# Appendix

```
library(tidyverse)
library(visdat)
library(kableExtra)
library(pheatmap)
library(broom)
library(ggrepel)
```

```
# Load the data
stock <- read_csv("Data/SampleA.csv")
market <- read_csv("Data/Market.csv")
```

```
# Transform to Date format
stock <- stock |>
  mutate(
    year = str_extract(Date, "\\d{4}"),
    month = str_extract(Date, "(?<=M)\\d+"),
    month = str_pad(month, width = 2, pad = "0"),
    Date = paste0(year, "-", month),
    Date = as.Date(paste0(Date, "-01"))
  ) |>  select(-year, -month)
market <- market |>
  mutate(
    year = str_extract(Date, "\\d{4}"),
    month = str_extract(Date, "(?<=M)\\d+"),
    month = str_pad(month, width = 2, pad = "0"),
    Date = paste0(year, "-", month),
    Date = as.Date(paste0(Date, "-01"))
  ) |>  select(-year, -month)
```

```
##| tbl-cap: "Industry codes"
industry_codes <- tibble(Code = c("B", "C", "D", "E", "F", "G", "H", "I"),
  Industry = c("Mining","Construction","Manufacturing",
               "Transportation and Public Utilities","Wholesale Trade",
               "Retail Trade","Finance, Insurance and Real Estate","Services"))
```

```
##| label: tbl-ind_sum
##| tbl-cap: "Industry summary statistics"
# Extract industry from stock names
long <- stock |>
  pivot_longer(-Date, names_to = "stock", values_to = "ret")
stock_ind <- long |>
  mutate(ind = str_extract(stock, "^[A-Za-z]+"))
industry_prop <- stock_ind |>
  distinct(stock, ind) |>
  count(ind, name = "n_stocks") |>
  mutate(prop = n_stocks / sum(n_stocks)) |>
  arrange(desc(prop))
ind_summary <- industry_codes |>
  left_join(industry_prop, by = c("Code" = "ind")) |>
  select(Code, Industry, n_stocks, prop)
kable(ind_summary)
```

```
#Check duplicate value
stock %>%  filter(duplicated(.))
```

```
# Check for missing months
seq_months <- tibble(Date = seq(min(stock$Date, na.rm = TRUE),
                                max(stock$Date, na.rm = TRUE),
                                by = "month"))
missing_months <- seq_months |>
  anti_join(stock |> distinct(Date), by = "Date")
```

```
# Compute z-scores for each stock
stock_z <- stock |>
  mutate(across(-Date, ~ scale(.)[, 1], .names = "{.col}_z"))
stock_z_long <- stock_z |>
  pivot_longer(
    cols = ends_with("_z"),
    names_to = "Stock_z",
    values_to = "Z_Score"
  ) |>  mutate(Stock = str_remove(Stock_z, "_z") )
# Filter rows where abs(z-score) > 3 (3-sigma outliers)
```

```
outlier_df <- stock_z_long |>
  filter(abs(Z_Score) > 3) |>
  select(Date, Stock, Z_Score) |>
  arrange(desc(abs(Z_Score)))
# join with raw returns
outlier_df <- outlier_df |>
  left_join(stock |>
              pivot_longer(-Date, names_to = "Stock", values_to = "Return"),
            by = c("Date", "Stock"))
```

```
##| label: tbl-outlier_table
##| tbl-cap: "Top 6 outliers detected using Z-scores"
top_outlier <- head(outlier_df,6)
kable(top_outlier)
```

```
# Per-stock summary stats
summ <- long |>
  group_by(stock) |>
  summarise(
    mean = mean(ret, na.rm = TRUE),
    sd = sd(ret, na.rm = TRUE),
    min = min(ret, na.rm = TRUE),
    p25 = quantile(ret, 0.25, na.rm = TRUE),
    median = median(ret, na.rm = TRUE),
    p75 = quantile(ret, 0.75, na.rm = TRUE),
    max = max(ret, na.rm = TRUE),
    .groups = "drop"
  ) |> arrange(desc(sd))
```

```
##| label: tbl-statistics-table
##| tbl-cap: "Summary statistics for stock returns"
sum_table <- summ |> slice_head(n = 9)
kable(sum_table)
```

```
# Boxplot of top volatile stocks
top9 <- summ |> slice_max(sd, n = pmin(9, nrow(summ))) |> pull(stock)
```

```
##| label: fig-boxplot
##| fig-cap: "Return distributions of top 9 volatile stocks"
stock |>
  select(Date, all_of(top9)) |>
```

```r
  pivot_longer(-Date, names_to = "stock", values_to = "ret") |>
  ggplot(aes(stock, ret)) +
  geom_boxplot(outlier.alpha = 0.5) +
  coord_flip() +
  labs(title = "Return distributions - top9 volatile stock",
       x = NULL,
       y = "Monthly return") +
  theme_minimal()
```

## Principal Component Analysis (PCA)

```r
##| label: fig-scree
##| fig-cap: "Scree plot of PCA"
# Prepare data for PCA
stock_pca <- stock |>
  select(-Date) |>
  as.matrix()
stock_pca_std <- scale(stock_pca)
# PCA
pca <- prcomp(stock_pca_std,center = FALSE,scale. = FALSE)
screeplot(pca, type = "lines")
```

```r
##| label: tbl-pca-summary
##| tbl-cap: "Variance explained by PC1-PC3"
var_explained <- pca$sdev^2
prop_var <- var_explained / sum(var_explained)
cum_var <- cumsum(prop_var)
# Combine into a table
pc_summary <- data.frame(
  PC = paste0("PC", 1:length(var_explained)),
  Variance = round(var_explained, 4),
  Proportion = round(prop_var, 4),
  Cumulative = round(cum_var, 4))
kable(pc_summary[1:3, ])
```

```r
##| label: tbl-cor
##| tbl-cap: "Correlation between industry mean returns and market return"
# Industry movement
ind_move <- stock_ind |>
```

```
  group_by(Date, ind) |>
  summarise(mean_ret = mean(ret))
ind_wide <- ind_move |>
  left_join(market) |>
  select(Date, ind, mean_ret, MarketReturn) |>
  pivot_wider(names_from = ind, values_from = mean_ret)
ind_wide_num <- ind_wide |>
  mutate(across(where(is.character), as.numeric)) |>
  as.data.frame() |>
  select(-Date)
# Compute correlation
cor_mat <- cor(ind_wide_num, use = "pairwise.complete.obs")
# Turn into table
cor_tbl <- as.data.frame(round(cor_mat, 3))
knitr::kable(cor_tbl)
```

```
##| label: tbl-loadings
##| tbl-cap: "Industry Loadings on PC1, PC2, and PC3"
load <- as.data.frame(pca$rotation[,1:3]) |>
  rownames_to_column("stock")
load$industry <- substr(load$stock, 1, 1)
industry_centroids <- load |>
  group_by(industry) |>
  summarise(PC1 = mean(PC1),PC2 = mean(PC2),PC3 = mean(PC3))
kable(industry_centroids)
```

```
##| label: fig-pca_cor
##| fig-cap: "Relationships of PC1-PC3 with market return"
scores <- as.data.frame(pca$x[, 1:3]) |>
  cbind(Date = stock$Date, Market = market$MarketReturn)
# Standardize
scores_std <- scores |>
  mutate(across(-Date, ~ as.numeric(scale(.))))
scores_long <- scores_std |>
  pivot_longer(cols = c(PC1, PC2, PC3),names_to = "PC",values_to = "Score")
pca_cor_labels <- scores_long |>
  group_by(PC) |>
  summarize(cor = cor(Market, Score, use = "complete.obs"),
    .groups = "drop") |>
  mutate(label = paste0("cor = ", sprintf("%.2f", cor)),
    x = Inf, y = Inf)
ggplot(scores_long, aes(x = Market, y = Score)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm") +
  geom_text(data = pca_cor_labels, aes(x = x, y = y, label = label),
            hjust = 1.1, vjust = 1.5, inherit.aes = FALSE) +
  facet_wrap(~PC, ncol = 1, scales = "free_y") +
  theme_minimal() +
  labs(title = "Relationship between PC1-PC3 and market return",
       x = "Market Return", y = "PC Score")
```

```
##| label: tbl-top5-loadings
##| tbl-cap: "Top 5 Stocks by Absolute PC1 Loading"
# Extract PCA loadings for PC1
pc_load_tbl <- as.data.frame(pca$rotation[,1:3]) |>
  rownames_to_column("stock") |>
  mutate(industry = substr(stock, 1, 1),
    abs_PC1 = abs(PC1)   # absolute loadings for ranking
  ) |> arrange(desc(abs_PC1))
# Top 5
top5_pc1 <- pc_load_tbl |>
  slice_head(n = 5)
kable(top5_pc1)
```

## Factor Modelling

```
##| label: fig-scree-eigen
##| fig-cap: "Scree plot of eigenvalues for factor analysis"
### Scree plot of eigenvalues
stock_only <- stock |> select(-Date)
X <- as.matrix(stock_only)
eig_vals <- eigen(cor(X))$values
eig_df <- data.frame(PC = 1:length(eig_vals),Eigenvalue = eig_vals)
ggplot(eig_df, aes(x = PC, y = Eigenvalue)) +
  geom_line() +
  geom_point() +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  theme_minimal() +
  labs(title = "Scree Plot of Eigenvalues",
    x = "Principal Component",y = "Eigenvalue")
```

```r
# Prepare matrix
stock_only <- stock |> select(-Date)
X <- as.matrix(stock_only)
# Estimate 3-factor model with Promax rotation
fa <- factanal(X, factors = 3, rotation = "promax",
               scores = "Bartlett", lower = 0.05)
print(fa, digits = 3, cutoff = 0.3)
```

```r
##| label: tbl-mat
##| tbl-cap: "Numbers of Each Industry on Factors 1-3"
# Tidy loadings + add industry code from ticker's first letter
load_df <- as_tibble(unclass(fa$loadings), rownames = "stock") |>
  rename(Factor1 = 2, Factor2 = 3, Factor3 = 4) |>
  mutate(ind = substr(stock, 1, 1))
# Industry proportions in the sample (unique stocks per industry)
ind_prop <- load_df |>
  distinct(stock, ind) |>
  count(ind, name = "n_stocks") |>
  mutate(prop = n_stocks / sum(n_stocks))
fac_ind_mat <- load_df |>
  pivot_longer(starts_with("Factor"),names_to = "Factor",
               values_to = "Loading")|> filter(abs(Loading) >= 0.4) |>
  count(ind, Factor, name = "n_strong") |>
  pivot_wider(names_from = Factor, values_from = n_strong, values_fill = 0) |>
  left_join(ind_prop, by = "ind") |>
  arrange(desc(prop))
kable(fac_ind_mat)
```

```r
##| label: tbl-fa-cor
##| tbl-cap: "Factor Correlation Matrix"
# Construct the symmetric correlation matrix
Phi_mat <- matrix(c(
  1.00, -0.44,  0.65,
 -0.44,  1.00, -0.24,
  0.65, -0.24,  1.00
), nrow = 3, byrow = TRUE)
dimnames(Phi_mat) <- list(
  c("Factor1","Factor2","Factor3"),
  c("Factor1","Factor2","Factor3"))
# Transform to tibble
Phi_df <- as_tibble(Phi_mat, rownames = "Factor")
kable(Phi_df)
```

```
##| label: tbl-fa-stock
##| tbl-cap: "Top 10 Most Systematic Stocks (Lowest Uniqueness)"
fa_tbl <- as_tibble(unclass(fa$loadings), rownames = "stock")
uniq_tbl <- tibble(
  stock = names(fa$uniquenesses),
  uniqueness = as.numeric(fa$uniquenesses))
fa_stock <- fa_tbl |>
  left_join(uniq_tbl, by = "stock") |>
  mutate(industry = substr(stock, 1, 1)) |>
  # nice formatting for printing only; keeps numeric math safe
  mutate(across(where(is.numeric), ~ round(.x, 3))) |>
  relocate(industry, stock)
fa_stock_ranked <- fa_stock |>
  arrange(uniqueness)
knitr::kable(
  fa_stock_ranked |> select(industry, stock, dplyr::starts_with("Factor"),
                            uniqueness) |> slice_head(n = 10))
```

```
##| label: fig-fa-stock
##| fig-cap: "Uniqueness vs strongest factor loading"
fa_stock_plot <- fa_stock |>
  rowwise() |>
  mutate(max_loading = max(abs(c_across(dplyr::starts_with("Factor"))))) |>
  ungroup()
ggplot(fa_stock_plot, aes(x = uniqueness,y = max_loading,color = industry,
                          label = stock)) +
  geom_point() +
  geom_text_repel(size = 2, max.overlaps = 100) +
  labs(title ="Uniqueness (idiosyncratic) vs strongest factor loading (systematic)",
    x = "Uniqueness (lower = more systematic)",y = "Max |Factor Loading|") +
  theme_minimal()
```

```
##| label: fig-fa-loadings
##| fig-cap: "Factor Loadings Plot"
fa_df <- tidy(fa)
ggplot(fa_df, aes(x = fl1, y = fl2, label = variable)) +
  geom_segment(aes(xend = fl1, yend = fl2, x = 0, y = 0),
               arrow = arrow(length = unit(2, "mm")), linewidth = 0.2) +
  geom_point(size = 0.8) +
  geom_text_repel(color='red',size = 2, max.overlaps = 50) +
  coord_equal() +
  labs(title = "Factor Loadings Plot (F1 vs F2)",
       x = "Factor 1 Loadings", y = "Factor 2 Loadings")
```

```
##| label: tbl-rank
##| tbl-cap: "Top 5 Most Systematic Stocks with FA and PCA Loadings"
# Factor Analysis loadings + uniqueness
fa_tbl <- as_tibble(unclass(fa$loadings), rownames = "stock") |>
  rename(F1 = Factor1, F2 = Factor2, F3 = Factor3)
uniq_tbl <- tibble(stock = names(fa$uniquenesses),
                   uniqueness = as.numeric(fa$uniquenesses))
fa_tbl <- fa_tbl |>
  left_join(uniq_tbl, by = "stock")
# PCA loadings (rotation)
pc_tbl <- as.data.frame(pca$rotation[,1:3]) |>
  rownames_to_column("stock") |>
  rename(PC1 = PC1, PC2 = PC2, PC3 = PC3)
# Combine both
rank_tbl <- fa_tbl |>
  left_join(pc_tbl, by = "stock") |>
  mutate(industry = substr(stock, 1, 1)) |>
  arrange(uniqueness)
# Preview top 5
rank_tbl_pretty <- rank_tbl |>
  mutate(across(c(F1,F2,F3,PC1,PC2,PC3,uniqueness), ~round(.x,3))) |>
  select(industry, stock, uniqueness, F1, F2, F3, PC1, PC2, PC3)
knitr::kable(rank_tbl_pretty |> slice_head(n = 5))
```

```
##| label: fig-fa-top5
##| fig-cap: "Uniqueness vs strongest factor loading, highlighting top 5 picks"
# Get absolute strongest factor loading for each stock
df <- rank_tbl %>%
  rowwise() %>%
  mutate(max_loading = max(abs(c(F1, F2, F3)))) %>%
  ungroup()
# Mark top 5 recommended stocks
top5 <- c("H90000", "H88215", "H75157", "H89050", "H77466")
df <- df %>%
  mutate(top_pick = ifelse(stock %in% top5, "Yes", "No"))
# Plot
ggplot(df,aes(x=uniqueness,y = max_loading,color=top_pick,label = stock)) +
  geom_point(size = 3) +
  geom_text_repel(aes(label = ifelse(top_pick == "Yes", stock, "")),
                  size = 3, box.padding = 0.25, max.overlaps = Inf) +
```

```r
  scale_color_manual(values = c("Yes" = "red", "No" = "grey")) +
  theme_minimal() +
  labs(title = "Uniqueness vs Strongest Factor Loading",
       subtitle = "Top 5 market-like stocks highlighted in red",
       x = "Uniqueness (idiosyncratic risk)",
       y = "Strongest Factor Loading (systematic risk)",
       color = "Top 5 Pick")
```

## Other

```r
## Check missing values and data types
vis_miss(stock) +
  theme(axis.text.x = element_blank(),
    axis.ticks.x = element_blank())
vis_dat(stock) +
  theme(axis.text.x = element_blank(),
    axis.ticks.x = element_blank())
```

```r
##| fig-cap: "Stock Price Movement by industry"
# Time series
ggplot(ind_move, aes(x = Date)) +
  geom_line(aes(y = mean_ret), color = "black") +
  facet_wrap(~ind) +
  theme_minimal() +
  labs(title = "Industry mean returns over time",
    x = "Date",y = "Return") +
  theme(legend.position = "bottom")
```