

# Online Tutorial on Regression Modeling with Actuarial and Financial Applications

Edward W. (Jed) Frees, University of Wisconsin-Madison



# Contents

<b>Preface</b>	<b>7</b>
About Regression Modeling . . . . .	7
Resources . . . . .	7
Tutorial Description . . . . .	7
Welcome to the Tutorial Video . . . . .	8
<b>1 Regression and the Normal Distribution</b>	<b>11</b>
1.1 Fitting a normal distribution . . . . .	11
1.1.1 Video . . . . .	11
1.1.2 Exercise. Fitting Galton's height data . . . . .	14
1.1.3 Exercise. Visualizing child's height distribution . . . . .	15
1.2 Visualizing distributions . . . . .	15
1.2.1 Video . . . . .	15
1.2.2 Exercise. Visualizing bodily injury claims with density plots . . . . .	18
1.3 Summarizing distributions . . . . .	18
1.3.1 Video . . . . .	19
1.3.2 Exercise. Summarizing bodily injury claims with box and qq plots . . . . .	22
1.3.3 Exercise. Effects on distributions of removing the largest claim . . . . .	23
1.4 Transformations . . . . .	23
1.4.1 Video . . . . .	23
1.4.2 Exercise. Distribution of transformed bodily injury claims . . . . .	25
<b>2 Basic Linear Regression</b>	<b>27</b>
2.1 Correlation . . . . .	27
2.1.1 Video . . . . .	27
2.1.2 Exercise. Correlations and the Wisconsin lottery . . . . .	28
2.2 Method of least squares . . . . .	29
2.2.1 Video . . . . .	29
2.2.2 Exercise. Least squares fit using housing prices . . . . .	30
2.3 Understanding variability . . . . .	31
2.3.1 Video . . . . .	31
2.3.2 Exercise. Summarizing measures of uncertainty . . . . .	32
2.3.3 Exercise. Effects of linear transforms on measures of uncertainty . . . . .	33
2.4 Statistical inference . . . . .	35
2.4.1 Video . . . . .	35
2.4.2 Exercise. Statistical inference and Wisconsin lottery . . . . .	36
2.5 Diagnostics . . . . .	37
2.5.1 Video . . . . .	37
2.5.2 Exercise. Assessing outliers in lottery sales . . . . .	38
<b>3 Multiple Linear Regression (MLR)</b>	<b>41</b>
3.1 Method of least squares . . . . .	41

3.1.1	Video (Exercise). Method of least squares . . . . .	41
3.1.2	Exercise. Least squares and term life data . . . . .	43
3.1.3	Exercise. Interpreting coefficients as proportional changes . . . . .	44
3.1.4	Exercise. Interpreting coefficients as elasticities . . . . .	45
3.2	Foundations of multiple linear regression . . . . .	47
3.2.1	Video (Exercise). Foundations of multiple linear regression . . . . .	47
3.2.2	Exercise. Multiple choice exercise on the theory... Maybee . . . . .	47
3.3	Statistical inference and multiple linear regression . . . . .	48
3.3.1	Video (Exercise). Statistical inference and multiple linear regression . . . . .	48
3.3.2	Exercise. Statistical inference and term life . . . . .	48
3.4	Binary variables . . . . .	49
3.4.1	Video (Exercise). Binary variables . . . . .	49
3.4.2	Exercise. Binary variables and term life . . . . .	51
3.5	Categorical variables . . . . .	52
3.5.1	Video (Exercise). Categorical variables . . . . .	52
3.5.2	Exercise. Categorical variables and Wisconsin hospital costs . . . . .	53
3.6	General linear hypothesis . . . . .	55
3.6.1	Video (Exercise). Hypothesis testing . . . . .	55
3.6.2	Exercise. Hypothesis testing and term life . . . . .	56
3.6.3	Exercise. Hypothesis testing and Wisconsin hospital costs . . . . .	58
3.6.4	Exercise. Hypothesis testing and auto claims . . . . .	59
<b>4</b>	<b>Variable Selection</b>	<b>61</b>
4.1	An iterative approach to data analysis and modeling . . . . .	61
4.1.1	Video (Exercise). An iterative approach to data analysis and modeling . . . . .	61
4.1.2	MC Exercise. An iterative approach to data modeling . . . . .	63
4.2	Automatic variable selection procedures . . . . .	64
4.2.1	Video (Exercise). Automatic variable selection procedures . . . . .	64
4.2.2	Exercise. Data-snooping in stepwise regression . . . . .	65
4.3	Residual analysis . . . . .	66
4.3.1	Video (Exercise). Residual analysis . . . . .	66
4.3.2	Exercise. Residual analysis and risk manager survey . . . . .	67
4.3.3	Exercise. Added variable plot and refrigerator prices . . . . .	68
4.4	Unusual observations . . . . .	69
4.4.1	Video (Exercise). Unusual observations . . . . .	69
4.4.2	Exercise. Outlier example . . . . .	71
4.4.3	Exercise. High leverage and risk manager survey . . . . .	72
4.5	Collinearity . . . . .	73
4.5.1	Video (Exercise). Collinearity . . . . .	73
4.5.2	Exercise. Collinearity and term life . . . . .	74
4.6	Selection criteria . . . . .	75
4.6.1	Video (Exercise). Selection criteria . . . . .	75
4.6.2	Exercise. Cross-validation and term life . . . . .	76
<b>5</b>	<b>Interpreting Regression Results</b>	<b>79</b>
5.1	Case study: MEPS health expenditures . . . . .	79
5.1.1	Video (Exercise). Case study: MEPS health expenditures . . . . .	79
5.1.2	Exercise. Summarizing data . . . . .	81
5.1.3	Exercise. Fit a benchmark multiple linear regression model . . . . .	82
5.1.4	Exercise. Variable selection . . . . .	84
5.1.5	Exercise. Model comparisons using cross-validation . . . . .	85
5.1.6	Exercise. Out of sample validation . . . . .	87
5.2	What the modeling procedure tells us . . . . .	89
5.2.1	Video (Exercise). What the modeling procedure tells us . . . . .	89

5.2.2	MC Exercise. What the modeling procedure tells us . . . . .	90
5.3	The importance of variable selection . . . . .	90
5.3.1	Video (Exercise). The importance of variable selection . . . . .	90
5.3.2	MC Exercise. Principle of parsimony . . . . .	91



# Preface

Date: 17 October 2018

## About Regression Modeling

Statistical techniques can be used to address new situations. This is important in a rapidly evolving risk management world. Analysts with a strong analytical background understand that a large data set can represent a treasure trove of information to be mined and can yield a strong competitive advantage. This book and online tutorial provides budding analysts with a foundation in multiple regression. Viewers will learn about these statistical techniques using data on the demand for insurance, healthcare expenditures, and other applications. Although no specific knowledge of actuarial or risk management is presumed, the approach introduces applications in which statistical techniques can be used to analyze real data of interest.

## Resources

- This tutorial is based on the book *Regression Modeling with Actuarial and Financial Applications*.
  - For resources associated with the book, please visit the *Regression Modeling* book web site.
- For advanced regression applications in insurance, you may be interested in the series, *Predictive Modeling Applications in Actuarial Science*.
  - Sample code and data for the series are available at series website.
- An earlier version of this tutorial, a Short Course constructed for Indonesian actuaries, uses the Datacamp learning platform.

## Tutorial Description

- This online tutorial is designed to guide you through the foundations of regression with applications in actuarial science.
- Anticipated completion time is approximately six hours.
- The tutorial assumes that you are familiar with the foundations in the statistical software **R**, such as Datacamp's Introduction to R.

**General Layout.** There are five chapters in this tutorial that summarize the foundations of multiple linear regression. Each chapter is subdivided into several sections. At the beginning of each section is a short video, typically 4-8 minutes, that summarizes the section key learning outcomes. Following the video, you can see more details about the underlying R code for the analysis presented in the video.

**Role of Exercises.** Following each video, there are one or two exercises that allow you to practice skills to make sure that you fully grasp the learning outcomes. The exercises are implemented using an online learning platform provided by Datacamp so that you need not install R. Feedback is programmed into the exercises so that you will learn a lot by making mistakes! You will be pacing yourself, so always feel free to reveal the

# Welcome to the Tutorial Video

- Describe regression briefly, i.e., in a nutshell
- Explain Galton's height example as a regression application

Show Overhead A. Galton's 1885 Regression Data

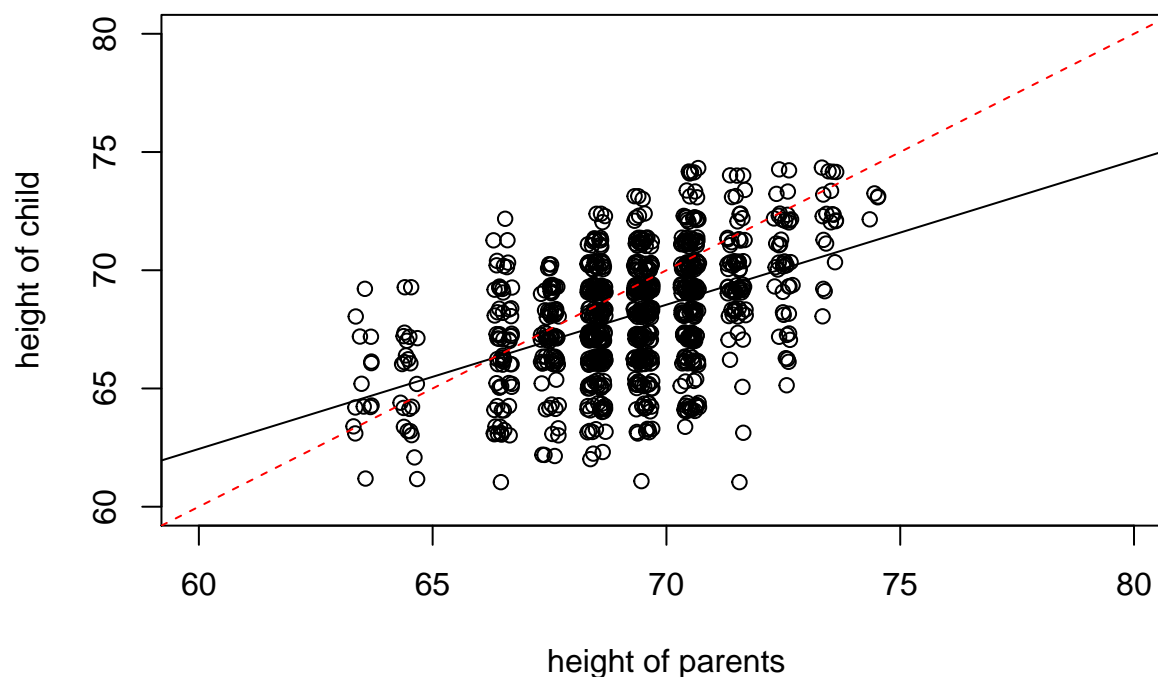
Height of adult child in inches	< 64.0	64.5	65.5	66.5	67.5	68.5	69.5	70.5	71.5	72.5	> 73.0	Totals
> 73.7	—	—	—	—	—	—	5	3	2	4	—	14
73.2	—	—	—	—	—	3	4	3	2	2	3	17
72.2	—	—	1	—	4	4	11	4	9	7	1	41
71.2	—	—	2	—	11	18	20	7	4	2	—	64
70.2	—	—	5	4	19	21	25	14	10	1	—	99
69.2	1	2	7	13	38	48	33	18	5	2	—	167
68.2	1	—	7	14	28	34	20	12	3	1	—	120
67.2	2	5	11	17	38	31	27	3	4	—	—	138
66.2	2	5	11	17	36	25	17	1	3	—	—	117
65.2	1	1	7	2	15	16	4	1	1	—	—	48
64.2	4	4	5	5	14	11	16	—	—	—	—	59
63.2	2	4	9	3	5	7	1	1	—	—	—	32
62.2	—	1	—	3	3	—	—	—	—	—	—	7
< 61.2	1	1	1	—	—	1	—	1	—	—	—	5
Totals	14	23	66	78	211	219	183	68	43	19	4	928

Show Overhead B. Supporting R Code

```
# Reformat Data Set
#heights <- read.csv("CSVData\\GaltonFamily.csv",header = TRUE)
heights <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/c85ede6c205d2204")
str(heights)
head(heights)
heights$child_ht <- heights$CHILDC
heights$parent_ht <- heights$PARENTC
heights2 <- heights[c("child_ht","parent_ht")]

#heights <- read.csv("CSVData\\galton_height.csv",header = TRUE)
heights <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/c85ede6c205d2204")
plot(jitter(heights$parent_ht),jitter(heights$child_ht), ylim = c(60,80), xlim = c(60,80),
     ylab = "height of child", xlab = "height of parents")
abline(lm(heights$child_ht~heights$parent_ht))
abline(0,1,col = "red", lty=2)
```





```
summary(lm(heights$child_ht~heights$parent_ht))
```

Call:

```
lm(formula = heights$child_ht ~ heights$parent_ht)
```

Residuals:

Min	1Q	Median	3Q	Max
-8.2577	-1.4280	0.1323	1.5720	5.7918

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	25.84856	2.69009	9.609	<2e-16 ***
heights\$parent_ht	0.60992	0.03882	15.710	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.26 on 926 degrees of freedom

Multiple R-squared: 0.2104, Adjusted R-squared: 0.2096

F-statistic: 246.8 on 1 and 926 DF, p-value: < 2.2e-16



# Chapter 1

## Regression and the Normal Distribution

### Chapter description

Regression analysis is a statistical method that is widely used in many fields of study, with actuarial science being no exception. This chapter introduces the role of the normal distribution in regression and the use of logarithmic transformations in specifying regression relationships.

### 1.1 Fitting a normal distribution

---

In this section, you learn how to:

- Calculate and interpret two basic summary statistics
  - Fit a data set to a normal curve
  - Calculate probabilities under a standard normal curve
- 

#### 1.1.1 Video

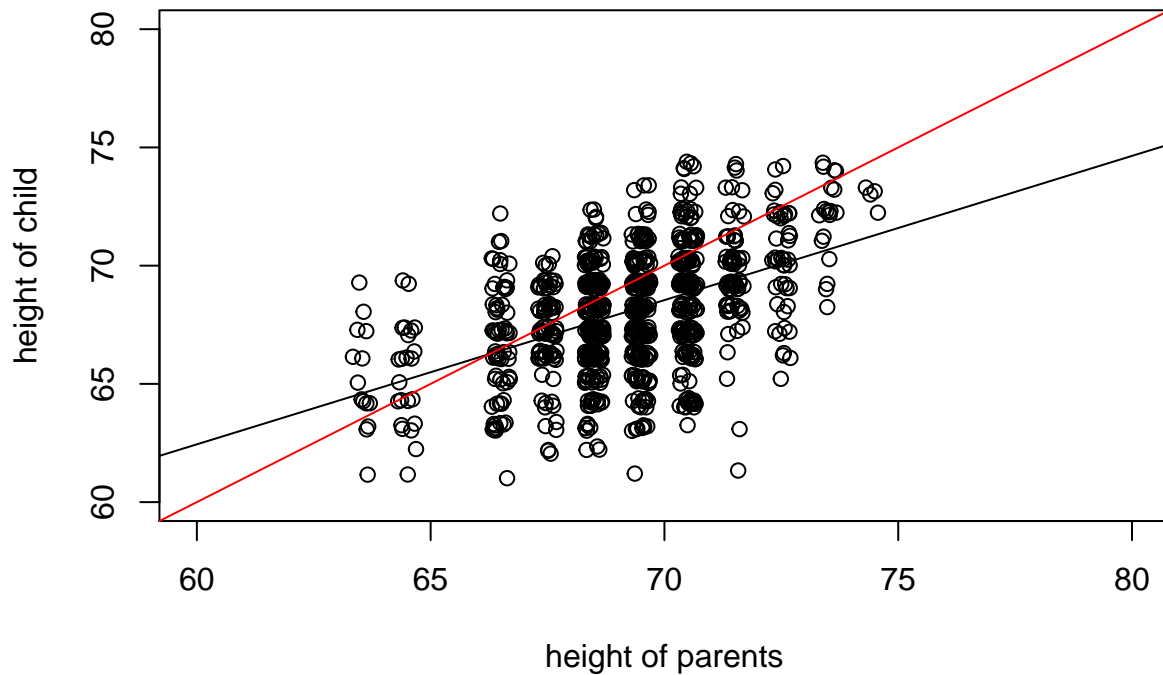
##### Video Overhead Details

Show Overhead A Details. Description of the data

To illustrate a data set that can be analyzed using regression methods, we consider some data included in Galton's 1885 paper. These data include the heights of 928 adult children (`child_ht`), together with an index of their parents' height (`parent_ht`). Here, all female heights were multiplied by 1.08, and the index was created by taking the average of the father's height and rescaled mother's height. Galton was aware that the parents' and the adult child's height could each be adequately approximated by a normal curve. In developing regression analysis, he provided a single model for the joint distribution of heights.

```
heights <- read.csv("CSVData\\galton_height.csv", header = TRUE)
#heights <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/c85ede6c205d220...")
plot(jitter(heights$parent_ht), jitter(heights$child_ht), ylim = c(60,80), xlim = c(60,80),
     ylab = "height of child", xlab = "height of parents")
```

```
abline(lm(heights$child_ht~heights$parent_ht))
abline(0,1,col = "red")
```



Show Overhead B Details. Read and examine data structure

The data has already been read into a dataset called `heights`. Examine the structure of the data with the function `str()` and use the `head()` command to look at the first few records.

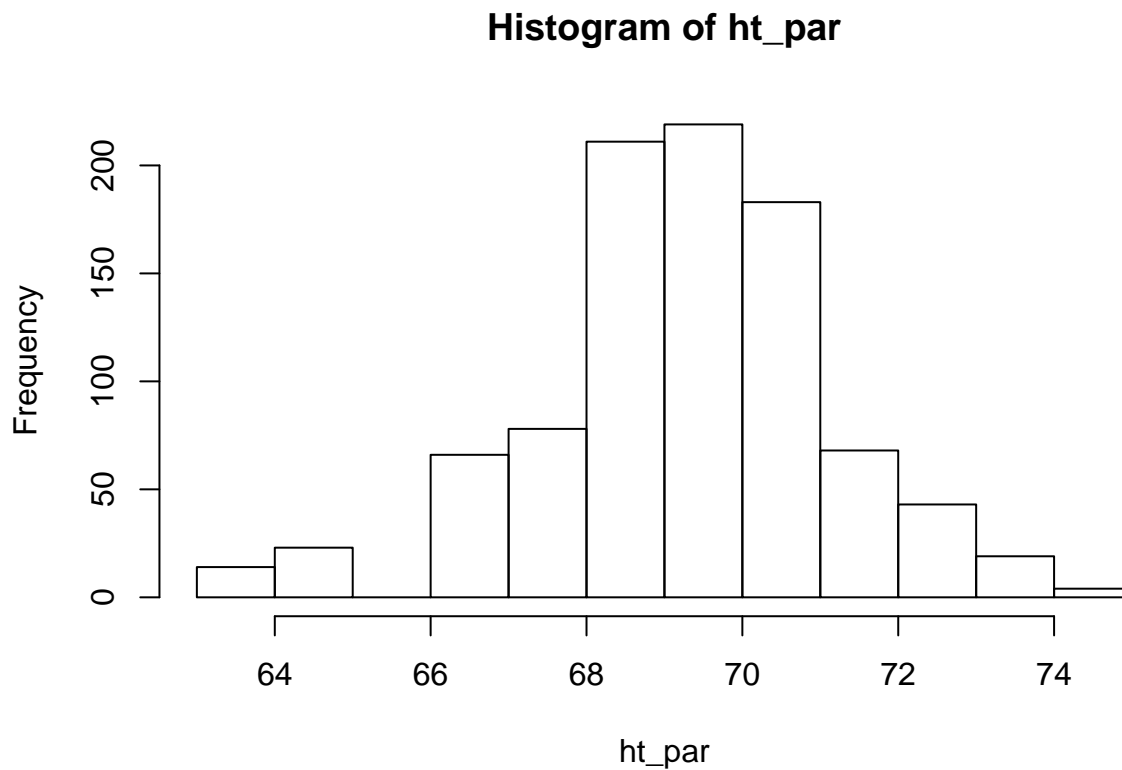
```
heights <- read.csv("CSVData\\galton_height.csv",header = TRUE)
#heights <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/c85ede6c205d220...")
str(heights)
head(heights)
```

```
'data.frame':  928 obs. of  2 variables:
 $ child_ht : num  72.2 73.2 73.2 73.2 68.2 ...
 $ parent_ht: num  74.5 74.5 74.5 74.5 73.5 73.5 73.5 73.5 73.5 73.5 ...
  child_ht parent_ht
1    72.2    74.5
2    73.2    74.5
3    73.2    74.5
4    73.2    74.5
5    68.2    73.5
6    69.2    73.5
```

Show Overhead C Details. Summary stats for parents' height

Next, examine the distribution of the child's height and then examine the distribution of the parents height.

```
ht_par <- heights$parent_ht  
hist(ht_par)
```

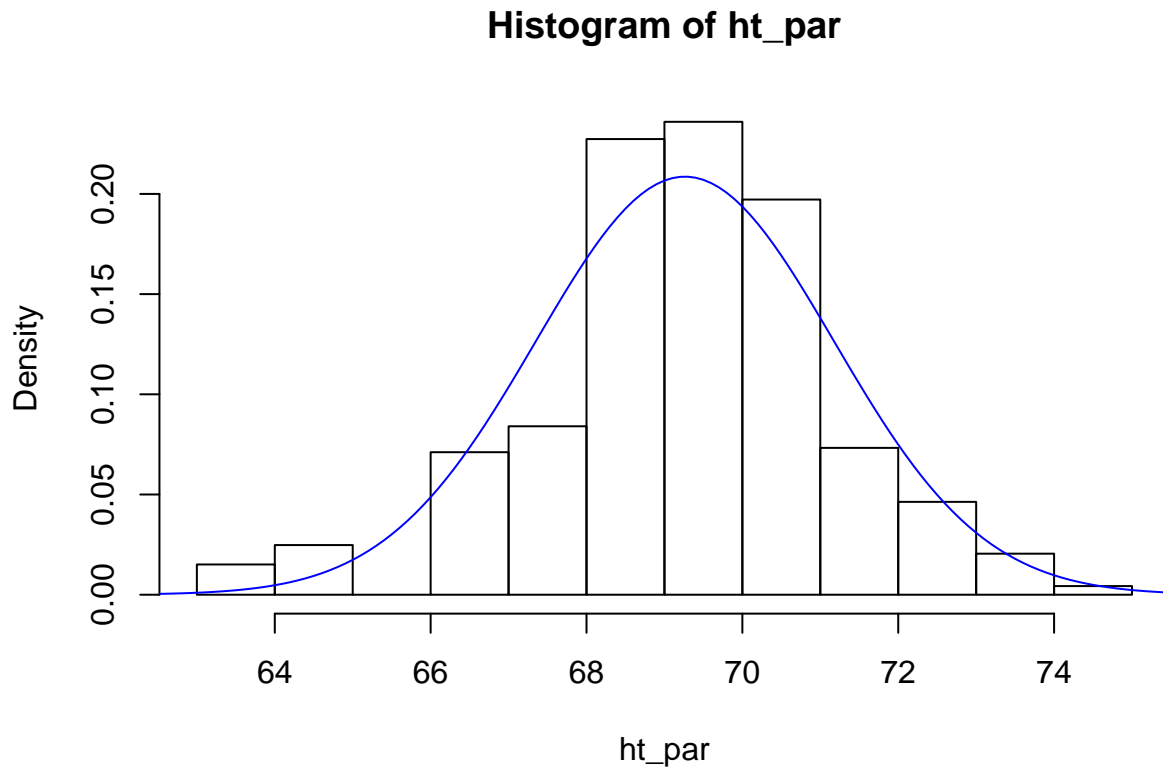


```
mean(ht_par)  
sd(ht_par)
```

```
[1] 69.26293  
[1] 1.912274
```

Show Overhead D. Fit a normal curve to parents' height details

```
(mparent <- mean(ht_par))  
(sdparent <- sd(ht_par))  
x <- seq(60, 80, by = 0.1)  
hist(ht_par, freq = FALSE)  
lines(x, dnorm(x, mean = mparent, sd = sdparent), col = "blue")
```



```
[1] 69.26293
[1] 1.912274
```

Show Overhead E Details. Use the normal approximation to determine the probability of the height of tall parents

```
TallHeight <- 72
pnorm(TallHeight, mean = mparent, sd = sdparent)
pnorm(72, mean = mean(ht_par), sd = sd(ht_par))
(StdUnitsTallHeight <- (TallHeight - mparent)/sdparent)
pnorm(StdUnitsTallHeight, mean = 0, sd = 1)
```

```
[1] 0.9238302
[1] 0.9238302
[1] 1.431317
[1] 0.9238302
```

### 1.1.2 Exercise. Fitting Galton's height data

#### Assignment Text

The Galton data has already been read into a dataframe called `heights`. These data include the heights of 928 adult children `child_ht`, together with an index of their parents' height `parent_ht`. The video explored the distribution of the parents' height; in this assignment, we investigate the distribution of the heights of the adult children.

#### Instructions

- Define the height of an adult child as a global variable
- Use the function `mean()` to calculate the mean and the function `sd()` to calculate the standard deviation
- Use the normal approximation and the function `pnorm()` to determine the probability that an adult child's height is less than 72 inches

**Hint.** Remember that we can reference a variable, say `var`, from a data set such as `heights`, as `heights$var`.

eyJsYW5ndWFnZSI6InIiLCJwcmVfZXhlcmlNpc2VfY29kZSI6IiNoZWlnaHRzIDwtIHJlYWQuY3N2KFwiQ1NWRGF0YVxc2

### 1.1.3 Exercise. Visualizing child's height distribution

#### Assignment Text

As in the prior exercise, from the Galton dataset `heights`, the heights of 928 adult children have been used to create a global variable called `ht_child`. We also have basic summary statistics, the mean height `mchild` and the standard deviation of heights in `sdchild`. In this exercise, we explore the fit of the normal curve to this distribution.

#### Instructions

- To visualize the distribution, use the function `hist()` to calculate the histogram. Use the `freq = FALSE` option to give a histogram with proportions instead of counts.
- Use the function `seq()` to determine a sequence that can be used for plotting. Then, with the function `lines()`, superimpose a normal curve on the histogram
- Determine the probability that a child's height is greater than 72 inches

**Hint 1.** Use the function `dnorm()` to calculate the normal density, similar to the cumulative probabilities that you calculated using `pnorm()`

**Hint 2.** To calculate probabilities greater than an amount, simply use 1 minus the cumulative probability

#### Pre-exercise code

eyJsYW5ndWFnZSI6InIiLCJwcmVfZXhlcmlNpc2VfY29kZSI6IiNoZWlnaHRzIDwtIHJlYWQuY3N2KFwiQ1NWRGF0YVxc2

## 1.2 Visualizing distributions

---

In this section, you learn how to:

- Calculate and interpret distributions using histograms
  - Calculate and interpret distributions using density plots
- 

### 1.2.1 Video

#### Video Overhead Details

Show Overhead Details. Data description

---

For our first look at an insurance data set, we consider data from Rempala and Derrig (2005). They considered claims arising from automobile bodily injury insurance coverages. These are amounts incurred for outpatient medical treatments that arise from automobile accidents, typically sprains, broken collarbones and the like. The data consists of a sample of 272 claims from Massachusetts that were closed in 2001 (by

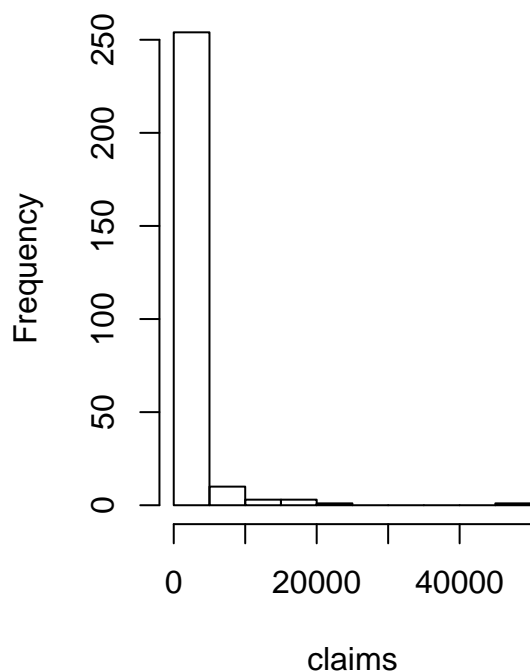
“closed,” we mean that the claim is settled and no additional liabilities can arise from the same accident). Rempala and Derrig were interested in developing procedures for handling mixtures of “typical” claims and others from providers who reported claims fraudulently. For this sample, we consider only those typical claims, ignoring the potentially fraudulent ones.

```
# Reformat Data Set
injury <- read.csv("CSVData\\MassBodilyInjury.csv",header = TRUE)
str(injury)
head(injury)
# PICK THE SUBSET OF THE DATA CORRESPONDING TO PROVIDER A
injury2 <- subset(injury, providerA != 0 )
injury2$claims <- 1000*injury2$claims
injury2$logclaims <- log(injury2$claims)
injury3 <- injury2[c("claims","logclaims")]
#write.csv(injury3,"CSVData\\MassBI.csv",row.names = FALSE)
```

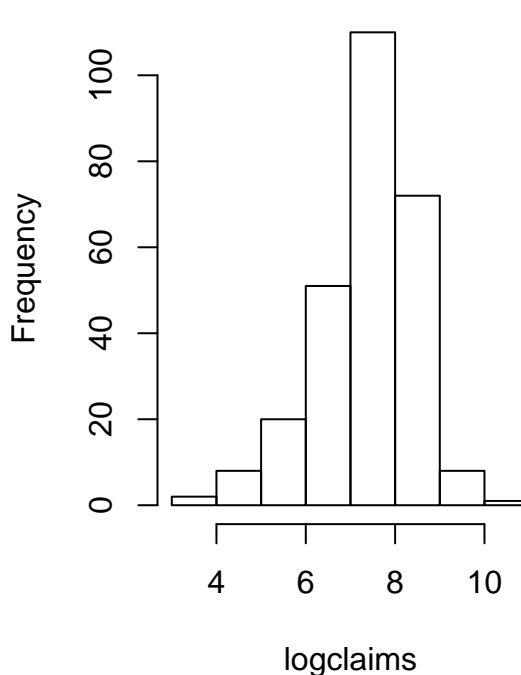
Show Overhead A Details. Bring in Data, Introduce Logarithmic Claims

```
injury <- read.csv("CSVData\\MassBI.csv",header = TRUE)
# CHECK THE NAMES, DIMENSION IN THE FILE AND LIST THE FIRST 8 OBSERVATIONS ;
str(injury)
head(injury)
attach(injury)
claims <- injury$claims
par(mfrow = c(1, 2))
hist(claims)
hist(logclaims)
```

**Histogram of claims**



**Histogram of logclaims**

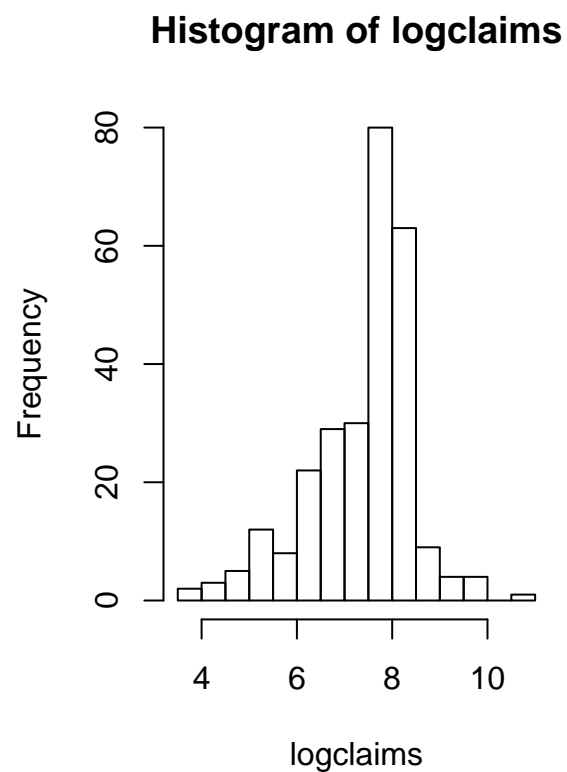
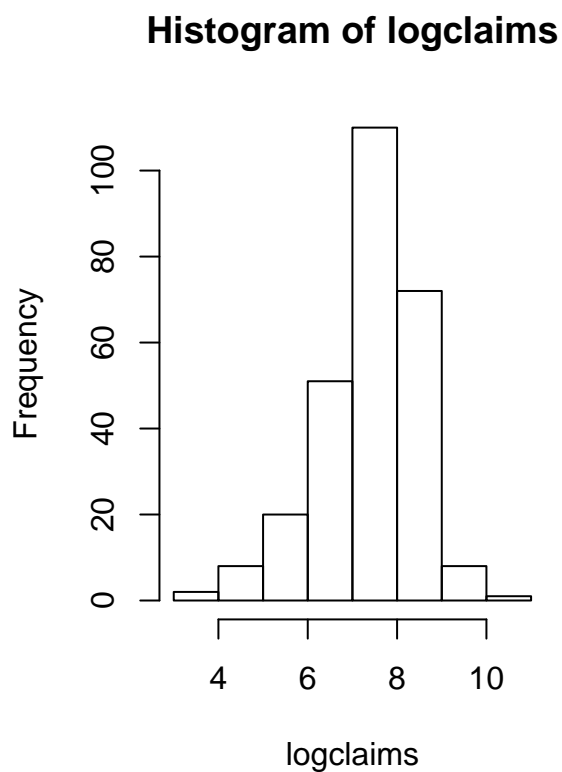




```
'data.frame': 272 obs. of 2 variables:
 $ claims : int 45 47 70 75 77 92 117 117 140 145 ...
 $ logclaims: num 3.81 3.85 4.25 4.32 4.34 ...
  claims logclaims
1     45  3.806662
2     47  3.850148
3     70  4.248495
4     75  4.317488
5     77  4.343805
6     92  4.521789
```

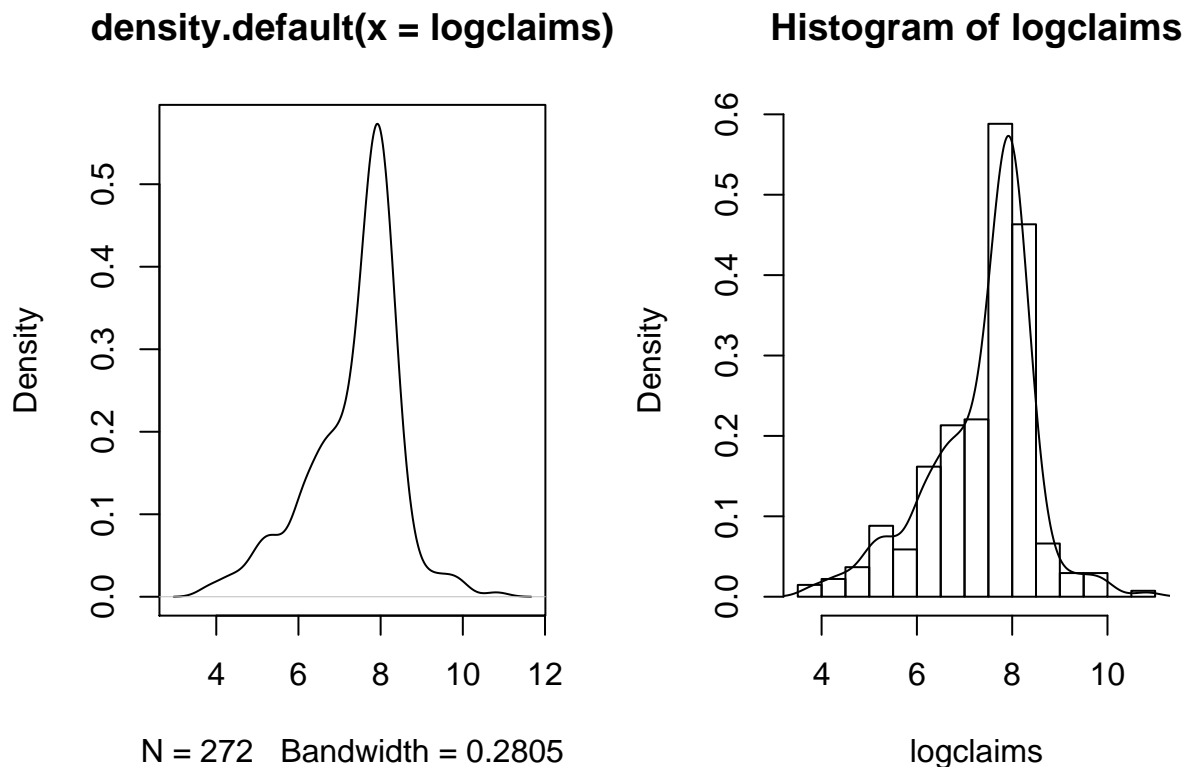
Show Overhead B Details. Show how to get a finer grid for histograms

```
par(mfrow = c(1, 2))
hist(logclaims)
hist(logclaims, breaks = 15)
```



Show Overhead C Details. Introduce the density plot

```
par(mfrow = c(1, 2))
plot(density(logclaims))
hist(logclaims, breaks = 15, freq = FALSE)
lines(density(logclaims))
```



### 1.2.2 Exercise. Visualizing bodily injury claims with density plots

#### Assignment Text

In the prior video, you learned about the Massachusetts bodily injury dataset. This dataframe, `injury`, has been read in and the global variable `claims` has been created. This assignment reviews the `hist()` function for visualizing distributions and allows you to explore density plotting, a smoothed version of the histogram.

#### Instructions

- Use the function `log()` to create the logarithmic version of the claims variable
- Calculate a histogram of logarithmic with 40 bins using an option in the `hist()` function, `breaks =`.
- Create a density plot of logarithmic claims using the functions `plot()` and `density()`.
- Repeat the density plot, this time using a more refined bandwidth equal to 0.03. Use an option in the `density()` function, `bw =`.

eyJsYW5ndWFnZSI6InIiLCJwcmVfZXhlcmNpc2VfY29kZSI6LiNpbmp1cnkgPC0gcmVhZC5jc3YoXCJDU1ZEYXRhXFxcXE

## 1.3 Summarizing distributions

---

In this section, you learn how to:

- Calculate and interpret basic summary statistics
- Calculate and interpret distributions using boxplots

- Calculate and interpret distributions using qq plots
- 

### 1.3.1 Video

#### Video Overhead Details

Show Overhead A Details. Summary statistics

```
injury <- read.csv("CSVData\\MassBI.csv",header = TRUE)
#injury <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/8cca19d0503fcf6e...
attach(injury)

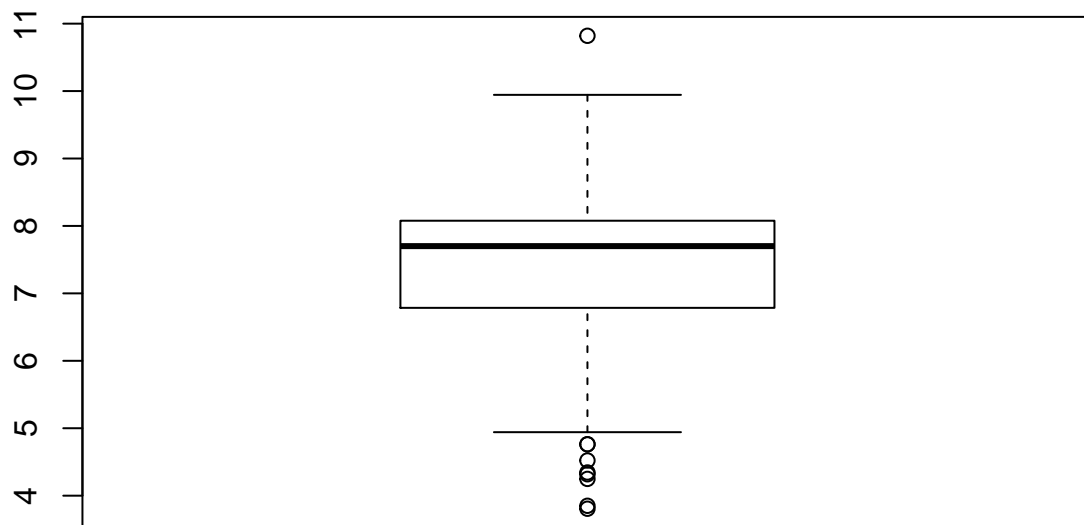
# SUMMARY STATISTICS
summary(injury)
sd(claims);sd(logclaims)
length(claims)
```

claims		logclaims	
Min.	: 45.0	Min.	: 3.807
1st Qu.:	892.5	1st Qu.:	6.794
Median :	2210.0	Median :	7.701
Mean :	2697.7	Mean :	7.388
3rd Qu.:	3215.0	3rd Qu.:	8.076
Max. :	50000.0	Max. :	10.820

```
[1] 3944.445
[1] 1.10093
[1] 272
```

Show Overhead B Details. Boxplot

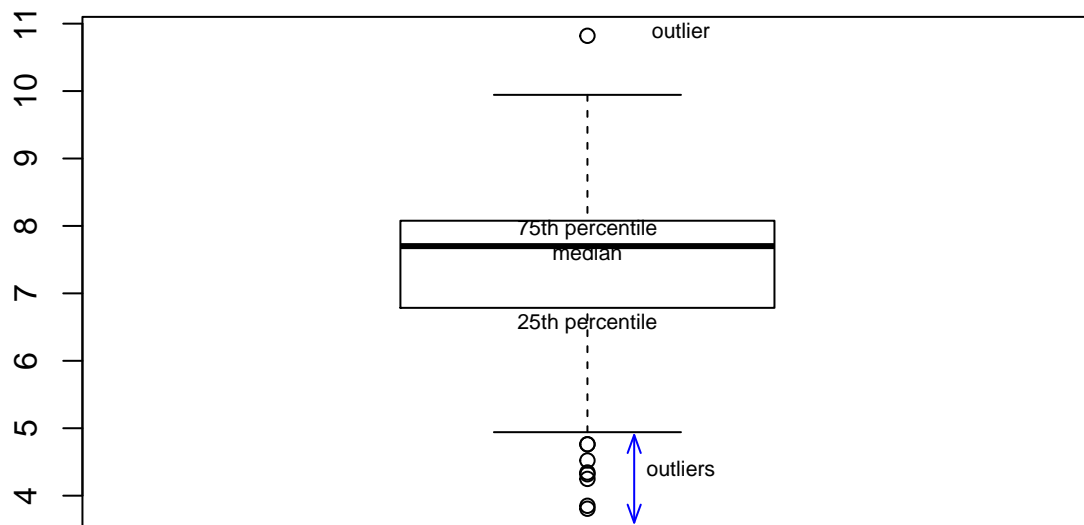
```
# BASIC BOXPLOT
boxplot(logclaims)
```



```
quantile(logclaims, probs = 0.75)

# BOXPLOT WITH ANNOTATION
boxplot(logclaims, main = "Boxplot of logclaims")
text(1, 7.6, "median", cex = 0.7)
text(1, 6.55, "25th percentile", cex = 0.7)
text(1, 7.95, "75th percentile", cex = 0.7)
arrows(1.05, 4.9, 1.05, 3.6, col = "blue", code = 3, angle = 20, length = 0.1)
text(1.1, 4.4, "outliers", cex = 0.7)
text(1.1, 10.9, "outlier", cex = 0.7)
```

### Boxplot of logclaims

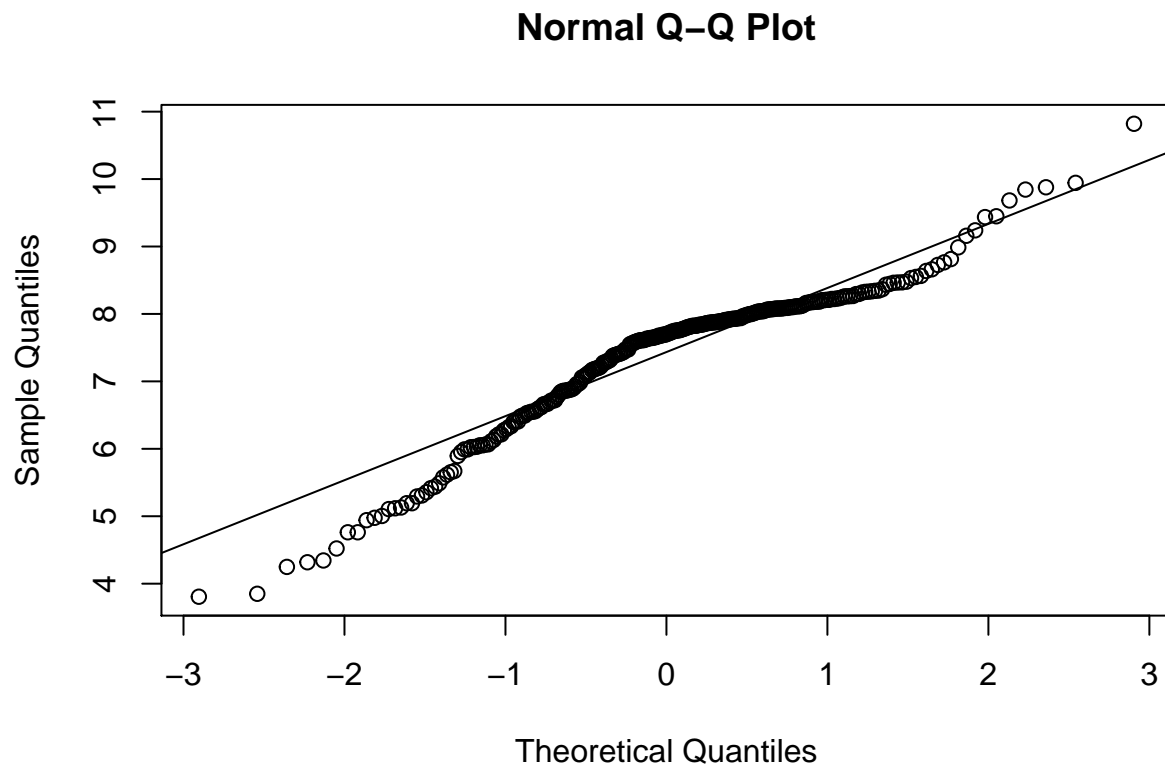


75%  
8.075579

Show Overhead C Details. QQ Plot

```
summary(injury)
quantile(claims, probs = 0.75)
quantile(logclaims, probs = 0.75)
log(quantile(claims, probs = 0.75))
qnorm(p = 0.75, mean = mean(logclaims), sd = sd(logclaims))
(qnorm(p = 0.75, mean = mean(logclaims), sd = sd(logclaims)) - mean(logclaims)) /
  sd(logclaims)
qnorm(p = 0.75, mean = 0, sd = 1)

# QUANTILE - QUANTILE PLOT
qqnorm(logclaims)
qqline(logclaims)
```



claims	logclaims
Min. : 45.0	Min. : 3.807
1st Qu.: 892.5	1st Qu.: 6.794
Median : 2210.0	Median : 7.701
Mean : 2697.7	Mean : 7.388
3rd Qu.: 3215.0	3rd Qu.: 8.076
Max. : 50000.0	Max. : 10.820

```

75%
3215
75%
8.075579
75%
8.075583
[1] 8.131056
[1] 0.6744898
[1] 0.6744898

```

### 1.3.2 Exercise. Summarizing bodily injury claims with box and qq plots

#### Assignment Text

The Massachusetts bodily injury data has already been read and used to create the global variable `claims` representing bodily injury claims. The previous video showed how to present the distribution of logarithmic claims which appeared to be approximately normally distributed. However, users are not really interested in log dollars but want to know about a unit of measurement that is more intuitive, such as dollars.

So this assignment is based on claims, not the logarithmic version. You will use the functions `boxplot()` and `qqnorm()` to visualize the distribution through boxplots and quantile-quantile, or qq-, plots. But, because we are working with such a skewed distribution, do not be surprised that it is difficult to interpret these results readily.

### Instructions

- Produce a box plot for claims
- Determine the 25th empirical percentile for claims using the `quantile()` function.
- Determine the 25th percentile for claims based on a normal distribution using the `qnorm()` function.
- Produce a normal qq plot for claims using the function `qqnorm()`. The `qqline()` function is handy for producing a reference line.

**Hint.** Note that `qnorm()` (one q) is for a normal quantile and `qqnorm()`. (two q's!) is for the normal qq plot

eyJsYW5ndWFnZSI6InIiLCJwcmVfZXhlcmlNpc2VfY29kZSI6LiNpbmp1cnkgPC0gcmVhZC5jc3YoXCJDU1ZEYXRhXFxcXE

### 1.3.3 Exercise. Effects on distributions of removing the largest claim

#### Assignment Text

The Massachusetts bodily injury dataframe `injury` has been read in; our focus is on the `claims` variable in that dataset.

In the previous exercise, we learned that the Massachusetts bodily injury `claims` distribution was not even close to approximately normal (as evidenced by the box and qq- plots). Non-normality may be induced by skewness (that we will handle via transformations in the next section). But, seeming non-normality can also be induced by one or two very large observations (called an outlier later in the course). So, this exercise examines the effects on the distribution of removing the largest claims.

### Instructions

- Use the function `tail()` to examine the `injury` dataset and identify the largest claim
- Use the function `subset()` to create a subset omitting the largest claim
- Compare the summary statistics of the omitted claim distribution to the full distribution
- Compare the two distributions visually via histograms plotted next to another. `par(mfrow = c(1, 2))` is used to organize the plots you create. Do not alter this code.

**Hint.** For this data set, the `[subset()]` argument `claims < 25000` will keep all but the largest claim

eyJsYW5ndWFnZSI6InIiLCJwcmVfZXhlcmlNpc2VfY29kZSI6LiNpbmp1cnkgPC0gcmVhZC5jc3YoXCJDU1ZEYXRhXFxcXE

## 1.4 Transformations

---

In this exercise, you learn how to:

- Symmetrize a skewed distribution using a logarithmic transformation
- 

### 1.4.1 Video

#### Video Overhead Details

Show Overhead A Details. Simulate a moderately skewed distribution, with transforms

```
# FIGURE 1.7 - SIMULATE CHI-SQUARE, CREATE 3 TRANSFORMATIONS
set.seed(1237)                # set the seed of the random number generator
                                # allows us to replicate results
X1 <- 10000*rchisq(500, df = 2) # generate variables randomly from a skewed distribution
X2 <- X1^(0.5)                 # square root transform, could also use sqrt(X1)
X3 <- log(X1)                  # logarithmic transform
X4 <- -1/X1                    # negative reciprocal transform
```

Show Overhead B Details. Visualize the distributions

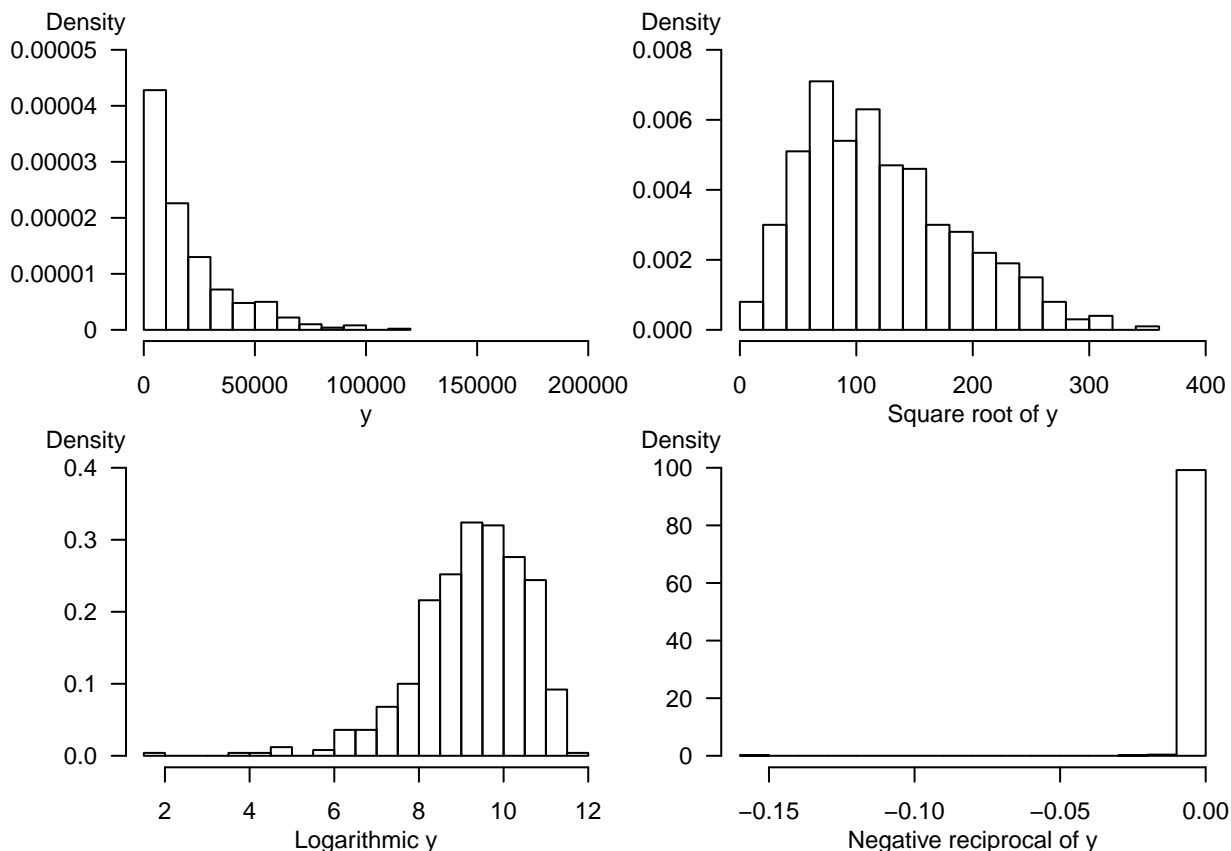
```
par(mfrow = c(2, 2), cex = .75, mar = c(3,5,1.5,0))
hist(X1, freq = FALSE, nclass = 16, main = "", xlab = "", ylab = "",
     las = 1, yaxt = "n", xlim = c(0,200000), ylim = c(0,.00005))
axis(2, at = seq(0,.00005,.00001), las = 1, cex = .3,
     labels = c("0", "0.00001", "0.00002", "0.00003", "0.00004", "0.00005"))
mtext("Density", side = 2, at = .000055, las = 1, cex = .75)
mtext("y", side = 1, cex = .75, line = 2)

par(mar = c(3,4,1.5,0.2))
hist(X2, freq = FALSE, nclass = 16, main = "", xlab = "", ylab = "",
     las = 1, xlim = c(0,400), ylim = c(0,.008))
mtext("Density", side = 2, at = .0088, las = 1, cex = .75)
mtext("Square root of y", side = 1, cex = .75, line = 2)

par(mar = c(3.2,5,1,0))
hist(X3, freq = FALSE, nclass = 16, main = "", xlab = "", ylab = "", las = 1, ylim = c(0,.4))
mtext("Density", side = 2, at = .44, las = 1, cex = .75)
mtext("Logarithmic y", side = 1, cex = .75, line = 2)

par(mar = c(3.2,4,1,0.2))
hist(X4, freq = FALSE, nclass = 16, main = "", xlab = "", ylab = "", las = 1, ylim = c(0,100))
mtext("Density", side = 2, at = 110, las = 1, cex = .75)
mtext("Negative reciprocal of y", side = 1, cex = .75, line = 2)
```





### 1.4.2 Exercise. Distribution of transformed bodily injury claims

#### Assignment Text

We have now examined the distributions of bodily injury claims and its logarithmic version. Grudgingly, we have concluded that to fit a normal curve the logarithmic version of claims is a better choice (again, we really do not like log dollars but you'll get used to it in this course). But, why logarithmic and not some other transformations?

A partial response to this question will appear in later chapters when we describe interpretation of regression coefficients. Another partial response is that the log transform seems to work well with skewed insurance data sets, as we demonstrate visually in this exercise.

#### Instructions

Use the code `par(mfrow = c(2, 2))` so that four graphs appear in a 2 by 2 matrix format for easy comparisons. Plot the `density()` of

- `claims`
- `square root of claims`
- `logarithmic claims`
- `negative reciprocal of claims`

**Hint.** For negative reciprocal claims, use `plot(density(-claims$(-1)))`

eyJsYW5ndWFnZSI6InIiLCJwcmVfZXhlcmlNpc2VfY29kZSI6LiNpbmp1cnkgPC0gcmlVhZC5jc3YoXCJDU1ZEYXRhXFxcXE



# Chapter 2

## Basic Linear Regression

### Chapter description

This chapter considers regression in the case of only one explanatory variable. Despite this seeming simplicity, many deep ideas of regression can be developed in this framework. By limiting ourselves to the one variable case, we can illustrate the relationships between two variables graphically. Graphical tools prove to be important for developing a link between the data and a predictive model.

### 2.1 Correlation

---

In this section, you learn how to:

- Calculate and interpret a correlation coefficient
  - Interpret correlation coefficients by visualizing scatter plots
- 

#### 2.1.1 Video

##### Video Overhead Details

Show Overhead A Details. Wisconsin lottery data description

```
Lot <- read.csv("CSVData\\Wisc_lottery.csv")
#Lot <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/a792b30fb32b0896dd6...")
str(Lot)
```

Show Overhead B Details. Summary statistics

```
#options(scipen = 100, digits = 4)
#numSummary(Lot[,c("pop", "sales")], statistics = c("mean", "sd", "quantiles"), quantiles = c(0,.5,1))
(as.data.frame(psych::describe(Lot)))[,c(2,3,4,5,8,9)]
#Rcmdr::numSummary(Lot[,c("pop", "sales")], statistics = c("mean", "sd", "quantiles"), quantiles = c(0,
```

Show Overhead C Details. Visualizing skewed distributions

```
par(mfrow = c(1, 2))
hist(Lot$pop, main = "", xlab = "population")
hist(Lot$sales, main = "", xlab = "sales")
```

Show Overhead D Details. Visualizing relationships with a scatter plot

```
plot(Lot$pop, Lot$sales, xlab = "population", ylab = "sales")
```

Show Overhead E Details. Correlation coefficient

```
cor(Lot$pop, Lot$sales)
```

## 2.1.2 Exercise. Correlations and the Wisconsin lottery

### Assignment Text

The Wisconsin lottery dataset, `Wisc_lottery`, has already been read into a dataframe `Lot`.

Like insurance, lotteries are uncertain events and so the skills to work with and interpret lottery data are readily applicable to insurance. It is common to report sales and population in thousands of units, so this exercise gives you practice in rescaling data via linear transformations.

### Instructions

- From the available population and sales variables, create new variables in the dataframe `Lot`, `pop_1000` and `sales_1000` that are in thousands (of people and of dollars, respectively).
- Create summary statistics for the dataframe that includes these new variables.
- Plot `pop_1000` versus `sales_1000`.
- Calculate the correlation between `pop_1000` versus `sales_1000` using the function `cor()`. How does this differ between the correlation between population and sales in the original units?

### Hint

Use the dataframe to refer to pop and sales as `Lot$pop` and `Lot$sales`, respectively

### Pre-exercise code

```
# Pre-exercise code
#library(Rcmdr)
#library(psych)
Lot <- read.csv("CSVData\\Wisc_lottery.csv", header = TRUE)
#Lot <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/a792b30fb32b0896dd6")
```

### Sample\_code

```
`@Sample_code`
# Create new variables, say, `pop_1000` and `sales_1000`
Lot$pop_1000 <- ___
___ <- Lot$sales/1000

# Create summary statistics for the dataframe
summary(___ )

# Plot `pop_1000` versus `sales_1000`.
plot(___, ___)

# Calculate the correlation between `pop_1000` versus `sales_1000`
cor(___, ___)
```

**Solution**

```
# Solution
Lot$pop_1000 <- Lot$pop/1000
Lot$sales_1000 <- Lot$sales/1000
summary(Lot)
#(as.data.frame(psych::describe(Lot)))[,c(2,3,4,5,8,9)]
#(as.data.frame(psych::describe(Lot[,c("pop_1000", "sales_1000")])))[,c(2,3,4,5,8,9)]
#Rcmdr::numSummary(Lot[,c("pop_1000", "sales_1000")], statistics = c("mean", "sd", "quantiles"), quanti
plot(Lot$pop_1000, Lot$sales_1000)
cor(Lot$pop_1000, Lot$sales_1000)
```

**Submission Correctness Tests (SCT)**

test\_error() test\_object("Lot", incorrect\_msg = "Looks like a variable is defined incorrectly. The hint may help.") success\_msg("Congratulations! We will rescale data using 'linear' transformations regularly. In part we do this for communicating our analysis to others. Also in part, this is for our own convenience as it can allow us to see patterns more readily.")

## 2.2 Method of least squares

---

In this section, you learn how to:

- Fit a line to data using the method of least squares
  - Predict an observation using a least squares fitted line
- 

### 2.2.1 Video

**Video Overheads**

Show Overhead A Details. Where to fit the line?

```
model_blr <- lm(sales ~ pop, data = Lot)
plot(Lot$pop, Lot$sales, xlab = "population", ylab = "sales")
abline(model_blr, col="blue")
abline(0,1, col="red")
```

Show Overhead B Details. Method of least squares

- For observation  $\{(y, x)\}$ , the height of the regression line is

$$b_0 + b_1x.$$

- Thus,  $y - (b_0 + b_1x)$  represents the deviation.
- The sum of squared deviations is

$$SS(b_0, b_1) = \sum (y - (b_0 + b_1x))^2.$$

- The method of least squares – determine values of  $b_0, b_1$  that minimize  $SS$ .

Show Overhead C Details. Regression coefficients

```
model_blr <- lm(sales ~ pop, data = Lot)
round(coefficients(model_blr), digits=4)
plot(Lot$pop, Lot$sales, xlab = "population", ylab = "sales")
abline(model_blr, col="blue")
```

Show Overhead D Details. Prediction

```
round(coefficients(model_blr), digits=6)
coefficients(model_blr)[1] + coefficients(model_blr)[2]*30000

newdata <- data.frame(pop = 30000)
predict(model_blr, newdata)
```

## 2.2.2 Exercise. Least squares fit using housing prices

### Assignment Text

The prior video analyzed the effect that a zip code's population has on lottery sales. Instead of population, suppose that you wish to understand the effect that housing prices have on the sale of lottery tickets. The dataframe `Lot`, read in from the Wisconsin lottery dataset `Wisc_lottery`, contains the variable `medhome` which is the median house price for each zip code, in thousands of dollars. In this exercise, you will get a feel for the distribution of this variable by examining summary statistics, examine its relationship with sales graphically and via correlations, fit a basic linear regression model and use this model to predict sales.

### Instructions

- Summarize the dataframe `Lot` that contains `medhome` and `sales`.
- Plot `medhome` versus `sales`. Summarize this relationship by calculating the corresponding correlation coefficient using the function `cor()`.
- Using the function `lm()`, regress `medhome`, the explanatory variable, on `sales`, the outcome variable. Display the regression coefficients to four significant digits.
- Use the function `predict()` and the fitted regression model to predict sales assuming that the median house price for a zip code is 50 (in thousands of dollars).

### Hint

#### Pre-exercise code

```
# Pre-exercise code
#library(Rcmdr)
Lot <- read.csv("CSVData\\Wisc_lottery.csv", header = TRUE)
#Lot <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/a792b30fb32b0896dd6")
```

#### Sample\_code

```
`@Sample_code`
# Summarize the dataframe `Lot` that contains `medhome` and `sales`
summary(Lot)
# Plot and calculate the correlation of `medhome` versus `sales`.
cor(___, ___)
plot(___, ___)

# Regress `medhome` on `sales`. Display the regression coefficients to four significant digits.
model_blr1 <- lm(___ ~ ___, data = Lot)
round(coefficients(model_blr1), digits= ---)

# Predict sales assuming that the median house price is 50
```

```
newdata <- data.frame(medhome = ___)
predict(model_blr1, newdata)
```

### Solution

```
# Solution
#numSummary(Lot[,c("medhome", "sales")], statistics = c("mean", "sd", "quantiles"), quantiles = c(0,.5,
#(as.data.frame(psych::describe(Lot[,c("medhome", "sales")])))[,c(2,3,4,5,8,9)]
summary(Lot)
cor(Lot$medhome, Lot$sales)
plot(Lot$medhome, Lot$sales)
model_blr1 <- lm(sales ~ medhome, data = Lot)
round(coefficients(model_blr1), digits=4)
newdata <- data.frame(medhome = 50)
predict(model_blr1, newdata)
```

### Submission Correctness Tests (SCT)

```
test_error() test_object("model_blr1", incorrect_msg = "The basic linear regression model is incor-
rectly specified.") test_object("newdata", incorrect_msg = "The new data is incorrectly specified.")
success_msg("Congratulations! You now have experience fitting a regression line and using this line for
predictions, just as Galton did when he used parents' heights to predict the height of an adult child. Well
done!")
```

## 2.3 Understanding variability

---

In this section, you learn how to:

- Visualize the ANOVA decomposition of variability
  - Calculate and interpret  $R^2$ , the coefficient of determination
  - Calculate and interpret  $s^2$  the mean square error
  - Explain the components of the ANOVA table
- 

### 2.3.1 Video

#### Video Overhead Details

Show OverheadS A and B Details. Visualizing the uncertainty about a line

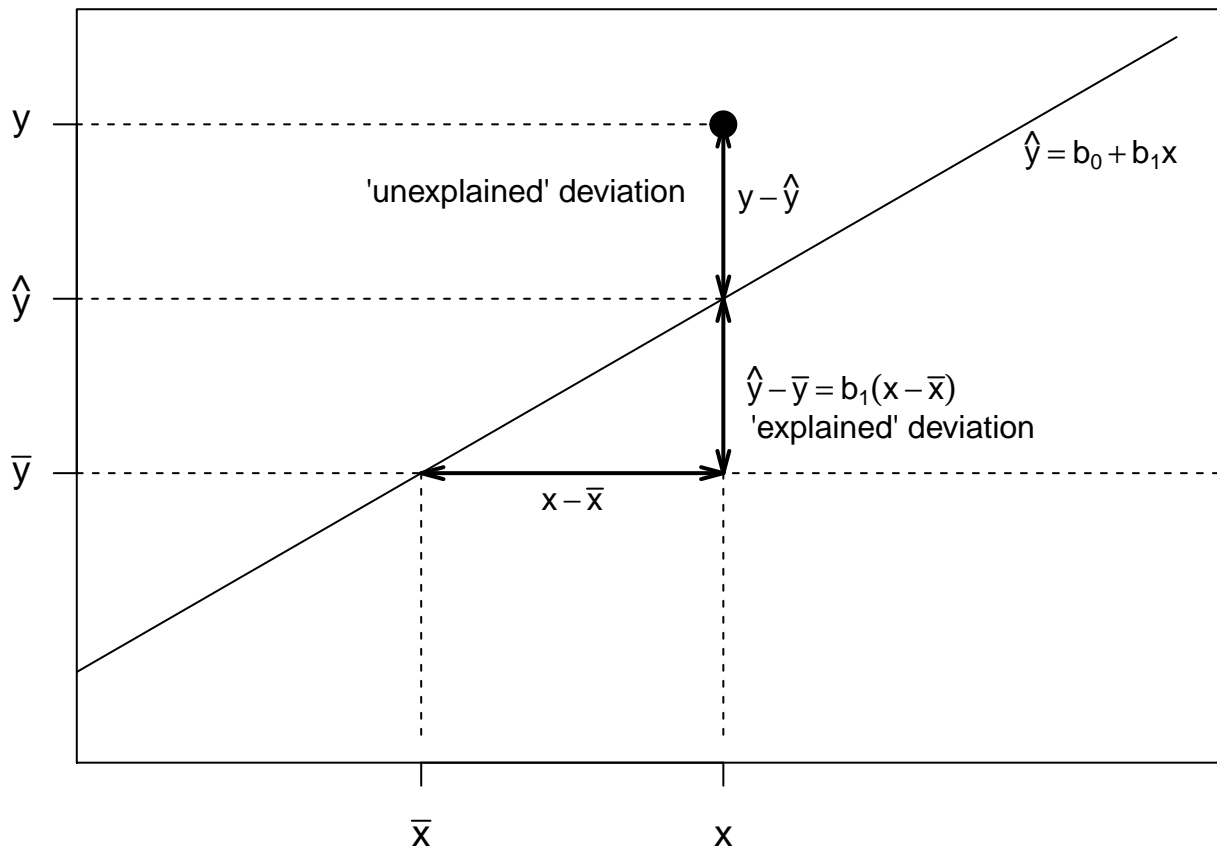
```
par(mar=c(2.2,2.1,.2,.2),cex=1.2)
x <- seq(-4, 4, len=101)
y <- x
plot(x, y, type = "l", xlim=c(-3, 4), xaxt="n", yaxt="n", xlab="", ylab="")
axis(1, at = c(-1, 1), lab = expression(bar(x), x))
axis(2, at = c(-1, 1, 3), lab = expression(bar(y), hat(y), y), las=1)
abline(-1, 0, lty = 2)
segments(-4, 1, 1, 1, lty=2)
segments(-4, 3, 1, 3, lty = 2)
segments(1, -4, 1, 3, lty = 2)
segments(-1, -4, -1, -1, lty = 2)
```

```

points(1, 3, cex=1.5, pch=19)

arrows(1.0, 1, 1.0, 3, code = 3, lty = 1, angle=15, length=0.12, lwd=2)
text(1.3, 2.2, expression( y-hat(y)), cex=0.8)
text(-.3, 2.2, "'unexplained' deviation", cex=.8)
arrows(1.0, -1, 1.0, 1, code = 3, lty = 1, angle=15, length=0.12, lwd=2)
text(1.85, 0, expression(hat(y)-bar(y) == b[1](x-bar(x)) ), cex=0.8 )
text(2.1, -0.5, " 'explained' deviation", cex=0.8)
arrows(-1, -1.0, 1, -1.0, code = 3, lty = 1, angle=15, length=0.12, lwd = 2)
text(0, -1.3, expression( x-bar(x)), cex=0.8 )
text(3.5, 2.7, expression( hat(y) == b[0] + b[1]*x), cex=0.8 )

```



Show Overheads C, D and E Details. ANOVA Table

```

model_blr <- lm(sales ~ pop, data = Lot)
anova(model_blr)
sqrt(anova(model_blr)$Mean[2])
summary(model_blr)$r.squared

```

### 2.3.2 Exercise. Summarizing measures of uncertainty

#### Assignment Text

In a previous exercise, you developed a regression line to fit the variable `medhome`, the median house price for each zip code, as a predictor of lottery sales. The regression of `medhome` on `sales` has been summarized



in the R object `model_blr`.

How reliable is the regression line? In this exercise, you will compute some of the standard measures that are used to summarize the goodness of this fit.

### Instructions

- Summarize the fitted regression model in an ANOVA table.
- Determine the size of the typical residual,  $s$ .
- Determine the coefficient of determination,  $R^2$ .

### Hint

Learn more about possibilities through the `Rdocumentation` site. If you have not done so already, check out the function `anova()`

### Pre-exercise code

```
# Pre-exercise code
Lot <- read.csv("CSVData\\Wisc_lottery.csv", header = TRUE)
#Lot <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/a792b30fb32b0896dd6
```

### Sample\_code

```
`@Sample_code`
model_blr <- lm(sales ~ medhome, data = Lot)

# Summarize the fitted regression model in an ANOVA table.
anova(____)

# Determine the size of the typical residual, $s$.
sqrt(anova(____)$Mean[2])

# Determine the coefficient of determination, $R^2$.
summary(____)$r.squared
```

### Solution

```
# Solution
model_blr <- lm(sales ~ medhome, data = Lot)
anova(model_blr)
sqrt(anova(model_blr)$Mean[2])
summary(model_blr)$r.squared
```

### Submission Correctness Tests (SCT)

```
test_error() test_object("model_blr", incorrect_msg = "The basic linear regression model is incorrectly specified.") success_msg("Congratulations! It will be helpful if you compare the results of this exercise to the regression of pop on sales from the prior video. We have seen that pop is more highly correlated with sales than medhome, so we are expecting greater uncertainty in this regression fit.")
```

## 2.3.3 Exercise. Effects of linear transforms on measures of uncertainty

### Assignment Text

Let us see how rescaling, a linear transformation, affects our measures of uncertainty. As before, the Wisconsin lottery dataset `Wisc_lottery` has been read into a dataframe `Lot` that also contains `sales_1000`, sales in thousands of dollars, and `pop_1000`, zip code population in thousands. How do measures of uncertainty change when going from the original units to thousands of those units?

## Instructions

- Run a regression of `pop` on `sales_1000` and summarize this in an ANOVA table.
- For this regression, determine the  $s$  and the coefficient of determination,  $R^2$ .
- Run a regression of `pop_1000` on `sales_1000` and summarize this in an ANOVA table.
- For this regression, determine the  $s$  and the coefficient of determination,  $R^2$ .

## Hint

The residual standard error is also available as `summary(model_blr1)$sigma`. The coefficient of determination is also available as `summary(model_blr1)$r.squared`.

## Pre-exercise code

```
# Pre-exercise code
#library(Rcmdr)
Lot <- read.csv("CSVData\\Wisc_lottery.csv", header = TRUE)
#Lot <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/a792b30fb32b0896dd6...")
Lot$pop_1000 <- Lot$pop/1000
Lot$sales_1000 <- Lot$sales/1000
```

## Sample\_code

```
`@Sample_code`
# Run a regression of `pop` on `sales_1000` and summarize this in an ANOVA table.
model_blr1 <- lm(sales_1000 ~ pop, data = Lot)
anova(____)

# Determine the $s$ and the coefficient of determination, $R^2$.
sqrt(anova(____)$Mean[2])
summary(____)$r.squared

# Run a regression of `pop_1000` on `sales_1000` and summarize this in an ANOVA table.
model_blr2 <- lm(____ ~ ____, data = Lot)
anova(model_blr2)

# Determine the $s$ and the coefficient of determination, $R^2$.
---
---
```

## Solution

```
# Solution
model_blr1 <- lm(sales_1000 ~ pop, data = Lot)
anova(model_blr1)
sqrt(anova(model_blr1)$Mean[2])
summary(model_blr1)$r.squared
model_blr2 <- lm(sales_1000 ~ pop_1000, data = Lot)
anova(model_blr2)
sqrt(anova(model_blr2)$Mean[2])
summary(model_blr2)$r.squared
```

## Submission Correctness Tests (SCT)

`test_error()` `test_object("model_blr1", incorrect_msg = "The basic linear regression model is incorrectly specified.")` `success_msg("Congratulations! In this exercise, you have seen that rescaling does not affect our measures of goodness of fit in any meaningful way. For example, the coefficient of determinations are`

completely unaffected. This is helpful because we will rescale variables extensively in our search for patterns in the data.”)

## 2.4 Statistical inference

---

In this section, you learn how to:

- Conduct a hypothesis test for a regression coefficient using either a rejection/acceptance procedure or a p-value
  - Calculate and interpret a confidence interval for a regression coefficient
  - Calculate and interpret a prediction interval at a specific value of a predictor variable
- 

### 2.4.1 Video

#### Video Overhead Details

Show Overhead A Details. Summary of basic linear regression model

Introduce the output in the summary of the basic linear regression model.

```
Lot <- read.csv("CSVData\\Wisc_lottery.csv", header = TRUE)
#Lot <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/a792b30fb32b0896dd6...")
#options(scipen = 8, digits = 4)
model_blr <- lm(sales ~ pop, data = Lot)
summary(model_blr)
```

Show Overhead B Details. Hypothesis testing

```
> summary(model_blr)$coefficients
      Estimate Std. Error t value Pr(>|t|)
(Intercept) 469.7036   702.90619   0.6682 5.072e-01
pop          0.6471     0.04881  13.2579 1.158e-17
```

Show Overhead C Details. Confidence intervals

```
Rcmdr::Confint(model_blr, level = .90)
Rcmdr::Confint(model_blr, level = .95)
```

Show Overhead D Details. Confidence intervals check

```
# Just for checking
summary(model_blr)$coefficients[2,1]
summary(model_blr)$coefficients[2,2]
qt(.975, 48)

summary(model_blr)$coefficients[2,1] -
  summary(model_blr)$coefficients[2,2]*qt(.975, 48)

Rcmdr::Confint(model_blr, level = .95)
confint(model_blr, level = .95)
```

Show Overhead E Details. Prediction intervals

```
NewData <- data.frame(pop = 10000)
predict(model_blr, NewData, interval = "prediction", level = .90)
predict(model_blr, NewData, interval = "prediction", level = .99)
```

## 2.4.2 Exercise. Statistical inference and Wisconsin lottery

### Assignment Text

In a previous exercise, you developed a regression line with the variable `medhome`, the median house price for each zip code, as a predictor of lottery sales. The regression of `medhome` on `sales` has been summarized in the R object `model_blr`.

This exercise allows you to practice the standard inferential tasks: hypothesis testing, confidence intervals, and prediction.

### Instructions

- Summarize the regression model and identify the t-statistic for testing the importance of the regression coefficient associated with `medhome`.
- Use the function `confint()` to provide a 95% confidence interval for the regression coefficient associated with `medhome`.
- Consider a zip code with a median housing price equal to 50 (in thousands of dollars). Use the function `predict()` to provide a point prediction and a 95% prediction interval for sales.

### Hint

Taking a `[summary()]` of a regression object produces a new object. You can use the `[str()]` structure command to learn more about the new object. Try out a command such as `str(summary(model_blr))`

### Pre-exercise code

```
# Pre-exercise code
Lot <- read.csv("CSVData\\Wisc_lottery.csv", header = TRUE)
#Lot <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/a792b30fb32b0896dd6")
```

### Sample\_code

```
`@sample_code`
model_blr1 <- lm(sales ~ medhome, data = Lot)
# Summarize the regression model and identify the $t$-statistic for testing the importance of the regression coefficient associated with `medhome`.
summary(___)
summary(___)$coefficients
summary(___)$coefficients[,3]

# Provide a 95% confidence interval for the regression coefficient associated with `medhome`.
confint(___, level = ___)

# Provide a point prediction and a 95% prediction interval for sales.
NewData1 <- data.frame(medhome = 50)
predict(___, NewData1, interval = "prediction", level = ___)
```

### Solution

```
# Solution
model_blr1 <- lm(sales ~ medhome, data = Lot)
summary(model_blr1)
summary(model_blr1)$coefficients
summary(model_blr1)$coefficients[,3]
```

```
#Rcmdr::Confint(model_blr1, level = .95)
confint(model_blr1, level = .95)

NewData1 <- data.frame(medhome = 50)
predict(model_blr1, NewData1, interval = "prediction", level = .95)
```

### Submission Correctness Tests (SCT)

test\_error() test\_object("model\_blr1", incorrect\_msg = "The basic linear regression model is incorrectly specified.") test\_object("NewData1", incorrect\_msg = "The new data object is incorrectly specified.") success\_msg("Congratulations! Much of what we learn from a data modeling exercise can be summarized using standard inferential tools: hypothesis testing, confidence intervals, and prediction.")

## 2.5 Diagnostics

In this section, you learn how to:

- Describe how diagnostic checking and residual analysis are used in a statistical analysis
- Describe several model misspecifications commonly encountered in a regression analysis

### 2.5.1 Video

#### Video Overhead Details

Show Overhead A Details. Unusual observations in regression

- We have defined regression estimates as minimizers of a least squares objective function.
- An appealing intuitive feature of linear regressions is that regression estimates can be expressed as weighted averages of outcomes.
- The weights vary by observation, some observations are more important than others.
- “Unusual” observations are far from the majority of the data set:
- Unusual in the vertical direction is called an outlier.
- Unusual in the horizontal directional is called a high leverage point.

Show Overhead B Details. Example. Outliers and High Leverage Points

```
outlr <- read.csv("CSVData\\Outlier.csv", header = TRUE)

# FIGURE 2.7
plot(outlr$x, outlr$y, xlim = c(0, 10), ylim = c(2, 9), xlab = "x", ylab = "y")
text(4.5, 8.0, "A")
text(9.8, 8.0, "B")
text(9.8, 2.5, "C")
```

Show Overhead C Details. Regression fit with 19 base observations

```
model_outlr0 <- lm(y ~ x, data = outlr, subset = -c(20,21,22))
summary(model_outlr0)
plot(outlr$x[1:19], outlr$y[1:19], xlab = "x", ylab = "y", xlim = c(0, 10), ylim = c(2, 9))
abline(model_outlr0)
```

Show Overhead D Details. Regression fit with 19 base observations plus C

```
model_outlrC <- lm(y ~ x, data = outlr, subset = -c(20,21))
summary(model_outlrC)
plot(outlr$x[c(1:19,22)], outlr$y[c(1:19,22)], xlab = "x", ylab = "y", xlim = c(0, 10), ylim = c(2, 9))
text(9.8, 2.5, "C", col = "blue")
abline(model_outlrC)
```

Show Overhead E Details. R code

```
model_outlr0 <- lm(y ~ x, data = outlr, subset = -c(20,21,22))
summary(model_outlr0)
model_outlrA <- lm(y ~ x, data = outlr, subset = -c(21,22))
summary(model_outlrA)
model_outlrB <- lm(y ~ x, data = outlr, subset = -c(20,22))
summary(model_outlrB)
model_outlrC <- lm(y ~ x, data = outlr, subset = -c(20,21))
summary(model_outlrC)
```

Show Overhead F Details. Visualizing four regression fits

```
plot(outlr$x, outlr$y, xlim = c(0, 10), ylim = c(2, 9), xlab = "x", ylab = "y")
text(4.5, 8.0, "A", col = "red")
text(9.8, 8.0, "B", col = "green")
text(9.8, 2.5, "C", col = "blue")
abline(model_outlr0)
abline(model_outlrA, col = "red")
abline(model_outlrB, col = "green")
abline(model_outlrC, col = "blue")
```

Show Overhead G Details. Results from four regression models

Results from Four Regressions

Data	$b_0$	$b_1$	$s$	$R^2(\%)$	$t(b_1)$
19 Base Points	1.869	0.611	0.288	89.0	11.71
19 Base Points + A	1.750	0.693	0.846	53.7	4.57
19 Base Points + B	1.775	0.640	0.285	94.7	18.01
19 Base Points + C	3.356	0.155	0.865	10.3	1.44

## 2.5.2 Exercise. Assessing outliers in lottery sales

### Assignment Text

In an earlier video, we made a scatter plot of population versus sales. This plot exhibits an outlier; the point in the upper left-hand side of the plot represents a zip code that includes Kenosha, Wisconsin. Sales for this zip code are unusually high given its population.

This exercise summarizes the regression fit both with and without this zip code in order to see how robust our results are to the inclusion of this unusual observation.

### Instructions

- A basic linear regression fit of population on sales has already been fit in the object `model_blr`. Re-fit this same model to the data, this time omitting Kenosha (observation number 9).
- Plot these two least squares fitted lines superimposed on the full data set.
- What is the effect on the distribution of residuals by removing this point? Calculate a normal qq plot with and without Kenosha.

### Hint

You can extract the residuals from a regression object with the function `[residuals()]`.

### Pre-exercise code

```
# Pre-exercise code
Lot <- read.csv("CSVData\\Wisc_lottery.csv", header = TRUE)
#Lot <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/a792b30fb32b0896dd6
```

### Sample\_code

```
`@sample_code`
model_blr <- lm(sales ~ pop, data = Lot)
summary(model_blr)
# Re-fit this model to the data, this time omitting Kenosha (observation number 9).
model_Kenosha <- lm(____ ~ ____, data = Lot, subset = -c(9))
summary(____)

# Plot these two least squares fitted lines superimposed on the full data set.
plot(____, ____, xlab = "population", ylab = "sales")
text(5000, 24000, "Kenosha")
abline(model_blr, col="blue")
abline(____, col="red")

# Calculate a normal qq plot with and without Kenosha.
par(mfrow = c(1, 2))
qqnorm(residuals(____), main = "")
qqline(residuals(____))
qqnorm(residuals(____), main = "")
qqline(residuals(____))
```

### Solution

```
# Solution
model_blr <- lm(sales ~ pop, data = Lot)
summary(model_blr)
model_Kenosha <- lm(sales ~ pop, data = Lot, subset = -c(9))
summary(model_Kenosha)

plot(Lot$pop, Lot$sales, xlab = "population", ylab = "sales")
text(5000, 24000, "Kenosha")
abline(model_blr, col="blue")
abline(model_Kenosha, col="red")

par(mfrow = c(1, 2))
qqnorm(residuals(model_blr), main = "")
qqline(residuals(model_blr))
qqnorm(residuals(model_Kenosha), main = "")
qqline(residuals(model_Kenosha))
```

### Submission Correctness Tests (SCT)

```
test_error() test_object("model_blr1", incorrect_msg = "The basic linear regression model is incorrectly specified.") test_object("model_Kenosha", incorrect_msg = "The linear regression model without Kenosha is incorrectly specified.") success_msg("Congratulations! Just because an observation is unusual does not make it bad or noninformative. Kenosha is close to the Illinois border; residents from Illinois probably participate in the Wisconsin lottery thus effectively increasing the potential pool of sales in Kenosha. Although unusual, there is interesting information to be learned from this observation.")
```





## Chapter 3

# Multiple Linear Regression (MLR)

### Chapter description

This chapter introduces linear regression in the case of several explanatory variables, known as multiple linear regression (MLR). Many basic linear regression concepts extend directly, including goodness of fit measures such as the coefficient of determination and inference using t-statistics. Multiple linear regression models provide a framework for summarizing highly complex, multivariate data. Because this framework requires only linearity in the parameters, we are able to fit models that are nonlinear functions of the explanatory variables, thus providing a wide scope of potential applications.

## 3.1 Method of least squares

### 3.1.1 Video (Exercise). Method of least squares

#### Learning Objectives

In this module, you learn how to:

- Interpret correlation coefficients by visualizing a scatterplot matrix
- Fit a plane to data using the method of least squares
- Predict an observation using a least squares fitted plane

#### Video Overheads

##### Overhead A. Demand for term life insurance

“Who buys insurance and how much do they buy?”

- Companies have data on current customers
- How do get info on potential (new) customers?

To understand demand, consider the Survey of Consumer Finances (SCF)

- This is a nationally representative sample that contains extensive information on potential U.S. customers.
- We study a random sample of 500 of the 4,519 households with positive income that were interviewed in the 2004 survey.
- We now focus on  $n = 275$  households that purchased term life insurance

**Overhead B. Term life insurance summary statistics**

We study  $y = \text{face}$ , the amount that the company will pay in the event of the death of the named insured.

We focus on  $k = 3$  explanatory variables - annual income, - the number of years of education of the survey respondent and - the number of household members, numhh.

The data suggest that income and face are skewed so we also introduce logarithmic versions.

**Overhead C. Summary statistics**

```
Term <- read.csv("CSVData\\term_life.csv", header = TRUE)
# PICK THE SUBSET OF THE DATA CORRESPONDING TO TERM PURCHASE
Term1 <- subset(Term, subset = face > 0)
str(Term1)
head(Term1)

library(Rcmdr)
Term2 <- Term1[, c("education", "face", "income", "logface", "logincome", "numhh")]
options(scipen = 100, digits = 4)
summvar <- numSummary(Term2, statistic = c("mean", "sd", "quantiles"), quantiles = c(0, .5, 1))
summvar
```

**Overhead C. Scatter plots of income versus face in original and logarithmic units**

```
par(mfrow = c(1, 2))
plot(Term2$income, Term2$face, xlab = "income", ylab = "face")
plot(Term2$logincome, Term2$logface, xlab = "log", ylab = "log face")
```

**Overhead D. Correlation table**

```
round(cor(Term2), digits=3)
```

**Overhead E. Scatterplot matrix**

```
Term3 <- Term1[,c("numhh", "education", "logincome", "logface")]
pairs(Term3, upper.panel = NULL, gap = 0, cex.labels = 1.25)
```

**Overhead F. Visualizing a regression plane**

```
education <- seq(3, 16, length = 15)
logincome <- seq(5, 15, length = 15)
f <- function(education, logincome){
  r <- 5 + 0.221*education + 0.354*logincome
}
logface <- outer(education, logincome, f)
persp(education, logincome, logface, theta = 30,
      phi = 30, expand = 0.5, ticktype = "detailed")
rm(education, logincome, logface)

education <- seq(3, 16, length = 15)
logincome <- seq(5, 15, length = 15)
f <- function(education, logincome){
  r <- 5 + 0.221*education + 0.354*logincome
}
logface <- outer(education, logincome, f)
persp(education, logincome, logface, theta = 30,
      phi = 30, expand = 0.5, ticktype = "simple", #ticktype = "detailed", #
      xlab = "x1", ylab="x2", zlab="y", nticks = 1)
rm(education, logincome, logface)
```

**Overhead G. Method of least squares**

- For observation  $\{(y, x_1, \dots, x_k)\}$ , the height of the regression plane is

$$b_0 + b_1x_1 + \dots + b_kx_k.$$

- Thus,  $y - (b_0 + b_1x_1 + \dots + b_kx_k)$  represents the deviation.
- The sum of squared deviations is

$$SS(b_0, \dots, b_k) = \sum (y - (b_0 + b_1x_1 + \dots + b_kx_k))^2.$$

- The method of least squares – determine values of  $b_0, \dots, b_k$  that minimize  $SS$ .

**Overhead H. Fit a multiple linear regression model**

```
Term_mlr <- lm(logface ~ education + numhh + logincome, data = Term2)
round(coefficients(Term_mlr), digits=4)
newdata <- data.frame(logincome = log(60000), education = 12, numhh = 3)
exp(predict(Term_mlr, newdata))
```

**3.1.2 Exercise. Least squares and term life data****Assignment Text**

The prior video introduced the Survey of Consumer Finances (SCF) term life data. A subset consisting of only those who purchased term life insurance, has already been read into a dataframe `Term2`.

Suppose that you wish to predict the amount of term life insurance that someone will purchase but are uneasy about the `education` variable. The SCF `education` variable is the number of completed years of schooling and so 12 corresponds to completing high school in the US. Your sense is that, for purposes of purchasing life insurance, high school graduates and those that attend college should be treated the same. So, in this exercise, you will create a new variable, `education1`, that is equal to years of education for those with education less than or equal to 12 and is equal to 12 otherwise.

**Instructions**

- Use the `pmin()` function to create the `education1` variable as part of the `Term2` dataframe.
- Check your work by examining summary statistics for the revised `Term2` dataframe.
- Examine correlations for the revised dataframe.
- Using the method of least squares and the function `lm()`, fit a MLR model using `logface` as the dependent variables and using `education`, `numhh`, and `logincome` as explanatory variables.
- With this fitted model and the function `predict()`, predict the face amount of insurance that someone with income of 40,000, 11 years of education, and 4 people in the household would purchase.

**Hint**

Remember that your prediction is in log dollars so you need to exponentiate it to get the results in the original dollar units

**Pre-exercise code**

```
# Pre-exercise code
#library(Rcmdr)
Term <- read.csv("CSVDData\\term_life.csv", header = TRUE)
#Term <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/efc64bc2d78cf6b48a
Term1 <- subset(Term, subset = face > 0)
Term2 <- Term1[, c("education", "face", "income", "logface", "logincome", "numhh")]
```

**Sample\_code**

```
`@sample_code`
# Create the `education1` variable as part of the `Term2` dataframe.
Term2$education1 <- pmin(12, Term2$education)

# Check your work by examining summary statistics for the revised `Term2` dataframe.
summary(____)

# Examine correlations for the revised dataframe.
round(cor(____), digits=3)

# Fit a MLR model using `logface` as the dependent variables and using `education`, `numhh`, and `logincome` as explanatory variables.
Term_mlr2 <- lm(logface ~ ____ + numhh + logincome, data = Term2)

# Predict the face amount of insurance that someone with income of 40,000, 11 years of education, and 4 years of experience would have.
newdata <- data.frame(logincome = log(40000), education1 = 11, numhh = 4)
exp(predict(____, newdata))
```

**Solution**

```
# Solution
Term2$education1 <- pmin(12, Term2$education)
#Rcmdr::numSummary(Term2, statistic = c("mean", "sd", "quantiles"), quantiles = c(0, .5, 1))
summary(Term2)
round(cor(Term2), digits=3)
Term_mlr2 <- lm(logface ~ education1 + numhh + logincome, data = Term2)
newdata <- data.frame(logincome = log(40000), education1 = 11, numhh = 4)
exp(predict(Term_mlr2, newdata))
```

**Submission Correctness Tests (SCT)**

```
test_error() test_object("Term2", incorrect_msg = "Is the new education variable properly defined using the pmin() function?") test_object("Term_mlr2", incorrect_msg = "The MLR model is incorrectly specified.") test_object("newdata", incorrect_msg = "The new data object is incorrectly specified.") success_msg("Congratulations! You now have experience fitting a regression plane and using this plane for predictions. Prediction is one of the key tasks of 'predictive modeling.' Well done!")
```

**3.1.3 Exercise. Interpreting coefficients as proportional changes****Assignment Text**

In a previous exercise, you fit a MLR model using `logface` as the outcome variable and using `education`, `numhh`, and `logincome` as explanatory variables; the resulting fit is in the object `Term_mlr`. For this fit, the coefficient associated with `education` is 0.2064. We now wish to interpret this regression coefficient.

The typical interpretation of coefficients in a regression model is as a partial slope. That is, for the coefficient  $b_1$  associated with  $x_1$ , we interpret  $b_1$  to be amount that the expected outcome changes per unit change in  $x_1$ , holding the other explanatory variables fixed.

For the term life example, the units of the outcome are in logarithmic dollars. So, for small values of  $b_1$ , we can interpret this to be a proportional change in dollars.

**Instructions**

- Determine least square fitted values for several selected values of `education`, holding other explanatory variables fixed. For this part of the demonstration, we used their mean values.

- Determine the proportional changes. Note the relation between these values from a discrete change approximation to the regression coefficient for `education` equal to 0.2064.

### Hint

#### Pre-exercise code

```
# Pre-exercise code
Term <- read.csv("CSVData\\term_life.csv", header = TRUE)
#Term <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/efc64bc2d78cf6b48a
Term1 <- subset(Term, subset = face > 0)
Term2 <- Term1[, c("education", "face", "income", "logface", "logincome", "numhh")]
Term_mlr <- lm(logface ~ education + numhh + logincome, data = Term2)
summary(Term_mlr)$coefficients[,1]
```

#### Sample\_code

```
`@sample_code`
Term_mlr <- lm(logface ~ education + numhh + logincome, data = Term2)
summary(Term_mlr)$coefficients[,1]

# Determine least square fitted values for several selected values of `education`, holding other explan
educ_predict <- c(14,14.1,14.2,14.3)
newdata1 <- data.frame(logincome = mean(Term2$logincome), education = educ_predict, numhh = mean(Term2$
lsfits1 <- predict(Term_mlr, newdata1)
lsfits1

# Determine the proportional changes. Note the relation between these values from a discrete change app
lsfits1[2:4] - lsfits1[1:3]
pchange_fits1 <- exp(lsfits1[2:4] - lsfits1[1:3])
pchange_fits1
```

### Solution

```
# Solution
educ_predict <- c(14,14.1,14.2,14.3)
newdata1 <- data.frame(logincome = mean(Term2$logincome), education = educ_predict, numhh = mean(Term2$
lsfits1 <- predict(Term_mlr, newdata1)
lsfits1
lsfits1[2:4] - lsfits1[1:3]
pchange_fits1 <- exp(lsfits1[2:4] - lsfits1[1:3])
pchange_fits1
```

### Submission Correctness Tests (SCT)

```
test_error() test_object("educ_predict", incorrect_msg = "Check to see that values of the education pre-
dictor variable are properly coded.") test_object("newdata1", incorrect_msg = "The new data object is
incorrectly specified.") test_object("lsfits1", incorrect_msg = "The predicted fits at different values of edu-
cation are incorrectly specified.") test_object("pchange_fits1", incorrect_msg = "The proportional changes
at different values of education are incorrectly specified.") success_msg("Congratulations! From calculus,
small changes in logarithmic values can be interpreted as proportional changes. This is the reason for using
natural logarithms.")
```

### 3.1.4 Exercise. Interpreting coefficients as elasticities

#### Assignment Text

In a previous exercise, you fit a MLR model using `logface` as the outcome variable and using `education`, `numhh`, and `logincome` as explanatory variables; the resulting fit is in the object `Term_mlr`. From this fit, the coefficient associated with `logincome` is 0.4935. We now wish to interpret this regression coefficient.

The typical interpretation of coefficients in a regression model is as a partial slope. When both  $x_1$  and  $y$  are in logarithmic units, then we can interpret  $b_1$  to be ratio of two percentage changes, known as an elasticity in economics. Mathematically, we summarize this as

$$\frac{\partial \ln y}{\partial \ln x} = \left( \frac{\partial y}{y} \right) / \left( \frac{\partial x}{x} \right).$$

### Instructions

- For several selected values of `logincome`, determine the corresponding proportional changes.
- Determine least square fitted values for several selected values of `logincome`, holding other explanatory variables fixed.
- Determine the corresponding proportional changes for the fitted values.
- Calculate the ratio of proportional changes of fitted values to those for income. Note the relation between these values (from a discrete change approximation) to the regression coefficient for `logincome` equal to 0.4935.

### Hint

When you calculate the ratio of proportional changes of fitted values to those for income, note the relation between these values (from a discrete change approximation) to the regression coefficient for `logincome` equal to 0.4935.

### Sample\_code

```
`@sample_code`
Term_mlr <- lm(logface ~ education + numhh + logincome, data = Term2)
summary(Term_mlr)$coefficients[,1]
# For several selected values of `logincome`, determine the corresponding proportional changes.
logincome_pred <- c(11,11.1,11.2,11.3)
pchange_income <- 100*(exp(logincome_pred[2:4])/exp(logincome_pred[1:3]))-1)
pchange_income

# Determine least square fitted values for several selected values of `logincome`, holding other explanatory variables fixed.
newdata2 <- data.frame(logincome = logincome_pred, education = mean(Term2$education), numhh = mean(Term2$numhh))
lsfits2 <- predict(Term_mlr, newdata2)

# Determine the corresponding proportional changes for the fitted values.
pchange_fits2 <- 100*(exp(lsfits2[2:4])/exp(lsfits2[1:3]))-1)
pchange_fits2

# Calculate the ratio of proportional changes of fitted values to those for income.
pchange_fits2/pchange_income
```

### Pre-exercise code

```
# Pre-exercise code
Term <- read.csv("CSVData\\term_life.csv", header = TRUE)
#Term <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/efc64bc2d78cf6b48a...")
Term1 <- subset(Term, subset = face > 0)
Term2 <- Term1[, c("education", "face", "income", "logface", "logincome", "numhh")]
Term_mlr <- lm(logface ~ education + numhh + logincome, data = Term2)
summary(Term_mlr)$coefficients[,1]
```

**Solution***# Solution*

```

logincome_pred <- c(11,11.1,11.2,11.3)
pchange_income <- 100*(exp(logincome_pred[2:4])/exp(logincome_pred[1:3])-1)
pchange_income
newdata2 <- data.frame(logincome = logincome_pred, education = mean(Term2$education), numhh = mean(Term2$numhh))
lsfits2 <- predict(Term_mlr, newdata2)
pchange_fits2 <- 100*(exp(lsfits2[2:4])/exp(lsfits2[1:3])-1)
pchange_fits2
pchange_fits2/pchange_income

```

**Submission Correctness Tests (SCT)**

```

test_error() test_object("logincome_pred", incorrect_msg = "Check to see that values of the logarithmic income predictor variable are properly coded.") test_object("pchange_income", incorrect_msg = "Check to see that the proportional changes of logarithmic income predictor variable are properly coded.") test_object("newdata2", incorrect_msg = "The new data object is incorrectly specified.") test_object("lsfits2", incorrect_msg = "The predicted fits at different values of logarithmic income are incorrectly specified.") test_object("pchange_fits2", incorrect_msg = "The proportional changes at different values of logarithmic income are incorrectly specified.") success_msg("Congratulations! When both  $x_1$  and  $y$  are in logarithmic units, then we can interpret  $b_1$  to be ratio of two percentage changes, known as an elasticity in economics.")

```

## 3.2 Foundations of multiple linear regression

### 3.2.1 Video (Exercise). Foundations of multiple linear regression

**Learning Objectives**

In this module, you learn how to:

- Contrast the observable to the error representation of the model
- Describe the unbiasedness and determine the variance of least squares regression coefficients
- Motivate the least squares method using the Gauss-Markov and normality of estimators

**Video Overheads**

Overhead A. xxx

### 3.2.2 Exercise. Multiple choice exercise on the theory... Maybee

**Assignment Text****Instructions***# Pre-exercise code**# Solution*

### 3.3 Statistical inference and multiple linear regression

#### 3.3.1 Video (Exercise). Statistical inference and multiple linear regression

##### Learning Objectives

In this module, you learn how to:

- Explain mean square error and residual standard error in terms of degrees of freedom
- Develop an ANOVA table and use it to derive the coefficient of determination
- Calculate and interpret the coefficient of determination adjusted for degrees of freedom
- Conduct a test of a regression coefficient
- Summarize regression coefficients using point and interval estimators

##### Video Overheads

##### Overhead A. Goodness of fit

Summarize

- deviations
- $s^2$
- $R^2$
- $R_a^2$
- ANOVA table

##### Overhead B. Goodness of fit and term life

```
Term_mlr <- lm(logface ~ education + numhh + logincome, data = Term2)
summary(Term_mlr)
anova(Term_mlr)
```

##### Overhead C. Statistical inference

- hypothesis testing of a regression coefficient
- confidence intervals

##### Overhead D. Statistical inference and term life

```
Term_mlr <- lm(logface ~ education + numhh + logincome, data = Term2)
model_sum <- summary(Term_mlr)
model_sum$coefficients

round(Rcmdr::Confint(Term_mlr, level = .95), digits = 3)

round(confint(Term_mlr, level = .95), digits = 3)
```

#### 3.3.2 Exercise. Statistical inference and term life

##### Assignment Text

In later chapters, we will learn how to specify a model using diagnostics techniques; these techniques were used to specify face in log dollars for the outcome and similarly income in log dollars as an explanatory variable. Just to see how things work, in this exercise we will create new variables **face** and **income** that are in the original units and run a regression with these. We have already seen that rescaling by constants



do not affect relationships but can be helpful with interpretations, so we define both `face` and `income` to be in thousands of dollars. A prior video introduced the term life dataframe `Term2`.

### Instructions

- Create `Term2$face` by exponentiating `logface` and dividing by 1000. For convenience, we are storing this variable in the data set `Term2`. Use the same process to create `Term2$income`.
- Run a regression using `face` as the outcome variable and `education`, `numhh`, and `income` as explanatory variables.
- Summarize this model and identify the residual standard error ( $s$ ) as well as the coefficient of determination ( $R^2$ ) and the version adjusted for degrees of freedom ( $R_a^2$ ).

### Hint

#### Pre-exercise code

```
# Pre-exercise code
Term <- read.csv("CSVData\\term_life.csv", header = TRUE)
#Term <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/efc64bc2d78cf6b48a...")
Term1 <- subset(Term, subset = face > 0)
Term2 <- Term1[, c("education", "face", "income", "logface", "logincome", "numhh")]
```

#### Sample\_code

```
`@sample_code`
# Create `Term2$face` and `Term2$income`
Term2$face <- exp(____)/____
Term2$income <- exp(____)/____

# Run a regression using `face` as the outcome variable and `education`, `numhh`, and `income` as explanatory variables
Term_mlr1 <- lm(face ~ ____, data = Term2)

# Summarize this model
summary(Term_mlr1)
```

### Solution

```
# Solution
Term2$face <- exp(Term2$logface)/1000
Term2$income <- exp(Term2$logincome)/1000
Term_mlr1 <- lm(face ~ education + numhh + income, data = Term2)
summary(Term_mlr1)
```

### Submission Correctness Tests (SCT)

success\_msg("Congratulations! Compare these goodness of fit measures to those where income and face are in logarithmic units. Although not the only indicators, you will see that the proportion of variability explained (R square) and the statistical significance of coefficients are strikingly higher in the model with variables in logged units.")

## 3.4 Binary variables

### 3.4.1 Video (Exercise). Binary variables

#### Learning Objectives

In this module, you learn how to:

- Interpret regression coefficients associated with binary variables
- Use binary variables and interaction terms to create regression models that are nonlinear in the covariates

## Video Overheads

### Overhead A. Binary variables

- We can define a new variable

$$single = \begin{cases} 0 & \text{for other respondents} \\ 1 & \text{for single respondents} \end{cases}$$

- The variable single is said to be an indicator, or dummy, variable.
- To interpret coefficients, we now consider the regression function

$$E \log face = \beta_0 + \beta_1 \log income + \beta_2 single$$

- This can be expressed as two lines

$$E \log face = \begin{cases} \beta_0 + \beta_1 \log income & \text{for other respondents} \\ \beta_0 + \beta_2 + \beta_1 \log income & \text{for single respondents} \end{cases}.$$

- The least squares method of calculating the estimators, and the resulting theoretical properties, are the still valid when using binary variables.

### Overhead B. Visualize effect of binary variables

### Overhead C. R script for visualization

```
Term4 <- Term1[,c("numhh", "education", "logincome", "logface", "marstat")]
Term4$marstat <- as.factor(Term4$marstat)
table(Term4$marstat)
Term4$single <- 1*(Term4$marstat == 0)
model_single <- lm(logface ~ logincome + single, data = Term4)
summary(model_single)

plot(Term4$logincome, Term4$logface, xlab="logarithmic income", ylab="log face",
     pch= 1+16*Term4$single, col = c("red", "black", "black")[Term4$marstat])
Ey1 <- model_single$coefficients[1]+model_single$coefficients[2]*Term4$logincome
Ey2 <- Ey1 + model_single$coefficients[3]
lines(Term4$logincome, Ey1)
lines(Term4$logincome, Ey2, col="red")
```

### Overhead D. Interaction Terms

- Linear regression models are defined in terms of linear combinations of explanatory variables but we can expand their scope through nonlinear transformations
- One type of nonlinear transform is the product of two variables that is used to create what is known as an interaction variable
- To interpret coefficients, we now consider the regression function

$$E \log face = \beta_0 + \beta_1 \log income + \beta_2 single + \beta_3 single * \log income$$

- This can be expressed as two lines with different slopes

$$E \log face = \begin{cases} \beta_0 + \beta_1 \log income & \text{for other respondents} \\ \beta_0 + \beta_2 + (\beta_1 + \beta_3) \log income & \text{for single respondents} \end{cases}.$$

## Overhead E. Visualizing binary variables with interactions terms

```
Term4 <- Term1[,c("numhh", "education", "logincome", "logface", "marstat")]
Term4$marstat <- as.factor(Term4$marstat)
table(Term4$marstat)
Term4$single <- 1*(Term4$marstat == 0)
model_single_inter <- lm(logface ~ logincome + single + single*logincome, data = Term4)
summary(model_single_inter)

plot(Term4$logincome, Term4$logface, xlab="logarithmic income", ylab="log face",
     pch= 1+16*Term4$single, col = c("red", "black", "black")[Term4$marstat])
Ey1 <- model_single_inter$coefficients[1]+model_single_inter$coefficients[2]*Term4$logincome
Ey2 <- Ey1 + model_single_inter$coefficients[3]+model_single_inter$coefficients[4]*Term4$logincome
lines(Term4$logincome, Ey1)
lines(Term4$logincome, Ey2, col="red")
```

## 3.4.2 Exercise. Binary variables and term life

## Assignment Text

In the prior video, we saw how the variable `single` can be used with logarithmic income to explain logarithmic face amounts of term life insurance that people purchase. The coefficient associated with this variable turns out to be negative which is intuitively appealing; if an individual is single, then that person may not have the strong need to purchase financial security for others in the event of unexpected death.

In this exercise, we will extend this by incorporating `single` into our larger regression model that contains other explanatory variables, `logincome`, `education` and `numhh`. The data have been pre-loaded into the dataframe `Term4`.

## Instructions

- Calculate a table of correlation coefficients to examine pairwise linear relationships among the variables `numhh`, `education`, `logincome`, `single`, and `logface`.
- Fit a MLR model of `logface` using explanatory variables `numhh`, `education`, `logincome`, and `single`. Examine the residual standard deviation  $s$ , the coefficient of determination  $R^2$ , and the adjusted version  $R_a^2$ . Also note the statistical significance of the coefficient associated with `single`.
- Repeat the MLR model fit while adding the interaction term `single*logincome`.

## Hint

## Pre-exercise code

```
# Pre-exercise code
Term <- read.csv("CSVData\\term_life.csv", header = TRUE)
#Term <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/efc64bc2d78cf6b48a")
Term1 <- subset(Term, subset = face > 0)
Term4 <- Term1[,c("numhh", "education", "logincome", "marstat", "logface")]
Term4$single <- 1*(Term4$marstat == 0)
```

## Sample\_code

```
`@sample_code`
# Calculate a table of correlation coefficients
round(___(Term4[,c("numhh", "education", "logincome", "single", "logface")]), digits = 3)

# Fit a MLR model of `logface` using explanatory variables `numhh`, `education`, `logincome`, and `sing
Term_mlr3 <- lm(logface ~ education + numhh + logincome + single, data = Term4)
summary(Term_mlr3)
```

```
# Repeat the MLR model fit while adding the interaction term `single*logincome`.
Term_mlr4 <- lm(logface ~ education + numhh + logincome + single + single*logincome, data = Term4)
summary(Term_mlr4)
```

### Solution

```
# Solution
round(cor(Term4[,c("numhh", "education", "logincome", "single", "logface")]), digits = 3)
Term_mlr3 <- lm(logface ~ education + numhh + logincome + single, data = Term4)
summary(Term_mlr3)
Term_mlr4 <- lm(logface ~ education + numhh + logincome + single + single*logincome, data = Term4)
summary(Term_mlr4)
```

### Submission Correctness Tests (SCT)

success\_msg("Congratulations! From a correlation table, you saw that there are relationships with among explanatory variables and so it is not clear whether adding **single** to the model would be helpful. You explored this by first fitting a model by just adding the binary variable **single**, examined summary statistics, and checked the significance of the variable. Then, you explored the utility of the interaction of **single** with logarithmic income. Well done!")

## 3.5 Categorical variables

### 3.5.1 Video (Exercise). Categorical variables

#### Learning Objectives

In this module, you learn how to:

- Represent categorical variables using a set of binary variables
- Interpret the regression coefficients associated with categorical variables
- Describe the effect of the reference level choice on the model fit

#### Video Overheads

##### Overhead A. Categorical variables

- Categorical variables provide labels for observations to denote membership in distinct groups, or categories.
- A binary variable is a special case of a categorical variable.
  - To illustrate, a binary variable may tell us whether or not someone has health insurance.
  - A categorical variable could tell us whether someone has (i) private individual health insurance, (ii) private group insurance, (iii) public insurance or (iv) no health insurance.
- For categorical variables, there may or may not be an ordering of the groups.
  - For health insurance, it is difficult to say which is ‘larger’, private individual versus public health insurance (such as Medicare).
  - However, for education, we may group individuals from a dataset into ‘low’, ‘intermediate’ and ‘high’ years of education.
- Factor is another term used for a (unordered) categorical explanatory variable.

##### Overhead B. Term life example

- We studied  $y = \text{logface}$ , the amount that the company will pay in the event of the death of the named insured (in logarithmic dollars), focusing on the explanatory variables  $\text{logincome}$ ,  $\text{education}$ , and  $\text{numhh}$ .
- We now supplement this by including the categorical variable,  $\text{marstat}$ , that is the marital status of the survey respondent. This may be:
  - 1, for married
  - 2, for living with partner
  - 0, for other (SCF actually breaks this category into separated, divorced, widowed, never married and inapplicable, for persons age 17 or less or no further persons)

#### Overhead C. Term life boxplots

```
# Pre-exercise code
Term <- read.csv("CSVData\\term_life.csv", header = TRUE)
#Term <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/efc64bc2d78cf6b48a
Term1 <- subset(Term, subset = face > 0)
Term4 <- Term1[,c("numhh", "education", "logincome", "marstat", "logface")]
Term4$single <- 1*(Term4$marstat == 0)
Term4$marstat <- as.factor(Term4$marstat)
boxplot(logface ~ marstat, ylab = "log face", xlab = "Marital Status", data = Term4)
table(Term4$marstat)
# SUMMARY BY LEVEL OF MARSTAT
library(Rcmdr)
numSummary(Term4[, "logface"], groups = Term4$marstat, statistics = c("mean", "sd"))
numSummary(Term4[, "logface"], statistics = c("mean", "sd"))
```

#### Overhead D. Regression with a categorical variable

```
Term4$marstat <- as.factor(Term4$marstat)
Term4$marstat <- relevel(Term4$marstat, ref = "2")
summary(lm(logface ~ logincome+education+numhh+marstat, data = Term4))
```

#### Overhead E. t-ratios depend on the reference level

	Model 1		Model 2		Model 3	
Var	Coef	t-stat	Coef	t-stat	Coef	t-stat
<i>logincome</i>	0.452	5.74	0.452	5.74	0.452	5.74
<i>education</i>	0.205	5.30	0.205	5.30	0.205	5.30
<i>numhh</i>	0.248	3.57	0.248	3.57	0.248	3.57
Intercept	3.395	3.77	2.605	2.74	2.838	3.34
mar=0	-0.557	-2.15	0.232	0.44		
mar=1			0.789	1.59	0.557	2.15
mar=2	-0.789	-1.59			-0.232	-0.44

### 3.5.2 Exercise. Categorical variables and Wisconsin hospital costs

#### Assignment Text

This exercise examines the impact of various predictors on hospital charges. Identifying predictors of hospital charges can provide direction for hospitals, government, insurers and consumers in controlling these variables that in turn leads to better control of hospital costs. The data, from 1989, are aggregated by:

- **drg**, diagnostic related groups of costs,
- **payer**, type of health care provider (Fee for service, HMO, and other), and
- **hsa**, nine major geographic areas in Wisconsin.

Some preliminary analysis of the data has already been done. In this exercise, we will analyze `logcharge`, the logarithm of total hospital charges per number of discharges, in terms of `log_numdschg`, the logarithm of the number of discharges. In the dataframe `Hcost` which has been loaded in advance, we restrict consideration to three types of drgs, numbers 209, 391, and 431.

### Instructions

- Fit a basic linear regression model using logarithmic number of discharges to predict logarithmic hospital costs and superimposed the fitted regression line on the scatter plot.
- Produce a scatter plot of logarithmic number of discharges to predict logarithmic hospital costs. Allow plotting symbols and colors to vary by diagnostic related group.
- Fit a MLR model using logarithmic number of discharges to predict logarithmic hospital costs, allowing intercepts and slopes to vary by diagnostic related groups.
- Superimpose the fits from the MLR model on the scatter plot of logarithmic number of discharges to predict logarithmic hospital costs.

### Hint

#### Pre-exercise code

```
# Pre-exercise code
Hcost <- read.csv("CSVDData\\WiscHcosts.csv", header = TRUE)
#Hcost <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/2cc1e2739bf827093
Hcost1 <- subset(Hcost, drg == 209|drg == 391|drg == 430)
```

#### Sample\_code

```
`@sample_code`
# Fit a basic linear regression model using logarithmic number of discharges to predict logarithmic hos
hosp_blr <- lm(logcharge~log_numdschg, data=Hcost1)
plot(logcharge~log_numdschg, data=Hcost1, xlab = "log number discharges", ylab = "log charge")
abline(hosp_blr, col="red")

# Produce a scatter plot of logarithmic number of discharges to predict logarithmic hospital costs. All
plot(logcharge~log_numdschg, data=Hcost1, xlab = "log number discharges", ylab = "log charge",
     pch= as.numeric(as.factor(Hcost1$drg)),
     col = c("red", "black", "blue")[as.factor(Hcost1$drg)])
legend("left", legend=c("drg 209", "drg 391", "drg 430"), col=c("red", "black", "blue"), pch = c(1,2,3))

# Fit a MLR model allowing intercepts and slopes to vary by drg.
hosp_mlr <- lm(logcharge~log_numdschg + as.factor(drg)*log_numdschg, data=Hcost1)

# Superimpose the fits from the MLR model on the scatter plot of logarithmic number of discharges to pr
plot(logcharge~log_numdschg, data=Hcost1, xlab = "log number discharges", ylab = "log charge",
     pch= as.numeric(as.factor(Hcost1$drg)),
     col = c("red", "black", "blue")[as.factor(Hcost1$drg)])
xseq <- seq(0,10,length.out=100)
coef <- summary(hosp_mlr)$coefficients[,1]
fit209 <- coef[1] + coef[2]*xseq
lines(xseq,fit209, col="red")
fit391 <- coef[1] + coef[3] + (coef[2] + coef[5])*xseq
lines(xseq,fit391, col="black")
fit430 <- coef[1] + coef[4] + (coef[2] + coef[6])*xseq
lines(xseq,fit430, col="blue")
```

### Solution

```
# Solution
#par(mfrow = c(2, 1))
hosp_blr <- lm(logcharge~log_numdschg, data=Hcost1)
plot(logcharge~log_numdschg, data=Hcost1, xlab = "log number discharges", ylab = "log charge")
abline(hosp_blr, col="red")

plot(logcharge~log_numdschg, data=Hcost1, xlab = "log number discharges", ylab = "log charge",
     pch= as.numeric(as.factor(Hcost1$drg)),
     col = c("red", "black", "blue")[as.factor(Hcost1$drg)])
legend("left", legend=c("drg 209", "drg 391", "drg 430"), col=c("red", "black", "blue"), pch = c(1,2,3))

hosp_mlr <- lm(logcharge~log_numdschg + as.factor(drg)*log_numdschg, data=Hcost1)
#summary(hosp_mlr)$coefficients[,1]
plot(logcharge~log_numdschg, data=Hcost1, xlab = "log number discharges", ylab = "log charge",
     pch= as.numeric(as.factor(Hcost1$drg)),
     col = c("red", "black", "blue")[as.factor(Hcost1$drg)])
xseq <- seq(0,10,length.out=100)
coef <- summary(hosp_mlr)$coefficients[,1]
fit209 <- coef[1] + coef[2]*xseq
lines(xseq,fit209, col="red")
fit391 <- coef[1] + coef[3] + (coef[2] + coef[5])*xseq
lines(xseq,fit391, col="black")
fit430 <- coef[1] + coef[4] + (coef[2] + coef[6])*xseq
lines(xseq,fit430, col="blue")
```

### Submission Correctness Tests (SCT)

success\_msg("Congratulations! When you superimposed the fits from the MLR model on the scatter plot of logarithmic number of discharges to predict logarithmic hospital costs, note how slopes differ dramatically from the slope from the basic linear regression model.")

## 3.6 General linear hypothesis

### 3.6.1 Video (Exercise). Hypothesis testing

#### Learning Objectives

In this module, you learn how to:

- Jointly test the significance of a set of regression coefficients using the general linear hypothesis
- Conduct a test of a regression coefficient versus one- or two-side alternatives

#### Video Overheads

##### Overhead A. Testing the significance of a categorical variable

```
Term <- read.csv("CSVData\\term_life.csv", header = TRUE)
#Term <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/efc64bc2d78cf6b48a
Term1 <- subset(Term, subset = face > 0)
Term4 <- Term1[,c("numhh", "education", "logincome", "marstat", "logface")]
Term_mlr1 <- lm(logface ~ logincome + education + numhh + as.factor(Term4$marstat), data = Term4)
anova(Term_mlr1)
Term_mlr2 <- lm(logface ~ logincome + education + numhh, data = Term4)
```



```
Fstat <- (anova(Term_mlr2)$`Sum Sq`[4] - anova(Term_mlr1)$`Sum Sq`[5])/(2*anova(Term_mlr1)$`Mean Sq`[5])
Fstat
cat("p-value is", 1 - pf(Fstat, df1 = 2 , df2 = anova(Term_mlr1)$Df[5]))
```

### Overhead B. Overview of the general linear hypothesis

- The likelihood ratio is a general statistical test procedure that compares a model to a subset
- The general linear hypothesis test procedure is similar.
  - Start with a (large) linear regression model, examine the fit to a set of data
  - Compare this to smaller model that is a subset of the large model.
  - “Subset” is the sense that regression coefficients from the small model are linear combinations of regression coefficients of the large model (e.g., set them to zero)
- Although the likelihood ratio test is more generally available, the general linear hypothesis test is more accurate for smaller data sets (for normally distributed data)

### Overhead C. Procedure for conducting the general linear hypothesis

- Run the full regression and get the error sum of squares and mean square error, which we label as  $(ErrorSS)_{full}$  and  $s_{full}^2$ , respectively.
- Run a reduced regression and get the error sum of squares, labelled  $(ErrorSS)_{reduced}$ .
- Using  $p$  for the number of linear restrictions, calculate

$$F - ratio = \frac{(ErrorSS)_{reduced} - (ErrorSS)_{full}}{ps_{full}^2}.$$

- The probability value is  $p - value = \Pr(F_{p,df} > F - ratio)$  where  $F_{p,df}$  has an F distribution with degrees of freedom  $p$  and  $df$ , respectively. (Here,  $df$  is the degrees of freedom for the full model.)

### Overhead D. The general linear hypothesis for a single variable

- Suppose that you wish to test the hypothesis that a regression coefficient equals 0.
  - One could use the general linear hypothesis procedure with  $p = 1$ .
  - One could also examine the corresponding  $t - ratio$ .
  - Which is correct?
- Both. One can show that  $(t - ratio)^2 = F - ratio$ , so they are equivalent statistics.
  - The general linear hypothesis is useful because it can be extended to multiple coefficients.
  - The t-ratio is useful because it can be used to examine one-sided alternative hypotheses.

## 3.6.2 Exercise. Hypothesis testing and term life

### Assignment Text

With our `Term life` data, let us compare a model based on the binary variable that indicates whether a survey respondent is single versus the more complex marital status, `marstat`. In principle, more detailed information is better. But, it may be that the additional information in `marstat`, compared to `single`, does not help fit the data in a significantly better way.

As part of the preparatory work, the dataframe `Term4` is available that includes the binary variable `single` and the factor `marstat`. Moreover, the regression object `Term_mlr` contains information in a multiple linear regression fit of `logface` on the base explanatory variables 'logincome, education, and numhh'.

### Instructions

- Fit a MLR model using the base explanatory variables plus `single` and another model using the base variables plus `marstat`.
- Use the F test to decide whether the additional complexity `marstat` is warranted by calculating the p-value associated with this test.



- Fit a MLR model using the base explanatory variables plus `single` interacted with `logincome` and another model using the base variables plus `marstat` interacted with `logincome`.
- Use the F test to decide whether the additional complexity `marstat` is warranted by calculating the p-value associated with this test.

### Hint

Here is the code to calculate it by hand

```
Fstat12 <- (anova(Term_mlr1)$`Sum Sq`[5] -
           anova(Term_mlr2)$`Sum Sq`[5])/(1*anova(Term_mlr2)$`Mean Sq`[5])
Fstat12
cat("p-value is", 1 - pf(Fstat12, df1 = 1 , df2 = anova(Term_mlr2)$Df[5]))
```

### Pre-exercise code

```
# Pre-exercise code
Term <- read.csv("CSVData\\term_life.csv", header = TRUE)
#Term <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/efc64bc2d78cf6b48a
Term1 <- subset(Term, subset = face > 0)
Term4 <- Term1[,c("numhh", "education", "logincome", "marstat", "logface")]
Term4$single <- 1*(Term4$marstat == 0)
Term4$marstat <- as.factor(Term4$marstat)
Term_mlr <- lm(logface ~ logincome + education + numhh , data = Term4)
anova(Term_mlr)
```

### Sample\_code

```
`@sample_code`
# Fit a MLR model using the base explanatory variables plus `single` and another model using the base v
Term_mlr1 <- lm(logface ~ logincome + education + numhh +single, data = Term4)
Term_mlr2 <- lm(logface ~ logincome + education + numhh +marstat, data = Term4)

# Use the F test to decide whether the additional complexity `marstat` is warranted by calculating the
anova(Term_mlr1,Term_mlr2)

# Fit a MLR model using the base explanatory variables plus `single` interacted with `logincome` and an
Term_mlr3 <- lm(logface ~ logincome + education + numhh + single*logincome, data = Term4)
Term_mlr4 <- lm(logface ~ logincome + education + numhh +marstat*logincome, data = Term4)

# Use the F test to decide whether the additional complexity `marstat` is warranted by calculating the
anova(Term_mlr3,Term_mlr4)
```

### Solution

```
# Solution
Term_mlr1 <- lm(logface ~ logincome + education + numhh +single, data = Term4)
Term_mlr2 <- lm(logface ~ logincome + education + numhh +marstat, data = Term4)
anova(Term_mlr1,Term_mlr2)
#Fstat12 <- (anova(Term_mlr1)$`Sum Sq`[5] -
#           anova(Term_mlr2)$`Sum Sq`[5])/(1*anova(Term_mlr2)$`Mean Sq`[5])
#Fstat12
#cat("p-value is", 1 - pf(Fstat12, df1 = 1 , df2 = anova(Term_mlr2)$Df[5]))
Term_mlr3 <- lm(logface ~ logincome + education + numhh + single*logincome, data = Term4)
Term_mlr4 <- lm(logface ~ logincome + education + numhh +marstat*logincome, data = Term4)
anova(Term_mlr3,Term_mlr4)
```

### Submission Correctness Tests (SCT)

success\_msg("Congratulations! Hypothesis testing is a primary tool for ‘inferring’ about the real world {in contrast to mathematical ‘deduction’}. Moreover, as we will see in the next chapter, it can also be used to develop a model.”)

### 3.6.3 Exercise. Hypothesis testing and Wisconsin hospital costs

#### Assignment Text

In a previous exercise, you were introduced to a dataset with hospital charges aggregated by:

- **drg**, diagnostic related groups of costs,
- **payer**, type of health care provider (Fee for service, HMO, and other), and
- **hsa**, nine major geographic areas.

We continue our analysis of the outcome variable **logcharge**, the logarithm of total hospital charges per number of discharges, in terms of **log\_numdschg**, the logarithm of the number of discharges, as well as the three categorical variables used in the aggregation. As before, we restrict consideration to three types of drgs, numbers 209, 391, and 431 that has been preloaded in the dataframe **Hcost1**.

#### Instructions

- Fit a basic linear regression model using logarithmic hospital costs as the outcome variable and explanatory variable logarithmic number of discharges.
- Fit a MLR model using logarithmic hospital costs as the outcome variable and explanatory variables logarithmic number of discharges and the categorical variable diagnostic related group. Identify the F statistic and p value that test the importance of diagnostic related group.
- Fit a MLR model using logarithmic hospital costs as the outcome variable and explanatory variable logarithmic number of discharges interacted with diagnostic related group. Identify the F statistic and p value that test the importance of diagnostic related group interaction with logarithmic number of discharges.
- Calculate a coefficient of determination,  $R^2$ , for each of these models as well as for a model using logarithmic number of discharges and categorical variable **hsa** as predictors.

#### Hint

#### Pre-exercise code

```
# Pre-exercise code
Hcost <- read.csv("CSVData\\WiscHcosts.csv", header = TRUE)
#Hcost <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/2cc1e2739bf827093")
Hcost1 <- subset(Hcost, drg == 209|drg == 391|drg == 430)
```

#### Sample\_code

```
`@sample_code`
# Regress log charges on log number of discharges
hosp_blr <- lm(logcharge ~ log_numdschg , data=Hcost1)
anova(hosp_blr)

# Regress log charges on log number of discharges and drg. Identify the *F* statistic and *p* value tha
hosp_mlr1 <- lm(logcharge ~ log_numdschg + as.factor(drg), data=Hcost1)
anova(hosp_mlr1)

# Regress log charges on the interaction of log number of discharges and drg.
hosp_mlr2 <- lm(logcharge ~ log_numdschg + as.factor(drg)*log_numdschg, data=Hcost1)
anova(hosp_mlr2)

# Calculate a coefficient of determination,  $R^2$ , for each of these models as well as for a model usin.
```

```
summary(hosp_blr)$r.squared
summary(hosp_mlr1)$r.squared
summary(hosp_mlr2)$r.squared

hosp_mlr3 <- lm(logcharge ~ log_numdschg + as.factor(hsa)*log_numdschg, data=Hcost1)
summary(hosp_mlr3)$r.squared
```

### Solution

```
# Solution

hosp_blr <- lm(logcharge ~ log_numdschg , data=Hcost1)
anova(hosp_blr)
hosp_mlr1 <- lm(logcharge ~ log_numdschg + as.factor(drg), data=Hcost1)
anova(hosp_mlr1)
hosp_mlr2 <- lm(logcharge ~ log_numdschg + as.factor(drg)*log_numdschg, data=Hcost1)
anova(hosp_mlr2)
summary(hosp_blr)$r.squared
summary(hosp_mlr1)$r.squared
summary(hosp_mlr2)$r.squared

hosp_mlr3 <- lm(logcharge ~ log_numdschg + as.factor(hsa)*log_numdschg, data=Hcost1)
summary(hosp_mlr3)$r.squared
```

### Submission Correctness Tests (SCT)

success\_msg(“Congratulations! By examining the coefficients of determination,  $R^2$ , for each of these models, you see that this provides one piece of evidence that the `hsa` is a far poorer predictor of costs than `drg`.”)

## 3.6.4 Exercise. Hypothesis testing and auto claims

### Assignment Text

As an actuarial analyst, you are working with a large insurance company to help them understand their claims distribution for their private passenger automobile policies. You have available claims data for a recent year, consisting of:

- **state**: codes 01 through 17 used, with each code randomly assigned to an actual individual state
- **class**: rating class of operator, based on age, gender, marital status, and use of vehicle
- **gender**: operator gender
- **age**: operator age
- **paid**: amount paid to settle and close a claim.

You are focusing on older drivers, 50 and higher, for which there are  $n = 6,773$  claims available.

### Instructions

- Run a regression of `logpaid` on `age`. Is `age` a statistically significant variable? To respond to this question, use a formal test of hypothesis. State your null and alternative hypotheses, decision-making criterion, and your decision-making rule. Also comment on the goodness of fit of this variable.
- Consider using `class` as a single explanatory variable. Use the one factor to estimate the model and respond to the following questions.
  - What is the point estimate of claims in class C7, drivers 50-69, driving to work or school, less than 30 miles per week with annual mileage under 7500, in natural logarithmic units?

- b (ii). Determine the corresponding 95% confidence interval of expected claims, in natural logarithmic units.
- b (iii). Convert the 95% confidence interval of expected claims that you determined in part b(ii) to dollars.
- c. Run a regression of `logpaid` on `age`, `gender` and the categorical variables `state` and `class`.
  - c (i). Is `gender` a statistically significant variable? To respond to this question, use a formal test of hypothesis. State your null and alternative hypotheses, decision-making criterion, and your decision-making rule.
  - c (ii). Is `class` a statistically significant variable? To respond to this question, use a formal test of hypothesis. State your null and alternative hypotheses, decision-making criterion, and your decision-making rule.
  - c (iii). Use the model to provide a point estimate of claims in dollars (not log dollars) for a male age 60 in STATE 2 in `class` C7.
  - c (iv). Write down the coefficient associated with `class` C7 and interpret this coefficient.

### Submission Correctness Tests (SCT)

```
success_msg("Congratulations!")
```

# Chapter 4

## Variable Selection

### Chapter description

This chapter describes tools and techniques to help you select variables to enter into a linear regression model, beginning with an iterative model selection process. In applications with many potential explanatory variables, automatic variable selection procedures are available that will help you quickly evaluate many models. Nonetheless, automatic procedures have serious limitations including the inability to account properly for nonlinearities such as the impact of unusual points; this chapter expands upon the Chapter 2 discussion of unusual points. It also describes collinearity, a common feature of regression data where explanatory variables are linearly related to one another. Other topics that impact variable selection, including out-of-sample validation, are also introduced.

```
# Reformat Data
RSurvey <- read.csv("CSVData\\RiskSurvey.csv", header = TRUE)
str(RSurvey)
RSurvey$firmcost <- RSurvey$FIRMCOST
RSurvey$assume <- RSurvey$ASSUME
RSurvey$cap <- RSurvey$CAP
RSurvey$logsize <- RSurvey$SIZELOG
RSurvey$indcost <- RSurvey$INDCOST
RSurvey$central <- RSurvey$CENTRAL
RSurvey$soph <- RSurvey$SOPH
RSurvey2 <- RSurvey[,c("firmcost", "assume", "cap", "logsize", "indcost", "central", "soph")]
str(RSurvey2)
#write.csv(RSurvey2, "CSVData\\Risk_survey.csv", row.names = FALSE)
```

## 4.1 An iterative approach to data analysis and modeling

### 4.1.1 Video (Exercise). An iterative approach to data analysis and modeling

#### Learning Objectives

In this module, you learn how to:

- Describe the iterative approach to data analysis and modeling.

## Video Overheads

### Overhead A. Iterative approach

- Model formulation stage
- Fitting
- Diagnostic checking - the data and model must be consistent with one another before additional inferences can be made.

```
plot.new()
par(mar=c(0,0,0,0), cex=0.9)
plot.window(xlim=c(0,18),ylim=c(-5,5))

text(1,3,labels="DATA",adj=0, cex=0.8)
text(1,0,labels="PLOTS",adj=0, cex=0.8)
text(1,-3,labels="THEORY",adj=0, cex=0.8)
text(3.9,0,labels="MODEL\nFORMULATION",adj=0, cex=0.8)
text(8.1,0,labels="FITTING",adj=0, cex=0.8)
text(11,0,labels="DIAGNOSTIC\nCHECKING",adj=0, cex=0.8)
text(15,0,labels="INFERENCE",adj=0, cex=0.8)
text(14.1,0.5,labels="OK",adj=0, cex=0.6)

rect(0.8,2.0,2.6,4.0)
arrows(1.7,2.0,1.7,1.0,code=2,lwd=2,angle=25,length=0.10)
rect(0.8,-1.0,2.6,1.0)
arrows(1.7,-2.0,1.7,-1.0,code=2,lwd=2,angle=25,length=0.10)
rect(0.8,-4.0,2.6,-2.0)

arrows(2.6,0,3.2,0,code=2,lwd=2,angle=25,length=0.10)

x<-c(5,7.0,5,3.2)
y<-c(2,0,-2,0)
polygon(x,y)
arrows(7.0,0,8.0,0,code=2,lwd=2,angle=25,length=0.10)

rect(8.0,-1.0,9.7,1.0)
arrows(9.7,0,10.2,0,code=2,lwd=2,angle=25,length=0.10)

x1<-c(12,14.0,12,10.2)
y1<-c(2,0,-2,0)
polygon(x1,y1)
arrows(14.0,0,14.8,0,code=2,lwd=2,angle=25,length=0.10)

rect(14.8,-1.0,17.5,1.0)
arrows(12,-2.0,12,-3,code=2,lwd=2,angle=25,length=0.10)
arrows(12,-3.0,5,-3,code=2,lwd=2,angle=25,length=0.10)
arrows(5,-3.0,5,-2,code=2,lwd=2,angle=25,length=0.10)
```

### Overhead B. Many possible models

$E y = \beta_0$	1 model with no variables
$E y = \beta_0 + \beta_1 x_i,$	4 models with one variable
$E y = \beta_0 + \beta_1 x_i + \beta_2 x_j,$	6 models with two variables
$E y = \beta_0 + \beta_1 x_1 + \beta_2 x_j + \beta_3 x_k,$	4 models with three variables
$E y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$	1 model with all variables

- With  $k$  explanatory variables, there are  $2^k$  possible linear models
- There are infinitely many nonlinear ones!!

### Overhead C. Model validation

- Model validation is the process of confirming our proposed model.
- Concern: data-snooping - fitting many models to a single set of data.
  - Response to concern: out-of-sample validation.
  - Divide the data into model development, or training and validation, or test, subsamples.

```
par(mai=c(0,0.1,0,0))
plot.new()
plot.window(xlim=c(0,18),ylim=c(-10,10))
rect(1,-1.2,14,1.2)
rect(7,4,15,8)
rect(1,-8,6,-4)
x<-seq(1.5,9,length=6)
y<-rep(0,6)
text(x,y,labels=c(1:6),cex=1.5)
x1<-seq(10.5,11.5,length=3)
y1<-rep(0,3)
text(x1,y1,labels=rep(".",3),cex=3)
text(13,0,labels="n",cex=1.5)

text(15,0,labels="ORIGINAL\nSAMPLE\nSIZE n",adj=0)
text(7.5,6,labels="MODEL DEVELOPMENT\nSUBSAMPLE SIZE",adj=0)
text(12.5,5.3, expression(n[1]), adj=0, cex=1.1)
text(1.4,-6,labels="VALIDATION\nSUBSAMPLE\nSIZE",adj=0)
text(2.8,-7.2,expression(n[2]),adj=0, cex=1.1)

arrows(1.8,0.8,8.3,3.9,code=2,lwd=2,angle=15,length=0.2)
arrows(4.8,0.8,9,3.8,code=2,lwd=2,angle=15,length=0.2)
arrows(9.1,0.9,9.5,3.8,code=2,lwd=2,angle=15,length=0.2)
arrows(12.8,0.8,10,3.8,code=2,lwd=2,angle=15,length=0.2)
arrows(2.9,-0.9,2.5,-3.8,code=2,lwd=2,angle=15,length=0.2)
arrows(5.9,-0.9,3.1,-3.8,code=2,lwd=2,angle=15,length=0.2)
arrows(7.4,-0.9,3.5,-3.8,code=2,lwd=2,angle=15,length=0.2)
```

### 4.1.2 MC Exercise. An iterative approach to data modeling

Which of the following is not true?

- A. Diagnostic checking reveals symptoms of mistakes made in previous specifications.
- B. Diagnostic checking provides ways to correct mistakes made in previous specifications.
- C. Model formulation is accomplished by using prior knowledge of relationships.
- D. Understanding theoretical model properties is not really helpful when matching a model to data or inferring general relationships based on the data.

## 4.2 Automatic variable selection procedures

### 4.2.1 Video (Exercise). Automatic variable selection procedures

#### Learning Objectives

In this module, you learn how to:

- Identify some examples of automatic variable selection procedures
- Describe the purpose of automatic variable selection procedures and their limitations
- Describe “data-snooping”

#### Video Overheads

##### Overhead A. Classic stepwise regression algorithm

Suppose that the analyst has identified one variable as the outcome,  $y$ , and  $k$  potential explanatory variables,  $x_1, x_2, \dots, x_k$ .

- (i). Consider all possible regressions using one explanatory variable. Choose the one with the highest t-statistic.
- (ii). Add a variable to the model from the previous step. The variable to enter is with the highest t-statistic.
- (iii). Delete a variable to the model from the previous step. Delete the variable with the small t-statistic if the statistic is less than, e.g., 2 in absolute value.
- (iv). Repeat steps (ii) and (iii) until all possible additions and deletions are performed.

##### Overhead B. Drawbacks of stepwise regression

- The procedure “snoops” through a large number of models and may fit the data “too well.”
- There is no guarantee that the selected model is the best.
  - The algorithm does not consider models that are based on nonlinear combinations of explanatory variables.
  - It ignores the presence of outliers and high leverage points.

##### Overhead C. Data-snooping in stepwise regression

- Generate  $y$  and  $x_1 - x_{50}$  using a random number generator
- By design, there is no relation between  $y$  and  $x_1 - x_{50}$ .
- **But**, through stepwise regression, we “**discover**” a relationship that explains 14% of the variation!!!

Call: `lm(formula = y ~ xvar27 + xvar29 + xvar32, data = X)`

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.04885	0.09531	-0.513	0.6094
xvar27	0.21063	0.09724	2.166	0.0328 *
xvar29	0.24887	0.10185	2.443	0.0164 *
xvar32	0.25390	0.09823	2.585	0.0112 *

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9171 on 96 degrees of freedom

Multiple R-squared: 0.1401, Adjusted R-squared: 0.1132

F-statistic: 5.212 on 3 and 96 DF, p-value: 0.002233



**Overhead D. Variants of stepwise regression**

This uses the R function `step()`

- The option `direction` can be used to change how variables enter
  - Forward selection. Add one variable at a time without trying to delete variables.
  - Backwards selection. Start with the full model and delete one variable at a time without trying to add variables.
- The option `scope` can be used to specify which variables must be included

**Overhead E. Automatic variable selection procedures**

- Stepwise regression is a type of automatic variable selection procedure.
- These procedures are useful because they can quickly search through several candidate models. They mechanize certain routine tasks and are excellent at discovering patterns in data.
- They are so good at detecting patterns that they analyst must be wary of overfitting (data-snooping)
- They can miss certain patterns (nonlinearities, unusual points)
- A model suggested by automatic variable selection procedures should be subject to the same careful diagnostic checking procedures as a model arrived at by any other means

**4.2.2 Exercise. Data-snooping in stepwise regression****Assignment Text**

Automatic variable selection procedures, such as the classic stepwise regression algorithm, are very good at detecting patterns. Sometimes they are too good in the sense that they detect patterns in the sample that are not evident in the population from which the data are drawn. They detect “spurious” patterns.

This exercise illustrates this phenomenon by using a simulation, designed so that the outcome variable ( $y$ ) and the explanatory variables are mutually independent. So, by design, there is no relationship between the outcome and the explanatory variables.

As part of the code set-up, we have  $n = 100$  observations generated of the outcome  $y$  and 50 explanatory variables, `xvar1` through `xvar50`. As anticipated, collections of explanatory variables are not statistically significant. However, with the `step()` function, you will find some statistically significant relationships!

**Instructions**

- Fit a basic linear regression model and MLR model with the first ten explanatory variables. Compare the models via an F test.
- Fit a multiple linear regression model with all fifty explanatory variables. Compare this model to the one with ten variables via an F test.
- Use the `step` function to find the best model starting with the fitted model containing all fifty explanatory variables and summarize the fit.

**Hint**

The code shows stepwise regression using BIC, a criterion that results in simpler models than AIC. For AIC, use the option `k=2` in the `[step()]` function (the default)

**Pre-exercise code**

```
# Pre-exercise code
set.seed(1237)
X <- as.data.frame(matrix(rnorm(100*50, mean = 0, sd = 1), ncol = 50))
colnames(X) <- paste("xvar", 1:50, sep = "")
X$y <- with(X, matrix(rnorm(100*1, mean = 0, sd = 1), ncol = 1))
#cor(X[,c("xvar1", "xvar2", "xvar3", "xvar4", "xvar5", "xvar6", "xvar7", "xvar8", "xvar9", "xvar10", "y")], use =
```

**Sample\_code**

```
`@sample_code`
# Fit a basic linear regression model and MLR model with the first ten explanatory variables. Compare t
model_step1 <- lm(y ~ xvar1, data = X)
model_step10 <- lm(y ~ xvar1 + xvar2 + xvar3 + xvar4 + xvar5 + xvar6 + xvar7 + xvar8 + xvar9 + xvar10, d
anova(____, ____)
```

```
# Fit a multiple linear regression model with all fifty explanatory variables. Compare this model to th
model_step50 <- lm(y ~ xvar1 + xvar2 + xvar3 + xvar4 + xvar5 + xvar6 + xvar7 + xvar8 + xvar9 + xvar10 +
anova(____, ____)
```

```
# Use the `step` function, starting with the fitted model containing all fifty explanatory variables an
#For BIC:
model_stepwise <- step(____, data = X, direction= "both", k = log(nrow(X)), trace = 0)
summary(model_stepwise)
```

**Solution**

```
# Solution
model_step1 <- lm(y ~ xvar1, data = X)
model_step10 <- lm(y ~ xvar1 + xvar2 + xvar3 + xvar4 + xvar5 + xvar6 + xvar7 + xvar8 + xvar9 + xvar10, d
anova(model_step1,model_step10)
model_step50 <- lm(y ~ xvar1 + xvar2 + xvar3 + xvar4 + xvar5 + xvar6 + xvar7 + xvar8 + xvar9 + xvar10 +
anova(model_step10,model_step50)
```

```
#For BIC:
model_stepwise <- step(model_step50, data = X, direction= "both", k = log(nrow(X)), trace = 0)
summary(model_stepwise)
```

```
# An example with scope
#model_step5a <- step(model_step4, data = X, direction= "both", k=log(nrow(X)), trace = 0,
#
# scope = list(lower = ~xvar1+xvar2, upper = model_step4))
#summary(model_step5a)
#For AIC:
#step(model_step4, data = X, direction= "both", k=2, trace = 0) # k=2 is by default
```

**Submission correctness test (SCT)**

success\_msg(“Excellent! The step procedure repeatedly fits many models to a data set. We summarize each fit with hypothesis testing statistics like t-statistics and p-values. But, remember that hypothesis tests are designed to falsely detect a relationship a fraction of the time (typically 5%). For example, if you run a t-test 50 times (for each explanatory variable), you can expect to get two or three statistically significant explanatory variables even for unrelated variables (because 50 times 0.05 = 2.5).”)

## 4.3 Residual analysis

### 4.3.1 Video (Exercise). Residual analysis

**Learning Objectives**

In this module, you learn how to:

- Explain how residual analysis can be used to improve a model specification
- Use relationships between residuals and potential explanatory variables to improve model specification

## Video Overheads

### Overhead A. Residual analysis

- Use  $e_i = y_i - \hat{y}_i$  as the  $i$ th residual.
- Later, I will discuss rescaling by, for example,  $s$ , to get a standardized residual.
- Role of residuals: If the model formulation is correct, then residuals should be approximately equal to random errors or “white noise.”
- Method of attack: Look for patterns in the residuals. Use this information to improve the model specification.

### Overhead B. Using residuals to select explanatory variables

- Residual analysis can help identify additional explanatory variables that may be used to improve the formulation of the model.
- If the model is correct, then residuals should resemble random errors and contain no discernible patterns.
- Thus, when comparing residuals to explanatory variables, we do not expect any relationships.
- If we do detect a relationship, then this suggests the need to control for this additional variable.

### Overhead C. Detecting relationships between residuals and explanatory variables

- Calculate summary statistics and display the distribution of residuals to identify outliers.
- Calculate the correlation between the residuals and additional explanatory variables to search for linear relationships.
- Create scatter plots between the residuals and additional explanatory variables to search for nonlinear relationships.

## 4.3.2 Exercise. Residual analysis and risk manager survey

### Assignment Text

This exercise examines data, pre-loaded in the dataframe `survey`, from a survey on the cost effectiveness of risk management practices. Risk management practices are activities undertaken by a firm to minimize the potential cost of future losses, such as the event of a fire in a warehouse or an accident that injures employees. This exercise develops a model that can be used to make statements about cost of managing risks.

A measure of risk management cost effectiveness, `logcost`, is the outcome variable. This variable is defined as total property and casualty premiums and uninsured losses as a proportion of total assets, in logarithmic units. It is a proxy for annual expenditures associated with insurable events, standardized by company size. Explanatory variables include `logsize`, the logarithm of total firm assets, and `indcost`, a measure of the firm’s industry risk.

### Instructions

- Fit and summarize a MLR model using `logcost` as the outcome variable and `logsize` and `indcost` as explanatory variables.
- Plot residuals of the fitted model versus `indcost` and superimpose a locally fitted line using the R function `lowess()`.
- Fit and summarize a MLR model of `logcost` on `logsize`, `indcost` and a squared version of `indcost`.
- Plot residuals of the fitted model versus ‘`indcost`’ and superimpose a locally fitted line using `lowess()`.

### Hint

You can access model residuals using `mlr.survey1$residuals` or `mlr.survey1($residuals)`

### Pre-exercise code

```
# Pre-exercise code
survey <- read.csv("CSVData\\Risk_survey.csv", header=TRUE)
#survey <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/dc1c5bce43ef076a
survey$logcost <- log(survey$firmcost)
#str(survey)
```

### Sample\_code

```
`@sample_code`
# Regress `logcost` on `logsize` and `indcost`
mlr.survey1 <- lm(logcost ~ logsize + indcost, data = survey)
summary(____)

# Plot residuals of the fitted model versus `indcost` and superimpose a locally fitted line using the
plot(survey$indcost, ____ )
lines(lowess(survey$indcost, ____))

# Regress `logcost` on `logsize` and `indcost` and `indcost` squared
mlr.survey2 <- lm(____ ~ logsize + poly(indcost,2), data = survey)
summary(____)

# Plot residuals of this fitted model and superimpose a locally fitted line using the function [lowess(
plot(survey$indcost, ____ )
lines(lowess(survey$indcost, ____))
```

### Solution

```
# Solution
mlr.survey1 <- lm(logcost ~ logsize + indcost, data = survey)
summary(mlr.survey1)

plot(survey$indcost, mlr.survey1$residuals)
lines(lowess(survey$indcost,mlr.survey1$residuals))

mlr.survey2 <- lm(logcost ~ logsize + poly(indcost,2), data = survey)
summary(mlr.survey2)

plot(survey$indcost, mlr.survey2$residuals)
lines(lowess(survey$indcost,mlr.survey2$residuals))
```

### Submission correctness test (SCT)

success\_msg("Excellent! In this exercise, you examined residuals from a preliminary model fit and detected a mild quadratic pattern in a variable. This suggested entering the squared term of that variable into the model specification. The refit of this new model suggests that the squared term has important explanatory information. The squared term is a nonlinear alternative that is not available in many automatic variable selection procedures.")

## 4.3.3 Exercise. Added variable plot and refrigerator prices

### Assignment Text

What characteristics of a refrigerator are important in determining its price (**price**)? We consider here several characteristics of a refrigerator, including the size of the refrigerator in cubic feet (**rsize**), the size of the freezer compartment in cubic feet (**fsize**), the average amount of money spent per year to operate the

refrigerator (**ecost**, for energy cost), the number of shelves in the refrigerator and freezer doors (**shelves**), and the number of features (**features**). The features variable includes shelves for cans, see-through crispers, ice makers, egg racks and so on.

Both consumers and manufacturers are interested in models of refrigerator prices. Other things equal, consumers generally prefer larger refrigerators with lower energy costs that have more features. Due to forces of supply and demand, we would expect consumers to pay more for these refrigerators. A larger refrigerator with lower energy costs that has more features at the similar price is considered a bargain to the consumer. How much extra would the consumer be willing to pay for this additional space? A model of prices for refrigerators on the market provides some insight to this question.

To this end, we analyze data from  $n = 37$  refrigerators.

### Instructions

```
# Pre-exercise code
Refrig <- read.table("CSVData\\Refrig.csv", header = TRUE, sep = ",")
summary(Refrig)
Refrig1 <- Refrig[c("price", "ecost", "rsize", "fsize", "shelves", "s_sq_ft", "features")]
round(cor(Refrig1), digits = 3)
refrig_mlr1 <- lm(price ~ rsize + fsize + shelves + features, data = Refrig)
summary(refrig_mlr1)
Refrig$residuals1 <- residuals(refrig_mlr1)
refrig_mlr2 <- lm(ecost ~ rsize + fsize + shelves + features, data = Refrig)
summary(refrig_mlr2)
Refrig$residuals2 <- residuals(refrig_mlr2)
plot(Refrig$residuals2, Refrig$residuals1)

library(Rcmdr)
refrig_mlr3 <- lm(price ~ rsize + fsize + shelves + features + ecost, data = Refrig)
avPlots(refrig_mlr3, terms = "ecost")
```

## 4.4 Unusual observations

### 4.4.1 Video (Exercise). Unusual observations

#### Learning Objectives

In this module, you learn how to:

- Compare and contrast three alternative definitions of a standardized residual
- Evaluate three alternative options for dealing with outliers
- Assess the impact of a high leverage observation
- Evaluate options for dealing with high leverage observations
- Describe the notion of influence and Cook's Distance for quantifying influence

#### Video Overheads

##### Overhead A. Unusual observations

- Regression coefficients can be expressed as (matrix) weighted averages of outcomes
  - Averages, even weighted averages can be strongly influenced by unusual observations
- Observations may be unusual in the y direction or in the X space
- For unusual in the y direction, we use a residual  $e = y - \hat{y}$

- By subtracting the fitted value  $\hat{y}$ , we look to the y distance from the regression plane
- In this way, we “control” for values of explanatory variables

### Overhead B. Standardized residuals

We standardize residuals so that we can focus on relationships of interest and achieve carry-over of experience from one data set to another.

Three commonly used definitions of standardize residuals are:

$$(a) \frac{e_i}{s}, \quad (b) \frac{e_i}{s\sqrt{1-h_{ii}}}, \quad (c) \frac{e_i}{s_{(i)}\sqrt{1-h_{ii}}}.$$

- First choice is simple
- Second choice, from theory,  $\text{Var}(e_i) = \sigma^2(1 - h_{ii})$ . Here,  $h_{ii}$  is the  $i$ th leverage (defined later).
- Third choice is termed “studentized residuals”. Idea: numerator is independent of the denominator.

### Overhead C. Outlier - an unusual standardized residual

- An outlier is an observation that is not well fit by the model; these are observations where the residual is unusually large.
- Unusual means what? Many packages mark a point if the  $|\text{standardized residual}| > 2$ .
- Options for handling outliers
  - Ignore them in the analysis but be sure to discuss their effects.
  - Delete them from the data set (but be sure to discuss their effects).
  - Create a binary variable to indicator their presence. (This will increase your  $R^2$ !)

### Overhead D. High leverage points

- A high leverage point is an observation that is “far away” in the  $x$ -space from others.
- One can get a feel for high leverage observations by looking a summary statistics (mins, maxs) for each explanatory variable.
- Options for dealing with high leverage points are comparable to outliers, we can ignore their effects, delete them, or mark them with a binary indicator variable.

### Overhead E. High leverage point graph

```
library(cluster)
#library(MASS)
par(mar=c(3.2,5.4,.2,.2))
plot(1,5,type="p",pch=19,cex=1.5,xlab="",ylab="",cex.lab=1.5,xaxt="n",yaxt="n",xlim=c(-3,5),ylim=c(-12,
mtext(expression(x[2]), side=1,line=2, cex=2.0)
mtext(expression(x[1]), side=2, line=2, las=2, cex=2.0)
arrows(1.5,5,4,5,code=1,lwd=2,angle=15,length=0.25)
xycov<-matrix(c(2, -5,-5, 20),nrow=2,ncol=2)
xyloc<-matrix(c(0, 0),nrow=1,ncol=2)
polygon(ellipsePoints(xycov, d2 = 2, loc=xyloc),col="black")
```

### Overhead F. Leverage

- Using matrix algebra, one can express the  $i$ th fitted value as a linear combination of observations

$$\hat{y}_i = h_{i1}y_1 + \cdots + h_{ii}y_i + \cdots + h_{in}y_n.$$

- The term  $h_{ii}$  is known as the  $i$ th leverage
  - The larger the value of  $h_{ii}$ , the greater the effect of the  $i$ th observation  $y_i$  on the  $i$ th fitted value  $\hat{y}_i$ .
  - Statistical routines have values of the leverage coded, so computing this quantity. The key thing to know is that  $h_{ii}$  is based solely on the explanatory variables. If you change the  $y$  values, the leverage does not change.

- As a commonly used rule of thumb, a leverage is deemed to be “unusual” if its value exceeds three times the average (= number of regression coefficients divided by the number of observations.)

### 4.4.2 Exercise. Outlier example

In chapter 2, we consider a fictitious data set of 19 “base” points plus three different types of unusual points. In this exercise, we consider the effect of one unusual point, “C”, this both an outlier (unusual in the “y” direction) and a high leverage point (usual in the x-space). The data have been pre-loaded in the dataframe `outlrC`.

#### Instructions

- Fit a basic linear regression model of `y` on `x` and store the result in an object.
- Use the function `rstandard()` to extract the standardized residuals from the fitted regression model object and summarize them.
- Use the function `hatvalues()` to extract the leverages from the model fitted and summarize them.
- Plot the standardized residuals versus the leverages to see the relationship between these two measures that calibrate how unusual an observation is.

#### Hint

#### Pre-exercise code

```
# Pre-exercise code
outlr <- read.csv("CSVData\\Outlier.csv", header = TRUE)
#outlr <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/7a38912e544c31fc6...")
outlrC <- outlr[-c(20,21),c("x","y")]
```

#### Sample code

```
`@sample_code`
outlrC <- outlr[-c(20,21),c("x","y")]

# Fit a basic linear regression model of `y` on `x` and store the result in an object.
model_outlrC <- lm(y ~ x, data = outlrC)

# Extract the standardized residuals from the fitted regression model object and summarize them.
ri <- rstandard(model_outlrC)
summary(ri)

# Extract the leverages from the model fitted and summarize them.
hii <- hatvalues(model_outlrC)
summary(hii)

# Plot the standardized residuals versus the leverages
plot(hii,ri)
```

#### Solution

```
# solution
plot(outlrC)
model_outlrC <- lm(y ~ x, data = outlrC)
ri <- rstandard(model_outlrC)
summary(ri)
hii <- hatvalues(model_outlrC)
summary(hii)
plot(hii,ri)
```

### Submission correctness test (SCT)

success\_msg("Excellent! With only two variables, we could argue graphically that observations were unusual. In this exercise, we showed how statistics could also be used to identify usual observations. Although not really necessary in basic linear regression, the main advantage of the statistics is that they work readily in a multivariate setting.")

### 4.4.3 Exercise. High leverage and risk manager survey

#### Assignment Text

In a prior exercise, we fit a regression model of `logcost` on `logsize`, `indcost` and a squared version of `indcost`. This model is summarized in the object `mlr_survey2`. In this exercise, we examine the robustness of the model to unusual observations.

#### Instructions

- Use the R functions `rstandard()` and `hatvalues()` to extract the standardized residuals and leverages from the model fitted. Summarize the distributions graphically.
- You will see that there are two observations where the leverages are high, numbers 10 and 16. On looking at the dataset, these turn out to be observations in a high risk industry. Create a histogram of the variable `indcost` to corroborate this.
- Re-run the regression omitting observations 10 and 16. Summarize this regression and the regression in the object `mlr_survey2`, noting differences in the coefficients.

#### Hint

#### Pre-exercise code

```
# Pre-exercise code
survey <- read.csv("CSVData\\Risk_survey.csv", header=TRUE)
#survey <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/dc1c5bce43ef076a
survey$logcost <- log(survey$firmcost)
mlr_survey2 <- lm(logcost ~ logsize + poly(indcost,2), data = survey)
```

#### Sample\_code

```
`@sample_code`
mlr_survey2 <- lm(logcost ~ logsize + poly(indcost,2), data = survey)
# Extract the standardized residuals and leverages from the model fitted. Summarize the distributions g
ri <- ____(mlr_survey2)
hii <- ____(mlr_survey2)
par(mfrow=c(1, 2))
hist(ri, nclass=16, main="", xlab="Standardized Residuals")
hist(hii, nclass=16, main="", xlab="Leverages")

# Create a histogram of the variable `indcost`
par(mfrow=c(1, 1))
hist(___, nclass=16)

# Re-run the regression omitting observations 10 and 16. Summarize this regression and the regression i
mlr_survey3 <- lm(___ ~ logsize + poly(indcost,2), data = survey, subset =-c(10,16))
summary(mlr_survey2)
summary(mlr_survey3)
```

#### Solution



```
# Solution
#summary(mlr.survey2)
ri <- rstandard(mlr.survey2)
hii <- hatvalues(mlr.survey2)

par(mfrow=c(1, 2))
hist(ri, nclass=16, main="", xlab="Standardized Residuals")
hist(hii, nclass=16, main="", xlab="Leverages")
par(mfrow=c(1, 1))
hist(survey$indcost, nclass=16)
mlr.survey3 <- lm(logcost ~ logsize + poly(indcost,2), data = survey, subset =-c(10,16))
summary(mlr.survey2)
summary(mlr.survey3)
```

### Submission correctness test (SCT)

success\_msg("Excellent! You will have noted that after removing these two influential observations from a high risk industry, the variable associated with the `indcost` squared became less statistically significant. This illustrates a general phenomena; sometimes, the 'significance' of a variable may actually due to a few unusual observations, not the entire variable.")

## 4.5 Collinearity

### 4.5.1 Video (Exercise). Collinearity

#### Learning Objectives

In this module, you learn how to:

- Define collinearity and describe its potential impact on regression inference
- Define a variance inflation factor and describe its effect on a regression coefficients standard error
- Describe rules of thumb for assessing collinearity and options for model reformulation in the presence of severe collinearity
- Compare and contrast effects of leverage and collinearity

#### Video Overheads

##### Overhead A. Collinearity

- Collinearity, or multicollinearity, occurs when one explanatory variable is, or nearly is, a linear combination of the other explanatory variables.
  - Useful to think of the explanatory variables as being highly correlated with one another.
- Collinearity neither precludes us from getting good fits nor from making predictions of new observations.
  - Estimates of error variances and, therefore, tests of model adequacy, are still reliable.
- In cases of serious collinearity, standard errors of individual regression coefficients can be large.
  - With large standard errors, individual regression coefficients may not be meaningful.
  - Because a large standard error means that the corresponding t-ratio is small, it is difficult to detect the importance of a variable.

##### Overhead B. Quantifying collinearity

A common way to quantify collinearity is through the variance inflation factor (VIF).

- Suppose that the set of explanatory variables is labeled  $x_1, x_2, \dots, x_k$ .
- Run the regression using  $x_j$  as the “outcome” and the other  $x$ ’s as the explanatory variables.
- Denote the coefficient of determination from this regression by  $R_j^2$ .
- Define the variance inflation factor

$$VIF_j = \frac{1}{1 - R_j^2}, \quad \text{for } j = 1, 2, \dots, k.$$

### Overhead C. Options for handling collinearity

- Rule of thumb: When  $VIF_j$  exceeds 10 (which is equivalent to  $R_j^2 > 90\%$ ), we say that severe collinearity exists. This may signal a need for action.
- Recode the variables by “centering” - that is, subtract the mean and divide by the standard deviation.
- Ignore the collinearity in the analysis but comment on it in the interpretation. Probably the most common approach.
- Replace one or more variables by auxiliary variables or transformed versions.
- Remove one or more variables. Easy. Which One? is hard.
  - Use interpretation. Which variable(s) do you feel most comfortable with?
  - Use automatic variable selection procedures to suggest a model.

### 4.5.2 Exercise. Collinearity and term life

**Assignment Text** We have seen that adding an explanatory variable  $x^2$  to a model is sometimes helpful even though it is perfectly related to  $x$  (such as through the function  $f(x) = x^2$ ). But, for some data sets, higher order polynomials and interactions can be approximately linearly related (depending on the range of the data).

This exercise returns to our term life data set **Term1** (preloaded) and demonstrates that collinearity can be severe when introducing interaction terms.

#### Instructions

- Fit a MLR model of **logface** on explanatory variables **education**, **numhh** and **logincome**
- Use the function `vif()` from the **car** package (preloaded) to calculate variance inflation factors.
- Fit and summarize a MLR model of **logface** on explanatory variables **education**, **numhh** and **logincome** with an interaction between **numhh** and **logincome**, then extract variance inflation factors.

#### Hint

If the **car** package is not available to you, then you could calculate vifs using the `[lm()]` function, treating each variable separately. For example

```
1/(1-summary(lm(education ~ numhh + logincome, data = Term1))$r.squared)
```

gives the **education** vif.

#### Pre-exercise code

```
# Pre-exercise code
Term <- read.csv("CSVData\\term_life.csv", header = TRUE)
#Term <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/efc64bc2d78cf6b48a
Term1 <- subset(Term, subset = face > 0)
#str(Term1)
```

\*Sample\_code\*\*

```
`@sample_code`
# Fit a MLR model of `logface` on explanatory variables `education`, `numhh` and `logincome`
Term_mlr <- lm(logface ~ education + numhh + logincome, data = Term1)
```

```
# Calculate the variance inflation factors.
car::vif(Term_mlr)

# Fit and summarize a MLR model of `logface` on explanatory variables `education`, `numhh` and `logincome`
Term_mlr1 <- lm(logface ~ education + numhh*logincome, data = Term1)
summary(Term_mlr1)
car::vif(Term_mlr1)
```

### Solution

```
# Solution
Term_mlr <- lm(logface ~ education + numhh + logincome, data = Term1)
car::vif(Term_mlr)
Term_mlr1 <- lm(logface ~ education + numhh*logincome, data = Term1)
summary(Term_mlr1)
car::vif(Term_mlr1)
```

### Submission correctness test (SCT)

success\_msg(“Excellent! This exercise underscores that collinearity among explanatory variables can be induced when introducing higher order terms such as interactions. Note that in the interaction model the variable ‘numhh’ does not appear to be statistically significant effect. This is one of the big dangers of collinearity - it can mask important effects.”)

## 4.6 Selection criteria

### 4.6.1 Video (Exercise). Selection criteria

#### Learning Objectives

In this module, you learn how to:

- Summarize a regression fit using alternative goodness of fit measures
- Validate a model using in-sample and out-of-sample data to mitigate issues of data-snooping
- Compare and contrast SSPE and PRESS statistics for model validation

#### Video Overheads

##### Overhead A. Goodness of fit

- Criteria that measure the proximity of the fitted model and realized data are known as goodness of fit statistics.
- Basic examples include:
  - the coefficient of determination ( $R^2$ ),
  - an adjusted version ( $R_a^2$ ),
  - the size of the typical error ( $s$ ), and
  - $t$ -ratios for each regression coefficient.

##### Overhead B. Goodness of fit and information criteria

A general measure is Akaike’s Information Criterion, defined as

$$AIC = -2 \times (\text{fitted log likelihood}) + 2 \times (\text{number of parameters})$$

- For model comparison, the smaller the  $AIC$ , the better is the fit.
- This measures balances the fit (in the first part) with a penalty for complexity (in the second part)
- It is a general measure - for linear regression, it reduces to

$$AIC = n \ln(s^2) + n \ln(2\pi) + n + k + 3.$$

- So, selecting a model to minimize  $s$  or  $s^2$  is equivalent to model selection based on minimizing  $AIC$  (same  $k$ ).

#### Overhead C. Out of sample validation

- When you choose a model to minimize  $s$  or  $AIC$ , it is based on how well the model fits the data at hand, or the model development, or training, data
- As we have seen, this approach is susceptible to overfitting.
- A better approach is to validate the model on a model validation, or test data set, held out for this purpose.

#### Overhead D. Out of sample validation procedure

- (i) Using the model development subsample, fit a candidate model.
- (ii) Using the Step (ii) model and the explanatory variables from the validation subsample, “predict” the dependent variables in the validation subsample,  $\hat{y}_i$ , where  $i = n_1 + 1, \dots, n_1 + n_2$ .
- (iii) Calc the \*sum of absolute prediction errors\*\*

$$SAPE = \sum_{i=n_1+1}^{n_1+n_2} |y_i - \hat{y}_i|.$$

Repeat Steps (i) through (iii) for each candidate model. Choose the model with the smallest SAPE.

#### Overhead E. Cross - validation

- With out-of-sample validation, the statistic depends on a random split between in-sample and out-of-sample data (a problem for data sets that are not large)
- Alternatively, one may use cross-validation
  - Use a random mechanism to split the data into  $k$  subsets, (e.g., 5-10)
  - Use the first  $k-1$  subsamples to estimate model parameters. Then, “predict” the outcomes for the  $k$ th subsample and use SAE to summarize the fit
  - Repeat this by holding out each of the  $k$  sub-samples, summarizing with a cumulative SAE.
- Repeat these steps for several candidate models.
  - Choose the model with the lowest cumulative SAE statistic.

### 4.6.2 Exercise. Cross-validation and term life

#### Assignment Text

Here is some sample code to give you a better feel for cross-validation.

The first part of the randomly re-orders (“shuffles”) the data. It also identifies explanatory variables `explvars`.

The function starts by pulling out only the needed data into `cvdata`. Then, for each subsample, a model is fit based on all the data except for the subsample, in `train_mlr` with the subsample in `test`. This is repeated for each subsample, then results are summarized.

```
# Randomly re-order data - "shuffle it"
n <- nrow(Term1)
set.seed(12347)
shuffled_Term1 <- Term1[sample(n), ]
explvars <- c("education", "numhh", "logincome")

## Cross - Validation
crossvalfct <- function(explvars){
  cvdata <- shuffled_Term1[, c("logface", explvars)]
  crossval <- 0
  k <- 5
  for (i in 1:k) {
    indices <- (((i-1) * round((1/k)*nrow(cvdata))) + 1):((i*round((1/k) * nrow(cvdata))))
    # Exclude them from the train set
    train_mlr <- lm(logface ~ ., data = cvdata[-indices,])
    # Include them in the test set
    test <- data.frame(cvdata[indices, explvars])
    names(test) <- explvars
    predict_test <- exp(predict(train_mlr, test))
    # Compare predicted to held-out and summarize
    predict_err <- exp(cvdata[indices, "logface"]) - predict_test
    crossval <- crossval + sum(abs(predict_err))
  }
  crossval/1000
}

crossvalfct(explvars)
```

### Instructions

- Calculate the cross-validation statistic using only logarithmic income, `logincome`.
- Calculate the cross-validation statistic using `logincome`, `education` and `numhh`.
- Calculate the cross-validation statistic using `logincome`, `education`, `numhh` and `marstat`.

The best model has the lowest cross-validation statistic.

### Hint

The function `[sample()]` is for taking random samples. We use it without replacement so it results in a re-ordering of data.

### Pre-exercise code

```
# Pre-exercise code
Term <- read.csv("CSVDData\\term_life.csv", header = TRUE)
#Term <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/efc64bc2d78cf6b48a
Term1 <- subset(Term, subset = face > 0)
Term1$marstat <- as.factor(Term1$marstat)

crossvalfct <- function(explvars){
  cvdata <- shuffled_Term1[, c("logface", explvars)]
  crossval <- 0
  k <- 5
  for (i in 1:k) {
    indices <- (((i-1) * round((1/k)*nrow(cvdata))) + 1):((i*round((1/k) * nrow(cvdata))))
    # Exclude them from the train set
    train_mlr <- lm(logface ~ ., data = cvdata[-indices,])
```

```

# Include them in the test set
test <- data.frame(cvdata[indices, explvars])
names(test) <- explvars
predict_test <- exp(predict(train_mlr, test))
# Compare predicted to held-out and summarize
predict_err <- exp(cvdata[indices, "logface"]) - predict_test
crossval <- crossval + sum(abs(predict_err))
}
crossval/1000000
}

```

### Sample\_code

```

`@sample_code`
# Calculate the cross-validation statistic using only logarithmic income, `logincome`.
explvars <- c("logincome")
crossvalfct(explvars)

# Calculate the cross-validation statistic using `logincome`, `education` and `numhh`.
explvars <- c("education", "numhh", "logincome")
crossvalfct(___)

# Calculate the cross-validation statistic using `logincome`, `education`, `numhh` and `marstat`.
explvars <- c(___)
crossvalfct(explvars)

```

### Solution

```

# Solution
# Randomly re-order data - "shuffle it"
n <- nrow(Term1)
set.seed(12347)
shuffled_Term1 <- Term1[sample(n), ]
## Cross - Validation
explvars <- c("logincome")
crossvalfct(explvars)
explvars <- c("education", "numhh", "logincome")
crossvalfct(explvars)
explvars <- c("education", "numhh", "logincome", "marstat")
crossvalfct(explvars)

```

### Submission correctness test (SCT)

success\_msg("Excellent! This exercise demonstrates the use of cross-validation, a very important technique in model selection. The exercise builds the procedure from the ground up so that you can see all the steps involved. Further, it illustrates how you can develop your own functions to automate procedures and save steps.")

## Chapter 5

# Interpreting Regression Results

### Chapter description

A case study, on determining an individual's characteristics that influence its health expenditures, illustrates the regression modeling process from start to finish. Subsequently, the chapter summarizes what we learn from the modeling process, underscoring the importance of variable selection.

## 5.1 Case study: MEPS health expenditures

### 5.1.1 Video (Exercise). Case study: MEPS health expenditures

#### Learning Objectives

In this module, you learn how to:

- 

#### Video Overheads

##### Overhead A. MEPS health expenditures

This exercise considers data from the Medical Expenditure Panel Survey (MEPS), conducted by the U.S. Agency of Health Research and Quality. MEPS is a probability survey that provides nationally representative estimates of health care use, expenditures, sources of payment, and insurance coverage for the U.S. civilian population. This survey collects detailed information on individuals of each medical care episode by type of services including physician office visits, hospital emergency room visits, hospital outpatient visits, hospital inpatient stays, all other medical provider visits, and use of prescribed medicines. This detailed information allows one to develop models of health care utilization to predict future expenditures. You can learn more about MEPS at <http://www.meps.ahrq.gov/mepsweb/>.

We consider MEPS data from the panels 7 and 8 of 2003 that consists of 18,735 individuals between ages 18 and 65. From this sample, we took a random sample of 2,000 individuals that appear in the file `HealthExpend`. From this sample, there are 1,352 that had positive outpatient expenditures.

Our dependent variable is the amount of expenditures for outpatient visits, `expndop`. For MEPS, outpatient events include hospital outpatient department visits, office-based provider visits and emergency room visits excluding dental services. (Dental services, compared to other types of health care services, are more predictable and occur in a more regular basis.) Hospital stays with the same date of admission and discharge, known as “zero-night stays,” were included in outpatient counts and expenditures. (Payments associated

with emergency room visits that immediately preceded an inpatient stay were included in the inpatient expenditures. Prescribed medicines that can be linked to hospital admissions were included in inpatient expenditures, not in outpatient utilization.)

### Overhead B. Overhead MEPS health expenditures

Data from the Medical Expenditure Panel Survey (MEPS), conducted by the U.S. Agency of Health Research and Quality (AHRQ).

- A probability survey that provides nationally representative estimates of health care use, expenditures, sources of payment, and insurance coverage for the U.S. civilian population.
- Collects detailed information on individuals of each medical care episode by type of services including
  - physician office visits,
  - hospital emergency room visits,
  - hospital outpatient visits,
  - hospital inpatient stays,
  - all other medical provider visits, and
  - use of prescribed medicines.
- This detailed information allows one to develop models of health care utilization to predict future expenditures.
- We consider MEPS data from the first panel of 2003 and take a random sample of  $n = 2,000$  individuals between ages 18 and 65.

### Overhead C. Outcome variable

Our dependent variable is expenditures for outpatient admissions.

- For MEPS, inpatient admissions include persons who were admitted to a hospital and stayed overnight.
- In contrast, outpatient events include hospital outpatient department visits, office-based provider visits and emergency room visits excluding dental services.
  - Hospital stays with the same date of admission and discharge, known as “zero-night stays,” were included in outpatient counts and expenditures.
  - Payments associated with emergency room visits that immediately preceded an inpatient stay were included in the inpatient expenditures.
  - Prescribed medicines that can be linked to hospital admissions were included in inpatient expenditures, not in outpatient utilization.

### Overhead D. Explanatory variables

9 variables in the database. Here 13 most relevant.

<i>expendop</i>	Amounts of expenditures for outpatient visits
<i>gender</i>	Indicate gender of patient (=1 if female, =0 if male)
<i>age</i>	Age in years between 18 and 65
<i>race</i>	Race of patient described as Asian, Black, Native, White and other
<i>region</i>	Region of patient described as WEST, NORTHEAST, MIDWEST and SOUTH
<i>educ</i>	Level of education received described by words (LHIGHSC, HIGHSCH and COLLEGE)
<i>phstat</i>	Self-rated physical health status described as EXCE, VGOO, GOOD, FAIR and POOR
<i>mpoor</i>	Self-rated mental health (=1 if poor or fair, =0 if good to excellent mental health)
<i>anylimit</i>	Any activity limitation (=1 if any functional/activity limitation, =0 if otherwise)
<i>income</i>	Income compared to poverty line described as POOR, NPOOR, LINCOME, MINCOME and HINCOME
<i>insure</i>	Insurance coverage (=1 if covered by public/private health insurance in any month of 1996, =0 otherwise)
<i>usc</i>	1 if dissatisfied with one’s usual source of care
<i>unemploy</i>	Employment status of patients
<i>managedcare</i>	1 if enrolled in an HMO or gatekeeper plan

### Overhead E. Case study outline



The next series of exercises leads you through an analysis of the steps for understanding a complex data set. Because of the complexity of the data, in each step only a sample of procedures will be executed.

The outline consists of:

- Summary statistics
- Splitting the data into training and testing portions with initial model fits
- Selecting variables to be included in the model

### 5.1.2 Exercise. Summarizing data

#### Assignment Text

With a complex dataset, you will probably want to take a look at the structure of the data. You are already familiar with taking a `[summary()]` of a dataframe which provides summary statistics for many variables. You will see that several variables in this dataframe are categorical, or factor, variables. We can use the `table()` function to summarize them.

After getting a sense of the distributions of explanatory variables, we want to take a deeper dive into the distribution of the outcome variable, `expndop`. We will do this by comparing the histograms of the variable to that of its logarithmic version.

To examine relationships of the outcome variable visually, we look to scatterplots for continuous variables (such as `age`) and boxplots for categorical variables (such as `phstat`).

#### Instructions

- Examine the structure of the `meps` dataframe using the `str()` function. Also, get a `[summary()]` of the dataframe.
- Examine the distribution of the `race` variable using the `table()` function.
- Compare the expenditures distribution to its logarithmic version visually via histograms plotted next to another. `par(mfrow = c(1, 2))` is used to organize the plots you create.
- Examine the distribution of logarithmic expenditures in terms of levels of `phstat` visually using the `boxplot()` function.
- Examine the relationship of age versus logarithmic expenditures using a scatter plot. Superimpose a local fitting line using the `lines()` and `lowess()` functions.

#### Hint

#### Pre-exercise code

```
# Pre-exercise code
meps <- read.csv("CSVData\\HealthMeps.csv", header = TRUE)
#meps <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/7b7dab6d0c528e4cd2,
```

#### Sample\_code

```
`@sample_code`
# Examine the structure and get a summary of the `meps` dataframe
str(___)
summary(___)

# Examine the distribution of the `race` variable
table(___)

# Compare the expenditures distribution to its logarithmic version visually
par(mfrow = c(1, 2))
hist(___, main = "", xlab = "outpatient expenditures")
hist(log(___), main = "", xlab = "log expenditures")
```

```
# Examine the distribution of logarithmic expenditures in terms of levels of `phstat`
par(mfrow = c(1, 1))
meps$logexpend <- log(meps$expendop)
boxplot(logexpend ~ ___, data = meps, main = "boxplot of log expend")

# Examine the relationship of age versus logarithmic expenditures. Superimpose a local fitting line.
plot(___, ___, xlab = "age", ylab = "log expend")
lines(lowess(___, ___), col="red")
```

### Solution

```
# Solution
str(meps)
summary(meps)
table(meps$race)
par(mfrow = c(1, 2))
hist(meps$expendop, main = "", xlab = "outpatient expenditures")
hist(log(meps$expendop), main = "", xlab = "log expenditures")
par(mfrow = c(1, 1))
meps$logexpend <- log(meps$expendop)
boxplot(logexpend ~ phstat, data = meps, main = "boxplot of log expend")
plot(meps$age, meps$logexpend, xlab = "age", ylab = "log expend")
lines(lowess(meps$age, meps$logexpend), col="red")
```

### Submission correctness test (SCT)

success\_msg("Excellent! Summarizing data, without reference to a model, is probably the most time-consuming part of any predictive modeling exercise. Summary statistics are also a key part of any report as they illustrate features of the data that are accessible to a broad audience.")

## 5.1.3 Exercise. Fit a benchmark multiple linear regression model

### Assignment Text

As part of the pre-processing for the model fitting, we will split the data into training and test subsamples. For this exercise, we use a 75/25 split although other choices are certainly suitable. Some analysts prefer to do this splitting before looking at the data. Another approach, adopted here, is that the final report typically contains summary statistics of the entire data set and so it makes sense to do so when examining summary statistics.

We start by fitting a benchmark model. It is common to use all available explanatory variables with the outcome on the original scale and so we use this as our benchmark model. This exercise shows that when you plot() a fitted linear regression model in R, the result provides four graphs that you have seen before. These can be useful for identifying an appropriate model.

### Instructions

- Randomly split the data into a training and a testing data sets. Use 75% for the training, 25% for the testing.
- Fit a full model using `expendop` as the outcome and all explanatory variables. Summarize the results of this model fitting.
- You can plot() the fitted model to view several diagnostic plots. These plots provide evidence that expenditures may not be the best scale for linear regression.
- Fit a full model using `logexpend` as the outcome and all explanatory variables and summarize the fit. Use the plot() function for evidence that this variable is more suited for linear regression methods than expenditures on the original scale.

**Hint**

A plot of a regression object such as `plot(mlr)` provides four diagnostic plots. These can be organized as a 2 by 2 array using `par(mfrow = c(2, 2))`.

**Pre-exercise code**

```
# Pre-exercise code
meps <- read.csv("CSVData\\HealthMeps.csv", header = TRUE)
#meps <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/7b7dab6d0c528e4cd2",
meps$logexpend <- log(meps$expendop)
```

**Sample\_code**

```
`@sample_code`
# Randomly split the data into a training and a testing data sets. Use 75% for the training, 25% for
n <- nrow(meps)
set.seed(12347)
shuffled_meps <- meps[sample(n), ]
train_indices <- 1:round(0.75 * n)
train_meps <- shuffled_meps[train_indices, ]
test_indices <- (round(0.25 * n) + 1):n
test_meps <- shuffled_meps[test_indices, ]

# Fit a full model using `expendop` as the outcome and all explanatory variables. Summarize the results
meps_mlr1 <- lm(____ ~ gender + age + race + region + educ + phstat + mpoor + anylimit + income + insure
summary(meps_mlr1)

# Provide diagnostic plots of the fitted model.
par(mfrow = c(2, 2))
plot(____)

# Fit a full model using `logexpend` as the outcome and all explanatory variables. Summarize the fit and
meps_mlr2 <- lm(____ ~ gender + age + race + region + educ + phstat + mpoor + anylimit + income + insure
summary(meps_mlr2)
plot(meps_mlr2)
```

**Solution**

```
# Solution

# Split the sample into a `training` and `test` data
n <- nrow(meps)
set.seed(12347)
shuffled_meps <- meps[sample(n), ]
train_indices <- 1:round(0.75 * n)
train_meps <- shuffled_meps[train_indices, ]
test_indices <- (round(0.25 * n) + 1):n
test_meps <- shuffled_meps[test_indices, ]

meps_mlr1 <- lm(expendop ~ gender + age + race + region + educ + phstat + mpoor + anylimit + income + insure
summary(meps_mlr1)
par(mfrow = c(2, 2))
plot(meps_mlr1)

meps_mlr2 <- lm(logexpend ~ gender + age + race + region + educ + phstat + mpoor + anylimit + income + insure
summary(meps_mlr2)
```

```
plot(meps_mlr2)
```

### Submission correctness test (SCT)

success\_msg(“Excellent! You may have compared the four diagnostic graphs from the MLR model fit of ‘expend’ to those created using the same procedure but with logarithmic expenditures as the outcome. This provides another piece of evidence that log expenditures are more suitable for regression modeling. Using logarithmic outcomes is a common feature of actuarial applications but can be difficult to diagnose and interpret without practice.”)

## 5.1.4 Exercise. Variable selection

### Assignment Text

Modeling building can be approached using a “ground-up” strategy, where the analyst introduces a variable, examines residuals from a regression fit, and then seeks to understand the relationship between these residuals and other available variables so that these variables might be added to the model.

Another approach is a “top-down” strategy where all available variables are entered into a model and unnecessary variables are pruned from the model. Both approaches are helpful when using data to specify models. This exercise illustrates the latter approach, using the `step()` function to help narrow our search for the best fitting model.

### Instructions

From our prior work, the training dataframe `train_meps` has already been loaded in. A multiple linear regression model fit object `meps_mlr2` is available that summarizes a fit of `logexpend` as the outcome variable using all 13 explanatory variables.

- Use the `step()` function to drop unnecessary variables from the full fitted model summarized in the object `meps_mlr2` and summarize this recommended model.
- As an alternative, use the explanatory variables in the recommended model and add the variable `phstat`. Summarize the fit and note that statistical significance of the new variable.
- You have been reminded by your boss that use of the variable `gender` is unsuitable for actuarial pricing purposes. As an alternative, drop `gender` from the recommended model (still keeping `phstat`). Note the statistical significance of the variable `uscwith` with this fitted model.

### Hint

#### Pre-exercise code

```
# Pre-exercise code
meps <- read.csv("CSVData\\HealthMeps.csv", header = TRUE)
#meps <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/7b7dab6d0c528e4cd2...")
meps$logexpend <- log(meps$expendop)
# Split the sample into a `training` and `test` data
n <- nrow(meps)
set.seed(12347)
shuffled_meps <- meps[sample(n), ]
train_indices <- 1:round(0.75 * n)
train_meps <- shuffled_meps[train_indices, ]
test_indices <- (round(0.25 * n) + 1):n
test_meps <- shuffled_meps[test_indices, ]
```

#### Sample\_code

```
@sample_code
meps_mlr2 <- lm(logexpend ~ gender + age + race + region + educ + phstat + mpoor + anylimit + income +
```

```
# Use the step() to drop unnecessary variables from the full fitted model summarized in the object `meps`
model_stepwise <- step(meps_mlr2, data = ___, direction= "both", k = log(nrow(X)), trace = 0)
summary(model_stepwise)

# As an alternative, use the explanatory variables in the recommended model and add the variables `mpoor`
meps_mlr4 <- lm(___ ~ gender + age + phstat + anylimit + insure + ___, data = train_meps)
summary(meps_mlr4)

# You have been reminded by your boss that use of the variable `gender` is unsuitable for actuarial pri
meps_mlr5 <- lm(logexpend ~ age + phstat + anylimit + insure + ___, data = train_meps)
summary(___)
```

### Solution

```
# Solution
meps_mlr2 <- lm(logexpend ~ gender + age + race + region + educ + phstat + mpoor + anylimit + income + )
#library(Rcmdr)
#temp <- stepwise(meps_mlr2, direction = 'backward/forward')
model_stepwise <- step(meps_mlr2, data = X, direction= "both", k = log(nrow(X)), trace = 0)
summary(model_stepwise)
meps_mlr3 <- lm(logexpend ~ gender + age + phstat + anylimit + insure , data = train_meps)
summary(meps_mlr3)
meps_mlr4 <- lm(logexpend ~ gender + age + phstat + anylimit + insure + mpoor, data = train_meps)
summary(meps_mlr4)
meps_mlr5 <- lm(logexpend ~ age + phstat + anylimit + insure + mpoor, data = train_meps)
summary(meps_mlr5)

# par(mfrow = c(2, 2))
# plot(meps_mlr3)
#
# meps_mlr4 <- lm(logexpend ~ gender + age + mpoor + anylimit + insure + usc + phstat, data = train_meps)
# summary(meps_mlr4)
#
#
# meps_mlr5 <- lm(logexpend ~ age + anylimit + mpoor + insure + usc + phstat, data = train_meps)
# summary(meps_mlr5)
# anova(meps_mlr4, meps_mlr5)
#
# #boxplot(train_meps$logexpend ~ train_meps$phstat*train_meps$usc)
```

### Submission correctness test (SCT)

success\_msg("Excellent! Sometimes variables may have good predictive power but are unacceptable for policy purposes - in insurance, ethnicity and sometimes sex are good examples. This implies that model interpretation can be just as important as the ability to predict.")

### 5.1.5 Exercise. Model comparisons using cross-validation

#### Assignment Text

To compare alternative models, you decide to utilize cross-validation. For this exercise, you split the training sample into six subsamples of approximately equal size.

In the sample code, the cross-validation procedure has been summarized into a function that you can call. The input to the function is a list of variables that you select as your model explanatory variables. With

this function, you can readily test several candidate models.

### Instructions

- Run the cross validation (`crossvalfct`) function using the explanatory variables suggested by the stepwise function.
- Run the function again but adding the `mpoor` variable
- Run the function again but omitting the `gender` variable

Note which model is suggested by the cross validation function.

### Hint

The cross validation function of this is very similar to the one we did earlier. Different number of subsamples, different test/training data and a different outcome variable. Except for these minor changes, it is the same function that we worked with earlier.

### Pre-exercise code

```
# Pre-exercise code
meps <- read.csv("CSVData\\HealthMeps.csv", header = TRUE)
#meps <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/7b7dab6d0c528e4cd2...")
meps$logexpend <- log(meps$expendop)
# Split the sample into a `training` and `test` data
n <- nrow(meps)
set.seed(12347)
shuffled_meps <- meps[sample(n), ]
train_indices <- 1:round(0.75 * n)
train_meps <- shuffled_meps[train_indices, ]
test_indices <- (round(0.25 * n) + 1):n
test_meps <- shuffled_meps[test_indices, ]

## Cross - Validation

crossvalfct <- function(explvars){
  cvdata <- train_meps[, c("logexpend", explvars)]
  crossval <- 0
  for (i in 1:6) {
    indices <- (((i-1) * round((1/6)*nrow(cvdata))) + 1):((i*round((1/6) * nrow(cvdata))))
    # Exclude them from the train set
    train_mlr <- lm(logexpend ~ ., data = cvdata[-indices,])
    # Include them in the test set
    test <- data.frame(cvdata[indices, explvars])
    names(test) <- explvars
    predict_test <- exp(predict(train_mlr, test))
    # Compare predicted to held-out and summarize
    predict_err <- exp(cvdata[indices, "logexpend"]) - predict_test
    crossval <- crossval + sum(abs(predict_err))
  }
  crossval/1000000
}
```

### Sample\_code

```
`@sample_code`
# Run the cross validation (`crossvalfct`) function using the explanatory variables suggested by the st
explvars <- c("gender", "age", "phstat", "anylimit", "insure")
crossvalfct(explvars)
```

```
# Run the function again but adding the `mpoor` variable
explvars <- c(____)
crossvalfct(explvars)

# Run the function again but omitting the `gender` variable
explvars <- c( ____ )
crossvalfct(explvars)
```

### Solution

```
#Solution

explvars <- c("gender", "age", "phstat", "anylimit", "insure")
crossvalfct(explvars)
explvars <- c("gender", "age", "phstat", "anylimit", "insure", "mpoor")
crossvalfct(explvars)
explvars <- c("age", "phstat", "anylimit", "insure", "mpoor")
crossvalfct(explvars)
```

### Submission correctness test (SCT)

success\_msg("Excellent! Cross-validation has become an essential piece of the data analysts toolkit. Good that you now have additional experience with it.")

## 5.1.6 Exercise. Out of sample validation

### Assignment Text

From our prior work, the training `train_meps` and test `test_meps` dataframes have already been loaded in. We think our best model is based on logarithmic expenditures as the outcome and the following explanatory variables:

```
explvars3 <- c("gender", "age", "phstat", "anylimit", "insure", "mpoor")
```

We will compare this to a benchmark model that is based on expenditures as the outcome and all 13 explanatory variables

```
explvars4 <- c(explvars3, "race", "income", "region", "educ", "unemploy", "managedcare", "usc")
```

The comparisons will be based on expenditures in dollars using the held-out validation sample.

### Instructions

- Use the training sample to fit a linear model with `logexpend` and explanatory variables listed in `explvars3`
- Predict expenditures (not logged) for the test data and summarize the fit using the sum of absolute prediction errors.
- Use the training sample to fit a benchmark linear model with `expendop` and explanatory variables listed in `explvars4`
- Predict expenditures for the test data and summarize the fit for the benchmark model using the sum of absolute prediction errors.
- Compare the predictions of the models graphically.

### Hint

### Pre-exercise code

```
# Pre-exercise code
meps <- read.csv("CSVData\\HealthMeps.csv", header = TRUE)
#meps <- read.csv("https://assets.datacamp.com/production/repositories/2610/datasets/7b7dab6d0c528e4cd2...")
meps$logexpend <- log(meps$expendop)
# Split the sample into a `training` and `test` data
n <- nrow(meps)
set.seed(12347)
shuffled_meps <- meps[sample(n), ]
train_indices <- 1:round(0.75 * n)
train_meps <- shuffled_meps[train_indices, ]
test_indices <- (round(0.25 * n) + 1):n
test_meps <- shuffled_meps[test_indices, ]
explvars3 <- c("gender", "age", "race", "mpoor", "anylimit", "income", "insure", "usc")
explvars4 <- c(explvars3, "region", "educ", "phstat", "unemploy", "managedcare")
```

### Sample\_code

```
`@sample_code`
# Regress `logexpend` on the explanatory variables listed in `explvars3`
meps_mlr3 <- lm(logexpend ~ gender + age + phstat + anylimit + insure + mpoor, data = train_meps)

# Predict expenditures (not logged) and summarize using the sum of absolute prediction errors.
explvars3 <- c("gender", "age", "phstat", "anylimit", "insure", "mpoor")
predict_meps3 <- test_meps[,explvars3]
predict_mlr3 <- exp(predict(meps_mlr3, predict_meps3))
predict_err_mlr3 <- test_meps$expendop - predict_mlr3
sape3 <- sum(abs(predict_err_mlr3))/1000

# Regress `expendop` on all 13 explanatory variables
meps_mlr4 <- lm(____ ~ gender + age + race + region + educ + phstat + mpoor + anylimit + income + insure + income)

# Predict expenditures and summarize using the sum of absolute prediction errors.
predict_meps4 <- test_meps[,explvars4]
predict_mlr4 <- predict(meps_mlr4, predict_meps4)
predict_err_mlr4 <- test_meps$expendop - predict_mlr4
sape4 <- sum(abs(predict_err_mlr4))/1000
sape3;sape4

# Compare the predictions of the models graphically.
par(mfrow = c(1, 2))
plot(predict_err_mlr4, predict_err_mlr3, xlab = "Benchmark Predict Error", ylab = "MLR Predict Error")
plot(predict_mlr3, test_meps$expendop, xlab = "MLR Predicts", ylab = "Held Out Expend")
```

### Solution

```
# Solution
meps_mlr3 <- lm(logexpend ~ gender + age + phstat + anylimit + insure + mpoor, data = train_meps)
predict_meps3 <- test_meps[,explvars3]
explvars3 <- c("gender", "age", "phstat", "anylimit", "insure", "mpoor")
predict_meps3 <- test_meps[,explvars3]
predict_mlr3 <- exp(predict(meps_mlr3, predict_meps3))
predict_err_mlr3 <- test_meps$expendop - predict_mlr3
sape3 <- sum(abs(predict_err_mlr3))/1000

meps_mlr4 <- lm(expendop ~ gender + age + race + region + educ + phstat + mpoor + anylimit + income + income)
```



```

predict_meps4 <- test_meps[,explvars4]
predict_mlr4 <- predict(meps_mlr4, predict_meps4)
predict_err_mlr4 <- test_meps$expendop - predict_mlr4
sape4 <- sum(abs(predict_err_mlr4))/1000

sape3;sape4

par(mfrow = c(1, 2))
plot(predict_err_mlr4, predict_err_mlr3, xlab = "Benchmark Predict Error", ylab = "MLR Predict Error")
plot(predict_mlr3, test_meps$expendop, xlab = "MLR Predicts", ylab = "Held Out Expend")

```

### Submission correctness test (SCT)

success\_msg("Excellent! We found that the model of log expenditures outperforms the benchmark that models expenditures, even when the out of sample criterion was in the original 'dollar' units. It is comoforting to know that a search for a good model does well when using different out of sample criteria.")

## 5.2 What the modeling procedure tells us

### 5.2.1 Video (Exercise). What the modeling procedure tells us

#### Learning Objectives

In this module, you learn how to:

- Interpret individual effects, based on their substantive and statistical significance
- Describe other purposes of regression modeling, including regression function for pricing, benchmarking studies, and predicting future observations.

#### Video Overheads

##### Overhead A. Interpreting individual effects

- Substantive Effect
  - Does a 1 unit change in  $x$  imply an economically meaningful change in  $y$ ?
  - Example: Looking at urban and rural claims experience, is there a big enough difference to warrant differentiating prices by location?
- Statistical Significance
  - We have standards for deciding whether or not a variable is statistically significant.
  - A “statistically significant effect” is the result of a regression coefficient that is large relative to its standard error.
- Statistical significance is driven by
  - precision of  $s$ ,
  - collinearity ( $VIF$ ) and
  - sample size
- Causal Effects
  - If we change  $x$ , would  $y$  change?

##### Overhead B. Other Interpretations

- Regression function and pricing
  - The regression function is  $E y = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k$ .
  - Think about expected claims as our baseline price for short-term insurance coverages.
- Benchmarking studies

- In studies of CEO’s salaries, who is making a lot (or a little), controlled for industry, years of experience and so forth?
- In studies of medical claims, who are the high-cost patients?
- Prediction
  - A new patient comes in with a given set of characteristics, what can I say about his or her future medical claims?

### 5.2.2 MC Exercise. What the modeling procedure tells us

Which of the following are not important when interpreting the effects of individual variables?

- A. Substantive significance
- B. Statistical significance
- C. The amount of effort that it took to gather the data and do the analysis
- D. Role of causality

Which of the following is not a potential explanation for the lack of statistical significance of an explanatory variable?

- A. Large variation of the disturbance term
- B. High collinearity, so that the variable may be confounded with other variables.
- C. The coefficient of determination,  $R^2$ , is not sufficiently large.

Which of the following is not an important purpose of regression modeling?

- A. Pricing of risks such as insurance contracts
- B. Benchmarking studies, to compare an observation to others
- C. Prediction
- D. Keeping a computer occupied with work

## 5.3 The importance of variable selection

### 5.3.1 Video (Exercise). The importance of variable selection

#### Learning Objectives

In this module, you learn how to:

- Describe the bias that can occur when omitting important variables
- Describe the principle of parsimony and reasons for adopting this approach

#### Video Overheads

##### Overhead A. The importance of variable selection

- With too many or too few variables,  $s$  is too large an estimate of  $\sigma$ .
  - Prediction intervals are too large
  - Standard errors for the partial slopes are too large
- With too few or incorrect variables, we produce biased estimates of the slopes  $\beta$ . Thus, our predictions are biased and hence inaccurate.

##### Overhead B. Example. Regression using one explanatory variable

- **Too Many Variables**
  - The “true” model is  $y_i = \beta_0 + \varepsilon_i$

- We mistakenly use  $y_i = \beta_0 + \beta_1 x_i^* + \varepsilon_i$
- The prediction at a generic level  $x$  is  $b_0^* + b_1^* x$ .
- It is not too hard to confirm that  $Bias = E(b_0^* + b_1^* x) - Ey = 0$ .
- **Too Few Variables**
  - The “true” model is  $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$ .
  - We mistakenly use  $y_i = \beta_0^* \varepsilon_i$ .
  - Under the true model,  $\bar{y} = \beta_0 + \beta_1 \bar{x} + \bar{\varepsilon}$
  - Thus, the bias is

$$Bias = E \bar{y} - E (\beta_0 + \beta_1 x + \varepsilon) = E (\beta_0 + \beta_1 \bar{x} + \bar{\varepsilon}) - (\beta_0 + \beta_1 x) = \beta_1 (\bar{x} - x).$$

There is a persistent, long-term error in omitting the explanatory variable  $x$ .

#### Overhead C. Principle of parsimony

- The principle of parsimony, also known as Occam’s Razor, states that when there are several possible explanations for a phenomenon, use the simplest.
  - A simpler explanation is easier to interpret.
  - Simpler models, also known as “more parsimonious” models, often do well on fitting out-of-sample data
  - Extraneous variables can cause problems of collinearity, leading to difficulty in interpreting individual coefficients.
- In contrast, in a quote often attributed to Albert Einstein, we should use “the simplest model possible, but no simpler.”
  - Omitting important variables can lead to biased results, a potentially serious error.
  - Including extraneous variables decreases the degrees of freedom and increases the estimate of variability, typically of less concern in actuarial applications.

#### 5.3.2 MC Exercise. Principle of parsimony

Which of the following is true about under- and over-fitting a model?

- A. When we over-fit a model, estimates of regression coefficients are over-biased as is  $s^2$ , the estimate of model variance  $\sigma^2$ .
- B. When we over-fit a model, estimates of regression coefficients are remain unbiased whereas  $s^2$ , the estimate of model variance  $\sigma^2$ , is over-biased.
- C. When we over-fit a model, estimates of regression coefficients are remain under-biased as is  $s^2$ , the estimate of model variance  $\sigma^2$ .
- D. When we under-fit a model, estimates of regression coefficients are remain unbiased whereas  $s^2$ , the estimate of model variance  $\sigma^2$ , is over-biased.

Which of the following is not true of Occam’s Razor?

- A. When there are several possible explanations for a phenomenon, use the simplest one.
- B. Simpler models are easier to interpret.
- C. Variables can be statistically significant but practically unimportant.
- D. Simpler models often do better for predicting out-of-sample data

#### Overhead D. Course wrap up

Thanks for watching