REVIEW OF BASIC STATISTICS

OBJECTIVES OF THIS CHAPTER

This chapter reviews descriptive statistics, simple linear regression, multiple linear regression, and the chi-square test using R. When introducing each type of inferential statistics, an example of research design is provided followed by the research questions, the R command, and the output. The R commands are explained and the output is interpreted in detail. In addition, a sample of reporting the results for each analysis is provided. Finally, the commands for creating publication-quality tables using R are introduced, and the guidelines for reporting results are discussed. This chapter focuses on conducting basic statistical analyses using R, as well as on interpreting and presenting the results. After reading this chapter, you should be able to:

- Conduct analysis of descriptive statistics for continuous and categorical variables.
- Conduct simple linear regression and multiple regression.
- Conduct the chi-square test.
- Interpret R output for these analyses.
- Make publication-quality tables using R.
- Write results for research reports.

This section reviews the basic statistics covered in most introductory statistics courses. It shows you how to use R to perform basic statistics analyses in descriptive statistics, simple linear regression, multiple regression, and the chi-square test.

2.1 UNDERSTAND YOUR DATA USING DESCRIPTIVE STATISTICS

Descriptive statistics should never be overlooked. Whenever you conduct statistical analysis, you first need to understand your data. The descriptive statistics analysis helps you describe your data by showing you types of data, graphs of the distribution of variables, and various statistical indices, such as the central tendency and variability of your data. For continuous variables, you can draw graphs, such as histograms, box plots, and stem-and-leaf plots to show frequency distributions of your data. You can also create a scatterplot to check the relationship between two variables.

The central tendency of a variable tells you the central values of a distribution. Common measures of central tendency include the mean, median, and mode. The mean of a variable is the arithmetic average of scores, the median is the middle score when all values of a variable are ordered, and the mode is the score that is most frequently occurring. In addition to the central tendency of a distribution, we also need to understand its variability. Measures of variability show the spread of a variable. They include the range, variance, and standard deviation. The range is simply the difference between the highest and lowest scores. Variance is the average summed square of each score from the mean. To compute the variance of a set of scores, we need to follow three steps. First, subtract each score from the mean. Second, square each deviation score from the first step and sum them together to get the total sum of squares. Third, get the average sum of squares by dividing the total sum of squares by the number of scores. The standard deviation is just the square root of the variance.

2.2 DESCRIPTIVE STATISTICS FOR CONTINUOUS VARIABLES USING R

2.2.1 The `summary()` Function

The `summary()` function can be used for descriptive statistics analysis. It provides the basic descriptive statistics, such as the minimum, the first quartile, the mean, the median, the third quartile, and the maximum values for a continuous variable and frequencies for a categorical variable. Before running a descriptive analysis, you may want to know whether the missing values are properly coded, and how they are coded. Although you may browse the data, you may not be able to identify all types of missing values. A better way is to use the `summary()` function.

```
> # Import GSS 2016 Stata data file
> library(foreign)
> chp2 <- read.dta("C:/CDA/gss2016-chap1.dta")
> attach(chp2)

> summary(chp2)
      age              degree              educ            health
Min.   :18.00    high school   :940    Min.   : 0.00    excellent:414
1st Qu.:35.00    bachelor      :358    1st Qu.:12.00    good     :914
Median :50.00    graduate      :220    Median :14.00    fair     :427
Mean   :49.95    lt high school:216    Mean   :13.79    poor     :118
3rd Qu.:63.00    junior college:139    3rd Qu.:16.00    dk       :  0
Max.   :89.00    dk            :  0    Max.   :20.00    iap      :  0
                 (Other)       :  0                     na       :  0
         marital         race          realinc          realrinc
married      :819    white:1374    Min.   :   234    Min.   :   234
widowed      :170    black: 315    1st Qu.: 11115    1st Qu.:  8775
divorced     :325    other: 184    Median : 25740    Median : 17550
separated    : 65    iap  :   0    Mean   : 32484    Mean   : 24140
never married:494                  3rd Qu.: 38610    3rd Qu.: 31590
na           :  0                  Max.   :131677    Max.   :164382
                                   NA's   :168       NA's   :798
     sex
male  : 831
female:1042
```

The output shows the minimum, the first quartile, the mean, the median, the third quartile, and the maximum values for a continuous variable and frequencies for a categorical variable. For example, let us look at the first variable `age`, which is the age of the respondent. It is numeric with a range from 18 to 89.

We can also use the `head(chp2)` command to see the first six cases of all the variables in the dataset.

```
> head(chp2)
  age      degree  educ     health  marital   race   realinc  realrinc     sex
1  47    bachelor    16       good  married  white  131676.7    164382    male
2  72    bachelor    16       good  married  white   38610.0        NA    male
3  43  high school    12       good  married  white  131676.7      5265  female
4  55     graduate    18  excellent  married  white  131676.7       936  female
5  50  high school    14       poor  married  white  131676.7    164382    male
6  23  high school    11       good  married  other   15210.0      7605  female
```

The `summary(chp2)` command provides descriptive statistics for all the variables in the dataset named `chp2`. If you want to get descriptive statistics for a particular variable, you just enter the variable name within the parentheses of the function. Let us

take a look at the summary of the variable `age`, which is the age of a respondent to the General Social Survey 2016 (GSS 2016) and run an analysis of descriptive statistics with the `summary(age)` command.

```
> summary(age)
 Min.  1st Qu.  Median   Mean  3rd Qu.   Max.
18.00   35.00   50.00  49.95   63.00  89.00
```

The R output displays the minimum, the first quartile, the mean, the median, the third quartile, and the maximum values of the variable. The mean age is 49.95 years. The minimum age is 18 years old, and the maximum is 89 years old.

We can also use the `str(age)` command to look at the structure of the variable `age`. The output shows that the class of the variable `age` is integer and there are 1,873 observations.

```
> str(age)
 int [1:1873] 47 72 43 55 50 23 45 71 86 33 ...
```

We can use the `mean()` function, `sd()` function, `max()` function, `min()` function, and `length()` function to obtain the mean, standard deviation, maximum, minimum, and number of observations of the variable `age`, respectively. The `na.rm=TRUE` argument is used to delete the missing values.

```
> mean(age, na.rm=TRUE)
[1] 49.95462
> sd(age, na.rm=TRUE)
[1] 17.44434
> max(age, na.rm=TRUE)
[1] 89
> min(age, na.rm=TRUE)
[1] 18
> length(age)
[1] 1873
```

2.2.2 The `tapply()` Function for Grouped Summaries

In R, if you are interested in descriptive statistics of a variable by another grouping variable, we can use the `tapply()` function to split the data file and apply a function. In this example, we would like to see the descriptive statistics for male and female respondents, respectively. We first use the `mean <- tapply(age, sex, mean, na.rm=TRUE)` command to obtain the mean for the variable `age` for each category

of sex. In the tapply() function, age is the continuous variable in which we are interested, sex is the grouping variable with two categories, and mean is the function. The na.rm=TRUE argument is used to remove the missing values. The computed means for the two groups are named the object mean. Then we repeat the process to obtain the standard deviations (sd), maximums (max), minimums (min), and numbers of observations (length) of age for males and female separately. Finally, we use the cbind(mean, sd, max, min, length) command to combine the results.

```
> # Descriptive statistics by group: method 1
> mean <- tapply(age, sex, mean, na.rm=TRUE)
> sd <- tapply(age, sex, sd, na.rm=TRUE)
> max <- tapply(age, sex, max, na.rm=TRUE)
> min <- tapply(age, sex, min, na.rm=TRUE)
> length <- tapply(age, sex, length)
> cbind(mean, sd, max, min, length)
            mean        sd  max  min  length
male    48.62214  17.07903   89   18     831
female  51.01727  17.66650   89   18    1042
```

The R output shows the descriptive statistics of age for males and female separately. Of 831 male respondents, the mean of the age is 48.622 years and the standard deviation is 17.079 years. Of 1,042 female respondents, the average age is 51.017 years and the standard deviation is 17.667 years.

2.2.3 The group_by() Function and the summarize() Function for Grouped Summaries

Another option is to use the group_by() function and the summarize() function in the dplyr package (Wickham et al., 2021). If not installed, you need to install the package first by typing install.packages("dplyr") and then load it by using the library(dplyr) function. In the following example, we first use the gender <- group_by(chp2, sex) function to create a grouping variable and name it as an object gender. In the group_by() function, chp2 is the data frame and sex is the grouping variable. We then use the summarize() function to compute the descriptive statistics for age for the two categories of the variable sex. The following is the output.

```
> # Descriptive statistics by group: method 2
> library(dplyr)
> gender <- group_by(chp2, sex)
> summarize(gender, mean(age, na.rm=TRUE), sd(age, na.rm=TRUE), max(age, na.rm=TRUE), min(age,
na.rm=TRUE))
# A tibble: 2 x 5
   sex    `mean(age, na.rm ~  `sd(age, na.rm =~  `max(age, na.rm~  `min(age, na.rm~
   <fct>               <dbl>              <dbl>             <int>             <int>
1  male                 48.6               17.1                89                18
2  female               51.0               17.7                89                18
```

2.3.4 The group_by() Function and the descr() Function for Grouped Summaries

The third option is to use the group_by() function in the dplyr package and the descr() function in the sjmisc package (Lüdecke, 2018a). Since both the dplyr and sjmisc packages have been loaded, we do not need to load them again. This single-line chp2 %>% group_by(sex) %>% descr(age) command tells R to run it as a chain. We first take the data frame chp2, then split the data by the grouping variable sex, and finally compute the descriptive statistics of age for the two categories of the variable sex. In this command, the pipe operator %>% means "then," which helps us to run a series of functions in sequence.

```
> # Descriptive statistics by group: method 3
> # library(sjmisc)
> # library(dplyr)
> chp2 %>% group_by(sex) %>% descr(age)

## Basic descriptive statistics

Grouped by: male

var     type  label    n  NA.prc   mean     sd    se  md  trimmed     range  skew
age  Integer    age  831       0  48.62  17.08  0.59  48    48.09 71 (18-89)  0.18

Grouped by: female

var     type  label     n  NA.prc   mean     sd    se  md  trimmed     range  skew
age  integer    age  1042       0  51.02  17.67  0.55  52    50.62 71 (18-89)  0.12
```

2.2.5 Descriptive Statistics for Multiple Variables With stat.desc()

The previous examples focus on the descriptive statistics for one continuous variable. If you would like to see the descriptive statistics for more than one variable, simply use the stat.desc() function in the pastecs package (Grosjean et al., 2018) and list the variables in the c() function. You need to install the package first by typing install.packages("pastecs") and then load the package with the library(pastecs) command. In the following example, we use the stat.desc(chp2[, c("age", "educ")]) command to compute the descriptive statistics for the two variables age and educ. The following results are displayed.

```
> # Descriptive statistics for two variables
> # Install pastecs by using install.packages()
> library(pastecs)
> stat.desc(chp2[, c("age", "educ")])
                      age          educ
nbr.val      1.873000e+03  1.873000e+03
nbr.null     0.000000e+00  2.000000e+00
nbr.na       0.000000e+00  0.000000e+00
min          1.800000e+01  0.000000e+00
max          8.900000e+01  2.000000e+01
range        7.100000e+01  2.000000e+01
sum          9.356500e+04  2.582100e+04
median       5.000000e+01  1.400000e+01
mean         4.995462e+01  1.378590e+01
SE.mean       4.030749e-01   6.953498e-02
CI.mean.0.95  7.905233e-01   1.363742e-01
var          3.043051e+02  9.056169e+00
std.dev      1.744434e+01  3.009347e+00
coef.var      3.492038e-01   2.182916e-01
```

The output shows that the means of `age` and `educ` are 49.955 and 13.786, respectively. The variable `educ` is the highest year of school completed. The smallest value is 0, whereas the largest value is 20. The variance is 9.056, and the standard deviation of 3.009.

2.2.6 Descriptive Statistics for Multiple Variables With `descr()`

Another option is to use the `descr()` function in the `sjmisc` package. You need to install the package first by typing `install.packages("sjmisc")` and then load it with the `library(sjmisc)` command. In the `descr(chp2, age, educ)` command, `chp2` is the data frame and `age` and `educ` are the two selected variables in the data.

```
> # Descriptive statistics for two variables: method 2
> library(sjmisc)
> descr(chp2, age, educ)

## Basic descriptive statistics

 var      type   label      n   NA.prc    mean      sd     se   md   trimmed       range   skew
 age   integer     age   1873        0   49.95   17.44   0.40   50     49.48  71 (18-89)   0.15
educ   integer    educ   1873        0   13.79    3.01   0.07   14     13.80  20 (0-20)  -0.20
```

The output shows the variable name, type, label, total number, percentage of missing data, mean, standard deviation, standard error, mode, range, and skewness. For example, the mean age is 49.95 years and the standard deviation is 17.44 years. The range is 71 years. The skewness is .15, which is less than 1. It indicates that the distribution is almost symmetric. We can use the Shapiro–Wilk test with the `shapiro.test()` function to test normality.

2.2.7 The `group_by()` Function and the `descr()` Function for Grouped Summaries of Multiple Variables

If we would like to see the mean of these two continuous variables across the categories of a third variable, we can use the `chp2 %>% group_by(sex) %>% select(age, educ) %>% descr()` command. The following example displays each mean of `age` and `educ` for males and females separately.

```
# Descriptive statistics for multiple variables by group
> chp2 %>% group_by(sex) %>% select(age, educ) %>% descr()

## Basic descriptive statistics

Grouped by: male

 var     type  label    n  NA.prc   mean     sd    se  md  trimmed       range
 age  integer    age  831       0  48.62  17.08  0.59  48    48.09  71 (18-89)
educ  integer   educ  831       0  13.74   2.99  0.10  14    13.76   20 (0-20)
  skew
  0.18
 -0.29

Grouped by: female

 var     type  label     n  NA.prc   mean     sd    se  md  trimmed       range
 age  integer    age  1042       0  51.02  17.67  0.55  52    50.62  71 (18-89)
educ  integer   educ  1042       0  13.82   3.03  0.09  14    13.83   20 (0-20)
  skew
  0.12
 -0.13
```

The R output shows the descriptive statistics of `age` and `educ` for males and female separately. In the first table, the mean, standard deviation, minimum, and maximum of

the variable age for males are 48.62, 17.08, 18, and 89, respectively. In addition, the mean, standard deviation, minimum, and maximum of the variable educ for males are 13.74, 2.99, 0, and 20, respectively. The descriptive statistics of age and educ for female are displayed in the second table.

2.3 FREQUENCY DISTRIBUTION FOR CATEGORICAL VARIABLES USING R

For categorical variables, such as gender or ethnicity, it does not make much sense to get summary statistics, such as the mean and the standard deviation. Instead we normally do a frequency analysis to get the frequency of each value for categorical variables. The table() function is used for creating frequency tables. The following example shows you how to create a frequency table for the nominal variable degree from the GSS 2016 dataset. Before we start, let us take a look at the structure of this variable with the str(degree) command first.

```
> str(degree)
 Factor w/ 8 levels "lt high school",..: 4 4 2 5 2 2 2 3 2 2 ...
```

The output shows that degree is a factor variable with eight levels. To see detailed levels and frequency for each level, we use the table() function below.

2.3.1 The table() Function for a Single Categorical Variable

Next, let us look at the frequency table of the variable degree using the table(degree) command. We can also use the table(marital) command for the other categorical variable, marital.

```
> # Frequency table for a factor or categorical variable
> table(degree)
degree
lt high school  high school  junior college  bachelor  graduate
           216          940              139       358       220
            dk          iap               na
             0            0                0
> table(marital)
marital
      married  widowed  divorced  separated never married
          819      170       325         65           494
           na
            0
```

The `table(degree)` command displays the frequency for each level or category of the variable `degree`. The output shows eight categories. Since "iap" (i.e., inapplicable), "dk" (i.e., don't know), and "na" (i.e., no answer) represent the missing values in the Stata dataset for this chapter, there are five actual levels ranging from "lt high school" to "graduate."

2.3.2 The `frq()` Function in the `sjmisc` Package for a Single Categorical Variable

To see more detailed information such as percentages and cumulative percentages in a frequency table, we use the `frq()` function in the `sjmisc` package. The package needs to be loaded first. In the `frq(chp2, degree)` command, `chp2` is the data frame and `degree` is the categorical variable. We also use the same `frq()` function for `marital`.

```
# Frequency table with frq() in sjmisc
> # library(sjmisc)
> frq(chp2, degree)


degree <categorical>
# total N=1873 valid N=1873 mean=2.69 sd=1.24

Value           |    N  |  Raw %  |  Valid %  |  Cum. %
------------------------------------------------------------
lt high school  |  216  |  11.53  |   11.53   |   11.53
high school     |  940  |  50.19  |   50.19   |   61.72
junior college  |  139  |   7.42  |    7.42   |   69.14
bachelor        |  358  |  19.11  |   19.11   |   88.25
graduate        |  220  |  11.75  |   11.75   |  100.00
dk              |    0  |   0.00  |    0.00   |  100.00
iap             |    0  |   0.00  |    0.00   |  100.00
na              |    0  |   0.00  |    0.00   |  100.00
<NA>            |    0  |   0.00  |    <NA>   |    <NA>



> frq(chp2, marital)

marital <categorical>
# total N=1873 valid N=1873 mean=2.60 sd=1.66

Value          |    N  |  Raw %  |  Valid %  |  Cum. %
-----------------------------------------------------------
married        |  819  |  43.73  |   43.73   |   43.73
widowed        |  170  |   9.08  |    9.08   |   52.80
divorced       |  325  |  17.35  |   17.35   |   70.15
separated      |   65  |   3.47  |    3.47   |   73.63
never married  |  494  |  26.37  |   26.37   |  100.00
na             |    0  |   0.00  |    0.00   |  100.00
<NA>           |    0  |   0.00  |    <NA>   |    <NA>
```

Interpreting R Output

The output shows the number of observations, valid observations with no missing values, mean, and standard deviation on the top of the frequency table. In the first table, the first column shows the levels of the degree, the labels of the categories. The frequency column (labeled "`frq`") shows the number of respondents who reported their highest degrees. The `Raw %` column shows that 11.53% of respondents have a degree less than high school, 50.19% have a high-school degree, 7.42% have a junior college degree, 19.11% have a bachelor's degree, and 11.75% have a graduate degree. The percentages from the missing values which are coded as "iap," "dk," "na," and "NA" are 0%. The `Valid %` column shows the percentages for data with no missing values. The last column labeled `Cum. %` provides the cumulative percentage for each category. For example, the cumulative percent of having a junior college degree or less is 69.14%, which equals the percent of respondents having less than a high-school degree (11.53%) plus the percent having a high-school degree (50.19%), plus the percent having a junior college degree (7.42%). The other frequency table for the variable `marital` can be interpreted in the same way.

2.3.3 The `table()` Function for a Two-Way Table

If we want to get a two-way cross-tabulation table for two categorical variables, we use the `table()` function with the two variables which are separated by a comma within the parentheses. For example, the command `tab <- table(degree, race)` tells R to create a two-way table of frequency counts for two nominal variables `degree` and `race` and assign `tab` as the object name. The `ftable(tab)` command also produces the same frequency table. To get the marginal totals of the rows and columns of the table, we use the `addmargins (tab)` command.

```
> # Cross-tabulation
> tab <- table(degree, race)
> summary(tab)
Number of cases in table: 1873
Number of factors: 2
Test for independence of all factors:
    Chisq = NaN, df = 21, p-value = NA
    Chi-squared approximation may be incorrect
> tab
                race
degree             white  black  other  iap
  lt high school     132     40     44    0
  high school        683    186     71    0
  junior college     104     23     12    0
  bachelor           286     44     28    0
  graduate           169     22     29    0
  dk                   0      0      0    0
  iap                  0      0      0    0
  na                   0      0      0    0
```

```
> ftable(tab)
                race
degree           white  black  other  iap
  lt high school   132     40     44    0
  high school      683    186     71    0
  junior college   104     23     12    0
  bachelor         286     44     28    0
  graduate         169     22     29    0
  dk                 0      0      0    0
  iap                0      0      0    0
  na                 0      0      0    0

> addmargins(tab)
                race
degree           white  black  other  iap   Sum
  lt high school   132     40     44    0   216
  high school      683    186     71    0   940
  junior college   104     23     12    0   139
  bachelor         286     44     28    0   358
  graduate         169     22     29    0   220
  dk                 0      0      0    0     0
  iap                0      0      0    0     0
  na                 0      0      0    0     0
  Sum             1374    315    184    0  1873
```

Interpreting R Output

The output produced by the `table(degree, race)` command or the `ftable(tab)` command displays a two-way cross-tabulation table, where the first column lists the categories for the variable `degree` and the first row lists the categories for the variable `race`. The variable `degree` is the row variable since its categories are across the rows of the table. The variable `race` is the column variable since its categories are across the top of the table. Each cell shows a relative frequency of subjects in each subgroup. For example, there were 176 White respondents who did not have a high-school degree.

The table produced by the `addmargins(tab)` command displays the marginal totals of the rows and columns. The last column is the row total. It shows the total frequency for each of the five degree levels. The last row is the column total. It provides the total frequency for each of the three categories of the variable `race`. The row and column totals are also called marginal totals or frequencies.

2.3.4 The `CrossTable()` Function in the `gmodels` Package

To get the relative frequency of each cell within its row and column, we use the `CrossTable()` function in the `gmodels` package (Warnes et al., 2018). We need to install the package first by typing `install.packages("gmodels")` and then load it by using the `library(gmodels)` command. In the following example, we use the `CrossTable(degree, race, digits=2)` command to create a frequency table. In the function, `degree` and `race` are the two categorical variables and the argument `digits=2` specifies the number of decimals.

```
> # Cross-tabulation with CrossTable() in gmodels
> library(gmodels)
> CrossTable(degree, race, digits=2)

  Cell Contents
|  ------------------------  |
|                         N  |
|   Chi-square contribution  |
|              N / Row Total  |
|              N / Col Total  |
|            N / Table Total  |
|  ------------------------  |


Total Observations in Table:  1873
```

degree	race: white	black	other	Row Total
lt high school	132	40	44	216
	4.42	0.37	24.46	
	0.61	0.19	0.20	0.12
	0.10	0.13	0.24	
	0.07	0.02	0.02	
high school	683	186	71	940
	0.06	4.93	4.93	
	0.73	0.20	0.08	0.50
	0.50	0.59	0.39	
	0.36	0.10	0.04	
junior college	104	23	12	139
	0.04	0.01	0.20	
	0.75	0.17	0.09	0.07
	0.08	0.07	0.07	
	0.06	0.01	0.01	
bachelor	286	44	28	358
	2.08	4.36	1.46	
	0.80	0.12	0.08	0.19
	0.21	0.14	0.15	
	0.15	0.02	0.01	
graduate	169	22	29	220
	0.36	6.08	2.53	
	0.77	0.10	0.13	0.12
	0.12	0.07	0.16	
	0.09	0.01	0.02	
Column Total	1374	315	184	1873
	0.73	0.17	0.10	

```
> chisq.test(degree, race)

    Pearson's Chi-squared test

data:  degree and race
X-squared = 56.286, df = 8, p-value = 2.481e-09
```

Interpreting R Output

In the output, each category of the first variable `degree` takes one row, and each category of the second variable `race` takes one column. The row percentage displays the relative frequency of a cell within each row (i.e., each degree level), and the column percentage displays the relative frequency of a cell with each category of race. For example, 683 White respondents had a high-school degree, which was 73% of the 940 respondents who had a high-school degree (683/940 = 73%). Its column percentage was 50%. This means 50% of the 1,374 White respondents had a high-school degree (683/1,374 = 50%).

2.4 SIMPLE LINEAR REGRESSION

2.4.1 Simple Linear Regression: An Introduction

Regression is used when we predict a dependent variable from an independent variable or multiple independent variables. When there is only one independent variable, the regression is simple linear regression; when there are two or more independent variables, it is called multiple linear regression. The independent variable is the explanatory or predictor variable, whereas the dependent variable is the outcome or response variable, the one we are interested in predicting from the predictor variable. In linear regression, the dependent variable is continuous and the independent variable(s) can be either continuous or categorical. The simple linear regression can be expressed as follows:

$$Y = \beta_0 + \beta_1 X_1 + e \tag{2.1}$$

where Y is the observed value of the dependent variable, X_1 is the value of the independent variable, β_0 is the intercept, β_1 is the regression coefficient that is the slope of the regression line, and e is the error term, also known as the residual. The error term e is the difference between the observed value and the predicted value of the dependent variable. It is assumed to have a normal distribution with a mean of 0 and a constant variance at every value of the independent variable. The values of e are mutually independent from each other.

The simple linear regression can also be expressed as a sample equation, which is the predicted regression equation:

$$\hat{Y} = b_0 + b_1 X_1 \tag{2.2}$$

where $\hat{Y}$ is the predicted or fitted value of the dependent variable, b_0 is the intercept, and b_1 is the regression coefficient. The estimated intercept and coefficient can also be expressed as $\hat{\beta}_0$ and $\hat{\beta}_1$, respectively.

To estimate the intercept β_0 and the regression coefficient β_1, the method of ordinary least squares (OLS) is used, which minimizes the sum of squared residuals. Since the residual e is the difference between the observed value Y and predicted value $\hat{Y}$, of the

dependent variable, the estimated intercept and regression coefficients are the values when the total amount of the squared errors is as small as possible.

In the following example, we are interested in how well annual family income can be predicted by a respondent's individual income using the GSS 2016 dataset. The dependent variable is annual family income, and the independent variable is the respondent's annual income.

Research Questions:

1. Can an individual's income predict the family income?
2. How well can the annual family income be predicted from an individual's income?

2.4.2 The `lm()` Function and Extractor Functions

The `lm()` function is used for linear regression analysis. The model formula in `lm()` specifies the dependent variable and the predictor variable(s), which are separated by the tilde (~). For example, in simple linear regression, there is only one dependent variable and one independent variable, so the model syntax in the function is `y ~ x`, where `y` is the dependent variable and `x` is the independent variable. The command `lm(y ~ x, data = data1)` tells R to run a simple regression analysis predicting the dependent variable `y` with an independent variable `x`. The data argument is `data = data1`. When there is more than one predictor variable in the formula, they are connected by plus (+) symbols. For more details on how to use this command, type `help(lm)` in the command prompt.

Once a model is fitted, we can use extractor functions to extract specific results. For example, we can use the `summary()` function to display the summary results of the fitted model, use the `coef()` function to extract the coefficients, use the `confint()` function to request the confidence intervals of the coefficients, and use the `anova()` function to request the ANOVA table. Other useful extractor functions include the `fitted()` function for creating the fitted values, the `residuals()` function for the residual values, the `predict()` function for the predicted values of an outcome variable at specific values of a predictor variable or multiple predictor variables, the `AIC()` function for the AIC statistic, and the `plot()` function for diagnostic plots. These are the generic functions and most of them can be applied to other models introduced in the book.

In the following example, the command `slm <-lm(realinc1 ~ realrinc1)` tells R to predict the dependent variable `realinc1` from the independent variable `realrinc1`. The fitted model is named `slm`. The output is shown by the `summary (slm)` command.

```
> # Simple linear regression
> realinc1 <- realinc/10000
> realrinc1 <- realrinc/10000
> slm <- lm(realinc1 ~ realrinc1)
> summary(slm)


Call:
lm(formula = realinc1 ~ realrinc1)


Residuals:
    Min      1Q  Median      3Q     Max
-1.8654 -1.4220 -0.7374  0.5563 11.2236


Coefficients:
             Estimate  Std. Error  t value   Pr(>|t|)
(Intercept)   1.87035     0.08842    21.15  <2e-16 ***
realrinc1     0.78711     0.02284    34.46  <2e-16 ***


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


Residual standard error: 2.211 on 1045 degrees of freedom
 (826 observations deleted due to missingness)
Multiple R-squared: 0.5319, Adjusted R-squared: 0.5315
F-statistic: 1188 on 1 and 1045 DF, p-value: < 2.2e-16
```

2.4.3 Interpreting R Output: The Coefficients Table

Two new variables `realinc1` and `realrinc1` are created so that the family income and the respondent's income are on a scale of $10,000. With this transformation, the R output would show the results properly. In the R output for the regression analysis, the first part is the call, which shows the R command for the model. The second part shows the residuals. The minimum, first quarter, median, third quarter, and maximum values of the residuals are shown here. The third part shows the coefficients table including the parameter estimates for the predictor variable and the intercept, their standard errors, the t statistics, and the associated p values. The fourth part shows the residual standard error and the degrees of freedom. The fifth part shows the multiple R^2 and the adjusted multiple R^2. Finally, the sixth part shows the F statistic, the degrees of freedom, and the associate p value.

First, let us take a look at the regression table. In this model, we have one independent variable, or the predictor, and the constant. The null hypothesis for the t test is that the coefficient of the predictor variable is zero, and the alternative hypothesis is that the coefficient of the predictor variable is significantly different from zero. In this example, $\beta_1 = .787$ is the regression coefficient. The t statistic tests whether the regression coefficient of the independent variable, respondent's income, is significantly different from zero. In other words, it tests whether the effect of the independent variable on the dependent variable is significant.

The t statistic equals the parameter estimate divided by its standard error. For the predictor variable `realrinc1`, $t = .787/.023 = 34.46$. The associated p value, `Pr(>|t|) < .001`, which means that the probability of having a t value larger than the absolute value of the observed t value of 34.46 is less than .001 if the null hypothesis is true, so we reject the null hypothesis. The rejection of the null hypothesis indicates that the regression coefficient of the independent variable, respondent's income, is significantly different from zero. Therefore, the predictor variable, respondent's income (`realrinc1`) is a significant predictor of the dependent variable, family income (`realinc1`).

In the model, the estimated constant (β_0) is shown as `(Intercept)`, = 1.870. This is the mean family income when the value of the independent variable equals zero.

The estimated regression coefficient $\beta_1 = .787$. It can be interpreted as follows: For a one-unit increase in the respondent's annual income, the dependent variable, the family income, is increased by a value of .787.

Substituting the values of the constant and regression coefficient into the equation of the estimated regression model:

$$\hat{Y} = b_0 + b_1 X_1$$

We get:

$$\hat{Y} = 1.870 + .787 X_1$$

When $X_1 = 0$, the predicted outcome variable = 1.870, which is the constant.

Next, the output reports the residual standard error and the degrees of freedom. The residual standard error = 2.211, which is the square root of the mean squares of variance for the `Residual`. The degrees of freedom for the residual = $n - 1 - k$ = 1,047 − 1 − 1 = 1,045 since n = 1,047 and the number of predictors $k = 1$.

2.4.4 Interpreting R Output: The Multiple R^2 and the F Statistic

Further, the multiple R^2 and the adjusted multiple R^2 are reported. R^2 is called the coefficient of determination, indicating the strength of the prediction. It is the proportion of error variance explained by the predictor. $R^2 = .532$, which indicates that 53.2% of the variance in the family income is explained by the respondent's income. The adjusted R^2 takes the sample size and the number of predictors into consideration, and it is a less biased estimate of the population R^2. Adjusted $R^2 = .352$.

Finally, the output shows the F statistic, the degrees of freedom, and the associate p value. The F statistic tests whether the overall model with one predictor in this example can significantly predict the dependent variable.

Null hypothesis: The overall model with one predictor in this example can significantly predict the dependent variable.

Alternative hypothesis: The overall model with one predictor in this example cannot significantly predict the dependent variable.

$F(1, 1{,}045) = 1187.50$, $p < .001$, which indicates that the model with one predictor, respondent's income, is significantly different from zero.

2.4.5 The ANOVA Table

We use the `anova()` function to request the ANOVA table, which was discussed in the one-way ANOVA section.

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

Response: realinc1
             Df   Sum Sq  Mean Sq  F value        Pr(>F)
realrinc1     1  5805.0   5805.0   1187.5  < 2.2e-16 ***
Residuals  1045  5108.4      4.9

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

The source of variance includes three components: the variance for the model, the variance for the residuals, and the total variance. The total sum of squares of the variance (SST) is partitioned into the model sum of squares (SSM) and the residual sum of squares (SSR). Since this model has only one independent variable or predictor variable, the SSM is the sum squares of the predictor variable `realrinc1`. The Analysis of Variance table displays the variance for `realrinc1` and the residuals, so we can compute the SST as follows.

$$\text{SST} = \text{SSM} + \text{SSR} = 5,805 + 5,108.4 = 10,913.4$$

The `Df` column lists the degrees of freedom for their respective variances. The total degrees of freedom $= n - 1 = 1{,}047 - 1 = 1{,}046$. The degrees of freedom for the model equal the number of predictors $k = 1$. The degrees of freedom for the residual $= n - 1 - k = 1{,}047 - 2 = 1{,}045$.

The `Mean Sq` column lists the mean squares of the variance, which is the ratio of the sum squares and the corresponding degrees of freedom. For example, MS (model) = 5,805.0/1 = 5,805.0.

The *F* statistic equals the ratio of the mean squares of variance for the model (i.e., `real rinc1`) to the mean squares of variance for the `Residuals`. F = MSM/MSR = 5,805.0/4.9 = 1,187.50. $F(1, 1{,}045) = 1{,}187.50$.

2.4.6 The `coef()` Function and the `confint()` Function

The regression coefficient and the constant can also be obtained using the `coef(slm)` command and their confident intervals can be obtained with `confint(slm)`.

```
> coef(slm)
(Intercept)  realrinc1
  1.8703542  0.7871115
> confint(slm)
                 2.5 %     97.5 %
(Intercept)  1.6968568  2.0438516
realrinc1    0.7422919  0.8319312
```

2.4.7 Effect Size With the `eta_sq()` Function

We can also compute the effect size using use the `eta_sq()` function in the `sjstats` package (Lüdecke, 2021). You need to install the package first by typing `install.packages("sjstats")` and then load it with `library(sjstats)`. In the following example, we use the `eta_sq(slm)` command to compute the eta square (η^2).

```
> # Install sjstats using install.packages()
> library(sjstats)
> eta_sq(slm)
     term etasq
1 realrinc1 0.532
```

In the output, the overall model $\eta^2 = .532$, which is the same as R^2 in the output of the `summary(slm)` command above.

2.4.8 Reporting the Results

A simple regression analysis was conducted to investigate whether an individual's annual income was a predictor of the family income and how accurate the prediction was: $F(1, 1{,}045) = 1{,}187.50$, $p < .001$, which indicated that the model with one predictor, respondent's income, was significantly different from zero. The regression coefficient $\beta = .787$, $p < .001$, which indicates that there was a significant effect of respondent's income on the dependent variable, family income. For a one-unit increase in the respondent's annual income, the family income was increased by a factor of .787.

2.5 MULTIPLE LINEAR REGRESSION

2.5.1 Multiple Linear Regression: An Introduction

Multiple linear regression is simply an extension of the simple linear regression when there are two or more independent variables. It is used to predict a continuous dependent variable from a combination of predictors that can be either continuous or categorical variables. Similar to simple regression, multiple linear regression can be expressed as follows:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k + e$$

where Y is the continuous dependent variable; X_1, X_2, …, and X_k are a set of independent variables; β_0 is the intercept; and β_1, β_2, and β_k are the regression coefficients for predictors. As with simple linear regression, the intercept and regression coefficients are estimated using the OLS method.

In the following example, we are interested in how well the family income can be predicted by a combination of three independent variables using the GSS 2016 dataset. The dependent variable is the annual family income, and the independent variables are the respondent's annual income, the highest years of education completed, and age.

Research Question: How accurately can the family income be predicted from a set of three independent variables, the respondent's annual income, the highest years of education completed, and age?

2.5.2 The `lm()` Function

As with simple regression, the function for multiple regression is still the same `lm()` function. The model formula in `lm()` specifies the dependent variable and the predictor variable(s), which are separated by the tilde (~). When there are multiple predictor variables, they are connected by plus (+) symbols. For more details on how to use this command, type `help(lm)` in the command prompt.

In the following example, the command `mlm <- lm(realinc1 ~ realrinc1 + educ + age)` tells R to predict the dependent variable `realinc1` from the three independent variables `realrinc1`, `educ`, and `age`. The fitted model is named `mlm`. The output is shown by the `summary(mlm)` command.

```
# Multiple linear regression
> mlm <- lm(realinc1 ~ realrinc1 + educ + age, data=chp2)
> summary(mlm)


Call:
lm(formula = realinc1 ~ realrinc1 + educ + age, data = chp2)


Residuals:
    Min       1Q   Median       3Q      Max
-3.3887  -1.2862  -0.6401  0.5646  11.6320
```

```
Coefficients:
              Estimate  Std. Error  t value      Pr(>|t|)
(Intercept)  -1.219610   0.394154   -3.094   0.002026 **
realrinc1     0.733981   0.023196   31.642    < 2e-16 ***
educ          0.168181   0.023660    7.108  2.17e-12 ***
age           0.018453   0.004787    3.855  0.000123 ***


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


Residual standard error: 2.147 on 1043 degrees of freedom
  (826 observations deleted due to missingness)
Multiple R-squared: 0.5594, Adjusted R-squared: 0.5581
F-statistic: 441.4 on 3 and 1043 DF, p-value: < 2.2e-16
```

2.5.3 Interpreting R Output: The Coefficients Table

In the R output for the regression analysis, the first part is the call, which shows the R command for the model. The second part shows the residuals. The minimum, first quarter, median, third quarter, and maximum values of the residuals are shown here. The third part shows the coefficients table including the parameter estimates for the predictor variable and the intercept, their standard errors, the t statistics, and the associated p values. The fourth part shows the residual standard error and the degrees of freedom. The fifth part shows the multiple R^2 and the adjusted multiple R^2. Finally, the sixth part shows the F statistic, the degrees of freedom, and the associated p value.

First, in the regression model table, we can see the estimated regression coefficients, their standard errors, the t values, and the associated p values for the constant and the three predictor variables. The 95% confident intervals can be obtained with the `confint(mlm)` command, which will be introduced later in this section.

The t statistic in the regression table tests whether each regression coefficient of the three independent variables is significantly different from zero, controlling for the other independent variables. It is the ratio of the estimated regression coefficient to its standard error.

The regression coefficient for the first predictor, respondent's annual income (`realrinc1`), $\beta = .734$. The t value equals the ratio of the regression coefficient to its standard error: $t = .7340/.0232 = 31.642$.

Under the heading, `P(>|t|)`, $p <$ `2e-16`. This means that the probability of having a t value larger than the absolute value of 31.64 is close to zero if the null hypothesis is true. Since $p < .001$, we can conclude that the regression coefficient of the independent variable, respondent's income, is significantly different from zero when holding the other two predictors constant. Therefore, the respondent's income is a significant predictor of the dependent variable, family income.

The regression coefficient $\beta_1 = .734$. This can be interpreted as follows: For a one-unit increase in the respondent's annual income, the dependent variable, the family income, is increased by a factor of .734 when holding the other two predictors constant.

The regression coefficient for the second predictor, the highest years of education completed (`educ`), $\beta_2 = .168$, $t = .1682/.0237 = 7.10$.

Under the heading, `P(>|t|)`, $p <$ `2e-16`. This means that the probability of having a t value larger than the absolute value of 7.10 is almost zero if the null hypothesis is true. Since $p < .001$, we can conclude that the regression coefficient of the independent variable, the highest years of education completed, is significantly different from zero when holding the other two predictors constant. Therefore, the predictor, the highest years of education completed, is a significant predictor of the dependent variable, family income.

The regression coefficient $\beta_2 = .168$, which means that for a one-unit increase in the highest years of education completed, the dependent variable, the family income, is increased by a factor of .168 when holding the other two predictors constant.

The regression coefficient for the third predictor age (`age`), $\beta_3 = .018$. $t = .0185/.0048 = 3.854$.

Under the heading, `P(>|t|)`, $p < .001$. We can conclude that the regression coefficient of the independent variable, age, is significantly different from zero when controlling for the other two predictors. Therefore, the predictor, age, is a significant predictor of the dependent variable, family income.

In the model, the constant (β_0), shown as `(Intercept)`, $= -1.220$. It is also known as the intercept of the model and is the mean family income when the values of the independent variables equal zero.

Substituting the values of the constant and regression coefficients into the equation of the estimated regression model:

$$\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$$

We get:

$$\hat{Y} = -1.220 + 0.734 X_1 + 0.168 X_2 + 0.018 X_3$$

Next, the output reports the residual standard error and the degrees of freedom. The residual standard error = 2.147, which is the square root of the mean squares for the residual (i.e., mean square error). The degrees of freedom for the residual $= n - 1 - k = 1{,}047 - 1 - 3 = 1{,}043$ since $n = 1{,}047$ and the number of predictors $k = 3$.

2.5.4 Interpreting R Output: The Multiple R^2 and the F Statistic

Further, the multiple R^2 and the adjusted multiple R^2 are reported. R^2 is the coefficient of multiple determination, which tells us the strength of the prediction with a set of

predictors. It is the proportion of error variance explained by the model with all the predictors. $R^2 = .559$, which indicates that 55.9% of the variance in the family income is explained using a fitted model with the three predictors overall. The adjusted R^2 takes the sample size and the number of predictors into consideration, and it is a less biased estimate of the population R^2. Adjusted $R^2 = .558$.

Finally, the output shows the F statistic, the degrees of freedom, and the associate p value. The F statistic in the multiple regression tests whether the overall model with all the predictors can significantly predict the dependent variable.

Null hypothesis: The regression coefficients of all three independent variables are equal to zero ($\beta_1 = \beta_2 = \beta_3 = 0$).

In other words, all three independent variables are not significant predictors of the dependent variable, family income.

Alternative hypothesis: At least one of the three regression coefficients of the independent variables is different from zero, controlling for the others. (At least one β_j is not equal to 0.)

In other words, at least one of the three independent variables is a significant predictor of the dependent variable, family income.

$F(3, 1{,}043) = 441.4$, $p < .001$, which indicates that the overall model with three predictors, respondent's income, the highest years of education completed, and age, is significant. In other words, at least one independent variable significantly predicts the dependent variable.

2.5.5 The `coef()` Function and the `confint()` Function

The regression coefficient and the constant can also be obtained using the `coef(mlm)` command and their confident intervals can be obtained with the `confint(mlm)` command.

```
> coef(mlm)
(Intercept)  realrinc1       educ        age
-1.2196105   0.7339815  0.1681815  0.0184535

> confint(mlm)
                    2.5 %       97.5 %
(Intercept)  -1.993036447  -0.44618454
realrinc1     0.688464467   0.77949850
educ          0.121755538   0.21460741
age           0.009059685   0.02784731
```

A 95% confidence interval for a regression coefficient means that we are 95% confident that this interval contains the true coefficient with repeated samples. The 95% confidence interval for the predictor `realrinc1` is [.688, .779]. It can be interpreted as follows: For a one-unit change in the respondent's annual income, we are 95% confident that the change in the dependent variable, the family income, is between .668 and .779 when controlling for the other two variables.

The 95% confidence interval for the predictor `educ` is [.122, .215]. This means that for a one-unit change in the highest years of education completed, we are 95% confident that the change in the dependent variable, the family income, is between .122 and .215 when holding the other two predictors constant.

The 95% confidence interval for the predictor `age` can be interpreted in a similar way.

2.5.6 The ANOVA Table

We use the `anova()` function to request the ANOVA table, which was discussed in the one-way ANOVA section. The command `anova(mlm)` requests the ANOVA table for the multiple regression model.

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

Response: realinc1
             Df   Sum Sq  Mean Sq   F value     Pr(>F)
realrinc1     1  5805.0   5805.0  1259.157  < 2.2e-16 ***
educ          1   231.4    231.4    50.188  2.567e-12 ***
age           1    68.5     68.5    14.859   0.000123 ***
Residuals  1043  4808.5      4.6
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Just as explained for simple regression, in the ANOVA table, the `Sum Sq` column displays the sum of squares for each predictor variable and the sum of squares for the residuals (also called the sum of squared errors). The total sum of squares can be obtained by adding all the sum of squares. The `Df` column displays the respective degrees of freedom related to each type of sum of squares. The `Mean Sq` column shows the mean squares of the variance, which is the ratio of the sum squares to the corresponding degrees of freedom. The `F value` column shows the *F* statistics. Finally, the `Pr(>F)` column displays the associated *p* values.

2.5.7 Effect Size With the `eta_sq()` Function

We again use the `eta_sq()` function in the `sjstats` package to compute the effect size. Since we have loaded the package in the previous section for simple linear

regression, you do not need to load it again. We can also use the `anova_stats()` function in the `sjstats` package to compute more effect size statistics. In the following example, we use the `eta_sq(mlm, ci.lvl = .95)` command to compute the eta square (η^2). In the `eta_sq()` function, `mlm` is the model object and the `ci.lvl = .95` argument requests the 95% confidence interval.

```
> eta_sq(mlm, ci.lvl = .95)
        term  etasq  conf.low  conf.high
1  realrinc1  0.532     0.466      0.598
2       educ  0.021     0.010      0.033
3        age  0.006     0.001      0.012
```

The η^2 for respondent's annual income (`realrinc1`) is .532, which indicates that 53.2% of the variance in the family income is explained by this predictor variable. The eta squares (η^2) for the other two predictor variables `educ` and `age` can be interpreted in the same way. The same results can be obtained with the `effectsize::eta_squared(mlm, partial = FALSE)` command.

We can also use the `eta_sq(mlm, partial=TRUE, ci.lvl = .95)` command to compute the partial η^2. The output is as follows.

```
> eta_sq(mlm, partial=TRUE, ci.lvl = .95)
        term  partial.etasq  conf.low  conf.high
1  realrinc1          0.547     0.510      0.580
2       educ          0.046     0.024      0.073
3        age          0.014     0.003      0.031
```

In the output, the partial η^2 for respondent's annual income (`realreinc1`) is .547, which indicates that 54.7% of the variance in the family income is explained by this predictor variable while excluding variance explained by the other two predictor variables. The partial eta squares (η^2) for the other two predictor variables `educ` and `age` can be interpreted in the same way.

2.5.8 Computing the Predicted Values With the `ggpredict()` Function in `ggeffects`

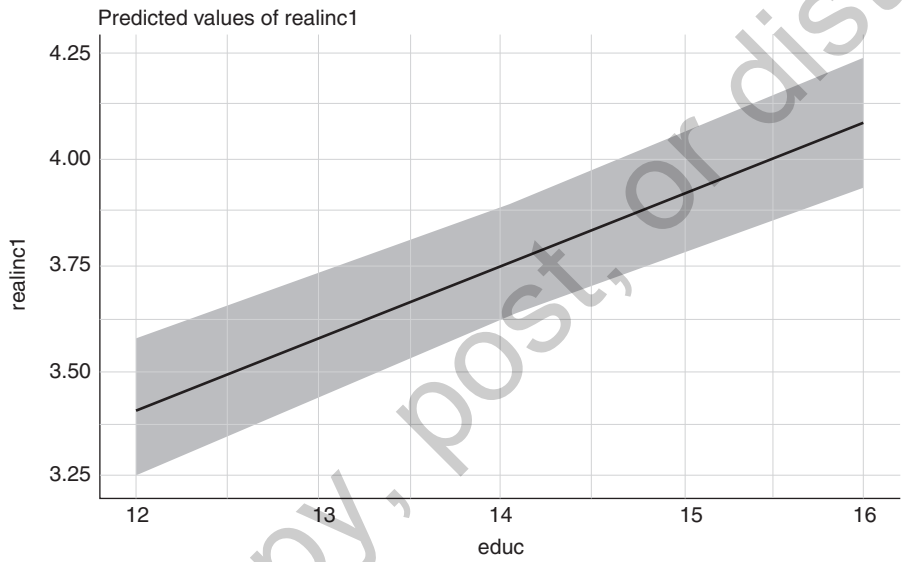
The `ggpredict()` function in the `ggeffects` package (Lüdecke, 2018b) is used to compute the predicted outcome variable if we know the specified values of a predictor variable or multiple predictor variables. This command is particularly useful for generalized linear models that will be introduced in the following chapters. The `ggeffects` package needs to be installed first by typing `install.packages("ggeffects")`. We then load it with `library(ggeffects)`. In the first

example, we compute the predicted values of the outcome variable when the predictor variable `educ` is specified at the values of 12, 14, and 16 and the other two predictors are held at their means. Please note when predictor variables are non-numeric or categorical, they are held at the reference level or category. The command is as follows: `mlm.educ <- ggpredict(mlm, terms="educ[12, 14, 16]")`. In the `ggpredict()` function, `mlm` is the model object and the `terms="educ[12, 14, 16]"` option specifies the predictor variable `educ` at the values of 12, 14, and 16. When there are more than one variable, the `terms` option can specify up to four variables, including the second to fourth grouping variables. In this example, we include only one variable in the `terms` option. The output is assigned to an object named `mlm.educ`. Either the `as.data.frame()` function or the `sqrt(diag(vcov()))` function can be used to request the standard errors of the predicted values.

```
> # Predicted values with ggpredict() in ggeffects
> # Install ggeffects using install.packages()
> library(ggeffects)
> mlm.educ <-ggpredict(mlm, terms="educ[12, 14, 16]")
> mlm.educ

# Predicted values of realinc1

educ  |  Predicted  |        95% CI
------------------------------------
  12  |       3.41  |  [3.25, 3.58]
  14  |       3.75  |  [3.62, 3.88]
  16  |       4.09  |  [3.93, 4.24]



Adjusted for:
* realrinc1 =  2.46
*       age = 44.00
> as.data.frame(mlm.educ)
    x  predicted   std.error  conf.low  conf.high  group
1  12   3.413709  0.08459629  3.247904   3.579515      1
2  14   3.750072  0.06672711  3.619290   3.880855      1
3  16   4.086435  0.07890954  3.931775   4.241095      1

> sqrt(diag(vcov(mlm.educ)))
          1           2           3
0.08459629  0.06672711  0.07890954

> plot(mlm.educ)
Loading required namespace: ggplot2
```

In the output, when `educ` equals 8, 13, and 16 and the two predicator variables `realreinc1` and `age` are held at their means, 2.46, and 44.00, respectively, the mean predicted outcomes are 3.41, 3.75, and 4.09, respectively. The same values can be obtained if we substitute the specified values of `educ` and the means of the other two predictor variables into the multiple regression equation provided earlier. Figure 2.1 shows the predicted values of the outcome variable for `educ` at 12, 14, and 16 when the other two predictor variables are held at their means.

FIGURE 2.1 ⬢ Predicted Values of the Outcome Variable for `educ` at 12, 14, and 16 With Others Fixed at Their Means



In the second example, we compute the predicted values of the outcome variable when the three predictor variables are specified at their mean values and at one standard deviation below and above the mean values with the `meansd` option. The command is as follows: `mlm.m <- ggpredict(mlm, terms=c("educ[meansd]", "realrinc1[meansd]", "age[meansd]"))`. In the `ggpredict()` function, `mlm` is the model object and the `terms= c("educ[meansd]", "realrinc1[meansd]", "age[meansd]")` option specifies the mean value and at one standard deviation below and above the mean value for each predictor variable. In this example, we use the `c()` function to include the three variables in the `terms` option. The output is assigned to an object named `mlm.m`.

```
> mlm.m <- ggpredict(mlm, terms=c("educ[meansd]", "realrinc1[meansd]", "age
[meansd]"))
> mlm.m

# Predicted values of realinc1
# realrinc1 = -0.54
#       age = 31.1

 educ  |  Predicted  |        95% CI
-------------------------------------
11.30  |       0.86  |  [0.62, 1.10]
14.20  |       1.35  |  [1.12, 1.57]
17.10  |       1.83  |  [1.55, 2.11]

# realrinc1 = 2.46
#       age = 31.1

 educ  |  Predicted  |        95% CI
-------------------------------------
11.30  |       3.06  |  [2.83, 3.29]
14.20  |       3.55  |  [3.36, 3.73]
17.10  |       4.04  |  [3.81, 4.26]

# realrinc1 = 5.45
#       age = 31.1

 educ  |  Predicted  |        95% CI
-------------------------------------
11.30  |       5.25  |  [4.97, 5.54]
14.20  |       5.74  |  [5.51, 5.98]
17.10  |       6.23  |  [5.98, 6.48]

# realrinc1 = -0.54
#       age = 45

 educ  |  Predicted  |        95% CI
-------------------------------------
11.30  |       1.11  |  [0.91, 1.32]
14.20  |       1.60  |  [1.41, 1.79]
17.10  |       2.09  |  [1.84, 2.34]

# realrinc1 = 2.46
#       age = 45

 educ  |  Predicted  |        95% CI
-------------------------------------
11.30  |       3.32  |  [3.13, 3.50]
14.20  |       3.80  |  [3.67, 3.93]
17.10  |       4.29  |  [4.11, 4.48]
```

```
# realrinc1 = 5.45
#       age = 45

 educ   |  Predicted  |        95% CI
------------------------------------------------
11.30   |       5.51  |  [5.26, 5.76]
14.20   |       6.00  |  [5.81, 6.19]
17.10   |       6.49  |  [6.28, 6.69]


# realrinc1 = -0.54
#       age = 59

 educ   |  Predicted  |        95% CI
------------------------------------------------
11.30   |       1.37  |  [1.12, 1.62]
14.20   |       1.86  |  [1.63, 2.10]
17.10   |       2.35  |  [2.06, 2.64]


# realrinc1 = 2.46
#       age = 59

 educ   |  Predicted  |        95% CI
------------------------------------------------
11.30   |       3.58  |  [3.35, 3.80]
14.20   |       4.06  |  [3.88, 4.25]
17.10   |       4.55  |  [4.32, 4.78]


# realrinc1 = 5.45
#       age = 59

 educ   |  Predicted  |        95% CI
------------------------------------------------
11.30   |       5.77  |  [5.49, 6.05]
14.20   |       6.26  |  [6.03, 6.48]
17.10   |       6.75  |  [6.50, 6.99]
```

Since each predictor variable is specified at the three values, the mean value and at one standard deviation below and above the mean value, the output displays nine tables for the combinations of the three variables. The mean for `educ` is 14.20 and the values for one standard deviation below and above the mean value are 11.30 and 17.10, respectively. Similarly, the mean for `realrinc1` is 2.46 and the values for one standard deviation below and above the mean value are −.54 and 5.45, respectively;

the mean for `age` is 45.00 and the values for one standard deviation below and above the mean value are 31.10 and 59.00, respectively.

Let us take a look at the first table in the output. When `educ` = 11.30, `realrinc1` = −.54, and `age` = 31.10, the estimated margin or the mean predicted outcome ($\hat{Y}$) is .86. When `educ` = 14.20, `realrinc1` = −.54, and `age` = 31.10, the estimated margin or the mean predicted outcome ($\hat{Y}$) is 1.35. When `educ` = 17.10, `realrinc1` = −.54, and `age` = 31.10, the estimated margin or the mean predicted outcome ($\hat{Y}$) is 1.83.

When all three predictor variables are held at their means (i.e., `educ` = 14.20, `realrinc1` = 2.46, and `age` = 45.00), the mean predicted outcome ($\hat{Y}$) is 3.80. The same result is displayed in the fifth table in the output.

2.5.9 Reporting the Results

A multiple regression analysis was conducted to predict the annual family income using three predictors, the respondent's annual income, the highest years of education completed, and age. $F(3, 1{,}043) = 441.4$, $p < .001$, which indicated that the overall model with the three predictors was significant. $R^2 = .559$, which indicated that 55.9% of the variance in the family income was explained by the fitted model with the three predictors overall. Adjusted $R^2 = .558$.

The respondent's income was a significant predictor of the dependent variable, family income ($\beta = .734$, $t = 31.642$, $p < .001$). For a one-unit increase in the respondent's annual income, the dependent variable, the family income, was increased by a factor of .734 when holding the other two predictors constant.

The predictor, the highest years of education completed, was also a significant predictor of the dependent variable, family income ($\beta = .168$, $t = 7.108$, $p < .001$). For a one-unit increase in the highest years of education completed, the dependent variable, the family income, was increased by a factor of .168 when holding the other two predictors constant.

In addition, age was a significant predictor of the dependent variable, family income ($\beta = .018$, $t = 3.855$, $p < .001$). For a one-unit increase in age, the family income was increased by a factor of .018 when holding the other two predictors constant.

2.6 CHI-SQUARE TEST

2.6.1 The Chi-Square Test: An Introduction

The chi-square test of independence is used to investigate the relationship between two categorical variables. Each categorical variable has two or more levels/categories. A two-way contingency table can be constructed with one variable as the row variable and the other as the column variable. The rows list different levels of the row variable, and the columns represent categories of the column variable. For example, a 4×5 contingency table shows frequencies for a row variable with four levels and a column variable with five levels.

A simple case for the chi-square test is a two-by-two frequency table, which includes two categorical variables with two levels for each category. In it the rows represent two categories of one variable and the columns represent two categories of the other. Each cell where a row and column intersects tells the frequency number of participants that fall into each subgroup.

In the following example, we are interested in whether two categorical variables, health status and marital status, are related using the GSS 2016 dataset. One variable, health status, has four levels, including poor, fair, good, and excellent. The other variable, marital status, has five levels: married, widowed, divorced, separated, and never married.

Research Question: Is there a relationship between the two categorical variables, health status and marital status? Or, in other words, are health status and marital status independent of each other?

Null hypothesis: There is a relationship between the two categorical variables, health status and marital status.

Alternative hypothesis: There is no relationship between the two categorical variables, health status and marital status.

2.6.2 The `CrossTable()` Function in the `gmodels` Package

To get the relative frequency of each cell within its row and column, we use the `CrossTable()` function in the `gmodels` package. Since the package has been installed earlier in this chapter, we just need to load it with `library(gmodels)`. If it has been loaded, you do not need to load it again. In the following example, we use the `CrossTable(health, marital, digits=2)` command to create a frequency table. In the function, `health` and `marital` are the two categorical variables and the argument `digits=2` specifies the number of decimals.

```
> # Cross-tabulation with CrossTable() in gmodels
> library(gmodels)
> CrossTable(health, marital, digits=2)
   Cell Contents
|  ---------------------------  |
|                           N  |
|   Chi-square contribution    |
|               N / Row Total  |
|               N / Col Total  |
|             N / Table Total  |
|  ---------------------------  |
Total Observations in Table:  1873
```

health	marital: married	widowed	divorced	separated	never married	Row Total
excellent	196	35	60	7	116	414
	1.24	0.18	1.95	3.78	0.42	
	0.47	0.08	0.14	0.02	0.28	0.22
	0.24	0.21	0.18	0.11	0.23	
	0.10	0.02	0.03	0.00	0.06	
good	430	76	162	32	214	914
	2.30	0.58	0.07	0.00	3.04	
	0.47	0.08	0.18	0.04	0.23	0.49
	0.53	0.45	0.50	0.49	0.43	
	0.23	0.04	0.09	0.02	0.11	
fair	160	46	65	21	135	427
	3.82	1.35	1.12	2.58	4.45	
	0.37	0.11	0.15	0.05	0.32	0.23
	0.20	0.27	0.20	0.32	0.27	
	0.09	0.02	0.03	0.01	0.07	
poor	33	13	38	5	29	118
	6.70	0.49	15.00	0.20	0.14	
	0.28	0.11	0.32	0.04	0.25	0.06
	0.04	0.08	0.12	0.08	0.06	
	0.02	0.01	0.02	0.00	0.02	
Column Total	819	170	325	65	494	1873
	0.44	0.09	0.17	0.03	0.26	

Interpreting R Output

The output provides a two-way contingency table. The row variable, health status, has four levels, and the column variable, marital status, has five levels. The last row shows the column totals (819, 170, 325, 65, and 494), and the last column displays the row totals (414, 914, 427, and 118). The row and column totals are also called the marginal totals according to the places where they are located. The total number of observations is shown at the lower right bottom of the table (n = 1,873). The row percentage and column percentage are also displayed in the table.

2.6.3 The `chisq.test()` Function

We use the `chisq.test()` function to conduct the chi-square test for the two categorical variables `health` and `marital`. The command is `chisq.test(health, marital)` and the resulting output is displayed as follows.

```
> chisq.test(health, marital)

        Pearson's Chi-squared test

data:  health and marital
X-squared = 49.423, df = 12, p-value = 1.764e-06


Warning message:
In chisq.test(health, marital) : Chi-squared approximation may be incorrect
```

The Pearson $\chi^2 = 49.423$ with the degrees of freedom of 12. The number of degrees of freedom in the χ^2 test = $(r - 1) \times (c - 1) = 3 \times 4 = 12$ since r is the number of rows and c is the number of columns.

$\chi^2_{(12)} = 49.423$, $p < .001$, which indicates that there is a significant relationship between health status and marital status.

2.6.4 Cramér's *V*

There are several ways to compute the Cramér's *V*. We first use the `xtab_statistics()` function in the `sjstats` package. The `sjstats` package needs to be installed first by typing `install.packages("sjstats")`. We then load it with `library(sjstats)`.

Before we use the `xtab_statistics()` function, we need to work on the data frame to correctly code the missing values as "NA." The current data frame includes missing values coded as "iap" (i.e., inapplicable), "dk" (i.e., don't know), and "na" (i.e., no answer). These three types of missing values need to be coded as "NA" in R. We first load the data frame `chp2.bs`, then we select the two factor variables `health` and `marital` by using the `dplyr::select(health, marital)` command.

In the command, `dplyr::select` means that we use the `select ()` function from the `dplyr` package with `::`, the double colons. Finally, we recoded the three types of missing values to "NA" by using the `mutate ()` function in the `dplyr` package and the `rec ()` function in the `sjmisc` packages. Since both the `dplyr` and `sjmisc` packages have been loaded in the previous section, we do not need to load them again. You need to load them if you have not done so. The command is as follows: `dplyr::mutate(health_re = rec(health, rec = "iap=NA; dk=NA; na=NA; else=copy"), marital_re = rec(marital, rec = "na=NA; else=copy"))`. The recoded two variables are named `health_re` and `marital_re`, respectively. The pipe operator `%>%` is used to run a series of functions in sequence. The new data frame is named as the object `new`.

```
> # Recode user-defined missing values to NA
> new<-chp2 %>%
+  dplyr::select(health, marital) %>%
+  dplyr::mutate(health_re = rec(health, rec = "iap=NA; dk=NA; na=NA; else=copy"),
marital_re = rec(marital, rec = "na=NA; else=copy") )


One or more of the old values are recoded into identical new values.
Please check if you correctly specified the recode-pattern,
else separate multiple values with comma, e.g. rec="a,b,c=1; d,e,f=2".
```

To check if the two variables are correctly recoded, we use the `table (new$health_re, new$marital_re)` command to create a two-way cross-tabulation table.

```
> table(new$health_re, new$marital_re)


           divorced  married  never married  separated  widowed
excellent       60      196            116          7       35
fair            65      160            135         21       46
good           162      430            214         32       76
poor            38       33             29          5       13
```

Since "iap," "dk," and "na" are not shown in the cross-tabulation table, these missing values are coded correctly.

We use the `xtab_statistics()` function in the `sjstats` package to compute the Cramér's *V*. In the `xtab_statistics(new, health_re, marital_re)` command, `new` is the data frame and `health_re` and `marital_re` are the two recoded factor variables.

```
# Compute chi-square test statistic and Cramer's V
> library(sjstats)
> xtab_statistics(new, health_re, marital_re)


# Measure of Association for Contingency Tables
      (using Fisher's Exact Test)


 Chi-squared: 49.4229
  Cramer's V : 0.0938
     p-value : <0.001
```

In the output, Cramér's $V = .094$. It also shows that the Pearson $\chi^2 = 49.423$ and $p < .001$.

The second method to compute the Cramér's V is to use the `cramer()` function in the `sjstats` package. In the `cramer()` function, `health_re ~ marital_re` is the model equation and `data=new` is the data argument. The command and the resulting output are as follows.

```
> cramer(health_re ~ marital_re, data=new)
[1] 0.09378524
```

The third method to compute the Cramér's V is to use the `assocstats()` function in the `vcd` package (Meyer et al., 2020). The `vcd` package needs to be installed first by typing `install.packages("vcd")`. We then load it with `library(vcd)`. We first create a two-way table by using the `tab <- table(new$health_re, new$marital_re)` command and the table is assigned to an object named `tab`. Then we use the `summary(assocstats(tab))` command to display the statistics of association between the two factor variables.

```
> # Install vcd by using install.packages()
> library(vcd)
Loading required package: grid
> tab <- table(new$health_re, new$marital_re)
> summary(assocstats(tab))

Number of cases in table: 1873
Number of factors: 2
Test for independence of all factors:
  Chisq = 49.42, df = 12, p-value = 1.764e-06
  Chi-squared approximation may be incorrect
                 X^2 df  P(> X^2)
Likelihood Ratio 47.973 12 3.1603e-06
Pearson          49.423 12 1.7639e-06


Phi-Coefficient      :  NA
Contingency Coeff.   :  0.16
Cramer's V           :  0.094
```

The output displays the likelihood ratio χ^2 test statistic, the Pearson χ^2 test statistic, the contingency coefficient, and the Cramér's V. The results are the same as those above.

We can also compute the Cramér's V manually by following the equation

$$\text{Cramér's } V = \sqrt{\frac{\chi^2}{N*(k-1)}}$$

where χ^2 is the Pearson chi-square statistic, N is the sample size, and k is the smaller number of categories of the two variables. In this example, $\chi^2 = 49.423$ and $N = 1{,}873$. Since the numbers of categories for `health_re` and `marital_re` are 4 and 5, respectively, the smaller number of categories is 4. $k - 1 = 3$. We compute the Cramér's V as follows.

```
> # Compute the Cramer's V
> sqrt(49.423/1873/3)
[1] 0.09378536
```

Cramér's V indicates the strength of association between two categorical variables. Its size lies between 0 and 1. In the output, Cramér's $V = .094$, which indicates that the relationship between two variables is weak.

In the output, we can easily see that each cell displays the row and column percentages in addition to the frequency. The row percentage equals the number of counts in a cell divided by its corresponding row total at the margin (i.e., the marginal total).

2.6.5 Follow-Up Chi-Square Test With the `chisq.test()` Function

The Pearson χ^2 test discussed earlier indicated that there is a significant relationship between health status and marital status. It is an omnibus test for the overall model. If we are interested in the relationship between the variables for subcategories, then we can conduct follow-up tests. For example, we can examine the health status between married and widowed. The research question would be as follows: Do the married and the widowed differ among the levels of health status?

Since we know the frequency distribution from the earlier output, we can use the `matrix()` function to create a 4 by 2 matrix with the command `matrix(c(196, 35, 160, 46, 430, 76, 33, 13), nrow=4, byrow=T)` and the matrix is assigned to an object named `list`. We then conduct a chi-square test with the `chisq.test(list)` command.

```
> # Create a matrix and conduct a chi-square test
> list <- matrix(c(196, 35, 160, 46, 430, 76, 33, 13), nrow=4, byrow=T)
> chisq.test(list)
```

```
  Pearson's Chi-squared test

data: list
X-squared = 10.133, df = 3, p-value = 0.01747
```

$\chi^2_{(3)} = 10.133$, $p < .05$, which indicates that there is a significant relationship between health status and marital status and that the married have better health status levels than the widowed.

Other follow-up comparisons can be done in a similar way. To control the Type I error due to multiple comparisons, we need to adjust the significant level. To do this, we can use the Bonferroni adjustment.

2.6.6 Reporting the Results

A Pearson chi-square test was conducted to investigate the relationship between two categorical variables, health status and marital status, using the GSS 2016 dataset. $\chi^2_{(12)} = 49.423$, $p < .001$, which indicated that there was a significant relationship between health status and marital status. Cramér's V was used to indicate the strength of association between two categorical variables. Cramér's $V = .094$, which indicated that the relationship between two variables was weak.

2.7 MAKING PUBLICATION-QUALITY TABLES USING R

Once you have conducted statistical analyses and interpreted results, the final step of the research process is to report the results and submit your manuscript for publication. While writing the manuscript, you may find that you need to summarize your research findings and display them in tables. You might wonder whether there are any tools to reduce your workload, or automatically combine results from the fitted models and generate a single publication-style regression table.

R has several user-written packages to accomplish this job with no hassle. We use the `stargazer` package (Hlavac, 2018) and the `texreg` package (Leifeld, 2013) to produce regression tables throughout this book since they work with most of the models covered here.

To install these two add-on packages, you can use the `install.packages()` function and then load it with `library()` function. For example, if you would like to install `stargazer`, type `install.packages("stargazer")`, choose a

mirror close to you, and then install the package. To use it, you need to load it with the `library(stargazer)` command.

The help files of these two packages provide various examples with explanations on how to use them. The following is an example of using the `stargazer()` function in the `stargazer` package to make a table for the results of two regression models. After fitting the single-predictor model `slm` and the multiple-predictor model `mlm` introduced in the previous section, we use the `stargazer(slm, mlm, type="text", align=TRUE, out="chp2.lrmod.txt")` command to create a table. In the `stargazer()` function, we first specify `slm` and `mlm`, the two model objects to be presented, and then the type of the table with the `type="text"`

```
> library(stargazer)

Please cite as:

 Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
 R package version 5.2.2. https://CRAN.R-project.org/package=stargazer

> stargazer(slm, mlm, type="text", align=TRUE, out="chp2.lrmod.txt")

================================================================
                                Dependent variable:
                    ---------------------------------------
                                       realinc1
                              (1)                  (2)
----------------------------------------------------------------
realrinc1                   0.787***             0.734***
                            (0.023)              (0.023)
educ                                             0.168***
                                                 (0.024)
age                                              0.018***
                                                 (0.005)
Constant                    1.870***             -1.220***
                            (0.088)              (0.394)
----------------------------------------------------------------
Observations                1,047                1,047
R2                          0.532                0.559
Adjusted R2                 0.531                0.558
Residual Std. Error         2.211 (df = 1045)    2.147 (df = 1043)
F Statistic                 1,187.513*** (df = 1; 1045)  441.401*** (df = 3; 1043)
================================================================
Note:                                *p<0.1; **p<0.05; ***p<0.01

> stargazer(slm, mlm, type="html", align=TRUE, out="chp2.lrmod.htm")
```

argument. The `align=TRUE` argument requests to align the results of the two models. The `out="chp2.lrmod.txt"` argument saves the output named `chp2.lrmod.txt`.

Table 2.1 shows the results of two regression models using the `stargazer()` function.

TABLE 2.1 ⬢ Results of Two Regression Models: An Example

	Dependent variable:	
	realinc1	
	(1)	(2)
realrinc1	0.787***	0.734***
	(0.023)	(0.023)
educ		0.168***
		(0.024)
age		0.018***
		(0.005)
Constant	1.870***	−1.220***
	(0.088)	(0.394)
Observations	1,047	1,047
R^2	0.532	0.559
Adjusted R^2	0.531	0.558
Residual Std. Error	2.211 (df = 1,045)	2.147 (df = 1,043)
F Statistic	1,187.513*** (df = 1; 1,045)	441.401*** (df = 3; 1,043)

Note:
*$p < 0.1$
**$p < 0.05$
***$p < 0.01$

A similar table can be created if you use the `screenreg()` and `htmlreg()` functions from the `texreg` package. You need to install `texreg` first by typing `install.packages("texreg")` if you haven't already done so and then load the package by typing `library(texreg)`. To create a plain text table similar to Table 2.1, we use the `screenreg(list(slm, mlm))` command. In the `screenreg()` function, we specify the two model objects to be presented with the `list()` function. We can also use the `htmlreg()` function to create the same regression table and save it to a Microsoft Word file named `chap2.doc` with the

following command: `htmlreg(list(slm, mlm), file="chap2.doc", doctype=TRUE, html.tag=TRUE, head.tag=TRUE)`. To save space, the created table is omitted here.

Both the `stargazer` package and the `texreg` package can be of good help when creating a regression table since you do not need to start from scratch. They are particularly useful when you fit a series of models and need to summarize parameter estimates from each model. Please note that different fields or journals have different requirements for the table format. You still need to do some editing to the created tables in the manuscript submitted for publication.

2.8 GENERAL GUIDELINES FOR REPORTING RESULTS

Once the data analysis is complete, the next step is to present and interpret the results, which normally are included in the results section of a research report. When reporting the results of your statistical analyses, a general rule is to provide sufficient information for readers to understand your analyses and the findings of your study. What are the major elements that should be included in the results section? The answer varies since different disciplines and journals may have their own reporting requirements, and one statistic commonly reported in one field may not be needed by another field. We provide the following general guidelines for reporting results.

First, describe the analyses you have conducted, explain the variables with descriptive statistics, and state what research questions have been addressed.

Second, when reporting the results of a statistical test, provide the value of the test, the degrees of freedom, and the associated p value, followed with an explanation of the meaning of your findings. You may also need to form a conclusion about whether the test is significant. If the test is significant, provide the effect size if available. The reporting of the effect sizes was recommended in APA (2020), and various measures of effect size for continuous and categorical outcome variables and their estimation methods were introduced in Kline (2013).

Third, when summarizing numerical information in tables, make sure they can be easily interpreted. The labels, categories, and numbers in the tables should be concise and clear, so that readers can understand them without much effort. Complex tables may confuse readers with excessive information.

Fourth, tables and written text should be complementary to each other. If the results can be summarized in a sentence, you do not need a table. On the other hand, if you have many categorical variables and you need report frequencies for all categories, it will be tedious to report them in paragraph after paragraph in written text, and it will be boring for readers to read your description. Instead, a table containing these frequencies is sufficient to help readers quickly understand this information.

Fifth, after the results of statistical tests are presented, you also need to interpret them. Readers are more interested in the meaning of the statistical results than in the technical information related to the tests. The numerical information is important since it is the evidence supporting your conclusion. To help readers understand these statistics, you need to interpret them in a clear manner.

Finally, when summarizing the results, also keep the audience in mind. If you submit your article to a journal for publication, reviewers are interested in reading the statistics, tables, and graphs. You might receive feedback from them asking you to provide more statistics. However, if your readers have limited backgrounds in statistics, you will need to explain your results in plain English in addition to the statistics you provide in the text.

2.9 SUMMARY OF R COMMANDS IN THIS CHAPTER

```
# Chap 2 R Script


# Remove all objects
rm(list = ls(all=TRUE))
# Set a working directory; here we create a folder named CDA in the C drive (C:) first
setwd("C:/CDA")


# The following user-written packages need to be installed first by using
install.packages(" ") and then by loading it with library()
# library(dplyr)      # It is already installed for Chapter 1
# library(sjmisc)     # It is already installed for Chapter 1
# library(gmodels)
# library(ggeffects)
# library(pastecs)
# library(stargazer)
# library(sjstats)
# library(vcd)


# Import GSS 2016 Stata data file
library(foreign)
chp2 <- read.dta("C:/CDA/gss2016-chap1.dta")
attach(chp2)
summary(chp2)
head(chp2)
str(age)
summary(age)
mean(age, na.rm=TRUE)
sd(age, na.rm=TRUE)
max(age, na.rm=TRUE)
min(age, na.rm=TRUE)
length(age)
```

```
# Descriptive statistics by group: method 1
mean <- tapply(age, sex, mean, na.rm=TRUE)
sd <- tapply(age, sex, sd, na.rm=TRUE)
max <- tapply(age, sex, max, na.rm=TRUE)
min <- tapply(age, sex, min, na.rm=TRUE)
length <- tapply(age, sex, length)
cbind(mean, sd, max, min, length)


# Descriptive statistics by group: method 2
library(dplyr)
gender <- group_by(chp2, sex)
summarize(gender, mean(age, na.rm=TRUE), sd(age, na.rm=TRUE), max(age, na.rm=TRUE),
min(age, na.rm=TRUE))


# Descriptive statistics by group: method 3
library(sjmisc)
chp2 %>% group_by(sex) %>% descr(age)


# Descriptive statistics for two variables
library(pastecs)
stat.desc(chp2[, c("age","educ")])


# Descriptive statistics for two variables: method 2
#library(sjmisc)
descr(chp2, age, educ)


# Descriptive statistics for multiple variables by group
chp2 %>% group_by(sex) %>% select(age, educ) %>% descr()


# Frequency table for a factor or categorical variable
str(degree)
table(degree)
table(marital)
# Frequency table with frq() in sjmisc
#library(sjmisc)
frq(chp2, degree)
frq(chp2, marital)


# Cross-tabulation
tab <- table(degree, race)
summary(tab)
tab
ftable(tab)
addmargins(tab)
prop.table(tab)


# Cross-tabulation with CrossTable() in gmodels
library(gmodels)
CrossTable(degree, race, digits=2)
chisq.test(degree, race)


# Simple linear regression
realinc1 <- realinc/10000
```

```
realrinc1 <- realrinc/10000
slm <- lm(realinc1 ~ realrinc1)
summary(slm)
anova(slm)
coef(slm)
confint(slm)
library(sjstats)
eta_sq(slm)


# Multiple linear regression
mlm <- lm(realinc1 ~ realrinc1 + educ + age, data=chp2)
summary(mlm)
anova(mlm)
coef(mlm)
confint(mlm)
eta_sq(mlm, ci.lvl = .95)
eta_sq(mlm, partial=TRUE, ci.lvl = .95)
anova_stats(mlm, digits = 3)


# Predicted values with ggpredict() in ggeffects
library(ggeffects)
mlm.educ <- ggpredict(mlm, terms="educ[12, 14, 16]")
mlm.educ
as.data.frame(mlm.educ)
sqrt(diag(vcov(mlm.educ)))
plot(mlm.educ)


mlm.m <- ggpredict(mlm, terms=c("educ[meansd]", "realrinc1[meansd]", "age
[meansd]"))
mlm.m


# Create a results table with stargazer()
library(stargazer)
stargazer(slm, mlm, type="text", align=TRUE, out="chp2.lrmod.txt")
stargazer(slm, mlm, type="html", align=TRUE, out="chp2.lrmod.htm")


# Cross-tabulation with CrossTable() in gmodels
library(gmodels)
CrossTable(health, marital, digits=2)
chisq.test(health, marital)


# Recode user-defined missing values to NA
new <- chp2 %>%
 dplyr::select(health, marital) %>%
 dplyr::mutate(health_re = rec(health, rec = "iap=NA; dk=NA; na=NA; else=copy"),
marital_re = rec(marital, rec = "na=NA; else=copy" ) )


table(new$health_re, new$marital_re)


# Compute chi-square test statistic and Cramer's V
library(sjstats)
xtab_statistics(new, health_re, marital_re)
cramer(health_re ~ marital_re, data=new)
```

```
library(vcd)
tab <- table(new$health_re, new$marital_re)
summary(assocstats(tab))


# Compute the Cramer's V
sqrt(49.423/1873/3)


# Create a matrix and conduct a chi-square test
list <- matrix(c(196, 35, 160, 46, 430, 76, 33, 13), nrow=4, byrow=T)
chisq.test(list)


 detach(chp2)
```

Glossary

Descriptive statistics helps you describe your data by showing types of data, graphs of the distribution of variables, and various statistical indices, such as the central tendency and variability of your data.

Frequency analysis provides the frequency of each value for categorical variables.

Multiple linear regression is used when we predict a dependent variable from two or more independent variables.

Simple linear regression is used when we predict a dependent variable from an independent variable.

The chi-square test of independence is used to investigate the relationship between two categorical variables.

Exercises

Use the GSS 2016 data available at **https://edge.sagepub.com/liu1e** for the following problems.

1. Run a descriptive statistics analysis for `coninc` and interpret the results.
2. Conduct a frequency analysis for `happy`. What percentages of respondents are very happy?
3. Make a two-way table for `degree` and `class`.
4. Run a chi-square test to investigate the relationship between `degree` and `class`.
5. Conduct a multiple regression analysis to estimate `tvhours` from the two predictor variables `educ` and `age`.
   a. Write a research question.
   b. Find the *F* statistic from the output and interpret whether the overall model is statistically significant.
   c. Which predictor variables are significant? Interpret the coefficients for the two predictor variables.
   d. Produce a table for the regression output using `stargazer`.
   e. Write a concise report to summarize the results.