# Warm-Up 02 - Vectors

Stat 133, Fall 2018, Prof. Sanchez

*Due date: Sep-11 (before midnight)*

## Introduction

The purpose of this assignment is to keep working with vectors of different data types, factors, and some basic plots. More specifically you will summarize quantitative and qualitative variables graphically and numerically. These tasks are often the first step in analyzing most data sets. Technically, this involves performing a univariate analysis, that is, analyzing one variable at a time. In future assignments you will work on analyzing two or more variables simultaneously.

Keep in mind that summarizing and describing data, as simple as it may sound, can be tricky. Why? Because there is not one right way to analyze data, but there are wrong ways. Do your best to describe what you see. Jot down notes to capture your thinking as you go. It does not matter if you lack the technical terminology to write such descriptions: use your own words. It takes practice to learn to describe distributions and write an analysis. You will work on these skills over the rest of the semester.

Use this assignment to keep developing your manipulation skills of basic data objects in R: use of bracket notation, understanding vectorization, coercion rules, recycling, etc.

### General Instructions

- Write your narrative and code in an `Rmd` (R markdown) file.
- Name this file as `warmup02-first-last.Rmd`, where `first` and `last` are your first and last names (e.g. `warmup02-gaston-sanchez.Rmd`).
- Please do not use code chunk options such as: `echo = FALSE`, `eval = FALSE`, `results = 'hide'`. All chunks must be visible and evaluated.
- Submit your Rmd and html files to bCourses.
- If you have questions/problems, don't hesitate to ask us for help in OH or piazza.

## Data

The data objects for this assignment are in the file `tents.RData`, inside the `data/` folder of course github repo. There is also the data dictionary file `tents-dictionary.md`.

### Download the data file

To read the data in R, we recommend that you download the `.RData` file to your computer. You can use the function `download.file()` to do this. The file will be downloaded to the specified destination (`destfile`). In the code below, the binary file will be downloaded to your working directory:

```
# assembling url so it fits on the screen
# (Do not include this code in your Rmd file)
github <- 'https://github.com/ucb-stat133/stat133-fall-2018/'
repo <- 'raw/master/data/tents.RData'

download.file(
  url = paste0(github, repo),
  destfile = "tents.RData")
```

You only have to download the file once. By the way, there is NO need to include the previous command in your source `.Rmd` file. Otherwise, everytime you knit the file, R will download the file.

### Importing the data

Open a new `.Rmd` file (this will be your source file). Once you have `tents.RData` in your computer, use the `load()` function, which allows you to import `.RData` files into R:

```
# load the data objects
# (assuming the data file is in your working directory)
load("tents.RData")

# list the available objects
ls()
```

### Inspect the objects in the `.RData` file

The first step is to inspect the data. This is actullay the first contact stage. And to be honest, most of the work you do in this stage never gets reported. But this does not mean that it is worthless or less important. Here are various questions for you to consider while "getting to know the data":

- Make sure you have all the objects described in the data dictionary. Hint: `ls()` is your friend.
- What class of objects are in the file?
- Are there any vectors, factors, lists?
- What flavor is each vector (i.e. variable)?
- Check that all objects have the same length.

## 1) A bit of data preprocessing

Once you have imported the `RData` file `tents`, the next step involves performing a first exploration.

- inspect the structure of objects in `tents.RData`
- use `str()` on `tent` names
- take a look at the first 5 tent names
- take a look at the last 3 tent names

### Quantitative Variable

- Use the `summary()` function to get a quick summary of descriptive statitstics for `price`.
- Now, look up for functions that allow you to get the following statistics:
  - mean (i.e. average)
  - standard deviation
  - minimum value
  - maximum value
  - median
  - quartiles
- Look at the distribution: use `hist()` and `boxplot()`.
- You can also try to get a density curve (or density polygon). Find out how to do this.
- Choose another quantitative variable (e.g. `height`, `weight`). Describe the overall pattern (shape, center, and spread) and striking deviation from the pattern.

### Qualitative Variable

- Pick one of the categorical variables: e.g. `bestuse`, `season`, `capacity`.
- If the variable that you choose is not an R factor, then use `factor()` to convert the object into a factor.
- Use `table()` to get a frequency table (i.e. counts of each category).
- Find out how to use the obtained frequency table to calculate relative frequencies (proportions).
- Use the frequencies (counts) and relative frequencies (proportions) to describe the overall distribution.
- Use `barplot()` to display the frequencies with a barchart.

## 2) Scatterplot of Height and Price

Study the relationship between `height` and `price` with a scatterplot. To make such a scatterplot you will have to use the "plotting from scratch" approach. See slides

https://github.com/ucb-stat133/stat133-fall-2018/blob/master/slides/08b-base-graphics2.pdf

use the following functions:

- `plot.new()`
- `plot.window()`
- `axis()`
- `title()`
- `points()`

Play with the following graphical parameters:

- `pch`: point character (plotted symbol)
- `col`: color of points
- `cex`: character expansion (size of points)
- `xlab`: x-axis label
- `ylab`: y-axis label
- `main`: main title

Don't worry too much about the visual appearance of your plot. Later in the course we will spend some time talking about data visualization in a more formal way. Focus instead on providing a concise description of the patterns observed in the scatterplot.

Optionally, you can also try using graphing scatterplots (and other graphs) with the package `"plotly"`. Feel free to play with it.

https://plot.ly/r/

## 3) Correlation between Height and Price

Study the relationship between `height` and `price` by calculating their correlation (i.e. linear correlation coefficient). R has the function `cor()` that computes this type of correlation. However, we want you to practice writing commands: creating objects, and working with vectors. So instead of using `cor()`—and other related functions—you will have to "manually" compute the correlation as well as other summary statistics. To achieve these tasks, you have to create R objects (and display their values) for:

- $n$: number of individuals
- $\bar{x}$: mean of variable $X$ (`height`)
- $\bar{y}$: mean of variable $Y$ (`price`)
- $var(X)$: variance of $X$
- $var(Y)$: variance of $Y$
- $sd(X)$: standard deviation of $X$
- $sd(Y)$: standard deviation of $Y$
- $cov(X, Y)$: covariance between $X$ and $Y$
- $cor(X, Y)$: correlation between $X$ and $Y$

Note: you are NOT allowed to use functions such as `mean()`, `var()`, `cov()`, `cor()`, or `lm()`. Nor you can use any type of loop (e.g. `for`, `while`, `repeat`). The only auxiliary function that you are allowed to use is `sum()` to implement operations that involve summation $\sum_{i=1}^{n}$. The most important concept here is knowing that most operations with vectors in R are **vectorized**.

The **mean** of a variable $X$, denoted by $\bar{x}$, is given by:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

where:

- $n$ is the number of individuals (i.e. number of elements in $X$)
- $x_i$ is the $i$-th element of $X$
- $\sum_{i=1}^{n}$ is the summation symbol

Similarly, the **mean** of $Y$, denoted by $\bar{y}$, is given by:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Use the function `sum()` to implement operations that involve summation $\sum_{i=1}^{n}$ (recall vectorized operations in R).

The (sample) **variance** of a variable is given by:

$$var(X) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

From the variance, you can derive the (sample) **standard deviation** as:

$$sd(X) = \sqrt{var(X)}$$

In turn, the (sample) **covariance** between two variables $X$ and $Y$ is given by:

$$cov(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

Finally, the **correlation** between $X$ and $Y$ is given by:

$$cor(X,Y) = \frac{cov(X,Y)}{sd(X)sd(Y)}$$

### More Manipulations

- Create a new vector `weight_lbs` for weight in pounds. Keep in mind that `weight` is given in grams.
- Create a new vector `height_in` for height in inches. Keep in mind that `height` is given in centimeters.

Write R code (using bracket notation) to answer the following questions:

- how many tents have a price less than or equal to $300
- how many tents have a price between $300 and $400 (including both $300 and $400 prices)
- what's the name of the tent with maximum price
- how many tents have a price > $400 AND weight < 1500 grams
- calculate the 90th percentile for height and assign it to the object `height_p90` (display this value)
- calculate the 90th percentile for weight and assign it to the object `weight_p90` (display this value)
- display the name of the tents with `height > height_p90` AND `weight > weight_p90`

### Working with factors

Use `cut()` to create a factor `weight_cut` by using the breaking points and labels according to the following table:

```
----------------------
     intervals | labels
---------------|-------
     (0, 1000] |  1kg
  (1000, 2000] |  2kg
  (2000, 3000] |  3kg
  (3000, 4000] |  4kg
  (4000, 5000] |  5kg
  (5000, 6000] |  6kg
  (6000, 7000] |  7kg
  (7000, 8000] |  8kg
  (8000, 9000] |  9kg
----------------------
```

The interval `(0, 1000]` means a range from 0 to 1000 (excluding 0, including 1000).

Verify that the frequencies given by `table(weight_cut)` are:

```
 1kg  2kg  3kg  4kg  5kg  6kg  7kg  8kg  9kg
   3   24   30    7    7    2    9    3    3
```

**Reordering a factor**

The variable `season` is a character vector. Convert it into a factor with three `levels`; `factor()` is your friend. You can use functions such as `summary()`, `nlevels()`, and `is.ordered()` to inspect any R factor object.

Find out how to use the `factor()` function in order to convert `season` as an **ordinal** factor. The resulting levels should be `3-season`, `3-4-season`, and `4-season` (in this order!). Furthermore, if you execute the command `is.ordered(season)` the answer should be `TRUE`.

## What are we looking for from your work?

- We're NOT expecting high production value (yet). Aesthetics, design elements, and grammar are not very important right now.
- Having said that, do NOT just include R output; make an effort to include concise descriptions of the results and displays that you're obtaining.
- Examine your graphs and summary statistics in order to make observations about shape, center, spread and outliers (if there are any).
- Try to describe the graphs in a comprehensive way:
  - What is the overall shape? (e.g. symmetryc, right skewed, left skewed)
  - What is the typical center? (e.g. mode, median, center)
  - Overall range, along with an interval of typical measurements. (e.g. range)
- Try to explain why your observations are important or interesting.
- Add transitions to your narrative that help tie your observations.
- Use lots of inline code! Especially when writing paragraphs describing data and related values. Try not to hard-code values—this breaks computational reproducibility.
- Keep practicing with Markdown syntax: use bullet/itemized lists, embed bullets within lists, italics, bold, links, headings of different levels, pre-format (i.e. code), horizontal rules, etc.

If this (and future) assignment looks more like an English (writing) assignment, you are right. We want to remind you that *Computing with Data* (CwD) is not just about computations, solving problems, and obtaining results. CwD also involves making sense of the computed results. Whether you work as a consultant, scientist, analyst, journalist, programmer, etc, you will have to explain your findings, report your work, and communicate it to several audiences.

## Comments and Reflections

Reflect on what was hard/easy, problems you solved, helpful tutorials you read, etc.

- What things were hard, even though you saw them in class?
- What was easy(-ish) even though we haven't done it in class?
- Did you need help to complete the assignment? If so, what kind of help? Who helped you?
- How much time did it take to complete this HW?
- What was the most time consuming part?
- Was there anything that you did not understand? or fully grasped?
- Was there anything frustrating in particular?