DATA 606 Final Project

Sam Reeves

Data Preparation

```
library(zoo)
library(tidyverse)
library(tidymodels)
library(TTR)

rates <- read.csv(
  "https://raw.githubusercontent.com/TheWerefriend/exchange-rate-prediction/master/rates.csv") %>%
  na.locf()

colnames(rates)[colnames(rates) == "X"] <- "date"
rates <- mutate(rates, date=as.Date(date[[1]]))
str(rates)

set.seed(1337)
```

Research question

Is it reasonable to predict the USD/MMK exchange rate using the historical data for all ***/MMK exchange rates? Does the prediction get better when you remove some currencies as independent variables? What if we engineer some technical features like momentum, MACD, and stochastic oscillators?

Cases

Each day (this has halted since the coup), the Central Bank of Myanmar declares an exchange rate for each of 38 currencies which can be used to by Kyats. We have nearly 10 years of data, each day is an observation, and there are 38 variables in each observation.

Data collection

This data was scraped from the Central Bank of Myanmar's main website. Since the coup, the website has gone offline and this data is no longer available. There was no formal API, however, by building URL strings it was possible to select data from specific dates.

Relevant summary statistics

Here's the model from my proposal:

```
xAndY <- function(days = 1, input = rates) {
  x <- input[1:(nrow(input)-days), 2:39]
  return(mutate(x, y = input[(days+1):nrow(input), "USD"]))
}

dataSplit <- function(data, ratio = 0.7) {
```

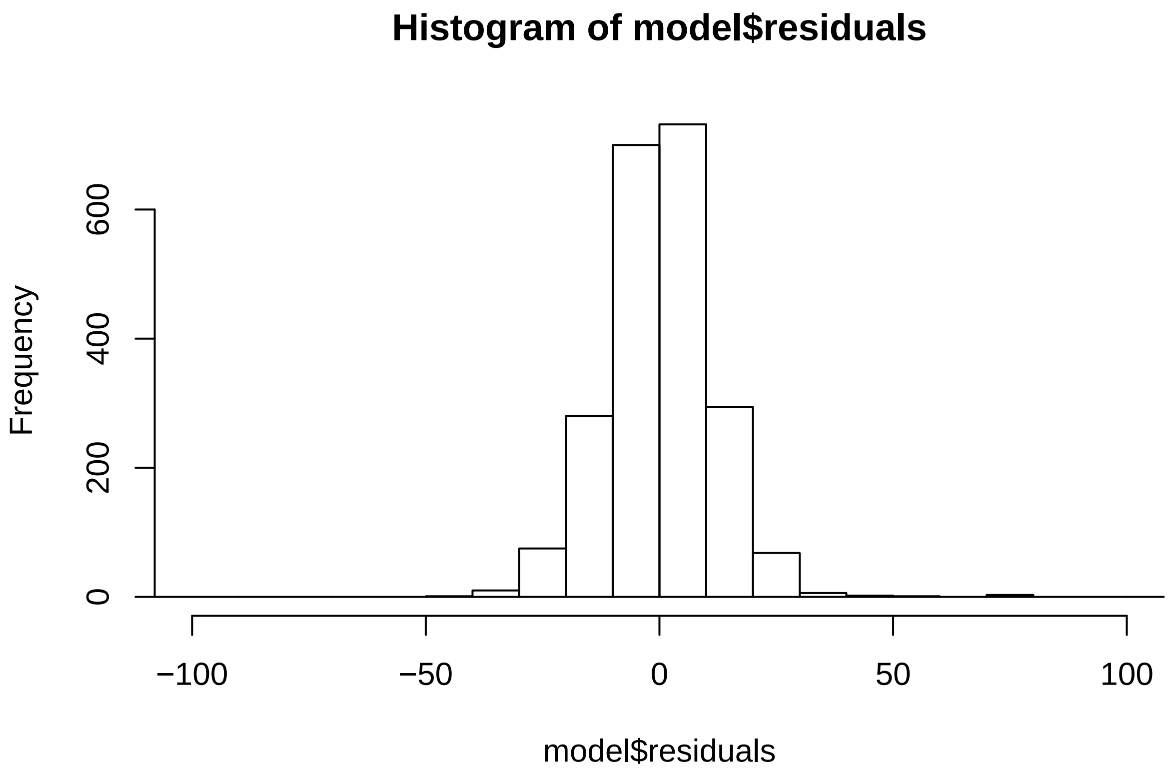
```
  n <- nrow(data) * ratio
  train <- data[1:n,]
  test <- data[(n+1):nrow(data),]
  return(list(train, test))
}

data <- xAndY(7) %>% dataSplit()
model <- lm(formula = y ~ ., data = data[[1]])
```

```
summary(model)
```

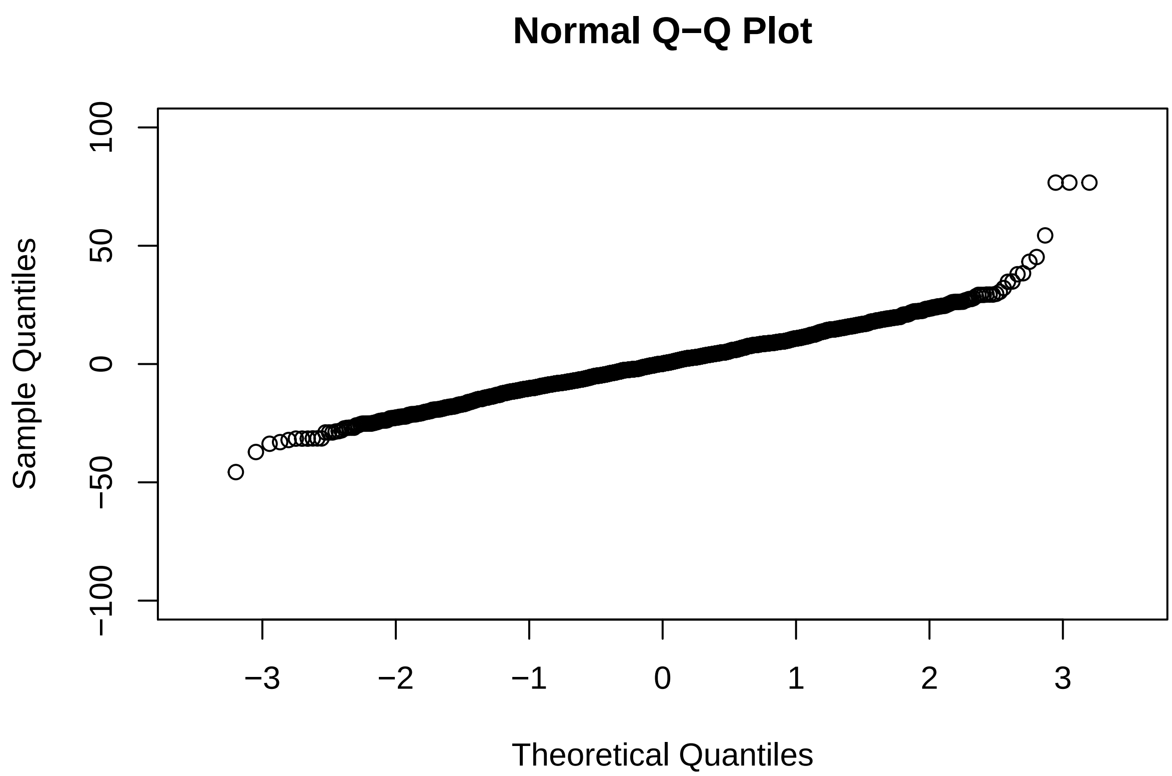
```
##
## Call:
## lm(formula = y ~ ., data = data[[1]])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -976.42   -7.18    0.14    7.91  248.42
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.366e+02  2.091e+01  35.224  < 2e-16 ***
## AUD          1.241e-01  5.123e-02   2.422 0.015502 *
## BDT          1.366e+01  7.119e+00   1.919 0.055085 .
## BND         -5.429e-04  4.619e-02  -0.012 0.990624
## BRL          6.034e-02  6.690e-02   0.902 0.367177
## CAD         -8.348e-03  4.840e-02  -0.172 0.863090
## CHF         -6.926e-02  3.515e-02  -1.970 0.048915 *
## CNY          4.432e-01  5.493e-01   0.807 0.419804
## CZK          4.409e+00  1.175e+00   3.752 0.000180 ***
## DKK         -4.872e-01  6.675e-01  -0.730 0.465510
## EGP          9.154e-02  8.527e-02   1.073 0.283203
## EUR          5.460e-02  8.913e-02   0.613 0.540201
## GBP          4.188e-02  2.010e-02   2.083 0.037336 *
## HKD          3.852e+00  4.247e+00   0.907 0.364462
## IDR          7.063e+00  4.382e+00   1.612 0.107179
## ILS          1.054e+00  1.637e-01   6.440 1.47e-10 ***
## INR          2.575e+00  6.848e+00   0.376 0.706908
## JPY         -7.383e-02  2.783e-02  -2.653 0.008031 **
## KES         -2.787e+01  5.716e+00  -4.875 1.17e-06 ***
## KHR         -1.058e+01  2.992e+00  -3.536 0.000415 ***
## KRW          2.344e-01  4.712e-01   0.497 0.618996
## KWD         -2.621e-01  6.599e-02  -3.972 7.35e-05 ***
## LAK          4.739e+01  1.101e+01   4.303 1.76e-05 ***
## LKR         -6.459e+00  8.948e+00  -0.722 0.470481
## MYR         -2.602e-01  1.157e-01  -2.249 0.024590 *
## NOK         -1.169e+00  2.928e-01  -3.992 6.77e-05 ***
## NPR          5.205e+00  1.087e+01   0.479 0.632043
## NZD         -4.202e-02  4.359e-02  -0.964 0.335107
## PHP         -2.768e+01  2.756e+00 -10.044  < 2e-16 ***
## PKR         -1.145e+01  4.743e+00  -2.415 0.015816 *
## RSD          1.524e+01  7.151e+00   2.130 0.033245 *
## RUB          7.793e-01  7.156e-01   1.089 0.276276
## SAR          6.129e-01  2.407e+00   0.255 0.799070
```

```
## SEK          -1.834e+00  3.414e-01  -5.370 8.71e-08 ***
## SGD          -4.002e-01  1.637e-01  -2.445 0.014580 *
## THB          -1.129e+01  2.105e+00  -5.362 9.14e-08 ***
## USD           1.384e+00  8.312e-01   1.665 0.096064 .
## VND          -1.969e+01  1.883e+01  -1.045 0.295961
## ZAR           5.369e-01  2.172e-01   2.472 0.013531 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.72 on 2135 degrees of freedom
## Multiple R-squared:  0.9824, Adjusted R-squared:  0.9821
## F-statistic:  3130 on 38 and 2135 DF,  p-value: < 2.2e-16
```

```r
hist(model$residuals, breaks = 100, xlim = c(-100, 100))
```




```r
qqnorm(model$residuals, ylim = c(-100, 100))
```



Interpreting the summary. . . .

So here we have an R-squared value which is very very close to the maximum value of 1. This means that nearly all of the variance in the dependent variable is explained by the model based on the independents. There's really not a lot left to gain. But, we are predicting timeseries data with historical information? The typical wisdom suggests this shouldn't work. Why does it?

This is not a free-floating currency! It isn't a monetary or inflationary target either, because the central bank doesn't have enough control over the money supply. There is a small committee who decides what the value of the kyat is going to be, and they don't make public their method of valuation. They introduce a small amount of randomness (and dispell many rumors about valuation by astrology) to discourage speculation in the market. However, a multiple regression model can cut right through, revealing the engineered value. I expect that the value is pegged to a weighted basket of currencies, and the results support this.

Improving the Model

It's cute to make these most basic functions by hand and rely on the summary() function, but there is much more available. To answer our questions, I will rebuild this model with the tidymodels package and assess the results. . . more thoroughly.

Q1: Does the model improve when we remove less-significant inputs?

```
data <- xAndY(7)

split <- initial_split(data, strata = y, p = 0.67)

trainer <- training(split)
tester <- testing(split)

mmkRecipe <- recipe(y ~ CZK + ILS + KES + KWD +
```

```r
                      LAK + NOK + PHP + SEK + THB, data = trainer) %>%
  step_center(all_predictors()) %>%
  step_scale(all_predictors()) %>%
  prep(training = trainer)

mmkTrain <- juice(mmkRecipe)
mmkTest <- bake(mmkRecipe, tester)

mmkModel <- linear_reg(mode = "regression") %>%
  set_engine("lm")

mmkFit <- mmkModel %>%
  fit(y ~ CZK + ILS + KES + KWD +
        LAK + NOK + PHP + SEK + THB, data = tester)

summary(mmkFit$fit)
```

```
##
## Call:
## stats::lm(formula = y ~ CZK + ILS + KES + KWD + LAK + NOK + PHP +
##     SEK + THB, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -973.49   -6.93    0.43    7.23   98.76
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  51.86471   32.14977   1.613   0.1070
## CZK           1.54007    1.05811   1.455   0.1458
## ILS           0.30372    0.13394   2.268   0.0236 *
## KES         -22.03902    5.07828  -4.340 1.57e-05 ***
## KWD           0.26062    0.03163   8.240 5.30e-16 ***
## LAK          27.34138    3.67620   7.437 2.18e-13 ***
## NOK          -0.13016    0.25218  -0.516   0.6059
## PHP           2.97459    2.50580   1.187   0.2355
## SEK          -2.04562    0.37968  -5.388 8.87e-08 ***
## THB           1.08317    1.48609   0.729   0.4662
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.55 on 1013 degrees of freedom
## Multiple R-squared:  0.9777, Adjusted R-squared:  0.9775
## F-statistic:  4936 on 9 and 1013 DF,  p-value: < 2.2e-16
```

The model performed a tiny bit worse, but is basically the same. Perhaps this is the best we can hope for with such a study. We note that the significance of CZK, NOK, PHP, and THB were reduced in this model. Maybe this means that those currencies are pegged to a similar basket, but are not a major part of the MMK valuation scheme.

Q2: Can we produce the same results or better using momentum and MACD from the target variable?

```
USD_MACD <- MACD(data$USD, nFast = 7, nSlow = 14)
USD_momentum <- ROC(data$USD, n = 7, type = "continuous")

data <- cbind(data, USD_MACD, USD_momentum)

data <- na.omit(data)
```

We train on the set one last time, removing CZK and THB and adding the new features:

```
split <- initial_split(data, strata = y, p = 0.67)

trainer <- training(split)
tester <- testing(split)

mmkRecipe <- recipe(y ~ ILS + KES + KWD + LAK + NOK + PHP + SEK +
                      macd + signal + USD_momentum, data = trainer) %>%
  step_center(all_predictors()) %>%
  step_scale(all_predictors()) %>%
  prep(training = trainer)

mmkTrain <- juice(mmkRecipe)
mmkTest <- bake(mmkRecipe, tester)

mmkModel <- linear_reg(mode = "regression") %>%
  set_engine("lm")

mmkFit <- mmkModel %>%
  fit(y ~ ILS + KES + KWD + LAK + NOK + PHP + SEK +
        macd + signal + USD_momentum, data = tester)

summary(mmkFit$fit)
```

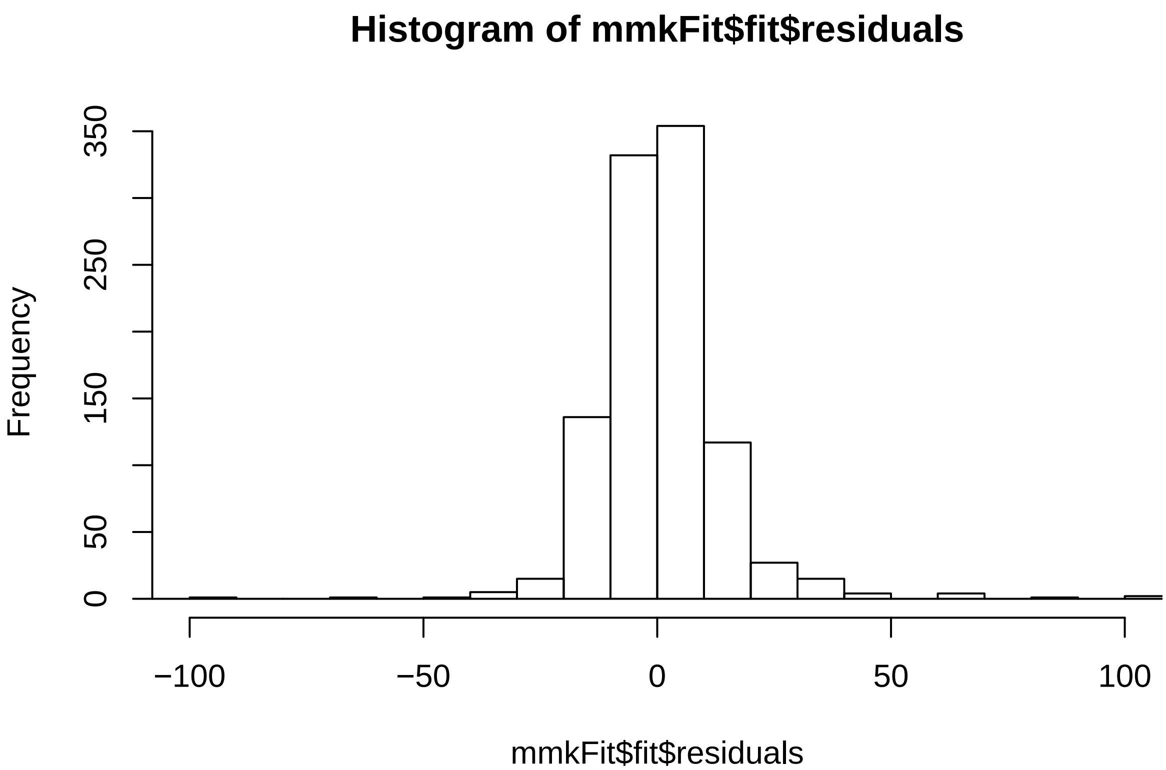
```
##
## Call:
## stats::lm(formula = y ~ ILS + KES + KWD + LAK + NOK + PHP + SEK +
##     macd + signal + USD_momentum, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -979.66   -6.25    0.53    7.07  100.27
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   41.87300   32.39481   1.293 0.196452
## ILS            0.39357    0.11912   3.304 0.000987 ***
## KES          -21.83457    4.82163  -4.528 6.65e-06 ***
## KWD            0.30448    0.02989  10.187  < 2e-16 ***
## LAK           20.97523    3.29134   6.373 2.82e-10 ***
## NOK            0.05753    0.24211   0.238 0.812238
## PHP            1.59663    2.09420   0.762 0.445996
## SEK           -1.90338    0.28244  -6.739 2.68e-11 ***
## macd          -2.38613    4.99233  -0.478 0.632784
```

```
## signal         4.74036    6.55059   0.724 0.469447
## USD_momentum  39.12456  159.33536   0.246 0.806082
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.78 on 1005 degrees of freedom
## Multiple R-squared:  0.9772, Adjusted R-squared:  0.977
## F-statistic:  4303 on 10 and 1005 DF,  p-value: < 2.2e-16
```

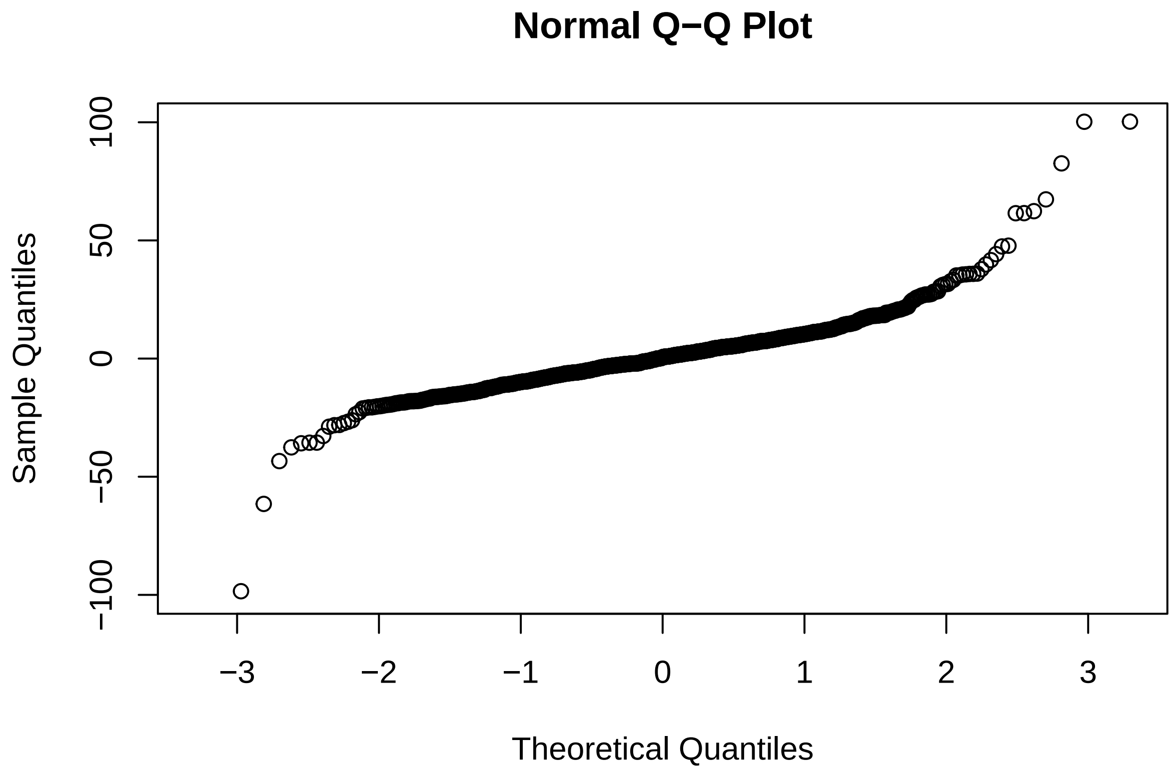
This model has again a slightly reduced performance, but it is still pretty accurate.

Conclusions

```r
hist(mmkFit$fit$residuals, breaks = 100, xlim = c(-100, 100))
```




```r
qqnorm(mmkFit$fit$residuals, ylim = c(-100, 100))
```



The lm() function, when applied correctly, is extremely powerful. This dataset lends itself to a linear model solution without much coercion, and it seems that feature engineering (as I have done, at least) weakens the model. Inclusion of all possible input data without any secondary stats or indicators is the best way to solve for the future value of MMK/USD.