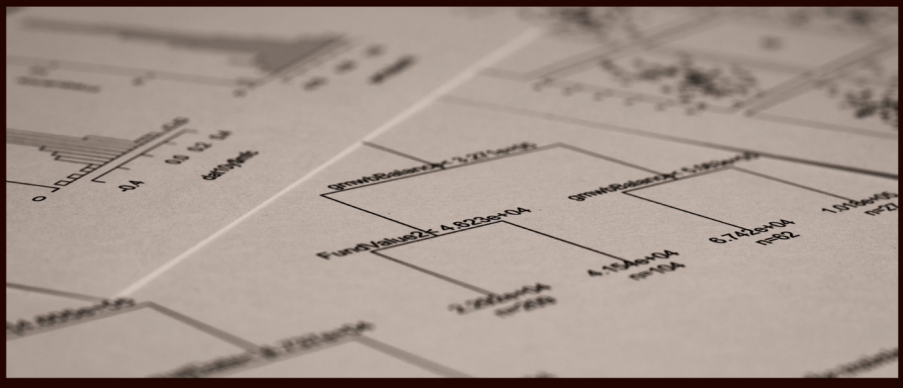
Actuarial Statistics with R

Theory and Case Studies



Guojun Gan, PhD, FSA Emiliano A. Valdez, PhD, FSA

ACTEX Learning

"A great introduction to statistical modeling for the working actuary or actuarial science college student—alternating between theory and application, it helps actuaries and aspiring actuaries round out their technical toolkit."

– Mary Pat Campbell, FSA, MAAA

This book covers several topics on data analysis and statistical learning prescribed by the International Actuarial Association (IAA). In particular, it has been designed to cover the learning objectives for the SOA's Statistics for Risk Modeling (SRM) Exam. Many materials from this book also cover parts of the syllabus for the CAS Modern Actuarial Statistics (MAS-I and MAS-II) Exam. It is broadly intended for students and practitioners to learn R programming and its applications in actuarial science, finance, and quantitative risk management.

This book is uniquely designed as a single source to cover traditional and modern methods of data analytics. It teaches the steps of how to implement and validate models in R at an elementary level and gives students and practitioners the opportunity to experience the power of applied statistics and R programming first-hand with real world problems. In addition, it provides a theoretical framework, but also utilizes the case study method to connect theory and practice and bridge the gap between academia and industry.

Guojun Gan, PhD, FSA, is an Assistant Professor in the Department of Mathematics at the University of Connecticut, Storrs, CT. He received a BS degree from Jilin University, Changchun, China, in 2001 and MS and PhD degrees from York University, Toronto, Canada, in 2003 and 2007, respectively. His research interests include data mining and actuarial science. He has published several books and research papers on a variety of topics.

Emiliano A. Valdez, PhD, FSA, is a Professor in the Department of Mathematics at the University of Connecticut, Storrs, CT. His academic experience includes several years of teaching and pursuing research in three different continents: North America, Australia, and Asia. Dr. Valdez is renowned for his work on copula models and has been awarded the Edward A. Lew Award, the Halmstad Memorial Prize, and the Hachemeister Prize.

ACTEX Learning
Learn Today. Lead Tomorrow.

9 78- 1- 63588- 548- 4

Actuarial Statistics with R

Theory and Case Studies

Guojun Gan, PhD, FSA Emiliano A. Valdez, PhD, FSA

ACTEX Learning

Requests for permission should be addressed to
ACTEX Learning
4 Bridge Street
New Hartford CT 06057

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Cover design by Jeff Melaragno
Cover photo by Brandon Hill. Website: www.brandonhill.photography

ISBN: 978-1-63588-549-1

Contents

List of Figures

List of Tables

Foreword

This timely book offers the reader a thorough introduction and overview of the topic of statistics, with applications in actuarial science, mathematical finance and quantitative risk management. The text contains sections on supervised learning, unsupervised learning, time-series models and simulation. These topics, related to both traditional and modern methods in data analytics are deemed important for today's actuaries according to the International Actuarial Association and the profession at large. The authors have brought together in one volume a unique and inspiring survey of these topics. The breadth of the coverage provides an almost full-scale picture of statistics that is applicable in actuarial, financial and quantitative risk management contexts.

One of the great merits of this book is that the presentation is far from encyclopedic. In the treatment of the different topics, several chapters contain the necessary theoretical background, immediately followed by many case studies as practical illustrations. The case studies are relevant, timely and trigger a reader's curiosity to learn more. They also cover a broad spectrum of highly relevant issues in insurance practice and will enable the reader to immediately apply what they learned to a practical situation. This pedagogical approach of blending theory with case studies using R makes this book unique in the area of statistics and at the same time relevant to both students and practitioners. The appendix offering an introduction to R provides a useful addition that helps to make the book self-contained and comprehensive. The mathematical skills required to be able to read the book are at a reasonable level. It assumes knowledge in probability and inference; a one-year course in mathematical statistics is sufficient.

The actuarial community is fortunate to have Guojun Gan and Emiliano Valdez write this significant book for the actuarial profession and the financial community. Dr. Valdez has built up a worldwide reputation over the years and Dr. Gan is quickly gaining momentum as a recognized expert in data mining and data analytics. Both authors are colleagues at the University of Connecticut, which offers an outstanding actuarial program. I have had the pleasure to work with Dr. Valdez on some research-related projects. During our collaborations

and accompanying discussions, I saw firsthand Dr. Valdez's strong commitment to improving and expanding the knowledge base for the actuarial profession. This book is a testament to his commitment. Both authors deserve praise and congratulations for this remarkable work!

Jan Dhaene

Prof. Dr. Jan Dhaene holds a Ph.D. in Actuarial Science from KU Leuven (Belgium) and is a full professor with the Actuarial Research Group of the Department of Accountancy, Finance and Insurance at the Faculty of Business and Economics of KU Leuven. He is a member of the Institute of Actuaries of Belgium.

He is the head of the Insurance Research Centre (Actuarial Research Group) at KU Leuven. He is the co-author of several books on actuarial science and has authored over 120 scientific papers in refereed journals. He is an Associate Editor of Insurance: Mathematics & Economics, a member of the editorial board of ASTIN Bulletin, Advisory Editor of the Journal of Computational and Applied Mathematics, and the Editor-in-Chief of the Iranian Journal of Risk and Insurance.

Preface

This book is written primarily for actuarial students and practitioners who wish to learn the basic fundamentals and applications of statistical and simulation models using R programming. We assume that readers have studied probability at the level of (Ross, 2012a). We also assume that readers come to it with some knowledge of mathematical statistics (e.g., descriptive statistics, hypothesis testing, and confidence intervals), finance (e.g., risk-free interest rate, stocks, and returns), linear algebra (e.g., matrix operations), and calculus. We do not assume any prior programming experience.

In this textbook, we use a series of case studies to introduce the applications of classical supervised learning, unsupervised learning, time series, and simulation models. The content covers several topics on data analytics that have been prescribed by the International Actuarial Association. In particular, it has been designed to cover the learning objectives for the Statistics for Risk Modeling (SRM) Exam established by the Society of Actuaries. Some materials from this textbook also cover parts of the syllabus for the Modern Actuarial Statistics (MAS-I and MAS-II) Exams of the Casualty Actuarial Society. The treatment in this textbook differs from existing books in the following ways. First, this textbook teaches the steps of how to implement models in R at an elementary level. Second, this book gives students the opportunity to recognize the power of applied statistics and R programming by exposing them to real world problems. Finally, this book uses the case study method that helps better connect theory to practice and bridge the gap between academia and the workforce (Barkley et al., 2014).

During the past two decades, the rapid advancement of information technology has led to an explosive increase of data in various fields. Beyond all doubt, we are living in the era of *big data*; the term *big data* was coined to describe the enormous amount of data captured by enterprises in our world. According to a report (Manyika et al., 2011) published by McKinsey & Company, an American multinational management consulting firm, big data is the "the Next Frontier for Innovation, Competition, and Productivity." Big data has also attracted significant attention from many national governments including the US government.

In March 2012, the Obama Administration announced more than $200 million in investment to launch the "Big Data Research and Development Initiative."

Big data is affecting almost all industries, including the insurance industry (Ferris et al., 2014). Actuaries generally have excellent business acumen, but many do not get the proper training in order to code in high-level programming languages such as Java, C++, C#, and R. As coding and programming are essential to conduct analysis for large datasets, we have written this textbook to teach R programming to undergraduate students who intend to pursue a career in actuarial science, finance, or quantitative risk management. At the same time, this textbook aims to teach students topics in applied statistics including supervised learning, unsupervised learning, and time series models. Although this book does not deal directly with big data, it helps students develop the skills that are necessary for big data analysis.

Among the many high-level programming languages, we have chosen to teach R to students for the following reasons. First, R is an open source programming language and software environment designed for data analysis and visualization. Students can use R free of charge. Second, R contains many packages that make the language versatile and because of this versatility, R has become very popular and is useful both in academia and in the professional world. Third, R is easy to learn compared to other high-level programming languages such as C++ and Java. Finally, R is one of the top five tools used for big data mining and analysis according to a survey conducted by KDNuggets in 2012 (Chen et al., 2014, Section 5.4).

This book can serve several purposes. First, this book can be used as a primary textbook for a course taken by students to study the SRM Exam or a similar one. Second, this book is useful for students and practitioners to learn R programming and its applications in actuarial science, finance, and quantitative risk management. Third, this book is also suitable as a supplementary book used by instructors for courses related to statistical data analysis. Finally, the uniqueness of the case study approach used by this book provides significant promise to recommend it as a reference for advanced actuarial exams and online modules requiring R programming.

This book would not have been made possible without the assistance of several people. First, we would like to extend our appreciation to the following reviewers whose comments helped to significantly improve the quality of this book: Mary Pat Campbell, Runhuan Feng, Louise Francis, Yuanying Michelle Guan, Emma Ran Li, Nicole Radziwill, Peng Shi, and Xiaofei Susan Wang. We would like to thank the students at the University of Connecticut who used and commented on earlier versions of this book. Our special thanks also go to Michal Pesta and the participants of the Workshop on Advanced Statistical Methods

hosted by the Czech Society of Actuaries and Charles University in Prague on June 25-26, 2018; many materials presented by Professor Valdez at that workshop were derived from this textbook. We are also very grateful to Garrett Doherty, who checked the final layout and shepherded the book through production, Brandon Hill, who took the cover pictures, and Jeff Melaragno, who designed the book cover. Last but not certainly the least, our warm-hearted thanks go to Stephen Camilli of ACTEX Learning, whose continued support and constant communication helped facilitate the completion of this book.

Guojun Gan and Emiliano A. Valdez
Storrs, Connecticut, USA
June 30, 2018

Access to R Code and Data Sets

As part of your purchase of this book, you may also access the R code and data sets mentioned within the book. To access these, navigate to the Product Supplements section of the publisher's website, www.actexmadriver.com. If you have any issues accessing this material, please contact the publisher at support@actexmadriver.com.

Chapter 2

Case Study: Implementing the Capital Asset Pricing Model

The capital asset pricing model (CAPM) is a widely used model in finance that is used to price an individual stock or portfolio. In this case study, we illustrate how to implement the CAPM using a simple linear regression model. In particular, readers will be able to do the following in R:

- summarize and visualize data
- build simple linear regression models
- use the method of least squares to estimate regression parameters
- calculate and interpret the coefficient of determination (R^2) and the mean squared error
- create and interpret the ANOVA table
- calculate confidence intervals of regression coefficients
- calculate prediction intervals
- understand the numbers in the model summary produced by R
- identify outliers and high leverage points

2.1 Problem Description

Sharpe (1964) and Lintner (1965) developed the capital asset pricing model (**CAPM**), which can be used to price an individual stock or portfolio. Under the CAPM model, all investors are assumed to be rational and risk-averse, have homogeneous expectations, be broadly diversified across a range of investments, and be able to borrow and lend money freely at the same risk-free rate. In such a

market, the expected return of a stock can be expressed as

$$E[R] = R_f + \beta(E[R_m] - R_f), \tag{2.1}$$

where R is the return on the stock, R_m is the return on the market portfolio, R_f is the return on the risk-free asset, and β is a parameter that can be interpreted as a measure of the riskiness of the stock. In the CAPM, β is determined by

$$\beta = \frac{\text{Cov}(R, R_m)}{\text{Var}(R_m)} = \rho(R, R_m)\frac{\sqrt{\text{Var}(R)}}{\sqrt{\text{Var}(R_m)}}, \tag{2.2}$$

where $\text{Cov}(R, R_m)$ denotes the covariance between R and R_m, $\rho(R, R_m)$ denotes the correlation between R and R_m, $\text{Var}(R)$ and $\text{Var}(R_m)$ denote the variances of R and R_m, respectively.

To implement the CAPM, we can fit a basic (or single) linear regression model of a stock's excess return on the market-portfolio's excess return as follows

$$R_i - R_{f,i} = \alpha + \beta(R_{m,i} - R_{f,i}) + \epsilon_i, \quad i = 1, 2, \ldots, n, \tag{2.3}$$

where R_i is the realized return on the stock in period i, $R_{m,i}$ is the realized return on the stock in period i, $R_{f,i}$ is the realized return on the risk-free asset in period i, n is the number of data points, and ϵ_i is random noise.

Under the assumptions of the CAPM, the regression coefficients (α, β) estimated from Equation (2.3) are such that α is zero and β is the same as in the CAPM model given in Equation (2.1). In the subsequent sections, we illustrate the implementation of the CAPM by estimating the β of Manulife Financial's stock. Manulife Financial is one of the largest insurance companies with its corporate headquarter in Toronto, Canada. In our implementation, we use the S&P 500 index as a proxy for the market-portfolio and the 3-month US treasury rate as the risk-free rate.

2.2 Data Description

To estimate the β of Manulife Financial's stock, we follow standard practice in the securities industry and use monthly prices. We downloaded historical monthly data from Yahoo Finance[1]. The symbols of Manulife Financial's stock, the S&P 500 index, and the 3-month treasury rate are MFC, ^GSPC, and ^IRX, respectively.

Since we downloaded the data from the same source, the formats of the files are the same. All the data files contain the following seven columns: Date,

[1] `http://finance.yahoo.com/`

Open, High, Low, Close, Volume, and Adj Close. Since the data is monthly data, the Date column contains the dates of the first business days of the available months. The Open, High, Low, and Close columns contain the open, the highest, the lowest, and the close prices of the corresponding months, respectively. The Volume column contains the number of transactions for the months. The Adj Close column contains the close price adjusted for dividends. We use the prices from the Adj Close column for our estimation.

The price data of the three securities was saved as three CSV (Comma-Separated Values) files named as `MFC.csv`, `sp500.csv`, and `irx.csv`, respectively. Although the three files have the same format, they contain a different number of data points. The file `MFC.csv` contains prices for 192 days, with one observation from each month, from September 24, 1999 to August 3, 2015. The file `sp500.csv` contains prices for 788 days from January 3, 1950 to August 3, 2015. The file `irx.csv` contains prices for 308 days from January 4, 1990 to August 3, 2015.

2.3 Loading the Data into R

Since the data was saved into CSV files, we can use the function `read.csv` to load the data into R. Suppose that the data files are in the current working directory. Then we can read the data as follows:

```
> mfc <- read.csv('MFC.csv',stringsAsFactors=FALSE)
> sp500 <- read.csv('sp500.csv',stringsAsFactors=FALSE)
> irx <- read.csv('irx.csv',stringsAsFactors=FALSE)
> head(mfc)
         Date  Open  High   Low Close  Volume Adj.Close
1 2015-08-03 17.74 18.00 14.26 15.76 2708000  15.76000
2 2015-07-01 18.76 18.91 17.08 17.73 1899400  17.59599
3 2015-06-01 18.30 19.61 18.04 18.59 2290100  18.44949
4 2015-05-01 18.18 19.34 18.07 18.35 1795600  18.21131
5 2015-04-01 16.97 18.58 16.79 18.21 1697600  17.93680
6 2015-03-02 17.48 17.73 16.57 17.01 2081400  16.75480
> dim(mfc)
[1] 192   7
> dim(sp500)
[1] 788   7
> dim(irx)
[1] 308   7
```

In the above code, we used the function `head` to display the first several rows of the data frame. Actually, this function can be used to display the first several elements of a vector and the first several rows of a matrix. From the first several

rows of the stock price data, we see that the prices are in reverse chronological order.

Exercise 2.1. Suppose that we read the file `MFC.csv` using the following command

```
mfc <- read.csv('MFC.csv')
```

What is the return value of the following call?

```
mode(mfc$Date)
```

Exercise 2.2. Let `mfc` be the data frame created by

```
mfc <- read.csv('MFC.csv', stringsAsFactors=FALSE)
```

Look at the help of the function `as.Date` and convert the vector of strings `mfc$Date` to a vector of date objects in R.

Now we have the raw price data of the three securities in the R workspace. Suppose that we want to use 10 years of monthly returns from January 2005 to December 2014 to estimate the β of Manulife Financial's stock. We need to extract the price data and calculate the returns for the stock and the S&P 500 index. Since the monthly prices are obtained from the first business days of the months, we can use the next month's price as this month's end price. Then we can extract the price data and calculate the returns as follows:

```
> ind <- seq(from=127,to=8,by=-1)
> mfcReturn <- mfc$Adj.Close[ind] / mfc$Adj.Close[ind+1]
    - 1
> sp500Return <- sp500$Adj.Close[ind] / sp500$Adj.Close[
    ind+1] - 1
> rfRate <- irx$Adj.Close[ind+1] / 1200
> dat10y <- data.frame(mfc=mfcReturn - rfRate, sp500=
    sp500Return - rfRate)
> head(dat10y)
            mfc         sp500
1  0.062016836  0.016885013
2  0.030516898 -0.021359322
3 -0.045465272 -0.022376915
```

```
4  0.006420319  0.027587880
5  0.038077092 -0.002544381
6  0.049531119  0.033418287
> tail(dat10y)
                mfc         sp500
115 -0.004217211  0.037640295
116 -0.046100286 -0.015528837
117 -0.014556400  0.023190627
118  0.056440711  0.024531089
119 -0.040707677 -0.004192755
120 -0.157704995 -0.031071639
```

In the above code, we first created a vector of indices named `ind` in order to extract the data we wanted for our analysis. The vector contains a sequence of decreasing integers so that we can change the order of the prices to a chronological order. Then we calculated the monthly returns for the stock and the index. We also calculated the risk-free rates from the 3-month US treasury rates by dividing them by 12×100 because these treasury rates are annualized percentages. In Line 5 of the above output, we calculated the excess monthly returns of the stock and the index and put them into a data frame named `dat10y`.

In Line 14 of the above output, we used the function `tail` to display the last several rows of the data frame. Similar to the function `head`, the function `tail` is a useful function for checking data in R.

2.4 Data Visualization and Summarization

We now have the excess returns on the stock and the index in the R workspace. We are ready to examine the data before fitting a regression model. We need to make sure the data satisfies the assumptions of linear regression models.

To get an impression of the distribution of each variable, we plot **histograms** of the two variables as follows:

```
par(mfrow=c(1,2))
hist(dat10y$mfc, breaks=20)
hist(dat10y$sp500, breaks=20)
```

The resulting histograms are shown in Figure 2.1. The histograms show that both variables are approximately normally distributed.

Instead of plotting histograms, we can also calculate the summary statistics to investigate the distribution of an individual variable in isolation of the other. Common summary statistics include the minimum, the maximum, the mean,

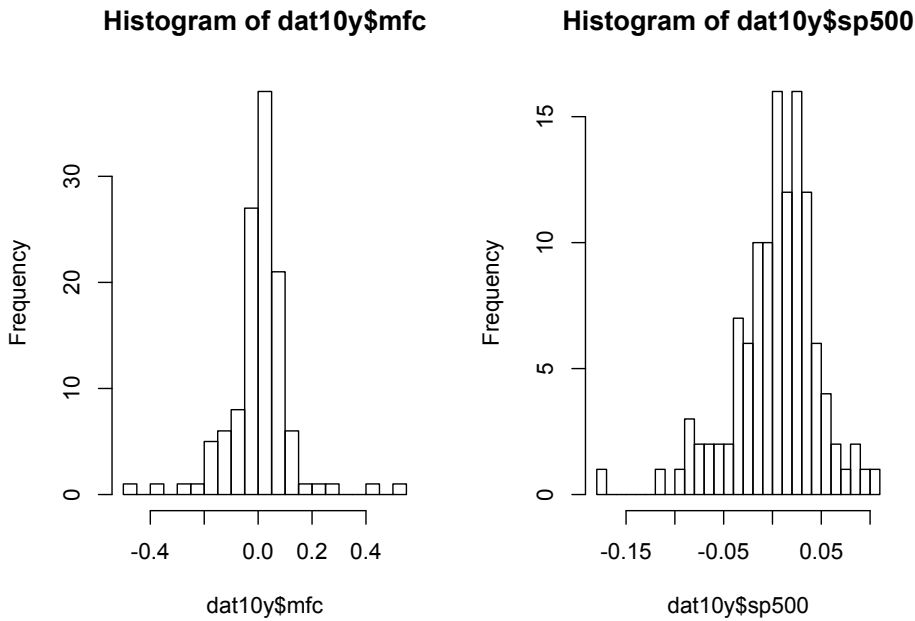


Figure 2.1: *Histograms of the excess returns on Manulife Financial's stock and the S&P 500 index.*

and the percentiles. To get the summary statistics of the variables, we use the function `summary` as follows:

```
> summary(dat10y)
      mfc                 sp500
 Min.   :-0.453189   Min.   :-0.170175
 1st Qu.:-0.025397   1st Qu.:-0.016219
 Median : 0.014469   Median : 0.008698
 Mean   : 0.005639   Mean   : 0.004129
 3rd Qu.: 0.051921   3rd Qu.: 0.028892
 Max.   : 0.526619   Max.   : 0.107715
```

The output of the function `summary` gives us an overview of the statistical properties of the data. From these summary statistics, we observe a wide range of excess returns on both Manulife Financial's stock and the S&P500 index, but more so with the excess returns on Manulife Financial's stock. For example, for Manulife Financial's stock, the maximum excess return is 52.7% and the minimum excess return is -45.3%. Furthermore, since the mean is less than the median, both

variables are slightly skewed to the left.

Histograms and summary statistics are useful to examine the distribution of an individual variable in isolation of the other. Since linear regression is used to model the linear relationship between variables, we can use **scatter plots** to visualize the relationship between the two variables. To produce a scatter plot of the excess returns on the stock and those on the index, we proceed as follows:

```
1 with(dat10y, plot(sp500, mfc))
```

In the above code, we used the function `with`, which applies an expression to a dataset. The above command is equivalent to the following command:

```
1 plot(dat10y$sp500, dat10y$mfc)
```

The resulting scatter plot is shown in Figure 2.2. The scatter plot shows a positive linear relationship.

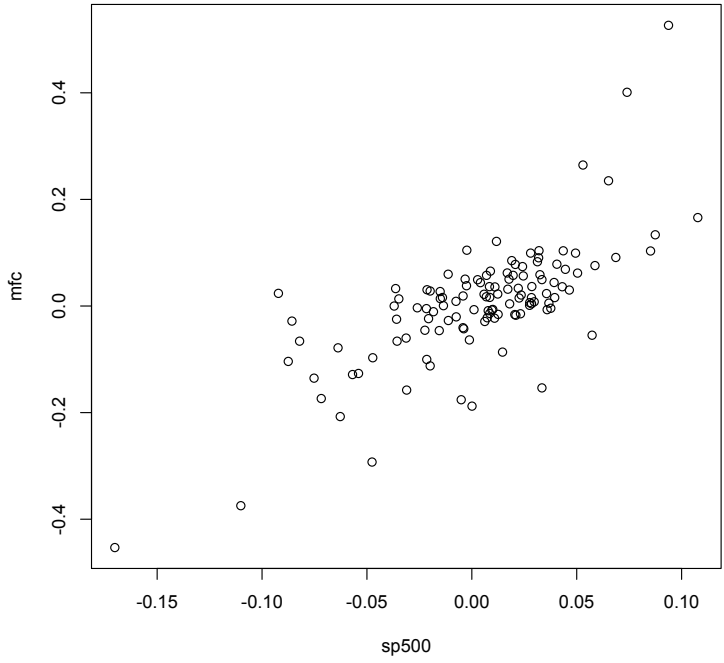


Figure 2.2: *A scatter plot of the excess returns on Manulife Financial's stock and those on the S&P 500 index.*

To calculate the correlation coefficient between the excess returns, we proceed as follows:

```
> with(dat10y, cor(mfc, sp500))
[1] 0.725671
```

The correlation coefficient of the excess returns turns out to be about 0.73, which indicates that the excess returns on the stock are positively correlated to those on the index. In this case, when the excess return on the index is high, the excess return on the stock is also high, and vice versa.

Exercise 2.3. The sample mean and the sample standard deviation of $\{x_1, x_2, \ldots, x_n\}$ are defined in Equation (1.2) and Equation (1.3), respectively.

(a) Write an R function called `calculateStd(x)` to calculate the sample standard deviation of the vector `x`. What is the return of the following call?

```
calculateStd(dat10y$mfc)
```

(b) Write an R function called `calculateCorr(x,y)` to calculate the Pearson correlation coefficient of the two vectors `x` and `y`. What is the return of the following call:

```
calculateCorr(dat10y$mfc, dat10y$sp500)
```

2.5 Fitting a Basic Linear Regression Model

According to our analysis in the previous section, the excess returns on the stock seem to be normally distributed and have a strong positive linear relationship with those of the index. In this section, we fit a linear regression model to the data by using the method of least squares to estimate the regression coefficients.

To fit a regression line to the data in R, we use the function `lm` as follows:

```
> fit <- lm(mfc ~ sp500, data=dat10y)
> summary(fit)

Call:
```

```
lm(formula = mfc ~ sp500, data = dat10y)

Residuals:
      Min        1Q    Median        3Q       Max
-0.21652 -0.04091 -0.00232  0.04084  0.34572

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.002436   0.007262  -0.335    0.738
sp500        1.955416   0.170676  11.457   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
   ' ' 1

Residual standard error: 0.07917 on 118 degrees of
   freedom
Multiple R-squared:  0.5266,  Adjusted R-squared:  0.5226
F-statistic: 131.3 on 1 and 118 DF,  p-value: < 2.2e-16
```

From Line 1 of the above output, we see that the function `lm` takes two arguments. The first argument is a formula. The variable on the left hand side of the symbol ~ is the dependent variable y; the variable on the right hand side of the symbol is the independent variable x. Table 2.1 gives some common names of regression variables. The second argument specifies the data set. We saved the fitting result to an object named `fit` and then used the function `summary` to show the summary of the fitted regression model.

Variable y	Variable x
Dependent variable	Independent variable
Response	Treatment
Output	Input
Endogenous variable	Exogenous variable
Predicted variable	Predictor variable
Regressand	Regressor

Table 2.1: *Common names of regression variables.*

The model summary contains the formula used to fit the model, summary statistics of the residuals, the regression coefficients, and other useful information. From the model summary, we get the intercept and slope estimates:

$$\hat{\alpha} = -0.002436, \quad \hat{\beta} = 1.955416. \tag{2.4}$$

We can also plot the fitted regression line and the data in the same figure using the function `abline` as follows:

```
with(dat10y, plot(sp500, mfc))
abline(fit)
```

The resulting plot is shown in Figure 2.3, from which we see that the data points surround the fitted regression line.

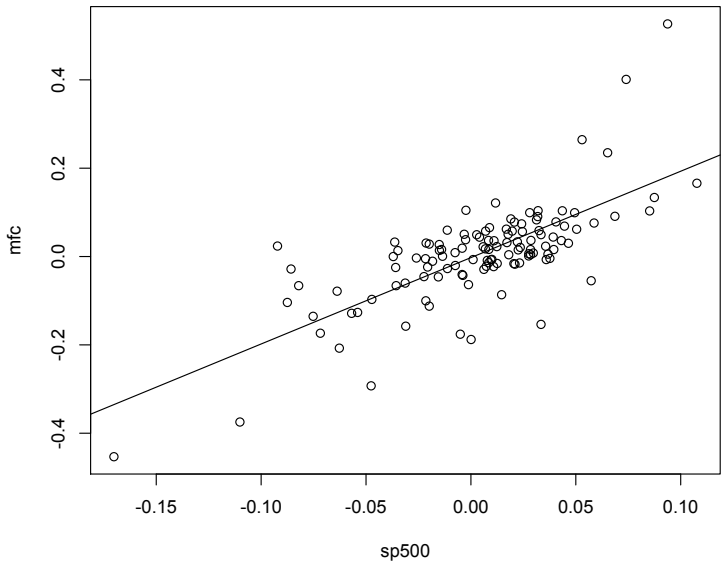


Figure 2.3: *A scatter plot of `sp500` and `mfc` with the fitted regression line.*

Exercise 2.4. Use the formulas given in Equation (1.7) and write a piece of R code to calculate the estimated regression coefficients $\hat{\alpha}$ and $\hat{\beta}$. Compare your results to the estimates shown in the model summary. You can use the R built-in functions `cor`, `sd`, and `mean` to calculate the correlation coefficient, the sample standard deviation, and the sample mean, respectively.

Exercise 2.5. Let x and y be the two vectors obtained from the following R code:

```
x <- dat10y$sp500
y <- dat10y$mfc
```

(a) Write R code to compute the following sum:

$$\sum_{i=1}^{n} w_i y_i, \tag{2.5}$$

where n is the length of the vector x and

$$w_i = \frac{x_i - \bar{x}}{s_x^2(n-1)}.$$

Here $\bar{x}$ and s_x denote the sample mean and the sample standard deviation of x, respectively.

(b) Does the value of the sum in Equation (2.5) equal to the slope estimate $\hat{\beta}$ given in Equation (2.4)?

2.6 Model Evaluation

Once we fit a basic linear regression model, we need to justify the quality of the fit of the regression model. To measure the fit of linear regression models, we can use the **coefficient of determination**, which is also referred to as R**-squared**.

Exercise 2.6. The fitted values $\{\hat{y}_i\}$ can be extracted from the R object produced by `lm` as follows:

```
1 fit <- lm(mfc ~ sp500, data=dat10y)
2 haty <- fit$fitted.values
```

(a) Write R code to calculate SST, SSE, and SSR defined in Equations (1.14), (1.15), and (1.16), respectively.

(b) Calculate $SSE + SSR - SST$. Is it equal to zero?

(c) Calculate the following sum

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)(\hat{y}_i - \bar{y}).$$

Is it equal to zero?

The R^2 of the linear regression model can be found in the model summary. For the model we just fitted, the R^2 is 0.5266. The R^2 shows that the basic linear regression model explains more than half of the total variability of the dependent variable.

Exercise 2.7. Suppose that we save the summary of the model into a variable as follows:

```
fit <- lm(mfc ~ sp500, data=dat10y)
fitsummary <- summary(fit)
```

(a) The R object `fitsummary` is a list. The estimated regression coefficients are stored in the object `coefficients` of the list. Use list operations to extract the estimated regression coefficients $\hat{\alpha}$ and $\hat{\beta}$ from the list `fitsummary`.

(b) Use the formula in Equation (1.17) and write R code to calculate the R^2.

In R, we can produce the ANOVA table using the function `anova` as follows:

```
> anova(fit)
Analysis of Variance Table

Response: mfc
           Df  Sum Sq Mean Sq F value    Pr(>F)
sp500       1 0.82277 0.82277  131.26 < 2.2e-16 ***
Residuals 118 0.73966 0.00627
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
   ' ' 1
```

The ANOVA table produced by R does not show the total sum of squares. However, we can derive the total sum of squares by the following formula

$$SST = SSR + SSE.$$

Exercise 2.8. Given the following ANOVA table produced by R:

```
            Df  Sum Sq Mean Sq F value    Pr(>F)
sp500        1 0.82277 0.82277  131.26 < 2.2e-16 ***
Residuals  118 0.73966 0.00627
```

Calculate the R^2 and the residual standard error s.

2.7 Residual Analysis

In this section, we examine the residuals to check if there are any unusual points. To find high leverage points, we can proceed as follows:

```
> barx <- with(dat10y, mean(sp500))
> sx <- with(dat10y, sd(sp500))
> n <- dim(dat10y)[1]
> h <- 1/n + (dat10y$sp500 - barx)^2 / ( (n-1) * sx^2 )
> indh <- h > 6/n
> HighLeveragePoints <- dat10y[indh,]
> HighLeveragePoints
             mfc        sp500
44   0.02373827 -0.09219977
45  -0.45318935 -0.17017452
49  -0.37464108 -0.11011453
81   0.16592281  0.10771471
> h[indh]
[1] 0.05145677 0.14952613 0.06898800 0.05819836
```

From the above R output, we see that there are four high leverage points. We can label the high leverage points in the scatter plot using the following piece of code:

```
with(dat10y, plot(sp500, mfc))
abline(fit)
for(i in 1:dim(HighLeveragePoints)[1] ) {
  p <- HighLeveragePoints[i,]
  text(p$sp500, p$mfc, labels=rownames(p), pos=3)
}
```

The resulting plot is shown in Figure 2.4. Since the leverages for the observations 44, 45, 49, and 81 are close to $6/120 = 0.05$, they are not severe high leverage

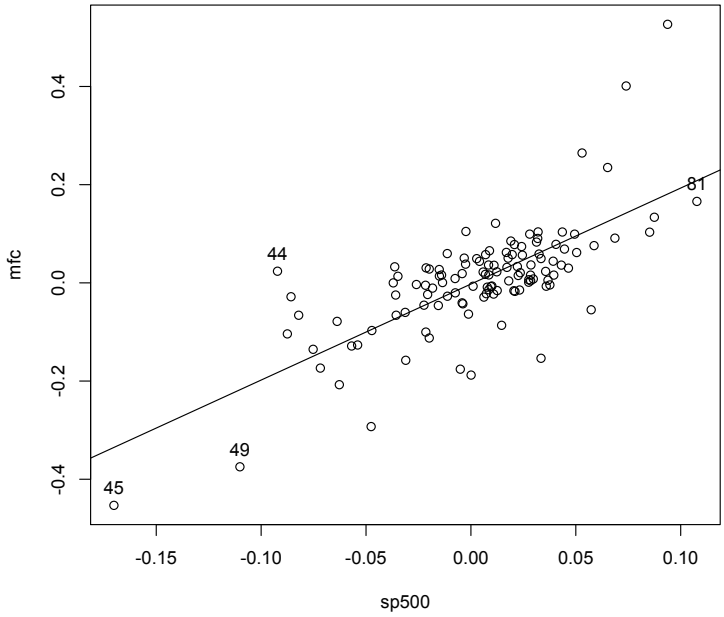


Figure 2.4: *Four high leverage points with labels above them.*

points. Here we call a point a severe high leverage point if its leverage is much larger than the average.

Now let us look at the standardized residuals to see if there are any severe outliers. To calculate the standardized residuals and identify outliers, we proceed as follows:

```
> e <- fit$residuals
> s <- sqrt( sum(e^2) / (n-2) )
> sr <- e / (s * sqrt(1-h) )
> indsr <- abs(sr) > 2
> Outliers <- dat10y[indsr,]
> sr[indsr]
        44         49         51         52         53
        54         55         58         67         82
 2.677566  -2.053673   4.469910   2.083716  -2.353886
    3.320130  -2.751849  -2.099776  -2.518124  -2.074143
```

We identified ten outliers. We can label these outliers in the scatter plot using the

following code:

```
with(dat10y, plot(sp500, mfc))
abline(fit)
for(i in 1:dim(Outliers)[1] ) {
  p <- Outliers[i,]
  text(p$sp500, p$mfc, labels=rownames(p), pos=4)
}
```

The resulting graph is shown in Figure 2.5. From the figure, we see that most of the identified outliers are not severe outliers. Observations 51 and 54 might have a large effect on the regression model.

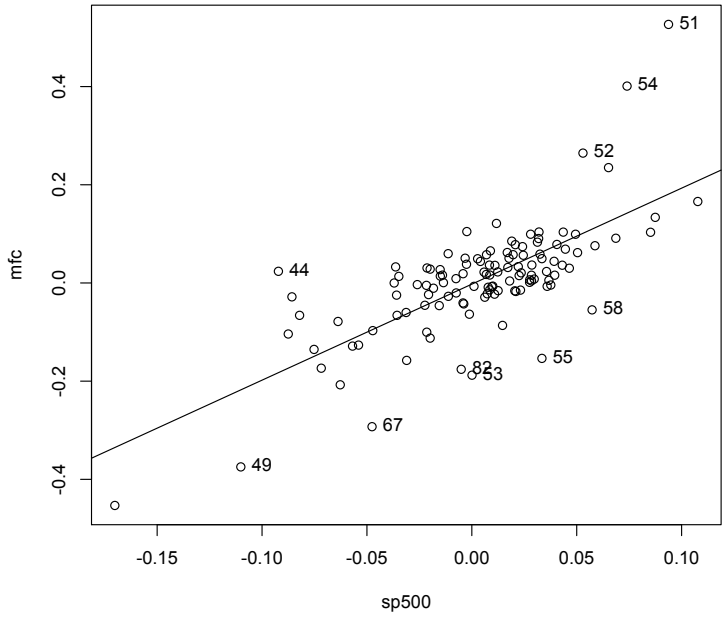


Figure 2.5: *Ten outliers with labels in their right-hand sides.*

From our above analysis, observations 45, 51, and 54 seem to be severe unusual points. We can remove them and fit a new linear regression model as follows:

```
> dat10yb <- dat10y[-c(45,51,54),]
> fitb <- lm(mfc ~ sp500, data=dat10yb)
```

```
> summary(fitb)

Call:
lm(formula = mfc ~ sp500, data = dat10yb)

Residuals:
      Min         1Q     Median         3Q        Max
-0.214103 -0.031488  0.002582  0.040869  0.187241

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.004871   0.006189  -0.787    0.433
sp500        1.550670   0.160483   9.663   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
   ' ' 1

Residual standard error: 0.06653 on 115 degrees of
   freedom
Multiple R-squared:  0.4481,  Adjusted R-squared:  0.4433
F-statistic: 93.36 on 1 and 115 DF,  p-value: < 2.2e-16
```

Comparing the R^2 from our first model, we see that the R^2 of the new model reduced to 0.4481. Since the three observations we removed are at both ends of the x-axis, deleting them from the data set makes the model worse. We can keep the original model because its R^2 is higher.

Exercise 2.9. Suppose that `dat22` is a data frame created by the following R code[2]:

```
x <- c(1.5, 1.7, 2, 2.2, 2.5, 2.5, 2.7, 2.9, 3, 3.5, 3.4,
    9.5, 9.5, 3.8, 4.2, 4.3, 4.6, 4, 5.1, 5.1, 5.2, 5.5)
y <- c(3, 2.5, 3.5, 3, 3.1, 3.6, 3.2, 3.9, 4, 4, 8, 8,
   2.5, 4.2, 4.1, 4.8, 4.2, 5.1, 5.1, 5.1, 4.8, 5.3)
dat22 <- data.frame(x=x, y=y)
```

(a) Fit a linear regression model to this data set using x as the explanatory variable and y as the dependent variable.

(b) Calculate the leverages for all the observations and identify which observations are high leverage points.

[2]This data set was obtained from (Frees, 2009, p. 43).

(c) Calculate the standardized residuals for all the observations and identify which observations are outliers.

2.8 Statistical Inference

To assess whether the explanatory variable (i.e., the excess returns on the S&P 500 index) is significant, we can investigate whether $\beta = 0$ using the t-test. We can calculate the t-ratio and compare it with the critical value $t_{n-2,1-\alpha/2}$ as follows:

```
> beta <- fit$coefficients[2]
> n <- dim(dat10y)[1]
> s <- sqrt( sum(fit$residuals^2) / (n-2) )
> sx <- sd(dat10y$sp500)
> sebeta <- s / (sx * sqrt(n-1) )
> d <- 0
> tratio <- (beta - d) / sebeta
> alpha <- 0.05
> tratio
    sp500
11.45687
> qt(1-alpha/2, n-2)
[1] 1.980272
```

Since the t-ratio is larger than the critical value, we reject the null hypothesis H_0 at the significance level of 5%. In the above code, we used the function `qt` to calculate the critical value $t_{n-2,1-\alpha/2}$. The hypothesis test we just performed is just one of many hypothesis tests (see Table 1.2).

Exercise 2.10. Follow the procedures given in Table 1.2 and write R code to perform the following hypothesis tests:

(a) $H_0 : \beta = 2$ versus $H_a : \beta \neq 2$ at the significance level of 1%.

(b) $H_0 : \beta = 1.5$ versus $H_a : \beta < 1.5$ at the significance level of 5%.

Exercise 2.11. Write R code to calculate a 95% confidence interval for the slope estimate $\hat{\beta}$.

Exercise 2.12. Suppose that the excess return on the S&P 500 index is −10%. Write R code to calculate a 99% prediction interval of the excess return on Manulife Financial's stock.

2.9 Summary

In this chapter, we introduced how to implement the capital asset pricing model using basic linear regression models. The CAPM is a model for pricing an individual security or portfolio. For more information on the CAPM, readers are referred to (Campbell et al., 1996, Chapter 5) and (Cochrane, 2001, Chapter 9). Through this case study, we introduced how to build and analyze basic linear regression models using R. In particular, we introduced how to visualize data and check some assumptions of regression models, fit a basic linear regression model to a dataset, and evaluate the fitted model, among others.

2.10 End-of-Chapter Exercises

Exercise 2.13. Given the following R output of a regression model

```
Call:
lm(formula = Y ~ X, data = data4c)

Residuals:
     Min       1Q   Median       3Q      Max
-0.87482 -0.02201  0.01517  0.05316  0.28862

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.17469    0.04537   -3.85  0.00014 ***
X            1.01923    0.01012  100.73  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.09373 on 353 degrees of freedom
Multiple R-squared:  0.9664,    Adjusted R-squared:  0.9663
F-statistic: 1.015e+04 on 1 and 353 DF,  p-value: < 2.2e-16
```

and

```
> qt(0.95,353)
[1] 1.649182
> qt(0.975,353)
[1] 1.966707
```

(a) What does −0.17469 mean?

(b) Is variable X significant? Why?

(c) What are the hypotheses associated with the t-value 100.73?

(d) What is the mean square error of this model?

(e) What is the sample standard deviation of the variable X?

(f) Suppose that the variable X has a change of 3. What is the expected change of Y?

(g) Suppose that the variable X has a change of 2. What is the 95% confidence interval of the expected change in Y?

(h) Test the following hypothesis at the 5% level of significance:

$$H_0 : \beta_1 = 1 \text{ versus } H_a : \beta_1 \neq 1$$

(i) Test the following hypothesis at the 5% level of significance:

$$H_0 : \beta_1 = 1 \text{ versus } H_a : \beta_1 > 1$$

Exercise 2.14. Summary statistics of the variables income (in thousands) and education (in years) are provided below:

```
> summary(income)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.611   4.106   5.930   6.798   8.187  25.880
> sd(income)
[1] 4.245922
> summary(education)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  6.380   8.445  10.540  10.740  12.650  15.970
> sd(education)
[1] 2.728444
```

The regression model

$$\text{income} = \beta_0 + \beta_1 \text{education} + \epsilon$$

was fitted to the data summarized above. The fitted regression line is shown in Figure 2.6.

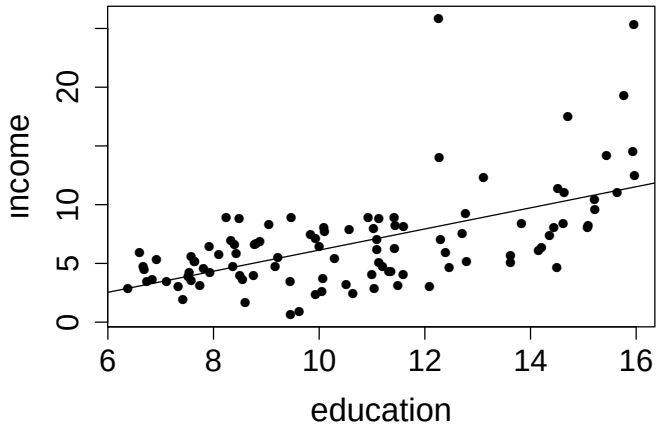


Figure 2.6: *The linear regression model fitted to the data.*

The ANOVA table from the fit and additional R outputs are given below:

```
> anova(fit)
Analysis of Variance Table

Response: income
           Df  Sum Sq Mean Sq F value    Pr(>F)
education   1  607.42  607.42   50.06 2.079e-10 ***
Residuals 100 1213.39   12.13
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> qt(.95,100)
[1] 1.660234
> qt(.975,100)
[1] 1.983972
```

(a) How many observations were used to fit the model?

(b) Calculate the coefficient of determination and interpret this value.