Statistics 305/605: Introduction to Biostatistical Methods for Health Sciences

R Demo for Chapters 2 & 3: Data presentations and summaries

Jinko Graham, Brad McNeney

Demo Slides

- This document shows and explains the R commands used to create the data summaries of the Chapters 2 & 3 lecture slides.
- This document should be read **after** reading the lecture slides for Chapters 2 & 3.

Data Presentation (Chapter 2)

Tables

- Tables can be used to display the frequency distribution of a categorical variable
- Example: Frequency distribution of gender among 21,737 bladder cancer patients. Data from Mungan et al. (2000)

```
uu <- url("http://people.stat.sfu.ca/~jgraham/Teaching/S305_17/Data/mung.csv")
Mungan <- read.csv(uu)
head(Mungan)
```

```
##   Gender Cancer.Stage
## 1   Male            I
## 2   Male            I
## 3   Male            I
## 4   Male            I
## 5   Male            I
## 6   Male            I
```

```
with(Mungan,table(Gender))
```

```
## Gender
## Female   Male
##   5536  16201
```

Software Notes: R objects

- ▶ When you start R you are starting a "session".
- ▶ Data that you read into R and the results of computations on data are stored as R "objects" within your "workspace" or "environment".
  - ▶ You can see a list of all objects in your environment in "Environment" tab of the the upper-right pane in RStudio, or you can type ls() in the R console.
- ▶ We assign values to objects with the assignment operator <−
  - ▶ For example, the following creates an object uu that contains the output of the function url():

```
uu <- url("http://people.stat.sfu.ca/~jgraham/Teaching/S305_17/Data/mung.csv")
```

Software Notes: R functions

- ▶ Most of the data analyses we do in R is with functions.
- ▶ You are familiar with functions such as `log` on your calculator that take a number as input, and output another number.
- ▶ In R, functions are used to do such calculations with data, but also to do non-data-related tasks such as managing R objects, reading/writing datasets, etc.
- ▶ R function inputs:
  - ▶ The inputs, sometimes called "arguments", are the R objects the function will work with.
- ▶ R function output:
  - ▶ The output, or what the function "returns", is an R object that summarizes the computations or other work done by the function.

Softward Notes: R functions, cont.

- ► `log(x)` takes a number `x` as input and returns its natural log; e.g.,

```r
log(1)
```

```
## [1] 0
```

- ► `ls()` lists the R objects in your workspace – no arguments needed.

```r
ls()
```

```
## [1] "Mungan" "uu"
```

Software Notes: Reading Data Into R

- `read.csv()` reads comma-separated-value (CSV) files into R.
  - By default this function reads files from the "working" directory in which R is running (e.g., the project directory of your RStudio project or the folder of your Jupyter notebook), but it can read files from URLs too.
  - The `url()` function takes a quoted URL as input and returns an object that other functions, such as `read.csv()`, can use to fetch the file from the internet.

- ▶ read.table() is a more flexible function than 'read.csv()' for reading data into R.

- ▶ It can easily read in comma-separated-value (CSV) files as well as files with values separated by other characters such as blank spaces or tabs.

- ▶ For example, the CSV file mung.csv can be read into R with read.table() as follows:

```
uu <- url("http://people.stat.sfu.ca/~jgraham/Teaching/S305_17/Data/mung.csv")
Mungan <- read.table(uu,header=TRUE,sep=",")
```

- ▶ read.table() options include the following:
    - ▶ header (default FALSE): Does the first line of the file contain the variable names?
    - ▶ sep (default "", for blank spaces)
- ▶ To get a full list of options for 'read.table()', type help("read.table") into R.

Software Notes: `head()`,`with()` and `table()`

- The `head()` function looks at the first few rows (default is six) of a dataset.
  - In the example, the dataset is called `Mungan`, and has variables `Gender` and `Cancer.Stage`.
  - Datasets have as many rows as there are sampled units (e.g., people) and as many columns as there are variables measured on the sampled units.
- The `with()` function takes a dataset as its first argument and the summary to compute as its second argument.
  - In the above example, the summary is a table of the values of the `Gender` variable in the `Mungan` dataset.
- The `table()` function tabulates the unique values of a variable, or, if given two variables, cross-tabulates the two variables (more on cross-tabulation in Chapter 15).

Tables, cont.

▶ Joint frequency distribution of two categorical variables:

```
with(Mungan,table(Gender,Cancer.Stage))
```
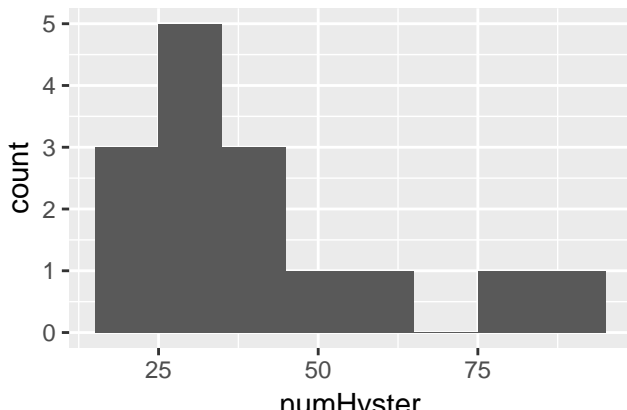
```
##         Cancer.Stage
## Gender      I    II   III    IV
##   Female 3926   402   356   852
##   Male  12418   995   883  1905
```

Example Histogram

▶ Data on the numbers of hysterectomies performed by 15 male Swiss doctors:
20 25 25 27 28 31 33 34 36 37 44 50 59 85 86

```
hyst <- data.frame(numHyster = c(20,25,25,27,28,31,33,34,
                                 36,37,44,50,59,85,86))
library(ggplot2)
ggplot(hyst,aes(x=numHyster)) + geom_histogram(binwidth=10)
```

Software Notes: Data Frames

- We used the `data.frame()` function to create a data frame with a single variable `numHyster`.
  - Data frames are objects used to store datasets in R.
  - Typically a data frame consists of multiple variables, such as the Mungan data frame with variables `Gender` and `Cancer.Stage`.
  - Use `names()` to find the names of variables in a data frame:

```
names(Mungan)
```

```
## [1] "Gender"       "Cancer.Stage"
```

Software Notes: Add-on Packages

- The code chunk that draws the histogram of the hysterectomy data loads an add-on package for R called `ggplot2`.
- R consists of a "base" distribution plus many add-on packages that contain useful functions.
  - For example, `ggplot2` is a package that contains the graphics function `ggplot()`.
- To use the functions in a package you must **first** load the package with `library()`.
  - For example, `library(ggplot2)` loads `ggplot2` and gives us access to `ggplot()`.
- If you don't load a package, R can't find its functions.
  - For example, if you haven't yet loaded `ggplot2` and you try to use `ggplot()` you will get an error message:

```
Error: could not find function "ggplot"
```

Software Notes: Installing Add-on Packages

- See this 2-minute Youtube video for a short backgrounder on R packages.
- RStudio users (RStudio Desktop or RStudio Cloud) will need to install packages before they can load them.
    - RStudio-Desktop users should consult the **R Packages** section of the R/Rstudio getting-started document (hover over preceding for link).
    - RStudio-Cloud users should consult step 5 of the RStudio-Cloud getting-started document (hover over preceding for link).
- The `tidyverse` packages are pre-installed for Jupyter users, but others, like `gapminder`, need to be installed every R session.

Software Notes: ggplot()

- ggplot2 is an add-on package for R that implements the graphics function ggplot().
    - We will use ggplot() throughout the course.
- To draw the histogram of the hysterectomy data, the call to ggplot() was

```
ggplot(hyst,aes(x=numHyster)) + geom_histogram(binwidth=10)
```

- This specifies the dataset (hyst) and the "aesthetic", which is a list of variables to plot as different features of the graph.
    - This example is a histogram of numHyster. We specify that numHyster is the x-axis variable with x=numHyster.
    - The function geom_histogram() adds the histogram; it takes the bin width as an optional argument.

Summary Statistics (Chapter 3)

Centre: The mean

- The population mean, μ, is the ordinary arithmetic average of a variable in the population.

- The corresponding statistic is the sample mean, $\bar{x}$.

- The sample mean is the ordinary arithmetic average of the observations in a random sample from the population.

- For example, the hysterectomy example data:

20 25 25 27 28 31 33 34 36 37 44 50 59 85 86

has sample mean

$$\bar{x} = \frac{20 + 25 + \ldots + 86}{n} = 41.3$$

```
library(dplyr)
summarize(hyst,mean(numHyster))
```

```
##   mean(numHyster)
## 1        41.33333
```

Software Note

- dplyr is an add-on package for R that includes useful tools for manipulating datasets in R.
  - The summarize() function takes the dataset as its first argument, and the summaries to compute as additional arguments.
  - In this example we could have instead used with(hyst,mean(numHyster)), but we will eventually want to use summarize() together with other tools from dplyr to produce data summaries.

Centre: The Median

- The population median is the "middle value" of the variable in the population.

- The corresponding statistic is the sample median, M.

- The sample median is the middle value of the variable in a random sample from the population.

- The sample median of the hysterectomy data is:

20, 25, 25, 27, 28, 31, 33, **34**, 36, 37, 44, 50, 59, 85, 86

- The centre observation is $M = 34$.

```
summarize(hyst,median(numHyster))
```

```
##   median(numHyster)
## 1              34
```

Spread: The Standard Deviation (SD) and Variance

- The variance, σ^2, is the average of squared deviations from the mean in the population

- The SD, σ, is the square-root of the variance and measures spread about the mean.

- As for the corresponding statistics:

    - The sample variance, s^2, is (almost) an average of squared deviations from the sample mean in a random sample from the population.
    - The sample SD, s, is the square root of the sample mean.

- Hysterectomy example: $s = 20.6$

```
summarize(hyst,sd(numHyster))
```

```
##   sd(numHyster)
## 1      20.60744
```

Spread: The Inter-Quartile Range (IQR)

- ▶ The first and third quartiles mark the first and third quarters of the observations, whether in a population or in a random sample from the population.
  - ▶ These are also called the 25th and 75th percentiles, respectively.

```
summarize(hyst,
          Q1=quantile(numHyster,probs=.25),
          Q3=quantile(numHyster,probs=.75))
```

```
##     Q1 Q3
## 1 27.5 47
```

- ▶ The middle half of the data lies between.
- ▶ The range of the middle half, or IQR, is 47-27.5=19.5.
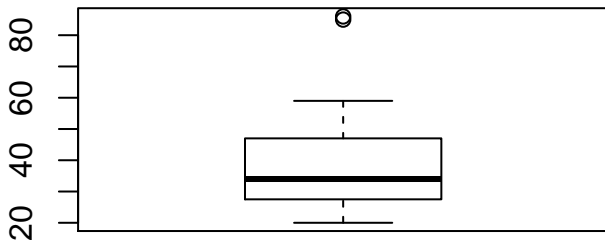
```
summarize(hyst,IQR(numHyster))
```

```
##   IQR(numHyster)
## 1           19.5
```

Boxplots

- The five number summary is the minimum, maximum, median, 1st and 3rd quartiles.
- Graphed with a boxplot in the hysterectomy data:

```
with(hyst,boxplot(numHyster))
```

Boxplots of Data from Multiple Groups

▶ Augment the hysterectomy example by numbers of
  hysterectomies performed by *female* Swiss doctors.

```
uu <- url("http://people.stat.sfu.ca/~jgraham/Teaching/S305_18/Data/hyst.csv")
hyst <- read.csv(uu)
head(hyst)
```
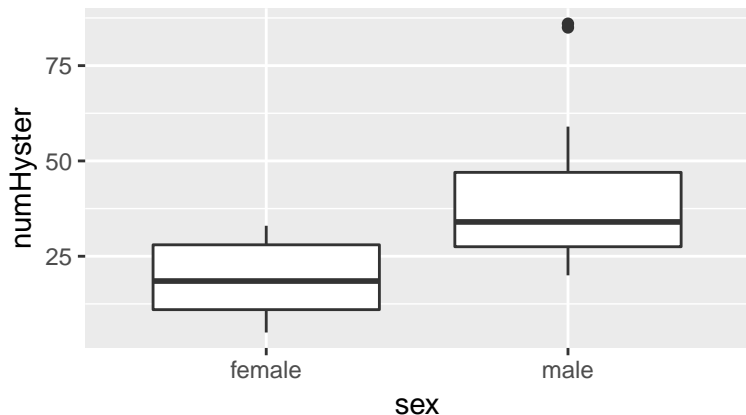
```
##   numHyster  sex
## 1       20 male
## 2       25 male
## 3       25 male
## 4       27 male
## 5       28 male
## 6       31 male
```

```
with(hyst,table(sex))
```

```
## sex
## female   male
##     10     15
```

Boxplots with ggplot()

```
ggplot(hyst,aes(x=sex,y=numHyster)) + geom_boxplot()
```


- ▶ The aes() specification: sex (grouping variable) is on the x-axis and numHyster is on the y-axis.
- ▶ Use geom_boxplot() to add the boxplots.