Lab 3 (Part 2) - Descriptive Statistics in R

Contents

Lab Objectives

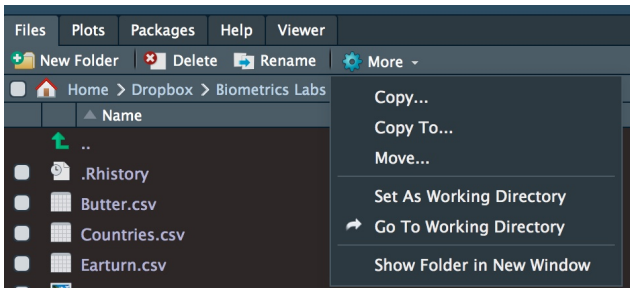
- Calculate descriptive statistics for continuous and categorical data
- Edit output tables

Set your working directory

Before getting started, set your working directory to where the course files are located on your computer. The code below is commented out, since you will have to provide your own working directory.

```
# setwd('Path to your files') # i.e.
# C:/Users/YourName/Documents/Biometrics Labs/ note: this
# depends on your computer's OS
```

In RStudio, you can also change your working directory using the graphical interface:



Exercise 1: Descriptive statistics

Since R is primarily a statistical computing language many descriptive statistics are available simply by coding. More complex approaches involving summarising large datasets can be achieved by installing certain packages.

It is always important to take a moment to think about the type of data you are using and what descriptive statistics will be most useful given the type. For numerical data, you typically report measures of central tendency and measures of variability. It is often useful to observe the frequency distributions or histograms of continuous distributions to note if they are normal or skewed. For categorical data you typically report the frequency or proportion of each value.

Descriptive Statistics:

Let's begin by calculating descriptive statistics for some of the data in the Appendix D file.

Open appendixd.csv and examine the structure of the data:

```
d <- read.csv("appendixd.csv")
str(d)
```

```
## 'data.frame':    88 obs. of  10 variables:
##  $ ID     : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ addsc  : int  45 50 49 55 39 68 69 56 58 48 ...
##  $ gender : Factor w/ 2 levels "female","male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ repeat.: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 2 1 1 1 ...
##  $ iq     : int  111 102 108 109 118 79 88 102 105 92 ...
##  $ engl   : int  2 2 2 2 2 2 2 2 3 2 ...
##  $ engg   : int  3 3 4 2 3 2 2 4 1 4 ...
##  $ gpa    : num  2.6 2.75 4 2.25 3 1.67 2.25 3.4 1.33 3.5 ...
##  $ socprob: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 1 ...
##  $ dropout: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 2 1 1 1 ...
```

Mean

In R, a mean can be calculated on an isolated variable via the mean(VAR) command, where VAR is the name of the variable whose mean you wish to compute. Alternatively, a mean can be calculated for each of the variables in a dataset by using the mean(DATAVAR) command, where DATAVAR is the name of the variable containing the data. The code sample below demonstrates both uses of the mean function.

Calculate the mean of a variable with mean(VAR), where VAR corresponds to one of the variables stored in the data.frame you just opened above:

```
mean(d$gpa)
```

```
## [1] 2.45625
```

Standard Deviation

Within R, standard deviations are calculated in the same way as means. The standard deviation of a single variable can be computed with the sd(VAR) command, where VAR is the name of the variable whose standard

deviation you wish to retrieve. Similarly, a standard deviation can be calculated for each of the variables in a dataset by using the sd(DATAVAR) command, where DATAVAR is the name of the variable containing the data. The code sample below demonstrates both uses of the standard deviation function.

Calculate the standard deviation of a variable with sd(VAR):

```
sd(d$gpa)
```

```
## [1] 0.8614307
```

Minimum and Maximum

Keeping with the pattern, a minimum can be computed on a single variable using the min(VAR) command. The maximum, via max(VAR), operates identically. However, in contrast to the mean and standard deviation functions, min(DATAVAR) or max(DATAVAR) will retrieve the minimum or maximum value from the entire dataset, not from each individual variable. Therefore, it is recommended that minimums and maximums be calculated on individual variables, rather than entire datasets, in order to produce more useful information. The sample code below demonstrates the use of the min and max functions.

Calculate the min and max of a variable with min(VAR) and max(VAR):

```
min(d$gpa)
```

```
## [1] 0.67
```

```
max(d$gpa)
```

```
## [1] 4
```

Range

The range of a particular variable, that is, its maximum and minimum, can be retrieved using the range(VAR) command. As with the min and max functions, using range(DATAVAR) is not very useful, since it considers the entire dataset, rather than each individual variable. Consequently, it is recommended that ranges also be computed on individual variables.

This operation is demonstrated in the following code sample.

```
range(d$gpa)
```

```
## [1] 0.67 4.00
```

Range returns two numbers in a vector.

Percentiles

Values from Percentiles (Quantiles)

Given a dataset and a desired percentile, a corresponding value can be found using the quantile(VAR, c(PROB1, PROB2,...)) command. Here, VAR refers to the variable name and PROB1, PROB2, etc., relate to desired probability values. The probabilities must be between 0 and 1, therefore making them equivalent to decimal versions of the desired percentiles (i.e. 50% = 0.5). The following example shows how this function can be used to find the data value that corresponds to a desired percentile.

Calculate the 25th and 75th percentile values using quantile(VAR, c(PROB1, PROB2,...)):

```
quantile(d$gpa, c(0.25, 0.75))
```

```
##  25%  75%
## 1.75 3.00
```

Note that quantile(VAR) command can also be used. When probabilities are not specified, the function will default to computing the 0, 25, 50, 75, and 100 percentile values, as shown in the following example.

```
quantile(d$gpa)
```

```
##    0%   25%   50%   75%  100%
## 0.670 1.750 2.635 3.000 4.000
```

Calculate the mean, standard deviation, variance, and SE mean for ADD score, IQ score, Grade in 9th grade english, GPA in the 9th grade

---

Summary

A very useful multipurpose function in R is summary(X), where X can be one of any number of objects, including datasets, variables, and linear models, just to name a few. When used, the command provides summary data related to the individual object that was fed into it. Thus, the summary function has different outputs depending on what kind of object it takes as an argument. Besides being widely applicable, this method is valuable because it often provides exactly what is needed in terms of summary statistics. A couple examples of how summary(X) can be used are displayed in the following code sample. I encourage you to use the summary command often when exploring ways to analyze your data in R.

Summarize a variable with summary(VAR):

```
summary(d$gpa)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.670   1.750   2.635   2.456   3.000   4.000
```

Summarize a dataset with summary(DATAVAR):

```
summary(d)
```

```
##        ID            addsc          gender   repeat.        iq
##  Min.   : 1.00   Min.   :26.00   female:33   No :76   Min.   : 75.00
##  1st Qu.:22.75   1st Qu.:44.75   male  :55   Yes:12   1st Qu.: 90.75
##  Median :44.50   Median :50.00                        Median :100.00
##  Mean   :44.50   Mean   :52.60                        Mean   :100.26
##  3rd Qu.:66.25   3rd Qu.:60.25                        3rd Qu.:108.25
##  Max.   :88.00   Max.   :85.00                        Max.   :137.00
##       engl            engg            gpa        socprob  dropout
##  Min.   :1.000   Min.   :0.000   Min.   :0.670   No :78   No :78
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:1.750   Yes:10   Yes:10
##  Median :2.000   Median :3.000   Median :2.635
##  Mean   :1.955   Mean   :2.659   Mean   :2.456
##  3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.:3.000
##  Max.   :3.000   Max.   :4.000   Max.   :4.000
```

Complete Summary Statistics Analysis

There is a fast way to obtain a broad range of summary statistics from an entire dataframe, using the **describe()** function contained in the **psych** package:

```
library(psych)
describe(d)
```

```
##          vars  n   mean    sd median trimmed   mad   min max range  skew
## ID          1 88  44.50 25.55  44.50   44.50 32.62  1.00  88 87.00  0.00
## addsc       2 88  52.60 12.42  50.00   52.18 10.38 26.00  85 59.00  0.39
## gender*     3 88   1.62  0.49   2.00    1.65  0.00  1.00   2  1.00 -0.51
## repeat.*    4 88   1.14  0.35   1.00    1.06  0.00  1.00   2  1.00  2.08
## iq          5 88 100.26 12.98 100.00   99.67 13.34 75.00 137 62.00  0.38
## engl        6 88   1.95  0.52   2.00    1.94  0.00  1.00   3  2.00 -0.06
## engg        7 88   2.66  0.95   3.00    2.71  1.48  0.00   4  4.00 -0.26
## gpa         8 88   2.46  0.86   2.63    2.49  0.93  0.67   4  3.33 -0.34
## socprob*    9 88   1.11  0.32   1.00    1.03  0.00  1.00   2  1.00  2.39
## dropout*   10 88   1.11  0.32   1.00    1.03  0.00  1.00   2  1.00  2.39
##          kurtosis   se
## ID          -1.24 2.72
## addsc       -0.11 1.32
## gender*     -1.76 0.05
## repeat.*     2.37 0.04
## iq          -0.28 1.38
## engl         0.57 0.06
## engg        -0.52 0.10
## gpa         -0.73 0.09
## socprob*     3.77 0.03
## dropout*     3.77 0.03
```

Additional Practice

Caffeine in your coffee

The data in the file caffeine.csv shows the amount of caffeine in a 16-oz cup of coffee obtained from various vendors. Import this data into R.

For context, doses of caffeine over 25 mg are enough to increase anxiety in some people, and doses over 300 mg are enough to significantly increase heart rate in most people. Red Bull contains 80mg of caffeine per serving. Analyze this data using the appropriate descriptive statistics function to view the mean amount of caffeine in a 16-oz coffee and the 95% confidence interval.

View the caffeine data in a histogram. Adjust the scale of the X-axis so it has a lower and upper margin of 0%, a minimum of 140, a maximum of 260, and a major increment of 20.

Now import the data contained in the file caffeine-starbucks.csv. This file has data on six 16 oz cups of coffee sampled on six different days from the same Starbucks location. Use R to calculate the mean and standard error for these data.

Answer the next three questions on Sakai.

Hemoglobin levels

Import the data from Hemoglobin.csv into R. This file contains data on blood hemoglobin level from three populations living at high-altitudes (Andes, Ethiopia, and Tibet) and a sea-level population from the USA.

Use the appropriate command to view the mean, standard deviation, and standard error of the mean for hemoglobin concentration (g/dL), according to population.

Answer the last two questions on Sakai.

Work in progress. . .