Statistics 305: Introduction to Biostatistical Methods for Health Sciences

R Demos for Chapters 8-10: Review of Statistical Inference

Jinko Graham, Brad McNeney

Confidence Intervals (Chapter 9)

Example

- Example from the text, page 223, summarizes data on plasma aluminum levels, in $\mu g/l$, for $n = 10$ infants receiving antacids that contain aluminum.
  - The sample mean of the plasma aluminum levels is $\bar{x} = 37.20 \mu g/l$, and the sample SD is $s = 7.13$
- I can't access the data from the text, but I have simulated a data set of $n = 10$ subject with similar properties for this demo.
  - Read these data into R:

```
uu <- url("http://people.stat.sfu.ca/~jgraham/Teaching/S305_18/Data/pa.csv")
plasmaAlu <- read.csv(uu)
head(plasmaAlu)
```

```
##        alu
## 1 33.20381
## 2 35.55883
## 3 48.31359
## 4 37.70272
## 5 38.12182
## 6 49.42841
```

Simulated Data Summary Statistics

- Our simulated data are similar to the data from the text, but the sample mean and SD are different:

```
library(dplyr)
summarize(plasmaAlu,mean(alu))
```

```
##   mean(alu)
## 1  37.73208
```

```
summarize(plasmaAlu,sd(alu))
```

```
##   sd(alu)
## 1 6.80048
```

Software Notes

- Recall from the chapter 2-3 demo that `dplyr` is an add-on package for R that includes useful tools for manipulating datasets in R.
  - To use `dplyr` functions we must first load the package with `library(dplyr)`.
- The package consists of 5 main "verbs" for manipulating a dataframe:

1. `select()`: select columns
2. `filter()`: filter rows
3. `arrange()`: re-order or arrange rows
4. `mutate()`: create new columns
5. `summarize()`: summarise columns

- The `summarize()` function takes the dataframe as its first argument, and the summaries to compute as additional arguments.

Do-It-Yourself CIs in R

- In R, the quantile function `qt()` for the t distribution can be used to find the quantile or critical value (t^*) for the CI when σ is unknown.
    - When the argument `lower.tail=FALSE`, the first argument, `p`, is the upper-tail area to the right of the desired quantile, under the t distribution.

```r
tstar<-qt(p=(1-0.95)/2,df=9,lower.tail=FALSE)
tstar
```

```
## [1] 2.262157
```

- Once we have the quantile or critical value, we can calculate the 95% CI.

- Use assignment operator <- to put the critical value into an R object called `tstar`.
- Then use `summarize()` function in `dplyr` package to get sample mean, sample sd and the number of observations, *n*. Assign these to an R object `mysummary`.
- Use the saved objects to calculate lower and upper bounds of CI and put them into the R object `CI`.

```r
tstar <- qt(p=(1-0.95)/2,df=9,lower.tail=FALSE)
mysummary <- plasmaAlu %>%
  summarize(xbar=mean(alu),s=sd(alu),n=n()) %>%
  mutate(CI.lower=xbar-tstar*s/sqrt(n),
         CI.upper=xbar+tstar*s/sqrt(n))
mysummary
```

```
##       xbar       s  n CI.lower CI.upper
## 1 37.73208 6.80048 10 32.86731 42.59685
```

Chaining Data Manipulations

- In the above code chunk we used the “forward pipe”, %>% to push the dataset from one data manipulation to another.
- Push the plasmaAlu dataframe through the pipe to use as input to summarize(), then take the dataframe output by summarize and push it through the pipe to use as input to mutate().
  - Each data-processing step is a verb, and we are to read the result like a sentence: Start with plasmaAlu, summarize it (mean, SD and sample size), then mutate it to add the lower and upper CI limits.
  - Notice that we do not need to explicitly specify the input data to summarize() and mutate() when they receive these data through the forward pipe.

CIs with the t.test() Function

- R's `t.test()` uses data to test hypotheses about a mean, or about differences between two means, but the function also returns summary statistics and a CI.
- Set the level or coverage probability of the CI with the argument `conf.level` (default $C = 0.95$):

```
with(plasmaAlu,t.test(alu,conf.level=0.95))
```

```
##
##  One Sample t-test
##
## data:  alu
## t = 17.546, df = 9, p-value = 2.872e-08
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  32.86731 42.59685
## sample estimates:
## mean of x
##  37.73208
```

Hypothesis Tests (Chapter 10)

Example

- In the **population** of infants **not taking antacids**, the mean plasma-aluminum levels are known to be $\mu_0 = 4.13\ \mu\text{g/l}$.
- Want to assess whether the mean level μ in infants taking antacids is the same as μ_0; i.e. whether

$$H_0 : \mu = 4.13.$$

- Our alternative hypothesis is $H_a : \mu \neq 4.13$; i.e.,the mean plasma-aluminum levels of infants taking antacids is different from infants not taking antacids.

Do-It-Yourself Hypothesis Test

- The *t*-statistic is

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

- We can implement this as follows using the summaries in the object `mysummary` from the CI demo:

```
mysummary
```

```
##       xbar       s  n CI.lower CI.upper
## 1 37.73208 6.80048 10 32.86731 42.59685
```

```
with(mysummary,(xbar - 4.13)/(s/sqrt(n)))
```

```
## [1] 15.62524
```

- The *p*-value is $2P(T \geq |15.63|)$ for T with $10 - 1 = 9$ df:

```
2*pt(15.63,df=9,lower.tail=FALSE)
```

```
## [1] 7.887651e-08
```

Using the `t.test()` Function

- The `t.test()` function will also perform the test.
    - Specify the null hypothesis $H_0 : \mu = 4.13$ with the `mu` argument.
    - The function's default is to use the two-sided alternative hypothesis, which in this case is $\mu \neq 4.13$:

```
with(plasmaAlu,t.test(alu,mu=4.13))
```

```
##
##  One Sample t-test
##
## data:  alu
## t = 15.625, df = 9, p-value = 7.909e-08
## alternative hypothesis: true mean is not equal to 4.13
## 95 percent confidence interval:
##  32.86731 42.59685
## sample estimates:
## mean of x
##  37.73208
```