Code Appendix

Erik Andersen

2024-05-16

```
### HW 4

here::i_am("HW4/code/HW4.R")

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

## Load data
ff_monthly_df = read_csv(here::here("HW4", "data", "F-F_Research_Data_Factors.CSV"),
                                # Skip the header
                                skip = 2,
                                # Just need excess and risk free returns
                                col_select = c(1 ,"Mkt-RF", "RF")) |>
  # There's annual data at the bottom for some reason so this get's rid of that
  filter(row_number() <= 1173) |>
  # Convert to useful date format
  mutate(date = ym(...1),
         date = as_date(date)) |>
  # Remove badly formatted date column
  select(-1)




ff_daily_df = read_csv(here::here("HW4", "data", "F-F_Research_Data_Factors_daily.CSV"),
                               # Header is 3 lines long this time
                               skip = 3,
                               col_select = c(1, "Mkt-RF", "RF")) |>
  rename("date" = ...1)




daily_industry_df = read_csv(here::here("HW4", "data", "30_Industry_Portfolios_Daily.csv"),
                             skip = 9) |>
  rename("date" = ...1)

daily_industry_df = daily_industry_df |>
  # Last row is copyright
  slice(-nrow(daily_industry_df))
```

```r
crsp_monthly_df = read_csv(here::here("HW4", "data", "CRSP_monthly.csv")) |>
  # Select share codes we need
  filter(SHRCD == 10 | SHRCD == 11) |>
  # We don't need this column anymore
  select(-SHRCD) |>
  # Set day of date to 1. I'm doing this because its monthly data, so the day doesn't matter and this m
  mutate(date = paste(year(date), month(date), 1, sep = "-"),
         date = as_date(date))




###### Clean data ########################

# Drop firms with no ticker
crsp_monthly_df = crsp_monthly_df |>
  mutate(RET = as.numeric(RET)) |>
  drop_na(RET)

# Filter so we only have firms with at least two years of data
# Calculate how many observations each firm has. Keep if greater than 24 months
date_range = map_df(unique(crsp_monthly_df$TICKER), function(x){

  dates = crsp_monthly_df |>
    # Select each firm
    filter(TICKER == {x}) |>
    nrow()

  # Convert into convenient format
  tmp = tibble(TICKER = {x}, data_length = dates)

return(tmp)
}) |>
  # Select only the firms with enough data#
  filter(data_length >= 24)


# Filter original data set to only include firms with enough data
crsp_monthly_df = crsp_monthly_df |>
  filter(TICKER %in% date_range$TICKER)




# Merge CRSP data and market data
monthly_df =
  crsp_monthly_df |>
    left_join(ff_monthly_df) |>
  # Calculate excess returns = individual return - risk free rate
```

```r
  # Multiply by 100 to make it same units as other data set
  mutate(excess_return = RET*100) |>
  janitor::clean_names()


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

# Calculate beta for individual firms,
monthly_df =
  monthly_df |>
  group_by(ticker) |>
  # Regress excess returns on market return to get beta. Nth(2) gives us beta rather than intercept
  summarise(beta = lm(excess_return ~ mkt_rf) |> coef() |> nth(2),

            # Need average returns to calculate cross sectional regression
            average_return = mean(excess_return)) |>
  # Join back into original data
  right_join(monthly_df)


# Calculate cross sectional regression
reg_cross_1 = monthly_df %>% lm(average_return ~ beta,.)


# Graph scatter plot
unconditional_plot =
  monthly_df |>
    ggplot(aes(x = beta, y = average_return)) +
    geom_point() +
    geom_smooth(method = 'lm', se = F) +
    labs(x = "Beta", y = "Average Return (%)") +
    ggtitle("Beta vs. Mean Return in Sample") +
    cowplot::theme_cowplot()
ggsave(here::here("HW4", "plots", "unconditional.png"))

# Calculate time series average
time_average = ff_monthly_df |> filter(year(date) >= 2014) |> summarise(mean = mean(`Mkt-RF`))


##### Question 2 ############

# Create a temporary data structure with the betas arranged by month
temp = monthly_df |>
  group_by(date, ticker) |> # Gather each month together by ticker
  summarise(beta = mean(beta), # Get the beta for each firm by month
            mkt = mean(mkt_rf), # Keep market rate
            ret = mean(ret)) |>
  arrange(date, beta) # Put in ascending order of betas so we can find the quantiles by month

# For each month split into 20 groups
portfolio_df =
  map_df(unique(temp$date), function(x){
    temp |>
```

```
      filter(date == {x}) |>
      mutate(portfolio = ntile(beta, 20)) |>  # Break each month into 20 groups
      group_by(date, portfolio) |>
      summarise(excess_returns = mean(ret)*100, # Calculate excess returns for each portfolio
                mkt = mean(mkt))})


#### c
# Run time series regression for each portfolio

portfolio_df =
  portfolio_df |>
    group_by(portfolio) |>
    summarise(beta = lm(excess_returns ~ mkt) |> coef() |> nth(2),
              alpha = lm(excess_returns ~ mkt) |> coef() |> nth(1),
              average_return = mean(excess_returns)) |>
  right_join(portfolio_df)

#### d
# Cross sectional regression

reg_cross_2 = portfolio_df %>% lm(average_return ~ beta,.)

#### e
# Graph cross sectional regression

portfolio_plot = portfolio_df |>
  ggplot(aes(x = beta, y = average_return)) +
    geom_point() +
    geom_smooth(method = 'lm', se = F) +
    labs(x = "Beta", y = "Average Returns (%)") +
    ggtitle("Beta vs Mean Portfolio Return") +
    cowplot::theme_cowplot()
ggsave(here::here("HW4", "plots", "portfolio.png"))

# Calculate time average
time_average_2 = portfolio_df |> summarise(mean = mean(mkt))

#### f)

# Extract residuals for each portfolio
# I'm running the regression again because its easier than shoehorning extracting the residuals into th

resids = map(unique(portfolio_df$portfolio), function(x){
  portfolio_df |>
    filter(portfolio == {x}) %>%
    lm(excess_returns ~ mkt,.) |> summary() |> resid() #
  }) |>
  unlist() |>
  matrix(nrow = 20) # We have 20 regressions so each row is the residuals from one regression

# Calculate GRS test
GRS = ((120 - 20 - 1)/20)*(1 + (mean(portfolio_df$mkt)/var(portfolio_df$mkt))^2)^(-1)* t(as.matrix(uniq
```

```r
p_value = 1 - pf(GRS, 20, 120 - 20 - 1)


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

# Merge industry and ff daily data

daily_df =
  left_join(ff_daily_df,
            daily_industry_df) |>
  mutate(date = ymd(date),
         date = paste(year(date), month(date), sep = "-")) |>
  drop_na(RF)

# Calculate monthly betas by industry

industry_monthy_beta =
  daily_df |>
  # This is so we can group by industry and date simultaneously
  pivot_longer(cols = -c(date, `Mkt-RF`, RF)) |>

  # Group by each industry in each month
  group_by(date, name) |>

  # Calculate monthly beta
  summarise(monthly_beta = lm(value ~ `Mkt-RF`) |> coef() |> nth(2),
            # This gets the standard error of the beta
            beta_var = lm(value ~ `Mkt-RF`) |> broom::tidy() |> select(std.error) |> slice(2) |> unlist

            # In question 4 we need the alphas so I calculate them here
            monthly_alpha = lm(value ~ `Mkt-RF`) |> coef() |> nth(1))


# Calculate time series standard deviations

industry_se_table =
  industry_monthy_beta |>
    group_by(name) |>

    # Adding up variance across months plus the mean error from calculating the betas themselves. Sqrt
    summarise(true_se = sqrt(var(monthly_beta) + mean(beta_var)^2)) |>
    rename("Industry" = name,
           "Time Series SD" = true_se)


##### Question 4 ############

# Calculate time series alpha and beta for each industry across entire time series

industry_betas =
  daily_df |>
    pivot_longer(cols = -c(date, `Mkt-RF`, RF)) |>
    group_by(name) |>
```

```
    summarise(alpha = lm(value ~ `Mkt-RF`) |> coef() |> nth(1),
              beta = lm(value ~ `Mkt-RF`) |> coef() |> nth(2),
              alpha_se = lm(value ~ `Mkt-RF`) |> broom::tidy() |> select(std.error) |> slice(1) |> unlist
  rename("Industry" = name,
         "Alpha" = alpha,
         "Beta" = beta,
         "SE" = alpha_se)



# Calculate conditional alphas. Using the Lewellen Nagel methodology this involves calculating the alph
industry_conditional_alphas =
  industry_monthy_beta |>
    group_by(name) |>
    summarise(conditional_alpha = mean(monthly_alpha),
              se = monthly_alpha |> var() |> sqrt()) |>
  rename("Industry" = name,
         "Conditional Alpha" = conditional_alpha,
         "Conditional SE" = se)

# Calculate conditional mean of alphas
conditional_alpha_mean = industry_conditional_alphas |>
  select(`Conditional Alpha`) |>
  unlist() |>
  mean()
```