Predicting Financial Asset Returns and Portfolio Allocation

Author: Ali Yasin
Supervisor: Dr Maria Kalli
Module Code: 6CCM345A

2024/25

Abstract

This project evaluates the Frequentist Regression approach (OLS, *ridge regression*, *lasso* and *elastic net*) and Conditional Volatility Modelling approach (ARCH/GARCH) for forecasting the monthly S&P 500 returns using 14 macro-financial indicators. This work will culminate in the production of a model which combines the two approaches in a hybrid model which outperforms every other model in both in-sample metrics and in the context of forecasting for portfolio allocation.

Contents

0 Introduction

Forecasting financial asset returns is crucial to portfolio construction and asset allocation. Past studies, such as; *A Comprehensive Look at the Empirical Performance of Equity Premium Prediction by Ivo Welch and Amit Goyal* [2], have attempted the regression approach, using ordinary least squares on a variety of financial ratios and macro economic variables. More modern methods such as shrinkage methods (*ridge regression*, *lasso* and *elastic net*) and modelling conditional volatility show promising potential to build upon this study.

This project aims to fill this gap by testing and comparing the predictive performance of each model using monthly S&P 500 returns data from January 1927 until December 2023 combined with the same 14 economically established financial ratios and macroeconomic indicators.

To start with, in Chapter 1 we introduce the concept of the return of an asset, after which we construct the usable data set and introduce the key variables that will be used. Then, Chapters 2 and 3 focus on covering the theoretical foundations to the frequentist regression approach, discussing ordinary least squares (OLS) and shrinkage methods, after which we apply each method to the usable data set and analyse each model to look for room for improvements. Penultimately, Chapter 4 introduces the theory behind an alternative approach, which relies on modelling the conditional volatility, and compares the models produced to the frequentist approach. This comparison leads to the end of this chapter and the production of a hybrid model which combines both the regression and modelling volatility approaches. Finally, we introduce the theoretical foundations for mean-variance portfolio allocation, after which we apply each model to the data and discuss our findings.

1 Introduction to Returns and Data Construction

This chapter follows the results and steps as found in this book and research paper: Analysis of Financial Time Series by Ruey S. Tsay [1] *and A Comprehensive Look at the Empirical Performance of Equity Premium Prediction by Ivo Welch and Amit Goyal* [2].

1.1 Asset Returns

This subsection follows the results and steps as found in this book: Analysis of Financial Time Series by Ruey S. Tsay [1].

Through this subsection, we will start by defining the concept of returns and different types of returns.

Let us start by understanding why returns are a useful metric; there are two main reasons as to why we use asset returns:

(i) The return of an asset is a scale-free and complete summary of the investment opportunity.

(ii) Return series have more desirable statistical properties that of price series, making them easier to use.

We will now define the notion of an asset return. Let P_t denote the price of an asset at time t. Also assume, for now, that the asset pays no dividends.

Definition 1 (One-Period Simple Return)**.** If we hold an asset for one period, this would result in the following:

(i) Simple Gross Return:

$$1 + R_t = \frac{P_t}{P_{t-1}} \tag{1.1}$$

(ii) Simple Net Return:

$$R_t = \frac{P_t}{P_{t-1}} - 1 = \frac{P_t - P_{t-1}}{P_{t-1}} \tag{1.2}$$

Definition 2 (Multi-Period Simple Return)**.** If we hold an asset for k-periods, then this results in the following:

(i) k-period Simple Gross Return:

$$1 + R_t[k] = \frac{P_t}{P_{t-k}} = \frac{P_t}{P_{t-1}} \cdot \frac{P_{t-1}}{P_{t-2}} \cdot \ldots \cdot \frac{P_{t-k+1}}{P_{t-k}} \tag{1.3}$$

$$= (1 + R_t)(1 + R_{t-1}) \ldots (1 + R_{t-k+1}) = \prod_{i=0}^{k-1} (1 + R_{t-i}) \tag{1.4}$$

(ii) k-period Simple Net Return:

$$R_t[k] = \frac{P_t}{P_{t-k}} - 1 = \frac{P_t - P_{t-k}}{P_{t-k}} \tag{1.5}$$

Remarks:

(i) The k-period simple gross return is just the product of the k one-period simple gross returns.

(ii) In general, we assume that one-period represents one year.

Definition 3 (Continuous Compounding)**.** The asset value A_n of continuous compounding is:

$$A_n = A_0 \cdot e^{r \cdot n} \tag{1.6}$$

,where A_0 is the initial capital, n is the number of years and r is the interest rate per year.

Definition 4 (Continuously Compounded One-Period Return)**.** The natural logarithm of the one-period simple gross return of an asset is called the continuously compounded return:

$$r_t = ln(1 + R_t) = ln\left(\frac{P_t}{P_{t-1}}\right) = p_t - p_{t-1} \tag{1.7}$$

,where $p_t = ln(P_t)$.

Definition 5 (Continuously Compounded Multiperiod Return)**.** The natural logarithm of the k-period simple gross return of an asset is called the continuously compounded return:

$$r_t[k] = ln(1 + R_t[k]) = \sum_{j=0}^{k} ln(1 + R_{t-j}) = \sum_{j=0}^{k} r_{t-j} \tag{1.8}$$

,where $p_t = ln(P_t)$. It is clear that the continuously compounded multiperiod return is just the sum of the continuously compounded one-period returns.

Definition 6 (Portfolio Return)**.** Let p be a portfolio that contains N assets and places weight w_i on asset i. Then the simple return of p at time t is:

$$R_{p,t} = \sum_{i=1}^{N} w_i R_{it} \tag{1.9}$$

where R_{it} is the simple return of asset i.

Definition 7 (Dividends)**.** Let D_t be the dividend payment of an asset, with P_t denoting the price of the asset at the end of period t. Then the :

(i) Simple Net Return:

$$R_t = \frac{P_t + D_t}{P_{t-1}} - 1 \tag{1.10}$$

(ii) Continuously Compounded Return:

$$r_t = ln(P_t + D_t) - ln(P_{t-1}) \tag{1.11}$$

Definition 8 (Excess Return)**.** The excess return of an asset at time t is simply the difference between the asset's return and the return on a reference asset.

(i) Simple Excess Return:

$$Z_t = R_t - R_{0t} \tag{1.12}$$

where R_{0t} is the simple return of the reference asset.

(ii) log Excess Return:

$$z_t = r_t - r_{0t} \tag{1.13}$$

where r_{0t} is the log return of the reference asset.

Remark: The reference asset is generally taken to be a risk-free asset, such as U.S. Treasury bill return or U.S. Treasury Bonds.

We can also think of the excess return as the payoff of an arbitrage portfolio that goes long in an and asset and short in a reference asset with no net initial investment.

We have now defined and understood the notion of an asset return and its different types.

1.2 Data Construction

This subsection follows the results and steps as found in this research paper: A Comprehensive Look at the Empirical Performance of Equity Premium Prediction by Ivo Welch and Amit Goyal [2].

The raw data set I will be using in this project has been provided by Prof. Amit Goyal.

Through this section I will introduce the variables in the raw data set and then show how I will clean and construct the variables my working data set.

1.2.1 Introducing Data

This raw data set contains data for every month from the start of 1871 to the end of 2023 on 16 variables, which I will introduce below. I will then use these variables to construct key financial ratios and macroeconomic indicators, which I will be using as the variables in my final data set.

Our response variable in this case will be Stock Returns of the S&P 500 index.

Variable	**Identifier**	**Description**
Stock Returns	`CRSP_SPvw`	Continuously compounded returns on the S&P 500 index, excluding dividends.
Index	`Index`	This is the Monthly closing price of the S&P 500 index.
Dividends	`D12`	The sum of the dividends paid out to shareholders by the companies in the S&P 500 index over the last 12 months.
Earnings	`E12`	The sum of the profits of the S&P 500 index companies over the last 12 months.
Stock Variance	`svar`	The sum of squared daily returns on the S&P 500 index.
Cross Sectional Premium	`csp`	A measure of the relative valuation of high and low-beta stocks.
Book to Market Ratio	`b.m`	The ratio of the book value to the market value. Where the book value is the total value of a company's assets after subtracting it's liabilities.[1]
Net Equity Expansion	`ntis`	This is a measure of corporate issuing activity. It is the ratio of the 12-month sum of net issues to the total end of year market capitalisation of NYSE stocks.
Risk-Free Rate	`tbl`	The rate of returns of the risk-free asset, in this case being the U.S. Treasury-bill rate.
Long Term Yield:	`lty`	The long term government bond yield.
Long Term Rate of Returns	`ltr`	The total return on long term government bonds.
Corporate Bond Returns	`corp`	The returns of corporate bonds (bonds which are issued by companies).
Corporate Bond Yields	`AAA,BAA`	This is the yields of corporate bonds. In this case, this is split up into two variables, one referring to the yields of AAA-rated bonds and the other referring to the yields of BAA bonds.
Inflation	`infl`	The monthly inflation rate of the U.S. economy, which is the rate of change of the general price level of goods and services.

1.2.2 Construction of the Usable Data Set

I will now describe the issues with this data set and how I will clean and construct the usable data set containing key financial ratios and macroeconomic indicators.

The main issue with this data set is that there are missing values for the earlier dates for a majority of the variables. To rectify this issue I choose to start my final data set from January 1927 instead of January 1871, as this is the earliest year that has complete data until end of 2023.

This issue is especially extreme for for the *Cross Sectional Premium (csp)* where there is data missing from the beginning of 1871 to April 1936, and from January 2003 to December 2023.

Since there are is around 85 years of data missing, in the interest of predictive performance for more modern data, I chose to omit this variable completely from my final data set.

We will now introduce all of the variables in the final data set.

There are 8 variables which are inherited from the raw data set:

1. **Stock Returns:** *(CRSP_SPvw)*
2. **Cross Sectional Premium:** *(csp)*
3. **Book to Market Ratio:** *(b.m)*
4. **Net Equity Expansion:** *(ntis)*
5. **Risk-free Rate:** *(tbl)*
6. **Long Term Yield:** *(lty)*
7. **Long Term Rate of Returns:** *(ltr)*
8. **Inflation:** *(infl)*

We also compute seven key financial ratios and macroeconomic indicators as additional variables.

Variable	Identifier	Description
Dividend-Price Ratio	`d.p`	The difference between the log of dividends and the log of the prices.[2]
Dividend-Yield	`d.y`	The difference between the log of dividends and log of lagged prices.[2]
Earning-Price Ratio	`e.p`	The difference between the log of the earnings and the log of the prices.[3]
Dividend-Payout Ratio	`d.e`	The difference between the log of the dividends and the log of the earnings.[3]
The Term Spread	`tms`	The difference between the long term yield on government bonds and the T-bill.[4]
The Default Yield Spread	`dfy`	The difference between AAA and BAA-rated corporate bond yields.[4]
The Default Return Spread	`dfr`	The difference between long-term corporate bond and long-term government bond returns.[4]

In summary, we have constructed a usable data set containing data for every month from the start of 1927 until the end of 2023 on the following 15 variables:

[1] For Mar-Dec: $b/m = Book_{t-1}^{Dec}/P_t$; for Jan-Feb: $b/m = Book_{t-2}^{Dec}/P_t$
[2] $d/p = log(D12) - log(Index)$; $d/y = log(D12) - log(lag(Index))$
[3] $e/p = log(E12) - log(Index)$; $d/e = log(D12) - log(E12)$
[4] $tms = lty - tbl$; $dfy = BAA - AAA$; $dfr = corp - ltr$

1. Stock Returns *(CRSP_SPvw)*
2. Dividend-Price Ratio *(d/p)*
3. Dividend-Yield *(d/y)*
4. Earnings Price Ratio *(e/p)*
5. Dividend Payout Ratio *(d/e)*
6. Stock Volatility *(svar)*
7. Book Market Ratio *(b/m)*
8. Net Issuing Activity *(ntis)*
9. Treasury-Bill Rate: *(tbl)*
10. Long-term Yield *(lty)*
11. Long-term Bond Rate of Return *(ltr)*
12. Term-Spread *(tms)*
13. The Default Yield *(dfy)*
14. The Default Rate of Return *(dfr)*
15. Inflation Rate *(infl)*

Remark: Everything relating to the importing, cleaning, and construction of the final data set has been done in R-script and the code can be found in Appendix A.

1.3 Empirical Properties of Returns

This subsection follows closely the results and steps as found in this book: Analysis of Financial Time Series by Ruey S. Tsay [1].

Through this chapter, we look to describe the empirical properties of our asset returns data.

1.3.1 Moments of Data: Describing Statistical Distributions

Consider a continuous random variable X which follows a probability density function $f_X(x)$.

Definition 9 (k^{th} Moment)**.** The k^{th} moment of a continuous random variable X is defined as:

$$m'_k = \mathbb{E}[X^k] = \int_{-\infty}^{\infty} x^k f_X(x) dx \tag{1.14}$$

Definition 10 (k^{th} Central Moment)**.** The k^{th} central moment of a continuous random variable X, which has mean μ_k, is defined as:

$$m_k = \mathbb{E}[(X - \mu_k)^k] = \int_{-\infty}^{\infty} (x - \mu_k)^k f_X(x) dx \tag{1.15}$$

Remark:

(i) Note that the 2^{nd} central moment is simply the variance. i.e. $m_2 = \text{var}(X) = \sigma^2$.

(ii) The third central moment measures the symmetry of the with respect to its mean.

(iii) The fourth central moment measures the behaviour of the tails of X.

Definition 11 (Skewness)**.** Skewness is defined as the normalised third central moment of X:

$$S(x) = \mathbb{E}\left[\frac{(X - \mu_x)^3}{\sigma_x^3}\right] \tag{1.16}$$

This summarises the extent of asymmetry.

Definition 12 (Kurtosis)**.** Kurtosis is defined as the normalised fourth central moment of X:

$$K(x) = \mathbb{E}\left[\frac{(X - \mu_x)^4}{\sigma_x^4}\right] \tag{1.17}$$

This summarises the extent of tail thickness.

Definition 13 (Excess Kurtosis)**.** Excess kurtosis is defined as Kurtosis of X minus 3 and is denoted γ_2:

$$\gamma_2 = K(x) - 3 \tag{1.18}$$

Remark: The reasoning behind why the number 3 is subtracted is simply that the kurtosis of any Gaussian distribution is 3, that is $K(X) = 3$, where $X \sim \mathcal{N}(\mu, \sigma^2)$.

What does the excess kurtosis tell us about they extent of tail thickness:

(i) Positive excess kurtosis: $K(x) > 0$
The distribution is said to have heavy tails, implying the distribution puts more mass on the tails of its support as compared to the normal distribution. This type of distribution is said to be leptokurtic.

(ii) Negative excess kurtosis: $K(x) < 0$
The distribution is said to have short tails, implying the distribution puts less mass on the tails of its support as compared to the normal distribution. This type of distribution is said to be platykurtic.

We can also estimate the skewness and kurtosis from a given sample. Let $\boldsymbol{x} = \{x_1, x_2, \ldots, x_N\}$ be a random sample of the random variable $\boldsymbol{X}$, then:

(i) Sample Mean:

$$\hat{\mu}_x = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1.19}$$

(ii) Sample Variance:

$$\hat{\sigma}_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \hat{\mu}_x)^2 \tag{1.20}$$

(ii) Sample Skewness:

$$\hat{S}(x) = \frac{1}{(n-1)\hat{\sigma}_x^3}\sum_{i=1}^{n}(x_i - \hat{\mu}_x)^3 \tag{1.21}$$

(ii) Sample Kurtosis:

$$\hat{K}(x) = \frac{1}{(n-1)\hat{\sigma}_x^4}\sum_{i=1}^{n}(x_i - \hat{\mu}_x)^4 \tag{1.22}$$

1.3.2 Empirical Properties of Returns

We will investigate the empirical properties of the S&P 500 returns in our data.

First, let us consider the empirical density plot of the returns and compare it with the normal distribution.

The plot below shows the empirical density plot of the stock returns *(CRSP_SPvw)* of the S&P 500 index from 1927 until 2023. The dotted line is the normal probability density function evaluated at the sample mean and sample variance of the returns.

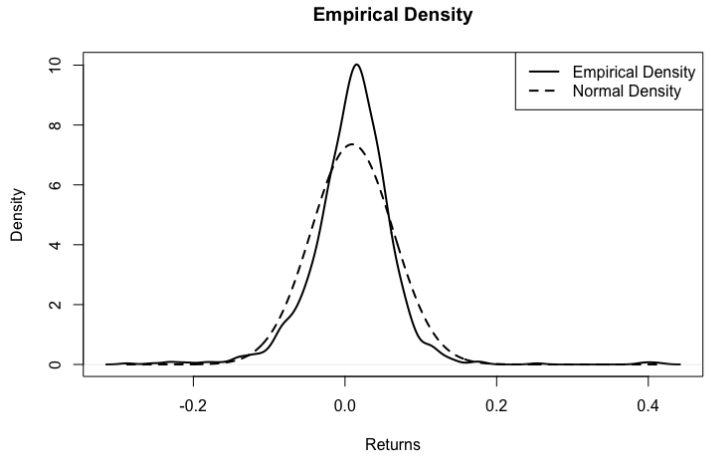

Figure 1: The solid line represents the empirical density function of the returns and the dotted line represents the Gaussian P.D.F evaluated at the sample mean and sample variance of the returns.

Remark: The above plot was generated using R-script using the code found in Appendix A.

The main observation we can make from this plot it that the empirical density function has a higher peak about it's mean but, it has heavier tails than that of the corresponding normal distribution.

In simpler terms, the empirical density is taller and thinner, but it has a wider support than that of the corresponding normal distribution.

Next, we will further justify these observations by calculating the summary statistics of our returns data.

By using the `basicStats()` function within the `fBasics` package in R, we are able to produce a variety of summary statistics of our asset returns data:

Sample Statistic	**Estimator**	**Result**
Mean	$\hat{\mu}_{r_t}$	0.009518
Variance	$\hat{\sigma}_{r_t}$	0.002942
Skewness	$\hat{S}(r_t)$	0.326531
Kurtosis	$\hat{K}(r_t)$	9.2318781

Table 1: Sample Summary Statistics of the S&P 500 Stock Returns

The observations we can make are as follows:

(i) The sample mean is very close to zero, as expected.

(ii) The sample variance is very small, indicating moderate volatility of returns.

(iii) The sample skewness is positive, which indicates positive skewness. i.e. the distribution is more concentrated on the right tail.

(iv) The excess sample kurtosis is positive, indicating the distribution is heavy tailed (leptokurtic).

These observations align with the observations made in the empirical density plot, and help give us a better understanding of the statistical nature of our returns data, something which will be key in later chapters.

2 Theoretical Foundations: Frequentist Regression

This chapter follows the results and steps as found in the book and research paper: An Introduction to Statistical Learning by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani [3] *and Regularization and variable selection via the elastic net by Hui Zou and Trevor Hastie* [4].

This chapter covers the frequentist approach to linear regression, first reviewing the Ordinary Least Squares procedure and then moving to cover shrinkage methods such as *Ridge regression*, *Lasso* and *Elastic net*.

2.1 Ordinary Least Squares (OLS)

We first present the method of Ordinary Least Squares, specifically applied to a multivariate linear model (MLR). We will then derive the estimator for our coefficients and some results.

Suppose we have n observations and p predictors. Then the multiple linear regression (MLR) takes the form:

$$Y_i = \beta_0 + \beta_1 X_{1,i} + \beta_2 X_{2,i} + \cdots + \beta_p X_{p,i} + \epsilon_i \quad , i = 1, \cdots, n \tag{2.1}$$

where X_j represents the jth predictor and β_j represents the coefficient of this predictor in our multiple linear regression model.

We can alternatively write this using the following notation:

$$Y_i = \mu_i + \epsilon_i \quad where \quad \mu_i = \mathbb{E}(Y_i) = \beta_0 + \beta_1 X_{1,i} + \beta_2 X_{2,i} + \cdots + \beta_p X_{p,i} \tag{2.2}$$

$$\text{var}(Y_i) = \sigma^2, \quad i = 1, \ldots, n \quad and \quad \text{cov}(Y_i, Y_j) = 0, \quad i \neq j \tag{2.3}$$

For the sake of testing we require the Normality of errors assumption which is presented as:

$$Y_i \overset{i.i.d}{\sim} \mathcal{N}(\mu_i, \sigma^2) \quad \implies \quad \epsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2) \tag{2.4}$$

To simplify this notation, we write the MLR in matrix form

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{2.5}$$

that is,

$$\underbrace{\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}}_{\mathbf{Y}} = \underbrace{\begin{pmatrix} 1 & x_{1,1} & \cdots & x_{p,1} \\ 1 & x_{1,2} & \cdots & x_{p,2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1,n} & \cdots & x_{p,n} \end{pmatrix}}_{\mathbf{X}} \underbrace{\begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}}_{\boldsymbol{\beta}} + \underbrace{\begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}}_{\boldsymbol{\epsilon}} \tag{2.6}$$

In this notation $\mathbf{Y}$ represents the vector of responses, $\mathbf{X}$ the design matrix, $\boldsymbol{\beta}$ the unknown vector of coefficients, and $\boldsymbol{\epsilon}$ the vector of random errors.

We also have that the errors, ϵ_i are independent and identically distributed:

$$\boldsymbol{\epsilon} \sim \mathcal{N}_n(\mathbf{0}_n, \sigma^2 \mathbf{I}_n) \quad \implies \quad \boldsymbol{Y} \sim \mathcal{N}_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}_n). \tag{2.7}$$

where $\mathcal{N}_n$ represents the nth dimensional multivariate normal distribution.

Remark: We have made four key assumptions that will become important in later chapters:

(i) Linearity: Assume the relationship between Y_i and $X_{j,i}$ is linear.

(ii) Normality: Assume that ϵ_i are identically distributed normal random variables with $\mu = 0$ and $\text{var}(\epsilon_i) = \sigma^2$.

(iii) No Autocorrelation: The errors are independent of each other.

(iv) Homoscedasticity: Assume that all the errors have the same finite variance.

2.1.1 Estimating the Coefficients

We will apply the Maximum Likelihood Estimation (MLE) to estimate the parameter $\boldsymbol{\beta}$ and show that this yields the Least Squares Procedure (LSE) which we will use to estimate our vector of coefficients $\boldsymbol{\beta}$.

First note that $\epsilon_i = Y_i$ - μ_i and that $\epsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2)$ by rearranging the notation used in (2.3). Then the likelihood function reads:

$$L(\boldsymbol{\theta}, \sigma \mid \boldsymbol{x}, \boldsymbol{Y}) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(y_i - \mu_i)^2\right) \tag{2.8}$$

$$= \frac{1}{\sigma^n (2\pi)^{n/2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mu_i)^2\right) \tag{2.9}$$

By the definition of maximum likelihood estimation, the MLE is the value that maximises the above likelihood function.
From the above it is clear that in order to maximise the likelihood function $L(\boldsymbol{\theta}, \sigma \mid \boldsymbol{x}, \boldsymbol{Y})$ we must minimise the sum below:

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - \mu_i)^2 \tag{2.10}$$

which is equivalent to the least square estimation (LSE) procedure. We proceed as follows:

$$S(\boldsymbol{\beta}) = (\boldsymbol{Y} - \boldsymbol{\mu})^T (\boldsymbol{Y} - \boldsymbol{\mu}) = (\boldsymbol{Y} - \boldsymbol{X\beta})^T (\boldsymbol{Y} - \boldsymbol{X\beta}) \tag{2.11}$$

$$= \boldsymbol{Y}^T \boldsymbol{Y} - 2\boldsymbol{\beta}^{\boldsymbol{T}} \boldsymbol{X}^{\boldsymbol{T}} \boldsymbol{Y} + \boldsymbol{\beta}^{\boldsymbol{T}} \boldsymbol{X}^T \boldsymbol{X\beta} \tag{2.12}$$

To find a stationary point on $S(\boldsymbol{\beta})$ we need:

$$\frac{dS(\boldsymbol{\beta})}{d\boldsymbol{\beta}} = 0 \quad \Longrightarrow \quad \hat{\boldsymbol{\beta}} = (\boldsymbol{X}^{\boldsymbol{T}} \boldsymbol{X})^{-1} \boldsymbol{X}^{\boldsymbol{T}} \boldsymbol{Y} \tag{2.13}$$

The second derivative is $\nabla^2 S(\boldsymbol{\beta}) = 2\boldsymbol{X}^T \boldsymbol{X}$, which is positive definite if $\boldsymbol{X}^T \boldsymbol{X}$ is non singular, thus the stationary point is a minimum.

To summarise, we have shown that in the case of Multiple Linear Regression model with normal errors, the MLE procedure yields the same estimator as the LSE procedure, and thus the methods are equivalent.

2.1.2 Model Evaluation in Brief

Now that we have introduced OLS and how to estimate the coefficients of the multivariate linear model (MLR), we would like to now briefly cover how we can evaluate the goodness of fit and compare different models to find the best one.

Let us first introduce the residual sum of squares.

Definition 14 (Residual Sum of Squares)**.** The residual sum of squares, denoted SS_E, is defined as:

$$SS_E = (\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^T(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) \tag{2.14}$$

$$= \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}\epsilon_i^2 = S(\hat{\beta}) \tag{2.15}$$

The SS_E is useful because it offers a measure of how well the model fits the data, by calculating the squared difference between the observed values and the predictions.

Definition 15 (Mean Square Error)**.** The mean square error (or residual mean square) , denoted MSE, is defined as:

$$s^2 = MSE = \frac{SS_E}{n-p} \tag{2.16}$$

The MSE is a better measure of model fit that just the SS_E since it gives a normalised measure of the error, therefore allowing for easier comparison between different data sets.

However, the main issue with the MSE is it doesn't penalise complexity which can lead to over fitting, therefore making it unable to provide useful comparisons between sparse models and more complex models.

Now we will look at our main point of model comparison, the AIC.

Definition 16 (Akaike's Information Criterion (AIC))**.** The AIC is defined as:

$$AIC = 2 \cdot dim(\boldsymbol{\theta}) - 2\ell(\hat{\boldsymbol{\theta}}) \tag{2.17}$$

,where $\ell(\boldsymbol{\theta}) = log - likelyhood$, $\boldsymbol{\theta} = parameters\ in\ the\ model$ and $dim(\theta) = dimension\ of\ \boldsymbol{\theta}$.

The AIC is the best metric of model evaluation for the following reasons:

(i) The first key point is that it balances goodness of fit and model complexity by penalising more complex models.

(ii) It also allows for comparisons between models that use different likelihoods.

In summary, the AIC makes sure the models do not become overly complex while also capturing the key patterns in the data.

Remark: The smaller the AIC, the better the model.

2.2 Shrinkage Methods

This section follows the results and steps as found in the book: An Introduction to Statistical Learning by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani [3].

In this subsection, we discuss an important alternative to Ordinary Least Squares. This approach of regression is called Shrinkage Methods. What differs this approach to OLS, is that the estimated coefficients have been shrunken towards zero relative to OLS. This has the effect of reducing the variance in exchange for a increase in bias, but depending on the type of shrinkage used, a subset of the coefficients can be set to be exactly zero.

2.2.1 Ridge Regression

Ridge regression, similar to the previously defined Ordinary Least Squares (OLS), selects coefficients which minimises the SS_E, except with the addition of an extra term as we will show below. Let us define the *Ridge regression*, choose $\boldsymbol{\beta}^{\boldsymbol{R}}_{\boldsymbol{\lambda}} = (\beta_0, \beta_1, \cdots, \beta_p)$ which minimise the below quantity:

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{n}(y_i - \mu_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{2.18}$$

where $\lambda \geq 0$ is called the *tuning parameter*.
The extra term, in this case, is called the *shrinkage penalty* and we will formally define it below.

Definition 17 (ℓ_2 Penalty)**.** The ℓ_2 shrinkage penalty is given by,

$$\|\beta\|_2 = \sum_{j=1}^{p} \beta_j^2 \tag{2.19}$$

and is small when β are close to zero, which gives the effect of shrinking the coefficients.

The purpose of the tuning parameter is to control the impact of the two terms on the regression coefficient estimates, and we will show this by considering two cases:

(i) $\lambda = 0 \implies \hat{\boldsymbol{\beta}}^{\boldsymbol{R}}_{\boldsymbol{\lambda}} = \hat{\boldsymbol{\beta}}$:
The shrinkage penalty term has no effect and the *ridge regression* will produce the OLS.

(ii) $\lambda \to \infty \implies \hat{\boldsymbol{\beta}}^{\boldsymbol{R}}_{\boldsymbol{\lambda}} \to \mathbf{0}$:
The shrinkage penalty increases, and the ridge regression estimates tend to zero.

It is critical to keep in mind is that *ridge regression* will produce a different set of coefficients, $\hat{\boldsymbol{\beta}}^{\boldsymbol{R}}_{\boldsymbol{\lambda}}$, for each value of λ, thus choosing a good value for λ is critical (see section 2.2.5).

Remark: Notice that *Ridge regression* does not apply the *shrinkage penalty* to β_0. This is because we want to shrink the estimated effect of each variable on the response; but not the intercept, which is a measure of the mean response when all of the predictors are set to zero.

That is, assuming the columns of the design matrix $\boldsymbol{X}$ have been centred to have mean zero before applying ridge regression, then:

$$\hat{\beta}_0 = \overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{2.20}$$

The LSE of $\boldsymbol{\beta}$ are *scale equivarient*, that is, multiplying X_j by a constant c leads to a scaling of the LSE $\hat{\boldsymbol{\beta}}$ by $\frac{1}{c}$, this implies that $X_j\beta_j$ is the same regardless of scaling.

However, the *ridge regression* coefficient estimates can change drastically due to the ℓ_2 penalty term. We can show this by considering a predictor X_k to be measured in hundreds, this results in a reduction of the observed values of this predictor by a factor of 100. It is clear that this kind of change won't just scale the associated predictor by a factor of 100 due to the ℓ_2 penalty. Therefore, $X_k\hat{\beta}^R_{k,\lambda}$ not only depends on the *tuning parameter* λ, but also on the scaling of this predictor and possibly even the scaling of all other predictors.

As a result of this it is important to apply *ridge regression* only after standardising the predictors, using the below formula:

$$\tilde{x}_{i,k} = \frac{x_{i,k}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{i,k} - \overline{x}_k)^2}} \tag{2.21}$$

so that they all use the same scale.

We can also derive a closed-form estimator in matrix form as we did with ordinary least squares (OLS) in the previous section; however, we must take some steps first so that we don't apply the shrinkage penalty to the intercept β_0. Suppose that the predictors have been centred by their means and scaled by their standard deviations and that the response has also been centred, that is:

$$\tilde{x}_{j,k} = \frac{x_{j,k} - \overline{x}_k}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{i,k} - \overline{x}_k)^2}} \quad \forall j \in \{1,\dots,n\} \quad \textit{and} \quad \forall k \in \{1,\dots,p\} \tag{2.22}$$

$$\tilde{y}_j = y_j - \overline{y} \quad \forall j \in \{1,\dots,n\} \tag{2.23}$$

Now that our observations have been rescaled appropriately such that the shrinkage penalty isn't applied to β_0, we proceed with least squares estimation. We must minimise the sum below :

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{n}(y_i - \mu_i)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 \tag{2.24}$$

$$= (\boldsymbol{Y} - \boldsymbol{X\beta})^T(\boldsymbol{Y} - \boldsymbol{X\beta}) + \lambda\boldsymbol{\beta^T\beta} \tag{2.25}$$

$$= \boldsymbol{Y}^T\boldsymbol{Y} - 2\boldsymbol{\beta^T X^T Y} + \boldsymbol{\beta^T X}^T\boldsymbol{X\beta} + \lambda\boldsymbol{\beta^T\beta} \tag{2.26}$$

To find a stationary point on $S(\boldsymbol{\beta})$ we need:

$$\frac{dS(\boldsymbol{\beta})}{d\boldsymbol{\beta}} = 0 \quad \Longrightarrow \quad \hat{\boldsymbol{\beta}}^{\boldsymbol{R}}_{\boldsymbol{\lambda}} = (\boldsymbol{X^T X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X^T Y} \tag{2.27}$$

The second derivative is identical to the LSE case, with the addition of $2\lambda \geq 0$, and so this point is a minimum.

Naturally, we need to answer the main question: Why can *ridge regression* produce a better model than OLS?

Ridge regression's main advantage over OLS lies mainly in the bias-variance trade off. As λ increases, the flexibility of the ridge regression fit decreases, which leads to the following:

(i) $\lambda = 0$: High variance but is unbiased.

(ii) $\lambda \to \infty$: The coefficient estimates shrink, which decreases the variance but increases bias.

In general, when the relationship between the response and predictors is approximately linear, the LSE's will have a low bias but possibly a high variance. This means that a small change in the data can cause a substantial change in the LSE coefficients, particularly in the two cases below:

(i) Number of predictors is close to the number of observations ($p \approx n$):
The LSEs are extremely variable.

(ii) Number of predictors is larger than the number of observations ($p > n$):
The OLS does not yield a unique solution.

On the other hand, *ridge regression* can still perform well by trading off a small increase in bias for a large decrease in variance, and therefore *ridge regression* performs best when the LSE estimates have a high variance.

Another advantage of *ridge regression* is that it has computational advantage over typical model selection techniques, such as best subset selection which requires testing 2^p models. This is clear as *ridge regression* only fits one model, coefficient estimation procedures can be carried out quickly.

2.2.2 The LASSO

There is one major issue with *ridge regression*, being that it will include all p predictors in the final model and that the shrinkage will shrink all the coefficients towards zero, but never set any of them to zero (unless $\lambda \to \infty$).

Including non-influential predictors unnecessarily increases model complexity, and this may not be a problem for prediction accuracy, but it can cause issues in model interpretation when the number of predictors p is large.

The *lasso* is an alternative that overcomes this issue, and is very similar to *ridge regression* except with a slight difference to the shrinkage penalty, as we will now show.

Let us define the *lasso*, choose $\boldsymbol{\beta}_{\boldsymbol{\lambda}}^{\boldsymbol{L}} = (\beta_0, \beta_1, \cdots, \beta_p)$ which minimise the below quantity:

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - \mu_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{2.28}$$

where $\lambda \geq 0$ is the *tuning parameter*, which controls the impact of the ℓ_1 shrinkage penalty $\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$.

The *lasso*, as we had before, also shrinks the coefficients towards zero but with a key difference, being that some of the coefficient estimates are set to be exactly zero for sufficiently large λ.

Therefore, similar to best subset and stepwise selection, the *lasso* performs variable selection. This means that the *lasso* yields sparse models (containing only a subset of the predictors), which improves the interpretability of the models, as compared to *ridge regression*.

As we had before, the tuning parameter controls the impact of the *shrinkage penalty* on the coefficient estimates, but using the ℓ_1 penalty instead yields slightly different effects:

(i) $\lambda = 0 \implies \hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}^{\boldsymbol{L}} = \hat{\boldsymbol{\beta}}$:
The shrinkage penalty term has no effect and the *lasso* will produce the OLS.

(ii) $\lambda \to \infty \implies \hat{\boldsymbol{\beta}}^{\boldsymbol{L}}_{\boldsymbol{\lambda}} \to \mathbf{0}$:

The shrinkage penalty increases, and the *lasso* estimates approach zero.

(iii) Between the two extremes $\lambda \in (0, \infty)$:

Based on the value of λ, the *lasso* can produce a model involving any number of variables.

2.2.3 Alternative Form of the Ridge Regression and LASSO

It can be shown that the *ridge regression* and *lasso* coefficient estimates, $\hat{\boldsymbol{\beta}}^{\boldsymbol{R}}_{\boldsymbol{\lambda}}$ and $\hat{\boldsymbol{\beta}}^{\boldsymbol{L}}_{\boldsymbol{\lambda}}$, solve the optimisation problems:

1. *Ridge regression* :

$$\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} (y_i - \mu_i)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \leq s \tag{2.29}$$

2. *Lasso* :

$$\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} (y_i - \mu_i)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq s \tag{2.30}$$

This is equivalent to the statement: $\forall\ \lambda \in (0, \infty)\ \exists\ s \geq 0$ such that the old formulations (2.18) and (2.28) give the same coefficient estimates in *ridge regression/lasso*.

One way to interpret this is to look at it as follows; consider the *lasso* formulation above, *lasso* attempts to find the set of coefficients $(\beta_0, \beta_1, \ldots, \beta_p)$, subject to the constraint for how large $\sum |\beta_j|$ can be, which leads to the smallest SS_E. If the budget is large enough that the LSE lies in it, then we get the LSE coefficient estimates (i.e. $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}^{\boldsymbol{L}}_{\boldsymbol{\lambda}}$). If the budget is small then $\sum |\beta_j|$ must be small and the amount of shrinkage applied is large.

We will now try to understand why the *lasso* has the variable selection property while *ridge regression* does not. Let's look at the case when the number of predictors is two, that is, $p = 2$:

(i) $\boldsymbol{\beta}^{\boldsymbol{R}}_{\boldsymbol{\lambda}}$ has the lowest SS_E out of all the points in the diamond: $|\beta_1| + |\beta_2| \leq s$.

(ii) $\boldsymbol{\beta}^{\boldsymbol{L}}_{\boldsymbol{\lambda}}$ has the lowest SS_E out of all the points in the circle: $\beta_1^2 + \beta_2^2 \leq s$.

We now plot β_2 against β_1, with the constraint regions as defined above.

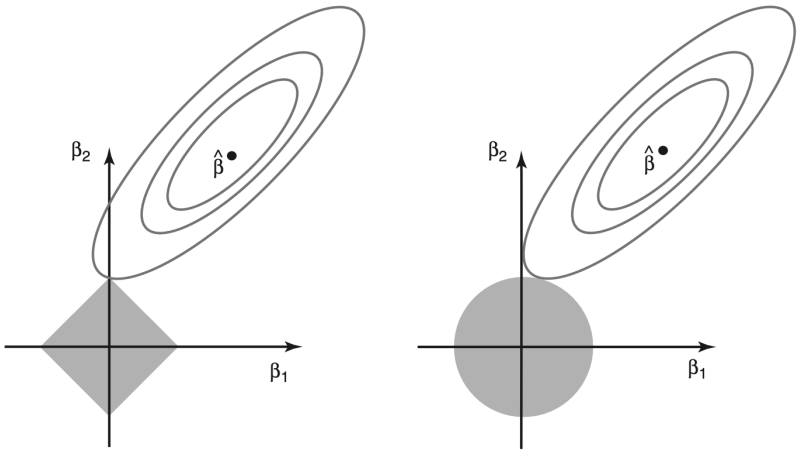

Figure 2: Contours of the error and constraint regions for the *lasso* (left) and *ridge regression* (right). *Reproduced from* An Introduction to Statistical Learning by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani under fair-use for educational purposes.

The LSE solution is denoted $\hat{\boldsymbol{\beta}}$ and the shaded area represents the constraint region. It is clear that for sufficiently large s, the constrain region will contain our LSE solution $\hat{\boldsymbol{\beta}}$ and the *lasso* and *ridge regression* estimates will be identical to LSE.

The ellipses centred about $\hat{\boldsymbol{\beta}}$ represent lines of constant SS_E. As the SS_E increases the ellipses expand away. The *lasso* and *ridge regression* estimates are given by the first contact point between these expanding ellipses and the shaded constraint region.

Let's analyse each shrinkage, *ridge regression* and *lasso* respectively:

(i) *Ridge regression* :
Since the constraint region is circular and has no sharp points, the intersection generally doesn't occur on an axis, so the coefficient estimates are non-zero.

(ii) *Lasso* :
The constraint region forms a diamond and so it has sharp corners at each axis. Therefore, the intersection generally occurs on an axis which means one of the coefficients will be zero.

In higher dimensions the above logic stays the same. Take for example the number of predictors being three, $p = 3$, then the *ridge regression* constraint region is a sphere and the *lasso* constraint region is a polyhedron with sharp corners. The only difference in higher dimensions is that the *lasso* constraint region can set many coefficients to be simultaneously zero.

2.2.4 Comparing the Ridge Regression and Lasso

We now look to discuss which of the shrinkage methods is better and under which circumstances.

As discussed before, one clear advantage of the *lasso*, as compared to *ridge regression*, is that it yields sparse models which include only a subset of the predictors. This means that the *lasso* produces models that are simpler and easier to interpret.

Something similar between the two shrinkage methods is that they yield similar behaviour in the sense that as our *tuning parameter*, λ, increases the variance decreases and the bias increases.

There is no general rule as to which shrinkage method dominates the other as it depends on the type of data we are dealing with. Here we state guidelines for which method performs better under specific circumstances:

(i) **When the** *lasso* **tends to outperform** *ridge regression* **:**
This occurs when a there is a relatively small number of predictors which have significant coefficients and the rest predictors have small (or zero) coefficients. This is clear since the *lasso* will shrink these non-influential predictors to zero, preventing issues like over fitting when including a large number of non-influential predictors in out model.

(ii) **When** *ridge regression* **tends to outperform the** *lasso* **:**
This occurs when the response variable is a function of many predictors, which all have approximately equal coefficients.

This general rule however is not convenient as before fitting a regression model the number of influential predictors is not known in real data sets.

Therefore, the best way to determine which model performs better is to use model comparison techniques, such as the AIC or cross validation.

2.2.5 Selecting the Tuning Parameter

In this section we will introduce cross validation, a method of model comparison. We will then show how this can be used to help us determine the *tuning parameter*.

Cross validation is a method of using the observed data to evaluate the model from a prediction of unseen data viewpoint.

Consider a data set with n observations and p explanatory variables. We first randomly split the n observations into two subsets; the training set and the test set.

After splitting the data set we use the training set to fit the model, denoted fit. We then make predictions of the unseen data based on on the explanatory variables data in the test set and this gives us our predictions $\hat{y}_i$, $i \in \{test\ set\}$.

Definition 18 (Predictive Mean Square Error)**.** The predictive mean squared error is defined as follows:

$$P_{MS_E} = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 \tag{2.31}$$

where m = The number of data points in the test set.

Let us now also give an algorithm which defines k-fold cross validation.

Consider data: $D = \{x_i, y_i\}_{i=1}^{n}$.

1. Partition the data D into k roughly equal size subsets: $D_1, D_2, \ldots, D_k$
2. For $j = 1, 2, \ldots, k$:
   $D_{train} = D - D_j$
   $D_{test} = D_j$
3. Obtain P_{MS_E} based on D_j, $\forall j \in 1, 2, ..., k$ and compute the mean predictive MSE:

$$\overline{P}_{MS_E} = \frac{1}{k}\sum_{j=1}^{k}(P_{MS_E})_j \tag{2.32}$$

Now that have introduced the concept of k-fold cross-validation, we can show that it provides a simple method to select the *tuning parameter*, λ.

We do this via the following:

1. Choose a grid for λ : $\lambda = h * i$ for $i = 0, 1, \ldots, N$ with $h = \frac{1}{N}$ such that λ is divisible by h.
2. Compute the 10-fold cross-validation error $(\overline{P}_{MS_E})_\lambda$ for each lambda, $\lambda = 0, h, 2h, \ldots, 1$.
3. Select the tuning parameter which gives the smallest cross validation error.
4. Refit the model using all of the observations and the selected *tuning parameter*.

2.3 Elastic Net Regression

This subsection follows the results found in this research paper: Regularization and variable selection via the elastic net by Hui Zou and Trevor Hastie [4].

We saw before that *ridge regression* produces better predictive performance via the bias-variance trade off but has the issue where it cannot produce sparse (and more interpretable) models. We overcame this issue by using the *lasso* which simultaneously does variable selection and shrinkage, yielding the positive effects of *ridge regression* with the additional ability to produce more interpretable models.

The is one issue however with the *lasso*. Suppose we have a group of variables, in which the pairwise correlations are high, the lasso will generally only select one variable and shrink the other variable's coefficient to zero, without caring which is selected.

This is an issue because this means that the *lasso* can misrepresent and neglect the effect of a variable that may be influential. Therefore, in the scenario where we have some pairwise correlation in our data, the *lasso* is inappropriate.

The *Elastic net* is an alternative that overcomes this issue, and is very similar to *ridge regression* and *lasso* except with a slight difference to the shrinkage penalty used.

Let us define the *elastic net*, choose $\boldsymbol{\beta}^E_{\lambda_1,\lambda_2} = (\beta_0, \beta_1, \ldots, \beta_p)$ which minimise the below quantity:

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{n}(y_i - \mu_i)^2 + \lambda_1 \sum_{j=1}^{p}|\beta_j| + \lambda_2 \sum_{j=1}^{p}\beta_j^2 \tag{2.33}$$

where $\lambda_1, \lambda_2 \geq 0$ are the tuning parameters.

Remark: Note that in this case the *elastic net* uses a linear combination of the ℓ_1 and ℓ_2 penalties we had before.

Similar to the *lasso* the *elastic net* simultaneously applies variable selection and shrinkage depending on the $\lambda_1, \lambda_2 \geq 0$ which are chosen.

The main difference between the *lasso* and the *elastic net* is that instead of carelessly selecting which variable is shrunk to zero, when we have pairwise correlation, the *elastic net* distributes the weight uniformly among the correlated predictors due to the ℓ_2 shrinkage.

This means that, not only does the *elastic net* yield sparse models which are more interpretable and also shrinks the coefficients, it has the additional ability of accounting for influential predictors that are correlated.

An additional advantage of the *elastic net* is that by including an additional parameter, this gives us more flexibility of what weight we assign to each shrinkage and therefore gives more freedom to tune our model.

The tuning parameters serve to control the impact of each type of shrinkage on the coefficient estimates, as we can show below:

(i) $\lambda_1 = 0$:
The ℓ_1 shrinkage penalty term has no effect and the *elastic net* will produce the *ridge regression*.

(ii) $\lambda_2 = 0$:
The ℓ_2 shrinkage penalty term has no effect and the *elastic net* will produce the *lasso*.

(ii) $\lambda_1, \lambda_2 \to \infty \implies \hat{\boldsymbol{\beta}}^{\boldsymbol{E}}_{\boldsymbol{\lambda_1,\lambda_2}} \to \mathbf{0}$:
The shrinkage penalties grow, and the *elastic net* estimates approach zero.

(iii) Between the two extremes $\lambda_1, \lambda_2 \in (0, \infty)$:
Depending on the values of λ_1 and λ_2 the *elastic net* can produce a model that has any number of predictors.

We can also define the *elastic net* using an alternative formulation, as we did before.

It can be shown that the *elastic net* coefficient estimates, $\hat{\boldsymbol{\beta}}^{E}_{\lambda_1,\lambda_2}$ solve the optimisation problem:

$$\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} (y_i - \mu_i)^2 \right\} \quad \text{subject to} \quad (1-\alpha) \sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} \beta_j^2 \leq t \tag{2.34}$$

$$where \quad \alpha = \frac{\lambda_2}{\lambda_1 + \lambda_2} \tag{2.35}$$

This is equivalent to: $\forall \alpha \in (0,1)\ \exists\, t \geq 0$ such that the formulation (2.33) yields the same coefficients as the *elastic net*.

In this case, *elastic net* attempts to find coefficients $(\beta_0, \beta_1, \ldots, \beta_p)$ subject to the constraint for how large $(1-\alpha)\sum \beta_j^2 + \alpha \sum |\beta_j|$ can be, which leads to the smallest SS_E. And again, if the budget, t, is large enough such that the LSE lies in is, then we get the LSE coefficient estimates (i.e $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}^{\boldsymbol{E}}_{\boldsymbol{\lambda_1,\lambda_2}}$) . If the budget is small, then the amount of shrinkage applied is large.

Another equivalent form which packages in R, such as `glmnet`, tend to use is as follows:

$$S(\boldsymbol{\beta}) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \mu_i)^2 + \lambda \left(\frac{(1-\alpha)}{2} \sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right) \tag{2.36}$$

We will now prove that this is equivalent to the *elastic net*.

Proposition 1. The form of the score function below,

$$S(\boldsymbol{\beta}) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \mu_i)^2 + \lambda \left(\frac{(1-\alpha)}{2} \sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right) \tag{2.37}$$

is equivalent to the *elastic net*.

Proof. First, we recall the form of the *elastic net* score function:

$$S_E(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - \mu_i)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2$$

with $\lambda_1, \lambda_2 \geq 0$ as the ℓ_1 and ℓ_2 tuning parameters respectively.
We must show that,

$$\min_{\boldsymbol{\beta}} \{S(\boldsymbol{\beta})\} = \min_{\boldsymbol{\beta}} \{S_E(\boldsymbol{\beta})\}$$

We proceed as follows:

$$\begin{aligned}\min_{\boldsymbol{\beta}} \{S(\boldsymbol{\beta})\} &= \min_{\boldsymbol{\beta}} \left\{ \frac{1}{2n} \sum_{i=1}^{n} (y_i - \mu_i)^2 + \lambda \left(\frac{(1-\alpha)}{2} \sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right) \right\} \\ &= \min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} (y_i - \mu_i)^2 + \lambda n (1-\alpha) \sum_{j=1}^{p} \beta_j^2 + 2n\alpha \sum_{j=1}^{p} |\beta_j|) \right\}\end{aligned}$$

Now, set $\lambda_1 = 2n\alpha\lambda$ and $\lambda_2 = n(1-\alpha)\lambda$ and it is clear that,

$$\min_{\boldsymbol{\beta}} \{S(\boldsymbol{\beta})\} = \min_{\boldsymbol{\beta}} \{S_E(\boldsymbol{\beta})\}$$

□

Now, let's also consider how we chose the tuning parameters λ_1 and λ_2 to produce the model with the most optimal predictive performance.

Recall: In the case of *ridge regression* and the *lasso* we had one tuning parameter to determine and we used 10-fold cross validation to do this. We will proceed very similarly in determining the two tuning parameters.

We will instead calculate the 10-fold cross validation error across a two-dimensional mesh and use this to determine which combination of λ_1, λ_2 works best.

The algorithm is as follows:

1. Choose a grid for λ_1 and λ_2 :
   $\lambda_1 = h * i$ for $i = 0, 1, \ldots, N$ with $h = \frac{1}{N}$ such that λ_1 is divisible by h.
   $\lambda_2 = h * i$ for $i = 0, 1, \ldots, N$ with $h = \frac{1}{N}$ such that λ_2 is divisible by h.
2. Compute the 10-fold cross-validation error $(\overline{P}_{MS_E})_{\lambda_1,\lambda_2}$ for each combination of λ_1 and λ_2, $\lambda_1, \lambda_2 = 0, h, 2h, \ldots, 1$.
   This produces the matrix:

$$\overline{\boldsymbol{P}}_{MS_E} = \begin{pmatrix} (\overline{P}_{MS_E})_{0,0} & (\overline{P}_{MS_E})_{h,0} & \cdots & (\overline{P}_{MS_E})_{1,0} \\ (\overline{P}_{MS_E})_{0,h} & (\overline{P}_{MS_E})_{h,h} & \cdots & (\overline{P}_{MS_E})_{1,h} \\ \vdots & \vdots & \ddots & \vdots \\ (\overline{P}_{MS_E})_{0,1} & (\overline{P}_{MS_E})_{h,1} & \cdots & (\overline{P}_{MS_E})_{1,1} \end{pmatrix} \tag{2.38}$$

3. Select the tuning parameter combination which give the smallest cross validation error.
4. Refit the model using all of the observations and the selected *tuning parameters*, λ_1, λ_2.

3 The Frequentist Approach: Application

Through this chapter, we will apply all of the frequentist statistical regression techniques presented in the previous chapter. That is, we will apply, OLS, *ridge regression*, *lasso* and *elastic net* to the usable data set, which we constructed in section (1.2).

3.1 Model Fitting

Through this section, we will fit each regression model to our data and make some inferences on our models.

3.1.1 OLS Model

We start by applying ordinary least squares regression, as introduced in Section (2.1), to our model.

Our model takes the form:

$$Y_i = \beta_0 + \beta_1 X_{1,i} + \beta_2 X_{2,i} + \cdots + \beta_p X_{p,i} + \epsilon_i \quad , i = 1, \cdots, 14 \tag{3.1}$$

$$, n = 1, \cdots, 1164 \tag{3.2}$$

More explicitly, in terms of our variables:

$$\begin{aligned} \texttt{CRSP\_SPvw} = \beta_0 &+ \beta_1 \cdot \texttt{d/p} + \beta_2 \cdot \texttt{d/y} + \beta_3 \cdot \texttt{e/p} + \beta_4 \cdot \texttt{d/e} + \beta_5 \cdot \texttt{svar} + \beta_6 \cdot \texttt{b/m} + \beta_7 \cdot \texttt{ntis} \\ &+ \beta_8 \cdot \texttt{tbl} + \beta_9 \cdot \texttt{lty} + \beta_{10} \cdot \texttt{ltr} + \beta_{11} \cdot \texttt{tms} + \beta_{12} \cdot \texttt{dfy} + \beta_{13} \cdot \texttt{dfr} + \beta_{14} \cdot \texttt{infl} \\ &+ \epsilon_i \qquad , \epsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2) \end{aligned} \tag{3.3}$$

Now, by using the `lm()` [5]function in R, we can apply OLS to our data set. The estimated coefficients in the full model are shown below:

$$\begin{aligned} \widehat{\texttt{CRSP\_SPvw}} = (7.52 \times 10^{-3}) &+ (-1.01) \cdot \texttt{d/p} + (1.01) \cdot \texttt{d/y} + (1.28 \times 10^{-3}) \cdot \texttt{e/p} + (0) \cdot \texttt{d/e} \\ &+ (3.98 \times 10^{-1}) \cdot \texttt{svar} + (2.81 \times 10^{-3}) \cdot \texttt{b/m} + (1.61 \times 10^{-3}) \cdot \texttt{ntis} + (3.22 \times 10^{-4}) \cdot \texttt{tbl} \\ &+ (-1.11 \times 10^{-2}) \cdot \texttt{lty} + (-8.20 \times 10^{-4}) \cdot \texttt{ltr} + (0) \cdot \texttt{tms} + (9.43 \times 10^{-2}) \cdot \texttt{dfy} \\ &+ (1.98 \times 10^{2}) \cdot \texttt{dfr} + (1.10 \times 10^{-4}) \cdot \texttt{infl} \end{aligned} \tag{3.4}$$

Remark: The code output showed two rows, associated to the d/e (dividend payout ratio) and *tms* (term-spread) variables, which show `NA` for all columns. This error typically occurs for any of the following reasons:

(i) Perfect Multicollinearity.

(ii) Redundancy.

(iii) Overspecified model.

In this case the *d/e* (dividend-payout ratio) and *tms* (term-spread) error can be explained by perfect multicollinearity or redundancy. The problem arises from the way we have calculated these quantities:

$$tms = lty - tbl, \qquad d/e = d/p - e/p \tag{3.5}$$

The issue is clear, since *lty* and *tbl* are variables already included in our dataset, so including *tms*, which is dependent on these two variables, will definitely create redundancy. We can apply similar reasoning to the *d/e* variable.

Remark: With regards to how this error affects our regression, the affected variables are simply not included in regression calculations, that is to say, the coefficients for these variables is set to zero before we perform Least Square Estimation (LSE). i.e. $\beta_4 = \beta_{11} = 0$

Now, let us analyse our summary output:

(i) **Multiple R-Squared:**
$R^2 = 0.9915 \implies 99.15\%$ of the variation in the stock returns, *(CRSP SPvw)*, is explained by the model.

(ii) **Global F-test:** Test: $H_0 : \boldsymbol{\beta} = \mathbf{0}$ vs $H_1 : \boldsymbol{\beta} \neq \mathbf{0}$
$p - value < 2.2e - 16 \implies$ reject the null at any sensible significance level. Therefore the S&P500 stock returns, *(CRSP SPvw)*, has some relation with the explanatory variables.

(iii) **T-tests:** Test: $H_0 : \beta_i = 0$ vs $H_1 : \beta_i \neq 0$ $\forall i \in \{1, 2, ..., 14\}$
$p - value < 0.05 \implies$ reject the null at the $\alpha = 0.05$ significance level for $i \in \{1, 2, 5, 6, 12\}$. This tells us that there is overwhelming evidence that the variables; Dividend-Price Ratio *(d/p)*, Dividend-Yield *(d/y)*, Stock Volatility *(svar)*, Book-Market Ratio *(b/m)* and Default Yield *(dfy)*, are related to the S&P500 stock returns *(CRSP SPvw)*.

Comments: We must be critical of this model for the following reasons:

(i) The R-Squared is unusually high which indicates there may be issues of over-fitting.

(ii) Initially, two variables, *d/e* and *tms*, were removed for regression calculations. This is an issue because, although this is mathematically justified, in practice these two variables still capture distinct econometrically meaningful effects, so omitting them forces the remaining coefficients to subsume their effect, which can lead to inaccuracies in the model.

This naturally leads us to apply a shrinkage method so that the collinear predictors can be included in the model, thus preserving their econometric effects without causing singularities.

Overall, this model is a good fit; however, we must also conduct model checking to see if it satisfies our model assumptions.

Let us now calculate some model evaluation metrics, in order to compare our models later on.

First, we calculate the 10-fold cross validation MSE using the `trainControl()` function within the `caret` package [10] in R. This yields:

$$\overline{P}_{MS_E} = 2.712819 \times 10^{-5} \tag{3.6}$$

Next, we calculate the Akaike Information Criterion (AIC) and find that:

$$AIC = -9003.921 \tag{3.7}$$

[5] Everything relating to the OLS model, described in this subsection, has been done in R-script and can be found in Appendix B.

3.1.2 Shrinkage Methods

Now, we apply the shrinkage methods, as introduced in Section (2.2), to our usable data set.

Our model, again, takes the form:

$$\begin{aligned}\texttt{CRSP\_SPvw} = \beta_0 &+ \beta_1 \cdot \texttt{d/p} + \beta_2 \cdot \texttt{d/y} + \beta_3 \cdot \texttt{e/p} + \beta_4 \cdot \texttt{d/e} + \beta_5 \cdot \texttt{svar} + \beta_6 \cdot \texttt{b/m} + \beta_7 \cdot \texttt{ntis} \\ &+ \beta_8 \cdot \texttt{tbl} + \beta_9 \cdot \texttt{lty} + \beta_{10} \cdot \texttt{ltr} + \beta_{11} \cdot \texttt{tms} + \beta_{12} \cdot \texttt{dfy} + \beta_{13} \cdot \texttt{dfr} + \beta_{14} \cdot \texttt{infl} \\ &+ \epsilon_i \qquad\qquad , \epsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2) \end{aligned} \tag{3.8}$$

Throughout this subsection, we will be using the `glmnet` package [9] in R.

The first thing we must do to fit the *ridge regression*, *lasso* and *elastic net* models is to determine the optimal tuning parameters which minimise the 10-fold cross-validation error ($(\overline{P}_{MS_E})_\lambda$) of our models. Specifically, for *ridge regression* and the *lasso* we pick the tuning parameter λ. This can be done using the `cv.glmnet()` [6]function, which automatically selects a suitable sequence of lambdas on which the algorithm, as described in Section (2.2.5), can be applied.

Then, for the *elastic net* we select the tuning parameters $\{\lambda, \alpha\}$. Unfortunately, unlike for the *ridge regression* and *lasso* models, the `glmnet` package does not contain a function which automatically selects a grid on which the algorithm, as described in Section (2.3), can be applied.

In order to determine the optimal tuning parameters, we must create the grid for the tuning parameters on which a for loop, in combination with `cv.glmnet()` can apply the algorithm.

Table 2, below summarises the optimal tuning parameters for each model, the predictors shrunk to zero, and the corresponding model comparison metrics for each model.

Model	$\boldsymbol{\lambda}$	$\boldsymbol{\alpha}$	**Zeroed Predictors**	**10-fold CV Error**	**AIC**
Ridge Regression	1.12×10^{-3}	-	-	1.556×10^{-3}	-4283.586
Lasso	1.52×10^{-5}	-	`tbl, tms, infl`	2.771×10^{-5}	-9000.305
Elastic Net	1.10×10^{-5}	0.9489	`ntis, tbl, tms, infl`	2.767×10^{-5}	-8995.06

Table 2: Comparison of shrinkage models: optimal tuning parameters, zeroed predictors (predictors coefficients have been set to zero by the shrinkage), 10-fold CV Error and AIC.

As before, we calculate some model evaluation metrics, after fitting the models with their respective optimal tuning parameters, in order to compare our models later on.

First, we calculate the 10-fold cross validation MSE for each model, which we have already found via our selection of the tuning parameters algorithms. We simply extract the 10-fold cross validation MSE, associated with our optimal tuning parameters.

Next, we calculate the Akaike Information Criteria (AIC) for each model, using the formula:

$$AIC = 2 \cdot dim(\boldsymbol{\theta}) - 2\ell(\hat{\boldsymbol{\theta}}) \tag{3.9}$$

The resulting 10-fold cross validation errors and AIC for each model are displayed in Table 2.

[6]The R-code which is used to determine the optimal tuning parameters, 10-fold cross validation error and the AIC can be found in Appendix B.

Ridge Regression Model:

Now, let us look closer at the *ridge regression* model we have produced.

We use the `glmnet()` [7]function to fit our model for value of the optimal tuning parameter and the coefficient estimates produced are below:

$$\begin{aligned}\widehat{\texttt{CRSP\_SPvw}} = {} & (5.84 \times 10^{-2}) + (-1.37 \times 10^{-1}) \cdot \texttt{d/p} + (2.61 \times 10^{-1}) \cdot \texttt{d/y} + (-1.13 \times 10^{-1}) \cdot \texttt{e/p} \\ & + (-1.11 \times 10^{-1}) \cdot \texttt{d/e} + (-1.56) \cdot \texttt{svar} + (-2.21 \times 10^{-2}) \cdot \texttt{b/m} + (-1.28 \times 10^{-1}) \cdot \texttt{ntis} \\ & + (-1.46 \times 10^{-2}) \cdot \texttt{tbl} + (-2.66 \times 10^{-2}) \cdot \texttt{lty} + (3.15 \times 10^{-1}) \cdot \texttt{ltr} + (-4.14 \times 10^{-2}) \cdot \texttt{tms} \\ & + (6.45 \times 10^{-1}) \cdot \texttt{dfy} + (6.84 \times 10^{-1}) \cdot \texttt{dfr} + (1.12 \times 10^{-1}) \cdot \texttt{infl} \end{aligned} \tag{3.10}$$

Previously, we found the optimal tuning parameter, $\lambda = 0.00112$, via 10-fold cross validation (Table 2). We observe that this value of the tuning parameter, λ, is very close to zero. This indicates that the amount of shrinkage applied is very small and the ridge regression coefficients are close to the OLS coefficients, for the reasoning used in Section (2.2.1).

We can also see that, as desired, the *ridge regression* model has included the *d/e* and *tms* variables thus preserving their econometric effects without causing singularities.

Lasso Model:

Now, we look closer at the *lasso* model we have produced.

Now, we use the `glmnet()` [7] function to fit our model for this value of the tuning parameter and the coefficient estimates produced are below:

$$\begin{aligned}\widehat{\texttt{CRSP\_SPvw}} = {} & (8.95 \times 10^{-3}) + (-9.99 \times 10^{-1}) \cdot \texttt{d/p} + (1.00) \cdot \texttt{d/y} + (-8.56 \times 10^{-4}) \cdot \texttt{e/p} \\ & + (-2.08 \times 10^{-3}) \cdot \texttt{d/e} + (3.71 \times 10^{-1}) \cdot \texttt{svar} + (2.15 \times 10^{-3}) \cdot \texttt{b/m} + (-8.10 \times 10^{-5}) \cdot \texttt{ntis} \\ & + (0) \cdot \texttt{tbl} + (-9.90 \times 10^{-3}) \cdot \texttt{lty} + (1.94 \times 10^{-3}) \cdot \texttt{ltr} + (0) \cdot \texttt{tms} \\ & + (1.00 \times 10^{-1}) \cdot \texttt{dfy} + (2.58 \times 10^{-2}) \cdot \texttt{dfr} + (0) \cdot \texttt{infl} \end{aligned} \tag{3.11}$$

As we had with our *ridge regression* tuning parameter, we note that our value of the tuning parameter, $\lambda = 1.52 \times 10^{-5}$, is very close to zero. This indicates that the amount of shrinkage applied is very small.

We can see that the *lasso*, as we wanted, conducted variable selection and has yielded a sparse model.

Therefore, the *lasso* has identified some predictors as non-influential and thus has shrunk their coefficients down to zero.

More specifically, the treasury-bill rate *(tbl)*, term-spread *(tms)* and inflation rate *(infl)* coefficients have all been shrunk to zero and the *lasso* has identified them as non- influential predictors.

This confirms the theory that we covered in Section (2.2.2), in that the *lasso* has automatically performed variable selection and produced a more interpretable sparse model as desired.

[7]Everything relating to the *ridge regression* and *lasso* models, described in this subsection, has been done in R-script and can be found in Appendix B.

We also notice that since: $tms = lty - tbl$

The variables *tms*, *lty* and *tbl* are all collinear, yet the *lasso* has carelessly selected the Long-Term Yield *(lty)* and shrunk the Treasury-Bill rate *(tbl)*, Term-Spread *(tms)* to zero.

This is clearly an issue, since it is possible that the Treasury-Bill rate *(tbl)* and Term-Spread *(tms)* are indeed influential predictors, but the *lasso* has not accounted for this.

Elastic Net Model:

Now, we look closer at the *elastic net* model we have produced.

We use the `glmnet()` [8]function to fit our model at these values of the tuning parameters and the coefficient estimates produced are below:

$$\begin{aligned}\widehat{\texttt{CRSP\_SPvw}} = {} & (8.56 \times 10^{-3}) + (-9.76 \times 10^{-1}) \cdot \texttt{d/p} + (1.00) \cdot \texttt{d/y} + (-2.44 \times 10^{-2}) \cdot \texttt{e/p} \\ & + (-2.56 \times 10^{-2}) \cdot \texttt{d/e} + (3.72 \times 10^{-1}) \cdot \texttt{svar} + (2.32 \times 10^{-3}) \cdot \texttt{b/m} + (0) \cdot \texttt{ntis} \\ & + (0) \cdot \texttt{tbl} + (-1.04 \times 10^{-2}) \cdot \texttt{lty} + (2.33 \times 10^{-3}) \cdot \texttt{ltr} + (0) \cdot \texttt{tms} \\ & + (1.00 \times 10^{-1}) \cdot \texttt{dfy} + (2.66 \times 10^{-2}) \cdot \texttt{dfr} + (0) \cdot \texttt{infl} \end{aligned} \tag{3.12}$$

We will first discuss two observations with regard to our tuning parameters; $\{\lambda, \alpha\} = \{1.10 \times 10^{-5}, 0.9489\}$ (Table 2).

The first observation is that since $\alpha = 0.9489$, is very close to one, the *elastic net* is very similar to the *lasso* model. We can show this when we look at the formulation used, (2.36), as $\alpha \to 1$:

$$\lim_{\alpha \to 1} \{S(\boldsymbol{\beta})\} = \lim_{\alpha \to 1} \left\{ \frac{1}{2n} \sum_{i=1}^{n} (y_i - \mu_i)^2 + \lambda \left(\frac{(1-\alpha)}{2} \sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right) \right\} \tag{3.13}$$

$$= \frac{1}{2n} \sum_{i=1}^{n} (y_i - \mu_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{3.14}$$

Thus, it is clear that for $\alpha = 1$, the *elastic net* yields the *lasso*.

The second observation we make is that since our value of λ, is again very close to zero, the amount of shrinkage applied is very small.

Now, as we had with the *lasso* model, we observe that the *elastic net*, has conducted variable selection and has yielded a sparse model.

Therefore, the *elastic net* has identified some predictors as non-influential, and thus has shrunk their coefficients down to zero.

More specifically, the net issuing activity *(ntis)*, treasury-bill rate *(tbl)*, term-spread *(tms)* and inflation rate *(infl)* coefficients have all been shrunk to zero and the *elastic net* has identified them as non-influential predictors.

Remark: The *elastic net* has identified the same non-influential predictors as the *lasso*, with the addition of the *(ntis)* variable.

[8]Everything relating to the *elastic net* model, described in this subsection, has been done in R-script and can be found in Appendix B.

We notice that again, although *tms*, *lty* and *tbl* are all collinear, the *elastic net* has carelessly selected the Long-Term Yield *(lty)* and shrunk the others to zero.

There are two main reasons why this has happened:

(i) **The α tuning parameter:**
We saw earlier that our alpha tuning parameter was very close to one, which produces a model very similar to the *lasso*. This explains why our *elastic net* model has produced the same problem we had before (that is, the issue of carelessly shrinking collinear variables).

(ii) **Cross-Validation:**
The cross-validation procedure may have decided that the slight improvement of including the treasury-bill rate *(tbl)* and the inflation rate *(infl)* variables is not worth the increase in complexity of the model or the shrinkage penalty.

3.1.3 Summary

Through this subsection, we will now make some observations of our results and compare our models.

Firstly, let us look at the variable coefficients that seem unusual:

(i) **Dividend-Price Ratio *(d.p)*:**
In the shrinkage models, the coefficient estimates of *d.p* are all negative, which is unusual as we typically expect the dividend-price ratio to be positively related to stock returns.

(ii) **Earning-Price Ratio *(e.p)*:**
The shrinkage estimates of *e.p* are also all negative, which is unusual as we typically expect the earning-price ratio to be positively related to stock returns.

(iii) **Dividend-Earning Ratio *(d.e)*:**
The shrinkage estimates of *d.e* are also all negative, which is unusual as we typically expect the dividend-earning ratio to be positively related to stock returns.

This issue likely arises due to the following issues:

(i) **Multicollinearity:**
Since $d/p = d/e + e/p$, including all three predictors may cause unstable coefficients as they are collinear (they explain the same variation).

(ii) **Shrinkage:**
In the *lasso* and *elastic net* models, regularisation penalises coefficients to reduce overfitting, which could cause this issue since variables are correlated.

(iii) **Non-influential Predictors:**
In the *lasso* and *elastic net* models, the coefficients of *e.p* and *d.e* are very small, which could just indicate that they are non-influential predictors.

Finally, we compare the model comparison metrics produced. We produce the summary table below:

Model	10-fold CV Error	AIC
OLS	2.713×10^{-5}	-9003.921
Ridge Regression	1.556×10^{-3}	-4283.586
Lasso	2.771×10^{-5}	-9000.305
Elastic Net	2.767×10^{-5}	-8995.06

Table 3: Comparison of models with 10-fold CV Error and AIC.

These results tell me that the OLS model, *lasso* model and *elastic net* model all perform very similarly with marginal performance difference. However, both the 10-fold cross validation error and AIC point to the OLS model performing the best.

This is an unexpected result since we know that our data set contains multicollinearity, an issue the shrinkage methods was supposed to rectify and improve upon.

The only plausible explanation of this is that in some real-world scenarios, the amount of shrinkage these shrinkage methods apply just isn't powerful enough to deal with these issues.

3.2 Model Checking

Through this subsection, we will verify whether or not our models satisfy the four assumptions of linear regression, as described in Section (2.1). Then, if issues are detected, we will describe the different possible methods of addressing them.

Recall: The four assumptions of linear regression are:

(i) Normality: The errors ϵ_i are identically distributed normal random variables.

(ii) Homoscedasticity: All of the errors have the same finite variance.

(iii) Linearity: The relationship between the response variable and the predictors is linear.

(iv) No Autocorrelation: The errors are independent of each other.

We will generate three plots and make some inferences from them which will tell us if our model satisfies the four assumptions. We can do this using the `plot()` function in R, and our plots are shown below:

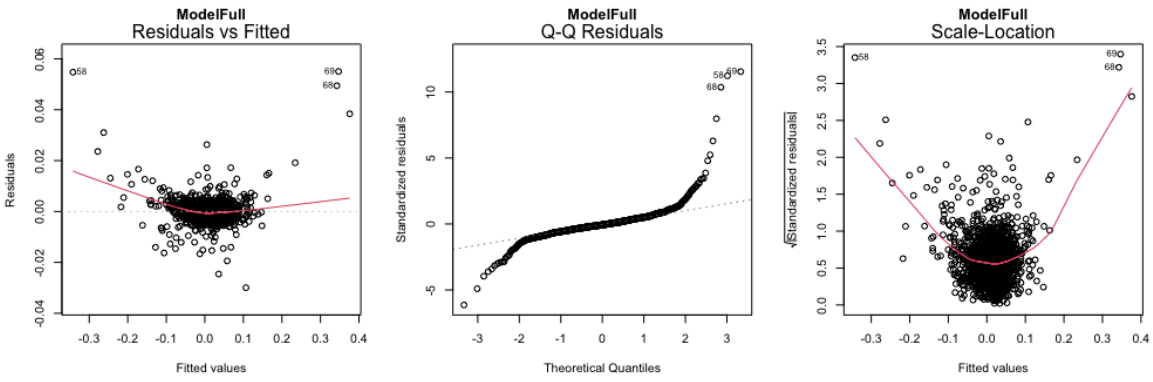

Figure 3: This figure shows the following three plots; Residuals vs Fitted (left), Q-Q Residuals (middle) and Scale-Location (right)

We make the following observations:

(i) **Q-Q Plot:**
The plot shows clear deviation at the tails implying the residuals are not normal and in particular they are heavy tailed. Therefore, the normality of errors assumption is violated.

(ii) **Scale-Location Plot:**
The plot shows a general increase in the spread of the standardised residuals as the fitted values increase. This resembles a funnel shape and implies that the variance increases with the expectation and therefore we detect heteroscedasticity (non-constant variance).

(iii) **Residuals vs Fitted:**
This plot shows a very slight curvature which could imply the existence of non-linear relationships.

It is clear that we have identified issues with the validity of our assumptions; in particular, the normality of the errors assumption is especially violated.

We must therefore apply some statistical techniques in order to rectify these issues.

3.2.1 Transformation of the Response

In this subsection, we will describe the first type of statistical technique to address the issues outlined previously.

If we detect heteroscedasticity (non-constant variance) or non-normality of the errors it is possible to obtain a better model by transforming the response variable Y_i.

We will describe a commonly used type of transformation of the response.

Box-Cox Transformation:

The most common types of transformations are:

(i) $\sqrt{Y_i}$; if $Var[Y_i] \propto \mathbb{E}[Y_i]$

(ii) $log(Y_i)$; if $Var[Y_i] \propto (\mathbb{E}[Y_i])^2$

(iii) $1/\sqrt{Y_i}$; if $Var[Y_i] \propto (\mathbb{E}[Y_i])^3$

(iv) $1/Y_i$; if $Var[Y_i] \propto (\mathbb{E}[Y_i])^4$

The above transformations are just special cases of the more general family of transformations, called the Box-Cox transformation,

$$y^{(\lambda)} = \begin{cases} \frac{y^\lambda - 1}{\lambda}, & \text{when } \lambda \neq 0; \\ \log(y), & \text{when } \lambda = 0. \end{cases} \tag{3.15}$$

The additional parameter, λ, used in the Box-Cox transformation can be estimated from the data using Maximum Likelihood Estimation (MLE). That is, we find the value of λ which maximises,

$$\underset{\lambda}{argmax} \left[L(\hat{\boldsymbol{\beta}}, \hat{\sigma}; \lambda, \boldsymbol{y}) \right] \tag{3.16}$$

Remark: It is clear from the definition of the Box-Cox transformation, it can only be applied to strictly positive values. This is an issue since, returns can take negative and positive values. To account for this, before we proceed with calculations, I have added the constant one to the returns.

We can use the `boxcox()` [9]function within the `MASS` package [8] in R, to produce a plot of the log-Likelihood and find the optimal value of λ. The plot can be found in Appendix D.

We find the optimal value of λ to be, $\lambda = -0.06060606$.

Since the value of λ that maximises the log-Likelihood is very close to zero, so we can simply use the $log(Y_i)$ transformation.

The transformed model takes the form:

$$\begin{aligned} log(\texttt{CRSP\_SPvw}) = \beta_0 &+ \beta_1 \cdot \texttt{d/p} + \beta_2 \cdot \texttt{d/y} + \beta_3 \cdot \texttt{e/p} + \beta_4 \cdot \texttt{d/e} + \beta_5 \cdot \texttt{svar} + \beta_6 \cdot \texttt{b/m} + \beta_7 \cdot \texttt{ntis} \\ &+ \beta_8 \cdot \texttt{tbl} + \beta_9 \cdot \texttt{lty} + \beta_{10} \cdot \texttt{ltr} + \beta_{11} \cdot \texttt{tms} + \beta_{12} \cdot \texttt{dfy} + \beta_{13} \cdot \texttt{dfr} + \beta_{14} \cdot \texttt{infl} \\ &+ \epsilon_i \qquad\qquad , \epsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2) \end{aligned} \tag{3.17}$$

As we did before, we can use the `plot()` function in R to produce the three plots as shown below:

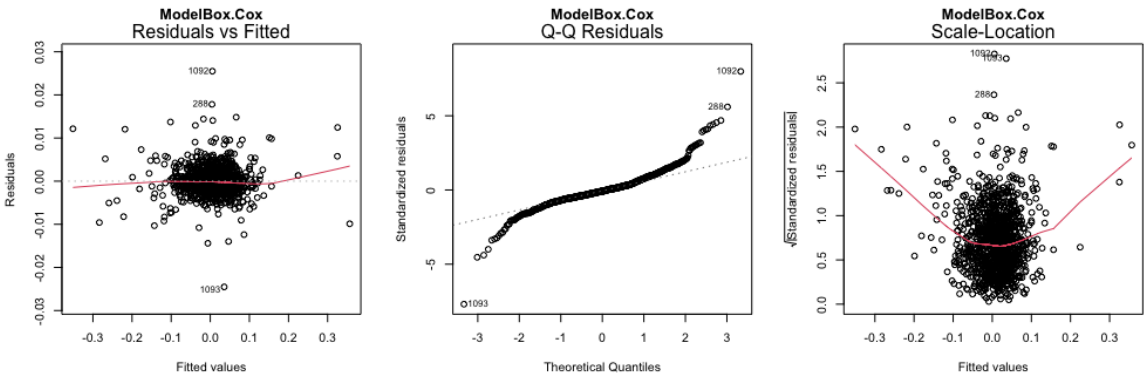

Figure 4: This figure shows the following three plots; Residuals vs Fitted (left), Q-Q Residuals (middle) and Scale-Location (right).

We make the following observations:

(i) **Q-Q Plot:**
The plot shows clear deviation at the tails implying the residuals are not normal and in particular they are heavy tailed. Therefore, the normality of errors assumption is violated.

(ii) **Scale-Location Plot:**
The plot again resembles a funnel shape. This implies that the variance increases with the expectation, and therefore we detect heteroscedasticity (non-constant variance), as we had before.

(iii) **Residuals vs Fitted:**
There is now less curvature than before, implying our linearity assumption is better satisfied.

[9]Everything relating to, the application of the Box-Cox transformation has been done in R and the code can be found in Appendix C.

In summary, we can see that the issues of non-normality and heteroscedasticity persist despite the application of the Box-Cox transformation. Therefore, the Box-Cox transformation has been ineffective in solving these issues, and we must look for an alternative.

3.2.2 Changing the Distribution of the Errors Assumption

In Section (1.3.2) we produced an empirical density plot, calculated the excess kurtosis, and deduced that our data is heavy tailed. That is, the excess kurtosis is positive and, therefore, the distribution of the returns is leptokurtic.

One of the most important assumptions made before constructing our regression models was that of the normality of the errors.

More precisely, in the formulation (2.2), we stated that:

$$\epsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2) \quad \Longrightarrow \quad Y_i \overset{i.i.d}{\sim} \mathcal{N}(\mu_i, \sigma^2) \tag{3.18}$$

As a result, we are assuming that our response variable (returns) is normally distributed. This is clearly not a valid assumption given the heavy tailed distributional nature of the asset returns.

This explains why the normality of errors assumption was initially violated and further persisted even after applying the Box-Cox transformation.

We can tweak the model assumptions slightly to account for this heavy tailed behaviour. That is, instead of assuming our data is normally distributed, we adopt a distribution with heavier tails.

Based on the empirical density plot, the Student t-distribution is the most appropriate distribution to fit. This is because it follows a bell-shaped curve, similar to that of the normal distribution, but it has heavier tails as compared to the normal distribution, making it a better fit for the data.

Altogether, the updated model assumptions are:

(i) Student t-distribution: The errors ϵ_i are identically t-distributed random variables with n-degrees of freedom.

(ii) Homoscedasticity: All of the errors have the same finite variance.

(iii) Linearity: The relationship between the response variable and the predictors is linear.

(iv) No Autocorrelation: The errors are independent

As before, our model takes the form:

$$\begin{aligned} \texttt{CRSP SPvw} = \beta_0 &+ \beta_1 \cdot \texttt{d/p} + \beta_2 \cdot \texttt{d/y} + \beta_3 \cdot \texttt{e/p} + \beta_4 \cdot \texttt{d/e} + \beta_5 \cdot \texttt{svar} + \beta_6 \cdot \texttt{b/m} + \beta_7 \cdot \texttt{ntis} \\ &+ \beta_8 \cdot \texttt{tbl} + \beta_9 \cdot \texttt{lty} + \beta_{10} \cdot \texttt{ltr} + \beta_{11} \cdot \texttt{tms} + \beta_{12} \cdot \texttt{dfy} + \beta_{13} \cdot \texttt{dfr} + \beta_{14} \cdot \texttt{infl} \\ &+ \epsilon_i \qquad\qquad , \epsilon_i \overset{i.i.d}{\sim} t_n \end{aligned} \tag{3.19}$$

The first step we must take is to estimate the degrees of freedom of the Student t-distribution which best fits the distribution of the returns data.

Maximum Likelihood Estimation:

We will first estimate the degrees of freedom using Maximum Likelihood Estimation (MLE).

We use the `fitdistr()` function within the `MASS` package [8] in R, to calculate the MLE of n as shown in Appendix E. We find that the MLE of n is, $\hat{n} = 3.68$.

Now we can use the `heavyLm()` function within the `heavy` package, [11], in R to fit the model with t-distributed errors with 3.68 degrees of freedom. The coefficient estimates are shown below:

$$\begin{aligned}\widehat{\texttt{CRSP\_SPvw}} = &(8.90 \times 10^{-3}) + (-1.00) \cdot \texttt{d/p} + (1.00) \cdot \texttt{d/y} + (5.00 \times 10^{-4}) \cdot \texttt{e/p} + (0) \cdot \texttt{d/e} \\ &+ (1.66 \times 10^{-1}) \cdot \texttt{svar} + (4.00 \times 10^{-4}) \cdot \texttt{b/m} + (-2.90 \times 10^{-3}) \cdot \texttt{ntis} \\ &+ (-1.00 \times 10^{-4}) \cdot \texttt{tbl} + (-2.00 \times 10^{-4}) \cdot \texttt{lty} + (1.60 \times 10^{-3}) \cdot \texttt{ltr} + (0) \cdot \texttt{tms} \\ &+ (3.67 \times 10^{-2}) \cdot \texttt{dfy} + (1.00 \times 10^{-4}) \cdot \texttt{dfr} + (1.18 \times 10^{-2}) \cdot \texttt{infl} \end{aligned} \tag{3.20}$$

Remark: In order to prevent singularity of the design matrix due to multicollinearity, I choose to omit the Dividend-Earning ratio *(d.e)* and the Term-Spread *(tms)* variables from our model. This means that we must again be critical of our model, as we did with the OLS model in Section (3.1)

We make the following observations from the summary output:

(i) **Z-tests:** Test: $H_0 : \beta_i = 0$ vs $H_1 : \beta_i \neq 0$ $\forall i \in \{1, 2, ..., 14\}$
$p - value < 0.05 \implies$ reject the null at the $\alpha = 0.05$ significant level for $i \in \{1, 2, 5, 12\}$. This tells us that there is overwhelming evidence that the variables; Dividend-Price Ratio *(d/p)*, Dividend-Yield *(d/y)*, Stock Volatility *(svar)* and Default Yield *(dfy)*, are related to the S&P500 stock returns *(CRSP SPvw)*.

(ii) **Scale Estimates:** $\hat{\sigma}^2 = 5.01 \times 10^{-6}$
The scale estimate tells me that the standard deviation of the residuals is very low. Therefore, only a very small amount of the variation in the data is not explained by the model.

(iii) **Log-Likelihood:** $\ell(\hat{\boldsymbol{\theta}}) = 5014.441$
We use the log-likelihood to calculate the Akaike Information Criterion (AIC) as follows:

$$AIC = 2 \cdot dim(\boldsymbol{\theta}) - 2\ell(\hat{\boldsymbol{\theta}}) = -10000.88 \tag{3.21}$$

Finally, we use the three plots below to check the model assumptions:

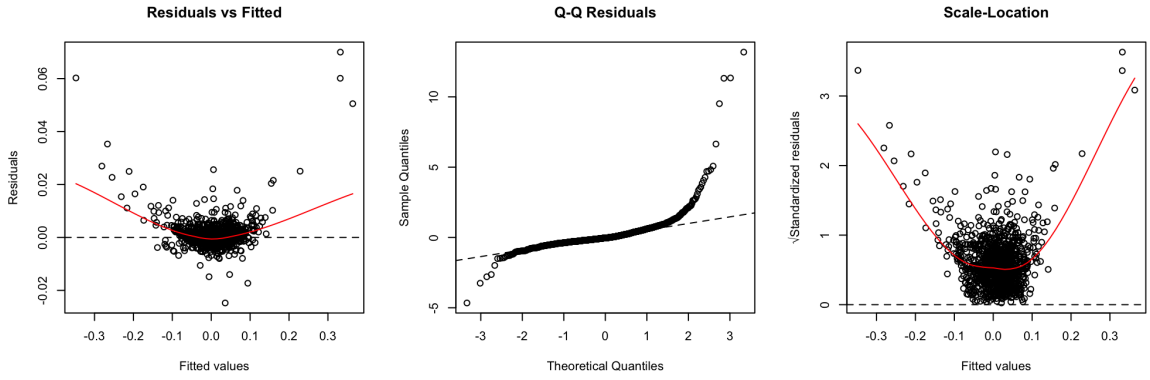

Figure 5: This figure shows the following three plots; Residuals vs Fitted (left), Q-Q Residuals (middle) and Scale-Location (right).

We make the following observations:

(i) **Q-Q Plot:**
The plot shows some deviations at the tails. However, the middle alignment with the line is much improved.

(ii) **Scale-Location Plot:**
The plot again resembles a funnel shape. This implies that the variance increases with the expectation, and therefore we detect heteroscedasticity (non-constant variance), as we had before.

(iii) **Residuals vs Fitted:**
This plot shows a very slight curvature which could imply the existence of non-linear relationships.

Expectation Maximisation (EM) Algorithm:

If I don't specify the degrees of freedom of in the `heavyLm()` function, it will automatically estimate the degrees of freedom by it self, using the Expectation Maximisation (EM) Algorithm[10].

The Expectation Maximisation (EM) algorithm finds an estimate for the degrees of freedom n, $n = 1.81639$. The output is shown below:

$$\begin{aligned}\widehat{\texttt{CRSP\_SPvw}} = {} & (9.30 \times 10^{-3}) + (-1.00) \cdot \texttt{d/p} + (1.00) \cdot \texttt{d/y} + (5.00 \times 10^{-4}) \cdot \texttt{e/p} + (0) \cdot \texttt{d/e} \\ & + (1.33 \times 10^{-1}) \cdot \texttt{svar} + (-6.00 \times 10^{-4}) \cdot \texttt{b/m} + (-1.20 \times 10^{-3}) \cdot \texttt{ntis} \\ & + (-1.90 \times 10^{-3}) \cdot \texttt{tbl} + (1.50 \times 10^{-3}) \cdot \texttt{lty} + (4.00 \times 10^{-4}) \cdot \texttt{ltr} + (0) \cdot \texttt{tms} \\ & + (6.80 \times 10^{-2}) \cdot \texttt{dfy} + (-3.80 \times 10^{-3}) \cdot \texttt{dfr} + (1.69 \times 10^{-2}) \cdot \texttt{infl} \end{aligned} \tag{3.22}$$

We make the following observations from the summary output:

(i) **Z-tests:** Test: $H_0 : \beta_i = 0$ vs $H_1 : \beta_i \neq 0$ $\forall i \in \{1, 2, ..., 14\}$
$p - value < 0.05 \implies$ reject the null at the $\alpha = 0.05$ significant level for $i \in \{1, 2, 5, 12\}$. This tells us that there is overwhelming evidence that the variables; Dividend-Price Ratio *(d/p)*, Dividend-Yield *(d/y)*, Stock Volatility *(svar)* and Default Yield *(dfy)*, are related to the S&P500 stock returns *(CRSP SPvw)*.

(ii) **Scale Estimates:** $\hat{\sigma}^2 = 3.04 \times 10^{-6}$
The scale estimate tells me that the standard deviation of the residuals is very low. Therefore, only a very small amount of the variation in the data is not explained by the model.

(iii) **Log-Likelihood:** $\ell(\hat{\boldsymbol{\theta}}) = 5045.865$
Alone, the log-likelihood just represents the level of plausibility (not probability) of the estimated parameter given the data.

The observations we make from this summary output is identical to what we had before.

[9]Everything relating to the *elastic net* model, described in this subsection, has been done in R-script and can be found in Appendix E.

[10]The `heavyLm()` function uses the Expectation Maximisation (EM) algorithm to estimate the degrees of freedom, and more information on this can be found in reference [5].

Next, let's compute the Akaike Information Criterion (AIC):

$$AIC = 2 \cdot dim(\boldsymbol{\theta}) - 2\ell(\hat{\boldsymbol{\theta}}) = -10063.73 \tag{3.23}$$

This AIC is much lower than using the MLE degrees of freedom. Therefore, this model is better.

Again, we use the three plots below to check the model assumptions:

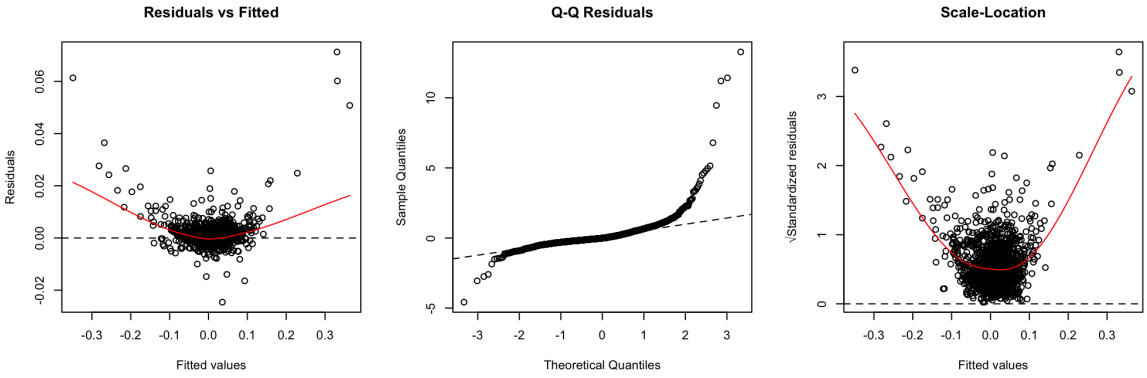

Figure 6: This figure shows the following three plots; Residuals vs Fitted (left), Q-Q Residuals (middle) and Scale-Location (right).

We make the following observations:

(i) **Q-Q Plot:**
The plot shows some deviations at the tails. However, the middle alignment with the line is much improved.

(ii) **Scale-Location Plot:**
The plot again resembles a funnel shape. This implies that the variance increases with the expectation, and therefore we detect heteroscedasticity (non-constant variance), as we had before.

(iii) **Residuals vs Fitted:**
This plot shows a very slight curvature which could imply the existence of non-linear relationships.

To summarise the performance of our models so far:

Model	**10-fold CV Error**	**AIC**
OLS	2.713×10^{-5}	-9003.921
Ridge Regression	1.556×10^{-3}	-4283.586
Lasso	2.771×10^{-5}	-9000.305
Elastic Net	2.767×10^{-5}	-8995.06
Heavy Tailed OLS	NA	-10063.73

Table 4: Comparison of models with 10-fold CV Error and AIC.

From this table, it is clear that, based on AIC, the most effective model thus far is the Ordinary Least Squares (OLS) with t-distributed errors.

This concludes the linear regression approach to predicting financial asset returns.

4 Modelling Conditional Volatility: ARCH and GARCH

The first three subsections of this chapter follow the results and steps as found in this book: Analysis of Financial Time Series by Ruey S. Tsay [1].

This chapter covers an important alternative to the multi-predictor Frequentist Regression methods covered in Chapter (2). This approach is called Modelling Conditional Volatility. What differs this approach to Frequentist regression, which directly predicts the asset returns, here we are constructing a univariate model for the time-varying volatility of returns. This means that the models we will introduce in this chapter capture key characteristics of financial returns that the homoskedastic frequentist regressions cannot.

4.1 Introduction to Modelling Conditional Volatility

This subsection follows the results and steps as found in this book: Analysis of Financial Time Series by Ruey S. Tsay [1]

Through this subsection, we start by introducing the concept of volatility and its characteristics, after which we will discuss the concept of stationarity, which will this will motivate our use of volatility model.

4.1.1 Motivating Modelling Conditional Volatility

Definition 19 (Volatility)**.** Volatility is defined as the standard deviation of the financial asset's returns. We will denote the volatility as, σ_t.

The volatility is useful because it has many financial applications, such as:

(i) Volatility plays an important role in portfolio allocation, as we will see in Chapter (5).

(ii) Modelling volatility improves efficiency in parameter estimation and accuracy in interval forecasting.

(iii) Modelling volatility also gives a simpler approach to calculating the Value at Risk (VaR) of an investment.

Now that we have understood the notion of volatility, we discuss some empirical properties of volatility:

(i) **Volatility Clustering:**
The volatility may be high/low over specific time periods.

(ii) **Continuous Evolution:**
The volatility evolves continuously, that is, there are no jump discontinuities in volatility.

(iii) **Finite Variance:**
The volatility is bounded above and below, that is, the volatility is finite.

(iv) **Leverage Effect:**
The volatility is larger for negative past returns as compared to a positive past return of the same magnitude.

In summary, these empirical properties lead us to produce a model in which the current volatility is dependent on the past volatility. This motivates our use of volatility modelling.

4.1.2 Stationarity

Now we will introduce the notion of stationarity. However, we must first discuss Stochastic Processes and their properties.

Definition 20 (Stochastic Process)**.** A Stochastic Process is a collection of random variables, which are indexed by T (time): $\{X_t : t \in T\}$.

Remark: A time series (return series in this case) is simply a realisation of the stochastic process.

Let us now define the statistical properties of a stochastic process.

Definition 21 (Mean and Variance)**.** The mean and variance of a stochastic process are respectively defined as:

$$\mu_t = \mathbb{E}[X_t], \qquad \sigma_t = Var[X_t] \tag{4.1}$$

Definition 22 (Autocovariance)**.** The autocovariance of two stochastic processes is defined as:

$$\gamma(s,t) = Cov[X_s, X_t] = \mathbb{E}[X_s X_t] - \mu_s \mu_t \tag{4.2}$$

Definition 23 (Autocorrelation)**.** The autocorrelation of two stochastic processes is defined as:

$$\rho(s,t) = \frac{\gamma(s,t)}{\sqrt{\gamma(s,s)\gamma(t,t)}} \tag{4.3}$$

Using these definitions, we can now define the notion of stationarity, of which there are two types.

Definition 24 (Strict Stationarity)**.** A stochastic process is strictly stationary if $\forall k \in \mathbb{Z}$, $\forall \{t_1, \ldots, t_k\}$ and $\forall \{C_1, \ldots, C_k\}$ constants, we have:

$$P[X_{t_1} \leq C_1, \ldots, X_{t_k} \leq C_k] = P[X_{t_1+h} \leq C_1, \ldots, X_{t_k+h} \leq C_k], \qquad \forall h = time\ shift \tag{4.4}$$

That is, the probabilistic behaviour of the random variables $\{X_{t_1}, \ldots, X_{t_k}\}$ does not change if the time is shifted by a level h.

Definition 25 (Weak Stationarity)**.** A stochastic process is weakly stationarity if:

(i) The long-run mean is constant: $\mu_t = \mu \qquad \forall \{t \in T\}$

(ii) The autocovariance depends on s and t only via the time difference $|s-t|$:

$$\gamma(s,t) = \gamma(|s-t|, 0) \tag{4.5}$$

The natural question we need to answer now is, why is stationarity important?

Stationarity is important because it guarantees that the mean and auto-covariance are constant over time. This is needed for ARCH and GARCH models, which we will introduce next, because it ensures that the model parameters stay consistent overtime, allowing for valid estimation and forecasting.

4.2 Theoretical Foundations: ARCH and GARCH

This subsection follows the results and steps as found in this book: Analysis of Financial Time Series by Ruey S. Tsay [1]

Through this subsection, we will now introduce and discuss the ARCH and GARCH models, which are models that model conditional volatility.

4.2.1 Autoregressive Conditional Heteroskedasticity Models (ARCH)

We start by defining the ARCH(p) model.

Definition 26 (Autoregressive Conditional Heteroskedasticity Model of order p)**.**
The ARCH(p) model is defined as follows:

$$y_t|Y_{t-1} \sim \mathcal{N}(\mu, \sigma_t^2) \qquad (y_t = \mu + \sigma_t \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0,1)) \tag{4.6}$$

$$with \quad \sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i (y_{t-i} - \mu)^2 \tag{4.7}$$

Remark: In this case μ represents the long-run unconditional mean of the process y_t.

Now, the properties of the ARCH(1) model give some useful insights, so we derive them below.

First, we derive the unconditional variance assuming stationarity and zero long run mean:

$$\begin{aligned} &\mathbb{E}[y_t^2] = \mathbb{E}[\sigma_t^2] = \mathbb{E}[\alpha_0 + \alpha_1 y_{t-1}^2] \quad \implies \quad \mathbb{E}[y_t^2] = \alpha_0 + \alpha_1 \mathbb{E}[y_{t-1}^2] \\ &\implies \quad Var[y_t] = \mathbb{E}[y_t^2] = \frac{\alpha_0}{1-\alpha_1} \geq 0 \quad \implies \quad 0 < \alpha_1 < 1 \quad and \quad \alpha_0 \geq 0 \end{aligned} \tag{4.8}$$

Next, we derive the unconditional fourth moment of y_t, assuming stationarity:

$$\begin{aligned} &\mathbb{E}[y_t^4] = 3 \cdot \mathbb{E}[\sigma_t^4] = 3 \cdot \mathbb{E}\left[(\alpha_0 + \alpha_1 y_{t-1}^2)^2\right] = 3 \cdot (\alpha_0^2 + 2\alpha_0\alpha_1 \mathbb{E}[y_{t-2}^2] + \alpha_1^2 \mathbb{E}[y_t^4]) \\ &\implies \quad \mathbb{E}[y_t^4] = \frac{3 \cdot \alpha_0^2 (1+\alpha_1)}{(1-\alpha_1)(1-3\alpha_1^2)} \geq 0 \quad \implies \quad 0 \leq 3 \cdot \alpha_1^2 < 1 \end{aligned} \tag{4.9}$$

Now using the above, we derive the kurtosis:

$$K(y_t) = \frac{\mathbb{E}[y_t^4]}{(\mathbb{E}[y_t^2])^2} = \frac{3 \cdot \alpha_0^2 (1+\alpha_1)}{(1-\alpha_1)(1-3\alpha_1^2)} \cdot \frac{(1-\alpha_1)^2}{\alpha_0^2} = 3 \cdot \frac{(1-\alpha_1^2)}{1-3\alpha_1^2} \geq 3 \tag{4.10}$$

Therefore, summarising the conditions for stationarity, we have:

(i) Require: $\alpha_0 \geq 0$, $\alpha_1 > 0$ so that $\sigma_t^2 \geq 0$

(ii) If $3 \cdot \alpha_1^2 < 1 \quad \implies \quad y_t^2$ is weakly stationary and in this case y_t is leptokurtic.

These properties hold for higher-order ARCH models; however, the computations become much more complex.

There are, however, some disadvantages of the ARCH model which we list below:

(i) The model assumes that positive and negative returns have the same effects on volatility which, according to the empirical properties of volatility, is incorrect in practice.

(ii) The model is restrictive in the sense that the constraints (i.e. $3\alpha_1^2 < 1$) become very complicated in higher orders. In practice this means that the models ability to capture the heavy tailed behaviour is limited.

(iii) The model is likely to over-predict volatility since it responds slowly to large isolated shocks.

(iv) The model often requires many parameters to adequately describe the volatility of the returns series.

4.2.2 Generalised Autoregressive Conditional Heteroskedasticity Models (GARCH)

The limitations of the ARCH model, specifically the need for higher orders, making the model more complex, and the slow reaction behaviour, motivates the use of the GARCH model, which overcomes these issues.

Now, let us define the GARCH(p, q) model.

Definition 27 (Generalised ARCH Model of order (p, q))**.**
The GARCH(p, q) model is defined as follows:

$$y_t|Y_{t-1} \sim \mathcal{N}(\mu, \sigma_t^2) \qquad (y_t = \mu + \sigma_t\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0,1)) \tag{4.11}$$

$$with \quad \sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i (y_{t-i} - \mu)^2 + \sum_{i=1}^{q} \beta_i \sigma_{t-i}^2 \tag{4.12}$$

Remark: As we had before, μ represents the long-run unconditional mean of the process y_t.

Similar to what we had for the ARCH model, we can derive stationarity conditions for stationarity.

Consider the general GARCH(p, q) model. We can derive the unconditional variance assuming stationarity and zero long run mean:

$$\begin{aligned}
\mathbb{E}[y_t^2] = \mathbb{E}[\sigma_t^2] = \mathbb{E}\left[\alpha_0 + \sum_{i=1}^{p} \alpha_i y_{t-i}^2 + \sum_{i=1}^{q} \beta_i \sigma_{t-i}^2\right] &= \alpha_0 + \mathbb{E}\sum_{i=1}^{p} \alpha_i \mathbb{E}[y_{t-i}^2] + \sum_{i=1}^{q} \beta_i \mathbb{E}[\sigma_{t-i}^2] \\
\implies \quad Var[y_t] = \mathbb{E}[y_t^2] &= \frac{\alpha_0}{1 - \sum_{i=1}^{p} \alpha_i - \sum_{i=1}^{q} \beta_i} \geq 0 \\
\implies \quad 0 < \sum_{i=1}^{p} \alpha_i + \sum_{i=1}^{q} \beta_i < 1 \quad &and \quad \alpha_0 \geq 0
\end{aligned} \tag{4.13}$$

Therefore, summarising the conditions for stationarity, we have:
Require: $\alpha_0 \geq 0$ and $1 > \sum_{i=1}^{p} \alpha_i + \sum_{i=1}^{q} \beta_i$ so that $\sigma_t^2 \geq 0$

4.3 Building the ARCH/GARCH Model

This subsection follows the results and steps as found in this book: Analysis of Financial Time Series by Ruey S. Tsay [1]

Through this subsection, we will describe how to build ARCH/GARCH models which we defined in Section (4.2). We first discuss how to determine the optimal order, then we discuss parameter estimation and finally how we perform diagnostics, all of which are crucial in producing an effective volatility model.

4.3.1 Determining the Order

A natural question after defining the ARCH and GARCH models is, how can we determine the optimal order for our model.

The easiest way to do this is by using Akaike's Information Criterion (AIC), which we introduced in Section (2.1.2). We can do this via the following:

(i) Choose a grid of ARCH lags p (or GARCH lags q).

(ii) Fit each ARCH(p) or GARCH(p, q) model to the data and record each model's AIC.

(iii) Choose the order which has the minimum AIC, and if two models have very similar AIC, we pick the lower order in the interest of parsimony.

Remark: In this setting we use the per-observation Akaike's Information Criterion (AIC), which is defined as:

$$AIC_{per\text{-}obs} = \frac{2 \cdot dim(\boldsymbol{\theta}) - 2\ell(\hat{\boldsymbol{\theta}})}{T} \qquad where \quad T = number\ of\ observations \tag{4.14}$$

4.3.2 Estimation

Now we describe estimating the model parameters of the ARCH(p) and GARCH(p,q) models.

The way we have defined the ARCH and GARCH models makes it clear that Maximum Likelihood Estimation (MLE) is the easiest and most efficient option.

Since for both the ARCH and GARCH models we had:

$$y_t | Y_{t-1} \sim \mathcal{N}(\mu, \sigma_t^2) \tag{4.15}$$

Remark: The difference between the ARCH and GARCH models only lies in the definition of the volatility σ_t^2.

In this case the log-likelihood is defined as:

$$\ell(\boldsymbol{\theta}) = \sum_{t=1}^{n} \{log(f(y_t | \boldsymbol{\theta}; \mathcal{F}_{t-1}))\} \tag{4.16}$$

First compute the likelihood function:

$$f(y_t | Y_{t-1}) = \frac{1}{\sigma_t \sqrt{2\pi}} exp\left\{-\frac{(y_t - \mu)^2}{2\sigma_t^2}\right\} \quad \implies \quad L(\boldsymbol{\theta} | y_t; Y_{t-1}) = \prod_{t=1}^{n} f(y_t | Y_{t-1}) \tag{4.17}$$

Then compute the log-likelihood function:

$$\ell(\boldsymbol{\theta}) = \sum_{t=1}^{n} \left\{-\frac{1}{2} log(\sigma_t^2) - \frac{1}{2} log(2\pi) - \frac{(y_t - \mu)^2}{2\sigma_y^2}\right\} \tag{4.18}$$

Remark: We require initialisation for the initial volatility, σ_1^2, which is often taken to be the long run variance of the returns.

Then we proceed as normal with MLE, finding the values of the parameters $\boldsymbol{\theta}$ which maximise the above log-likelihood:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{argmax} \{\ell(\boldsymbol{\theta})\} \tag{4.19}$$

4.3.3 Diagnostics (Model Checking)

The final part to building a volatility model for a return series is to perform diagnostics which we will cover here.

First we discuss the Probability Integral Transform which provides a general method for checking diagnostics and model fit.

Proposition 2 (The Probability Integral Transform)**.**
Let $X \overset{i.i.d}{\sim} F_X(\cdot)$ be the conditional distribution function of the random variable X.
Then:

$$U = F_X(X) \overset{i.i.d}{\sim} U_n(0,1) \tag{4.20}$$

Proof. First,

$$P[U \leq u] = P[F_X(X) \leq u] = P[X \leq F_X^{-1}(u)] = u$$
$$\implies \quad F_U(u) = P[U \leq u] = u$$

Therefore,

$$U = F_X(X) \overset{i.i.d}{\sim} U_n(0,1)$$

□

In the case of stochastic volatility models, this allows us to define the residuals u_t which we use for diagnostics:

$$Y_t \sim F_{y_t|Y_{t-1}}(\cdot) \quad \implies \quad u_t = F_{y_t|Y_{t-1}}(y_t) \tag{4.21}$$

And more specifically, in the case of ARCH and GARCH models we have:

$$f(y_t|Y_{t-1}) = \frac{1}{\sigma_t\sqrt{2\pi}} exp\left\{-\frac{(y_t-\mu)^2}{2\sigma_t^2}\right\} = \frac{1}{\sigma_t}\Phi\left(\frac{y_t-\mu}{\sigma_t}\right) \tag{4.22}$$

$$\implies F_{y_t|Y_{t-1}}(y_t) = \Phi\left(\frac{y_t-\mu}{\sigma_t}\right) \quad \implies \quad u_t = \Phi\left(\frac{y_t-\mu}{\sigma_t}\right) \tag{4.23}$$

If the model parameters are correct, then: $u_t \overset{i.i.d}{\sim} U_n(0,1)$.

There are two parts to this hypothesis:

(i) **Independence:**
Test for autocorrelation by running regression:

$$(U_t - 0.5)^j = \alpha_1(U_{t-1} - 0.5)^j + \alpha_2(U_{t-2} - 0.5)^j + \cdots + \alpha_p(U_{t-p} - 0.5)^j \tag{4.24}$$

Test: $H_0 : \alpha_1 = \cdots = \alpha_p$ vs $H_1 : otherwise$ $\forall j \in \{1,2,3,4\}$

(ii) **Uniformity:**
We use the Kolmogorov Smirnov Test:
Test: $H_0 : X \overset{i.i.d}{\sim} F_X(\cdot)$ vs $H_1 : X \overset{i.i.d}{\not\sim} F_X(\cdot)$
The test statistic is:

$$D_n = \underset{x}{sup}|F_n(x) - F(x)| \quad with \quad F_n(x) = \frac{1}{n}\sum_{i=1}^{n} I_{(-\infty,x]}(X_i) \tag{4.25}$$

There is also an alternative and much easier way to test this hypothesis. That is, we produce Q-Q plots and we should see that the conditional distribution function resembles an approximate 45° line.

This concludes the theoretical foundations which cover modelling conditional volatility returns.

4.4 Modelling Conditional Volatility: Application

Through this subsection, we aim to apply the modelling conditional volatility methods as discussed in the previous subsections. That is, we fit the ARCH and GARCH models to the stock returns data from the usable data set, which we constructed in Section (1.2).

4.4.1 Model Fitting: ARCH and GARCH

We will start by determining the optimal order of the volatility models, then we make some comments on the optimal-order models, and finally we will derive the AIC of the models and compare them.

ARCH Model:

Our model, takes the form:

$$r_t|R_{t-1} \sim \mathcal{N}(\mu, \sigma_t^2) \qquad with \qquad \sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i (r_{t-i} - \mu)^2 \tag{4.26}$$

Where r_t = `CRSP_SPvw`, represents the S&P 500 monthly stock returns in our data set, and $(\mu, \{\alpha_i\}_{i=0}^{p})$ are the model parameters to be estimated from the data.

We, take $p \in \{1, 2, \dots 12\}$ to be a suitable grid of ARCH orders to fit our models on. The reason behind this is so that we can try to detect annual or near-annual patterns in the variation of volatility of the monthly S&P 500 returns.

Now, we use the `ugarchspec()` [11] and `ugarchfit()` functions, within the `rugarch` package [12], alongside a for loop to fit each ARCH model to our data.

Once each model is fitted, we extract the per-observation AIC from each model, using the `infocriteria()` function. The results are displayed in Table 5, below.

ARCH Order	1	2	3	4	5	6	7	8	9	10	11	12
$\mathbf{AIC}_{\boldsymbol{per\text{-}obs}}$	-3.073	-3.208	-3.245	-3.267	-3.266	-3.281	-3.283	-3.314	-3.334	-3.333	-3.332	-3.330

Table 5: Comparison of the per-observation AIC for each ARCH model of order: $p \in \{1, 2, \dots 12\}$

Thus, we see that the per-observation AIC is minimised for $p = 9$, and we consider this the optimal order on which we fit the ARCH model to our data.

We can extract the model coefficients of the ARCH(9) model, and write the full model below:

$$r_t|R_{t-1} \sim \mathcal{N}(\hat{\mu}, \sigma_t^2) \qquad , \ \hat{\mu} = 0.0107$$

$$\begin{aligned} with \quad \sigma_t^2 = {} & (4.68 \times 10^{-4}) + (0.130) \cdot (r_{t-1} - \hat{\mu})^2 + (0.136) \cdot (r_{t-2} - \hat{\mu})^2 + (0.168) \cdot (r_{t-3} - \hat{\mu})^2 \\ & + (0.0491) \cdot (r_{t-4} - \hat{\mu})^2 + (0.0524) \cdot (r_{t-5} - \hat{\mu})^2 + (0.114) \cdot (r_{t-6} - \hat{\mu})^2 \\ & + (7.93 \times 10^{-3}) \cdot (r_{t-7} - \hat{\mu})^2 + (0.0887) \cdot (r_{t-8} - \hat{\mu})^2 + (0.137) \cdot (r_{t-9} - \hat{\mu})^2 \end{aligned} \tag{4.27}$$

[11] Everything relating to the ARCH and GARCH models, described in this section, has been done in R-script and the code can be found in Appendix F.

Now, we check that this model is stationary. First, all of the coefficient estimates are positive, so this result is satisfied. Next, by generalising the conditions found in the computations in (4.8), we have:

$$\sum_{i=1}^{9} \hat{\alpha}_i = 0.882 < 1 \quad \Longrightarrow \quad \textit{Stationarity} \tag{4.28}$$

Now, we discuss the estimated long-run mean and variance:

(i) **Estimated Unconditional Mean:** $\hat{\mu} = 0.0107$
This tells us that we see an average monthly return of the S&P 500 of about 1.07%.

(ii) **Estimated Unconditional Variance:**
We can calculate this, by generalising equation (4.8), as follows:

$$\widehat{Var[r_t]} = \frac{\hat{\alpha}_0}{1 - \sum_{i=1}^{9} \hat{\alpha}_i} = \frac{0.000468}{1 - 0.882} = 3.97 \times 10^{-3} \tag{4.29}$$

Next, we discuss some observations based on the model coefficients:

(i) Relatively speaking the magnitude of the 9th order coefficient term is large, which tells us that the effects of shocks can last up to 9 months.

(ii) The sum of the first three coefficients is $\sum_{i=1}^{3} \alpha_i = 0.434$, this is also relatively large, which tells us that the last three months returns affects the current volatility the most as compared to any other months.

Finally, we compute the overall model Akaike Information Criteria (AIC)[12], in order to compare our models later on:

$$AIC = T \cdot AIC_{\textit{per-obs}} = 1164 \cdot (-3.334) = -3880.776 \tag{4.30}$$

There is one major issue with this model however, that is the order of the model being nine ,which is quiet high, making the model very complex.

This confirms one of the earlier disadvantages of the ARCH model stated in Section (4.2.1), and thus motivates our use of the GARCH model in hopes of a more parsimonious model.

GARCH Model:

Our model, takes the form:

$$r_t | R_{t-1} \sim \mathcal{N}(\mu, \sigma_t^2) \qquad \textit{with} \qquad \sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i (r_{t-i} - \mu)^2 + \sum_{i=1}^{q} \beta_i \sigma_{t-i}^2 \tag{4.31}$$

Where r_t is as we defined for the ARCH model, and $(\mu, \{\alpha_i\}_{i=0}^{p}, \{\beta_i\}_{i=1}^{q})$ are the model parameters to be estimated from the data.

We, take $p \in \{1, 2, 3\}$ *and* $q \in \{1, 2, 3\}$ to be the suitable grid of GARCH orders to fit our model on. We consider much fewer orders, as compared to the ARCH model, since the GARCH model requires far fewer parameters to capture the returns series behaviour.

Similarly, we fit each GARCH model to the data and extract the per-observation AIC from each model, using the `infocriteria()` function. The results are displayed in Table 6, below.

[12] $AIC = 2 \cdot dim(\boldsymbol{\theta}) - 2\ell(\hat{\boldsymbol{\theta}}) = T \cdot (\frac{2 \cdot dim(\boldsymbol{\theta}) - 2\ell(\hat{\boldsymbol{\theta}})}{T}) = T \cdot AIC_{\textit{per-obs}}$

GARCH Order	$\mathbf{AIC}_{\boldsymbol{per\text{-}obs}}$		
$\boldsymbol{q} \setminus \boldsymbol{p}$	$p=1$	$p=2$	$p=3$
$q=1$	-3.326	-3.326	-3.325
$q=2$	-3.325	-3.324	-3.323
$q=3$	-3.323	-3.323	-3.328

Table 6: Comparison of the per-observation AIC for each GARCH model of order: $(p,q) \in \{1,2,3\} \times \{1,2,3\}$.

We can see that the per-observation AIC is minimised for order $(p,q) = (3,3)$. However, the difference is marginal as compared to order $(p,q) = (1,1)$ model (difference of 0.002), so in interest of parsimony this is the optimal order on which we fit the GARCH model to our data.

We can extract the model coefficients of the GARCH(1,1) model, and write the full model below:

$$r_t | R_{t-1} \sim \mathcal{N}(\hat{\mu}, \sigma_t^2) \qquad , \ \hat{\mu} = 0.0106$$

$$with \quad \sigma_t^2 = (6.80 \times 10^{-5}) + (0.141) \cdot (r_{t-1} - \hat{\mu})^2 + (0.840) \cdot \sigma_{t-1}^2 \tag{4.32}$$

Now, we check that this model is stationary. First, all of the coefficient estimates are positive, so this result is satisfied.

$$\hat{\alpha}_1 + \hat{\beta}_1 = 0.981 < 1 \quad \Longrightarrow \quad Stationary \tag{4.33}$$

Now, we discuss the estimated long-run mean and variance:

(i) **Estimated Unconditional Mean:** $\hat{\mu} = 0.0106$
This tells me that we see an average monthly return of the S&P 500 of about 1.06%, which is very close the ARCH result.

(ii) **Estimated Unconditional Variance:**
We can calculate this, by generalising equation (4.8), as follows:

$$\widehat{Var[r_t]} = \frac{\hat{\alpha}_0}{1 - \hat{\alpha}_1 - \hat{\beta}_1} = \frac{0.000068}{1 - 0.981} = 3.58 \times 10^{-3} \tag{4.34}$$

The key observation we can make based on the model coefficients is that the GARCH coefficient is very large, $\beta_1 = 0.840$, as compared to the ARCH term, $\alpha_1 = 0.141$. This tells us that the current month's volatility is largely affected by the previous month's volatility.

Finally, we compute the overall model Akaike Information Criteria (AIC):

$$AIC = T \cdot AIC_{per\text{-}obs} = 1164 \cdot (-3.328) = -3873.792 \tag{4.35}$$

4.4.2 Model Checking

Now that we have fitted our models, we may begin the model checking step to verify whether or not our models satisfy the normality assumption, as presented in Subsection (2.2).

We first generate the residuals via the probability integral transform as follows:

$$r_t | R_{t-1} \sim \mathcal{N}(\mu, \sigma_t^2) \quad \Longrightarrow \quad u_t = \Phi\left(\frac{y_t - \hat{\mu}}{\sigma_t}\right) \tag{4.36}$$

If the model parameters are correct, then: $u_t \overset{i.i.d}{\sim} U_n(0,1)$.

Then for the ARCH(9) and GARCH(1,1) models, the easiest way to check this is to create a Q-Q plot. If the residuals are uniformly distributed then we should see an approximate 45° line.

The plots are displayed below in Figure 7.

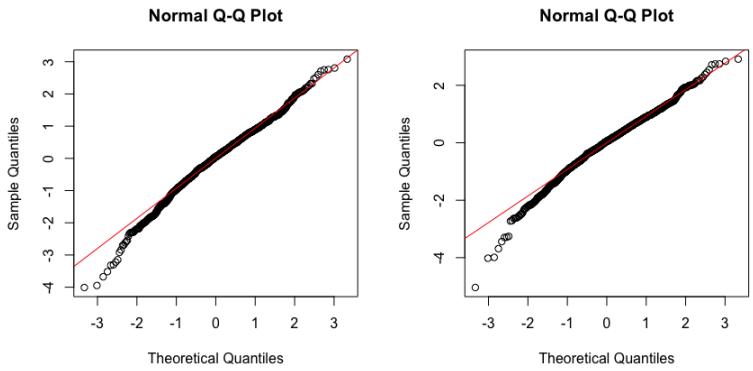

Figure 7: This figure shows the Q-Q Residuals plots for the following models: ARCH(9) Model (left), GARCH(1,1) Model(right).

We make the following observations:

(i) **ARCH(9) Q-Q Plot:**
There is slight deviation in the upper tail but clear heavy deviation in the lower tail, this implies that there are issues in assuming the conditional distribution of the returns is normal. However, the middle alignment with the line is very strong.

(ii) **GARCH(1,1) Q-Q Plot:**
We make similar observations to the ARCH(9) model, but we can see how the GARCH model has tried to pull things closer to the line in the lower tail. Thus, despite some remaining lower tail deviation, the GARCH model yields a better overall fit.

Therefore we can see that there are issues with the distributional assumption of the ARCH and GARCH models.

4.4.3 ARCH and GARCH Models using the Student's t-distribution

Naturally, we look for way to improve the Q-Q plots, mirroring the work done in Chapter (3.2).

We found that due to the leptokurtic nature of the S&P 500 monthly returns, slightly altering the normality of errors assumption and changing to a t-distribution yielded huge success in both predictive performance and satisfying the model assumptions. Similarly, we can alter the conditional distribution of the returns to take on a t-distribution in order to better capture the heavy tailed nature of the returns data.

More formally, we write:

Definition 28 (t-distributed ARCH/GARCH Model)**.**
The t-distributed ARCH/GARCH model is defined as follows:

$$y_t = \mu + \sigma_t \epsilon_t, \quad \epsilon_t \sim t_n \quad with \qquad \sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i (y_{t-i} - \mu)^2 + \sum_{i=1}^{q} \beta_i \sigma_{t-i}^2 \tag{4.37}$$

$$where\ in\ the\ case\ of\ ARCH,\ we\ take\ \beta_i = 0 \quad \forall i \in \{1, 2, \ldots, q\}\,. \tag{4.38}$$

Through this section, we will refit the t-distributed ARCH and GARCH models to our data and make some observations.

t-distributed ARCH Model:

Our model, takes the form:

$$r_t = \mu + \sigma_t \epsilon_t, \quad \epsilon_t \sim t_n \quad with \qquad \sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i (r_{t-i} - \mu)^2 \tag{4.39}$$

Again, for the reasons previously stated in Section (4.4.1), we take $p \in \{1, 2, \ldots, 12\}$ as the grid of ARCH lags on which we fit our models on.

Similarly, we fit each ARCH model to the data and we extract the per-observation AIC from each model, using the `infocriteria()` [13]function. The results are displayed in Table 7, below.

ARCH Order	1	2	3	4	5	6	7	8	9	10	11	12
$\mathbf{AIC}_{\boldsymbol{per\text{-}obs}}$	-3.261	-3.293	-3.319	-3.324	-3.324	-3.332	-3.333	-3.342	-3.350	-3.349	-3.347	-3.346

Table 7: Comparison of the per-observation AIC for each t-distributed ARCH model of order: $p \in \{1, 2, \ldots 12\}$

Again, we find that $p = 9$ is the optimal order on which we we fit the ARCH model to our data.

We can extract the t-distributed ARCH(9) model coefficients, and we write the full model below:

$$r_t = \hat{\mu} + \sigma_t \epsilon_t \quad , \epsilon_t \sim t_n \qquad , \hat{\mu} = 0.0117 \quad and \quad \hat{n} = 9.127$$

$$\begin{aligned} with \quad \sigma_t^2 = {} & (5.57 \times 10^{-4}) + (0.140) \cdot (r_{t-1} - \hat{\mu})^2 + (0.127) \cdot (r_{t-2} - \hat{\mu})^2 + (0.176) \cdot (r_{t-3} - \hat{\mu})^2 \\ & + (0.0330) \cdot (r_{t-4} - \hat{\mu})^2 + (0.0561) \cdot (r_{t-5} - \hat{\mu})^2 + (0.0993) \cdot (r_{t-6} - \hat{\mu})^2 \\ & + (7.39 \times 10^{-10}) \cdot (r_{t-7} - \hat{\mu})^2 + (0.0801) \cdot (r_{t-8} - \hat{\mu})^2 + (0.117) \cdot (r_{t-9} - \hat{\mu})^2 \end{aligned} \tag{4.40}$$

We check that this model is stationary: $\sum_{i=1}^{9} \hat{\alpha}_i = 0.829 < 1 \quad \Longrightarrow \quad Stationarity$

Now, we discuss the estimated long-run mean and variance:

(i) **Estimated Unconditional Mean:** $\hat{\mu} = 0.0117$
This tells us that we see an average monthly return of the S&P 500 of about 1.17%.

(ii) **Estimated Unconditional Variance:** $\widehat{Var[r_t]} = 3.26 \times 10^{-3}$

The observations on the model coefficients are identical to what we had before.

And we compute the overall model Akaike Information Criteria (AIC):

$$AIC = T \cdot AIC_{per\text{-}obs} = 1164 \cdot (-3.350) = -3899.40 \tag{4.41}$$

In this case the t-distributed ARCH(9) produced a model with a significantly lower AIC than that of it's normaly distributed counterpart.

t-distributed GARCH Model:

Our model, takes the form:

$$r_t = \mu + \sigma_t \epsilon_t, \quad \epsilon_t \sim t_n \quad with \qquad \sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i (r_{t-i} - \mu)^2 + \sum_{i=1}^{q} \beta_i \sigma_{t-i}^2 \tag{4.42}$$

As before, we take $p \in \{1, 2, 3\} \quad and \quad q \in \{1, 2, 3\}$ as the suitable be the grid of GARCH lags to fit our models on.

Similarly, we fit each GARCH model to the data and we extract the per-observation AIC from each model, using the `infocriteria()` function. The results are displayed in Table 8, below.

GARCH Order	**AIC$_{per\text{-}obs}$**		
$\boldsymbol{q} \setminus \boldsymbol{p}$	$p = 1$	$p = 2$	$p = 3$
$q = 1$	-3.351	-3.350	-3.348
$q = 2$	-3.350	-3.348	-3.347
$q = 3$	-3.348	-3.347	-3.350

Table 8: Comparison of the per-observation AIC for each t-distributed GARCH model of order: $(p, q) \in \{1, 2, 3\} \times \{1, 2, 3\}$.

Thus, we can see that the per-observation AIC is minimised for $(p, q) = (1, 1)$, so this the optimal order on which we fit the GARCH model to our data.

Now, we can extract the model coefficients, and we write the full model below:

$$r_t = \hat{\mu} + \sigma_t \epsilon_t \quad , \epsilon_t \sim t_n \qquad , \ \hat{\mu} = 0.0118 \quad and \quad \hat{n} = 8.08$$

$$with \quad \sigma_t^2 = (9.24 \times 10^{-5}) + (0.142) \cdot (r_{t-1} - \hat{\mu})^2 + (0.827) \cdot \sigma_{t-1}^2 \tag{4.43}$$

We check that this model is stationary: $\hat{\alpha}_1 + \hat{\beta}_1 = 0.969 < 1 \quad \implies \quad Stationary$

Now, we discuss the estimated long-run mean and variance:

(i) **Estimated Unconditional Mean:** $\hat{\mu} = 0.0118$
This tells me that we see an average monthly return of the S&P 500 of about 1.18%, which is very close the ARCH result.

(ii) **Estimated Unconditional Variance:** $\widehat{Var[r_t]} = 0.00295$

The observations on the model coefficients are identical to what we had before.

And we compute the overall model Akaike Information Criteria (AIC):

$$AIC = T \cdot AIC_{per\text{-}obs} = 1164 \cdot (-3.351) = -3900.564 \tag{4.44}$$

Now, we produce two Q-Q plots to recheck the uniformity of the residuals:

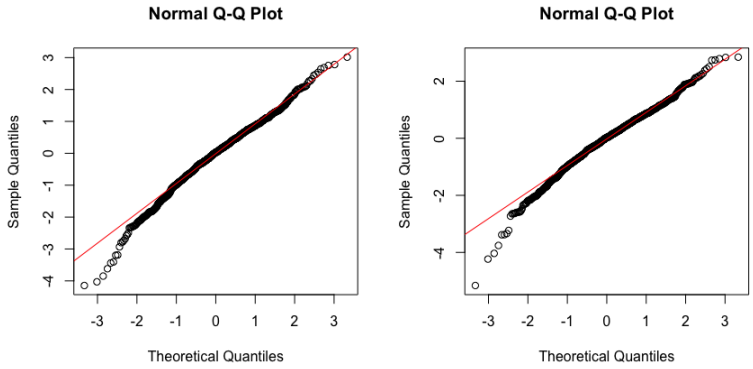

Figure 8: This figure shows the Q-Q Residuals plots for the following models: t-distributed ARCH(9) Model (left), t-distributed GARCH(1,1) Model(right).

We make the following observations:

(i) **t-distributed ARCH(9) Q-Q Plot:**
There is still a slight deviation in the upper tail but the lower tail deviation is slightly improved but still prominent, this implies that using the t-distribution has improved the model. Again, we observe that the middle alignment with the line is very strong.

(ii) **t-distributed GARCH(1,1) Q-Q Plot:**
Again, we make similar observations to the GARCH model and we see that the GARCH model has tried to pull things closer to the line in the lower tail.

Thus, using the t-distribution, instead of the normal distribution, to model volatility has improved the model's fit.

Finally, we summarise the performance of our models so far:

Model	**AIC**
ARCH(9)	-3880.776
GARCH(1,1)	-3873.792
Heavy ARCH(9)	-3899.40
Heavy GARCH(1,1)	-3900.564

Table 9: Comparison of conditional volatility models overall AIC.

From this table, it is clear that both t-distributed ARCH/GARCH have outperformed the normally distributed ARCH/GARCH in terms of predictive performance.

Furthermore, we can see that the most effective model thus far is the t-distributed GARCH(1,1) model.

This concludes the modelling conditional volatility approach to predicting financial asset returns.

4.5 Frequentist Regression vs Volatility Modelling

Through this section, we will discuss how the frequentist regression approach compares to the conditional volatility modelling approach when modelling the monthly returns of the S&P 500 index, and we will attempt a final improvement on the models created so far.

We, will start by summarising the performance of all of our models so far, and discuss our observations. Table 10, below, displays the model evaluation metrics of all of the models thus far:

Model	**10-fold CV Error**	**AIC**
OLS	2.713×10^{-5}	-9003.921
Ridge Regression	1.556×10^{-3}	-4283.586
Lasso	2.771×10^{-5}	-9000.305
Elastic Net	2.767×10^{-5}	-8995.06
Heavy Tailed OLS	NA	-10063.73
ARCH(9)	NA	-3880.776
GARCH(1,1)	NA	-3873.792
Heavy ARCH(9)	NA	-3899.40
Heavy GARCH(1,1)	NA	-3900.564

Table 10: Comparison of all models of the monthly S&P500 returns with 10-fold CV Error and AIC.

These results tell me that all of the frequentist regression models have far outperformed any of the conditional volatility models by up to 2.6× in the most extreme case.

This is easily explained by the way we have constructed each approach.

Firstly, we recall that the ARCH/GARCH models are univariate models, relying solely on the S&P500 returns to predict the following month's volatility.

In contrast, frequentist regression has access to a wide array of macroeconomic indicators (dividend yield, term spread, default spread, etc.), many of which were found to be highly influential, which we used as explanatory variables. This means the frequentist regression models have access to a far richer set of information on which to model the returns on, as compared to the ARCH/GARCH models which are completely blind to these highly influential macroeconomic indicators.

Therefore, frequentist regression on its own will almost always outperform the conditional volatility modelling methods described so far.

4.5.1 Hybrid Predictive Regression and t-distributed GARCH model

These conclusions lead to seeking a model which has both the advantage of having access to a rich set of information on which to model the returns on, while also having the ability to model the conditional volatility. This type of model would be taking the best of both worlds in a sense and we will look to create such a model now.

One way to do this would be to take another look formulation of the t-distributed GARCH model (4.42). In this case we assumed that the mean term was a constant and estimated it from the data. We can instead assume that this mean term is not constant, and even better than that we take it to be the mean comprised of the macroeconomic indicators as the variables in Multiple Linear Regression from that we used in Chapter 3.

This upgraded model takes the form:

$$r_t = \mu_t + \sigma_t \epsilon_t, \quad \epsilon_t \sim t_n \quad with \qquad \sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i (r_{t-i} - \mu_{t-i})^2 + \sum_{i=1}^{q} \beta_i \sigma_{t-i}^2 \tag{4.45}$$

Where the conditional mean is itself a regression:

$$\mu_t = \beta_0 + \beta_1 \cdot (\texttt{d/p})_t + \beta_2 \cdot (\texttt{d/y})_t + \beta_3 \cdot (\texttt{e/p})_t + \beta_4 \cdot (\texttt{d/e})_t + \beta_5 \cdot (\texttt{svar})_t + \beta_6 \cdot (\texttt{b/m})_t + \beta_7 \cdot (\texttt{ntis})_t$$
$$+ \beta_8 \cdot (\texttt{tbl})_t + \beta_9 \cdot (\texttt{lty})_t + \beta_{10} \cdot (\texttt{ltr})_t + \beta_{11} \cdot (\texttt{tms})_t + \beta_{12} \cdot (\texttt{dfy})_t + \beta_{13} \cdot (\texttt{dfr})_t + \beta_{14} \cdot (\texttt{infl})_t$$

This model has access to a rich set of macroeconomic indicators whilst also being able to model the conditional volatility and even further it uses a t-distribution, allowing it to capture the heavy tailed nature of the returns.

Now let us fit this model, determine the optimal orders and see if it improves the AIC any further.

Remark: In order to prevent singularities, I omit the Dividend-Earning ratio *(d.e)* and the Term-Spread *(tms)* variables from our model. Thus we must again be critical of our model.

We take $p \in \{1, 2, 3\} \quad and \quad q \in \{1, 2, 3\}$ to be the grid of orders on which we fit our models on. We fit each model to the data and extract the per-observation AIC. Table 11 displays the results:

GARCH Order	**AIC$_{per\text{-}obs}$**		
$\boldsymbol{q} \setminus \boldsymbol{p}$	$p = 1$	$p = 2$	$p = 3$
$q = 1$	-8.860	-8.856	-8.936
$q = 2$	-8.866	-8.865	-8.936
$q = 3$	-8.868	-8.866	-9.033

Table 11: Comparison of the per-observation AIC for each t-distributed GARCH regression model of order: $(p, q) \in \{1, 2, 3\} \times \{1, 2, 3\}$.

Thus, we can see that the per-observation AIC is minimised for orders $(p, q) = (3, 3)$, and so this is the optimal order on which we fit the hybrid regression GARCH model to our data.

Now, we extract the jointly estimated model coefficients and write the full model below:

$$r_t = \hat{\mu}_t + \sigma_t \epsilon_t \quad , \epsilon_t \sim t_n \qquad , where \quad \hat{n} = 3.799$$
$$with \quad \sigma_t^2 = (0) + (0.0127) \cdot (r_{t-1} - \hat{\mu}_{t-1})^2 + (9.99 \times 10^{-3}) \cdot (r_{t-2} - \hat{\mu}_{t-2})^2$$
$$+ (0.287) \cdot (r_{t-3} - \hat{\mu}_{t-3})^2 + (0) \cdot \sigma_{t-1}^2 + (0) \cdot \sigma_{t-2}^2 + (0.690) \cdot \sigma_{t-3}^2 \tag{4.46}$$

$$\hat{\mu}_t = (9.94 \times 10^{-3}) + (-1.01) \cdot \texttt{d/p} + (1.01) \cdot \texttt{d/y} + (7.17 \times 10^{-4}) \cdot \texttt{e/p} + (0) \cdot \texttt{d/e}$$
$$+ (0.295) \cdot \texttt{svar} + (-2.27 \times 10^{-3}) \cdot \texttt{b/m} + (4.53 \times 10^{-3}) \cdot \texttt{ntis}$$
$$+ (-2.60 \times 10^{-3}) \cdot \texttt{tbl} + (4.33 \times 10^{-3}) \cdot \texttt{lty} + (6.07 \times 10^{-4}) \cdot \texttt{ltr} + (0) \cdot \texttt{tms}$$
$$+ (7.41 \times 10^{-2}) \cdot \texttt{dfy} + (4.48 \times 10^{-3}) \cdot \texttt{dfr} + (4.65 \times 10^{-3}) \cdot \texttt{infl} \tag{4.47}$$

We make the following observations from the summary output:

(i) **T-tests:** Test: $H_0 : \beta_i = 0$ vs $H_1 : \beta_i \neq 0$ $\forall i \in \{1, 2, ..., 14\}$
$p - value < 0.05 \implies$ reject the null at the $\alpha = 0.05$ significant level for $i \in \{1, 2, 3, 5, 12\}$. This tells us that there is overwhelming evidence that the variables; Dividend-Price Ratio *(d/p)*, Dividend-Yield *(d/y)*, Earning-Price ratio *(e/p)*, Stock Volatility *(svar)* and Default Yield *(dfy)*, are related to the S&P500 stock returns *(CRSP SPvw)*.

(ii) **Stationarity:**

$$\sum_{i=1}^{3} \hat{\alpha}_i + \sum_{i=1}^{3} \hat{\beta}_i = 0.99969 < 1 \quad \implies \quad stationary \tag{4.48}$$

We now compute the overall model Akaike Information Criteria (AIC):

$$AIC = T \cdot AIC_{per\text{-}obs} = 1164 \cdot (-9.033) = -10514.412 \tag{4.49}$$

This AIC is significantly lower than even our best model, the t-distributed OLS model which had an AIC of -10063.73.

This shows us that this model surpasses all of the other models in terms of in-sample predictive performance.

This confirms our speculations and shows that by combining the frequentist regression (OLS) model and t-distributed GARCH model, we have been able to produce a model which far surpasses every model created thus far.

This concludes the predicting financial asset returns part of this project.

5 Portfolio Allocation

The first two parts of this chapter follow the results and steps as found in the book: Mathematics for Finance: An Introduction to Financial Engineering by Capiński, Marek, and Tomasz Zastawniak [6].

This chapter covers the theory behind portfolio allocation, after which we will use the models constructed so far to build portfolios based on the following month's returns.

5.1 Introduction to Portfolio Management

Through this subsection, we will introduce the key concepts and tools used to construct a portfolio of two assets.

Throughout this section, we will only consider the case where we work in a single-period model.

Recall: Using the definition of the One-Period Simple Return, as defined in Section (1.1), we now have: $R = \frac{P_1 - P_0}{P_0}$

Now, we will formally define the notion of a portfolio weight.

Definition 29 (Portfolio Weights)**.** The weights of two assets, asset A and asset B, in a portfolio with initial wealth V_0, are defined as:

$$w_A = \frac{x_A \cdot A_0}{V_0} \quad and \quad w_B = \frac{x_B \cdot B_0}{V_0} \tag{5.1}$$

,where w denotes the weight of the asset in the portfolio and x denotes the position of the asset.

Properties of the portfolio weights:

(i) The weights w_i can be either positive, negative or zero and if $w_i < 0 \implies$ short selling.

(ii) The amount invested in asset i $= x_i \cdot P_0 = w_i \cdot V_0$

(iii) The initial value of the portfolio is given by: $V_0 = x_A \cdot A_0 + x_B \cdot B_0$

(iv) We will always have: $w_A + w_B = \frac{x_A \cdot A_0}{V_0} + \frac{x_B \cdot B_0}{V_0} = 1$

Now we state a theorem which links the One-Period Simple Return of a portfolio with its weights and individual asset values.

Proposition 3. For any portfolio with two assets, we have:

$$R_v = w_A \cdot R_A + w_B \cdot R_B \tag{5.2}$$

Proof. First, we know that:

$$R_V = \frac{V_1 - V_0}{V_0} \quad , R_A = \frac{A_1 - A_0}{A_0} \quad and \quad R_B = \frac{B_1 - B_0}{B_0} \tag{5.3}$$

Then:

$$V_1 = x_A \cdot A_1 + x_B \cdot B_1 = x_A \cdot [A_0(1 + R_A)] + x_B \cdot [B_0(1 + R_B)] \tag{5.4}$$

This implies that:

$$V_1 - V_0 = (x_A \cdot A_0(1 + R_A) + x_B \cdot B_0(1 + R_B)) - (x_A \cdot A_0 + x_B \cdot B_0) \tag{5.5}$$

$$= x_A A_0 \cdot R_A + x_B B_1 \cdot R_B \tag{5.6}$$

Thus,

$$R_V = \frac{V_1 - V_0}{V_0} \quad = \quad \frac{x_A A_0}{V_0} \cdot R_A + \frac{x_B B_0}{V_0} \cdot R_B \quad = \quad w_A \cdot R_A + w_B \cdot R_B \tag{5.7}$$

□

5.1.1 The Expected Return and Risk of a Portfolio

Let us now define the expected return, risk and correlation of an asset.

Definition 30 (Expected Return)**.** The expected return of asset i is defined as:

$$\mu_i = \mathbb{E}[R_i] \tag{5.8}$$

Definition 31 (Risk)**.** The risk of asset i is defined as the standard deviation of the return, that is:

$$\sigma_i = \sqrt{Var[R_i]} \tag{5.9}$$

Definition 32 (Correlation Coefficient)**.** The correlation coefficient of asset i and asset j is defined as:

$$\rho_{i,j} = \frac{Cov[R_i, R_j]}{\sigma_i \cdot \sigma_j} \tag{5.10}$$

Now that we have defined the expected return and risk of an asset, we may now define the expected return and risk of a portfolio of two assets.

Theorem 1. For a portfolio made up two assets, asset A and asset B, we have:

(i) The Expected Return of the portfolio:

$$\mu_V = w_A \cdot \mu_A + w_B \cdot \mu_B \tag{5.11}$$

(ii) The Risk of the portfolio:

$$\sigma_V^2 = w_A^2 \cdot \sigma_A^2 + w_B^2 \cdot \sigma_B^2 + 2 w_A w_B \cdot \rho_{A,B} \cdot \sigma_A \sigma_B \tag{5.12}$$

Proof. First, for the expected return we have:

$$\begin{aligned} \mu_V = \mathbb{E}[R_V] = \mathbb{E}[w_A \cdot R_A + w_B \cdot R_B] &= w_A \cdot \mathbb{E}[R_A] + w_B \cdot \mathbb{E}[R_B] \\ &= w_A \cdot \mu_A + w_B \cdot \mu_B \end{aligned} \tag{5.13}$$

Similarly, for the risk we have:

$$\begin{aligned} \sigma_V^2 = Var[R_V] &= \mathbb{E}[(w_A \cdot R_A + w_B \cdot R_B)^2] - (\mathbb{E}[w_A \cdot R_A + w_B \cdot R_B])^2 \\ &= w_A^2 \cdot \mathbb{E}[R_A^2] + 2 w_A w_B \cdot \mathbb{E}[R_A R_B] + w_B^2 \cdot \mathbb{E}[R_B^2] \\ &\qquad - \left(w_A^2 \cdot (\mathbb{E}[R_A])^2 + 2 w_A w_B \cdot \mathbb{E}[R_A R_B] + w_B^2 \cdot (\mathbb{E}[R_B])^2 \right) \\ &= w_A^2 \cdot \sigma_A^2 + w_B^2 \cdot \sigma_B^2 + 2 w_A w_B \cdot \rho_{A,B} \cdot \sigma_A \sigma_B \end{aligned} \tag{5.14}$$

□

5.1.2 Diversification

Now we will show how diversification can be used to reduce the risk associated to a portfolio.

Proposition 4. Assume that short selling is not allowed in a portfolio of, assets A and B.

Then, we have:

$$\sigma_V \leq max\left\{\sigma_A, \sigma_B\right\} \tag{5.15}$$

Proof. Since short selling is banned: $\implies$ $w_A, w_B \geq 0$

Now, assume that $\sigma_A > \sigma_B$. Then:

$$\begin{aligned}\sigma_V^2 &\leq w_A^2 \cdot \sigma_A^2 + w_B^2 \cdot \sigma_B^2 + 2w_A w_B \cdot \sigma_A \sigma_B \\ &= (w_A \cdot \sigma_A + w_B \cdot \sigma_B)^2 \leq (w_A \cdot \sigma_A + w_B \cdot \sigma_A)^2 \\ &= \sigma_A^2 \cdot (w_A + w_B)^2 = \sigma_A^2 = (max\left\{\sigma_A, \sigma_B\right\})^2\end{aligned} \tag{5.16}$$

□

From this proposition, we can see that we can decrease the overall risk by investing in two assets instead of just one. This justifies the idea of constructing a portfolio with a lower-return asset to help reduce the overall risk.

5.2 Minimum Variance Portfolio given a Fixed level of return

Our goal now is to find the weights of the portfolio which has the minimum possible risk for a fixed, desired, level of return.

Theorem 2. Let asset A and asset B have expected returns μ_A, μ_B (where $\mu_A > \mu_B$) and risk σ_A, σ_B respectively. Furthermore, fix a level of return μ_v.
Then, the minimum variance portfolio, given the expected return μ_v, has weights:

$$w_A^{\mu_v} = \frac{\mu_v - \mu_B}{\mu_A - \mu_B} \qquad and \qquad w_B^{\mu_v} = \frac{\mu_A - \mu_v}{\mu_A - \mu_B} \tag{5.17}$$

Furthermore, the square of the risk of this portfolio is given by:

$$\sigma_v^2 = \frac{(\mu_v - \mu_B)^2\sigma_A^2 + (\mu_A - \mu_v)^2\sigma_B^2 + 2\sigma_A\sigma_B \cdot \rho_{A,B} \cdot (\mu_v - \mu_B)(\mu_A - \mu_v)}{(\mu_A - \mu_B)^2} \tag{5.18}$$

Proof. The aim is to minimise the risk, which is the same as minimising the variance.
Therefore we can formulate the optimisation problem as follows:

$$\begin{aligned}&\min_{(w_1,w_2)} \left\{\sigma_V^2\right\} \qquad subject\ to \qquad \begin{cases} w_A + w_B = 1, \\ \mu_A \cdot w_A + \mu_B \cdot w_B = \mu_v \end{cases} \\ &\qquad where \quad \sigma_V^2 = w_A^2 \cdot \sigma_A^2 + w_B^2 \cdot \sigma_B^2 + 2w_A w_B \cdot \rho_{A,B} \cdot \sigma_A \sigma_B\end{aligned} \tag{5.19}$$

The Lagrangian can be written as:

$$\mathcal{L}(w_A, w_B, \lambda_1, \lambda_2) = \sigma_V^2 + \lambda_1 \cdot (1 - w_A - w_B) + \lambda_2 \cdot (\mu_v - \mu_A \cdot w_A - \mu_B \cdot w_B) \tag{5.20}$$

Then:

$$(1) \quad \frac{\partial \mathcal{L}}{\partial w_A} = 2w_A \cdot \sigma_A^2 + 2w_B \cdot \rho_{A,B} \cdot \sigma_A \sigma_B - \lambda_1 - \lambda_2 \mu_A = 0 \tag{5.21}$$

$$(2) \quad \frac{\partial \mathcal{L}}{\partial w_B} = 2w_B \cdot \sigma_B^2 + 2w_A \cdot \rho_{A,B} \cdot \sigma_A \sigma_B - \lambda_1 - \lambda_2 \mu_B = 0 \tag{5.22}$$

$$(3) \quad \frac{\partial \mathcal{L}}{\partial \lambda_1} = 1 - w_A - w_B = 0 \quad \implies \quad w_A + w_B = 1 \tag{5.23}$$

$$(4) \quad \frac{\partial \mathcal{L}}{\partial \lambda_2} = \mu_v - \mu_A \cdot w_A - \mu_B \cdot w_B = 0 \quad \implies \quad \mu_A \cdot w_A + \mu_B \cdot w_B = \mu_v \tag{5.24}$$

Solving equations (3) and (4) simultaneously:

$$\mu_A \cdot w_A + \mu_B \cdot (1 - w_A) = \mu_v \qquad \Longrightarrow \qquad (\mu_A - \mu_B) \cdot w_A = \mu_v - \mu_B \tag{5.25}$$
$$\Longrightarrow \qquad w_A^{\mu_v} = \frac{\mu_v - \mu_B}{\mu_A - \mu_B} \qquad \Longrightarrow \qquad w_B^{\mu_v} = 1 - w_A = \frac{\mu_A - \mu_v}{\mu_A - \mu_B}$$

Finally, by substituting these weights back into formula (5.19), we find:

$$\sigma_v^2 = \frac{(\mu_v - \mu_B)^2\sigma_A^2 + (\mu_A - \mu_v)^2\sigma_B^2 + 2\sigma_A\sigma_B \cdot \rho_{A,B} \cdot (\mu_v - \mu_B)(\mu_A - \mu_v)}{(\mu_A - \mu_B)^2} \tag{5.26}$$

□

This concludes the theory covering portfolio allocation, which we will now use in the context of the models we have constructed so far.

5.3 Portfolio Allocation: Application

Through this subsection, we will produce forecasts based on the various models produced, for use in the construction of optimal portfolios, given a fixed level of return, of the S&P500 index with a risk-free asset.

We will first reduce the usable data set, to contain only data starting from the beginning of 1927 until the end of 2022. We will then refit the respective models on to this reduced data set and produce one-month ahead forecasts (expected return, risk, and 95% confidence interval)for January 2023.

Finally, we will use these predictions, together with the U.S. Treasury bill as the risk-free asset, to construct minimum variance portfolios under the method outlined in Section (5.2).

5.3.1 Refitting Models

First we will begin by refitting each model on the reduced data set, and reporting the model coefficients.

Remark: We will only fit the following four models:

(i) ***Elastic Net* :** This encapsulates all of the shrinkage models produced (Combines the ℓ_1 and ℓ_2 penalties).

(ii) **Heavy OLS:** This was the best model we could produce via frequentist regression.

(iii) **t-distributed GARCH(1,1):** This was the best model produced via modelling conditional volatility.

(iv) **Hybrid Regression and t-distributed GARCH(3,3):** This was the best model produced overall.

Going forward we will take the expected return to be the predicted next-step return and we take the risk to be the standard deviation (volatility) of this prediction.

Now, we begin fitting each model.

***Elastic Net Model* :**

We use the `glmnet()` [14]function to fit our model at optimal tuning parameter values and the coefficient estimates produced are below:

$$\begin{aligned}\widehat{\texttt{CRSP\_SPvw}} = (8.60\times10^{-3}) &+ (-0.979)\cdot\texttt{d/p} + (1.00)\cdot\texttt{d/y} + (-2.11\times10^{-2})\cdot\texttt{e/p}\\ &+ (-2.23\times10^{-2})\cdot\texttt{d/e} + (0.373)\cdot\texttt{svar} + (2.33\times10^{-3})\cdot\texttt{b/m} + (0)\cdot\texttt{ntis}\\ &+ (-4.13\times10^{-4})\cdot\texttt{tbl} + (-9.83\times10^{-3})\cdot\texttt{lty} + (2.11\times10^{-3})\cdot\texttt{ltr} + (0)\cdot\texttt{tms}\\ &+ (9.89\times10^{-2})\cdot\texttt{dfy} + (2.59\times10^{-2})\cdot\texttt{dfr} + (0)\cdot\texttt{infl} \qquad (5.27)\end{aligned}$$

Remark: In order to compute the 95% confidence interval, we required the use of bootstrapping[15].

Heavy Tailed OLS:

We use the `heavyLm()` function within the `heavy` package, [11], in R to fit the model with t-distributed errors, without specifying the degrees of freedom. The Expectation Maximisation (EM) algorithm[16]produces an estimate of the degrees of freedom to be $\hat{n} = 1.82$ degrees of freedom.

The coefficient estimates are shown below:

$$\begin{aligned}\widehat{\texttt{CRSP\_SPvw}} = (9.30\times10^{-3}) &+ (-1.00)\cdot\texttt{d/p} + (1.00)\cdot\texttt{d/y} + (5.00\times10^{-4})\cdot\texttt{e/p} + (0)\cdot\texttt{d/e}\\ &+ (1.32\times10^{-1})\cdot\texttt{svar} + (-6.00\times10^{-4})\cdot\texttt{b/m} + (-1.10\times10^{-3})\cdot\texttt{ntis}\\ &+ (-3.30\times10^{-3})\cdot\texttt{tbl} + (2.80\times10^{-3})\cdot\texttt{lty} + (0)\cdot\texttt{ltr} + (0)\cdot\texttt{tms}\\ &+ (6.57\times10^{-2})\cdot\texttt{dfy} + (-4.80\times10^{-3})\cdot\texttt{dfr} + (1.58\times10^{-2})\cdot\texttt{infl} \qquad (5.28)\end{aligned}$$

t-distributed GARCH(1,1):

We use the `ugarchspec()` and `ugarchfit()` functions, within the `rugarch` package [12], to fit the model and the model coefficients produced are below:

$$\begin{aligned}r_t &= \hat{\mu} + \sigma_t\epsilon_t \quad ,\epsilon_t \sim t_n \qquad ,\ \hat{\mu} = 0.0117 \quad and \quad \hat{n} = 8.02\\ with \quad \sigma_t^2 &= (9.14\times10^{-5}) + (0.143)\cdot(r_{t-1}-\hat{\mu})^2 + (0.826)\cdot\sigma_{t-1}^2 \qquad (5.29)\end{aligned}$$

Hybrid Regression and t-distributed GARCH(3,3):

Again, we use the `ugarchspec()` and `ugarchfit()` functions, within the `rugarch` package [12], to fit the model and the model coefficients produced are below:

$$\begin{aligned}r_t &= \hat{\mu}_t + \sigma_t\epsilon_t \quad ,\epsilon_t \sim t_n \qquad ,where \quad \hat{n} = 3.818\\ with \quad \sigma_t^2 &= (0) + (0.0130)\cdot(r_{t-1}-\hat{\mu}_{t-1})^2 + (0.0102)\cdot(r_{t-2}-\hat{\mu}_{t-2})^2\\ &\quad + (0.289)\cdot(r_{t-3}-\hat{\mu}_{t-3})^2 + (0)\cdot\sigma_{t-1}^2 + (0)\cdot\sigma_{t-2}^2 + (0.687)\cdot\sigma_{t-3}^2 \qquad (5.30)\end{aligned}$$

$$\begin{aligned}\hat{\mu}_t = (0.0101) &+ (-1.01)\cdot\texttt{d/p} + (1.01)\cdot\texttt{d/y} + (7.02\times10^{-4})\cdot\texttt{e/p} + (0)\cdot\texttt{d/e}\\ &+ (0.294)\cdot\texttt{svar} + (-2.26\times10^{-3})\cdot\texttt{b/m} + (4.75\times10^{-3})\cdot\texttt{ntis}\\ &+ (-4.50\times10^{-3})\cdot\texttt{tbl} + (6.29\times10^{-3})\cdot\texttt{lty} + (3.67\times10^{-4})\cdot\texttt{ltr} + (0)\cdot\texttt{tms}\\ &+ (7.08\times10^{-2})\cdot\texttt{dfy} + (3.72\times10^{-3})\cdot\texttt{dfr} + (3.38\times10^{-3})\cdot\texttt{infl} \qquad (5.31)\end{aligned}$$

[14]Everything relating to refitting of the models has been done in R-script and the code can is in Appendix H.
[15]More information on bootstrapping can be found in reference [7]
[16]More information on the Expectation Maximisation (EM) algorithm can be found in reference [5]

To summarise the key findings from these models, Table 12, displayed the AIC, expected return, risk and 95% confidence interval for each model:

Model	**AIC**	**Expected Return**	**Risk**	**95% CI**
Elastic Net	-8896.198	0.0628	6.57×10^{-4}	$[0.0615, 0.0641]$
Heavy Tailed OLS	-9944.326	0.0686	3.09×10^{-6}	$[0.06859, 0.06861]$
Heavy GARCH(1,1)	-3860.813	0.0117	0.0655	$[-0.139, 0.163]$
Hybrid Regression + GARCH(3,3)	-10391.962	0.0624	2.13×10^{-3}	$[0.0564, 0.0685]$

Table 12: Comparison of all the models, which have been refitted on the reduced data set, AIC, Expected Return and Risk and 95% Confidence Interval

We make the following observations based on this table.

First, let us discuss the AIC. Based purely on AIC, which tells us about the in-sample fit, we have that the hybrid regression + GARCH(3,3) model is the best model and the heavy GARCH(1,1) model is the worst.

Now, let's look into the expected return, risk, and confidence intervals of each model, given that the true return for January 2023 is: $\mu_{r_t} = 0.0638$.

***Elastic Net Model* :**

(i) **Expected Return:** Very close to the true return, negligible bias.

(ii) **Risk:** The estimated risk is extremely low because the model treats the variance as constant, thus not capturing any of the volatility clustering.

(iii) **CI:** Very narrow interval, indicating high over-confidence although it captures the true value.

Heavy Tailed OLS:

(i) **Expected Return:** Slightly biased estimate, over predicts the return value.

(ii) **Risk:** The risk is approximately zero since all of volatility behaviour is absorbed by the heavy tails. Extreme underestimate of the true risk.

(iii) **CI:** Extremely narrow confidence interval, indicating unreasonably high over-confidence. Furthermore, the interval doesn't even capture the true value of the return.

t-distributed GARCH(1,1):

(i) **Expected Return:** Very poor and biased estimate of the return, vastly under predicts the return value. This is due to only modelling volatility dynamics with a constant mean.

(ii) **Risk:** Unreasonably large risk, most likely due to the slow decay of shocks ($\alpha_1 + \beta_1 = 0.969$).

(iii) **CI:** Unreasonably wide confidence interval, which provides very little information for practical use.

Hybrid Regression and t-distributed GARCH(3,3):

(i) **Expected Return:** Also extremely close to the true return, negligible bias.

(ii) **Risk:** Reasonable risk, since it captures the volatility clustering.

(iii) **CI:** Reasonably wide confidence interval, which also captures the true return value.

Remark: The frequentist interpretation of a 95% confidence interval is that as the number independent of samples grows to infinity the proportion of confidence intervals which contain the true value of the return, tends to 95%. This is an issue because in our case, we are only working with one sample of the S&P 500 monthly returns (impossible to get any more). Therefore these intervals should be viewed as indication of relative precision instead of precise probability statements.

5.3.2 Construction of the Portfolios

Now that we have the expected return and risk for each model, we now construct a portfolio of the S&P500 with the risk free asset in hopes of reducing the overall risk, as shown in Section (5.1.2).

We must first fix a level of return that we aim for the portfolio to achieve, on which we will apply Theorem 2. We take the target return to be $\mu_v = 0.04$.

Finally, the return and risk of the risk free asset (taken to be the U.S. Treasury Bills), are:
$\mu_{rfree} = 0.008$ and $\sigma_{rfree} = 0$.

Now we apply Theorem 2, to give the portfolio weights alongside the associated portfolio risks. Table 13 displays the results.

Model	**Weights:** (w_A, w_B)	**Expected Return:** μ_v	**Risk of Portfolio:** σ_v
Elastic Net	(0.584,0.416)	0.04	3.84×10^{-4}
Heavy Tailed OLS	(0.528,0.472)	0.04	1.63×10^{-6}
Heavy GARCH(1,1)	(8.65,-7.65)	0.04	0.567
Hybrid Regression + GARCH(3,3)	(0.588,0.412)	0.04	1.25×10^{-3}

Table 13: Comparison of the minimum variance portfolios, given the target expected risk of 0.04, for each model

Between these four models, the hybrid regression GARCH(3,3) model produces the most balanced and best portfolio. It reaches the 0.04 target return, incorporates influential macroeconomic indicators, incorporates volatility clustering, and produces a confidence interval that reflects real-world uncertainty.

In contrast, the heavy OLS and *elastic net* models underestimate the risk, while the t-distributed GARCH(1,1) model overestimates the risk and greatly underestimates the expected return.

Furthermore, we observe that the heavy GARCH(1,1) model takes a short position in the risk-free asst (U.S. Treasury Bills). In practice, this type of allocation is unwise and exposes the portfolio to unnecessary risk.

Therefore, from this work, I conclude that for practical and reliable monthly expected return, risk forecasts and appropriate portfolio allocation, the hybrid regression GARCH(3,3) model is the best model.

This concludes the portfolio allocation section of this project.

6 Conclusion/Summary

We now summarise our key findings, after which we will discuss some limitations and possible directions to further this work.

Key Findings:

Firstly, during the frequent regression chapter, we found that the use of t-distributed errors in the OLS yielded a model whose in-sample performance outperformed the standard OLS and all shrinkage models. Although this model failed to perform well in expected return and risk forecasting.

Then, during the volatility modelling chapter, we saw how the use of t-distributed GARCH outperformed not only its ARCH counterpart but also the standard ARCH/GARCH models.

Then the biggest improvement we could find was the introduction of the hybrid regression GARCH(3,3) model, which yielded the most balanced and best performing model not only in regards to in-sample tests, but also in regards to expected return and risk forecasts.

Limitations:

(i) **Over Fitting Risk:**
We focused mainly on in-sample performance metrics (AIC), which is an issue because it can lead to over fitting and bias our forecasts. Therefore, we would need to also check the out of sample performance metrics to get a truly robust indication of the out of sample performance of our models.

(ii) **Leverage Effect:**
With regard to the volatility models, a key limitation is that we never accounted for the leverage effect, which may decrease forecast accuracy.

Future Directions:

With regards to general improvements or directions to could further this work, what stuck out the most was **Bayesian regression**. At almost every turn, the use Bayesian regression seemed to be a possible improvement, whether it be the possibility of modelling heavy tailed regression combined with shrinkage, which I was unable to do in the frequentist approach, or the issue of using confidence intervals in Section (5.3.1) which could be solved via the use of credible intervals instead. That is to say that, the use of Bayesian regression to incorporate a prior distribution to each parameter seems like a possible direction go in.

Furthermore, with respects to the volatility modelling and hybrid regression GARCH model, a possible direction to go in would be look into utilising **EGARCH** models instead, to capture the leverage effect and even possibly adding **ARMA(p,q)** terms to further improve the model.

Final Comments:

Overall, our hybrid regression GARCH framework provides a statistically strong and economically meaningful model for forecasting and portfolio allocation of the S&P 500 index, but there is definitely room for possible improvements.

This concludes my work on predicting financial asset returns and portfolio allocation.

Acknowledgements

Firstly, I would like to express my deepest gratitude to my family, especially my parents, for their unwavering support and encouragement throughout my studies.

I am also very grateful to Dr Maria Kalli for her exceptional supervision. The invaluable guidance, insightful feedback and patience in reading through my draft has been instrumental in shaping this project.

I also wish to thank my friends, for their continuous moral support and for making this academic journey much more enjoyable.

Finally, I would like to include a quote from the great Sir Isaac Newton as a sign of respect to the countless mathematicians and statisticians who put immeasurable time and effort into the development of the theories in this project:

> *"If I have seen further, it is by standing on the shoulders of giants".*

Appendix A

```
#1. Cleaning The Data
library(readxl)
returns <- read_excel("PredictorData2023.xlsx", col_names = TRUE)
returns[returns == "NaN"] <- NA
returns_1 <- returns[ , !(names(returns) %in% c("csp"))]
x <- max(which(!complete.cases(returns_1)))
returns_2 <- returns_1[(x + 1):nrow(returns_1), ]
returns_cleaned <- data.frame(lapply(returns_2, function(x) as.numeric(as.character(x))))

#2. Creating a usable data set:
library(dplyr)
data_new <- returns_cleaned %>%
mutate(
    d.p = log(D12) - log(Index),
    d.y = log(D12) - log(lag(Index)),
    e.p = log(E12) - log(Index),
    d.e = log(D12) - log(E12),
    tms = lty - tbl,
    dfy = BAA - AAA,
    dfr = corpr - ltr)
data_final <- data_new %>%
    select(yyyymm, d.p, d.y, e.p, d.e, svar, b.m, ntis, tbl, lty, ltr, tms, dfy, dfr,
    infl, CRSP_SPvw)
data_final <- data_final[-1, ]

#Empirical Density Plot:
install.packages("ggplot2")
library(ggplot2)
set.seed(123)
empirical_density <- density(data_final$CRSP_SPvw)
mean_return <- mean(data_final$CRSP_SPvw)
sd_return <- sd(data_final$CRSP_SPvw)
xs <- seq(min(data_final$CRSP_SPvw), max(data_final$CRSP_SPvw), length.out = 1000)
normal_density <- dnorm(xs, mean = mean_return, sd = sd_return)
plot(empirical_density,
    main = "Empirical Density",
    xlab = "Returns",
    ylab = "Density",
    lwd = 2)
lines(xs, normal_density,lwd=2, lty=2)
legend("topright",
    legend = c("Empirical Density", "Normal Density"),
    lwd = 2, lty = c(1,2))
```

Appendix B

```
#OLS Model
ModelFull <- lm(CRSP_SPvw ~ d.p+d.y+e.p+d.e+svar+b.m+ntis+tbl+lty+ltr+tms+dfy
                                                    +dfr+infl, data = data_final)
summary(ModelFull)

library(caret)

#OLS Model Evaluation Metrics
set.seed(1)
train.control <- trainControl(method = "cv", number = 10)
model.cv <- train(CRSP_SPvw ~ d.p + d.y + e.p + d.e + svar + b.m + ntis + tbl + lty
                  + ltr + tms + dfy + dfr + infl,
                  data = data_final,
                  method = "lm",
                  trControl = train.control)
OLS.cv <- mean((model.cv$resample$RMSE)^2)  # Convert RMSE to MSE
print(OLS.cv)
AIC(ModelFull)

#Ridge Regression Model Fitting
install.packages("glmnet")
library(glmnet)

data <- subset(data_final, select = -yyyymm)
x <- model.matrix(CRSP_SPvw ~ ., data)[, -1]
y <- data_final$CRSP_SPvw
ridge.model = glmnet(x,y,alpha=0)
set.seed (1)
cv.ridge= cv.glmnet(x,y,alpha=0)
plot(cv.ridge)
bestlam.ridge = cv.ridge$lambda.min
bestlam.ridge
predict(ridge.model,s=bestlam.ridge,type= "coefficients")

#Ridge Regression Model Evaluation Metrics
ridge.mse <- min(cv.ridge$cvm)  #Stores the MSE value
ridge.mse

fitted.ridge <- predict(ridge.model, newx=x, s=bestlam.ridge)
rss.ridge = sum((data_final$CRSP_SPvw - fitted.ridge)^2)
coefficients.ridge = predict(ridge.model, s = bestlam.ridge, type = "coefficients")
df.ridge = sum(coefficients.ridge != 0) - 1
```

```
n = length(data_final$CRSP_SPvw)
log_likelihood.ridge = -n / 2 * (log(2 * pi * rss.ridge / n) + 1)

aic.ridge = -2 * log_likelihood.ridge + 2 * df.ridge
aic.ridge

#Lasso Model Fitting
lasso.model = glmnet(x,y,alpha=1)

set.seed (1)
cv.lasso = cv.glmnet(x,y,alpha=1)
plot(cv.lasso)
bestlam.lasso = cv.lasso$lambda.min
bestlam.lasso

predict(lasso.model,s=bestlam.lasso,type= "coefficients")

#Lasso Model Evaluation Metrics
lasso.mse <- min(cv.lasso$cvm)
lasso.mse

fitted.lasso <- predict(lasso.model, newx=x, s=bestlam.lasso)
rss.lasso = sum((data_final$CRSP_SPvw - fitted.lasso)^2)
coefficients.lasso = predict(lasso.model, s = bestlam.lasso, type = "coefficients")
df.lasso = sum(coefficients.lasso != 0) - 1

log_likelihood.lasso = -n / 2 * (log(2 * pi * rss.lasso / n) + 1)

aic.lasso = -2 * log_likelihood.lasso + 2 * df.lasso
aic.lasso

#Elastic Net Model Fitting
alphas <- seq(0,1, by = 0.0001)
results <- data.frame(Alpha = numeric(), Lambda = numeric(), MSE = numeric())

for (a in alphas) {
  set.seed(1)
  cv.elasticnet = cv.glmnet(x,y,alpha=a)
  bestlam.elasticnet = cv.elasticnet$lambda.min
  best_mse <- min(cv.elasticnet$cvm)
  results <- rbind(results, data.frame(Alpha = a, Lambda = bestlam.elasticnet
                                                          , MSE = best_mse))
}
best.combination <- results[which.min(results$MSE), ]
best.combination
```

```
elasticnet.model = glmnet(x,y,alpha=best.combination$Alpha)
predict(elasticnet.model,s=best.combination$Lambda,type= "coefficients")
fitted.elasticnet <- predict(elasticnet.model, newx=x, s=bestlam.elasticnet)

#Elastic Net Model Evaluation
elasticnet.mse <- best.combination$MSE
elasticnet.mse

rss.elasticnet = sum((data_final$CRSP_SPvw - fitted.elasticnet)^2)
coefficients.elasticnet = predict(elasticnet.model, s = bestlam.elasticnet
                                                          , type = "coefficients")
df.elasticnet = sum(coefficients.elasticnet != 0) - 1
log_likelihood.elasticnet = -n / 2 * (log(2 * pi * rss.elasticnet / n) + 1)

aic.elasticnet = -2 * log_likelihood.elasticnet + 2 * df.elasticnet
aic.elasticnet
```

Appendix C

```
#Box-Cox Transformation
library(MASS)

data_boxcox <- data_final
data_boxcox$CRSP_SPvw <-data_boxcox$CRSP_SPvw + 1

ModelFull2 <- lm(CRSP_SPvw ~ d.p+d.y+e.p+d.e+svar+b.m+ntis+tbl+lty+ltr+tms
                                              +dfy+dfr+infl, data = data_boxcox)

boxcox_result <- boxcox(ModelFull2)
lambda <- boxcox_result$x[which.max(boxcox_result$y)]
lambda

ModelBox.Cox <- lm(log(CRSP_SPvw) ~ d.p+d.y+e.p+d.e+svar+b.m+ntis+tbl+lty+ltr+tms
                                                 +dfy+dfr+infl, data = data_boxcox)

par(mfrow=c(1,3))
plot(ModelBox.Cox, which = c(1,2,3), main="ModelBox.Cox")
```

Appendix D

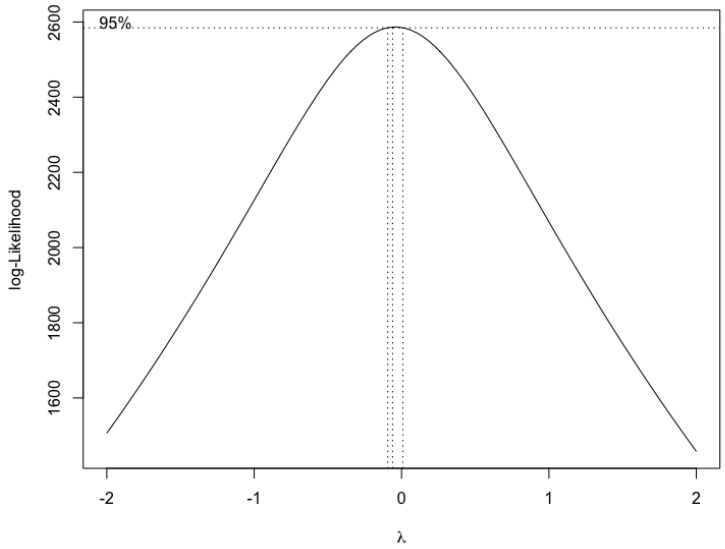

Figure 9: This figure shows the log-Likelihood plotted against the values of λ.

Appendix E

```
#MLE of t-distribution d.f.
library(MASS)

set.seed(123)
data <- data_final$CRSP_SPvw
fit <- fitdistr(data, "t",start = list(m=mean(data),s=sd(data),df=5), lower=c(-1,0.001,1))
print(fit)

#Heavy Tailed Model
library(heavy)

heavy.model <- heavyLm(CRSP_SPvw ~ d.p+d.y+e.p+svar+b.m+ntis+tbl+lty+ltr+dfy+dfr
+infl, data = data_final, family = Student(3.68), control = heavy.control(fix.shape = TRUE))
summary(heavy.model)
AIC.heavy <- 2*14 - 2*as.numeric(5014.441)
print(AIC.heavy)

heavy.model.2 <- heavyLm(CRSP_SPvw ~ d.p+d.y+e.p+svar+b.m+ntis+tbl+lty+ltr
+dfy+dfr+infl, data = data_final)
summary(heavy.model.2)
AIC.heavy.2 <- 2*14 - 2*as.numeric(5045.865)
print(AIC.heavy.2)
```

Appendix F

```
#TimeSeries Data Set
library(tseries)
returns_ts <- ts(data_final$CRSP_SPvw, start = c(1927, 1), frequency = 12)

#ARCH Models
library(rugarch)
arch_models <- list()
for (i in 1:12) {
  spec <- ugarchspec(
    variance.model = list(model = "sGARCH", garchOrder = c(i, 0)),
    mean.model     = list(armaOrder = c(0, 0), include.mean = TRUE),
    distribution.model = "norm"
  )
  fit <- ugarchfit(spec = spec, data = returns_ts)
  arch_models[[i]] <- fit
  aic_value <- infocriteria(fit)[1]
  cat("ARCH order:", i, "AIC:", aic_value, "\n")
}
coef(arch_models[[9]])


#GARCH Models
garch_models <- list()
model_info <- data.frame(p = integer(), q = integer(), AIC = numeric(),
                                                    stringsAsFactors = FALSE)
counter <- 1
for(p in 1:3) {
  for(q in 1:3) {
    spec <- ugarchspec(
      variance.model = list(model = "sGARCH", garchOrder = c(p, q)),
      mean.model     = list(armaOrder = c(0, 0), include.mean = TRUE),
      distribution.model = "norm"
    )
    fit <- ugarchfit(spec = spec, data = returns_ts)
    aic_value <- infocriteria(fit)[1]
    model_info[counter,] <- c(p, q, aic_value)
    garch_models[[counter]] <- fit
    cat("GARCH order: (", p, ",", q, ") - AIC:", aic_value, "\n")
    counter <- counter + 1
  }
}
garch33_coefs <- coef(garch_models[[1]])
print(garch33_coefs)
```

```
#Model Checking
fit9 <- arch_models[[9]]
u_t_fit9 <- residuals(fit9, standardize=TRUE)
qqnorm(u_t_fit9); qqline(u_t_fit9, col="red")

fit1_1 <- garch_models[[1]]
u_t_fit1_1 <- residuals(fit1_1, standardize=TRUE)
qqnorm(u_t_fit1_1); qqline(u_t_fit1_1, col="red")
```

Appendix G

```
#Heavy ARCH Models
library(rugarch)
heavy_arch_models <- list()
for (i in 1:12) {
  spec <- ugarchspec(
    variance.model = list(model = "sGARCH", garchOrder = c(i, 0)),
    mean.model     = list(armaOrder = c(0, 0), include.mean = TRUE),
    distribution.model = "std"
  )
  fit <- ugarchfit(spec = spec, data = returns_ts)
  heavy_arch_models[[i]] <- fit
  heavy_aic_value <- infocriteria(fit)[1]
  cat("Heavy ARCH order:", i, "AIC:", heavy_aic_value, "\n")
}
coef(heavy_arch_models[[9]])

#Heavy GARCH Models
heavy_garch_models <- list()
heavy_model_info <- data.frame(p = integer(), q = integer(), AIC = numeric()
                                                          , stringsAsFactors = TRUE)
heavy_counter <- 1
for(p in 1:3) {
  for(q in 1:3) {
    spec <- ugarchspec(
      variance.model = list(model = "sGARCH", garchOrder = c(p, q)),
      mean.model     = list(armaOrder = c(0, 0), include.mean = TRUE),
      distribution.model = "std"
    )
    fit <- ugarchfit(spec = spec, data = returns_ts)
    heavy_aic_value <- infocriteria(fit)[1]
    heavy_model_info[heavy_counter,] <- c(p, q, heavy_aic_value)
    heavy_garch_models[[heavy_counter]] <- fit
```

```
    cat("Heavy GARCH order: (", p, ",", q, ") - AIC:", heavy_aic_value, "\n")
    heavy_counter <- heavy_counter + 1
  }
}
heavy_garch11_coefs <- coef(heavy_garch_models[[1]])
print(heavy_garch11_coefs)

#Heavy Model Checking
heavy_fit9 <- heavy_arch_models[[9]]
u_t_heavy_fit9 <- residuals(heavy_fit9, standardize=TRUE)
qqnorm(u_t_heavy_fit9); qqline(u_t_heavy_fit9, col="red")

heavy_fit1_1 <- heavy_garch_models[[1]]
u_t_heavy_fit1_1 <- residuals(heavy_fit1_1, standardize=TRUE)
qqnorm(u_t_heavy_fit1_1); qqline(u_t_heavy_fit1_1, col="red")

#Heavy OLS+GARCH
X2 <- as.matrix(data_final[, c("d.p","d.y","e.p",
                               "svar","b.m","ntis",
                               "tbl","lty","ltr",
                               "dfy","dfr","infl")])
heavy_OLS_garch_models <- list()
heavy_model_info2 <- data.frame(p = integer(), q = integer(), AIC = numeric(), stringsAsFactors =

heavy_counter <- 1
for(p in 1:3) {
  for(q in 1:3) {
    spec2 <- ugarchspec(
      variance.model = list(model      = "sGARCH", garchOrder = c(p,q)),
      mean.model     = list(armaOrder = c(0,0), include.mean = TRUE, external.regressors = X2),
      distribution.model = "std"
    )
    fit <- ugarchfit(spec = spec2, data = data_final$CRSP_SPvw)
    heavy_aic_value <- infocriteria(fit)[1]
    heavy_model_info2[heavy_counter,] <- c(p, q, heavy_aic_value)
    heavy_OLS_garch_models[[heavy_counter]] <- fit
    cat("Heavy OLS+GARCH order: (", p, ",", q, ") - AIC:", heavy_aic_value, "\n")
    heavy_counter <- heavy_counter + 1
  }
}
heavy_fit14.11 <- heavy_OLS_garch_models[[9]]
show(fit14.11)
```

Appendix H

```
#Reducing the data set
data_pa <- data_final[1:1152, ]
data_pa2 <- subset(data_pa, select = -yyyymm)
x2 <- model.matrix(CRSP_SPvw ~ ., data_pa2)[, -1]
y2 <- data_pa2$CRSP_SPvw

newdata <- data_final[1153,]
newdata <- subset(newdata, select = -yyyymm)
x.new <- model.matrix(CRSP_SPvw ~ ., newdata)[, -1]

#Model Prediction CI Frequentest Regression
elasticnet.model2 = glmnet(x2,y2,alpha=best.combination$Alpha)
predict(elasticnet.model2,s=best.combination$Lambda,type= "coefficients")
library(boot)
boot_predict_elasticnet <- function(data, indices) {
  d <- data[indices, ]
  x_boot <- model.matrix(CRSP_SPvw ~ ., d)[,-1]
  y_boot <- d$CRSP_SPvw
  model <- glmnet(x_boot, y_boot, alpha = best.combination$Alpha)
  pred <- predict(model, newx = x.new, s = best.combination$Lambda)
  return(pred)
}
set.seed(123)
boot_results <- boot(data = data_pa2, statistic = boot_predict_elasticnet, R = 1000)

predictions.elasticnet <- boot.ci(boot_results, type = "perc")
predictions.elasticnet
forecast_return <- as.numeric(predict(elasticnet.model2, newx = x.new, s = best.combination$Lambda
forecast_return
forecast_se <- sd(boot_results$t)

n <- length(y2)
fitted.elasticnet2 <- as.numeric(
  predict(elasticnet.model2,
          newx = x2,
          s    = best.combination$Lambda,
          type = "response")
)
rss.elasticnet2 <- sum((y2 - fitted.elasticnet2)^2)
coefficients.elasticnet2 <- as.numeric(
  predict(elasticnet.model2,
          s    = best.combination$Lambda,
          type = "coefficients")
```

```
)
df.elasticnet2 <- sum(coefficients.elasticnet2 != 0) - 1
log_likelihood.elasticnet2 <- -n/2 * (log(2 * pi * rss.elasticnet2 / n) + 1)
aic.elasticnet2 <- -2 * log_likelihood.elasticnet2 + 2 * df.elasticnet2
aic.elasticnet2

#Model Prediction CI Heavy Time Series
returns_ts_pa <- ts(data_pa$CRSP_SPvw, start = c(1927, 1), frequency = 12)

spec <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model     = list(armaOrder = c(0, 0), include.mean = TRUE),
  distribution.model = "std"
)
garch11.3 <- ugarchfit(spec = spec, data = returns_ts_pa)
garch11.3_forecast <- ugarchforecast(garch11.3, n.ahead = 1)

forecast_mean <- fitted(garch11.3_forecast)
forecast_mean
forecast_sigma <- sigma(garch11.3_forecast)
forecast_sigma

model_coef <- coef(garch11.3)
show(model_coef)

t_multiplier <- qt(0.975, df = 8.02)
lower_bound <- forecast_mean - t_multiplier * forecast_sigma
upper_bound <- forecast_mean + t_multiplier * forecast_sigma

#Model Prediction Heavy OLS+GARCH
X3 <- as.matrix(data_pa[, c("d.p","d.y","e.p",
                            "svar","b.m","ntis",
                            "tbl","lty","ltr",
                            "dfy","dfr","infl")])
spec3 <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(3,3)),
  mean.model     = list(armaOrder = c(0,0), include.mean = TRUE, external.regressors = X3),
  distribution.model = "std"
)
heavy_OLS_garch14_11.2 <- ugarchfit(spec = spec3, data = data_pa$CRSP_SPvw)

new_row <- data_final[1153, ]
X3_next <- as.matrix(new_row[, colnames(X3), drop = FALSE])

garch14_11_forecast <- ugarchforecast(
```

```
  fit                = heavy_OLS_garch14_11.2,
  n.ahead            = 1,
  external.forecasts = list(mregfor = X3_next)
)
forecast_mean2 <- fitted(garch14_11_forecast)
forecast_mean2
forecast_sigma2 <- sigma(garch14_11_forecast)
forecast_sigma2

model_coef2 <- coef(heavy_OLS_garch14_11.2)
show(heavy_OLS_garch14_11.2)

t_multiplier <- qt(0.975, df = 3.817657)
lower_bound2 <- forecast_mean2 - t_multiplier * forecast_sigma2
upper_bound2 <- forecast_mean2 + t_multiplier * forecast_sigma2
```

References

[1] Tsay, Ruey S. 2010. Analysis of Financial Time Series. 3rd ed. Hoboken, New Jersey: Wiley & Sons, Inc. https://www.wiley.com/en-gb/Analysis+of+Financial+Time+Series%2C+3rd+Edition-p-9780470414354#tableofcontents-section.

[2] Welch, Ivo, and Amit Goyal. *"A Comprehensive Look at the Empirical Performance of Equity Premium Prediction."* The Review of Financial Studies 21, no. 4 (2008): 1455-1508. Accessed November 25, 2024. https://doi.org/10.1093/rfs/hhm014.

[3] James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2017. *An Introduction to Statistical Learning.* 1st ed. New York: Springer. https://doi.org/10.1007/978-1-4614-7138-7.

[4] Zou, Hui, and Trevor Hastie. *"Regularization and Variable Selection Via the Elastic Net."* Journal of the Royal Statistical Society Series B: Statistical Methodology 67, no. 2 (2005): 301–320. Accessed January 4, 2025. https://doi.org/10.1111/j.1467-9868.2005.00503.x.

[5] Dempster, A. P., N. M. Laird, and D. B. Rubin. *"Maximum Likelihood from Incomplete Data Via the EM Algorithm."* Journal of the Royal Statistical Society: Series B (Methodological) 39, no. 1 (1977): 1-22. Accessed January 11, 2025. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x.

[6] Capiński, Marek, and Tomasz Zastawniak. 2003. *Mathematics for Finance: An Introduction to Financial Engineering.* 1st ed. Springer London. https://doi.org/10.1007/b97511.

[7] Efron, Bradley, and Ryan J. Tibshirani. 1994. An Introduction to the Bootstrap. 1st ed. Chapman and Hall/CRC. https://doi.org/10.1201/9780429246593.

[8] Venables WN, Ripley BD (2002). Modern Applied Statistics with S, Fourth edition. Springer, New York. ISBN 0-387-95457-0, https://www.stats.ox.ac.uk/pub/MASS4/.

[9] Friedman J, Tibshirani R, Hastie T (2010). "Regularization Paths for Generalized Linear Models via Coordinate Descent." Journal of Statistical Software, 33(1), 1–22. doi:10.18637/jss.v033.i01.

[10] Kuhn, Max (2008). "Building Predictive Models in R Using the caret Package." Journal of Statistical Software, 28(5), 1–26. doi:10.18637/jss.v028.i05, https://www.jstatsoft.org/index.php/jss/article/view/v028i05.

[11] Osorio, F. (2019). heavy: Robust estimation using heavy-tailed distributions. R package version 0.38.196. URL: CRAN.R-project.org/package=heavy

[12] Galanos A (2024). rugarch: Univariate GARCH models.. R package version 1.5-3.