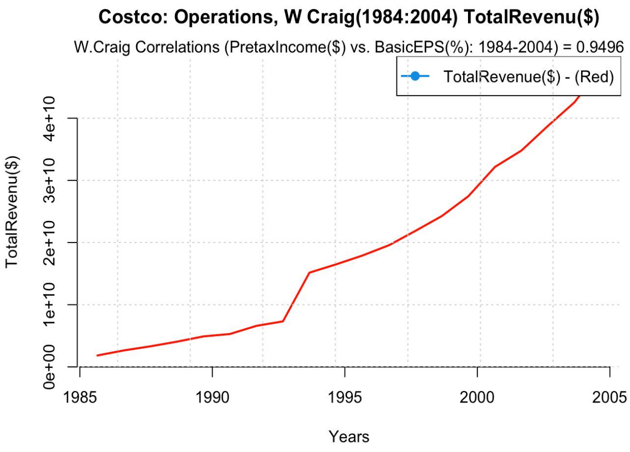
C) Costco:

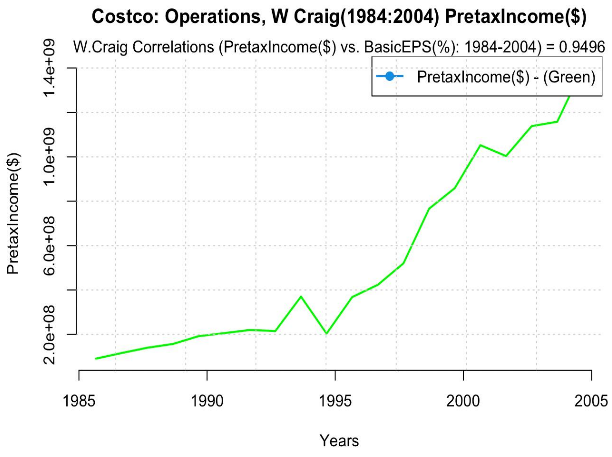
TotalRevenue and PretaxIncome Correlation is “0.967” i.e, strong positive correlation.
TotalRevenue and BasicEPS Correlation is “0.947” i.e, strong positive correlation.
PretaxIncome and BasicEPS Correlation is “0.978” i.e, strong positive correlation.

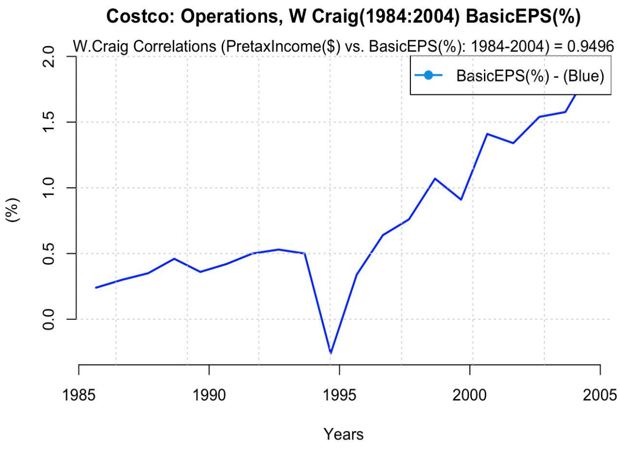
D) Amerisoucebergen:

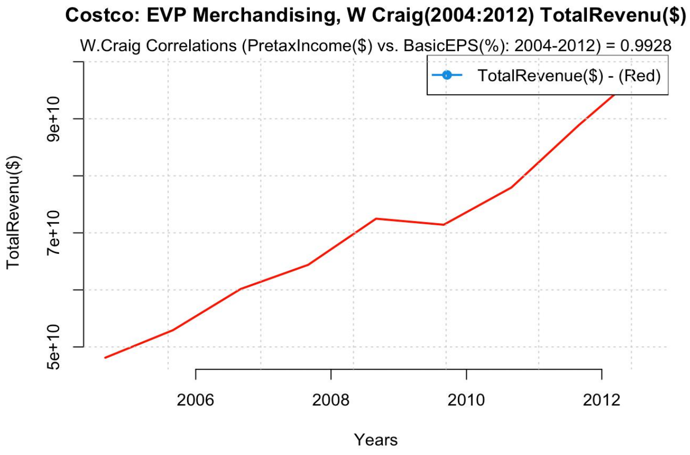
Total Revenue and Pretax income is “0.112” i.e, weak positive correlation.
Total Revenue and Basic EPS is “0.251” i.e, also weak positive correlation.
Pretax Income and Basic EPS is “0.945” i.e, strong positive correlation.

We then took a look at the Costco CEO W. Craig ’s tenure, to look at his career to see his performance. For that we created function “costco_records”. we separated the data in three 1984 - 2004, 2004 - 2012 and 2012 onwards. It extracts the total revenue,, pretax income and basic EPS of the company. And then calculated correlation coefficients between pretax income and basic EPS for the periods to determine the relationship between the metrics. Then we even looked into plots to visualize the trends of the metrics.



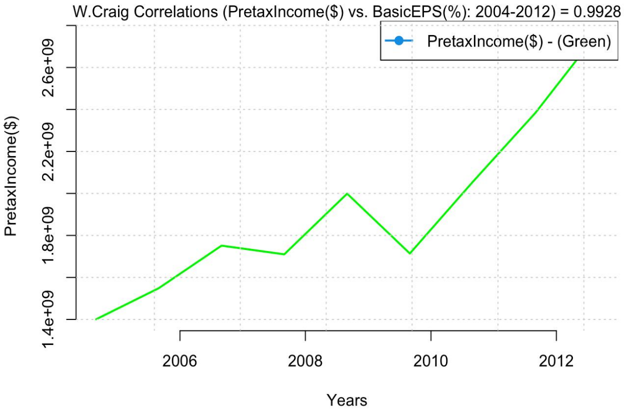






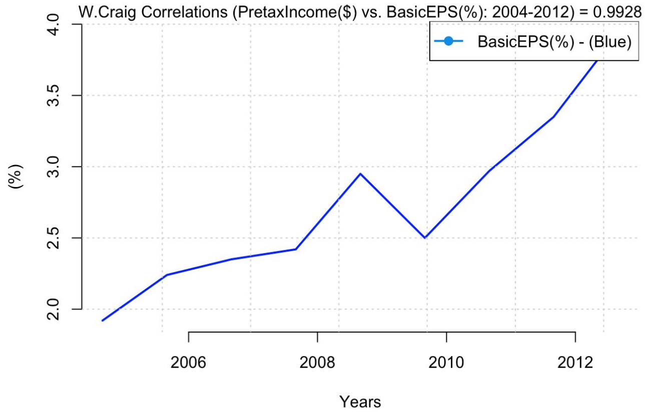
Costco: EVP Merchandising, W Craig(2004:2012) PretaxIncome($)

W.Craig Correlations (PretaxIncome($) vs. BasicEPS(%): 2004-2012) = 0.9928



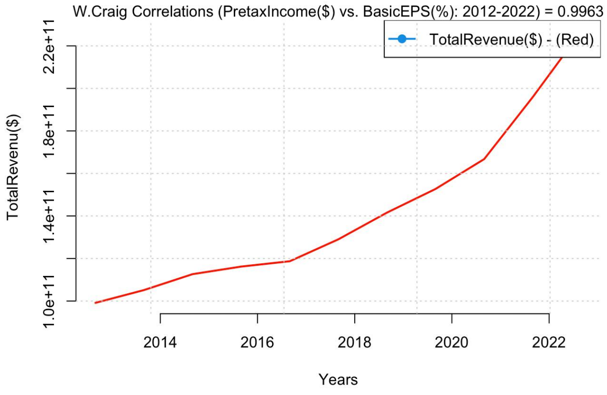
Costco: EVP Merchandising, W Craig(2004:2012) BasicEPS(%)

W.Craig Correlations (PretaxIncome($) vs. BasicEPS(%): 2004-2012) = 0.9928



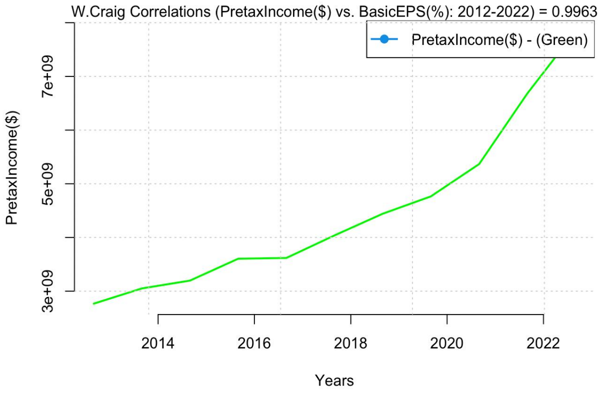
Costco: CEO, W Craig(2012:2022) TotalRevenu($)

W.Craig Correlations (PretaxIncome($) vs. BasicEPS(%): 2012-2022) = 0.9963



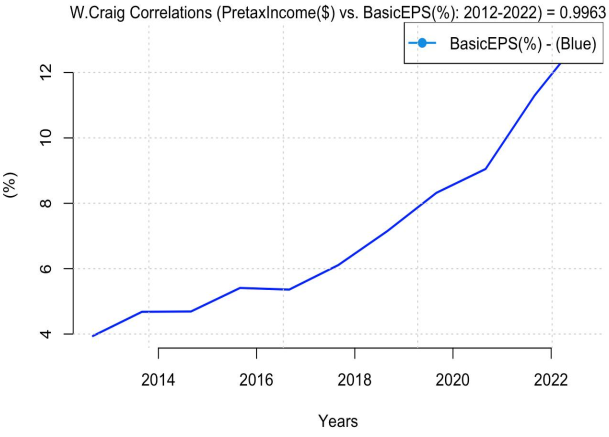
Costco: CEO, W Craig(2012:2022) PretaxIncome($)

W.Craig Correlations (PretaxIncome($) vs. BasicEPS(%): 2012-2022) = 0.9963



Costco: CEO, W Craig(2012:2022) BasicEPS(%)

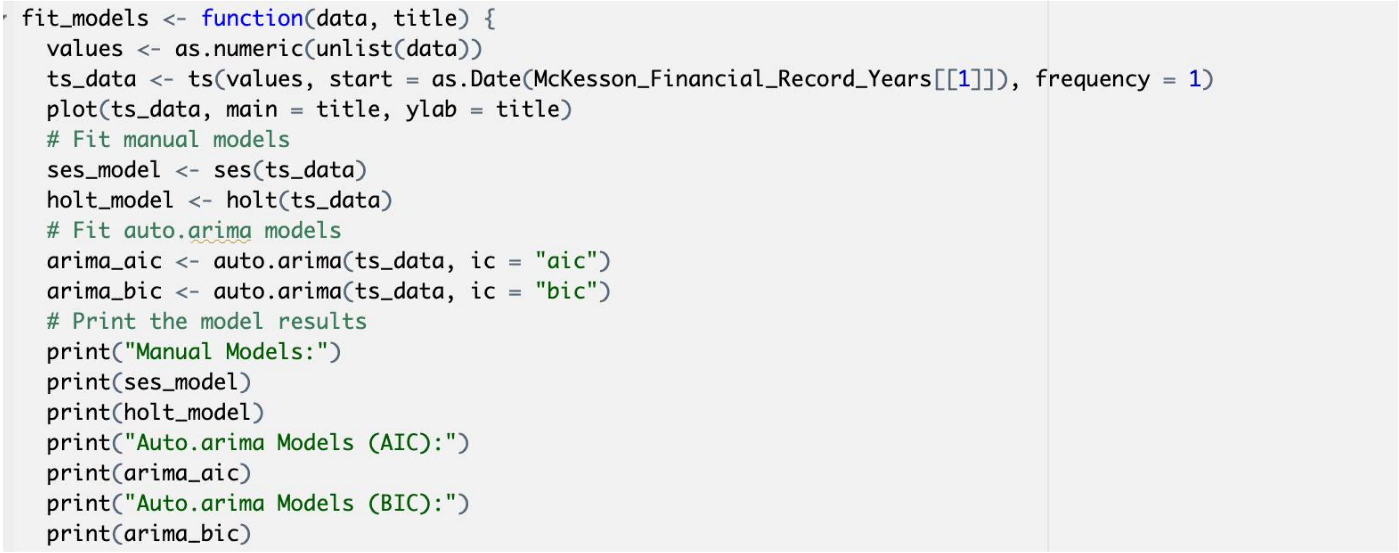
W.Craig Correlations (PretaxIncome($) vs. BasicEPS(%): 2012-2022) = 0.9963



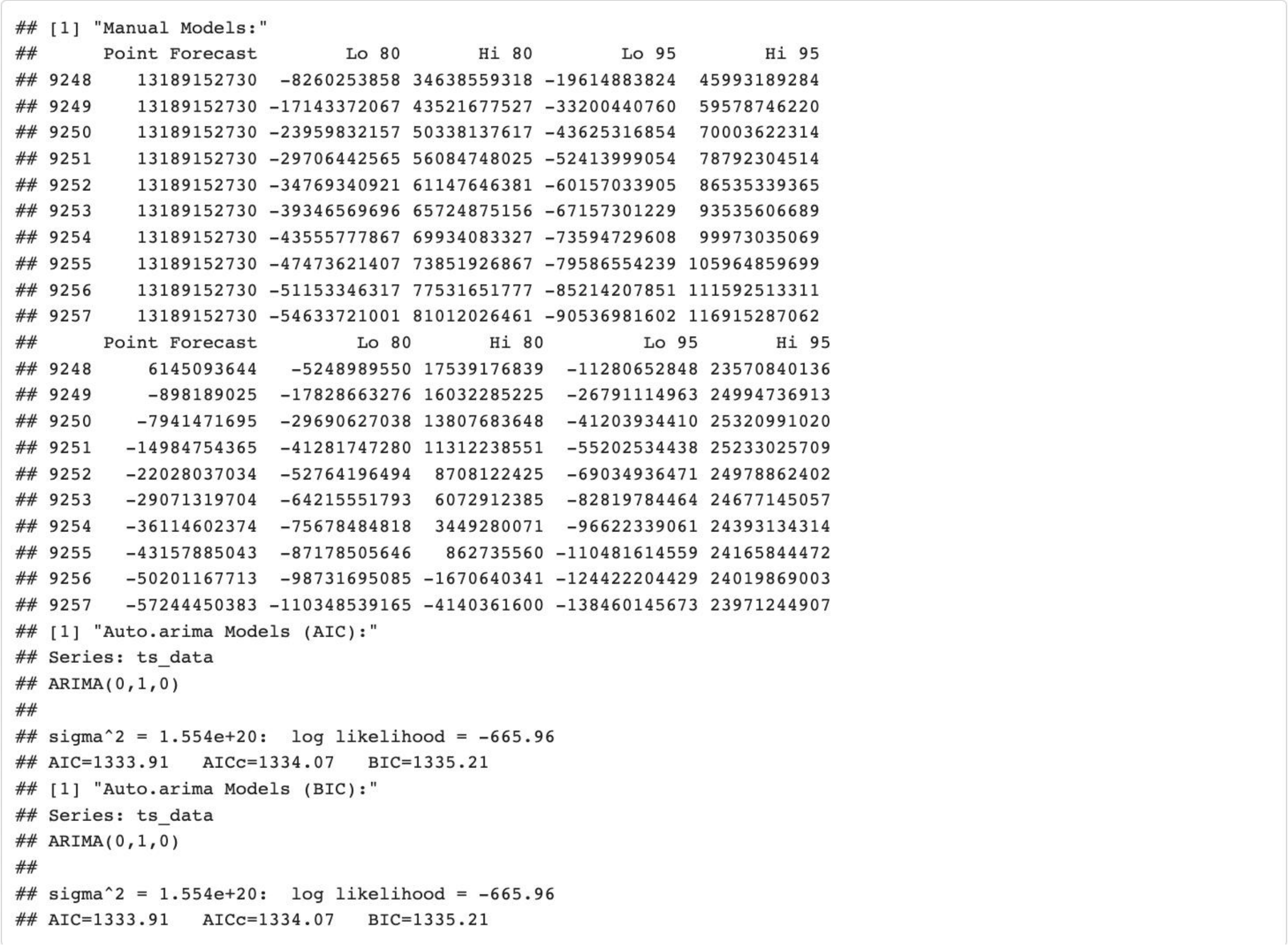
MODEL FITTING:

We then performed model fitting by defining a function called “fit_models” which performs model fitting, analysis and to visualize the data. We performed models like:

A) Simple exponential smoothing
B) Holts linear trend method
C) Auto ARIMA using AIC and BIC.



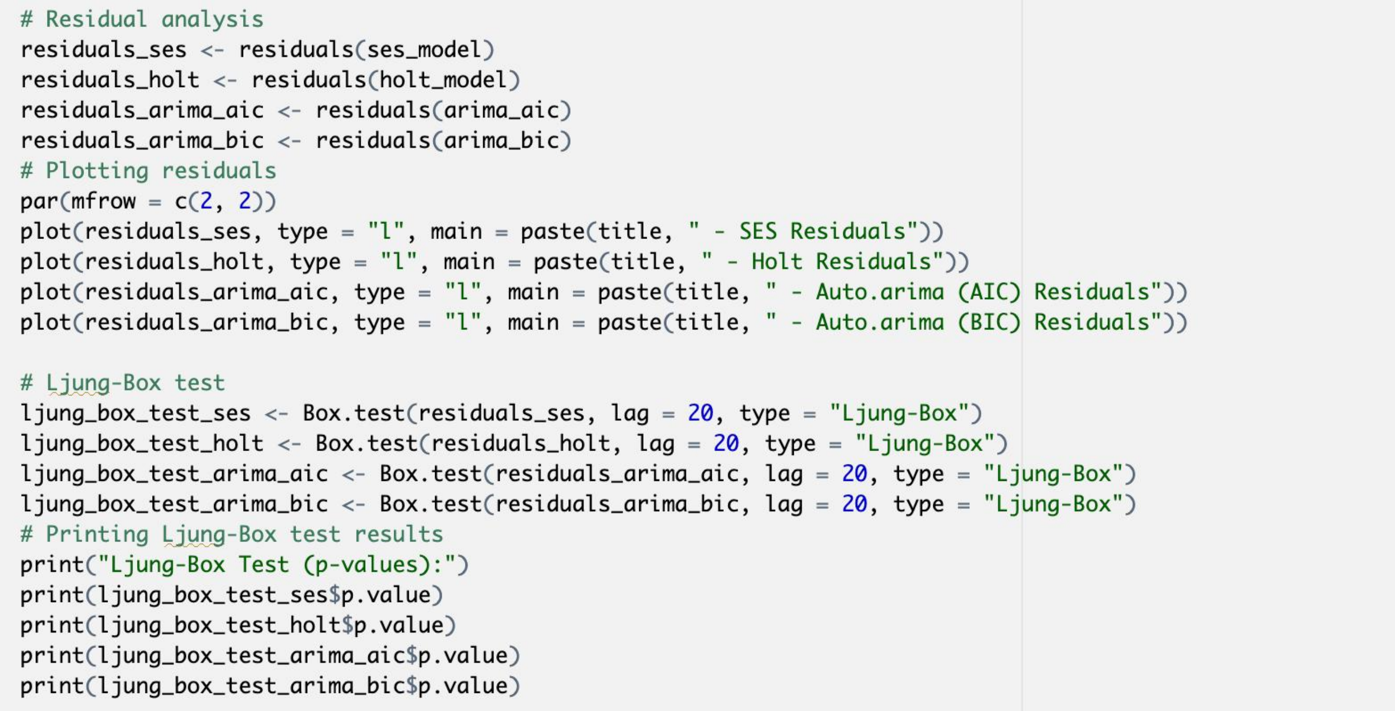
```r
fit_models <- function(data, title) {
  values <- as.numeric(unlist(data))
  ts_data <- ts(values, start = as.Date(McKesson_Financial_Record_Years[[1]]), frequency = 1)
  plot(ts_data, main = title, ylab = title)
  # Fit manual models
  ses_model <- ses(ts_data)
  holt_model <- holt(ts_data)
  # Fit auto.arima models
  arima_aic <- auto.arima(ts_data, ic = "aic")
  arima_bic <- auto.arima(ts_data, ic = "bic")
  # Print the model results
  print("Manual Models:")
  print(ses_model)
  print(holt_model)
  print("Auto.arima Models (AIC):")
  print(arima_aic)
  print("Auto.arima Models (BIC):")
  print(arima_bic)
```



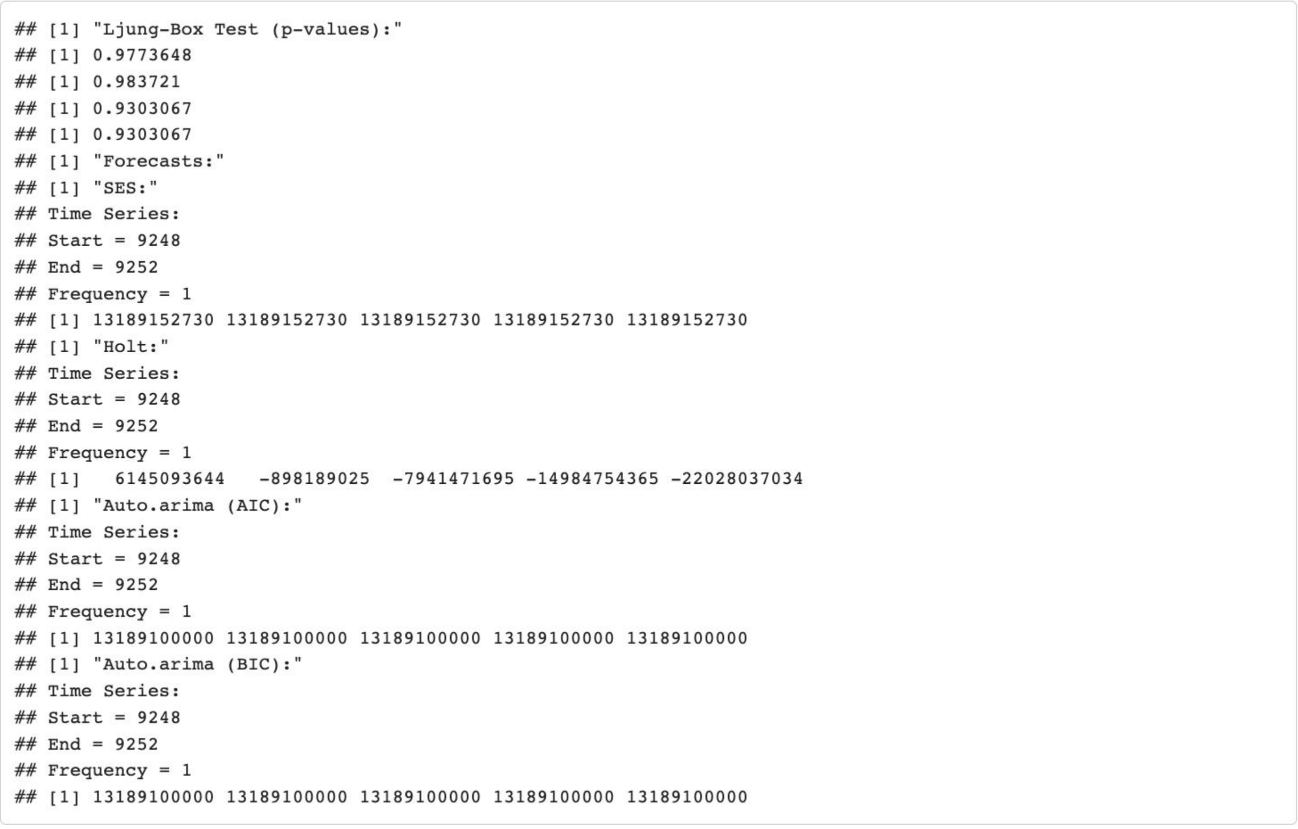
```
## [1] "Manual Models:"
##      Point Forecast        Lo 80       Hi 80        Lo 95        Hi 95
## 9248    13189152730  -8260253858 34638559318 -19614883824  45993189284
## 9249    13189152730 -17143372067 43521677527 -33200440760  59578746220
## 9250    13189152730 -23959832157 50338137617 -43625316854  70003622314
## 9251    13189152730 -29706442565 56084748025 -52413999054  78792304514
## 9252    13189152730 -34769340921 61147646381 -60157033905  86535339365
## 9253    13189152730 -39346569696 65724875156 -67157301229  93535606689
## 9254    13189152730 -43555777867 69934083327 -73594729608  99973035069
## 9255    13189152730 -47473621407 73851926867 -79586554239 105964859699
## 9256    13189152730 -51153346317 77531651777 -85214207851 111592513311
## 9257    13189152730 -54633721001 81012026461 -90536981602 116915287062
##      Point Forecast          Lo 80       Hi 80          Lo 95       Hi 95
## 9248     6145093644   -5248989550 17539176839  -11280652848 23570840136
## 9249     -898189025  -17828663276 16032285225  -26791114963 24994736913
## 9250    -7941471695  -29690627038 13807683648  -41203934410 25320991020
## 9251   -14984754365  -41281747280 11312238551  -55202534438 25233025709
## 9252   -22028037034  -52764196494  8708122425  -69034936471 24978862402
## 9253   -29071319704  -64215551793  6072912385  -82819784464 24677145057
## 9254   -36114602374  -75678484818  3449280071  -96622339061 24393134314
## 9255   -43157885043  -87178505646   862735560 -110481614559 24165844472
## 9256   -50201167713  -98731695085 -1670640341 -124422204429 24019869003
## 9257   -57244450383 -110348539165 -4140361600 -138460145673 23971244907
## [1] "Auto.arima Models (AIC):"
## Series: ts_data
## ARIMA(0,1,0)
##
## sigma^2 = 1.554e+20:  log likelihood = -665.96
## AIC=1333.91   AICc=1334.07   BIC=1335.21
## [1] "Auto.arima Models (BIC):"
## Series: ts_data
## ARIMA(0,1,0)
##
## sigma^2 = 1.554e+20:  log likelihood = -665.96
## AIC=1333.91   AICc=1334.07   BIC=1335.21
```

RESIDUAL ANALYSIS AND MODEL DIAGNOSTICS:

After fitting the models, we then performed residual analysis by computing the residuals for each model and plotting by using “plot()” and then conduct ljungbox test for residuals using the “Box.test()” .



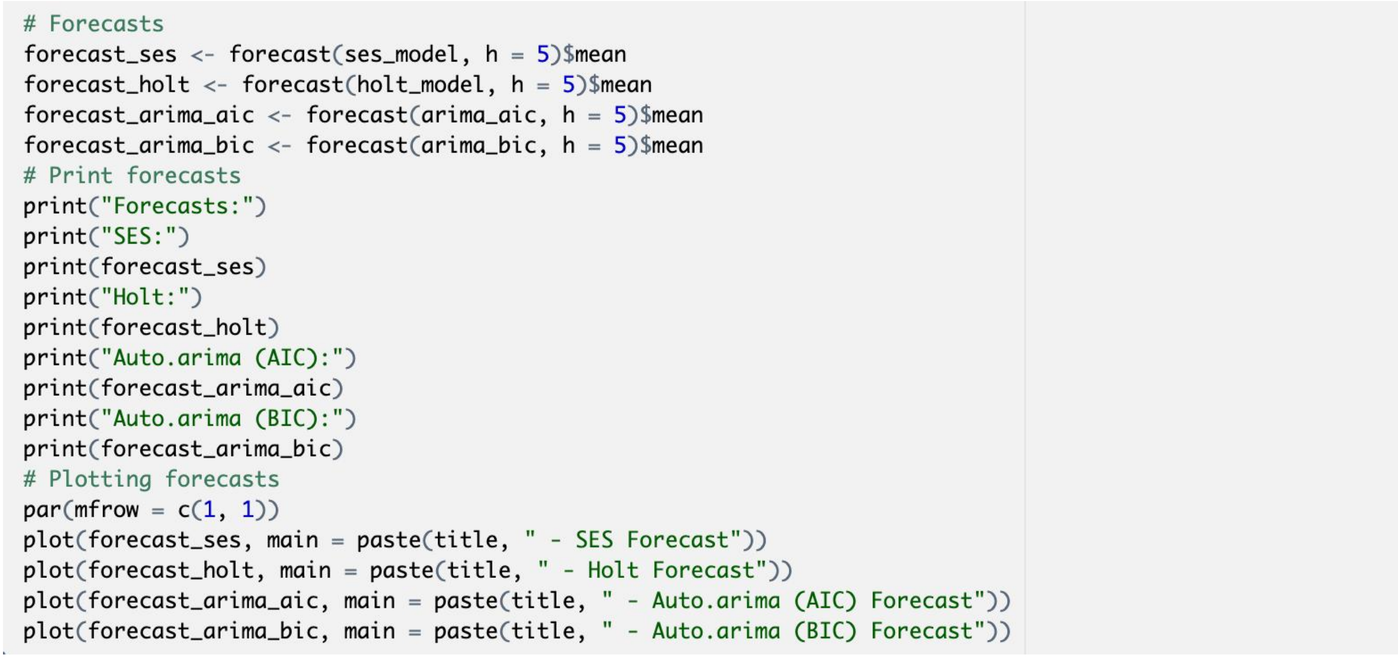
```r
# Residual analysis
residuals_ses <- residuals(ses_model)
residuals_holt <- residuals(holt_model)
residuals_arima_aic <- residuals(arima_aic)
residuals_arima_bic <- residuals(arima_bic)
# Plotting residuals
par(mfrow = c(2, 2))
plot(residuals_ses, type = "l", main = paste(title, " - SES Residuals"))
plot(residuals_holt, type = "l", main = paste(title, " - Holt Residuals"))
plot(residuals_arima_aic, type = "l", main = paste(title, " - Auto.arima (AIC) Residuals"))
plot(residuals_arima_bic, type = "l", main = paste(title, " - Auto.arima (BIC) Residuals"))

# Ljung-Box test
ljung_box_test_ses <- Box.test(residuals_ses, lag = 20, type = "Ljung-Box")
ljung_box_test_holt <- Box.test(residuals_holt, lag = 20, type = "Ljung-Box")
ljung_box_test_arima_aic <- Box.test(residuals_arima_aic, lag = 20, type = "Ljung-Box")
ljung_box_test_arima_bic <- Box.test(residuals_arima_bic, lag = 20, type = "Ljung-Box")
# Printing Ljung-Box test results
print("Ljung-Box Test (p-values):")
print(ljung_box_test_ses$p.value)
print(ljung_box_test_holt$p.value)
print(ljung_box_test_arima_aic$p.value)
print(ljung_box_test_arima_bic$p.value)
```



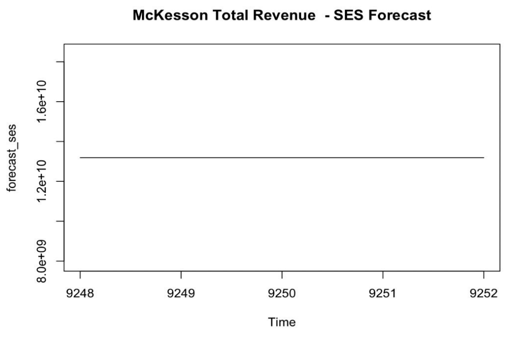
```
## [1] "Ljung-Box Test (p-values):"
## [1] 0.9773648
## [1] 0.983721
## [1] 0.9303067
## [1] 0.9303067
## [1] "Forecasts:"
## [1] "SES:"
## Time Series:
## Start = 9248
## End = 9252
## Frequency = 1
## [1] 13189152730 13189152730 13189152730 13189152730 13189152730
## [1] "Holt:"
## Time Series:
## Start = 9248
## End = 9252
## Frequency = 1
## [1]   6145093644   -898189025   -7941471695 -14984754365 -22028037034
## [1] "Auto.arima (AIC):"
## Time Series:
## Start = 9248
## End = 9252
## Frequency = 1
## [1] 13189100000 13189100000 13189100000 13189100000 13189100000
## [1] "Auto.arima (BIC):"
## Time Series:
## Start = 9248
## End = 9252
## Frequency = 1
## [1] 13189100000 13189100000 13189100000 13189100000 13189100000
```

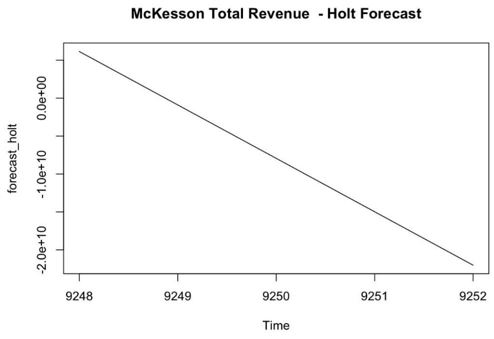
FORECAST ANALYSIS:

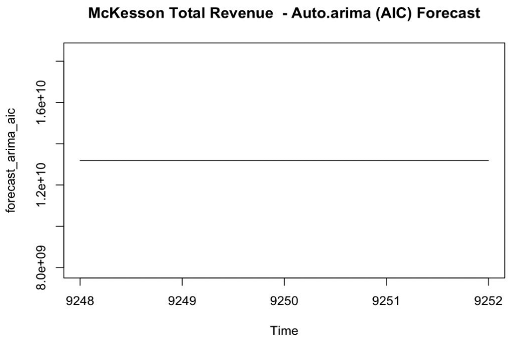
Then we will generate the forecasts for each model using “forecast()”.

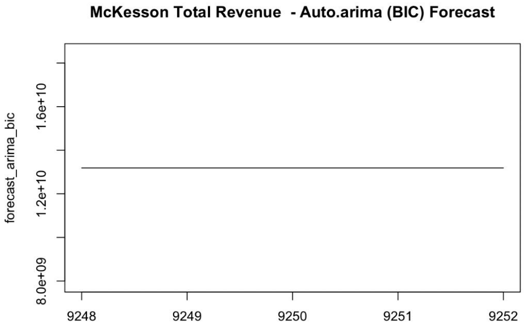


```
# Forecasts
forecast_ses <- forecast(ses_model, h = 5)$mean
forecast_holt <- forecast(holt_model, h = 5)$mean
forecast_arima_aic <- forecast(arima_aic, h = 5)$mean
forecast_arima_bic <- forecast(arima_bic, h = 5)$mean
# Print forecasts
print("Forecasts:")
print("SES:")
print(forecast_ses)
print("Holt:")
print(forecast_holt)
print("Auto.arima (AIC):")
print(forecast_arima_aic)
print("Auto.arima (BIC):")
print(forecast_arima_bic)
# Plotting forecasts
par(mfrow = c(1, 1))
plot(forecast_ses, main = paste(title, " - SES Forecast"))
plot(forecast_holt, main = paste(title, " - Holt Forecast"))
plot(forecast_arima_aic, main = paste(title, " - Auto.arima (AIC) Forecast"))
plot(forecast_arima_bic, main = paste(title, " - Auto.arima (BIC) Forecast"))
```









INDIVIDUAL ANALYSIS

LOUIS NEWMAN:

Since the project began, I have been instrumental in discussing the topic we chose and how to run data in order to run our analysis in a way that would be a good fit for the project. We met a few times over video chat and have corresponded via Slack. Given that we decided not to use pre-existing datasets, along with the fact that our project is financial related, I proposed using Yahoo Finance premium to download historical data for the top 12 Fortune 500 companies in the United States.

We then divided up the work so that each of us chose 4 companies to evaluate. I registered and paid for a Yahoo Finance premium account so that I could download the historical financial data and disburse to my teammates. I then cleaned up the date prior to sending out to the team to aid in the analysis. This involved renaming columns and transposing the data in a way that would be favorable to our analysis in R. Since then, I have added to our ongoing conversations as to which metrics to use to evaluate. I suggested a few that I believe might be instrumental in evaluating CEO performance of the Top 12 Fortune 500 companies. The team agreed so then we knew what metrics to evaluate. I also proposed that we look at the individual CEO metrics of their prior roles to determine their impact on the overall company financials, not only their current roles. This would allow us to determine impact of the CEO from other outside factors. Arte wrote the initial code in which I was assisting him when he had questions. I then worked with his code on my four companies. We are all working together to finalize the presentation and also the time series analysis.

Now that we have finished our project, I have learned a lot about some of the time series packages that were presented in the class. While I used them for the homework, it stretched my coding skills to have to tailor the code to our project. The datasets were also not completely the same which required me to do a lot of data cleaning. From a positive standpoint, this has given me more confidence to complete additional time series tasks on my own. There is a project that I would like to work on at my job which I would like to try to use the skills I learned in the class. I am a Sales Analyst for a large medical supply manufacturer/distributor and I am planning on running an analysis which would be used to evaluate trends among our largest and most profitable customers to determine if there are products that would be better alternatives for the less profitable customers.

ARTE MOALIN:

So, given the fact that everyone on our team of three is an adult with personal, professional, and academic lives, we coordinated our strategy into a set of three, simply due to the enormity of the companies we wanted to study, and the volume of Data sets involved. So,
a) Lewis Newman was assigned the task of Data Harvesting,
b) I, Arte Moalin, was assigned with writing the underling main code for the project, performing Data Wrangling, Exploration, Visualization, and providing initial test staging tests for all of our 12 Fortune 500 entities' time-series analysis, making sure that, we would, each, be able to individually complete the remaining in-depth time-series analysis for each of us, on our assigned 4 Companies out of 12 Fortune 500 Companies.

c) Harnain Kaur Sardani was assigned as our liaison with the Professor for guiding us throughout the term on the project, as well as the person responsible for putting all our works together and submitting in all our work along the way. And to accomplish our goal to understand, explore and analyze how the top 12 Fortune 500 companies operate, given that there are so many variables involved in running a successful business, in addition to assigning each of us a set of companies, we also assigned each of us 3 business parameters to study and analyze, so that we wouldn't be limiting ourselves to the same, repetitive parameters across all of our 12 Fortune 500 Companies. As such,

a) Lewis Newman, was assigned: 1-Walmart, 2- Amazon, 3- Apple, 4- CVS, with the business parameters 1-TotalRevenue, 2- OperatingExpenses & 3- OperatingIncome;
b) Harnain Kaur Sardani was assigned: 5- McKesson, 6-Microsoft, 7-Costco, 8-AmerisourceBergen, with the business parameters 1-TotalRevenue, 2- PretaxIncome & 3- BasicEPS;
c) I was assigned: 9-United Health Group, 10-ExxonMobil, 11-Birkshire Heather way, and 12-Alphabet, with the business parameters 1-TotalRevenue, 2-CostOfRevenue & 3-GrossProfit.

Therefore, my role was focused on writing the underling code that we built to wrangle, explore and visualize the Data for all of our 12 Fortune 500 companies, and for each company, I evaluated the given business performance indicators against our selections of business performance metrics, and in doing so, I did the time-series graphs for each sets of three business performance indicators, collectively per each entity, and against each other within entity, thus, relating each to the other and all for each company, with correlations, based, how, for instance: TotalRevenue($) related to OperatingExpenses($) and OperatingIncome($) for Lous' set of 4 companies; TotalRevenue($) related to PretaxIncome($) and BasicEPS(%Share$) for Harnain's set of 4 companies; and finally, TotalRevenue($) related to CostOfRevenue($) and GrossProfit($) for my set of 4 companies. Then, I run the auto.arima() for every three performance indicators on every one of the 12 Fortune 500 companies, so that, when we each got to the in-depth time series analysis four respective selected set of 4 companies, we'd validate the correlation permutations we saw in our exploration and visualization stages.

Finally, I run a trace code on each of our selected, single assigned CEO, for each of us, on his entire associated tenure, that evaluated his current performance as a CEO against his past business experiences, looking for correlations and similarities between his previous performance and his current role, as a CEO, that would provide a hint of a trend, a criteria of a selectin in other words, on why and how those who make CEO on this list of Fortune 500 companies, are filtered for their current roles as CEO's. In addition, I was always available to help with extended coding support, as well as help with all further time-series analysis.

As for a as what I have learned throughout the course, for one, I have gained a lot of experience using R for time-series based Data Science Applications, two, the importance of autocorrelations, moving averages as well as GARCH residual analysis on financial commodities, and the validity in the general application of the Efficient Market Hypothesis that, and because of it, a lot of financial time series analysis has no autocorrelations.

HARNAIN KAUR SARDARNI:

For this project my work involved on performing financial analysis and time series modeling for multiple companies.We worked on importing libraries, manipulating data, visualizing them and analyzing them. The companies i worked for were McKesson, AmerisourceBergen, Microsoft and Costco. We focused on extracting operating performance parameters, populating the financial indicators lists, and conducting auto correlation analysis. Hash() was used to store the lists of financial indicators for each company. The analysis was performed iteratively for each company, and the extracted values for the "Total Revenue" indicator were printed.

Several functions were defined to handle different aspects of the analysis, including processing financial records, conducting auto correlation analysis, calculating the correlations, and visualizing the data. Descriptive statistics were also provided for the financial metrics. The code iterated over each company and performed various steps, such as renaming columns, combining financial record years and financials, displaying current business performance, and creating time series plots. The analysis focused on understanding the correlations between different financial indicators and forecasting future values using time series models. Overall, the work involved data preparation, analysis, visualization, and forecasting for multiple companies' financial data.

REFERNCES:

1. https://en.wikipedia.org/wiki/List_of_largest_companies_in_the_United_States_by_revenue
2. https://finance.yahoo.com/quote/COST/financials?p=COST
3. https://rpubs.com/tedding/long-term-time-series-forecasting
4. https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/
5. https://www.dominodatalab.com/blog/time-series-with-r