Forecasting Stock Returns with Model Uncertainty and Parameter Instability

Nicholas J. DiBitetto

2024-12-11

This research project is based on the dataset 'Forecasting Stock Returns with Model Uncertainty and Parameter Instability (replication data)', initially analyzed by Hongwei Zhang, Qiang He, Ben Jacobsen, and Fuwei Jiang. I have conducted my own regressions using data sourced from notable databases and indexes, including The Chicago Fed National Activity Index, the Federal Reserve Economic Data (FRED), the Industrial Production Index Data from the Federal Reserve, the U.S. Bureau of Labor Statistics, and financial databases associated with the National Bureau of Economic Research (NBER). For detailed reference, the source of data used in this analysis is cited below.

# Load necessary libraries  
library(zoo)

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(knitr)  
library(readxl)

The zoo package was loaded to ensure independence of the index and to facilitate data manipulation, specifically for merging date functions of each independent variable and transforming the numeric columns. Additionally, the package offers functions for plotting zoo objects, which simplifies the visualization of time series data.

# Load datasets  
cay <- read.csv("C:/Users/DiBip/Downloads/cay.csv")  
cfnai <- read.csv("C:/Users/DiBip/Downloads/cfnai.csv")  
fred\_md\_factors <- read.csv("C:/Users/DiBip/Downloads/fred\_md\_factors.csv")  
gap <- read.csv("C:/Users/DiBip/Downloads/gap.csv")  
ipg <- read.csv("C:/Users/DiBip/Downloads/ipg.csv")  
mu <- read.csv("C:/Users/DiBip/Downloads/mu.csv")  
srp <- read.csv("C:/Users/DiBip/Downloads/srp.csv")  
unrate <- read.csv("C:/Users/DiBip/Downloads/unrate.csv")  
goyaldata\_nber <- read.csv("C:/Users/DiBip/Downloads/goyaldata\_nber.csv")  
  
# Convert yyyymm to numeric in `fred\_md\_factors` and `goyaldata\_nber`  
fred\_md\_factors$Date <- as.numeric(fred\_md\_factors$yyyymm)  
goyaldata\_nber$Date <- as.numeric(goyaldata\_nber$yyyymm)  
  
# Merge datasets using the common 'date' column  
data <- merge(merge(merge(merge(merge(merge(merge(cay, cfnai, by = "date"),  
 gap, by = "date"),  
 ipg, by = "date"),  
 mu, by = "date"),  
 srp, by = "date"),  
 unrate, by = "date"),  
 fred\_md\_factors, by.x = "date", by.y = "Date")  
data <- merge(data, goyaldata\_nber, by.x = "date", by.y = "Date")

Essentially, these codes convert the fred\_md\_factors and goyaldata\_nber datasets into numeric formats to be processed in a new column named date, along with the other datasets. The merging process uses nested merge functions to combine all datasets on the shared "date" column, allowing for the subsequent regression analyses.

# Define the numeric columns  
numeric\_cols <- c("Cay", "CFNAI", "Gap", "IPG", "MU", "SRP", "UNRATE", "factor1", "factor2", "factor3", "factor4", "factor5", "factor6", "factor7", "Index")  
  
# Convert all relevant columns to numeric, catching any NAs introduced by coercion  
data[numeric\_cols] <- lapply(data[numeric\_cols], function(x) as.numeric(as.character(x)))

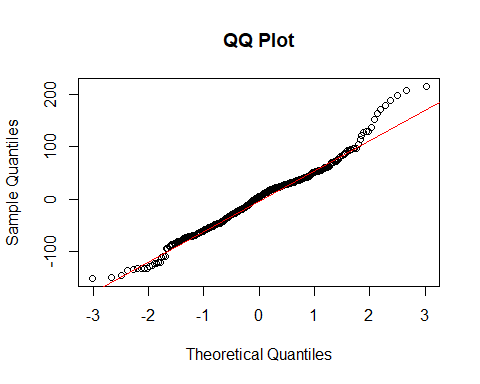
## Warning in FUN(X[[i]], ...): NAs introduced by coercion

# Remove rows with any NA, NaN, or Inf values across the entire dataset  
data <- data[complete.cases(data), ]

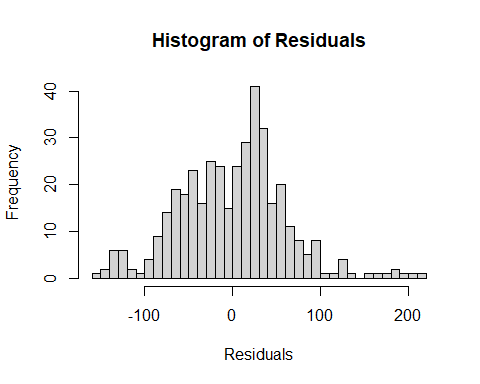
# Re-define the dependent variable (stock returns)  
y <- data$Index  
  
# Re-define the independent variables (factors)  
X <- data[, numeric\_cols[-length(numeric\_cols)]]  
  
# Fit the baseline regression model  
model <- lm(y ~ ., data = data.frame(y, X))  
summary(model)

##   
## Call:  
## lm(formula = y ~ ., data = data.frame(y, X))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -151.32 -43.38 4.91 35.17 215.75   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -397.8397 42.3263 -9.399 < 2e-16 \*\*\*  
## Cay -589.0160 355.7414 -1.656 0.098602 .   
## CFNAI 28.6565 19.1648 1.495 0.135677   
## Gap -1648.4710 385.9893 -4.271 2.47e-05 \*\*\*  
## IPG 15.4276 0.2802 55.057 < 2e-16 \*\*\*  
## MU 233.1081 58.0365 4.017 7.12e-05 \*\*\*  
## SRP -0.1367 0.2259 -0.605 0.545302   
## UNRATE -58.7809 5.1224 -11.475 < 2e-16 \*\*\*  
## factor1 79.5239 52.7055 1.509 0.132174   
## factor2 74.0608 16.3904 4.519 8.33e-06 \*\*\*  
## factor3 88.2429 15.8732 5.559 5.12e-08 \*\*\*  
## factor4 31.7302 14.9795 2.118 0.034805 \*   
## factor5 71.5730 21.4014 3.344 0.000907 \*\*\*  
## factor6 42.6750 20.5092 2.081 0.038126 \*   
## factor7 11.4053 20.1300 0.567 0.571333   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 61.63 on 379 degrees of freedom  
## Multiple R-squared: 0.9434, Adjusted R-squared: 0.9413   
## F-statistic: 451.5 on 14 and 379 DF, p-value: < 2.2e-16

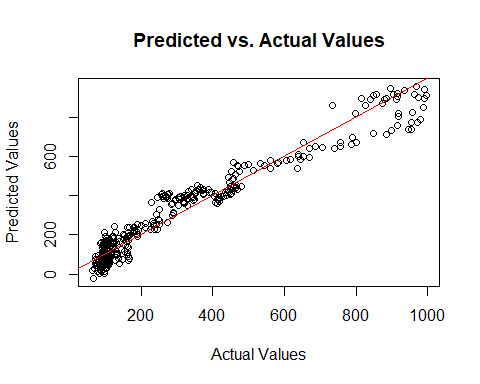
# QQ Plot (Quantile-Quantile Plot)  
qqnorm(model$residuals, main = "QQ Plot")  
qqline(model$residuals, col = "red")



# Histogram of Residuals  
hist(model$residuals, main = "Histogram of Residuals", xlab = "Residuals", breaks = 30)



# Predicted vs. Actual Values  
plot(data$Index, predict(model), main = "Predicted vs. Actual Values", xlab = "Actual Values", ylab = "Predicted Values")  
abline(0, 1, col = "red")



# Fit the rolling regression model  
window\_size <- 60 # 60-month rolling window  
rolling\_reg <- rollapply(data.frame(y, X), width = window\_size, by = 1, FUN = function(df) {  
 lm(y ~ ., data = as.data.frame(df))$coef  
}, by.column = FALSE)  
  
# Convert results to a data frame for plotting  
rolling\_reg <- as.data.frame(t(rolling\_reg))  
colnames(rolling\_reg) <- c("(Intercept)", numeric\_cols[-length(numeric\_cols)])  
  
# Plot the rolling regression coefficients  
matplot(rolling\_reg, type = "l", lty = 1, col = 1:ncol(rolling\_reg), xlab = "Time", ylab = "Coefficient Value", main = "Rolling Regression Coefficients")  
legend("topright", legend = colnames(rolling\_reg), col = 1:ncol(rolling\_reg), lty = 1, cex = 0.8)

A graph of a graph with different colored lines

Description automatically generated

Reference

Koijen, Ralph S. J., and Van Nieuwerburgh, Stijn (2020): "Forecasting Stock Returns With Model Uncertainty and Parameter Instability." ZBW - Leibniz Information Centre for Economics. Dataset. <https://journaldata.zbw.eu/dataset/forecasting-stock-returns-with-model-uncertainty-and-parameter-instability/resource/5b2a84a7-fa0b-4a93-b409-524ec31eae57?view_id=415ced0d-923f-4c34-82b2-d8fd4543b6c5>