STAT 610 Project Draft

Connor Kennedy

12/2/2021

```
library(tidyverse)
library(tidyquant)
library(lubridate)
library(bsts)
```

Data and pre-processing

```
# Load data
df <- tq_get("GOOGL", from = "2011-01-01", to = "2021-06-28") %>%
  mutate(st.close = log(close)) # perform stationary transformation

# Reduce dimensions
df.st <- df %>%
  filter(year(date) < 2021) %>%
  dplyr::select(date, close, st.close)

df.test <- df %>%
  filter(year(date) > 2020) %>%
  dplyr::select(date, close, st.close)

# Check observation counts
nrow(df.st)
```

```
## [1] 2517
```

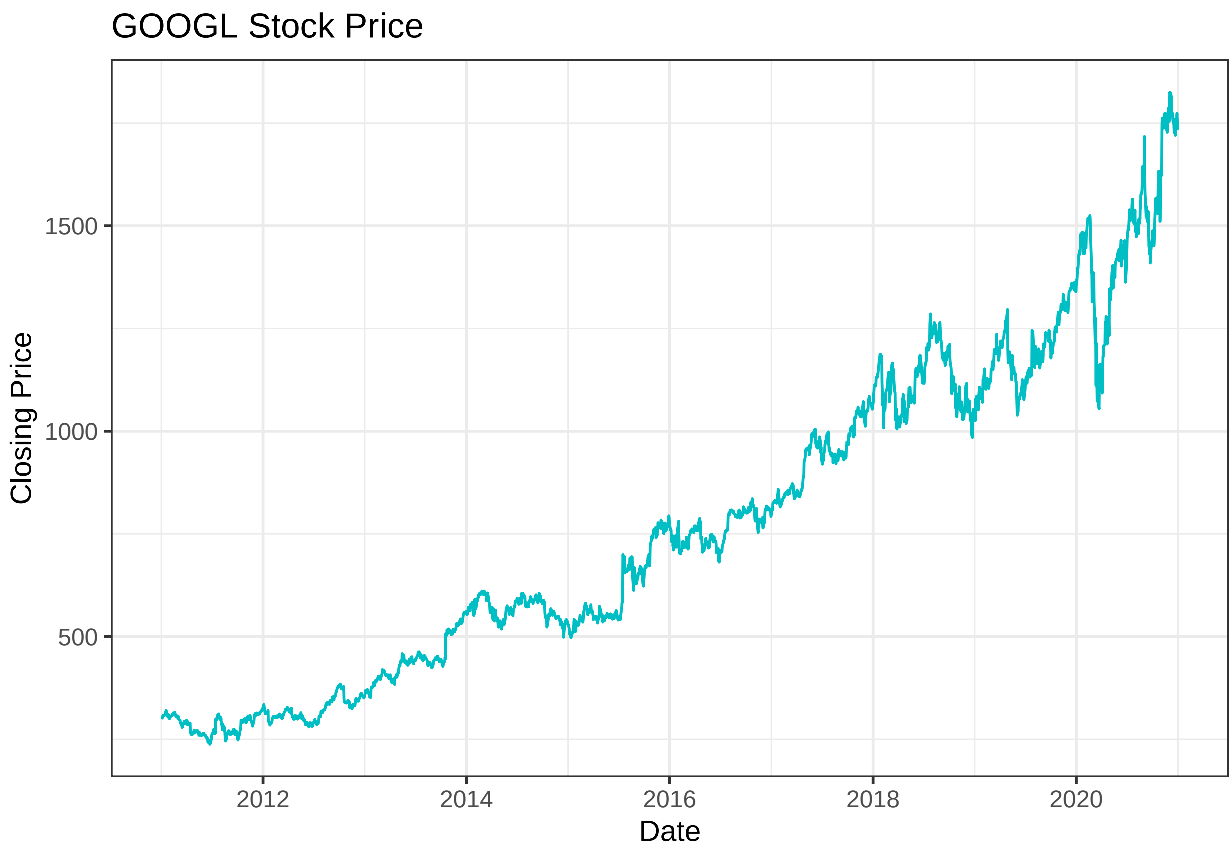
```
round((365.25 * 10) * 5 / 7 - 9 * (10)) # roughly how many days the market was open
```

```
## [1] 2519
```

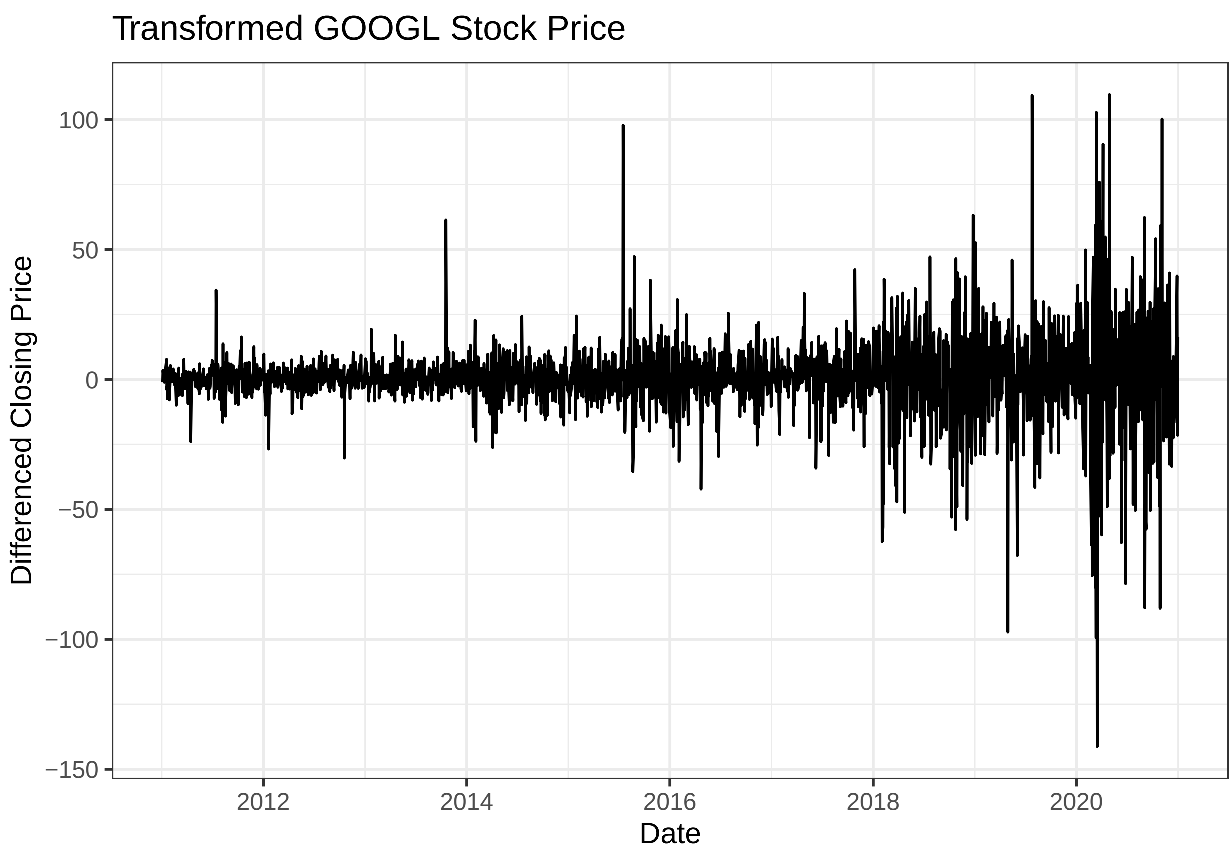
We will load daily 'GOOGL' closing price stock data from 1 January 2011 to 28 June 2021. The closing price will be transformed as described in our proposal. The data will then be split into training and validation sets. Lastly we check the number of observations from a rough formula of open stock market days over a 10 year stretch.

Exploratory Data Analysis

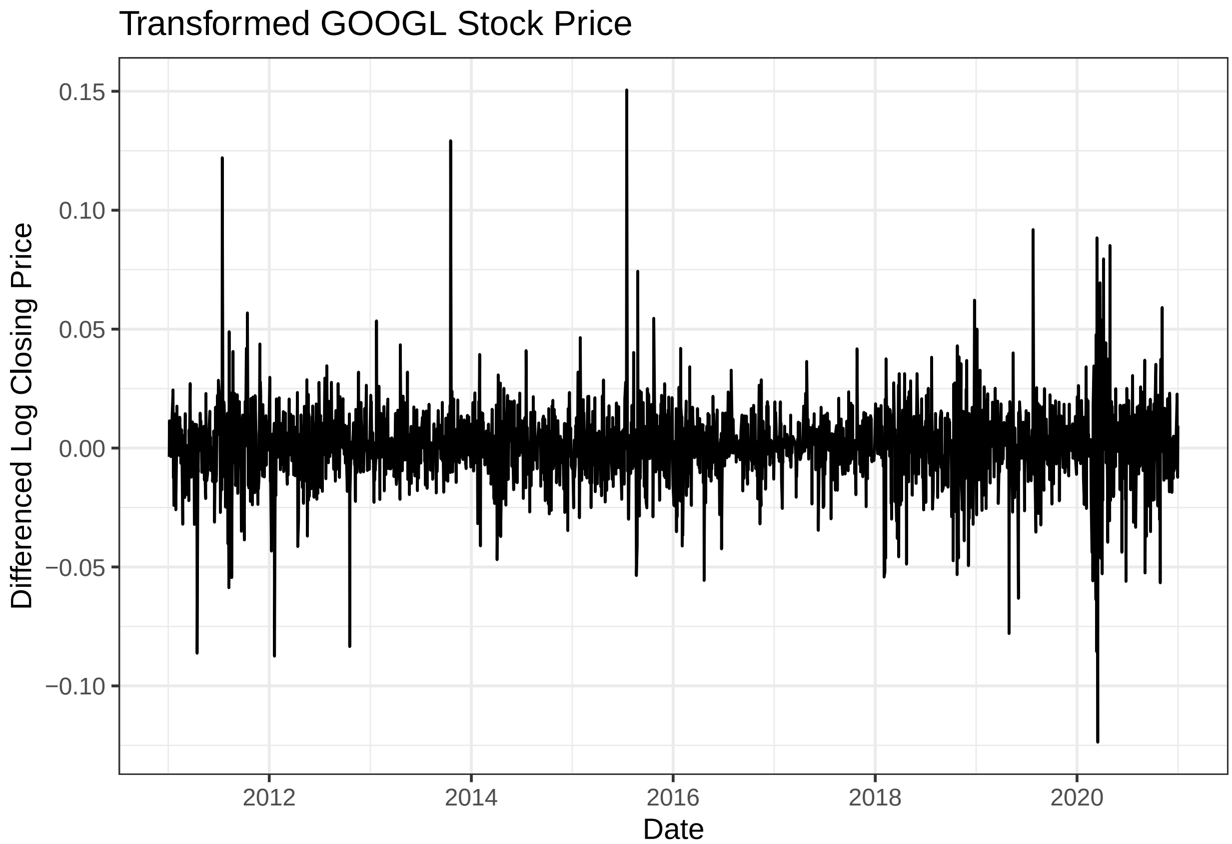
```r
# Plot data
ggplot(data = df.st, aes(x = date)) +
  geom_line(aes(y = close), color = "#00BFC4") +
  labs(x = "Date", y = "Closing Price", title = "GOOGL Stock Price") +
  theme_bw() +
  theme(legend.title = element_blank())
```




```r
# Find differenced data
dif <- diff(df.st$close)
log.dif <- diff(df.st$st.close)

df.dif <- df.st %>%
  slice(-1) %>%
  mutate(dif, log.dif)

# Plot differenced log transformed data
ggplot(data = df.dif, aes(x = date)) +
  geom_line(aes(y = dif)) +
  labs(x = "Date", y = "Differenced Closing Price",
       title = "Transformed GOOGL Stock Price") +
  theme_bw() +
  theme(legend.title = element_blank())
```




```r
# Plot differenced log transformed data
ggplot(data = df.dif, aes(x = date)) +
  geom_line(aes(y = log.dif)) +
  labs(x = "Date", y = "Differenced Log Closing Price",
       title = "Transformed GOOGL Stock Price") +
  theme_bw() +
  theme(legend.title = element_blank())
```



By visualizing the closing price, we can see there is an underlying trend in the data. Checking the differenced data over time, we can still see that the magnitude will increase over time as the stock price increases exponentially. A log transformation will mostly remove this trend rendering our data stationary.

Model fitting

```
# Fit bsts model
ss <- AddAr(y = df.st$st.close)
mod.bts <- bsts(df.st$st.close, state.specification = ss, niter = 1000,
                seed = 610)
```

```
## =-=-=-=-= Iteration 0 Fri Dec 10 14:25:09 2021
##  =-=-=-=-=
## =-=-=-=-= Iteration 100 Fri Dec 10 14:25:11 2021
##  =-=-=-=-=
## =-=-=-=-= Iteration 200 Fri Dec 10 14:25:14 2021
##  =-=-=-=-=
## =-=-=-=-= Iteration 300 Fri Dec 10 14:25:16 2021
##  =-=-=-=-=
## =-=-=-=-= Iteration 400 Fri Dec 10 14:25:18 2021
##  =-=-=-=-=
## =-=-=-=-= Iteration 500 Fri Dec 10 14:25:20 2021
##  =-=-=-=-=
## =-=-=-=-= Iteration 600 Fri Dec 10 14:25:22 2021
```

```
##  =-=-=-=-=
## =-=-=-=-= Iteration 700 Fri Dec 10 14:25:24 2021
##  =-=-=-=-=
## =-=-=-=-= Iteration 800 Fri Dec 10 14:25:26 2021
##  =-=-=-=-=
## =-=-=-=-= Iteration 900 Fri Dec 10 14:25:28 2021
##  =-=-=-=-=
```

```
# Extracting model
phi <- mean(mod.bts$AR1.coefficients)
sigma <- mean(mod.bts$AR1.sigma)

# Discard first 'n' MCMC iterations
burn <- SuggestBurn(0.1, mod.bts)
```

We fit the model on the 10 year stretch of training data. We will use 1000 MCMC iterations. To ensure a better starting value it is recommended that we find the suggested first 'n' iterations to 'burn' or discard. We will extract the parameters to compare bayesian structural methods 'bsts' to Gibbs sampling 'JAGS'. Our bayesian structural posterior results in $\phi = 0.998$ and $\sigma = 0.023$.

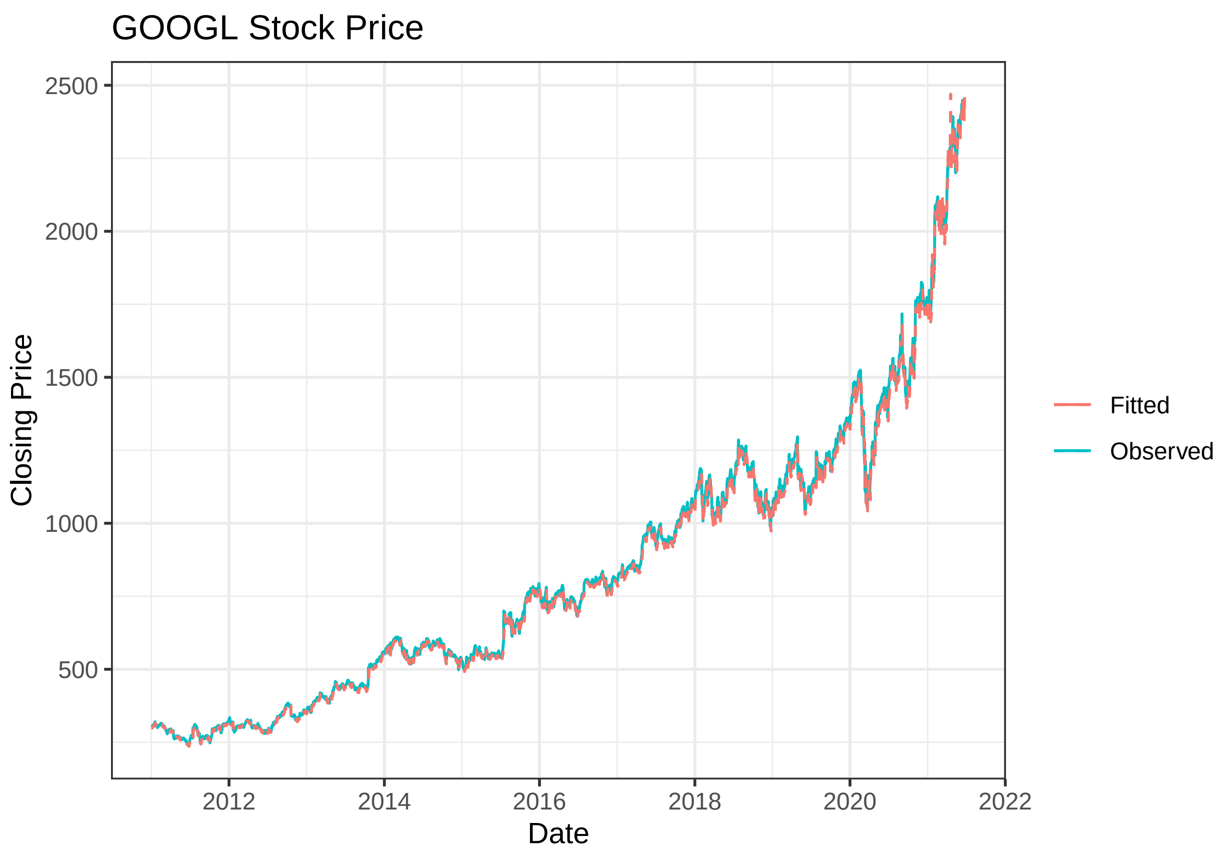
Model Output Evaluation

```
# Save predictions
preds <- predict.bsts(mod.bts, newdata = df.test, horizon = 122, burn = burn,
                      quantiles = c(0.025, 0.975))

# Transform back to compare predicted vs fitted values
steps = -colMeans(mod.bts$one.step.prediction.errors[-(1:burn),])

df.out <- df %>%
  mutate(fitted = exp(steps + df$st.close)) %>%
  dplyr::select(date, close, st.close, fitted) %>%
  slice(-1, -2518) # remove first observation for each set since no lag data

# Validation set
df.val <- df.out %>%
  filter(year(date) > 2020)

# Mean Absolute Percentage Error
MAPE <- mean(abs(df.val$close - df.val$fitted) / df.val$close)
```

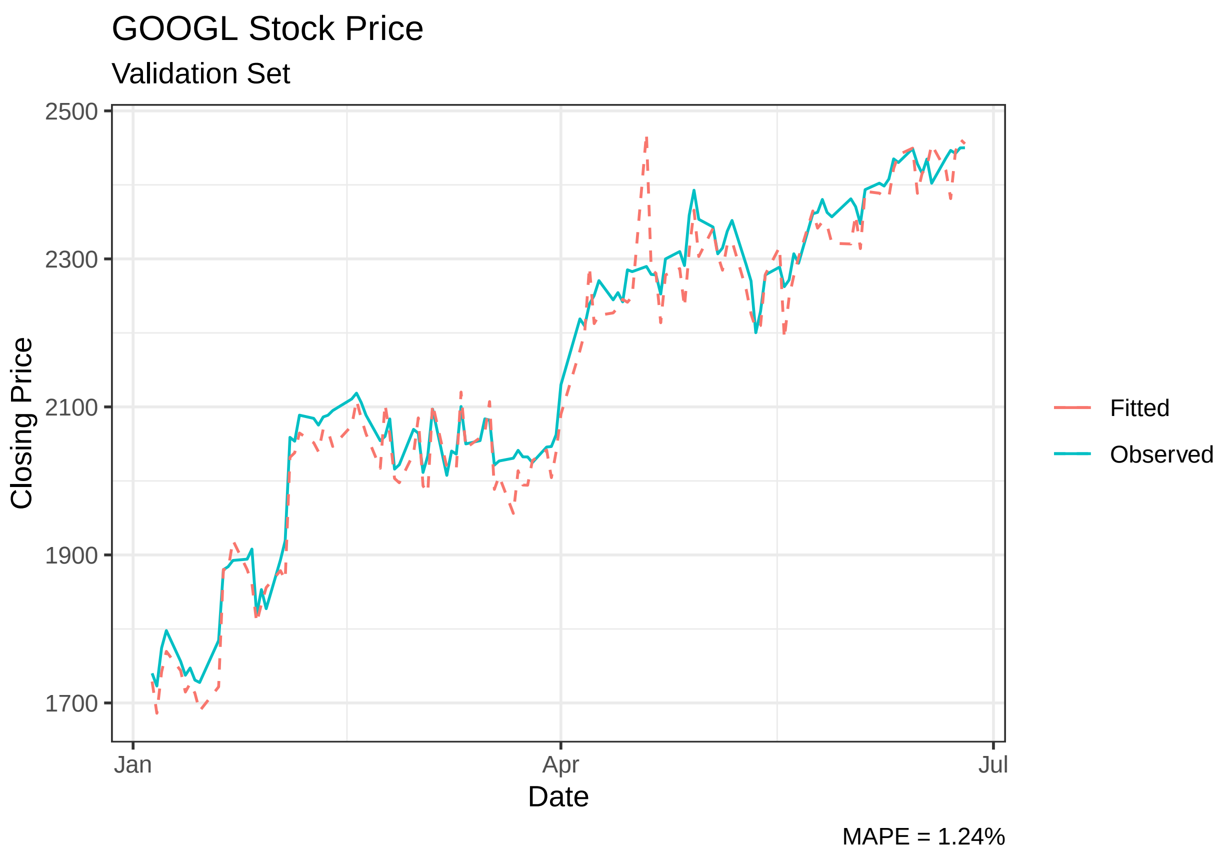
Use the fitted model to predict closing prices using the validation set. To compare frequentist and bayesian time series models we will calculate mean absolute percentage error (MAPE). While auto regressive will fit data closely, forecasting far ahead without data can result in wide and unusable CIs. With that in mind, the bayesian model results in MAPE = 1.23% when predicting values for the validation set.

Plotting Results

```r
# Plot data & predictions
ggplot(data = df.out, aes(x = date)) +
  geom_line(aes(y = close, color = "Observed")) +
  geom_line(aes(y = fitted, color = "Fitted"), linetype = 2) +
  labs(x = "Date", y = "Closing Price", title = "GOOGL Stock Price") +
  theme_bw() +
  theme(legend.title = element_blank())
```




```r
# Plot validation data & predictions
ggplot(data = df.val, aes(x = date)) +
  geom_line(aes(y = close, color = "Observed")) +
  geom_line(aes(y = fitted, color = "Fitted"), linetype = 2) +
  labs(x = "Date", y = "Closing Price", title = "GOOGL Stock Price",
       subtitle = "Validation Set", caption = "MAPE = 1.24%") +
  theme_bw() +
  theme(legend.title = element_blank())
```



GOOGL Stock Price
Validation Set
2500
2300
2100
1900
1700
Closing Price
Jan
Apr
Jul
Date
Fitted
Observed
MAPE = 1.24%