~~~~
---
title: "DATA 621 - Final Project"
subtitle: "Forecast Financial Stock Market & Returns"
author: "Ramnivas Singh, Deepak Sharma"
date: "`r Sys.Date()`"
output:
  html_document:
    theme: default
    highlight: espresso
    toc: yes
    toc_depth: 5
    toc_float:
      collapsed: yes
  pdf_document:
    toc: yes
    toc_depth: '5'
  editor_options:
   chunk_output_type: inline
  always_allow_html: true
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE)
```

\clearpage

--------------------------------------------------------------------------------

```{r options_pkgs, echo=F, warning=F, message=F, results=F}
# knitr::opts_chunk$set(error = F, message = F, # tidy = T,
#                       cache = T, warning = T,
#                       results = 'hide', # suppress code output
#                       echo = F,         # suppress code
#                       fig.show = 'hide' # suppress plots
#                       )
library(readr)
library(skimr)
library(knitcitations)
library(pander)
library(tidyverse)
library(corrplot)
library(quantmod)
library(data.table)
library(car)
library(caret)
library(nlme)
library(kableExtra)
library(plm)
library(broom)
library(tseries)
library(forecast)
library(tidyquant)
```
~~~~

# Abstract

Financial markets refer broadly to any marketplace where the trading of securities occurs, including the stock market, bond market, forex market, and derivatives market, among others. Financial markets are vital to the smooth operation of capitalist economies. The stock market is just one type of financial market. Financial markets are made by buying and selling numerous types of financial instruments including equities, bonds, currencies, and derivatives.

Financial markets rely heavily on informational transparency to ensure that the markets set prices that are efficient and appropriate. The market prices of securities may not be indicative of their intrinsic value because of macroeconomic forces like taxes.

Since predicting the exact percent increase of a stock the next day is difficult, we can instead try instead to predict the direction of the movement. Alternatively, we can look at the trends and seasonality in the data and try to predict with some level of confidence where the price would like in an arbitrary number of days.

The logistic models which try to predict the direction of the next day's movement yielded poor results. Hardly any of the predictor variable were significant, and the classification accuracies were around 52% which is hardly better than guessing. Panel Regression as well as the autoregressive models using ARIMA attempt to predict closing prices of stocks using one stock at a time.

Panel Regression attempts to build one model for any number of stocks across the same time periods. Doing this also yielded poor and sporadic results when predicting per stock. On the other hand, the auto-regressive models that predict subsequent values of a stock time series were of greater use. This required finding a suitable combination of parameters to fit the data closely.

# Key Words

1. ARIMA
2. Time series
3. Logistic regression
4. Auto-regressive models
5. Panel regression

# Introduction

It is absolutely impossible to predict accurate market movement and stock prices. To find an accurate reliable ways to predict stock prices is a difficult exercise and the methods used often yield unsatisfactory results. If we believe that the price of a stock, or the direction of its movement can be predicted using recent information then we can look at variables associated with recent movements with indicators such as RSI or moving averages as well as intraday variations in price via the OHLC prices. A lot of stock market data is readily available and it is important to find significant predictors variables amongst all the options.

If on other the hand we believe that trends and long-term movements are a better way to predict prices then time series analysis can be used. The analysis consists of comparing a series of prices such as the close price of a stock with a lagged or shifted version of itself in order to discern any temporal correlations. This technique requires certain assumptions to be met that require transformations of the data. Through different modeling approaches and parameter tuning, we can prediction what future prices will be to a certain degree of confidence.

An autoregressive integrated moving average, or ARIMA, is a statistical analysis model that uses time series data to either better understand the data set or to predict future trends. A statistical model is autoregressive if it predicts future values based on past values. For example, an ARIMA model might seek to predict a stock's future prices based on its past performance or forecast a company's earnings based on past periods.

An autoregressive integrated moving average model is a form of regression analysis that gauges the strength of one dependent variable relative to other changing variables. The model's goal is to predict future securities or financial market moves by examining the differences between values in the series instead of through actual values.

An ARIMA model can be understood by outlining each of its components as follows:

* Autoregression (AR): refers to a model that shows a changing variable that regresses on its own lagged, or prior, values.

* Integrated (I): represents the differencing of raw observations to allow for the time series to become stationary (i.e., data values are replaced by the difference between the data values and the previous values).

* Moving average (MA): incorporates the dependency between an observation and a residual error from a moving average model applied to lagged observations.

# Methodology

The dataset contains typical pricing variables and it is also augmented with engineered features which are believed to have predictive power. The dataset is composed of a variety of stocks. For the purpose of brevity, the data exploration will be limited to a single stock, Apple (AAPL). The engineered features used are technical indicators such as the RSI (Relative Strength Indicator), the MACD (Moving Average Convergence Divergence), or simple moving averages.

The logistic models require the creation of a binary target variable. It is calculated by comparing the price of the stock with the price on the following day. If the difference is positive, the target is evaluated as 1, and 0 otherwise. The logistic models are built using Generalized Linear Models suited for the exponential family of distributions (in this case binomial) with the help of the `glm` function.

The Panel regression models also use the engineered features such as RSI, MACD, closing price moving averages, and trade volume moving averages. Since our entire dataset consists of several different stocks over given time periods, the individuality of each company is taken into consideration using a fixed effects panel model. The closing price is then predicted factoring in the individual specific company effects.

The autoregressive models only make use of the price time series and fits a model that explains a given time series based on its previous values and how correlated it is to its lags (previous values). autoregressive Integrated Moving Average models (ARIMA) are 3 parameter models that can be used to forecast future values. The parameter p represents the AR (autoregressive) term, d is the MA (moving average) term and I (integration) is the order of differentiation required to make the series stationary.

# Experimentation and Results

## Data Description

The variables description are as follows:

Variable	Description
date	Trading Date
open	Price of the stock at market open
high	Highest price reached in the trade day
low	Lowest price reached in the trade day
close	Price of the stock at market close
volume	Number of shares traded
unadjustedVolume	Volume for stocks, unadjusted by stock splits
change	Change in closing price from prior trade day close
changePercent	Percentage change in closing price from prior trade day close
vwap	Volume weighted average price (VWAP) is the ratio of the value traded to total volume traded
label	Trading Date
changeOverTime	Percent change of each interval relative to first value. Useful for comparing multiple stocks.
ticker	Abbreviation used to uniquely identify publicly traded shares

```{r message=FALSE, warning=FALSE}
dataset <- read_csv('stocks_combined.csv')
tickers <- read_csv('tickers.csv')
```

The tickers available in this dataset are listed below. We will be focusing on the AAPL stock which is described below.

```{r}
tickers
```

~~~~
```{r}
skim(dataset)
```

## Data Processing

The only necessary processing of the data is formatting the date column
appropriately.

```{r}
dataset$date <- as.Date(dataset$date, format="%m/%d/%Y")
```

## Feature Engineering

The dataset is augmented with technicals indicators from the `ta-lib` package
and the target variable for the logistic model is created. Observations are
required to be dropped where NA values are introduced due to the shifting
required by the window functions. We are interested in rates of change instead
of value in order to be able to any stocks, which will not have the same
price.

- *TARGET*: the binaby target variable for the logistic model
- *MACD*: the MACD signal (only) using the typical 12 and 26 window sizes
- *m5*: momentum indicator over the last 5 days (moving average of one trading
week)
- *m20*: momentum indicator over the last 20 days (moving average of one
trading month)
- *vol1*: rate of change of the volume over the last day
- *vol5*: rate of change of the volume over the last five dasy (one trading
week)


```{r}
AAPL <- dataset %>% filter(ticker=='AAPL')

# Augment data frame
AAPL_full <- AAPL %>% mutate(macd=MACD(Cl(AAPL), nFast=12, nSlow=26, nSig=9,
maType=SMA)[,1],
                        m5=momentum(AAPL$close, n=5),
                        m20=momentum(AAPL$close, n=20),
                        rsi=RSI(AAPL$close, n=14),
                        vol1=ROC(AAPL$volume, n=1),
                        vol5=ROC(AAPL$volume, n=5)) %>%
                        drop_na() %>%
                      select(-c(date,ticker,label, changeOverTime))

# Create target variable
AAPL_model1_data <- AAPL_full%>%
                  mutate(TARGET=if_else(shift(close, n=1, fill=NA,
type="lead") > close, 1, 0)) %>%
~~~~

~~~
                  #select(-
c(date,ticker,label,open,high,low,close,volume,unadjustedVolume,vwap,change,changeOverTime
%>%
                  drop_na()

AAPL_model1_data$TARGET <- factor(AAPL_model1_data$TARGET)
```

```{r}
head(AAPL_model1_data,10)
```

## Data Exploration

The stock price for AAPL over the years are risen steadily. There are two
important periods of decline but the overall trends remains positive. These
periods of decline should provide balance to the dataset.
```{r}

AAPL %>% ggplot(aes(x=date,y=close), colour = 'red', size = 3) +
  geom_point()+ geom_hline(yintercept =230, color = "red")

```

The distributions of the predictor variables are mostly centered. The
variables related to volume are skewed to the right tail. The variables close,
high and low

```{r variables_distribution, fig.height=8, fig.width=10, message=FALSE,
warning=FALSE}
AAPL_full %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(x= value)) +
  geom_histogram(fill='grey') +
  facet_wrap(~key, scales = 'free')
```

The correlation plot reveals that some variable are nearly perfectly
correlated. This makes sense given that some of these variables generally move
together or are derived from another.

```{r}
library(corrplot)
corr_dataframe <- AAPL_model1_data %>% mutate_if(is.factor, as.numeric)
corr.d <- cor(corr_dataframe)
corr.d[ lower.tri( corr.d, diag = TRUE ) ] <- NA
corrplot( corr.d, type = "upper", diag = FALSE )
```

The target variable we created has a few more observations where the movement
was positive. This seems reasonable.

```{r message=FALSE, warning=FALSE}
~~~

~~~
AAPL_model1_data %>% ggplot(aes(x=TARGET)) + geom_histogram(stat="count")
```

## Modeling

### Model 1 (Logistic)

```{r}
# Initialize a df that will store the metrics of models
models.df <- tibble(id=character(), formula=character(),
res.deviance=numeric(), null.deviance=numeric(),
                 aic=numeric(), accuracy=numeric(), sensitivity=numeric(),
specificity=numeric(),
                precision.deviance=numeric(), stringsAsFactors=FALSE)
```



```{r include=FALSE}
# A function to extract the relevant metrics from the summary and confusion
matrix
build_model <- function(id, formula, data) {
  glm.fit <- glm(formula, data=data, family=binomial(link="logit"))
  print(summary(glm.fit))
  glm.probs <- predict(glm.fit, type="response")
  # Confirm the 0.5 threshold
  glm.pred <- ifelse(glm.probs > 0.5, 1, 0)
  results <- tibble(target=data$TARGET, pred=glm.pred)
  results <- results %>%
    mutate(pred.class = as.factor(pred), target.class = as.factor(target))

  #print(confusionMatrix(results$pred.class,results$target.class, positive =
"1"))

  acc <- confusionMatrix(results$pred.class,results$target.class, positive =
"1")$overall['Accuracy']
  sens <- confusionMatrix(results$pred.class,results$target.class, positive =
"1")$byClass['Sensitivity']
  spec <- confusionMatrix(results$pred.class,results$target.class, positive =
"1")$byClass['Specificity']
  prec <- confusionMatrix(results$pred.class,results$target.class, positive =
"1")$byClass['Precision']
  res.deviance <- glm.fit$deviance
  null.deviance <- glm.fit$null.deviance
  aic <- glm.fit$aic
  metrics <- list(res.deviance=res.deviance,
null.deviance=null.deviance,aic=aic, accuracy=acc, sensitivity=sens,
specificity=spec, precision=prec)
  metrics <- lapply(metrics, round, 3)

  #plot(roc(results$target.class,glm.probs), print.auc = TRUE)
  model.df <- tibble(id=id, res.deviance=metrics$res.deviance,
null.deviance=metrics$null.deviance,
~~~

~~~~
                         aic=metrics$aic, accuracy=metrics$accuracy,
sensitivity=metrics$sensitivity, specificity=metrics$specificity,
precision=metrics$precision)
  model.list <- list(model=glm.fit, df_info=model.df)
  return(model.list)
}
```

#### M1A Full Model

The model should be reduced by removing the least significant predictors until
the model is significant. The first logistic model yields unsatisfactory
results. Only the `close` variable is significant. It is revealed that a lot
of the variables are  multicollinear. This makes sense given that a number of
variables are related to the price are generally move together (close, open,
low, high), and some are derived from other variables.

```{r}
model.full <- build_model('model.full', "TARGET ~ .", data = AAPL_model1_data)
models.df <- rbind(models.df,model.full$df_info)
#summary(model.full)
```

##### Variance Inflation Factors

```{r echo=FALSE}
car::vif(model.full$model)
```


#### M1B Small Model

Reducing the model down to only the significant predictors leads to a
nonsensical results where the two variables cancel each other out.

```{r}
model.small <- build_model('model.small', "TARGET ~ .-vol5 -high -rsi -open -
vol1-changePercent-low-macd-m20-volume-unadjustedVolume-m5-change", data =
AAPL_model1_data)
models.df <- rbind(models.df,model.small$df_info)
summary(model.small)
```

#### M1C Multicollinearity Removed

This model removes a lot of the predictors (open,high,low,vwap,change,macd)
but adds `Close2Open` and `Open2Open` in order to  try to retain some
information from the variables that were dropped. The removed variables were
determined to be too correlated and we retained only the predictors with
Variance Inflation Factors below an allowable treshold as seen below.
Unfortunately, this model also lacks significance.

```{r}
m1c_data <- AAPL_model1_data
~~~~

~~~
m1c_data$Close2Open <- m1c_data$open - shift(m1c_data$close, n=1, fill=NA,
type="lag")
m1c_data$Open2Open <- m1c_data$open - shift(m1c_data$open, n=1, fill=NA,
type="lag")
m1c_data_trimmed <- m1c_data %>% select(-c(open,high,low,vwap,change,macd))
%>% drop_na()

model.m1c  <- build_model('model.m1c', "TARGET ~ .", data = m1c_data_trimmed )
models.df <- rbind(models.df,model.m1c$df_info)
summary(model.m1c)
```
##### Variance Inflation Factors
```{r}
car::vif(model.m1c$model)
```

#### Logistic Model Results

The results of the logistic models are displayed below. The in-sample testing
accuracy is fairly low at 52%. No model has a particular edge so we proceed to
move on to other types of model. This exercise could be enhanced by using data
from other stocks by removing all the variables that are stock dependent.
Additionally, more target could be used such as the the direction of the price
movement over 5 days or 20 days.

```{r}
models.df
```

### Model 2 (Autoregressive Models)

In this section we drop all the predictors but the close price of the time
series. We only use information contain the series itself to fit a model and
make predictions. The AAPL stock price had an underlying trend but large
fluctuations. We need to verify if certain conditions are met in order to
model this time series. We first look at the autocorrelation of the series and
see that the series is highly correlated with its previous values with a
significant lag. We also consider the PACF which removes variations explained
by earlier lags so we get only the relevant features.

Identification of an AR model is often done using the PACF while MA models use
the ACF. The order of the model is determined by the number of significant
lags taken into account. The PACF suggests ar AR(1) model and a regression
line of time t onto time t-1 shows a close linear fit and a highly statistical
coefficient of approximately 1 for `t-1`. A few methods are used to arrive at
the estiamte but the coefficient is consistent. However, the slowly decaying
ACF suggests that the series is not stationary and that some degree of
differentiation might be required to stabilize the mean.


```{r}
par(mfrow=c(2,2))
acf(AAPL$close)
pacf(AAPL$close)
~~~

~~~
t <- AAPL$close[-1]
t_1 <- AAPL$close[-length(AAPL$close)]
m2.lm <- lm(t ~ t_1)

plot(t_1, t)
abline(m2.lm, col=3, lwd=2)
```

```{r}
summary(m2.lm)
```

We use the generalized least squares method to obtain more information on the
mode. We specify an error correlation structure of order 1. We note that the
coefficient estiamte is again 1 and the AIC is 5478. This will serve as a
comparison of more complex models.

```{r}
AAPL_close <- ts(AAPL$close)
m2.gls <- gls(AAPL_close ~ time(AAPL_close), correlation = corAR1(form=~1))
summary(m2.gls)
```

We test the justification of the added autoregressive structure using a
likelyhood ratio test. With a very small p value, we can reject the null
hypothesis that the added term is not necessary.

```{r}
m2.gls.0 <- update(m2.gls, correlation=NULL)
anova(m2.gls, m2.gls.0) # AR(1) vs Uncorrelated errors
```

Fitting an AR(1) model using the `ar` function confirms the coefficient of
approxmiately 1 (0.997).

```{r}
m2.ar1 <- ar(AAPL_close)
m2.ar1_fitted <- AAPL$close - residuals(m2.ar1)
m2.ar1
```

We can view the fitted of the GLS and AR(1) models side by side.

```{r}
par(mfrow=c(1,2))
m2.gls_fitted <- AAPL_close - residuals(m2.gls)
plot(AAPL$close, type = "l", col = 4, lty = 1, main="GLS")
points(m2.gls_fitted, type = "l", col = 2, lty = 2)

plot(AAPL$close, type = "l", col = 4, lty = 1, main="AR(1)")
points(m2.ar1_fitted, type = "l", col = 2, lty = 2)
```
~~~

We noted earlier that the slowly decaying structure of the ACF suggested that the series is not stationary. We verify this claim using the Dickey Fuller tests below. The null hypothesis that the series is not stationary is not rejected for the original series, but rejected for the once differentiated series. Therefore, the original series is not stationary and should be differentiated to stabilize the mean.

Dickey Fuller Test on time series:
```{r message=FALSE, warning=FALSE}
adf.test(diff(AAPL$close, differences=1))
```

Dickey Fuller Test on differentiated time series:
```{r message=FALSE, warning=FALSE}
adf.test(diff(AAPL$close, differences=1))
```

#### ARIMA

ARIMA models allows to combine the AR structure seen above with differentiation and moving average terms.

Here we look at the first two differentials. The higher order differentials have mean 0 but the variance seems to increase in the last part of the series.

##### Order 1 differential

The order 1 differential shows the highest correlation at position 0 and a significant lag at position 6, while the partial ACF seenms to show periodic behavior with a few significant lags at 7 and 8.

```{r}
APPLdiff1 <- diff(AAPL$close, differences=1)
par(mfrow=c(2,2))
acf(APPLdiff1, lag.max=20)
pacf(APPLdiff1, lag.max=20)
plot.ts(APPLdiff1)
```

##### Order 2 differential

On the other hand, the order two differential has several meaningful lags at position 1 and 7, and the partial ACF decays and stays significant until lag 6.

```{r}
APPLdiff2 <- diff(AAPL$close, differences=2)
par(mfrow=c(2,2))
acf(APPLdiff2, lag.max=20)
pacf(APPLdiff2, lag.max=20)
plot.ts(APPLdiff2)
```

Form these observations, we can study the following ARMA (autoregressive moving average) models are possible for the differentiated time series:

* First Order ARIMA(7,1,0): AR(7) high order on the autoregressive structure
* First Order ARIMA(0,1,2): MA(1) moving average structure
* First Order ARIMA(1,1,1): ARMA(1,1) combination

* Second Order ARIMA(1,2,0): AR(1) autoregressive structure
* Second Order ARIMA(0,2,1): MA(1) moving average structure
* Second Order ARIMA(6,2,0): AR(6) autoregressive structure
* Second Order ARIMA(6,2,1): ARMA(6,1) combination

```{r}
arimadf <- tibble(model=character(), coef=character(), loglik=numeric(), aic=numeric())

arimamodels <- function(p,d,q) {
  # A specification of the non-seasonal part of the ARIMA model: the three components (p, d, q) are the AR order, the degree of differencing, and the MA order. ARMA(p,q)
  arima.model <- arima(AAPL$close, order = c(p,d,q))
  arima.model_fitted <- AAPL$close - residuals(arima.model)
  df <- tibble(model=paste0(p,d,q), loglik=arima.model$loglik, aic=arima.model$aic)
  return.list <- list(model=arima.model,fitted=arima.model_fitted, df=df)
  return(return.list)
}
```

#### Model Results

The results of the ARIMA models listed above are printed here. We find that the model with the lowest AIC is the first order differential AR(7) model with a value of 5466. However is is also a complex model with 8 parameters. A smaller model with marginally decreased AIC is preffered. We can choose for the any of the remaining 3 parameter models. We select the second order MA(1) model at the best model.

```{r}
# First order models
m710 <- arimamodels(p=7, d=1, q=0)
arimadf <- rbind(arimadf, m710$df)
m012 <- arimamodels(p=0, d=1, q=2)
arimadf <- rbind(arimadf, m012$df)
m111 <- arimamodels(p=1, d=1, q=1)
arimadf <- rbind(arimadf, m111$df)

# Second order models
m120 <- arimamodels(p=1, d=2, q=0)
arimadf <- rbind(arimadf, m120$df)
m021 <- arimamodels(0, 2, 1)
arimadf <- rbind(arimadf, m021$df)
m620 <- arimamodels(6, 2, 0)
```

~~~
arimadf <- rbind(arimadf, m620$df)
m621 <- arimamodels(6, 2, 1)
arimadf <- rbind(arimadf, m621$df)

arimadf
```

The coefficients for the best model are listed below and we can also see from
the graph that the model is a good fit.

```{r}
m021$model
```

```{r}
plot(AAPL$close, type = "l", col = 4, lty = 1)
points(m021$fitted, type = "l", col = 2, lty = 2)
```

We can now use the model to forecast the price of the stock. In the example
below the price is predicted for the next 150 trading days. Confidence
intervals (shown in blue) are provided at 80% and 95% levels.

```{r}
AAPLforecast <- forecast(m021$model, h=150)
autoplot(AAPLforecast)
```

#### Forecast vs Actual

```{r}
arimaforecast <- function(stock, ticker) {
  # Subset the training data, fit model and get fitted valyes
  training_subset <- stock$close[1:round(0.7*length(AAPL$close))]
  model <- arima(training_subset, order = c(0, 2, 1))
  fitted_values <- training_subset - residuals(model)

  boundary <- round(length(stock$close)*.7)
  end <- length(stock$close)
  data <- tibble(date=stock$date, price=stock$close, prediction = NA,
fitted=NA, Low95 = NA,
         Low80 = NA,
         High95 = NA,
         High80 = NA)
  stocksubsetforecast <- forecast(model, h=377)
  forecast_df <- fortify(as.data.frame(stocksubsetforecast)) %>% as_tibble()
  forecast_df <- forecast_df %>%
    rename("Low95" = "Lo 95",
           "Low80" = "Lo 80",
           "High95" = "Hi 95",
           "High80" = "Hi 80",
           "Forecast" = "Point Forecast")
  data$fitted <- c(as.vector(fitted_values),rep(NA,(end-boundary)))
  data$prediction <- c(rep(NA,boundary),forecast_df$Forecast)
~~~

~~~
  data$Low80 <- c(rep(NA,boundary),forecast_df$Low80)
  data$High80 <- c(rep(NA,boundary),forecast_df$High80)
  data$Low95 <- c(rep(NA,boundary),forecast_df$Low95)
  data$High95 <- c(rep(NA,boundary),forecast_df$High95)

  ggplot(data, aes(x = date)) +
    geom_ribbon(aes(ymin = Low95, ymax = High95, fill = "95%")) +
    geom_ribbon(aes(ymin = Low80, ymax = High80, fill = "80%")) +
    geom_point(aes(y = price, colour = "price"), size = 1) +
    geom_line(aes(y = price, group = 1, colour = "price"),
              linetype = "dotted", size = 0.75) +
    geom_line(aes(y = fitted, group = 2, colour = "fitted"), size = 0.75) +
    geom_line(aes(y = prediction, group = 3, colour = "prediction"), size = 0.
75) +
    scale_x_date(breaks = scales::pretty_breaks(), date_labels = "%b %y") +
    scale_colour_brewer(name = "Legend", type = "qual", palette = "Dark2") +
    scale_fill_brewer(name = "Intervals") +
    guides(colour = guide_legend(order = 1), fill = guide_legend(order = 2)) +
    theme_bw(base_size = 14) +
    ggtitle(ticker)
}
```
~~~

The actual price is slightly lower that the linearly predicted price. We also tested how the structure of this ARIMA model generalized to other stock price time series. Below are price prediction for Microsoft and IBM. In the case of the former, the price stays in the upper range of the of the 95% confidence interval and even exceeds it. The final price is well above the prediction. The forecast for IBM is more within the range of the linear predictor seems a reasonable estimation of the movement.

Now that we have a well-fiting model that can predict the stock price we can compare the prediction performance to the actual price of the stock by subsetting a training set using the first 70% of the data and predicting on the remaining 30% of the trading days. We find that the actual price of AAPL remain within the 80% and 95% confidence interval ribbons.

~~~
```{r message=FALSE, warning=FALSE}
arimaforecast(AAPL,'AAPL')
```

```{r}
MSFT <- dataset %>% filter(ticker=='MSFT')
IBM <- dataset %>% filter(ticker=='IBM')
JPM <- dataset %>% filter(ticker=='JPM')
PFE <- dataset %>% filter(ticker=='PFE')
INTC<- dataset %>% filter(ticker=='INTC')
WMT<- dataset %>% filter(ticker=='WMT')
```

```{r message=FALSE, warning=FALSE}
arimaforecast(MSFT,'MSFT')
```
~~~

~~~
```{r message=FALSE, warning=FALSE}
arimaforecast(IBM,'IBM')
```

```{r message=FALSE, warning=FALSE}
arimaforecast(MSFT,'JPM')
```

```{r message=FALSE, warning=FALSE}
arimaforecast(IBM,'INTC')
```
```{r message=FALSE, warning=FALSE}
arimaforecast(MSFT,'PFE')
```

```{r message=FALSE, warning=FALSE}
arimaforecast(IBM,'WMT')
```



### Model 3 (Panel Regression)
For the 3rd model we will attempt to panel models on the stocks dataset as
this dataset provides data across stocks and over time, thus having both
cross-sectional and time-series dimensions. The general assumption for this
dataset is that there is correlation over time for a given stock, but each
stock is independent of other stocks. This dataset has not have any time-
invariant regressors.

#### Balance dataset
Determine if all stocks are observed for all time periods and if unbalanced,
balance so that all stocks are observed on the same trade dates. This
indicates that some observations will be dropped so that all stocks have an
observations on the same trading days
```{r}
dataset2 <- dataset %>% select(-label)
is.pbalanced(dataset2)
dataset2 <- make.pbalanced(dataset2,balance.type = "shared.times")
is.pbalanced(dataset2)

```
#### Create Predictors
For all stock observations in our dataset, we will create the engineered
features: MACD, RSI, m5, m20, v1, and v5
```{r}

create_predictors <- function(df){
  df_full <- df %>% mutate(macd=MACD(Cl(df), nFast=12, nSlow=26, nSig=9,
maType=SMA)[,1],
                        m5=momentum(df$close, n=5),
                        m20=momentum(df$close, n=20),
                        rsi=RSI(df$close, n=14),
                        vol1=ROC(df$volume, n=1),
                        vol5=ROC(df$volume, n=5))
  return(df_full)
~~~

~~~~
}

balanced_tickers <- unique(dataset2$ticker)
dataset2_full <- dataset2 %>% filter(ticker=="AAPL") %>% create_predictors()

for(i in balanced_tickers[2:length(balanced_tickers)])
{
 dataset2_full <- rbind(dataset2_full, dataset2 %>% filter(ticker==i) %>%
create_predictors())
}

dataset2_full %>% filter(date=="2014-05-21")

```

#### Test and Train datasets
Create test and training datasets for prediction purposes, and ensure both
datasets are balanced
```{r}
stock_dates <- unique(dataset2_full$date)

test_dates <- tail(stock_dates, 250)
train_dates <- head(stock_dates, length(stock_dates)-250)

test_stocks <- dataset2_full %>% dplyr::filter(date %in% test_dates)
train_stocks <- dataset2_full %>% dplyr::filter(date %in% train_dates)
is.pbalanced(test_stocks)
is.pbalanced(train_stocks)
train_stocks <- train_stocks %>% drop_na()
panel_stocks <- pdata.frame(train_stocks, index = c("ticker", "date"))
```
#### Model 3A: Pooled OLS
A Pooled model on panel data is applies the Ordinary Least Squares technique
on the data. It is the most restrictive panel model as cross-sectional
dimensions are ignored:
$y_{it}=\alpha+\beta x_{it}+u_{it}$


```{r}
stocks_m3_pooled <- plm(close ~ macd + m5 + m20,  data = panel_stocks, model =
"pooling")
summary(stocks_m3_pooled)
Pooled_Model <- glance(stocks_m3_pooled)
```

#### Fixed Effect Models
In Fixed affects models we will assume there is an unobserved heterogeneity
across the stocks such as each company's core competency, or anything else
that is unique to each company and thus something unobserved factored into
each company's stock price. This heterogeneity ($\alpha_i$) is not known but
we would want to investigate it's correlation with the created predictor
variables see it's impact on the closing price:
$y_{it}=\alpha_i+\beta x_{it}+u_{it}$
~~~~

~~~
##### Model 3B: Fixed Model - Within Estimator
```{r}
stocks_m3_within <- plm(close ~ macd + m5 + m20 + rsi + vol5,  data =
panel_stocks, model = "within")
summary(stocks_m3_within)
Fixed_Model <- glance(stocks_m3_within)
```

##### Model 3C: Fixed Model - First Difference Estimator
The First difference estimator uses period changes for each stock, where the
individual specific effects (unobserved heterogeneity) is canceled out:
$y_{it}-y_{i,t-1} = \beta(x_{it}-x_{i,t-1})+(e_{it}-e_{i,t-1})$

Looking at the result, we see that while the adjusted $R^2$ is fairly high
compared to the fitted values are nonsensical
```{r}
stocks_m3_fd <- plm(close ~ macd + m5 + m20 + rsi + vol1 + vol5,  data =
panel_stocks, model = "fd")
summary(stocks_m3_fd)
Fixed_Model_FD <- glance(stocks_m3_fd)
fitted.values(stocks_m3_fd)[1:10]

```
#### Model 3D: Random Effect Model
In Random affects models assumes no fixed effects.The individual specific
effects (unobserved heterogeneity) are not correlated with the predictor
variables and are independent of the predictor variables. This would result in
the assumption that any residual variation on the dependent variable ( closing
price) is random and should randomly distributed with the error term:
$y_{it}=\beta x_{it}+(\alpha_i+ e_{it})$

```{r}
stocks_m3_random <- plm(close ~ macd + m5 + m20 + rsi + vol5,  data =
panel_stocks, model = "random")
summary(stocks_m3_random)
Random_Model <- glance(stocks_m3_random)

```


#### Panel Model Testing & Selection
Below is a summary glance of the 4 Panel models run on the stocks dataset.
Choosing between either a pooled, fixed effect, or random effect model
requires running Breusch–Pagan Lagrange Multiplier and Hausman tests. Of the 4
models below, we will discard the First Difference model due to its
nonsensical fitted values
```{r, warning=FALSE}
Panel_Model_Summary <- data.frame(Pooled_Model)

Panel_Model_Summary <- rbind(Panel_Model_Summary, Fixed_Model)
Panel_Model_Summary <- rbind(Panel_Model_Summary, Fixed_Model_FD)
Panel_Model_Summary <- rbind(Panel_Model_Summary, Random_Model)
~~~

```
rownames(Panel_Model_Summary) <- c("Pooled Model", "Fixed Model", "First Difference Model", "Random Model")
Panel_Model_Summary
```
The Breusch–Pagan Lagrange Multiplier Test (LM Test) is used to test for heteroskedasticity in a linear regression model. The null hypothesis assumes homoskedasticity and in the alternate hypothesis heteroskedasticity is assumed.

First, we test for Random Effects against OLS. From the test we see that since the p-value is near 0, we reject the null hypothesis. The individual specific effects (unobserved heterogeneity) of each stock are significant and therefore we should not use the Pooled OLS model.

```{r}
plmtest(stocks_m3_pooled, effect = "individual")
```

We test the Fixed Effects model against OLS Model, we again see that we reject the null for the alternative hypothesis as the test suggest more support for the Fixed Effect model

```{r}
#LM test for fixed effects vs OLS
#significant result, more support for fixed effects model over ols
pFtest(stocks_m3_within, stocks_m3_pooled)
```
From the two tests above, we can set aside the Pooled OLS model which ignores both the cross-sectional and time-series dimensions. Next, we compare the Fixed Effects and Random Effects model. We do this using the Hausman Test, which evaluates the consistency of estimators, in our case the fixed effect and random effect estimators. If there is no correlation between the independent variables and the individual specific effects, then both Fixed Effect and Random Effect models are consistent, but the Fixed Effect model is not efficient. Should there be a correlation, then the Fixed Effects model is consistent and the Random Effects model is inconsistent.

From the result we see that the null hypothesis is rejected. This indicates that the hypothesis is that individual random effects of each stock are uncorrelated with the error term does not have support. We therefore choose the alternative hypothesis and select the Fixed Effect model

```{r}
phtest(stocks_m3_random, stocks_m3_within)
```

# Discussion and Conclusion

In this project, we implemented models using logistic regression, panel regression, and auto-regressive models using ARIMA.

The logistic models which try to predict the direction of the next day's movement yielded poor results. Hardly any of the predictor variable were significant, and the classification accuracies were around 52% which is hardly better than guessing. The multicollinearity issues that were addressed did not improve the results so this model was abandoned. Some improvements are possible using a larger dataset of multiple stocks or using different target variables.

For Panel Regression we created and compared the significance of 3 models, the Pooled OLS, Fixed Effects, and Random Effects models. Within the Fixed Model, we tried the First Differences estimator which while having explaining the most variability produced results which do not make sense therefore was discarded. The remaining 3 models had similar explanation of variability, but given our dataset and the nature of it only the Fixed Effect model should be applicable.

The auto-regressive models that predict subsequent values of a stock time series were of greater use. This required finding a suitable combination of parameters to fit the data closely. Simple linear regression suggested that an AR(1) could be used where the predictor is the previous value of the price. We estimated the coefficients and fit a model but also noted that the series was not stationary and should be differentiated in order to stabilize the mean to conform with the modeling assumptions.

We progress some a simple AR(1) auto-regressive model to include the differentiation parameter as well as the moving average to the structure. A variety of models were fit and it was found that smalle models were preferred. The model selected as the best model was a second order differential model with a MA(1) structure.

We were able to test its validity by splitting stock data into training and test sets to evaluate the performance of the model. Surprisingly, the MA(1) consistently capture the movement of the stock price within its confidence intervals. The intervals remain fairly wide and the linear trend is not guaranteed to reflect the actual price of the stock at any time, but only centers the trend.

Given these 3 models, the auto-regressive models using ARIMA produced the most stable results. The output of each of the 3 models is different making comparison of each to each other problematic. The logistic model produces binary results, the panel model predicts actual close values, and the auto-regressive model estimates general stock trend over a given period of time. Given that it is a difficult task to produce a fairly accurate prediction of stock prices by trade day, the auto-regressive ARIMA model that gives trends over periods of time is a better and more suitable method determining/ estimating stock price movement

# Reference

## Books & Papers

* Stock price forecasting by Alibris
* Stock Market Prediction by Bradley
* Machine Learning Solutions by Jalaj Thanaki
* The Intelligent Investor by Benjamin Graham.

* Value investing and behavioral finance by Parag Parikh
* Common stocks and uncommon profits by Philip A. Fisher
* How to avoid loss and earn consistently in stock market by Prasenjit Paul
* Stock Prediction with Deep Learning Paperback by Ethan Shaotran, Mark Munoz, Foreword

## Studies & Links

* Hyndman, Rob J., and George Athanasopoulos. “Forecasting: Principles and Practice” Otexts. N.p., May 2012. Web.
* A. Trapletti and K. Hornik (2016). tseries: Time Series Analysis and Computational Finance. R package version 0.10-35.
* R. J. Hyndman(2016). forecast: Forecasting functions for time series and linear models . R package version 7.2, http://github.com/robjhyndman/forecast>.
* Irizzary,R., 2018,Introduction to Data Science,github page,https://rafalab.github.io/dsbook/
* Sean J Taylor and Benjamin Letham., 2017, Forecasting at scale, https://facebook.github.io/prophet/

* https://stats.idre.ucla.edu/r/dae/negative-binomial-regression/
* https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/
* https://saas.berkeley.edu/rp/arima
* https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/
* https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average
* https://people.duke.edu/~rnau/411arim.htm
* https://www.analyticsvidhya.com/blog/2021/11/performing-time-series-analysis-using-arima-model-in-r/
* https://towardsdatascience.com/an-introduction-to-time-series-analysis-with-arima-a8b9c9a961fb