Chapter 3: Descriptive Statistics: Numerical Methods

Example 1

```
mean(housing$rent)
```

Example 2

```
median(housing$rent)
```

Example 7

```
quantile(housing$rent, type = 2, probs = c(0.25, 0.50, 0.75, 0.90))
```

Example 8

```
boxplot(Salary ~ Position, data = jobs,
      xlab = 'Starting Salaries For Four Areas of Study',
      ylab = 'Starting Annual Salaries ($000)', col='grey', pch = 19)
```

Example 9

```
max(housing$rent) - min(housing$rent)
```

Example 10

```
IQR(housing$rent, type = 2)
```

Example 11

```
C3_1 <- c(40, 45, 38, 50, 47)
var(C3_1)
```

Example 12

```
sd(C3_1)
```

Example 13

```
sd(C3_1) / mean(C3_1)
```

Example 14

```
sd(housing$rent) / mean(housing$rent)
```

Example 16

```
(C3_1 - mean(C3_1)) / sd(C3_1)
```

Example 18

```
min(scale(housing$rent))

max(scale(housing$rent))
```

Example 19

```
cov(city$high_cent, city$low_cent)
```

Example 20

```
cov(polling$x1, polling$x4)

cov(polling$x1, polling$x5)
```

Example 21

```
cor(polling$x1, polling$x4)

cor(polling$x1, polling$x5)

plot(polling$x1, polling$x4, xlab = 'Age',
      ylab = 'Hours Weekly TV Viewing', pch = 19)

plot(polling$x1, polling$x2, xlab = 'Age',
      ylab = 'Annual Income ($)', pch = 19)

plot(polling$x1, polling$x3, xlab = 'Age',
      ylab = 'Views of Same-Sex Marriage', pch = 19)
```

Optional Material: The Geometric Mean

```
(0.630 + 1.2646 + 1.1506 + 1.0211 + 1.1600 + 1.3239 + 1.1369 +
      1.0138) / 8
```

End-of-Chapter 3 Exercises

Exercise 1

```
quantile(LakeHuron, prob = c(0.00, 0.25, 0.50, 0.75, 0.90, 1.00))

mean(LakeHuron)

median(LakeHuron)
```

Exercise 2

```
options(digits = 4)

max(LakeHuron) - min(LakeHuron)

IQR(LakeHuron)

var(LakeHuron)

sd(LakeHuron)

sd(LakeHuron) / mean(LakeHuron)
```

Exercise 3

```
options(scipen=99)

E3_2 <- c(-37.7, -0.3, 0.00, 0.91, exp(1), pi, 5.1, 2*exp(1),
113754)

mean(E3_2)

median(E3_2)
```

```
quantile(E3_2, prob = c(0.78))

var(E3_2)

sd(E3_2)
```

Exercise 4

```
x <- c(24, 22, 22, 21, 19)

y <- c(27, 24, 23, 21, 19)

data <- data.frame(X = x, Y = y)

cor(data$X, data$Y)

plot(data$X, data$Y, pch = 19, xlab = 'x', ylab = 'y')
```

Exercise 5

```
E3_3 <- seq(from = 10, to = 100, by = 10)

xbar <- mean(E3_3)

devs <- (E3_3 - xbar)

sqrd.devs <- (devs)^2

sum.sqrd.devs <- sum(sqrd.devs)

variance <- sum.sqrd.devs / (length(E3_3) - 1)
```

```
standard.deviation <- sqrt(variance)

var(E3_3)

sd(E3_3)
```

R Functions

. boxplot() A method of exploratory data analysis, boxplot() creates a box plot display of quantitative data. It is mostly based on the Five-Number Summary (see summary(), below).

. cor() Provides the correlation coefficient between two quantitative variables.

. cov() Provides the covariance between two quantitative variables.

. IQR() Reports the inter-quartile range, the difference between the third and first quartiles of a set of quantitative data.

. length() Reports the number of data items in an object.

. max(scale()) Note that scale() is nested within max(), and provides the maximum standardized value in a set of quantitative data.

. mean() Reports the arithmetic mean of a set of quantitative data.

. median() Reports the median of a set of quantitative data.

. min(scale()) Here scale() is nested within min(), and provides the minimum standardized value in a set of quantitative data.

. quantile() Reports the values of any desired percentiles.

. rexp() Generates a series of data values that are exponentially distributed.

. rnorm() Generates a series of data values that are normally distributed. The default normal distribution is the standard normal, but it is possible to override by specifying the preferred values for the mean and standard deviation.

. scale() Provides the standardized values, or z scores, of all data items.

. sd() Provides the standard deviation of a quantitative variable.

. seq() Can be used to generate a sequence of numbers whose minimum, maximum, and inter-value distance is defined by the arguments in ().

. var() Provides the variance of a quantitative variable.

Appendix

```
vector1 <- c(1, 2, 3, 4, 5)

vector2 <- vector1 – 3


vector3 <- vector1 + 5
```

```
vector4 <- vector1 * 2

vector5 <- vector1 / 2

vector6 <- (vector1) ^ 2

sum(vector1)

prod(vector1)

sum((vector1) ^ 2)

sum(vector1) / length(vector1)
```