# Lesson 01: Introduction to Statistics

**References**

- Black, Chapter 1 Introduction to Statistics (pp. 3-49)
- Kabakoff, Chapter 2.2.5 Factors (pp. 30), Chapter 5.2.3 Setting the Seed for RNG (pp. 96)
- Davies, Chapter 2.3 Vectors (pp. 23-27), Chapter 8.2 Reading in External Data Files (pp 153-154)
- Stowell, Chapter 3 Selecting a Random Sample from a Dataset (pp. 44)

**Data set: home_prices.csv**

**Description:** This data file is derived from a random sample of home resale records maintained by realtors. There are 117 observations and eight variables:

1. PRICE = Selling price ($hundreds)
2. SQFT = Square feet of living space
3. YEAR = Year of construction (year)
4. BATHS = Number of bathrooms
5. FEATS = Number out of 11 features (dishwasher, refrigerator, microwave, disposal, washer, intercom, skylight(s), compactor, dryer, handicap fit, cable TV access)
6. NBR = Located in northeast sector of city (YES) or not (NO)
7. CORNER = Corner location (YES) or not (NO)
8. TAX = Annual taxes ($)

```r
# To read the comma-separated value (.csv) file into R, we can use the read.csv()
# function. We will read-in the home_prices.cvs and assign the data frame to "houses."

houses <- read.csv("home_prices.csv")

# R will treat most of the variables as numeric (or integer), but note that the NBR
# and CORNER variables are character strings and the default behavior of the read.csv()
# function is to make these factor variables, with internal codes being NO = 1 and
# YES = 2 (not 0 and 1)

# We will examine the structure of the data frame:

str(houses)
```

```
## 'data.frame':    117 obs. of  8 variables:
##  $ PRICE : num  1350 2550 1550 1828 1800 ...
##  $ SQFT  : int  1142 1478 1480 1299 1121 1400 1505 1050 900 1215 ...
##  $ YEAR  : int  1959 1961 1965 1967 1968 1969 1969 1970 1971 1971 ...
##  $ BATHS : num  1.5 2 1.5 1 1.5 1.5 1.5 1 1 1.5 ...
##  $ FEATS : int  0 3 4 6 4 1 2 1 3 3 ...
##  $ NBR   : Factor w/ 2 levels "NO","YES": 1 2 1 2 2 1 1 2 1 2 ...
##  $ CORNER: Factor w/ 2 levels "NO","YES": 1 2 1 1 1 2 2 1 1 1 ...
##  $ TAX   : num  558 1565 1275 1462 995 ...
```

```r
# We could look at the first or last few records of the data frame, or at summary
# statistics using the head(), tail() and summary() functions.

# ?read.csv(), ?str(), ?head(), ?tail(), ?summary() to review documentation pages
```

**Exercises:**

a) What are the measurement levels of each of the eight variables?

```
PRICE, SQFT, TAX and FEATS are ratio variables; YEAR is an interval variable. Ratio and
interval measurements are referred to as quantitative variables and can be either discrete
or continuous depending on their nature. FEATS is a count. Since there can be a zero count
for FEATS, and the arithmetic difference between two counts is meaningful, FEATS is a
discrete quantitative variable at the ratio level. YEAR is an interval variable since
there is no firm zero starting time. YEAR equal to zero is by definition or convention.

BATHS is an ordinal variable; NBR and CORNER are nominal variables. Nominal or ordinal
measurements cannot be measured numerically and are referred to as categorical variables.
BATHS is not really a count. This is an instance where numbers are assigned to ordered
categories. Although there can be zero baths theoretically, the ratios and differences
that may be formed are not meaningful. For example, what is meaningful about the ratio
between 1.5 baths and 2.5 baths?  From another point of view, is the difference
between 2 and 3 baths twice that of between 1.0 and 1.5 baths? Since the ratios and
differences are not meaningful, BATHS is categorical at the ordinal level measurement.
```

b) Should any variable have its values changed to better reflect its true nature?

```
YEAR could be expressed in terms of the age of a house.  This would not change its nature
since the age would keep changing depending on what was taken to be the present date.
BATHS could be expressed in terms of an ordered scale as long as each category in the
scale had a definition.
```

c) From the vector "price", select a simple random sample of size 12. Assign the sample to the name "SRS."" Print SRS and determine the mean value.

```r
# This and the remaining exercises concern the variable PRICE, so we will create
# a vector, "price"
price <- houses$PRICE

# Next, we will select a simple random sample of size 12. However, prior to sampling,
# we will seed the random number generator so that results will be reproducible:
set.seed(9999)

# We will use the sample() function to select a random sample of size 12, assigning
# our n = 12 sample vector to "SRS."
SRS <- sample(price, 12)

# Lastly, we will print the values of SRS and determine its mean
print(SRS)
```

```
##  [1] 4500.0 5250.0 2575.0 2550.0 2625.0 1875.0 3882.5 2612.5 4360.0 1915.0
## [11] 2187.5 1550.0
```

```r
mean(SRS)
```

```
## [1] 2990.208
```

```
# ?set.seed(), ?sample(), ?print(), ?mean() to review documentation pages
```

d) From the vector "price", select a systematic sample of twelve observations. Start with the seventh observation and pick every 10th observation thereafter (i.e. 7, 17, 27,). You should end with the 117th observation. Assign the sample vector to "SS."" Print the values of SS and compute the mean value.

```
# We will select a systematic sample of twelve observations. Starting with the
# seventh observation, we'll select every 10th observation thereafter (i.e. 7, 17, 27,..)

# ?seq() to review documentation page
SS <- price[seq(from = 7, to = 117, by = 10)]

# We could verify that the first three and the last value of SS are as expected.
# Each of the below equalities shoudl evaluate to TRUE.
# price[7] == SS[1]
# price[17] == SS[2]
# price[27] == SS[3]
# price[117] == SS[12]

# We will print the values of SS and compute its mean:
print(SS)
```

```
##  [1] 1750.0 2347.5 3250.0 3997.5 3125.0 2950.0 4062.5 5250.0 2822.5 3325.0
## [11] 5375.0 3900.0
```

```
mean(SS)
```

```
## [1] 3512.917
```

e) Examine the printed values and mean values obtained from the two sampling procedures. Do you see a difference? Try the commands summary(SRS) and summary(SS).

```
# In items (c) and (d), we printed the SRS and SS vectors and determined the mean of
# each. Here, we will use the summary() function:
summary(SRS)
```

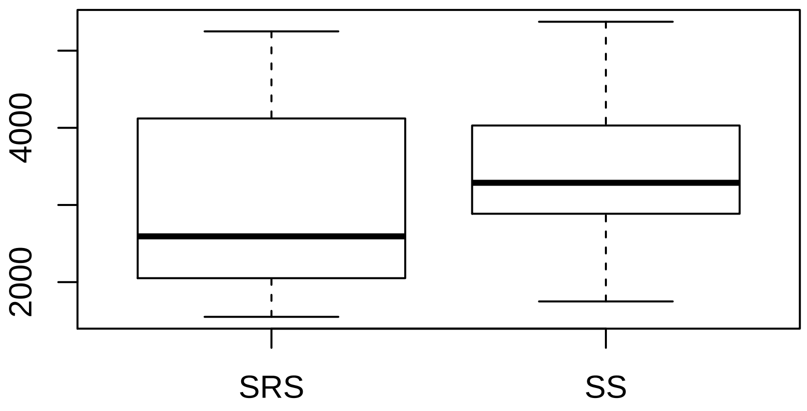
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1550    2119    2594    2990    4002    5250
```

```
summary(SS)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1750    2918    3288    3513    4014    5375
```

f) Create boxplots for SRS and SS using boxplot(). How do the two samples compare?

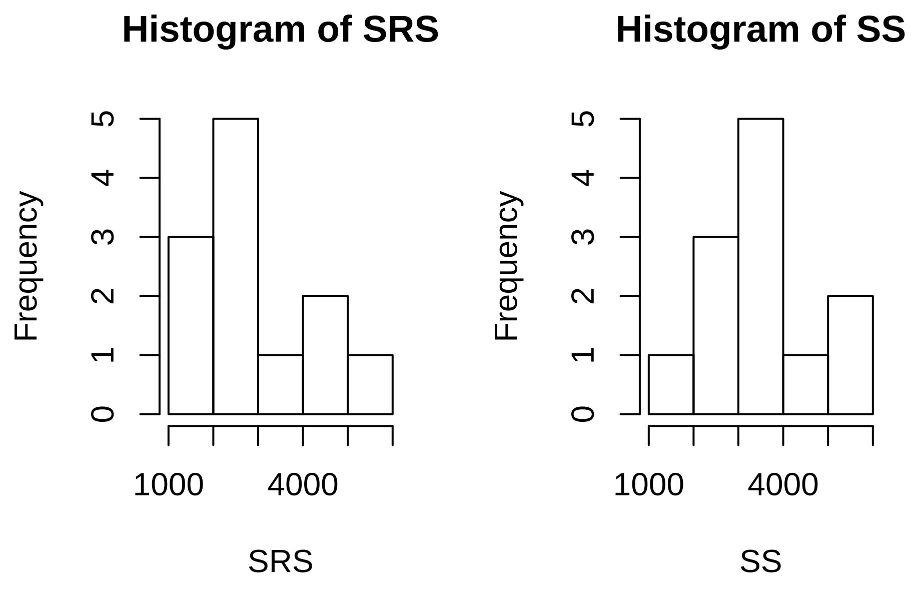
```
boxplot(list(SRS = SRS, SS = SS))
```



```r
# boxplot(SRS, SS) also produces the desired figure, but "loses" the vector names.
# We could, rather than specifying a named list like we did above, provide the vector
# names to the "names" argument; i.e. boxplot(SRS, SS, names = c("SRS", "SS")).
```

g) Create histograms and stem-and-leaf plots for SRS and SS using hist() and stem(). How do the two samples compare?

```r
# ?hist(), ?stem() to review documentation pages
par(mfrow = c(1, 2))
hist(SRS)
hist(SS)
```



```r
par(mfrow = c(1, 1))

stem(SRS)
```

```
##
##   The decimal point is 3 digit(s) to the right of the |
##
##   1 | 699
##   2 | 26666
##   3 | 9
##   4 | 45
##   5 | 3
```

```r
stem(SS)
```

```
##
##   The decimal point is 3 digit(s) to the right of the |
##
##   1 | 8
##   2 | 38
##   3 | 01339
##   4 | 01
##   5 | 34
```