Portfolio Analysis Using Fama-Macbeth Regression

Introduction:

Fama-Macbeth procedure:

The Fama-MacBeth 1973 procedure is a two-step regression method primarily used in finance to empirically test asset pricing models like the Capital Asset Pricing Model (CAPM).

It involves running monthly cross-sectional regressions of the form:

Where p = 1,..., the portfolios and t are the time periods (months).

The Fama- macbeth in general involves the following steps.

Step 1: Cross-Sectional Regressions

Our objective is to estimate the relationship between returns and risk factors for each period separately. For each time period (month), we run a regression of the returns of portfolios on the risk factors predicted by the CAPM. Then we obtain a series of risk premiums (coefficients on the risk factors) for each period, which are estimates of how much return is associated with each unit of risk.

Step 2: Time-Series Analysis

Our objective is to determine the average risk premium over time and test its significance.We calculate the average of the estimated risk premiums from the first step over all periods. Then we conduct statistical tests to assess if these average premiums are significantly different from zero. This provides evidence on whether the risk factors (like the market premium in CAPM) are consistently priced in the market and whether the CAPM holds empirically.

The above are a generalization of the procedure which we will use more specifically to empirically test the context of the CAPM in our provided data.

By applying the Fama- Macbeth to CAPM, we can empirically test one of the model's core predictions: that the expected excess return on an asset is proportional to its market beta. The Fama-MacBeth method provides a way to estimate and test the risk premium associated with market risk. If the procedure finds that the average risk premium for market risk is significantly positive and other factors do not significantly explain returns, this supports CAPM's assertion that market risk is the primary determinant of asset returns.

The Fama-MacBeth procedure is particularly useful for this kind of testing because it accounts for potential issues like autocorrelation and heteroskedasticity in the error terms of financial data, providing more reliable estimates of the relationship between returns and risk as proposed by asset pricing models like CAPM.

Data analysis and discussion:

On the table below, we present the sample estimates obtained from the Fama – Macbeth procedure.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sample Estimates |  |  |  |  | rm\_r0 | rft |
| 1980-1986  1987-1991  1992-1995 | 3.24 | -6.98 | 4.29 | -0.06 | -2.4 | 0.34 |
| 1984-1990  1991-1995  1996-1999 | 0.72 | 0.197 | 1.1 | -0.14 | 1.37 | 0.406 |
| 1988-1994  1995-1999  2000-2003 | 0.552 | 1.261 | -1.55 | 0.024 | -0.498 | 0.246 |
| 1992-1998  1999-2003  2004-2007 | -0.83 | 1.45 | -1.41 | 0.19 | 1.43 | 0.28 |
| 1996-2002  2003-2007  2008-2011 | 0.82 | 0.42 | -0.043 | -0.141 | -0.28 | 0.035 |
| 2000-2006  2007-2011  2012-2015 | 1.66 | -2.06 | 0.93 | 0.004 | -0.56 | 0.005 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linearity (γ2) | t-statistic | p-value | Confidence interval | Reject/Accept  CAPM |
| 1980-1986  1987-1991  1992-1995 | 2.22 | 0.03 | 0.4, 8.2 | Reject |
| 1984-1990  1991-1995  1996-1999 | 0.72 | 0.48 | -1.98, 4.18 | Accept |
| 1988-1994  1995-1999  2000-2003 | -0.84 | 0.41 | -5.27, 2.17 | Accept |
| 1992-1998  1999-2003  2004-2007 | -1.62 | 0.11 | -3.15, 0.34 | Accept |
| 1996-2002  2003-2007  2008-2011 | -0.038 | 0.9697 | -2.32, 2.23 | Accept |
| 2000-2006  2007-2011  2012-2015 | 0.78 | 0.44 | -1.46, 3.32 | Accept |

Testing the Linearity checks if the relationship between variables (e.g., stock returns and risk factors) is linear, meaning a change in one variable results in a proportional change in another. In period 1, the t-stat is 2.2166 with a p-value of 0.03152, indicating that we can reject the null hypothesis of the expected value of γ\_2t being zero at the 5% significance level. In this case, a significant γ\_2t would imply a non-linear risk-return relationship, which is inconsistent with the predictions of CAPM. Similarly, we can argue for the results of each of the other time periods and why we reject or accept the CAPM.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No systematic effects of non-**β **risk (γ3)** | t-statistic | p-value | Confidence interval | Reject/Accept  CAPM |
| 1980-1986  1987-1991  1992-1995 | -0.483 | 0.631 | -0.324, 0.198 | Accept |
| 1984-1990  1991-1995  1996-1999 | -1.05 | 0.299 | -0.41, 0.13 | Accept |
| 1988-1994  1995-1999  2000-2003 | 0.15 | 0.88 | -0.29, 0.34 | Accept |
| 1992-1998  1999-2003  2004-2007 | 1.4 | 0.169 | -0.083, 0.46 | Accept |
| 1996-2002  2003-2007  2008-2011 | -1.09 | 0.28 | -0.403, 0.12 | Accept |
| 2000-2006  2007-2011  2012-2015 | 0.03 | 0.98 | -0.23, 0.24 | Accept |

Non-Systematic Effects of non-β risk evaluates the impact of factors unique to a particular asset or market that do not affect the market as a whole, such as company-specific news.For period 1, the t-statistic is -0.48318 with a p-value of 0.6312, indicating that we cannot reject the null hypothesis of E[γ\_3 ]=0 at conventional significance levels, including 1% and 5%. This suggests that there is insufficient evidence to conclude that there is a systematic non-beta risk effect during this testing period. It would be consistent with the CAPM prediction that non-market risk does not have a systematic effect on asset returns. With the same thought process, we can derive the results for the other periods.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Positive expected return-risk tradeoff (γ1)** | t-statistic | p-value | Confidence interval | Reject/Accept  CAPM |
| 1980-1986  1987-1991  1992-1995 | -1.17 | 0.25 | -12.43, 3.25 | Accept |
| 1984-1990  1991-1995  1996-1999 | -0.465 | 0.643 | -6.196, 3.852 | Accept |
| 1988-1994  1995-1999  2000-2003 | 0.615 | 0.54 | -3.95, 7.47 | Accept |
| 1992-1998  1999-2003  2004-2007 | 0.009 | 0.99 | -3.57, 3.61 | Accept |
| 1996-2002  2003-2007  2008-2011 | 0.224 | 0.823 | -5.54, 6.95 | Accept |
| 2000-2006  2007-2011  2012-2015 | -0.467 | 0.642 | -7.87, 4.89 | Accept |

Positive Expected Return Trade-off investigates the assumption that higher risk (volatility) should be compensated by higher expected returns, as per traditional finance theory. We filter for the first firm permno=10145 and extract the vector of values for the market return. The t-statistic is -1.1707 with a p-value of 0.2465, indicating that the null hypothesis, which states there is no significant difference in the means of γ\_1tand the excess returns over a risk-free asset, cannot be rejected at the conventional 5% or even at the 10% significance levels. This suggests that there is no evidence of a positive expected return-risk tradeoff, which is a deviation from the CAPM expectations. With the explanation method we can interpret the results for the other time periods.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sharpe-Lintner (S-L) Hypothesis (γ0)** | t-statistic | p-value | Confidence interval | Reject/Accept  CAPM |
| 1980-1986  1987-1991  1992-1995 | 2.21 | 0.03 | 0.26, 5.54 | Reject |
| 1984-1990  1991-1995  1996-1999 | 0.3 | 0.765 | -1.79, 2.42 | Accept |
| 1988-1994  1995-1999  2000-2003 | 0.235 | 0.815 | -2.31, 2.93 | Accept |
| 1992-1998  1999-2003  2004-2007 | -0.92 | 0.36 | -3.52, 1.3 | Accept |
| 1996-2002  2003-2007  2008-2011 | 0.578 | 0.5658 | -1.94, 3.51 | Accept |
| 2000-2006  2007-2011  2012-2015 | 1.39 | 0.17 | -0.73, 4.05 | Accept |

Sharpe-Linder Hypothesis explores the idea that there's a direct relationship between the slope of the security market line (a representation of the Capital Asset Pricing Model) and the level of market integration, suggesting that more integrated markets will exhibit a steeper slope.The t-statistic is 2.2144 and the p-value is 0.03168, which suggests that the null hypothesis of no difference between the mean intercept and mean risk-free return can be rejected at both the 5% and 10% significance levels. This indicates that the Sharpe-Lintner hypothesis does not likely hold, thus rejecting the CAPM. Similarly, we can explain the results of each of the other time periods and why we reject or accept the CAPM.

|  |  |  |  |
| --- | --- | --- | --- |
| **Capital market efficiency** | γ1  (p-value, reject/accept CAPM) | γ2 | γ3 |
| 1980-1986  1987-1991  1992-1995 | 0.84,  Accept CAPM | 0.8,  Accept CAPM | 0.65,  Accept CAPM |
| 1984-1990  1991-1995  1996-1999 | 0.24,  Accept | 0.14,  Accept | 0.197,  Accept |
| 1988-1994  1995-1999  2000-2003 | 0.172,  Accept | 0.258,  Accept | 0.333,  Accept |
| 1992-1998  1999-2003  2004-2007 | 0.739,  Accept | 0.958,  Accept | 0.715,  Accept |
| 1996-2002  2003-2007  2008-2011 | 0.056,  Accept | 0.086,  Accept | 0.75,  Accept |
| 2000-2006  2007-2011  2012-2015 | 0.99,  Accept | 0.989,  Accept | 0.098,  Accept |

Capital Market Efficiency assesses the market's ability to accurately reflect all available information in the prices of securities, implying that it's difficult to consistently achieve higher returns without taking on additional risk.For the γ\_1t coefficient, the calculated Ljung-Box statistic is 0.037153 with a p-value of 0.8472, which does not lead us to reject the null hypothesis of zero autocorrelation. Similarly, for γ\_2t, the statistic is 0.063252 with a p-value of 0.8014, and for γ\_3t, the statistic is 0.20364 with a p-value of 0.6518. In all these cases, the null hypothesis of zero autocorrelation stands, and we conclude that the coefficients γ\_1t, γ\_2t, and γ\_3tare likely independent over time.

During the first testing period of 1992-1995, the evidence does not compel us to reject the null hypothesis for any of the coefficients, suggesting that the CAPM assumptions hold for this timeframe. Keeping the same thought process in mind, we interpret the results for the other time periods similarly and that is how we conclude whether to accept or reject the CAPM. The capital market efficiency results show that we always accept in the 6 time periods.

APPENDIX:

Below find the related rscripts used to obtain the results presented in the assignment.

5069-GROUP-Period1.R

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.0 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(broom)  
  
df <- read\_csv("USstocks\_balanced.csv")

## Rows: 188778 Columns: 6  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## dbl (6): permno, year, month, ri, rm, rf  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

#Portfolio Formation  
  
#testing period 1 1992-1995  
pf\_groups <- df %>% filter(between(year, 1980, 1986)) %>%   
 group\_by(permno) %>%   
 nest() %>%   
 mutate(betaest = map(data, ~ coef(lm(ri ~ rm, data = .x))[2])) %>%   
 unnest(betaest) %>%   
 ungroup() %>%   
 mutate(pfid = ntile(betaest, 20)) %>%   
 select(permno, pfid)  
  
#creat match between permno and portfolio groups  
#esimation period 1987-1991,testing period 1992-1995  
df\_withpfid <- df %>%   
 filter(between(year, 1987, 1995)) %>%   
 left\_join(pf\_groups)

## Joining with `by = join\_by(permno)`

#Portfolio Estimation  
  
#estimate market betas and standard deviation of residuals  
#Regressions start with the estimation period 1987-1991.   
#The beta and residual standard deviations are stored for the first year of testing period   
#which is the year 1992  
beta\_sigma\_92 <- df\_withpfid %>%   
 filter(year <= 1991) %>%   
 group\_by(permno) %>%   
 nest() %>%   
 mutate(betas = map(data, ~ coef(lm(ri ~ rm, data = .x))[2]),  
 sigmas = map(data, ~ sigma(lm(ri ~ rm, data = .x))),  
 yeartesting = 1992) %>%   
 unnest(c(betas, sigmas)) %>%   
 select(permno, betas, sigmas, yeartesting)  
  
#To get the estimates for the year 1992  
#we can repeat the same procedure but including the additional year of 1992  
testingperiod\_92to95 <- data.frame()  
  
for(i in 0:3){  
   
 beta\_sigma <- df\_withpfid %>%   
 filter(year <= 1991 + i) %>%   
 group\_by(permno) %>%   
 nest() %>%   
 mutate(betas = map(data, ~ coef(lm(ri ~ rm, data = .x))[2]),  
 sigmas = map(data, ~ sigma(lm(ri ~ rm, data = .x))),  
 yeartesting = 1992 + i) %>%   
 unnest(c(betas, sigmas)) %>%   
 select(permno, betas, sigmas, yeartesting)  
   
 # estimates is for 1992  
 testingperiod\_92to95 <- bind\_rows(testingperiod\_92to95, beta\_sigma)  
   
}  
  
  
testingperiod\_92to95 <- rename(testingperiod\_92to95, year = yeartesting)  
  
# join with df\_withpfid for all the testing years  
df\_testingperiod1 <- df\_withpfid %>%   
 filter(between(year, 1992, 1995)) %>%   
 left\_join(testingperiod\_92to95)

## Joining with `by = join\_by(permno, year)`

testing\_period1 <- df\_testingperiod1 %>%   
 group\_by(pfid, year, month) %>%   
 summarise(rp = mean(ri, na.rm = T),  
 betap = mean(betas, na.rm = T),  
 betap\_squared = mean(betas^2, na.rm = T),  
 sigmap = mean(sigmas, na.rm = T)) %>%   
 ungroup()

## `summarise()` has grouped output by 'pfid', 'year'. You can override using the  
## `.groups` argument.

#Portfolio Testing  
  
testing\_period1\_gammas <- testing\_period1 %>%   
 group\_by(year, month) %>%   
 nest(data = !c(year, month)) %>%   
 mutate(estim = map(data, ~lm(lead(rp) ~ betap + betap\_squared + sigmap, data = .x)),   
 estim = map(estim, tidy)) %>%   
 unnest(estim) %>%   
 select(year, month, term, estimate) %>%   
 ungroup()  
  
#Linearity  
gamma\_2t <- testing\_period1\_gammas %>%   
 filter(term == "betap\_squared") %>%   
 pull(estimate)  
  
# T test   
t.test(gamma\_2t, mu = 0)

##   
## One Sample t-test  
##   
## data: gamma\_2t  
## t = 2.2166, df = 47, p-value = 0.03152  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 0.3967325 8.1868271  
## sample estimates:  
## mean of x   
## 4.29178

#No systematic effects of non-β risk  
gamma\_3t <- testing\_period1\_gammas %>%   
 filter(term == "sigmap") %>%   
 pull(estimate)  
  
# T test   
t.test(gamma\_3t)

##   
## One Sample t-test  
##   
## data: gamma\_3t  
## t = -0.48318, df = 47, p-value = 0.6312  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -0.3240014 0.1985054  
## sample estimates:  
## mean of x   
## -0.062748

#Positive expected return-risk tradeoff  
gamma\_1t <- testing\_period1\_gammas %>%   
 filter(term == "betap") %>%   
 pull(estimate)  
  
# Rm  
rm\_data <- df\_testingperiod1 %>%   
 filter(permno == 10145) %>%   
 pull(rm)  
  
# Rm - R0  
gamma\_0t <- testing\_period1\_gammas %>%   
 filter(term == "(Intercept)") %>%   
 pull(estimate)  
  
rm\_r0 <- rm\_data - gamma\_0t  
  
t.test(gamma\_1t, rm\_r0) # t test

##   
## Welch Two Sample t-test  
##   
## data: gamma\_1t and rm\_r0  
## t = -1.1707, df = 58.518, p-value = 0.2465  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -12.425875 3.254121  
## sample estimates:  
## mean of x mean of y   
## -6.985560 -2.399683

length(gamma\_1t)

## [1] 48

length(rm\_r0)

## [1] 48

#Sharpe-Lintner (S-L) Hypothesis  
rft <- df\_testingperiod1 %>%   
 filter(permno == 10145) %>%   
 pull(rf)  
  
t.test(gamma\_0t, rft)

##   
## Welch Two Sample t-test  
##   
## data: gamma\_0t and rft  
## t = 2.2144, df = 47.009, p-value = 0.03168  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 0.2659096 5.5442506  
## sample estimates:  
## mean of x mean of y   
## 3.2418683 0.3367882

Box.test(x = gamma\_1t, lag = 1, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: gamma\_1t  
## X-squared = 0.037153, df = 1, p-value = 0.8472

Box.test(x = gamma\_2t, lag = 1, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: gamma\_2t  
## X-squared = 0.063252, df = 1, p-value = 0.8014

Box.test(x = gamma\_3t, lag = 1, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: gamma\_3t  
## X-squared = 0.20364, df = 1, p-value = 0.6518

5069-GROUP-Period2.R

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.0 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(broom)  
  
df <- read\_csv("USstocks\_balanced.csv")

## Rows: 188778 Columns: 6  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## dbl (6): permno, year, month, ri, rm, rf  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

#Portfolio Formation  
  
#testing period 2 1996-1999  
pf\_groups <- df %>% filter(between(year, 1984, 1990)) %>%   
 group\_by(permno) %>%   
 nest() %>%   
 mutate(betaest = map(data, ~ coef(lm(ri ~ rm, data = .x))[2])) %>%   
 unnest(betaest) %>%   
 ungroup() %>%   
 mutate(pfid = ntile(betaest, 20)) %>%   
 select(permno, pfid)  
  
#creat match between permno and portfolio groups  
#esimation period 1991-1995,testing period 1996-1999  
df\_withpfid <- df %>%   
 filter(between(year, 1991, 1999)) %>%   
 left\_join(pf\_groups)

## Joining with `by = join\_by(permno)`

#Portfolio Estimation  
  
#estimate market betas and standard deviation of residuals  
#Regressions start with the estimation period 1991-1995.   
#The beta and residual standard deviations are stored for the first year of testing period   
#which is the year 1996  
beta\_sigma\_96 <- df\_withpfid %>%   
 filter(year <= 1995) %>%   
 group\_by(permno) %>%   
 nest() %>%   
 mutate(betas = map(data, ~ coef(lm(ri ~ rm, data = .x))[2]),  
 sigmas = map(data, ~ sigma(lm(ri ~ rm, data = .x))),  
 yeartesting = 1996) %>%   
 unnest(c(betas, sigmas)) %>%   
 select(permno, betas, sigmas, yeartesting)  
  
#To get the estimates for the year 1996  
#we can repeat the same procedure but including the additional year of 1996  
testingperiod\_96to99 <- data.frame()  
  
for(i in 0:3){  
   
 beta\_sigma <- df\_withpfid %>%   
 filter(year <= 1995 + i) %>%   
 group\_by(permno) %>%   
 nest() %>%   
 mutate(betas = map(data, ~ coef(lm(ri ~ rm, data = .x))[2]),  
 sigmas = map(data, ~ sigma(lm(ri ~ rm, data = .x))),  
 yeartesting = 1996 + i) %>%   
 unnest(c(betas, sigmas)) %>%   
 select(permno, betas, sigmas, yeartesting)  
   
 # estimates is for 1996  
 testingperiod\_96to99 <- bind\_rows(testingperiod\_96to99, beta\_sigma)  
   
}  
  
testingperiod\_96to99 <- rename(testingperiod\_96to99, year = yeartesting)  
  
# join with df\_withpfid for all the testing years  
df\_testingperiod2 <- df\_withpfid %>%   
 filter(between(year, 1996, 1999)) %>%   
 left\_join(testingperiod\_96to99)

## Joining with `by = join\_by(permno, year)`

testing\_period2 <- df\_testingperiod2 %>%   
 group\_by(pfid, year, month) %>%   
 summarise(rp = mean(ri, na.rm = T),  
 betap = mean(betas, na.rm = T),  
 betap\_squared = mean(betas^2, na.rm = T),  
 sigmap = mean(sigmas, na.rm = T)) %>%   
 ungroup()

## `summarise()` has grouped output by 'pfid', 'year'. You can override using the  
## `.groups` argument.

#Portfolio Testing  
  
testing\_period2\_gammas <- testing\_period2 %>%   
 group\_by(year, month) %>%   
 nest(data = !c(year, month)) %>%   
 mutate(estim = map(data, ~lm(lead(rp) ~ betap + betap\_squared + sigmap, data = .x)),   
 estim = map(estim, tidy)) %>%   
 unnest(estim) %>%   
 select(year, month, term, estimate) %>%   
 ungroup()  
  
#Linearity  
gamma\_2t <- testing\_period2\_gammas %>%   
 filter(term == "betap\_squared") %>%   
 pull(estimate)  
  
# T test   
t.test(gamma\_2t, mu = 0)

##   
## One Sample t-test  
##   
## data: gamma\_2t  
## t = 0.71998, df = 47, p-value = 0.4751  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -1.975434 4.177499  
## sample estimates:  
## mean of x   
## 1.101033

#No systematic effects of non-β risk  
gamma\_3t <- testing\_period2\_gammas %>%   
 filter(term == "sigmap") %>%   
 pull(estimate)  
  
# T test   
t.test(gamma\_3t)

##   
## One Sample t-test  
##   
## data: gamma\_3t  
## t = -1.0505, df = 47, p-value = 0.2989  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -0.4074739 0.1279079  
## sample estimates:  
## mean of x   
## -0.139783

#Positive expected return-risk tradeoff  
gamma\_1t <- testing\_period2\_gammas %>%   
 filter(term == "betap") %>%   
 pull(estimate)  
  
# Rm  
rm\_data <- df\_testingperiod2 %>%   
 filter(permno == 10145) %>%   
 pull(rm)  
  
# Rm - R0  
gamma\_0t <- testing\_period2\_gammas %>%   
 filter(term == "(Intercept)") %>%   
 pull(estimate)  
  
rm\_r0 <- rm\_data - gamma\_0t  
  
t.test(gamma\_1t, rm\_r0) # t test

##   
## Welch Two Sample t-test  
##   
## data: gamma\_1t and rm\_r0  
## t = -0.46547, df = 68.135, p-value = 0.6431  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -6.196382 3.852301  
## sample estimates:  
## mean of x mean of y   
## 0.1971796 1.3692202

length(gamma\_1t)

## [1] 48

length(rm\_r0)

## [1] 48

#Sharpe-Lintner (S-L) Hypothesis  
rft <- df\_testingperiod2 %>%   
 filter(permno == 10145) %>%   
 pull(rf)  
  
t.test(gamma\_0t, rft)

##   
## Welch Two Sample t-test  
##   
## data: gamma\_0t and rft  
## t = 0.30084, df = 47.001, p-value = 0.7649  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -1.790495 2.420156  
## sample estimates:  
## mean of x mean of y   
## 0.7206985 0.4058681

Box.test(x = gamma\_1t, lag = 1, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: gamma\_1t  
## X-squared = 1.3965, df = 1, p-value = 0.2373

Box.test(x = gamma\_2t, lag = 1, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: gamma\_2t  
## X-squared = 2.1816, df = 1, p-value = 0.1397

Box.test(x = gamma\_3t, lag = 1, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: gamma\_3t  
## X-squared = 1.6623, df = 1, p-value = 0.1973

Period-3.R

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.0 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(broom)  
  
df <- read.csv("USstocks\_balanced.csv")  
  
# Portfolio formation  
  
# Testing period 3  
pf\_groups <- df %>% filter(between(year, 1988, 1994)) %>%   
 group\_by(permno) %>%   
 nest() %>%   
 mutate(betaest = map(data, ~ coef(lm(ri ~ rm, data = .x))[2])) %>%   
 unnest(betaest) %>%   
 ungroup() %>%   
 mutate(pfid = ntile(betaest, 20)) %>%   
 select(permno, pfid)  
  
#create match between permno and portfolio groups  
#estimation period 1995-1999,testing period 2000-2003  
df\_withpfid <- df %>%   
 filter(between(year, 1995, 2003)) %>%   
 left\_join(pf\_groups)

## Joining with `by = join\_by(permno)`

#Portfolio Estimation  
  
#estimate market betas and standard deviation of residuals  
#Regressions start with the estimation period 1995-1999.   
#The beta and residual standard deviations are stored for the first year of  
#testing period which is the year 2000  
beta\_sigma\_00 <- df\_withpfid %>%   
 filter(year <= 1999) %>%   
 group\_by(permno) %>%   
 nest() %>%   
 mutate(betas = map(data, ~ coef(lm(ri ~ rm, data = .x))[2]),  
 sigmas = map(data, ~ sigma(lm(ri ~ rm, data = .x))),  
 yeartesting = 2000) %>%   
 unnest(c(betas, sigmas)) %>%   
 select(permno, betas, sigmas, yeartesting)  
  
#To get the estimates for the year 2000  
#we repeat the same procedure but including the additional year of 2000  
testingperiod\_00to03 <- data.frame()  
  
for(i in 0:3){  
   
 beta\_sigma <- df\_withpfid %>%   
 filter(year <= 1999 + i) %>%   
 group\_by(permno) %>%   
 nest() %>%   
 mutate(betas = map(data, ~ coef(lm(ri ~ rm, data = .x))[2]),  
 sigmas = map(data, ~ sigma(lm(ri ~ rm, data = .x))),  
 yeartesting = 2000 + i) %>%   
 unnest(c(betas, sigmas)) %>%   
 select(permno, betas, sigmas, yeartesting)  
   
 # estimates is for 2000  
 testingperiod\_00to03 <- bind\_rows(testingperiod\_00to03, beta\_sigma)  
   
}  
  
testingperiod\_00to03 <- rename(testingperiod\_00to03, year = yeartesting)  
  
# join with df\_withpfid for all the testing years  
df\_testingperiod3 <- df\_withpfid %>%   
 filter(between(year, 2000, 2003)) %>%   
 left\_join(testingperiod\_00to03)

## Joining with `by = join\_by(permno, year)`

testing\_period3 <- df\_testingperiod3 %>%   
 group\_by(pfid, year, month) %>%   
 summarise(rp = mean(ri, na.rm = T),  
 betap = mean(betas, na.rm = T),  
 betap\_squared = mean(betas^2, na.rm = T),  
 sigmap = mean(sigmas, na.rm = T)) %>%   
 ungroup()

## `summarise()` has grouped output by 'pfid', 'year'. You can override using the  
## `.groups` argument.

#Portfolio Testing !!!!!!  
  
testing\_period3\_gammas <- testing\_period3 %>%   
 group\_by(year, month) %>%   
 nest(data = !c(year, month)) %>%   
 mutate(estim = map(data, ~lm(lead(rp) ~ betap + betap\_squared + sigmap, data = .x)),   
 estim = map(estim, tidy)) %>%   
 unnest(estim) %>%   
 select(year, month, term, estimate) %>%   
 ungroup()  
  
#Linearity  
gamma\_2t <- testing\_period3\_gammas %>%   
 filter(term == "betap\_squared") %>%   
 pull(estimate)  
  
# T test   
t.test(gamma\_2t, mu = 0)

##   
## One Sample t-test  
##   
## data: gamma\_2t  
## t = -0.83779, df = 47, p-value = 0.4064  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -5.272721 2.172246  
## sample estimates:  
## mean of x   
## -1.550238

#No systematic effects of non-β risk  
gamma\_3t <- testing\_period3\_gammas %>%   
 filter(term == "sigmap") %>%   
 pull(estimate)  
  
# T test   
t.test(gamma\_3t)

##   
## One Sample t-test  
##   
## data: gamma\_3t  
## t = 0.15357, df = 47, p-value = 0.8786  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -0.2884140 0.3360862  
## sample estimates:  
## mean of x   
## 0.02383607

#Positive expected return-risk tradeoff  
gamma\_1t <- testing\_period3\_gammas %>%   
 filter(term == "betap") %>%   
 pull(estimate)  
  
# Rm  
rm\_data <- df\_testingperiod3 %>%   
 filter(permno == 10145) %>%   
 pull(rm)  
  
# Rm - R0  
gamma\_0t <- testing\_period3\_gammas %>%   
 filter(term == "(Intercept)") %>%   
 pull(estimate)  
  
rm\_r0 <- rm\_data - gamma\_0t  
  
t.test(gamma\_1t, rm\_r0) # t test

##   
## Welch Two Sample t-test  
##   
## data: gamma\_1t and rm\_r0  
## t = 0.61503, df = 66.997, p-value = 0.5406  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -3.950295 7.468887  
## sample estimates:  
## mean of x mean of y   
## 1.2612655 -0.4980304

length(gamma\_1t)

## [1] 48

length(rm\_r0)

## [1] 48

#Sharpe-Lintner (S-L) Hypothesis  
rft <- df\_testingperiod3 %>%   
 filter(permno == 10145) %>%   
 pull(rf)  
  
t.test(gamma\_0t, rft)

##   
## Welch Two Sample t-test  
##   
## data: gamma\_0t and rft  
## t = 0.23506, df = 47.031, p-value = 0.8152  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -2.314171 2.926518  
## sample estimates:  
## mean of x mean of y   
## 0.5523888 0.2462153

Box.test(x = gamma\_1t, lag = 1, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: gamma\_1t  
## X-squared = 1.8654, df = 1, p-value = 0.172

Box.test(x = gamma\_2t, lag = 1, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: gamma\_2t  
## X-squared = 1.2784, df = 1, p-value = 0.2582

Box.test(x = gamma\_3t, lag = 1, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: gamma\_3t  
## X-squared = 0.93495, df = 1, p-value = 0.3336

Period-4.R

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.0 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(broom)  
  
df <- read.csv("USstocks\_balanced.csv")  
  
# Portfolio formation  
  
# Testing period 4  
pf\_groups <- df %>% filter(between(year, 1992, 1998)) %>%   
 group\_by(permno) %>%   
 nest() %>%   
 mutate(betaest = map(data, ~ coef(lm(ri ~ rm, data = .x))[2])) %>%   
 unnest(betaest) %>%   
 ungroup() %>%   
 mutate(pfid = ntile(betaest, 20)) %>%   
 select(permno, pfid)  
  
#create match between permno and portfolio groups  
#estimation period 1999-2003,testing period 2004-2007  
df\_withpfid <- df %>%   
 filter(between(year, 1999, 2007)) %>%   
 left\_join(pf\_groups)

## Joining with `by = join\_by(permno)`

#Portfolio Estimation  
  
#estimate market betas and standard deviation of residuals  
#Regressions start with the estimation period 1999-2003.   
#The beta and residual standard deviations are stored for the first year of  
#testing period which is the year 2004  
beta\_sigma\_04 <- df\_withpfid %>%   
 filter(year <= 2003) %>%   
 group\_by(permno) %>%   
 nest() %>%   
 mutate(betas = map(data, ~ coef(lm(ri ~ rm, data = .x))[2]),  
 sigmas = map(data, ~ sigma(lm(ri ~ rm, data = .x))),  
 yeartesting = 2004) %>%   
 unnest(c(betas, sigmas)) %>%   
 select(permno, betas, sigmas, yeartesting)  
  
#To get the estimates for the year 2004  
#we repeat the same procedure but including the additional year of 2004  
testingperiod\_04to07 <- data.frame()  
  
for(i in 0:3){  
   
 beta\_sigma <- df\_withpfid %>%   
 filter(year <= 2003 + i) %>%   
 group\_by(permno) %>%   
 nest() %>%   
 mutate(betas = map(data, ~ coef(lm(ri ~ rm, data = .x))[2]),  
 sigmas = map(data, ~ sigma(lm(ri ~ rm, data = .x))),  
 yeartesting = 2004 + i) %>%   
 unnest(c(betas, sigmas)) %>%   
 select(permno, betas, sigmas, yeartesting)  
   
 # estimates are for 2004  
 testingperiod\_04to07 <- bind\_rows(testingperiod\_04to07, beta\_sigma)  
   
}  
  
testingperiod\_04to07 <- rename(testingperiod\_04to07, year = yeartesting)  
  
# join with df\_withpfid for all the testing years  
df\_testingperiod4 <- df\_withpfid %>%   
 filter(between(year, 2004, 2007)) %>%   
 left\_join(testingperiod\_04to07)

## Joining with `by = join\_by(permno, year)`

testing\_period4 <- df\_testingperiod4 %>%   
 group\_by(pfid, year, month) %>%   
 summarise(rp = mean(ri, na.rm = T),  
 betap = mean(betas, na.rm = T),  
 betap\_squared = mean(betas^2, na.rm = T),  
 sigmap = mean(sigmas, na.rm = T)) %>%   
 ungroup()

## `summarise()` has grouped output by 'pfid', 'year'. You can override using the  
## `.groups` argument.

#Portfolio Testing  
  
testing\_period4\_gammas <- testing\_period4 %>%   
 group\_by(year, month) %>%   
 nest(data = !c(year, month)) %>%   
 mutate(estim = map(data, ~lm(lead(rp) ~ betap + betap\_squared + sigmap, data = .x)),   
 estim = map(estim, tidy)) %>%   
 unnest(estim) %>%   
 select(year, month, term, estimate) %>%   
 ungroup()  
  
#Linearity  
gamma\_2t <- testing\_period4\_gammas %>%   
 filter(term == "betap\_squared") %>%   
 pull(estimate)  
  
# T test   
t.test(gamma\_2t, mu = 0)

##   
## One Sample t-test  
##   
## data: gamma\_2t  
## t = -1.6225, df = 47, p-value = 0.1114  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -3.1542300 0.3378479  
## sample estimates:  
## mean of x   
## -1.408191

#No systematic effects of non-β risk  
gamma\_3t <- testing\_period4\_gammas %>%   
 filter(term == "sigmap") %>%   
 pull(estimate)  
  
# T test   
t.test(gamma\_3t)

##   
## One Sample t-test  
##   
## data: gamma\_3t  
## t = 1.3969, df = 47, p-value = 0.169  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -0.08290222 0.45962106  
## sample estimates:  
## mean of x   
## 0.1883594

#Positive expected return-risk tradeoff  
gamma\_1t <- testing\_period4\_gammas %>%   
 filter(term == "betap") %>%   
 pull(estimate)  
  
# Rm  
rm\_data <- df\_testingperiod4 %>%   
 filter(permno == 10145) %>%   
 pull(rm)  
  
# Rm - R0  
gamma\_0t <- testing\_period4\_gammas %>%   
 filter(term == "(Intercept)") %>%   
 pull(estimate)  
  
rm\_r0 <- rm\_data - gamma\_0t  
  
t.test(gamma\_1t, rm\_r0) # t test

##   
## Welch Two Sample t-test  
##   
## data: gamma\_1t and rm\_r0  
## t = 0.009038, df = 89.75, p-value = 0.9928  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -3.574592 3.607264  
## sample estimates:  
## mean of x mean of y   
## 1.446957 1.430622

length(gamma\_1t)

## [1] 48

length(rm\_r0)

## [1] 48

#Sharpe-Lintner (S-L) Hypothesis  
rft <- df\_testingperiod4 %>%   
 filter(permno == 10145) %>%   
 pull(rf)  
  
t.test(gamma\_0t, rft)

##   
## Welch Two Sample t-test  
##   
## data: gamma\_0t and rft  
## t = -0.92453, df = 47.019, p-value = 0.3599  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -3.523704 1.304696  
## sample estimates:  
## mean of x mean of y   
## -0.8262051 0.2832986

Box.test(x = gamma\_1t, lag = 1, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: gamma\_1t  
## X-squared = 0.11083, df = 1, p-value = 0.7392

Box.test(x = gamma\_2t, lag = 1, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: gamma\_2t  
## X-squared = 0.0026911, df = 1, p-value = 0.9586

Box.test(x = gamma\_3t, lag = 1, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: gamma\_3t  
## X-squared = 0.13357, df = 1, p-value = 0.7148

Period-5.R

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.0 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(broom)  
  
df <- read.csv("USstocks\_balanced.csv")  
  
# Portfolio formation  
  
# Testing period 5  
pf\_groups <- df %>% filter(between(year, 1996, 2002)) %>%   
 group\_by(permno) %>%   
 nest() %>%   
 mutate(betaest = map(data, ~ coef(lm(ri ~ rm, data = .x))[2])) %>%   
 unnest(betaest) %>%   
 ungroup() %>%   
 mutate(pfid = ntile(betaest, 20)) %>%   
 select(permno, pfid)  
  
#create match between permno and portfolio groups  
#estimation period 2003-2007,testing period 2008-2011  
df\_withpfid <- df %>%   
 filter(between(year, 2003, 2011)) %>%   
 left\_join(pf\_groups)

## Joining with `by = join\_by(permno)`

#Portfolio Estimation  
  
#estimate market betas and standard deviation of residuals  
#Regressions start with the estimation period 2003-2007.   
#The beta and residual standard deviations are stored for the first year of  
#testing period which is the year 2008  
beta\_sigma\_08 <- df\_withpfid %>%   
 filter(year <= 2007) %>%   
 group\_by(permno) %>%   
 nest() %>%   
 mutate(betas = map(data, ~ coef(lm(ri ~ rm, data = .x))[2]),  
 sigmas = map(data, ~ sigma(lm(ri ~ rm, data = .x))),  
 yeartesting = 2008) %>%   
 unnest(c(betas, sigmas)) %>%   
 select(permno, betas, sigmas, yeartesting)  
  
#To get the estimates for the year 2008  
#we repeat the same procedure but including the additional year of 2008  
testingperiod\_08to11 <- data.frame()  
  
for(i in 0:3){  
   
 beta\_sigma <- df\_withpfid %>%   
 filter(year <= 2007 + i) %>%   
 group\_by(permno) %>%   
 nest() %>%   
 mutate(betas = map(data, ~ coef(lm(ri ~ rm, data = .x))[2]),  
 sigmas = map(data, ~ sigma(lm(ri ~ rm, data = .x))),  
 yeartesting = 2008 + i) %>%   
 unnest(c(betas, sigmas)) %>%   
 select(permno, betas, sigmas, yeartesting)  
   
 # estimates are for 2008  
 testingperiod\_08to11 <- bind\_rows(testingperiod\_08to11, beta\_sigma)  
   
}  
  
testingperiod\_08to11 <- rename(testingperiod\_08to11, year = yeartesting)  
  
# join with df\_withpfid for all the testing years  
df\_testingperiod5 <- df\_withpfid %>%   
 filter(between(year, 2008, 2011)) %>%   
 left\_join(testingperiod\_08to11)

## Joining with `by = join\_by(permno, year)`

testing\_period5 <- df\_testingperiod5 %>%   
 group\_by(pfid, year, month) %>%   
 summarise(rp = mean(ri, na.rm = T),  
 betap = mean(betas, na.rm = T),  
 betap\_squared = mean(betas^2, na.rm = T),  
 sigmap = mean(sigmas, na.rm = T)) %>%   
 ungroup()

## `summarise()` has grouped output by 'pfid', 'year'. You can override using the  
## `.groups` argument.

#Portfolio Testing  
  
testing\_period5\_gammas <- testing\_period5 %>%   
 group\_by(year, month) %>%   
 nest(data = !c(year, month)) %>%   
 mutate(estim = map(data, ~lm(lead(rp) ~ betap + betap\_squared + sigmap, data = .x)),   
 estim = map(estim, tidy)) %>%   
 unnest(estim) %>%   
 select(year, month, term, estimate) %>%   
 ungroup()  
  
#Linearity  
gamma\_2t <- testing\_period5\_gammas %>%   
 filter(term == "betap\_squared") %>%   
 pull(estimate)  
  
# T test   
t.test(gamma\_2t, mu = 0)

##   
## One Sample t-test  
##   
## data: gamma\_2t  
## t = -0.038132, df = 47, p-value = 0.9697  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -2.318859 2.232587  
## sample estimates:  
## mean of x   
## -0.04313607

#No systematic effects of non-β risk  
gamma\_3t <- testing\_period5\_gammas %>%   
 filter(term == "sigmap") %>%   
 pull(estimate)  
  
# T test   
t.test(gamma\_3t)

##   
## One Sample t-test  
##   
## data: gamma\_3t  
## t = -1.0894, df = 47, p-value = 0.2815  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -0.4027281 0.1197795  
## sample estimates:  
## mean of x   
## -0.1414743

#Positive expected return-risk tradeoff  
gamma\_1t <- testing\_period5\_gammas %>%   
 filter(term == "betap") %>%   
 pull(estimate)  
  
# Rm  
rm\_data <- df\_testingperiod5 %>%   
 filter(permno == 10145) %>%   
 pull(rm)  
  
# Rm - R0  
gamma\_0t <- testing\_period5\_gammas %>%   
 filter(term == "(Intercept)") %>%   
 pull(estimate)  
  
rm\_r0 <- rm\_data - gamma\_0t  
  
t.test(gamma\_1t, rm\_r0) # t test

##   
## Welch Two Sample t-test  
##   
## data: gamma\_1t and rm\_r0  
## t = 0.22411, df = 71.126, p-value = 0.8233  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -5.544332 6.948495  
## sample estimates:  
## mean of x mean of y   
## 0.4191055 -0.2829762

length(gamma\_1t)

## [1] 48

length(rm\_r0)

## [1] 48

#Sharpe-Lintner (S-L) Hypothesis  
rft <- df\_testingperiod5 %>%   
 filter(permno == 10145) %>%   
 pull(rf)  
  
t.test(gamma\_0t, rft)

##   
## Welch Two Sample t-test  
##   
## data: gamma\_0t and rft  
## t = 0.57838, df = 47.003, p-value = 0.5658  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -1.943164 3.511334  
## sample estimates:  
## mean of x mean of y   
## 0.81958872 0.03550345

Box.test(x = gamma\_1t, lag = 1, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: gamma\_1t  
## X-squared = 3.6331, df = 1, p-value = 0.05664

Box.test(x = gamma\_2t, lag = 1, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: gamma\_2t  
## X-squared = 2.9474, df = 1, p-value = 0.08601

Box.test(x = gamma\_3t, lag = 1, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: gamma\_3t  
## X-squared = 0.10196, df = 1, p-value = 0.7495

Period-6.R

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.0 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(broom)  
  
df <- read.csv("USstocks\_balanced.csv")  
  
# Portfolio formation  
  
# Testing period 6  
pf\_groups <- df %>% filter(between(year, 2000, 2006)) %>%   
 group\_by(permno) %>%   
 nest() %>%   
 mutate(betaest = map(data, ~ coef(lm(ri ~ rm, data = .x))[2])) %>%   
 unnest(betaest) %>%   
 ungroup() %>%   
 mutate(pfid = ntile(betaest, 20)) %>%   
 select(permno, pfid)  
  
#create match between permno and portfolio groups  
#estimation period 2007-2011,testing period 2012-2015  
df\_withpfid <- df %>%   
 filter(between(year, 2007, 2015)) %>%   
 left\_join(pf\_groups)

## Joining with `by = join\_by(permno)`

#Portfolio Estimation  
  
#estimate market betas and standard deviation of residuals  
#Regressions start with the estimation period 2007-2011.   
#The beta and residual standard deviations are stored for the first year of  
#testing period which is the year 2012  
beta\_sigma\_12 <- df\_withpfid %>%   
 filter(year <= 2011) %>%   
 group\_by(permno) %>%   
 nest() %>%   
 mutate(betas = map(data, ~ coef(lm(ri ~ rm, data = .x))[2]),  
 sigmas = map(data, ~ sigma(lm(ri ~ rm, data = .x))),  
 yeartesting = 2012) %>%   
 unnest(c(betas, sigmas)) %>%   
 select(permno, betas, sigmas, yeartesting)  
  
#To get the estimates for the year 2012  
#we repeat the same procedure but including the additional year of 2012  
testingperiod\_12to15 <- data.frame()  
  
for(i in 0:3){  
   
 beta\_sigma <- df\_withpfid %>%   
 filter(year <= 2011 + i) %>%   
 group\_by(permno) %>%   
 nest() %>%   
 mutate(betas = map(data, ~ coef(lm(ri ~ rm, data = .x))[2]),  
 sigmas = map(data, ~ sigma(lm(ri ~ rm, data = .x))),  
 yeartesting = 2012 + i) %>%   
 unnest(c(betas, sigmas)) %>%   
 select(permno, betas, sigmas, yeartesting)  
   
 # estimates are for 2012  
 testingperiod\_12to15 <- bind\_rows(testingperiod\_12to15, beta\_sigma)  
   
}  
  
testingperiod\_12to15 <- rename(testingperiod\_12to15, year = yeartesting)  
  
# join with df\_withpfid for all the testing years  
df\_testingperiod6 <- df\_withpfid %>%   
 filter(between(year, 2012, 2015)) %>%   
 left\_join(testingperiod\_12to15)

## Joining with `by = join\_by(permno, year)`

testing\_period6 <- df\_testingperiod6 %>%   
 group\_by(pfid, year, month) %>%   
 summarise(rp = mean(ri, na.rm = T),  
 betap = mean(betas, na.rm = T),  
 betap\_squared = mean(betas^2, na.rm = T),  
 sigmap = mean(sigmas, na.rm = T)) %>%   
 ungroup()

## `summarise()` has grouped output by 'pfid', 'year'. You can override using the  
## `.groups` argument.

#Portfolio Testing  
  
testing\_period6\_gammas <- testing\_period6 %>%   
 group\_by(year, month) %>%   
 nest(data = !c(year, month)) %>%   
 mutate(estim = map(data, ~lm(lead(rp) ~ betap + betap\_squared + sigmap, data = .x)),   
 estim = map(estim, tidy)) %>%   
 unnest(estim) %>%   
 select(year, month, term, estimate) %>%   
 ungroup()  
  
#Linearity  
gamma\_2t <- testing\_period6\_gammas %>%   
 filter(term == "betap\_squared") %>%   
 pull(estimate)  
  
# T test   
t.test(gamma\_2t, mu = 0)

##   
## One Sample t-test  
##   
## data: gamma\_2t  
## t = 0.78154, df = 47, p-value = 0.4384  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -1.460344 3.315843  
## sample estimates:  
## mean of x   
## 0.9277498

#No systematic effects of non-β risk  
gamma\_3t <- testing\_period6\_gammas %>%   
 filter(term == "sigmap") %>%   
 pull(estimate)  
  
# T test   
t.test(gamma\_3t)

##   
## One Sample t-test  
##   
## data: gamma\_3t  
## t = 0.031412, df = 47, p-value = 0.9751  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -0.2293928 0.2366700  
## sample estimates:  
## mean of x   
## 0.003638616

#Positive expected return-risk tradeoff  
gamma\_1t <- testing\_period6\_gammas %>%   
 filter(term == "betap") %>%   
 pull(estimate)  
  
# Rm  
rm\_data <- df\_testingperiod6 %>%   
 filter(permno == 10145) %>%   
 pull(rm)  
  
# Rm - R0  
gamma\_0t <- testing\_period6\_gammas %>%   
 filter(term == "(Intercept)") %>%   
 pull(estimate)  
  
rm\_r0 <- rm\_data - gamma\_0t  
  
t.test(gamma\_1t, rm\_r0) # t test

##   
## Welch Two Sample t-test  
##   
## data: gamma\_1t and rm\_r0  
## t = -0.46746, df = 64.325, p-value = 0.6418  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -7.870614 4.885474  
## sample estimates:  
## mean of x mean of y   
## -2.0552326 -0.5626627

length(gamma\_1t)

## [1] 48

length(rm\_r0)

## [1] 48

#Sharpe-Lintner (S-L) Hypothesis  
rft <- df\_testingperiod6 %>%   
 filter(permno == 10145) %>%   
 pull(rf)  
  
t.test(gamma\_0t, rft)

##   
## Welch Two Sample t-test  
##   
## data: gamma\_0t and rft  
## t = 1.3939, df = 47, p-value = 0.1699  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -0.7345644 4.0487328  
## sample estimates:  
## mean of x mean of y   
## 1.661858527 0.004774306

Box.test(x = gamma\_1t, lag = 1, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: gamma\_1t  
## X-squared = 0.00015755, df = 1, p-value = 0.99

Box.test(x = gamma\_2t, lag = 1, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: gamma\_2t  
## X-squared = 0.0001653, df = 1, p-value = 0.9897

Box.test(x = gamma\_3t, lag = 1, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: gamma\_3t  
## X-squared = 2.7284, df = 1, p-value = 0.09858