FOREX Volatility

Introduction

The foreign exchange (FOREX) market is one of the most liquid financial markets in the world, where currency exchange rates fluctuate due to a number of economic and political factors. This project focuses on analyzing weekly percent changes in the USD/TRY exchange rate, which represents the value of the Turkish Lira (TRY) against the US Dollar (USD). The Turkish Lira is historically known for its high volatility due to political instability, unpredictable inflation, and other economic pressures. Understanding the behavior of USD/TRY exchange rate is critical for identifying periods of heightened risk or opportunity within the market.

This paper focuses on the following question: can we predict volatility of the USD/TRY FOREX rate based on the date, open price, close price, high price for the week, low price, 10Y treasury rate, buying power of the USD and the S&P 500?

Data Exploration

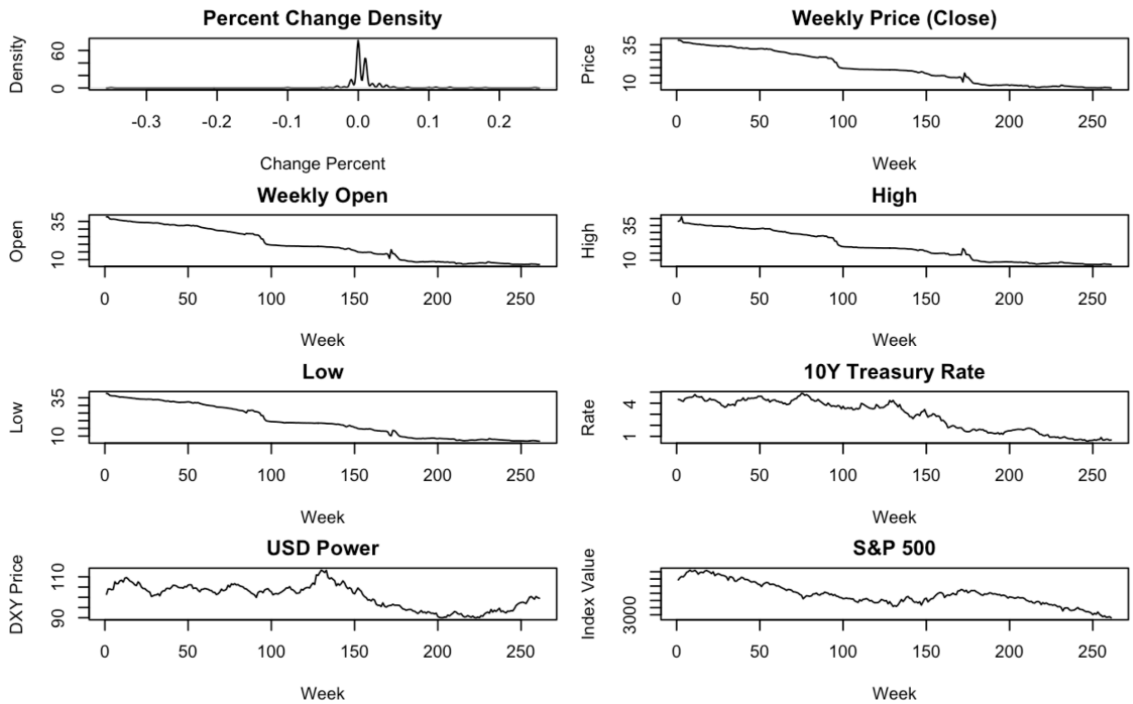
We will start by presenting a summary of the chosen data in tabular form, including the variable name (as seen in the code), a brief description, the data type, and the variable's mean.

Variable	Description	Data Type	Mean
`Date`	MM/DD/YYYY	Factor	None
`ChangePercent`	Percent change in USD/TRY	Numeric	0.007969
`LagChangePercent`	Percent change in USD/TRY from previous week	Numeric	0.007969
`Open`	Opening price of USD/TRY for the week	Numeric	19.3379
`High`	Highest price reached during the week	Numeric	19.6928
`Low`	Lowest price reached during the week	Numeric	19.13592
`Price`	Closing price of USD/TRY for the week	Numeric	19.43924
`DGS10`	10-year U.S. Treasury yield	Numeric	2.87843

Variable	Description	Data Type	Mean
`DXYPrice`	U.S. Dollar Index strength	Numeric	100.4202
`SP500`	S&P 500 stock index level	Numeric	4412.283

Note: The `ChangePercent` is calculated by: $\frac{\text{current price - previous closing price}}{\text{previous closing price}}$

Visual Summaries



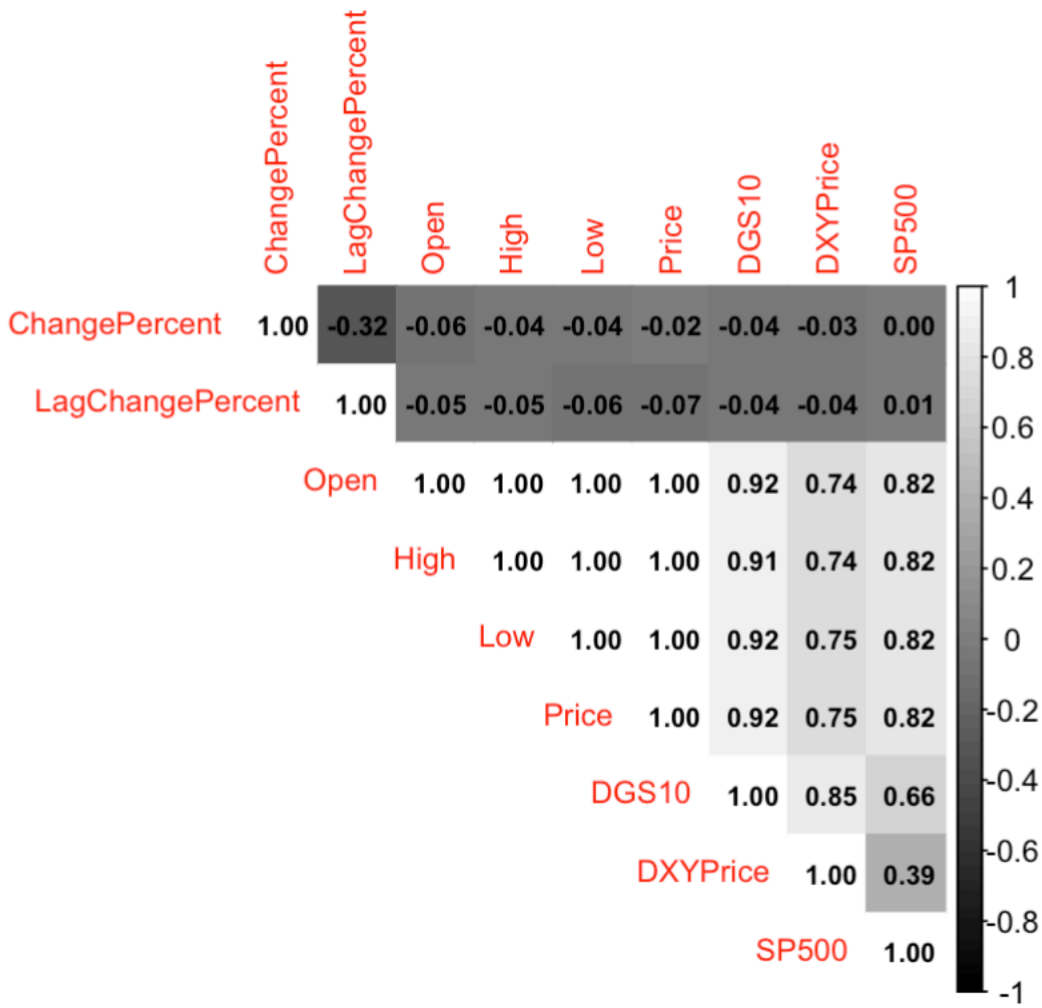
Based on the visual summaries, it is important to note that periods of large movements (high volatility) tend to cluster together. This is a strong indication that our data is time-dependent and hence volatility observations will be highly correlated. This notion will be formalized in the **Final Model** section.

Another pattern to notice is that `Open`, `Close`, `High`, and `Low` provide very similar structures. This could indicate that they provide very similar information and my not all need to be included in our final model. This will be revisited in the **Model Fitting & Evaluations** section.

Model Fitting & Evaluations

Please reference **Appendix B** for code associated to Model Fitting and Evaluations.

Given the observed clustering of volatility in USD/TRY exchange rate, an AR(1) - Order One Autoregressive Model - was incorporated by including a new predictor, `LagChangePercent`. Time series financial data is frequently highly correlated. That is, periods of high volatility tend to group together and hence our observations are not independent. If we were to ignore this lack of independence, we would get highly biased estimates with understated uncertainty. But, the AR(1) model ideally captures these dynamics making this a more realistic model for forecasting short-term movements. To confirm our intuition of variable correlation, the heatmap below indicates strong correlation between `Open`, `High`, `Low`, `Price`, and even `DGS10`.

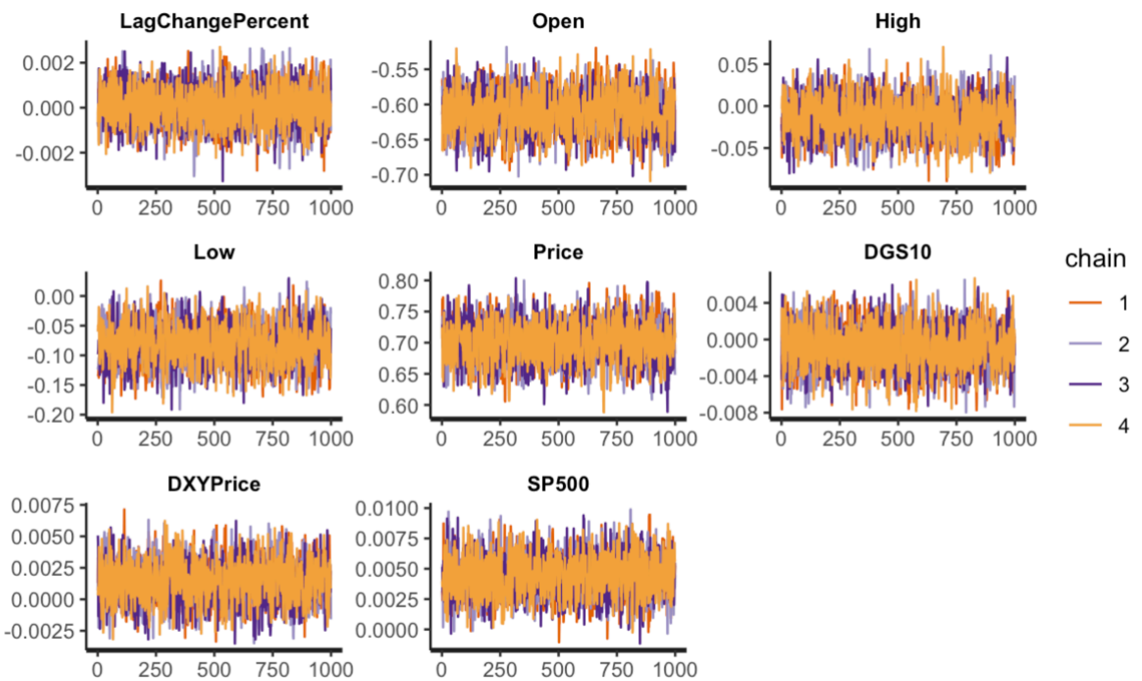


I fit three MCMC models to understand how different combinations of predictors influence model performance. These models parameters were chosen in hopes of removing highly correlated variables. All models listed below assume normal errors for `PercentChange` and weakly informative priors for the model parameters ($\sim N(0,1)$). The priors were chosen to leverage the benefits of the Bayesian framework while still allowing data to "speak for itself". The first model included all available predictors, while models 2 and 3 used smaller subsets of predictors which are mentioned in the table below. After model fitting, I used WAIC and LOOIC to determine which model had the best performance. The WAIC and LOOIC results follow.

	LOOIC	WAIC
Model 1 (full model)	0.0	0.0
Model 3 (`LagChangePercent`, `Price`)	-285.2	-289.4
Model 2 (`LagChangePercent`, `DGS10`, `Price`)	-287.0	-290.3

Based on the previous table, WAIC and LOOIC both strongly favor Model 1. The "next best model" has a LOOIC difference of 285.2 and WAIC difference of 289.4. Based on this evidence, Model 1 was selected as the final model for interpretation and further analysis. We will now assess MCMC convergence.

To determine if Model 1 converges properly, we will look at traceplots, the Gelman-Rubin Statistic $\hat{R}$, and effective sample size. Note again that `LagChangePercent` incorporates `ChangePercent` and `Date` in order to implement the AR(1) algorithm and hence we have 8 variables for which we will look at MCMC Convergence Diagnostics.



The trace plots for Model 1 show good MCMC chain mixing for all 4 chains and we don't have any evidence to believe our model may diverge.

Secondly, we can assess convergence by looking at the Gelman-Rubin Diagnostic for the predictors in the model.

Variable	Point Estimate	Upper CI
`LagChangePercent`	1	1.00
`Open`	1	1.01
`High`	1	1.01
`Low`	1	1.01
`Price`	1	1.01
`DGS10`	1	1.00
`DXYPrice`	1	1.01
`SP500`	1	1.00

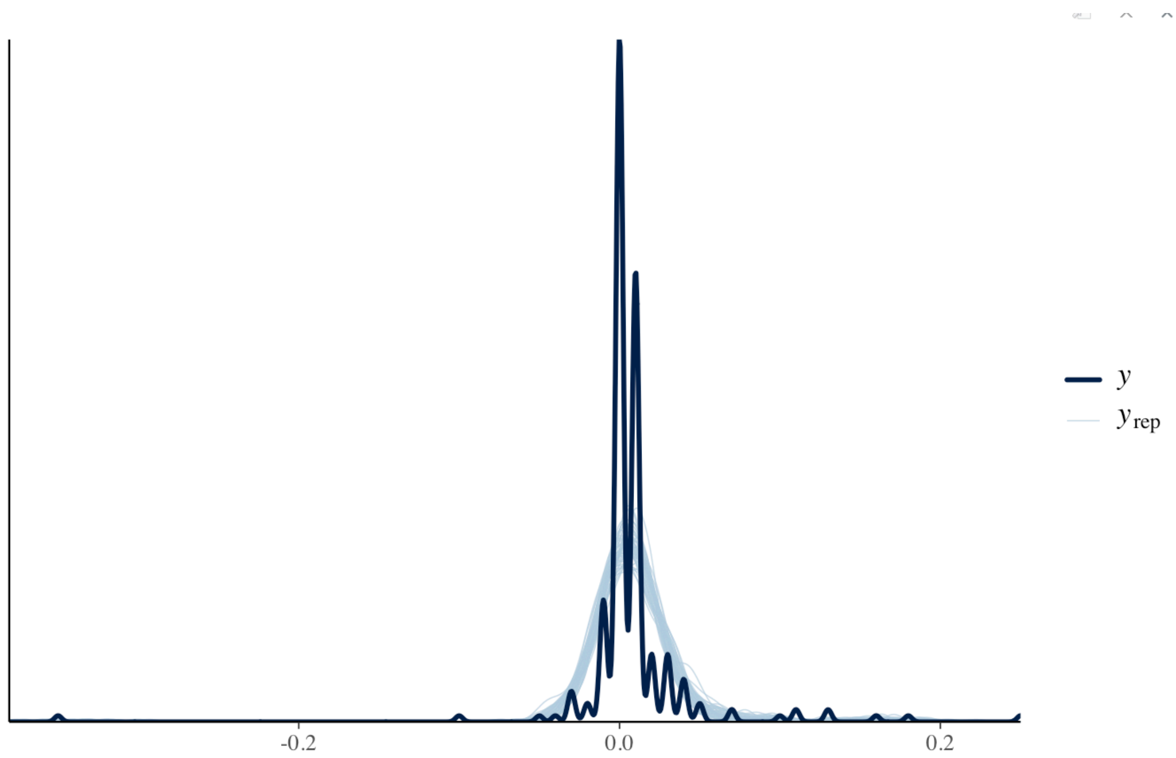
All $\hat{R}$ point estimates are all 1 and all Upper Confidence Intervals are ≤ 1.01 which is a great outcome of this diagnostic test. This indicates we have convergence in our model and we have excellent chain mixing and convergence.

Final, we can analyze Effective Sample Size to decide if our posterior is based on reliable sampling. Below are the results.

Variable	Effective Sample Size
`LagChangePercent`	3282.5
`Open`	1749.0
`High`	1798.5
`Low`	1454.8
`Price`	1637.1
`DGS10`	2933.52
`DXYPrice`	2429.6
`SP500`	2771.5

Our effective sample size for each variable is over 1400, indicating that our posterior is generated through reliable sampling. The combined success of trace plots, Gelman-Rubin $\hat{R}$, and Effective Sample Size indicates that our model accurately converged.

Finally, we must test if our model fits the data well. Since the model was fit from `rstanarm`, we can leverage the function `pp_check` to see how well our model fits the data.



Within the graphic above, the replicated datasets, y_{rep}, are represented in light blue and the observed data is represented in dark blue. The goal of y_{rep} is to mimic the distribution of y. The model (y_{rep}) seems to capture how spread out the data is. However, our model underestimates how concentrated the real data is at its typical values. That is, the peak around the mean of y is much higher than the peak of y_{rep}. This could be caused by USD/TRY percent changes clustering around small moves with few weeks that are extreme. This is common in financial data and does not strongly indicate that our model has a deficiency.

Ultimately, we can conclude that our model fits the data well, converges, and depicts proper MCMC mixing.

Final Model

The final fitted model predicts the weekly percent change in the USD/TRY exchange rate using autoregressive and macroeconomic framework. The response variable is `ChangePercent`, which represents the weekly percent change in the USD/TRY currency pair. Our final model includes the following predictors:

`LagChangePercent`: The percent change from the previous week, included to account for autocorrelation in volatility.

`Open`: The opening exchange rate for the week.

`High`: The highest exchange rate observed during the week.

`Low`: The lowest exchange rate observed during the week.

`Price`: The closing exchange rate for the week.

`DGS10`: The yield on the 10 year US Treasury bond.

`DXYPrice`: The value of the US Dollar index (DXY), measuring the strength of the US Dollar.

`SP500`: The level of the S&P 500 index.

The model assumes normal errors for the residuals and applies weakly informative priors for the regression coefficients. An autoregressive structure is induced by including the lagged outcome `LagChangePercent` as a predictor allowing the model to capture time dependence in exchange rate movements. Posterior predictive checks indicate the model adequately captures the variability and central tendency of the observed data, supporting its use for interpretation and forecasting.

Coefficient Interpretation & Wrap-Up

To understand the impact of each variable present in our final model, we can summarize the mean, standard deviation and 95% bayesian credible interval.

Variable	Mean	SD	0.025	0.975
β_0	0.008	0.00097	0.0061	0.010
`LagChangePercent`	0.0004	0.0010	-0.0015	0.0024
`Open`	-0.587	0.0352	-0.655	-0.521
`High`	-0.023	0.0283	-0.079	0.0344
`Low`	-0.068	0.0406	-0.147	0.0136
`Price`	0.679	0.0379	0.604	0.7513
`DGS10`	-0.0025	0.0032	-0.0088	0.0036
`DXYPrice`	-0.00073	0.0022	-0.0049	0.0036
`SP500`	0.0023	0.0022	-0.0021	0.0065
σ	0.0159	0.00074	0.0145	0.0174

The previous table depicts the mean, standard deviation, and 95% bayesian credible interval. This is important because it indicates the importance of a predictor for the output of the model. If 0 is included in the 95% credible interval, then we cannot confidently say the predictor matters. If 0 is outside the 95% credible interval, then we can conclude the predictor is important. Here is a brief summary of the importance and interpretation of each predictor.

Variable	Interpretation	Importance
$\beta_0 = 0.008$	This is the intercept, which is the expected weekly percent change when all predictors are set to 0.	Low; Mean is 0.008 and CI doesn't contain 0 but is close, indicating low impact.
`LagChangePercent` $= 0.004$	For every percentage point increase in the previous week's percent change, the current week's percent change is expected to increase by 0.004 when all other variables are held constant.	Low; Mean is 0.0004 and CI contains 0 indicating low impact.
`Open` $= -0.587$	For every unit increase in the weekly opening price of USD/TRY, the weekly percent change is expected to decrease by 0.587 percentage points.	High; CI does not contain 0 and the coefficient magnitude is 0.587 indicating high impact on percent change.

Variable	Interpretation	Importance
`High` $= -0.023$	For every dollar the weekly high increases, the percent change will decrease by 0.023 percentage points when all other variables are held constant.	Low; 0.023 is very close to 0 and the CI contains 0, indicating low impact on percent change.
`Low` $= -0.068$	When the weekly low increases by one unit, the percent change is expected to decrease by 0.068 percentage points when all other variables are held constant.	Low; -0.068 is near 0 and the CI contains 0 indicating low impact on percent change.
`Price` $= 0.679$	For every unit increase in weekly closing price, the weekly percent change is expected to increase by 0.679 percentage points when all other variables are held constant.	High; The magnitude of the `Price` coefficient and the absence of 0 in the CI indicates high importance of the variable in our model.
`DGS10` $= -0.0025$	For every unit increase in the 10 year treasury yield, the weekly percent change is expected to decrease by 0.0025 when all other variables are held constant.	Low; the magnitude of the coefficient is small and the CI contains 0, indicating low importance.
`DXYPrice` $= -0.00073$	For every unit increase in the DXY index value, the weekly percent change in USD/TRY is expected to decrease by 0.00073 when all other variables are held constant.	Low; The magnitude of the DXY Price coefficient is very small and the CI contains 0, indicating low importance.
`SP500` $= 0.0023$	For every 1 point increase in the S&P500 index, the weekly percent change in USD/TRY is expected to increase by 0.0023 when all other variables are held constant	Medium; the magnitude of the S&P 500 index coefficient is small indicating low importance. However, the CI does not contain 0, so we have a slightly greater chance of impact.

This project modeled the weekly percent change in the USD/TRY exchange rate using Bayesian GLMs. The model incorporated both autoregressive behavior and financial market variables, including weekly opening, high, low, closing prices, US Treasury rates, the US Dollar index, and the S&P 500. Posterior inference indicated the opening and closing prices were the strongest predictors of weekly exchange rate movements. Other variables showed weak or negligible predictive power after adjusting for the primary market price variables.

Posterior predictive checks and MCMC convergence diagnostics confirmed the model fit the data reasonably well. However, the normal error assumption may not fully account for potential outliers observed in financial return data.

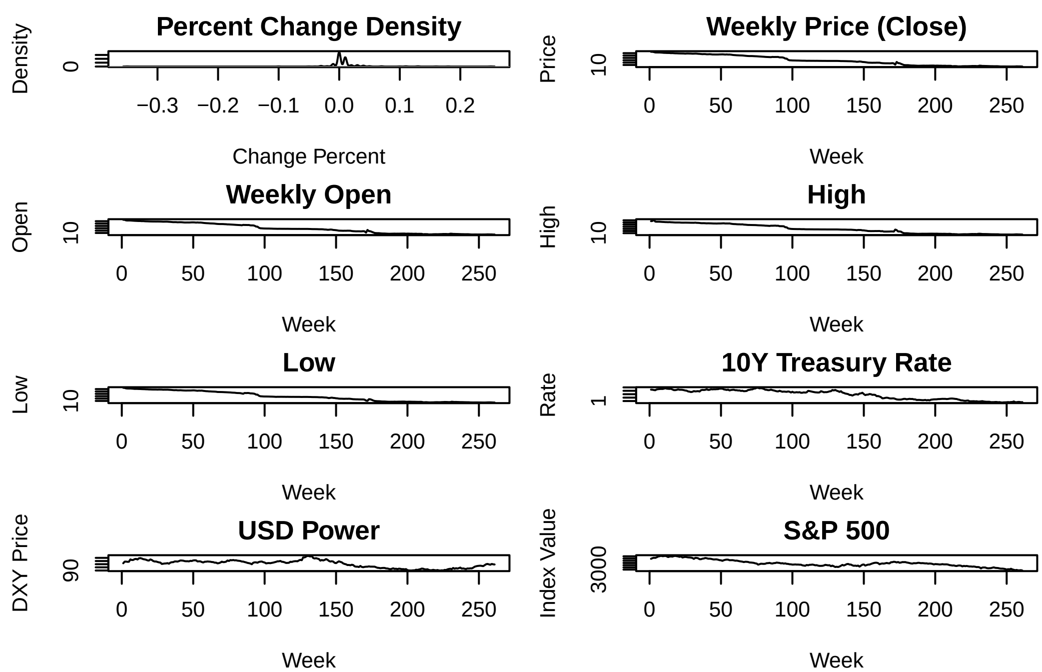
A potential improvement for future work could include switching to a t-distribution likelihood to fit the data even better. In the future, the model could also benefit from including additional financial metrics to more accurately handle variance in exchange rate changes.

Appendix A - Data Exploration

```
data <- read.table("/Users/madisonpoore/Desktop/BayesData.csv",sep=",",header=TRUE,stringsAsI
#summary(data$ChangePercent)
```

```
library(ggplot2)
```

```
par(mfrow = c(4, 2), mar = c(4, 4, 2, 1))
dens <- density(data$ChangePercent)

plot(dens, main = "Percent Change Density", xlab = "Change Percent")
plot(data$Price, type = "l", main = "Weekly Price (Close)", xlab = "Week", ylab = "Price")
plot(data$Open, type = "l", main = "Weekly Open", xlab = "Week", ylab = "Open")
plot(data$High, type = "l", main = "High", xlab = "Week", ylab = "High")
plot(data$Low, type = "l", main = "Low", xlab = "Week", ylab = "Low")
plot(data$DGS10, type = "l", main = "10Y Treasury Rate", xlab = "Week", ylab = "Rate")
plot(data$DXYPrice, type = "l", main = "USD Power", xlab = "Week", ylab = "DXY Price")
plot(data$SP500, type = "l", main = "S&P 500", xlab = "Week", ylab = "Index Value")
```




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

```r
mean(data$Price)
```

```
[1] 19.43924
```

```r
mean(data$Open)
```

```
[1] 19.33797
```

```r
mean(data$High)
```

```
[1] 19.69282
```

```r
mean(data$Low)
```

```
[1] 19.13592
```

```r
mean(data$ChangePercent)
```

```
[1] 0.007969349
```

```r
mean(data$DGS10)
```

```
[1] 2.878429
```

```r
mean(data$DXYPrice)
```

```
[1] 100.4202
```

```r
mean(data$SP500)
```

```
[1] 4412.283
```

```r
#head(data$ChangePercent)
```

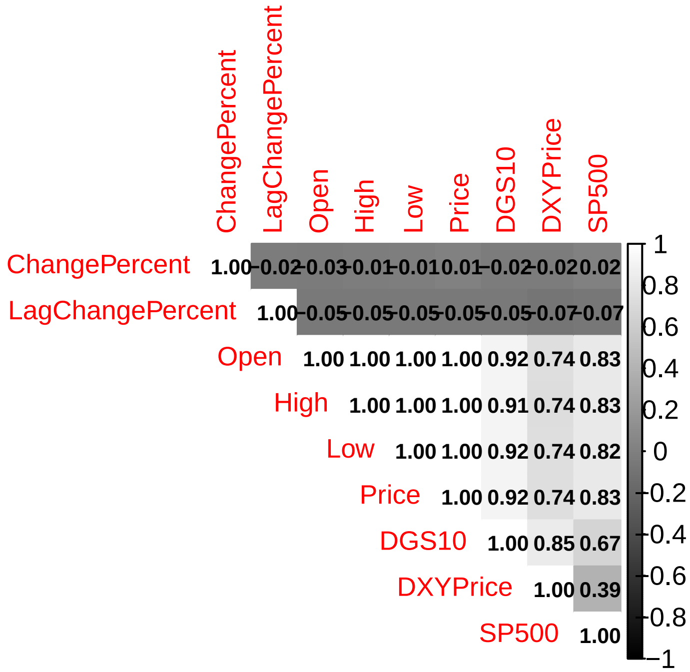
```r
str(data$ChangePercent)
```

```
 num [1:261] 0.16 0.01 0.02 0 0 0 0 0.01 0.01 0 ...
```

Appendix B - Model Fitting & Evaluations

```r
#introducing LagChangePercent
data$ChangePercent <- as.numeric(gsub("%", "", data$ChangePercent))

#here we are introducing our lag variable!!!
#this is maybe the most important step in an AR1, allowing the model
#to see last week's volatility and predict the next time step
#this is especially helpful here because we expect next week to be
#correlated to the previous week's observation
data <- data %>%
  arrange(data$Date) %>%
  mutate(
    LagChangePercent = lag(ChangePercent)
```

```
  ) %>%
  select(ChangePercent, LagChangePercent, Open, High, Low, Price, DGS10, DXYPrice, SP500) %>%
  drop_na()
X <- scale(select(data, -ChangePercent))
model_data <- as.data.frame(cbind(ChangePercent = data$ChangePercent, X))
#generating correlation heatmap

cor_vars <- data %>%
  select(ChangePercent, LagChangePercent, Open, High, Low, Price, DGS10, DXYPrice, SP500)

cor_matrix <- cor(cor_vars, use = "complete.obs")

#correlations in grayscale just in case!
gray_colors <- colorRampPalette(c("black", "white"))(200)
corrplot(cor_matrix, method = "color", type = "upper",
         addCoef.col = "black",
         tl.cex = 0.8,
         number.cex = 0.7,
         col = gray_colors)
```




```
#in this code block we will fit multiple models and select the best one
#full model
```

```r
model1 <- stan_glm(
  ChangePercent ~ LagChangePercent + Open + High + Low + Price + DGS10 + DXYPrice + SP500,
  data = model_data,
  family = gaussian(),
  chains = 4,
  iter = 2000,
  seed = 42,
  refresh=0
)

#model 2 (lag change percent, price, DGS10)
model2 <- stan_glm(
  ChangePercent ~ LagChangePercent + Price + DGS10,
  data = model_data,
  family = gaussian(),
  chains = 4,
  iter = 2000,
  seed = 42,
  refresh=0
)

#model 3 (lag and price)
model3 <- stan_glm(
  ChangePercent ~ LagChangePercent + Price,
  data = model_data,
  family = gaussian(),
  chains = 4,
  iter = 2000,
  seed = 42,
  refresh=0
)
```

```r
#choosing between models
#waic
waic1 <- waic(model1)
```

```
Warning:
9 (3.5%) p_waic estimates greater than 0.4. We recommend trying loo instead.
```

```r
waic2 <- waic(model2)
```

```
Warning:
```

```
4 (1.5%) p_waic estimates greater than 0.4. We recommend trying loo instead.
```

```
waic3 <- waic(model3)
```

```
Warning:
4 (1.5%) p_waic estimates greater than 0.4. We recommend trying loo instead.
```

```
#looic
loo1 <- loo(model1)
```

```
Warning: Found 2 observation(s) with a pareto_k > 0.7. We recommend calling 'loo' again with
```

```
loo2 <- loo(model2)
```

```
Warning: Found 1 observation(s) with a pareto_k > 0.7. We recommend calling 'loo' again with
```

```
loo3 <- loo(model3)
```

```
Warning: Found 2 observation(s) with a pareto_k > 0.7. We recommend calling 'loo' again with
```

```
loo_compare(loo1, loo2, loo3)
```

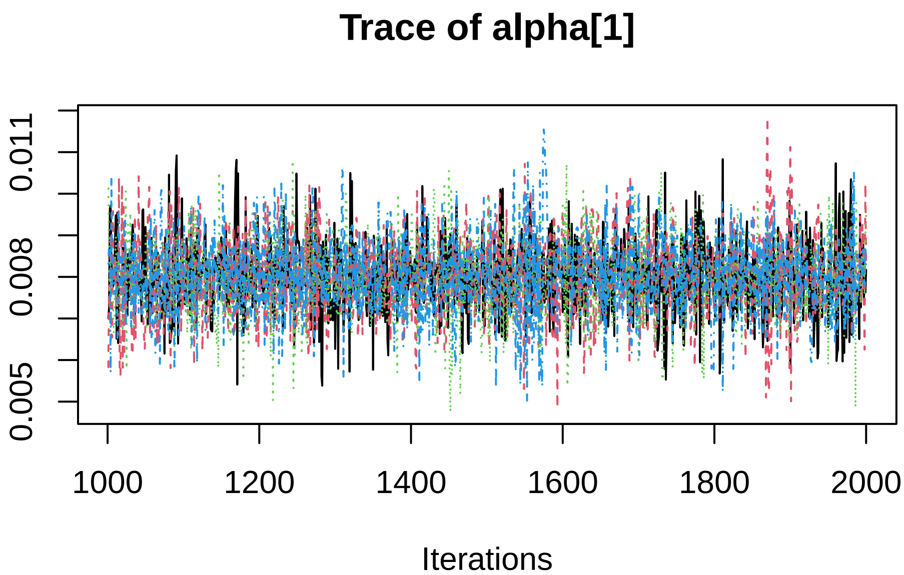
```
       elpd_diff se_diff
model1    0.0       0.0
model2 -216.6      78.6
model3 -218.4      79.1
```

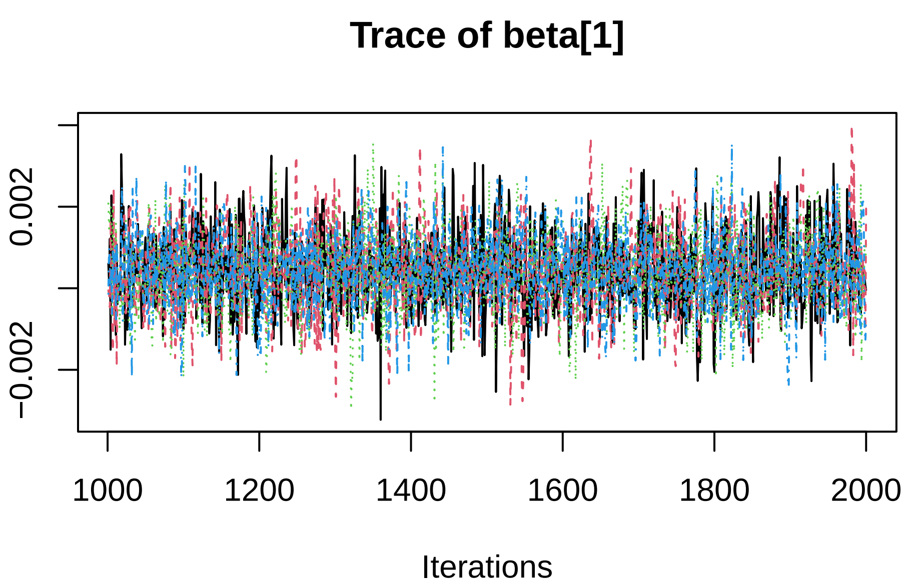
```
loo_compare(waic1, waic2, waic3)
```

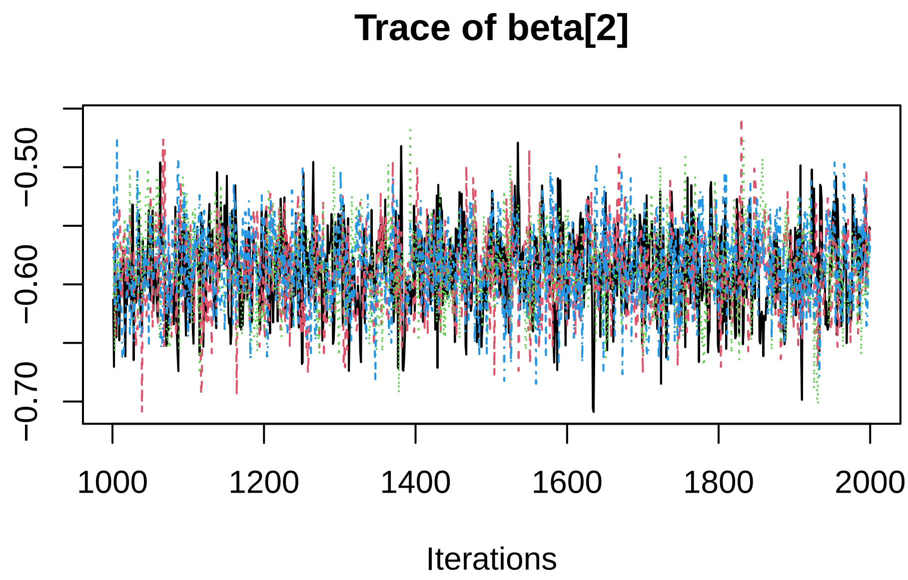
```
       elpd_diff se_diff
model1    0.0       0.0
model2 -213.8      84.7
model3 -214.4      84.4
```

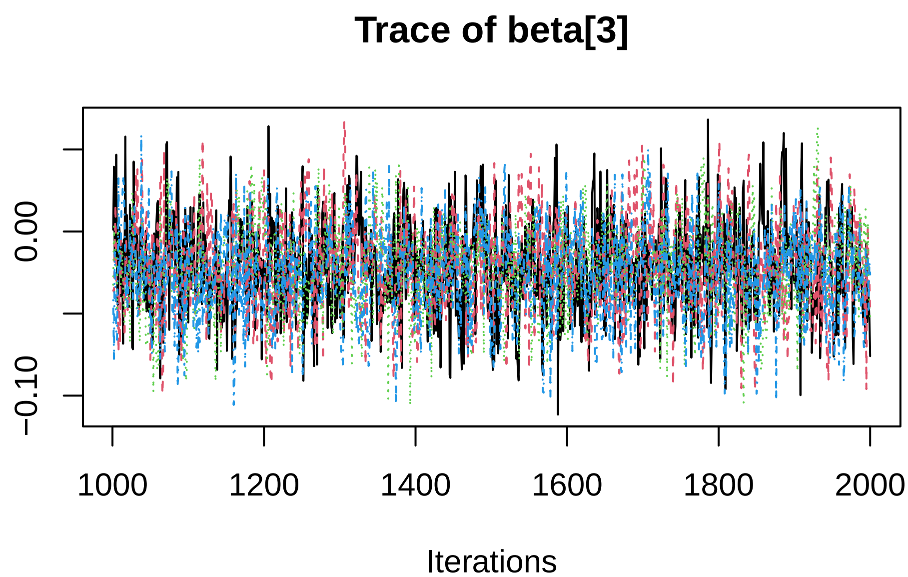
```
#MCMC Convergence

#generating traceplots
#traceplot(model1$stanfit, pars = c("LagChangePercent", "Open", "High", "Low", "Price", "DGS
```

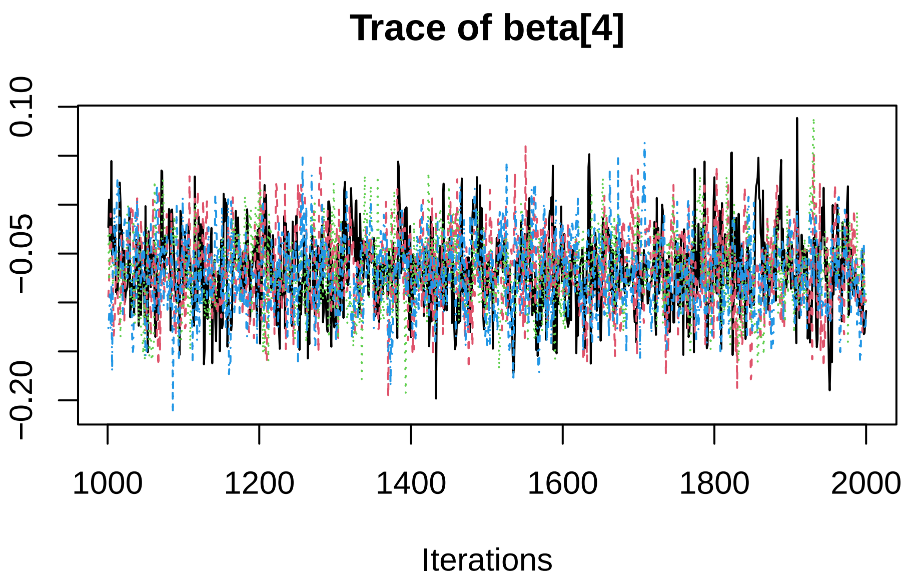
```r
stanfit_object <- model1$stanfit
posterior_mcmc_list <- rstan::As.mcmc.list(stanfit_object)
traceplot(posterior_mcmc_list)
```

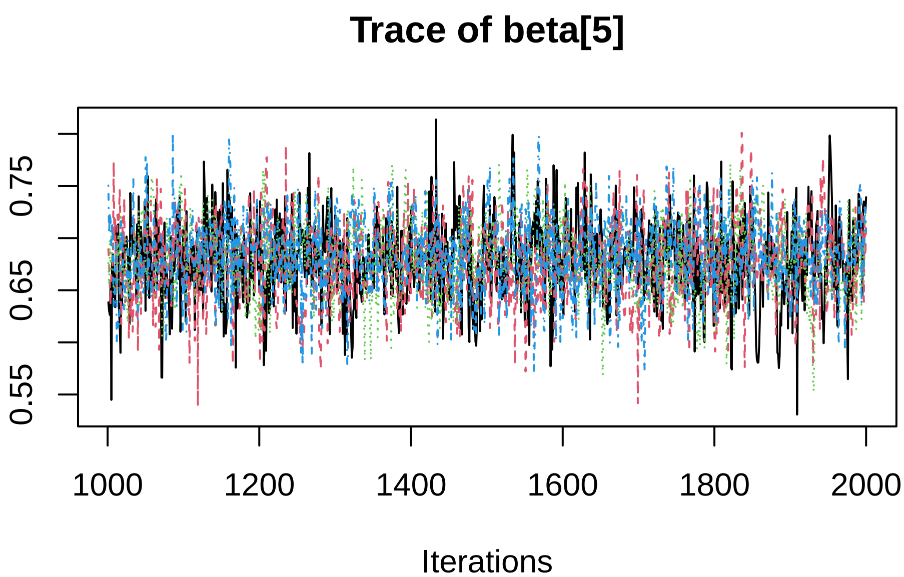


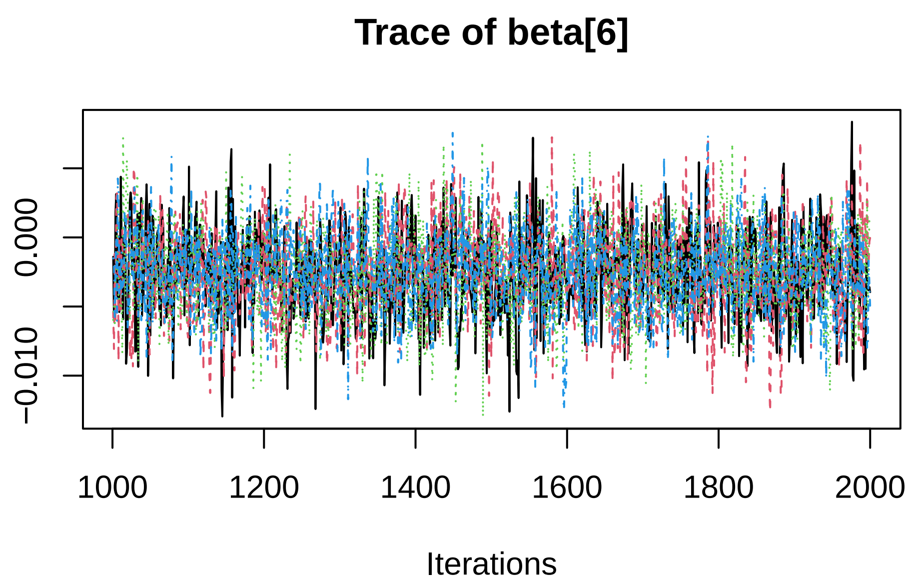


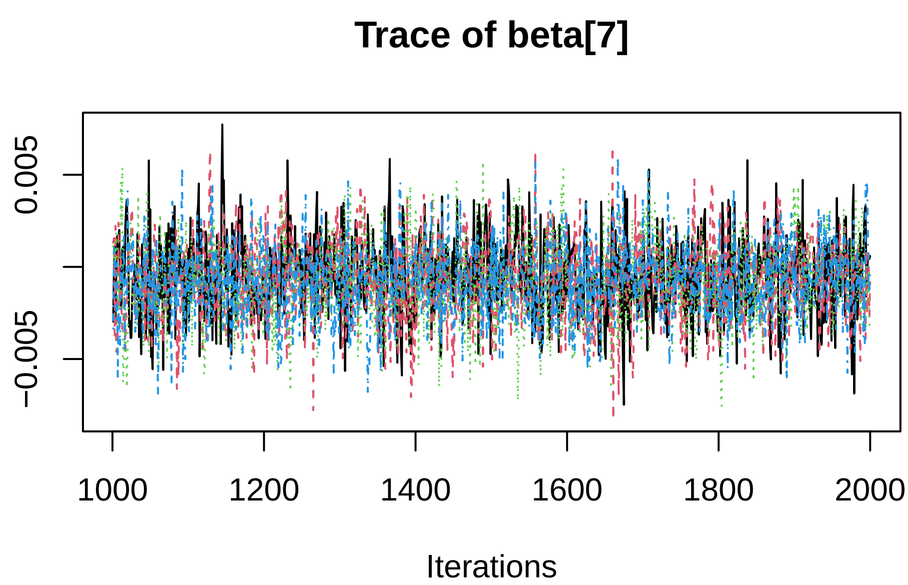

Trace of beta[2]
−0.70
−0.60
−0.50
1000
1200
1400
1600
1800
2000
Iterations

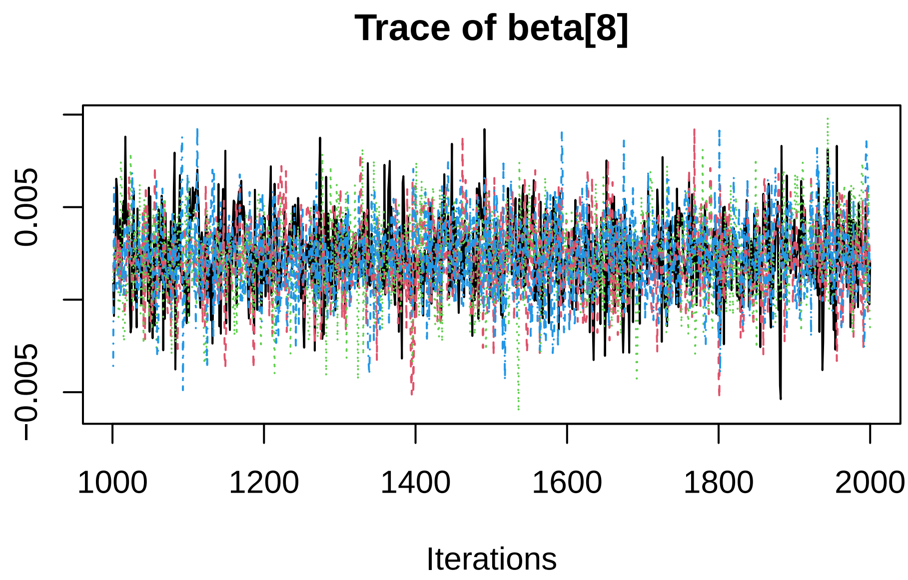

Trace of beta[3]
−0.10
0.00
1000
1200
1400
1600
1800
2000
Iterations

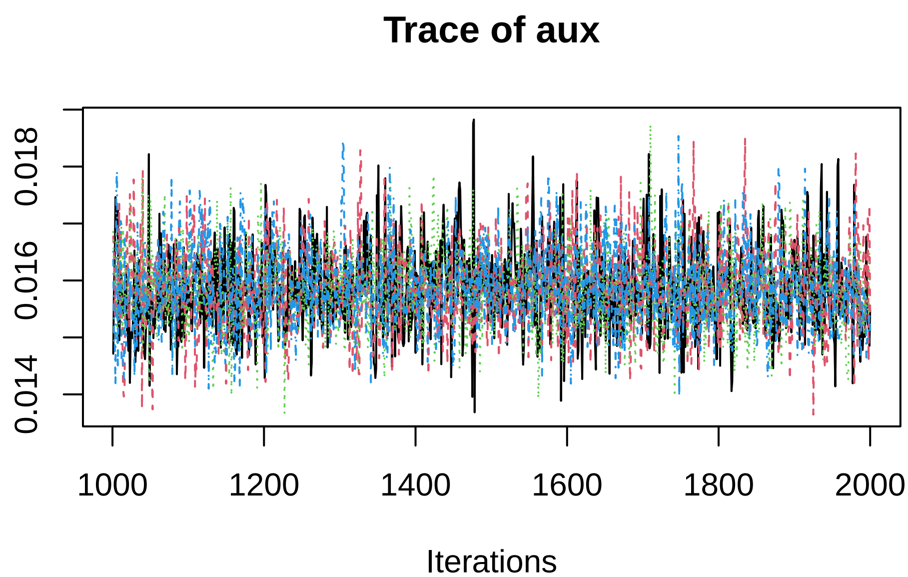

Trace of beta[4]
0.10
−0.05
−0.20
1000
1200
1400
1600
1800
2000
Iterations

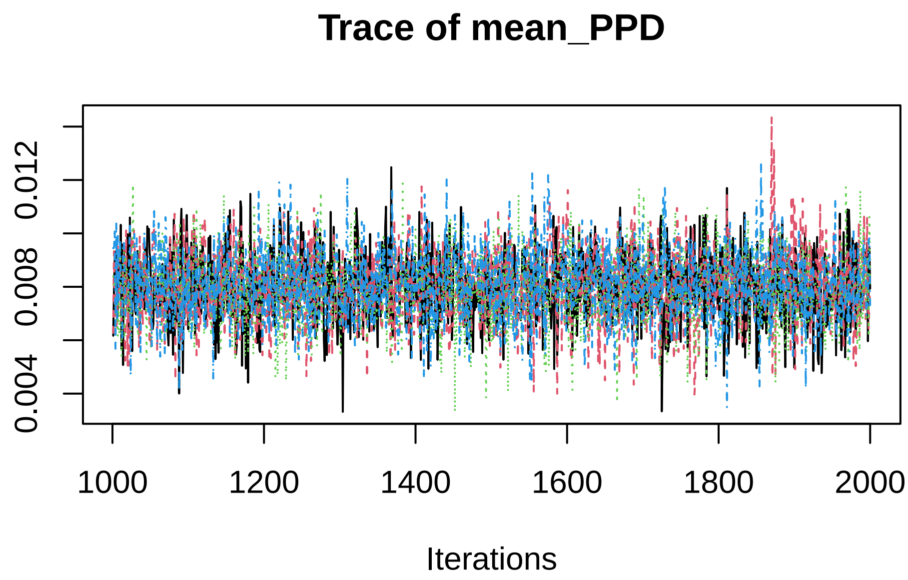

Trace of beta[5]
0.75
0.65
0.55
1000
1200
1400
1600
1800
2000
Iterations

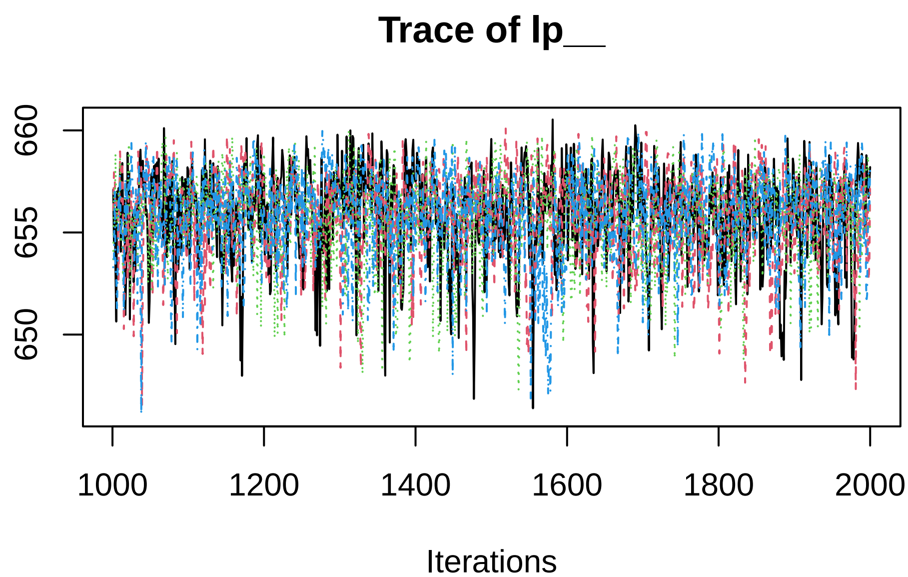

Trace of beta[6]
0.000
−0.010
1000
1200
1400
1600
1800
2000
Iterations


Trace of beta[7]
0.005
−0.005
1000
1200
1400
1600
1800
2000
Iterations


Trace of beta[8]
0.005
−0.005
1000
1200
1400
1600
1800
2000
Iterations


Trace of aux
0.018
0.016
0.014
1000
1200
1400
1600
1800
2000
Iterations






```
gelman.diag(posterior_mcmc_list, autoburnin = FALSE)
```

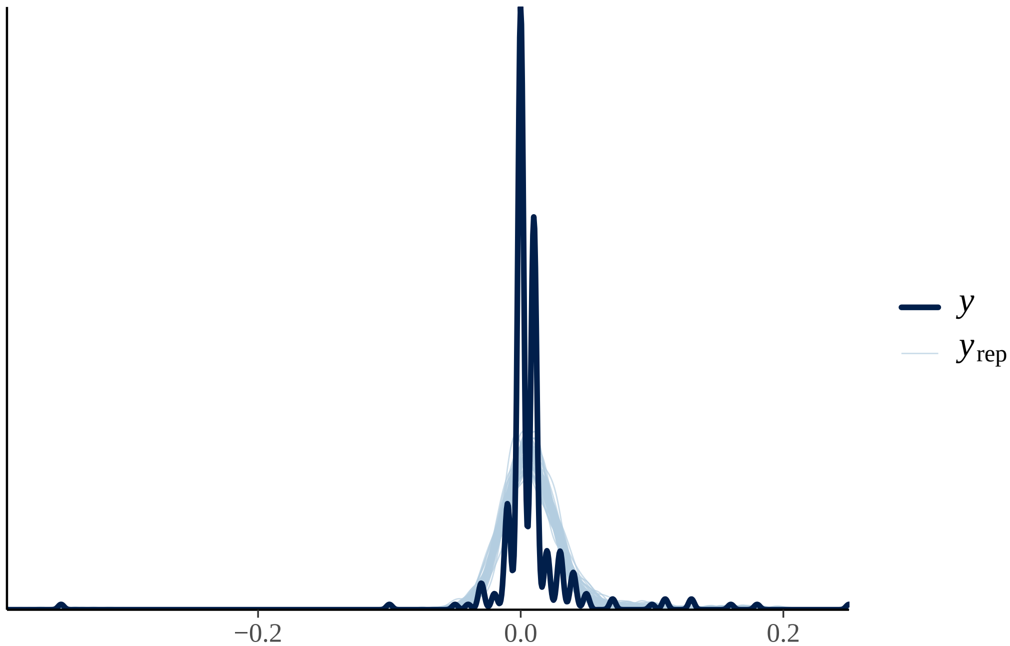
```
Potential scale reduction factors:
```

```
          Point est. Upper C.I.
alpha[1]           1       1.00
beta[1]            1       1.00
beta[2]            1       1.01
beta[3]            1       1.01
beta[4]            1       1.01
beta[5]            1       1.01
beta[6]            1       1.00
beta[7]            1       1.01
beta[8]            1       1.00
aux                1       1.00
mean_PPD           1       1.00
lp__               1       1.00

Multivariate psrf

1
```

```r
effectiveSize(posterior_mcmc_list)
```

```
alpha[1]  beta[1]  beta[2]  beta[3]  beta[4]  beta[5]  beta[6]  beta[7] 
4860.025 3282.513 1749.039 1798.519 1454.760 1637.059 2933.516 2429.596 
 beta[8]      aux mean_PPD     lp__ 
2771.507 3042.787 4357.310 1618.533 
```

```r
pp_check(model1)
```



Appendix C - Coefficients & Wrap-Up

```
summary(model1,digits=6,probs=c(0.025,0.975))
```

```
Model Info:
 function:     stan_glm
 family:       gaussian [identity]
 formula:      ChangePercent ~ LagChangePercent + Open + High + Low + Price +
       DGS10 + DXYPrice + SP500
 algorithm:    sampling
 sample:       4000 (posterior sample size)
 priors:       see help('prior_summary')
 observations: 260
 predictors:   9

Estimates:
                   mean      sd        2.5%      97.5%
(Intercept)       0.008010  0.000970  0.006108  0.009956
LagChangePercent  0.000429  0.001003 -0.001507  0.002386
Open             -0.587264  0.035234 -0.654997 -0.520563
```

```
High            -0.022681  0.028380 -0.079049  0.034492
Low             -0.068268  0.040590 -0.147385  0.013591
Price            0.678822  0.037910  0.603662  0.751342
DGS10           -0.002521  0.003224 -0.008763  0.003610
DXYPrice        -0.000727  0.002170 -0.004947  0.003571
SP500            0.002259  0.002203 -0.002130  0.006515
sigma            0.015855  0.000741  0.014501  0.017439

Fit Diagnostics:
           mean     sd       2.5%     97.5%
mean_PPD 0.007987 0.001391 0.005301 0.010668

The mean_ppd is the sample average posterior predictive distribution of the outcome variable

MCMC diagnostics
                 mcse     Rhat     n_eff
(Intercept)      0.000014 0.999934 4958
LagChangePercent 0.000018 0.999951 3188
Open             0.000851 1.000754 1715
High             0.000700 1.001640 1643
Low              0.001054 1.001348 1482
Price            0.000961 1.001577 1555
DGS10            0.000062 1.000196 2684
DXYPrice         0.000046 1.000967 2210
SP500            0.000042 1.000899 2804
sigma            0.000013 0.999832 3022
mean_PPD         0.000021 1.000822 4461
log-posterior    0.056685 1.000496 1547

For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effectiv
```