ALY6050 Introduction to Enterprise Analytics

Dr. Michael Prokle

Module 3

Time Series Analysis

Jeff Hackmeister

7/27/2025

Contents

Introduction

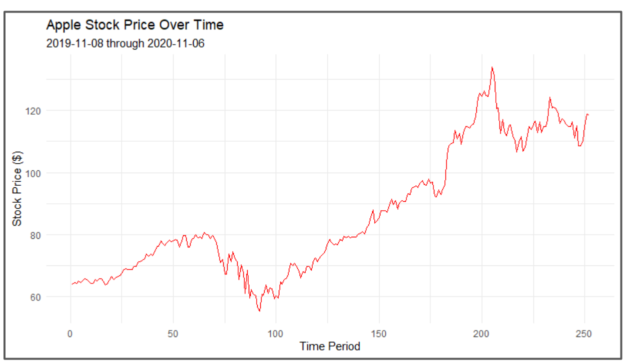
Predicting stock performance is crucial for investors and financial analysts. This study will center around historical stock prices for Apple Inc (AAPL) and Honeywell International Inc (HON). Using data provided for the study, short and long-term forecasting methods will be deployed to predict future performance of both stocks. Statistical tests will be performed on each forecast to ensure appropriate use of techniques and to determine the optimal variable values to maximize the effectiveness of the models.

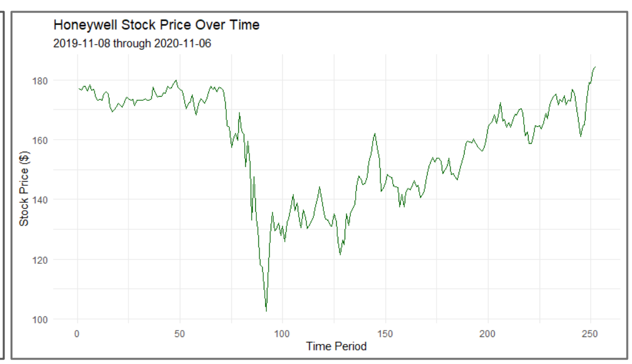
The dataset provided for this project contained the closing stock price and total volume for each company from November 8th, 2019, through November 11th, 2020. Each observation represents one trading day and is represented both by the date and a numerical period value which ranged from 1 to 252. The first steps of the project involved loading the data into R Studio, along with necessary packages for analysis. The data was also split into unique datasets for each company.

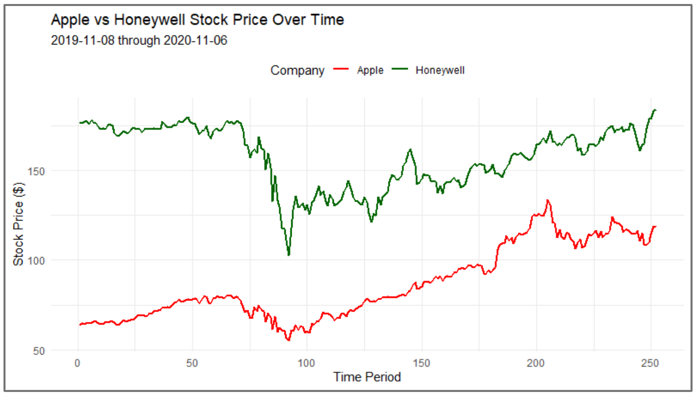
Part 1: Short-Term Forecasting

i. Plotting the data

To begin the analysis, plots were created to show movement of each company's stock price over the study period. The first plots were of the companies separately, and then a combined plot







In these plots, the effect of the Covid-19 pandemic is clear as both stock prices plunge in March of 2020. In the case of Honeywell, it takes nearly the entire study period to recover back to the late 2019 values. With Apple, by period 125 (May 8th, 2020) March losses are recovered, and a long positive trend is observed going forward.

ii. Exponential Smoothing

To produce forecasted values for the next period in the study, period 253, an exponential smoothing function will be used. Successive values for alpha will be deployed (0.15, 0.35, 0.55, and 0.75) and each result will be evaluated based on the resulting MAPE (Mean Absolute Percentage Error).

First, the exponential smoothing function is defined, utilizing the Holt-Winters

```
exp_smoothing <- function(data, alpha) {
  model <- HoltWinters(data, alpha = alpha, beta = FALSE, gamma = FALSE)
  forecast <- forecast(model, h = 1)
```

This function will also calculate the MAPE of each resulting function.

```
mape <- mean(abs((data - forecast$fitted) / data), na.rm = TRUE) * 100
return(list(forecast = forecast, mape = mape))
```

With the function defined, the list of alpha values can be passed through the function and applied to each company's dataset.

```
appl_exsmo <- lapply(c(0.15, 0.35, 0.55, 0.75), function(alpha) exp_smoothing(appl, alpha))
hon_exsmo <- lapply(c(0.15, 0.35, 0.55, 0.75), function(alpha) exp_smoothing(hon, alpha))

sapply(appl_exsmo, function(x) x$mape)
sapply(hon_exsmo, function(x) x$mape)
```

With the calculations completed, the MAPE values for each can be evaluated.

```
> hon_mapes
[1] 3.061768 2.349148 2.032281 1.895824
> appl_mapes
[1] 3.874227 2.468633 2.082012 1.969072
```

For both companies, the best performance came from the fourth alpha value considered, which was 0.75. In both examples, an alpha of 0.75 resulted in the smallest MAPE value, indicating the smallest deviation from observed values.

iii. Adjusted Exponential Smoothing

To increase the predictive capabilities of the model, the next step is to introduce adjusted exponential smoothing techniques. For this, alpha will be set at 0.55 and the trend (or beta) will be tested at values of 0.15, 0.25, 0.45, and 0.85. This technique is best deployed when trends are evident in the data and the initial plots do indicate a trend, particularly in the Apple prices.

As with the simple exponential smoothing, a function will be designed to implement the adjusted exponential smoothing calculation along with the MAPE values.

```
adj_exp_smoothing <- function(data, alpha, beta) {
  model <- HoltWinters(data, alpha = alpha, beta = beta, gamma = FALSE)
  forecast <- forecast(model, h = 1)
  mape <- mean(abs((data - forecast$fitted) / data), na.rm = TRUE) * 100
  return(list(forecast = forecast, mape = mape))
}
```

Then, the dataset for each company will be passed through the function for each of the desired beta values.

```
appl_adj_exsmo <- lapply(c(0.15, 0.25, 0.45, 0.85), function(beta) adj_exp_smoothing(appl, 0.55, beta))
hon_adj_exsmo <-  lapply(c(0.15, 0.25, 0.45, 0.85), function(beta) adj_exp_smoothing(hon, 0.55, beta))
```

```
> sapply(appl_adj_exsmo, function(x) x$mape)
[1] 2.026042 2.048863 2.170707 2.303869
> sapply(hon_adj_exsmo, function(x) x$mape)
[1] 2.063393 2.084121 2.167818 2.276372
```

With these calculations complete, the final forecasts for period 253 can be compared.

	Exponential Smoothing (α=0.75)	Adj. Exponential Smoothing (β=0.15)
Apple	118.39	118.08
Honeywell	183.72	185.14

Part 2: Long-Term Forecasting

i. 3-Period Weighted Average

For the next stage of the analysis, a 3-period weighted average will be used to forecast values for the first 100 periods. Then weights will be added for each period at 0.5 for the most recent, 0.3 for the next most recent, and 0.2 for two periods ago. Then, the observed value from period 100 be used as the base for a linear trend which will be used to predict values for period 101-257.

To start, a function is defined to perform the 3-period weighted average

```
weighted_moving_average <- function(data, weights) {
  n <- length(data)
  forecast <- rep(NA, n)  # Initialize with NA
  weights <- weights / sum(weights)  # Normalize weights to sum to 1

  if (length(weights) != 3) {
    stop("Weights vector must have exactly 3 elements.")
  }

  for (i in 3:n) {
    forecast[i] <- sum(data[(i-2):i] * weights)
  }
  return(forecast)
}
```

Next, the weights are defined, and the function is applied to each company's dataset for the first 100 periods.

```
# Define weights
weights <- c(0.2, 0.3, 0.5)

# Apply 3-period weighted moving average for AAPL and HON data (first 100 periods)
appl_wma <- weighted_moving_average(appl[1:100], weights)
hon_wma <- weighted_moving_average(hon[1:100], weights)
```

With this calculated, the linear model can be fit for the remaining periods.

```
appl_trend <- lm(Value ~ Period, data = appl_train)

hon_trend <- lm(Value ~ Period, data = hon_train)
```

And predictions can be made.

```
> appl_forecasts_253_257
     253      254      255      256      257 
128.0162 128.4151 128.8139 129.2128 129.6117 
> hon_forecasts_253_257
     253      254      255      256      257 
176.0496 176.3539 176.6582 176.9625 177.2668 
```

And these predictions can be compared with the actual closing values

Date	Period	APPL Forecast	APPL Actual	HON Forecast	HON Actual
11/9/2020	253	128.02	116.32	176.05	196.99
11/10/2020	254	128.42	115.97	176.35	201.98
11/11/2020	255	128.81	119.49	176.66	199.29
11/12/2020	256	129.21	119.21	176.96	197.24
11/13/2020	257	129.61	119.26	177.27	201.54

The forecast for periods 101-257 can be found in the appendix.

ii. Mean Absolute Percentage Error (MAPE)

To evaluate the performance of this model, the MAPE is calculated for each company.

```
> # Display MAPE results for long-term forecasts
> appl_long_mape
[1] 4.778476
> hon_long_mape
[1] 3.052453
```

In both cases, these MAPE values for the 3-period weighted average are higher than those seen in the exponential smoothing or adjusted exponential smoothing forecast. This is a foreseeable outcome as the starting point of the linear regression was in the depths of the Covid market crash, a period that was a historic outlier to nearly all publicly traded companies.

Part 3: Regression

i. Simple Regression

For further comparison, a simple regression model is fit for each company and the predictions are made to period 257.

```
# Perform simple regression for AAPL
appl_regression <- lm(`AAPL (Apple Inc) / $` ~ Period, data = data)
appl_regression_forecast <- predict(appl_regression, newdata = data.frame(Period = 1:257))

# Perform simple regression for HON
hon_regression <- lm(`HON (Honeywell Inc)  /  $` ~ Period, data = data)
hon_regression_forecast <- predict(hon_regression, newdata = data.frame(Period = 1:257))
```

With the predictions made, the MAPE can be evaluate for both companies.

```
> # Display MAPE results for regression forecasts
> appl_regression_mape
[1] 10.56076
> hon_regression_mape
[1] 9.758568
>
```

Company	Exponential Smoothing	Adj Exponential Smoothing	3-Period Weighted Average	Simple Regression
Apple	1.9691	2.0260	4.7785	10.5608
Honeywell	1.8958	2.0634	3.5245	9.7586

Predictably, the linear regression model has the worst performance of all techniques. This technique has no ability to measure the complicated nuances seen in the performance data.

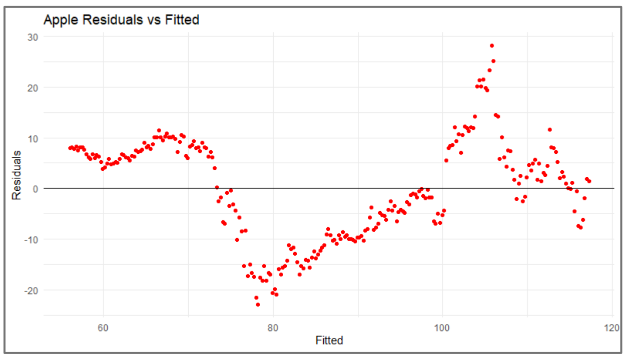
ii. Appropriateness of Regression Model

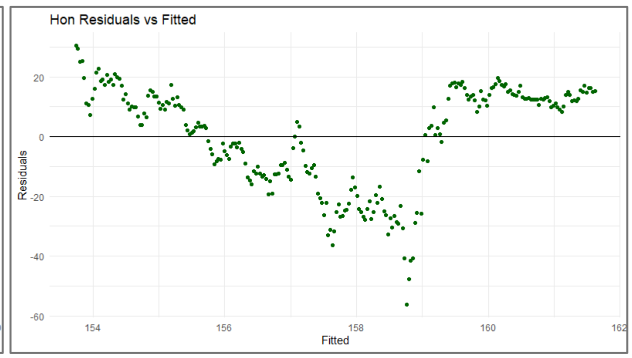
To further evaluate the use of the regression model, tests will be conducted to determine if the

- The residuals are independent
- The residuals are homoscedastic
- The residuals are normally distributed (both through a Q-Q plot and a statistical test)

Independent residuals

To test for the independence of the residuals, they are plotted against the fitted values. If the residuals are independent, there would be not discernible pattern in the plot.





In this case, there are clear patterns for both Apple and Honeywell in the plot.

Homoscedasticity

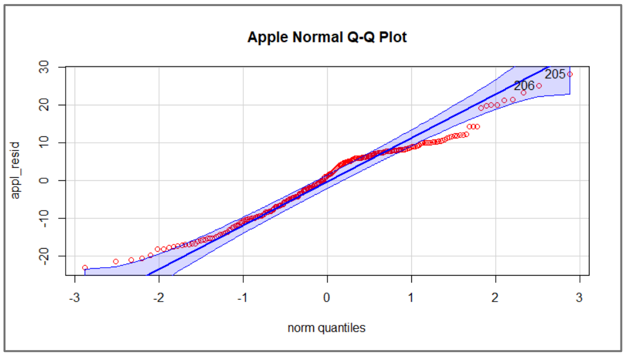
To test for homoscedasticity, the ncvTest is deployed. This uses the Breusch-Pagen test for non-constant variance in the residuals of the liner regression model.

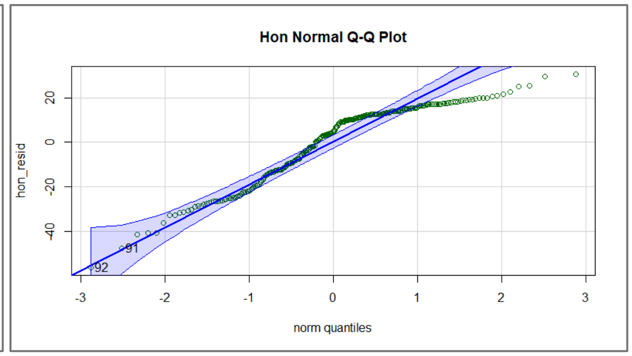
```
> ncvTest(appl_regression)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.0001407281, Df = 1, p = 0.99054
> ncvTest(hon_regression)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 1.241827, Df = 1, p = 0.26512
```

While both results display p-values greater than 0.05, the Apple results are particularly strong. In both cases, the null hypothesis of constant variance in the residuals is accepted.

Normal Distribution

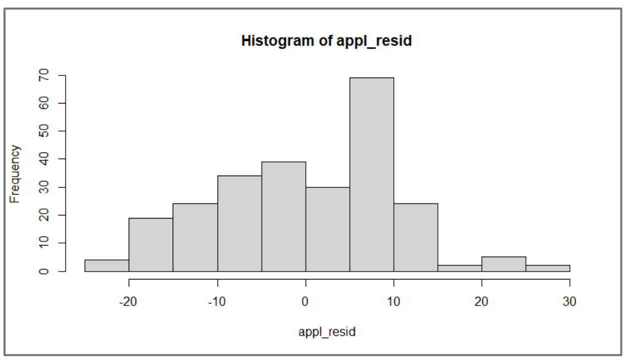
To test for a normal distribution of the residuals, first a Q-Q Plot is created for each model.

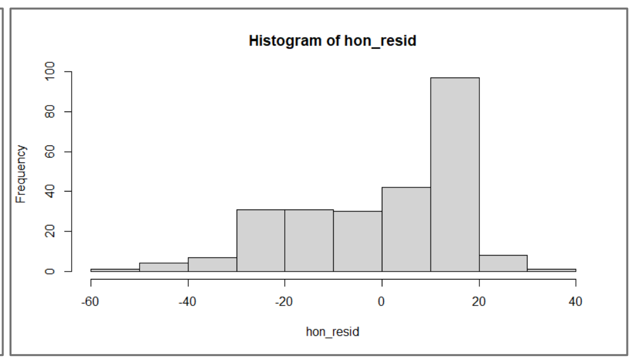




In each of these plots, considerable deviation from the normal distribution can be seen, especially at the right side of plots.

Also looking at histograms for each, they do not match the normal distribution.





To confirm the deviation from a normal distribution, the Shapiro-Wilk test is conducted on each model. This test is similar to the Chi-Squared goodness of fit test but is better suited for continuous data, such as the results of the linear regression model.

```
> shapiro.test(appl_resid)

        Shapiro-Wilk normality test

data:  appl_resid
W = 0.97284, p-value = 9.674e-05

> shapiro.test(hon_resid)

        Shapiro-Wilk normality test

data:  hon_resid
W = 0.92097, p-value = 2.591e-10
```

Both results of both the w and p-value scores indicate a rejection of the null hypothesis that the residuals are normally distributed. However, the large w values, especially for the Honeywell model, indicate that the deviation is not severe.

Conclusion

After examining the data from Apple and Honeywell in the given period, the prediction model with the best performance has been the exponential smoothing model. Given the nature of stock performance and the specific time of consideration, this does seem like a likely outcome. The exponential smoothing method is best suited for short-term predictions and best categorizes short-term momentum over long-term trends.

If considering a new stock portfolio, only containing these two stocks, the recommendation would be 70/30 split in favor of Apple. Many of the models in this study unduly punished the Apple stock due to the massive dip in performance around period 90, but this was due to the unprecedented global impact of Covid-19. Rarely in stock performance can such a large dip be easily explained by short-term (in hindsight) external factors. In the period after the Covid dip, the Apple growth has far exceeded Honeywell. Maintaining the 30% investment in Honeywell is a nod to the stable growth seen in the company after the 2020 dip, and future predictions of performance mirror that slow, steady growth.

References

1. 5 Homoscedasticity | Regression Diagnostics with R. (n.d.). In sscc.wisc.edu. https://sscc.wisc.edu/sscc/pubs/RegDiag-R/homoscedasticity.html
2. Albright, S. (2016) Business Analytics. Sixth Edition. Cengage Learning. Boston, MA.
3. Evans, J. R. (2013). Statistics, data analysis, and decision modeling: International Edition. Pearson Higher Ed.
4. GeeksforGeeks. (2024, August 29). Time Series Forecasting in R with HoltWinters. GeeksforGeeks. https://www.geeksforgeeks.org/r-language/time-series-forecasting-in-r-with-holt-winters/
5. HoltWinters function | R Documentation. (n.d.). Www.rdocumentation.org. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/HoltWinters
6. Lee, S. (2025). 5 Techniques Using Exponential Smoothing for Accurate Forecasts. Numberanalytics.com. https://www.numberanalytics.com/blog/exponential-smoothing-techniques-forecasting
7. Test for Normality. (n.d.). Www.statistics4u.com. http://www.statistics4u.com/fundstat_eng/cc_normality_test.html

Appendix

Apple Forecast

```
> setNames(appl_forecasts[101:257], 101:257)
      101       102       103       104       105       106       107       108       109       110       111 
 67.38491  67.78380  68.18269  68.58158  68.98047  69.37936  69.77824  70.17713  70.57602 
70.97491  71.37380 
      112       113       114       115       116       117       118       119       120       121       122 
 71.77269  72.17158  72.57047  72.96936  73.36825  73.76714  74.16603  74.56492  74.96381 
75.36270  75.76159 
      123       124       125       126       127       128       129       130       131       132       133 
 76.16048  76.55937  76.95826  77.35715  77.75604  78.15493  78.55382  78.95271  79.35160 
79.75049  80.14938 
      134       135       136       137       138       139       140       141       142       143       144 
 80.54827  80.94716  81.34605  81.74494  82.14383  82.54272  82.94161  83.34050  83.73939 
84.13828  84.53717 
      145       146       147       148       149       150       151       152       153       154       155 
 84.93606  85.33495  85.73384  86.13273  86.53162  86.93051  87.32940  87.72829  88.12718 
88.52607  88.92496 
      156       157       158       159       160       161       162       163       164       165       166 
 89.32385  89.72274  90.12163  90.52052  90.91941  91.31830  91.71719  92.11608  92.51497 
92.91386  93.31275 
      167       168       169       170       171       172       173       174       175       176       177 
 93.71164  94.11053  94.50942  94.90831  95.30720  95.70609  96.10498  96.50387  96.90276 
97.30165  97.70054 
      178       179       180       181       182       183       184       185       186       187       188 
```

98.09943 98.49832 98.89721 99.29610 99.69499 100.09387 100.49276 100.89165 101.29054 101.68943 102.08832

189 190 191 192 193 194 195 196 197 198 199

102.48721 102.88610 103.28499 103.68388 104.08277 104.48166 104.88055 105.27944 105.67833 106.07722 106.47611

200 201 202 203 204 205 206 207 208 209 210

106.87500 107.27389 107.67278 108.07167 108.47056 108.86945 109.26834 109.66723 110.06612 110.46501 110.86390

211 212 213 214 215 216 217 218 219 220 221

111.26279 111.66168 112.06057 112.45946 112.85835 113.25724 113.65613 114.05502 114.45391 114.85280 115.25169

222 223 224 225 226 227 228 229 230 231 232

115.65058 116.04947 116.44836 116.84725 117.24614 117.64503 118.04392 118.44281 118.84170 119.24059 119.63948

233 234 235 236 237 238 239 240 241 242 243

120.03837 120.43726 120.83615 121.23504 121.63393 122.03282 122.43171 122.83060 123.22949 123.62838 124.02727

244 245 246 247 248 249 250 251 252 253 254

124.42616 124.82505 125.22394 125.62283 126.02172 126.42061 126.81950 127.21839 127.61728 128.01617 128.41506

255 256 257

128.81395 129.21284 129.61173

Honeywell Forecast

101 102 103 104 105 106 107 108 109 110 111 112 113

129.7957 130.1000 130.4043 130.7086 131.0129 131.3172 131.6215 131.9258 132.2301 132.5344 132.8387 133.1430 133.4473

114 115 116 117 118 119 120 121 122 123 124 125 126

133.7516 134.0559 134.3602 134.6645 134.9688 135.2731 135.5774 135.8817 136.1860
136.4903 136.7946 137.0989 137.4032

127 128 129 130 131 132 133 134 135 136 137 138 139

137.7075 138.0118 138.3161 138.6204 138.9247 139.2290 139.5333 139.8376 140.1419
140.4462 140.7505 141.0548 141.3591

140 141 142 143 144 145 146 147 148 149 150 151 152

141.6634 141.9677 142.2721 142.5764 142.8807 143.1850 143.4893 143.7936 144.0979
144.4022 144.7065 145.0108 145.3151

153 154 155 156 157 158 159 160 161 162 163 164 165

145.6194 145.9237 146.2280 146.5323 146.8366 147.1409 147.4452 147.7495 148.0538
148.3581 148.6624 148.9667 149.2710

166 167 168 169 170 171 172 173 174 175 176 177 178

149.5753 149.8796 150.1839 150.4882 150.7925 151.0968 151.4011 151.7054 152.0097
152.3140 152.6183 152.9226 153.2269

179 180 181 182 183 184 185 186 187 188 189 190 191

153.5312 153.8355 154.1398 154.4441 154.7485 155.0528 155.3571 155.6614 155.9657
156.2700 156.5743 156.8786 157.1829

192 193 194 195 196 197 198 199 200 201 202 203 204

157.4872 157.7915 158.0958 158.4001 158.7044 159.0087 159.3130 159.6173 159.9216
160.2259 160.5302 160.8345 161.1388

205 206 207 208 209 210 211 212 213 214 215 216 217

161.4431 161.7474 162.0517 162.3560 162.6603 162.9646 163.2689 163.5732 163.8775
164.1818 164.4861 164.7904 165.0947

218 219 220 221 222 223 224 225 226 227 228 229 230

165.3990 165.7033 166.0076 166.3119 166.6162 166.9205 167.2248 167.5292 167.8335 168.1378 168.4421 168.7464 169.0507

231 232 233 234 235 236 237 238 239 240 241 242 243

169.3550 169.6593 169.9636 170.2679 170.5722 170.8765 171.1808 171.4851 171.7894 172.0937 172.3980 172.7023 173.0066

244 245 246 247 248 249 250 251 252 253 254 255 256

173.3109 173.6152 173.9195 174.2238 174.5281 174.8324 175.1367 175.4410 175.7453 176.0496 176.3539 176.6582 176.9625

257

177.2668