New Regression

Nam Tran

Packages used

pacman::p\_load(tidyverse,   
 magrittr,  
 skimr,  
 psych,  
 broom)  
options(scipen = 999)

# Data Preparation

Import the full data to the working environment

full\_data <- read.csv('full\_data.csv', stringsAsFactors = T)

Convert PERMNO to factor and format the date column

full\_data %<>%  
 select(-X) %>%   
 mutate(PERMNO = as.factor(PERMNO),  
 date = ymd(date)  
 )

# Period 1

Extract data for the first period only

sample\_1 <- full\_data %>%   
 filter(year %in% c(2016, 2017, 2018)) %>%   
 select(everything())

# Customization Function

## 1. Model Execution

model\_execution <- function(data){  
 # Create an empty data frame to store the results  
 results <- data.frame(PERMNO = character(),  
 Reg1\_B1 = numeric(),  
 Reg1\_P1 = numeric(),  
 Reg1\_ARS = numeric(),  
 stringsAsFactors = T)  
   
 # Loop through all unique company names  
 for (company in unique(data[,'PERMNO'])) {  
 # Filter the data for the current company  
 company\_data <- data %>%  
 filter(PERMNO == company)  
   
 # Create a linear regression model with the constraint on the intercept  
 model <- lm(excess\_ret ~ 0 + market\_prem,   
 data = company\_data)  
   
 # Extract the coefficient values, p-values, and adjusted R-squared  
 coefficient <- coef(model)["market\_prem"]  
 p\_value <- summary(model)$coefficients["market\_prem",   
 "Pr(>|t|)"]  
 adjusted\_r\_squared <- summary(model)$adj.r.squared  
   
 # Store the extracted values in the results data frame  
 results <- rbind(results,   
 data.frame(PERMNO = company,  
 Reg1\_B1 = coefficient,  
 Reg1\_P1 = p\_value,  
 Reg1\_ARS = adjusted\_r\_squared,  
 stringsAsFactors = T))  
 rownames(results) <- NULL  
   
 # Round output to 4 decimal places  
 results %<>%   
 mutate\_if(is.numeric, round, digits = 4)  
 }  
 return(results)  
  
}

## 2. Statistical summary

statistic\_summary <- function(data){  
  
 train\_1 %>%   
 select(-PERMNO) %>%   
 describe() %>%   
 select(mean, sd, skew, kurtosis)  
}

## 3. Visualization

histogram\_viz <- function(input, variable, binwdith = 0.05, time){  
 input %>%  
 ggplot(aes\_string(x = variable)) +  
 geom\_histogram(binwidth = binwdith, fill = "#5DADE2", color = "#2C3E50", alpha = 0.8) +  
 labs(title = paste("Distribution of", variable, time),  
 x = variable,  
 y = "Frequency") +  
 theme\_minimal() +  
 theme(plot.title = element\_text(hjust = 0.5, size = 12, face = "bold"),  
 axis.title.x = element\_text(size = 10, face = "bold"),  
 axis.title.y = element\_text(size = 10, face = "bold"),  
 axis.text = element\_text(size = 8))  
}

## Model 1: Stock’s Excess Return ~ Unadjusted Market Return

### 1. Model’s Execution

train\_1 <- model\_execution(sample\_1)  
  
# Sample Result  
train\_1 %>% head()

PERMNO Reg1\_B1 Reg1\_P1 Reg1\_ARS  
1 10104 1.0007 0 0.3815  
2 10107 1.3631 0 0.6133  
3 10138 1.2792 0 0.5538  
4 10145 0.9291 0 0.5516  
5 10516 0.8986 0 0.2653  
6 10696 0.8886 0 0.4528

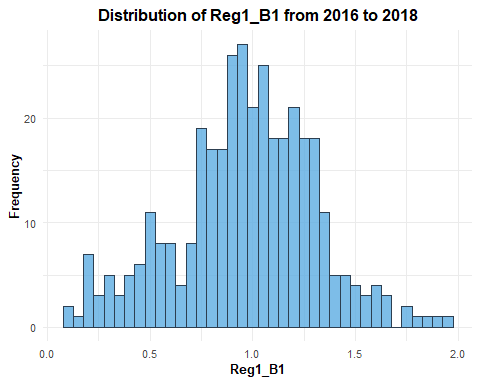
### 2. Statistical Summary

train\_1\_summary <- statistic\_summary(train\_1)

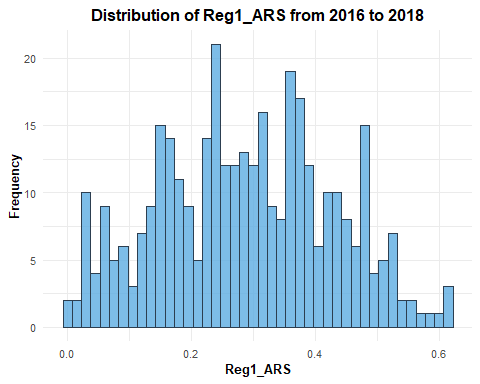
### 3. Visualization

histogram\_viz(train\_1,   
 variable = 'Reg1\_B1',  
 time = 'from 2016 to 2018')

Warning: `aes\_string()` was deprecated in ggplot2 3.0.0.  
ℹ Please use tidy evaluation ideoms with `aes()`



histogram\_viz(train\_1, variable = 'Reg1\_ARS',   
 binwdith = 0.015,  
 time = 'from 2016 to 2018')



# Period 2

Change the sub sample dataset

sample\_2 <- anti\_join(full\_data, sample\_1, by = 'date')

## Model 1: Stock’s Excess return ~ Market Premium

### 1. Model’s Execution

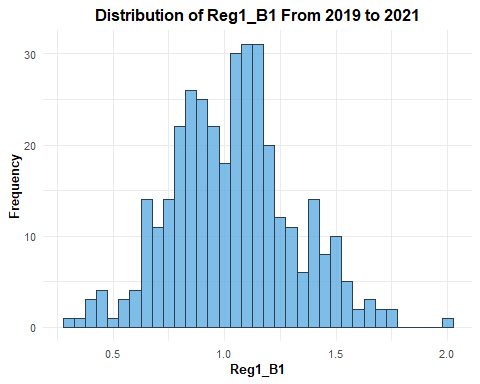
validation\_1 <- model\_execution(sample\_2)

### 2. Statistical Summary

valid\_1\_summary <- statistic\_summary(validation\_1)

### 3. Visualization

histogram\_viz(validation\_1,   
 variable = 'Reg1\_B1',   
 time = 'From 2019 to 2021')



histogram\_viz(validation\_1, 'Reg1\_ARS',   
 binwdith = 0.015,  
 'From 2019 to 2021')

