Analysis

Erik Andersen

2024-05-25

Question 1

In this question, I will replicate the mean return patterns for the Fama-French 25. These are the portfolios formed on size, and book-to-market value weighted. The values are given in table 1 (This table didn't round for some reason I don't know why).

Table 1:

	Mean Return (% Per Month)				
	Low B/M	2	3	4	High B/M
Small	0.305390300546448	0.775708333333333	0.779899590163935	0.983272131147541	1.1271162568306
2	0.522531557377049	0.795162295081967	0.860273360655738	0.909621994535519	1.03100163934426
3	0.546485109289617	0.809150546448087	0.754173360655738	0.894013387978142	1.01733633879781
4	0.652604918032787	0.657038114754098	0.727220081967213	0.851180327868853	0.884083879781421
Big	0.594943169398907	0.555158333333333	0.589210519125683	0.531210519125683	0.70213306010929

Question 2

In this question, we will see if the Fama, French finding that market β's are approximately 1 holds up when we only include the market risk premium as a risk factor in the estimating regrssion.

a)

To do that we will run the regression shown in equation (1) below. The α's, β's, and their requisite t-statistics are shown in table 2.

$$R^e_{i,t} = \alpha_i + \beta_i RmRf_t + \epsilon_{i,t} \tag{1}$$

b)

i. The GRS test gives a test statistic of 4.3713 which has a p-value of 1.778×10^{-11}. It is highly significant. The critical values it would need to achieve for 1, 5, and 10 percent significance are reported in table 3.

The $\chi\hat{}2$ test gives a value of 98.4161 which has a p-value of 1.158×10^{-10}. Table 4 reports reports the same critical values but for the chi-squared test as above.

Both tests strongly reject the null of the $alpha$'s being jointly 0.

ii. The mean absolute value of the α's is 0.208.

Table 2:

	Estimate					T-Statistic				
	Low B/M	2	3	4	High B/M	Low B/M	2	3	4	High B/M
Alpha										
Small	-0.5107	0.0690	0.1391	0.3825	0.5095	-2.7940	0.4311	1.070	2.8940	3.332
2	-0.2738	0.1124	0.2391	0.3190	0.3659	-2.0063	1.0082	2.318	3.0369	2.758
3	-0.2069	0.1610	0.1694	0.3134	0.3881	-1.8610	1.8716	2.005	3.2946	3.084
4	-0.0403	0.0334	0.1426	0.2740	0.2589	-0.4733	0.4900	1.804	3.0236	2.151
Big	0.0252	0.0228	0.0927	0.0176	0.1337	0.4213	0.3977	1.246	0.1811	1.002
Beta										
Small	1.4144	1.2249	1.1107	1.0412	1.0704	34.9893	34.6218	38.644	35.6202	31.651
2	1.3802	1.1834	1.0766	1.0237	1.1528	45.7287	48.0001	47.207	44.0621	39.285
3	1.3058	1.1233	1.0135	1.0063	1.0906	53.0963	59.0276	54.243	47.8262	39.189
4	1.2009	1.0809	1.0133	1.0004	1.0837	63.8357	71.7057	57.949	49.9177	40.719
Big	0.9874	0.9228	0.8606	0.8902	0.9852	74.5303	72.8998	52.297	41.3587	33.383

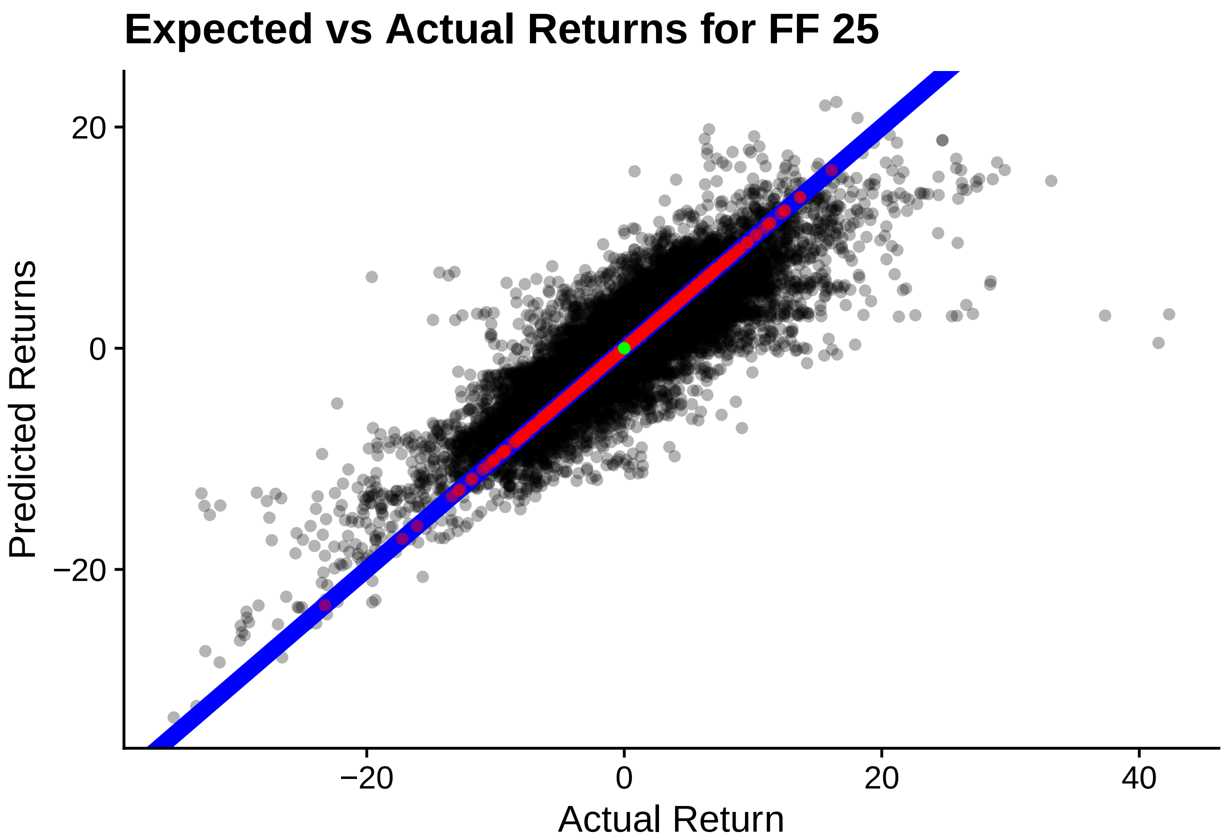
Table 3:

Critical Values		
10%	5%	1%
1.386	1.522	1.799

Table 4:

Critical Values		
10%	5%	1%
34.38	37.65	44.31

iii. Below is the plot of actual versus predicted returns. The blue line is the 45 degree line. The red points on the line are the market excess returns, and the green dot is the riks free rate.



c)

The CAPM does not work well on these portfolios. The null hypothesis of the α's being jointly 0 is strongly rejected by both tests. Individually, most of the α's also have large t-statistics. Statistically, the α's definitely matter. The mean absolute α is 0.208. This is fairly large I think. It is a fifth of the market β which seems like it matters economically.

There are predictable patterns in the mean returns we see in table 1. Generally, small firms have bigger returns that large firms, and high book to market firms have bigger returns than low book to market firms. This pattern seems to vaguely match the β's in table 2. The largest β's are the smaller firms with lower book to market ratios which were also the firms with larger mean returns.

d)

Now, I'll run the cross sectional regressions using the equation given by (2). I'll run it using only the Fama French assets, and then also include the market excess returns and the risk free asset. The results are given in table 5.

The χ^2 test only using the ff25 is 1.6056 which has p-value 1. Including the other assets they are 2.8648 and 1. The test statistics are very small which is why the p-values are so high. Thus we fail to reject the null that the α's are jointly 0.

This result is wildly different from the time series regression. The λ's are much smaller than any of the β's. When we don't include the market and risk free rates, the λ is negative. We also don't reject the null here of all α's being 0 which we did for the time series.

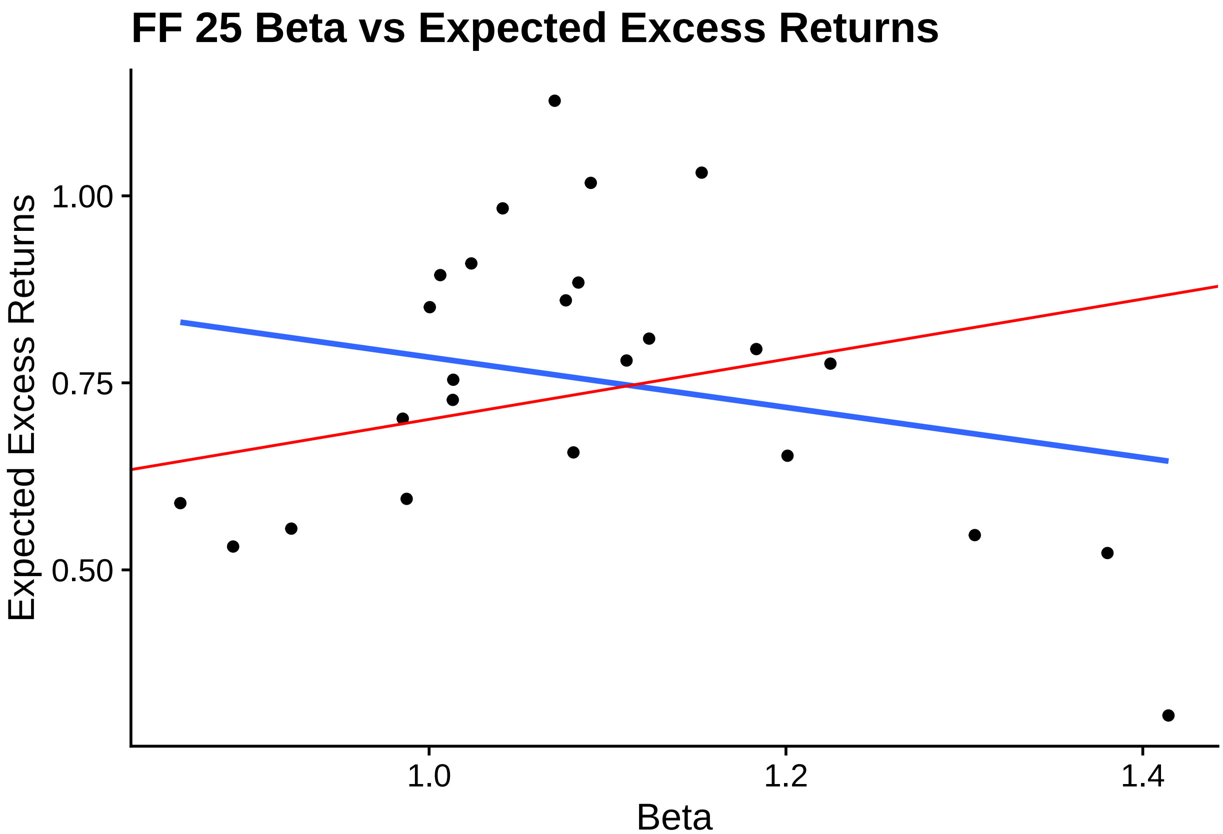
$$E[R_i^e] = \gamma + \beta_i \lambda + \alpha_i \tag{2}$$

Table 5:

	Gamma	Lambda	SE Lambda	Mean Absolute Alpha
FF 25	1.1197	-0.3354	0.2821	0.1536
FF25 + RMRF + RF	0.2991	0.4021	0.1729	0.1641

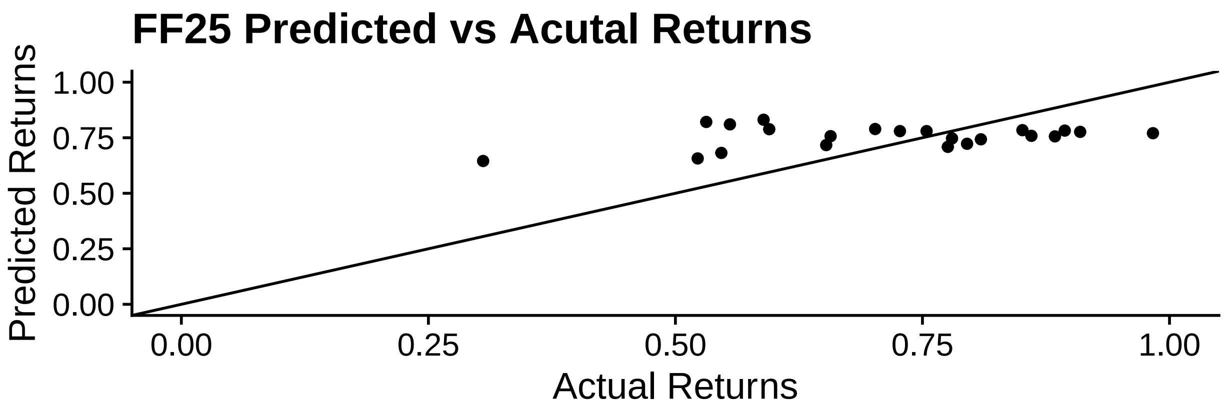
e)

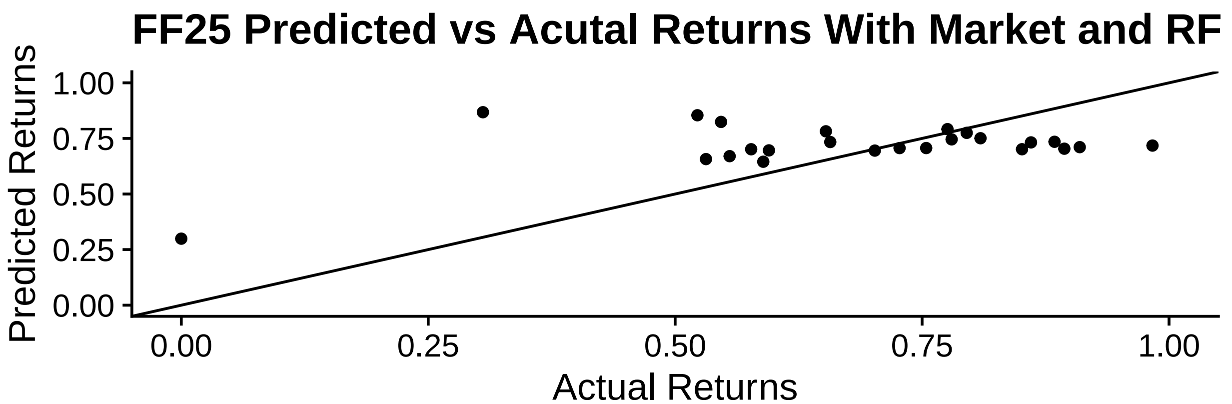
i. Below is the graph showing the relationship between β's and expected excess returns for the test assets. The blue line is the relationship not including the market and risk free assets. The red line includes those. We can see they are substantially different.



ii. Below is a plot of predicted versus actual returns for the test assets. The top plot uses the regression with only the test assets. The bottom uses the regression that uses the the market and risk free assets to make the predictions. It looks like using the whole sample leads to a better fit. The bottom graph has the points more clustered around the 45 degree line.

```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_point()`).
## Removed 3 rows containing missing values or values outside the scale range
## (`geom_point()`).
```





Question 3

In this question, I will rerun the analyis in question 2 but using equation (3).

a)

Table 6 shows the estimates from the regression. Table 7 shows the standard errors of a, and the R^2 values.

$$R_i - R_f = a_i + b_i(R_M - R_f) + s_i SMB + h_i HML + e_i \tag{3}$$

b)

Fama and French's patterns do hold up in the longer sample. First we can notice that there is still no spread in the β's. The other two factors move in the same way they do in the Fama French paper. The R^2's are similarly high. And finally there are several high t-statistics on the a's.

c)

The GRS test for the three factor portfolio is 3.7968 which has a p-value of 2.3793×10^{-9}. The χ^2 statistic is 98.4161 which has a p-value of 1.158×10^{-10}. The mean absolute value of the α's is 0.0944.

Table 6:

Mean Return (% Per Month)					
	Low B/M	2	3	4	High B/M
a					
Small	-0.5013	-0.0200	-0.0413	0.1316	0.1731
2	-0.2097	0.0127	0.0511	0.0588	0.0042
3	-0.1109	0.0707	-0.0142	0.0521	0.0308
4	0.0792	-0.0506	-0.0293	0.0441	-0.0783
Big	0.1748	0.0075	-0.0045	-0.2236	-0.1844
b					
Small	1.0964	0.9598	0.9290	0.8883	0.9417
2	1.1288	1.0144	0.9741	0.9540	1.0820
3	1.1042	1.0235	0.9817	1.0026	1.0882
4	1.0695	1.0609	1.0369	1.0302	1.1352
Big	0.9875	0.9706	0.9451	1.0222	1.1286
s					
Small	1.4084	1.3274	1.0985	1.0813	1.1083
2	1.0233	0.9140	0.7567	0.7234	0.8893
3	0.7503	0.5897	0.4330	0.4301	0.5766
4	0.3988	0.2223	0.1663	0.2307	0.3031
Big	-0.2371	-0.1901	-0.2243	-0.2090	-0.1380
h					
Small	-0.2597	0.0112	0.2889	0.4764	0.6958
2	-0.3388	0.1084	0.3659	0.5605	0.7986
3	-0.3767	0.1381	0.4083	0.6125	0.8392
4	-0.3795	0.1829	0.4224	0.5634	0.8322
Big	-0.3521	0.0716	0.2920	0.6666	0.8558

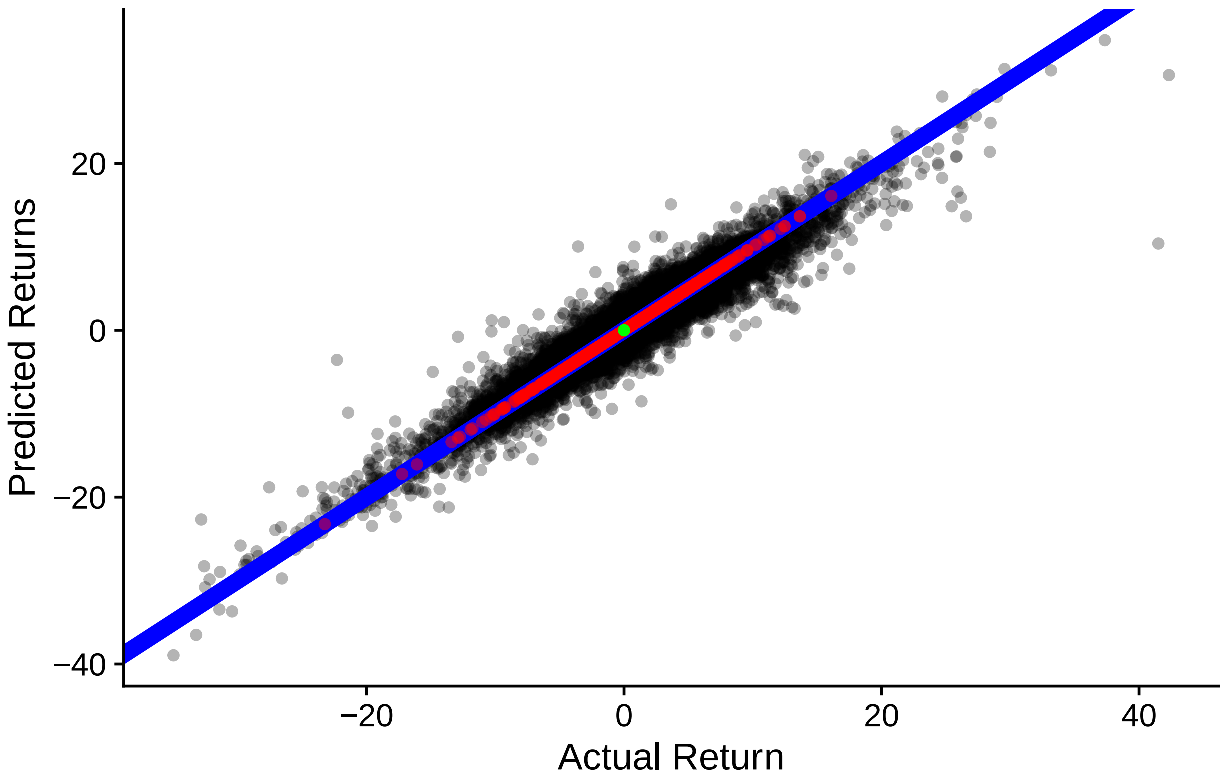
Table 7:

	Low B/M	2	3	4	High B/M
t(a)					
Small	c(statistic = -5.40638791927437)	c(statistic = -0.280699034536286)	c(statistic = -0.810730554444522)	c(statistic = 2.6838036809442)	c(statistic = 2.38444178634222)
2	c(statistic = -3.23869265059623)	c(statistic = 0.239849361011243)	c(statistic = 0.918955247638113)	c(statistic = 1.24560545633914)	c(statistic = 0.0818439114197013)
3	c(statistic = -1.8924543672117)	c(statistic = 1.21715946975781)	c(statistic = -0.240273087281493)	c(statistic = 0.928156088901413)	c(statistic = 0.44078797189263)
4	c(statistic = 1.37002516788438)	c(statistic = -0.816236509098277)	c(statistic = -0.460118353009339)	c(statistic = 0.678316497027607)	c(statistic = -1.01730961898882)
Big	c(statistic = 4.23926690262167)	c(statistic = 0.142142809733532)	c(statistic = -0.0727842940660477)	c(statistic = -3.83807131147297)	c(statistic = -1.96988458194656)
R2					
Small	0.905818519462402	0.926566280143147	0.950549343044562	0.950756556165132	0.90691229198422
2	0.942949665883881	0.946959672419274	0.929809992700147	0.945943083902876	0.953770136367553
3	0.944016912424872	0.922700882463926	0.905249081182746	0.917534356321696	0.902720608243064
4	0.931198459283688	0.899092017107044	0.886244841021857	0.885738206078375	0.877467518936404
Big	0.946079995687769	0.898360370550247	0.856603355166194	0.894974276219905	0.809302322835628

d)

Below is a plot of the predicted vs the actual returns. As before, the blue line is the 45 degree line, the red points are the market returns, and the green point is the risk free asset.

Expected vs Actual Returns Using 3 Factors



e)

Finally, I will compare the two models. The first thing to note is that both models are “rejected” by the GRS and χ^2 tests. We easily reject the null that the α's are jointly 0. However, the 3 factor model has much smaller α's. The mean absolute α is 0.0944 versus 0.208 for the CAPM. So the pricing error is halved.

This tells us that reducing the testing of models to a single test statistic is overly simplistic. Despite having similar test statistics, the 3F model clearly explains more of the variation here. The graphs show it is a much better fit. As discussed in class, the 3 factor model shrinks the α's but also shrinks the variance-covariance matrix. These shrinking simultaneously means we end up with a similar test statistic because the smaller α's are measured better.

Code Appendix

Erik Andersen

2024-05-27

```r
here::i_am("HW5/code/HW5.R")

# Load packages
pacman::p_load(tidyverse, ggplot2, magrittr)


########### Load data ####################

portfolios_df = read_csv(here::here("HW5", "data", "25_Portfolios_5x5.CSV"),
                         skip = 15) |> # This is just trial and error to get rid of extra lines at the
  mutate(date = ym(`...1`)) |>
  # Rearrange and drop badly formatted date column
  select(date, everything(), -1) |>
  # The last row is just NAs so remove it
  filter(row_number() < 8723) |>
  # We only need after January 1963 and before December 2023
  filter(year(date) >= 1963 & year(date) <= 2023) |>
  # This data set has annualized returns and several other equal rated returns and several other sepeci
  filter(row_number() <= 732) |>
  # Clean the names
  janitor::clean_names()

ff_df = read_csv(here::here("HW5", "data", "F-F_Research_Data_Factors.CSV"),
                 skip = 3) |>
  # Same process
  mutate(date = ym(...1)) |>
  select(date, everything(), -1) |>
  # There are annual factors at the bottom we don't need
  filter(row_number() <= 1173) |>
  # Select same interval
  filter(year(date) >= 1963 & year(date) <= 2023) |>
  # Clean names
  janitor::clean_names()

# Merge data sets
returns_df = left_join(ff_df, portfolios_df)

# Create excess returns for each portfolio
returns_df %<>%
  # 6:length(returns_df) are the columns of interest
  # All the returns are characters for some reason so make them numeric
  mutate(across(6:length(returns_df), ~ as.numeric(.x)),
```

```r
         # Subtract the risk free rate from each of the return columns
         across(6:length(returns_df), ~ .x - rf))

# Extract the relevant names of the columns to loop over
name = names(returns_df)
# We only want the names of the different sortings, so select those
name = name[6:length(name)]

########## Question 1 ####################

# Calculate mean returns for each portfolio
mean_returns = map(name, function(x){

  # Calculate mean returns. Selecting the column this way gives a list data type so we have to unlist i
  return(mean(unlist(returns_df[,x])))
}) |>
  # Organize sensibly
  matrix(nrow = 5, byrow = T)

# Set the names
rownames(mean_returns) = c("Small","2","3","4","Big")
colnames(mean_returns) = c("Low B/M", "2", "3", "4", "High B/M")
```

```r
########## Question 2 ####################

# Run the regression for each portfolio
coefficients_1f = map(name, function(x){

  # Run regression for each column
  estimates = lm(unlist(returns_df[,x]) ~ returns_df$mkt_rf) |>
    broom::tidy() |>
    select(estimate, statistic)


  return(estimates)
})

# Extract alphas
alpha = map_dbl(1:length(coefficients_1f), function(i){
  coefficients_1f[[i]] |>
    select(estimate) |>
    nth(1) |>
    unlist()
}) |>
  # Organize by portfolio
  matrix(nrow = 5, byrow = T)

# Set the names
```

```r
rownames(alpha) = c("Small","2","3","4","Big")
colnames(alpha) = c("Low B/M", "2", "3", "4", "High B/M")

# Extract alpha t-stats
alpha_t_stat = map_dbl(1:length(coefficients_1f), function(i){
  coefficients_1f[[i]] |>
    select(statistic) |>
    nth(1) |>
    unlist()
}) |>
  # Organize by portfolio
  matrix(nrow = 5, byrow = T)

# Set the names
rownames(alpha_t_stat) = c("Small","2","3","4","Big")
colnames(alpha_t_stat) = c("Low B/M", "2", "3", "4", "High B/M")


# Extract betas
beta = map_dbl(1:length(coefficients_1f), function(i){
  coefficients_1f[[i]] |>
    select(estimate) |>
    nth(2) |>
    unlist()
}) |>
  # Organize by portfolio
  matrix(nrow = 5, byrow = T)

# Set the names
rownames(beta) = c("Small","2","3","4","Big")
colnames(beta) = c("Low B/M", "2", "3", "4", "High B/M")

# Extract beta t-stats
beta_t_stat = map_dbl(1:length(coefficients_1f), function(i){
  coefficients_1f[[i]] |>
    select(statistic) |>
    nth(2) |>
    unlist()
}) |>
  # Organize by portfolio
  matrix(nrow = 5, byrow = T)

# Set the names
rownames(beta_t_stat) = c("Small","2","3","4","Big")
colnames(beta_t_stat) = c("Low B/M", "2", "3", "4", "High B/M")


# Rerun regression to get residuals
residuals_1f = sapply(name, function(x){

  # Run regression for each column
  res = lm(unlist(returns_df[,x]) ~ returns_df$mkt_rf)$residuals

```

```r
  return(res)
}) |>
  # Transpose to get into right form
  t()

# Calculate GRS test
# Parameters
alphas = c(t(alpha))
N = nrow(residuals_1f)
t = ncol(residuals_1f)

# Covariance
cov_p = residuals_1f %*% t(residuals_1f)/ t

# Two bits of formula
first = (1 + mean(returns_df$mkt_rf)^2/var(returns_df$mkt_rf))^(-1)
second = as.numeric(t(alphas) %*% solve(cov_p) %*% alphas)

# GRS statistic
GRS_1f = (t - N - 1)/N * first * second

# P-value
GRS_1f_p_value = 1 - pf(GRS_1f, N, t-N-1)

# Critical values
f_crits = cbind(round(qf(1-0.1, N, t-N-1), 4),
                round(qf(1-0.05, N, t-N-1), 4),
                round(qf(1-0.01, N, t-N-1), 4))

colnames(f_crits) = c("10%", "5%", "1%")

# Chi-squared critical values
chi_crits = cbind(round(qchisq(1-0.1, N), 4),
                  round(qchisq(1-0.05, N), 4),
                  round(qchisq(1-0.01, N), 4))

colnames(chi_crits) = c("10%", "5%", "1%")

### Calculate mean absolute alphas
mean_abs_alpha = mean(abs(alpha))

### Plot expected vs actual returns

# Calculate the predicted returns
predicted_returns = map(name, function(x){
  reg = lm(unlist(returns_df[,x]) ~ returns_df$mkt_rf)

  predicted = predict(reg, returns_df[,x])

  return(predicted)
})

# Plot
```

```r
plot_1f = returns_df |>
  # Select only the portfolios
  select(name) |>
  # Pivot longer for plotting
  pivot_longer(cols = everything()) |>
  # The previous function doesn't group the portfolios together. This does. The weird factor thing lets
  arrange(factor(name, levels = names(returns_df)[6:length(returns_df)]), name) |>
  cbind(unlist(predicted_returns)) |>
  # Cbind gives stupid names so fix that
  rename("actual" = value,
         "predicted" = `unlist(predicted_returns)`) |>
  ggplot(aes(x = actual, y = predicted)) +
  geom_point(alpha = 0.3) +
  # 45 Degree line
  geom_abline(intercept = 0, slope = 1, size = 3, color = 'blue') +
  # Market excess returns
  geom_point(data = returns_df, aes(x = mkt_rf, y = mkt_rf), col = 'red', alpha = 0.5) +
  # Risk free rate
  annotate("point", x = 0, y = 0, col = 'green') +
  cowplot::theme_cowplot() +
  labs(x = "Actual Return", y = "Predicted Returns", title = "Expected vs Actual Returns for FF 25")

ggsave(here::here("HW5", "plots", "one_factor.pdf"))



# Cross sectional regression on only ff25
reg_cross = lm(as.numeric(mean_returns) ~ as.numeric(beta))

# Cross sectional regression on that plus the whole market and the risk free asset
reg_cross_rf = lm(c(as.numeric(mean_returns), mean(returns_df$mkt_rf), 0) ~ c(as.numeric(beta), 1, 0))


# make the table
ff25_cross = c(reg_cross$coefficients, broom::tidy(reg_cross)$std.error[2], mean(abs(reg_cross$residuals
ff25_extra_cross = c(reg_cross_rf$coefficients, broom::tidy(reg_cross_rf)$std.error[2], mean(abs(reg_cro

cross_table = rbind(ff25_cross, ff25_extra_cross)
colnames(cross_table) = c("Gamma", "Lambda", "SE Lambda", "Mean Absolute Alpha")
rownames(cross_table) = c("FF 25", "FF25 + RMRF + RF")

# Chi square test for both regressions
ff25_chi = chisq.test(abs(reg_cross$residuals))
ff25_extra_chi = chisq.test(abs(reg_cross_rf$residuals))


# Plot expected excess returns versus betas
plot_ff25 = tibble(excess_return = as.numeric(mean_returns),
       beta = as.numeric(beta)) |>
  ggplot(aes(x = beta, y = excess_return)) +
  geom_point() +
  geom_smooth(method = 'lm', se = F) +
  geom_abline(intercept = reg_cross_rf$coefficients[1], slope = reg_cross_rf$coefficients[2], col = 're
```

```r
  cowplot::theme_cowplot() +
  labs(x = "Beta", y = "Expected Excess Returns", title = "FF 25 Beta vs Expected Excess Returns")

ggsave(here::here("HW5", "plots", "ff25.pdf"))



# Predicted versus actual returns for only ff25
plot_ff_predicted = tibble(actual = as.numeric(mean_returns),
       predicted = actual - reg_cross$residuals) |>
  ggplot(aes(x = actual, y = predicted)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1) +
  cowplot::theme_cowplot() +
  ylim(0,1) +
  xlim(0,1) +
  labs(x = "Actual Returns", y = "Predicted Returns", title =
         "FF25 Predicted vs Acutal Returns")
ggsave(here::here("HW5", "plots", "ff.pdf"))

# Predicted versus actual returns for whole sample
plot_all_predicted = tibble(actual = c(as.numeric(mean_returns), mean(returns_df$mkt_rf), 0),
       predicted = actual - reg_cross_rf$residuals) |>
  ggplot(aes(x = actual, y = predicted)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1) +
  cowplot::theme_cowplot() +
  ylim(0,1) +
  xlim(0,1) +
  labs(x = "Actual Returns", y = "Predicted Returns", title =
         "FF25 Predicted vs Acutal Returns With Market and RF Assets")
ggsave(here::here("HW5", "plots", "full_sample.pdf"))



########## Question 3 ####################




# Run the regression for each of the relevant columns
coefficients_3f = map(name, function(x){

  # Run regression for each column
  estimates = lm(unlist(returns_df[,x]) ~ returns_df$mkt_rf + returns_df$smb + returns_df$hml) |>
            broom::tidy() |>
            select(estimate)


  return(estimates)
})

# Get a vector for each different coefficient
```

```r
# Constant
a = map_dbl(1:length(coefficients_3f), function(i){
  coefficients_3f[[i]] |> nth(1) |> unlist()
}) |>
  # Organize
  matrix(nrow = 5, byrow = T)

# Set the names
rownames(a) = c("Small","2","3","4","Big")
colnames(a) = c("Low B/M", "2", "3", "4", "High B/M")


# Excess market risk
b = map_dbl(1:length(coefficients_3f), function(i){
  coefficients_3f[[i]] |> nth(2) |> unlist()
}) |>
  # Organize
  matrix(nrow = 5, byrow = T)

# Set the names
rownames(b) = c("Small","2","3","4","Big")
colnames(b) = c("Low B/M", "2", "3", "4", "High B/M")


# Small minus big
s = map_dbl(1:length(coefficients_3f), function(i){
  coefficients_3f[[i]] |> nth(3) |> unlist()
}) |>
  # Organize
  matrix(nrow = 5, byrow = T)

# Set the names
rownames(s) = c("Small","2","3","4","Big")
colnames(s) = c("Low B/M", "2", "3", "4", "High B/M")


# Value minus growth
h = map_dbl(1:length(coefficients_3f), function(i){
  coefficients_3f[[i]] |> nth(4) |> unlist()
}) |>
  # Organize
  matrix(nrow = 5, byrow = T)

# Set the names
rownames(h) = c("Small","2","3","4","Big")
colnames(h) = c("Low B/M", "2", "3", "4", "High B/M")

# Standard errors of alphas
a_se = map(name, function(x){

  # Run regression for each column
  estimates = lm(unlist(returns_df[,x]) ~ returns_df$mkt_rf + returns_df$smb + returns_df$hml) |>
    broom::tidy() |>
```

```
    select(statistic) |>
    nth(1) |> unlist()


  return(estimates)
}) |>
  matrix(nrow = 5, byrow = T)
rownames(a_se) = c("Small","2","3","4","Big")
colnames(a_se) = c("Low B/M", "2", "3", "4", "High B/M")

# Get R^2
R2 = map(name, function(x){

  # Run regression for each column
  estimates = lm(unlist(returns_df[,x]) ~ returns_df$mkt_rf + returns_df$smb + returns_df$hml) |>
    summary()


  return(estimates$r.squared)
}) |>
  matrix(nrow = 5, byrow = T)
rownames(R2) = c("Small","2","3","4","Big")
colnames(R2) = c("Low B/M", "2", "3", "4", "High B/M")



# Rerun regression to get residuals
residuals_3f = sapply(name, function(x){

  # Run regression for each column
  res = lm(unlist(returns_df[,x]) ~ returns_df$mkt_rf + returns_df$smb + returns_df$hml)$residuals


  return(res)
}) |>
  # Transpose to get into right form
  t()

# Calculate GRS test
# Parameters
alphas = c(t(a))
N = nrow(residuals_3f)
t = ncol(residuals_3f)

# Covariance
cov_p = residuals_3f %*% t(residuals_3f)/ t

# Risk factors
mean_rf = c(mean(returns_df$mkt_rf), mean(returns_df$smb), mean(returns_df$hml))
var_rf = var(returns_df[,2:4])

# Two bits of formula
first = (1 + t(mean_rf) %*% solve(var_rf) %*% mean_rf)^(-1)
```

```
second = as.numeric(t(alphas) %*% solve(cov_p) %*% alphas)

# GRS statistic
GRS_3f = (t - N - 1)/N * first * second

# P-value
GRS_3f_p_value = 1 - pf(GRS_3f, N, t-N-1)


# Chi-square test
chi_3f = t*first*second
chi_3f_p = 1 - pchisq(t*first*second, N)

# MEan absolute alpha
mean_abs_alpha_3f = mean(abs(alphas))

# Calculate the predicted returns
predicted_returns_3f = map(name, function(x){
  reg = lm(unlist(returns_df[,x]) ~ returns_df$mkt_rf + returns_df$smb + returns_df$hml)

  predicted = predict(reg, returns_df[,x])

  return(predicted)
})

# Plot
plot_predicted_3f = returns_df |>
  # Select only the portfolios
  select(name) |>
  # Pivot longer for plotting
  pivot_longer(cols = everything()) |>
  # The previous function doesn't group the portfolios together. This does. The weird factor thing lets
  arrange(factor(name, levels = names(returns_df)[6:length(returns_df)]), name) |>
  cbind(unlist(predicted_returns_3f)) |>
  # Cbind gives stupid names so fix that
  rename("actual" = value,
         "predicted" = `unlist(predicted_returns_3f)`) |>
  ggplot(aes(x = actual, y = predicted)) +
  geom_point(alpha = 0.3) +
  # 45 Degree line
  geom_abline(intercept = 0, slope = 1, size = 3, color = 'blue') +
  # Market excess returns
  geom_point(data = returns_df, aes(x = mkt_rf, y = mkt_rf), col = 'red', alpha = 0.5) +
  # Risk free rate
  annotate("point", x = 0, y = 0, col = 'green') +
  cowplot::theme_cowplot() +
  labs(x = "Actual Return", y = "Predicted Returns", title = "Expected vs Actual Returns Using 3 Factors

ggsave(here::here("HW5", "plots", "expected_3f.pdf"))
```