Lab 2: Inference Review and Intro. to Simulation

Statistics 139 (special thanks to Julie Vu!)

September 15, 2023

Topics

- Data-driven Inference (a little review)
- Using `sample()`, `set.seed()` and `for` loops
- Using `if` statements
- Probability distributions in R

Problem 1: Data Analysis and Inference Review

A survey was conducted over the last few years to determine what factors are related to heart rate (our first day survey from Lecture 0). The results are saved in 'survey0.csv'. Use this dataset to answer the following questions:

a) Is drinking coffee associated with heart rate? Perform a formal hypothesis test to answer this question and provide a confidence interval to estimate the true difference.

```r
# load data
survey <- read.csv("data/survey0.csv")

# look at data
str(survey)
```

```
## 'data.frame':    176 obs. of  6 variables:
##  $ time     : chr  "9/3/19 9:44" "9/3/19 9:48" "9/3/19 10:22" "9/3/19 10:22" ...
##  $ heartrate: int  60 68 88 170 70 64 66 68 48 74 ...
##  $ exercise : num  10 4 5 3 4 7 2 3 5 2 ...
##  $ gender   : chr  "Male" "Male" "Male" "Male" ...
##  $ classyear: chr  "Grad Student" "Grad Student" "Sophomore" "Junior" ...
##  $ coffee   : chr  "Yes" "Yes" "No" "No" ...
```

```r
table(survey$gender)
```

```
##
##         Female   Male
##      1     67    108
```

```
table(survey$classyear)
```

```
##
## Grad Student       Junior       Senior    Sophomore
##           40           86           34           16
```

```
table(survey$coffee)
```

```
##
##  No Yes
## 116  60
```

```
# two sample t test
test.1a <- t.test(heartrate ~ coffee, data = survey, alternative = "two.sided")
test.1a
```

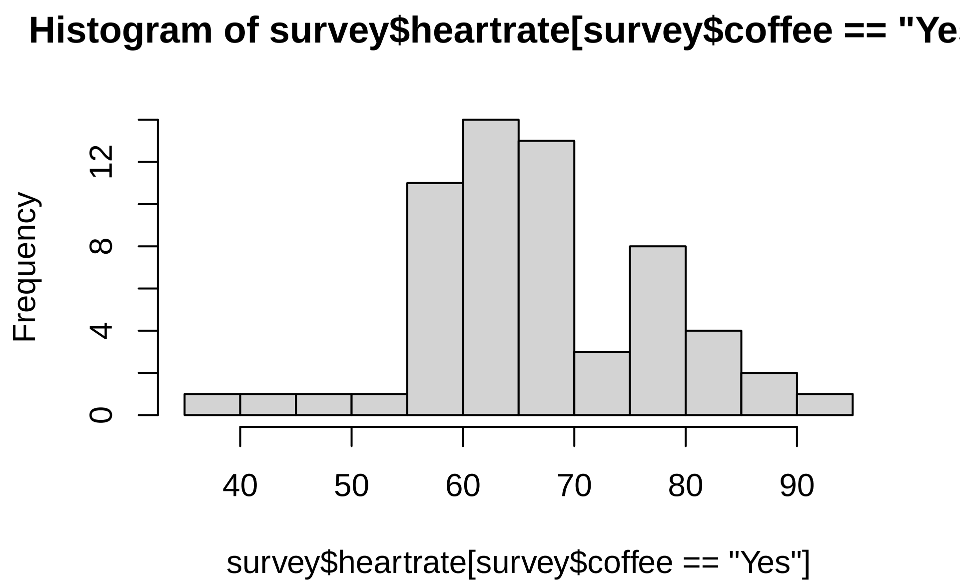
```
##
##  Welch Two Sample t-test
##
## data:  heartrate by coffee
## t = 2.7123, df = 152.37, p-value = 0.007451
## alternative hypothesis: true difference in means between group No and group Yes is not equal
## 95 percent confidence interval:
##  1.426449 9.078149
## sample estimates:
##  mean in group No mean in group Yes
##          72.56897          67.31667
```

- Two sample t test (not paired)
- 95% confidence interval:

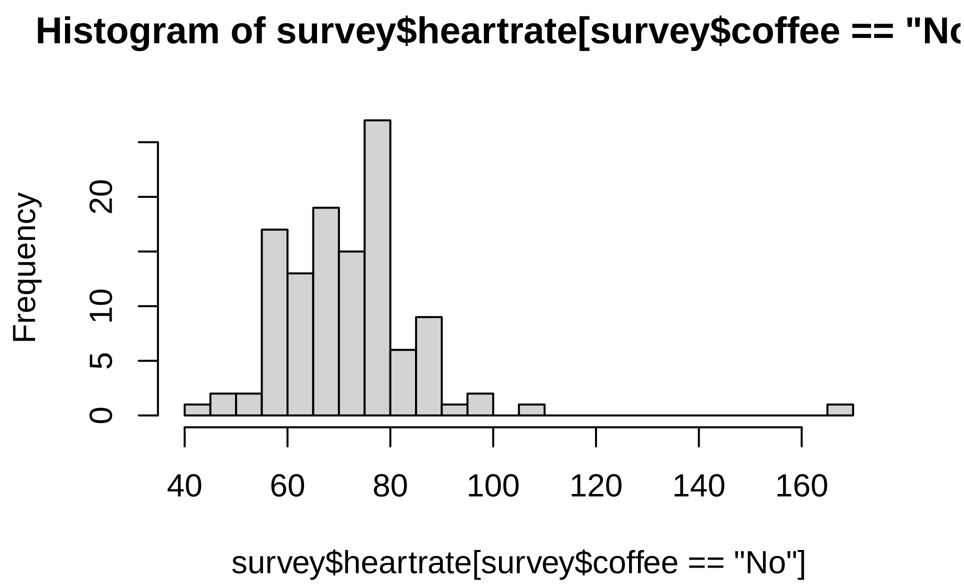
b) What assumptions go into the inference in the previous part? Check these assumptions using the data.

- Assumptions
  - Independence of observations within each sample: people's heart rates are generally unaffected by each other
  - Independence between each sample
  - Observations are distributed normally

```r
hist(survey$heartrate[survey$coffee=="Yes"], breaks = 20)
```




```r
hist(survey$heartrate[survey$coffee=="No"], breaks = 20)
```



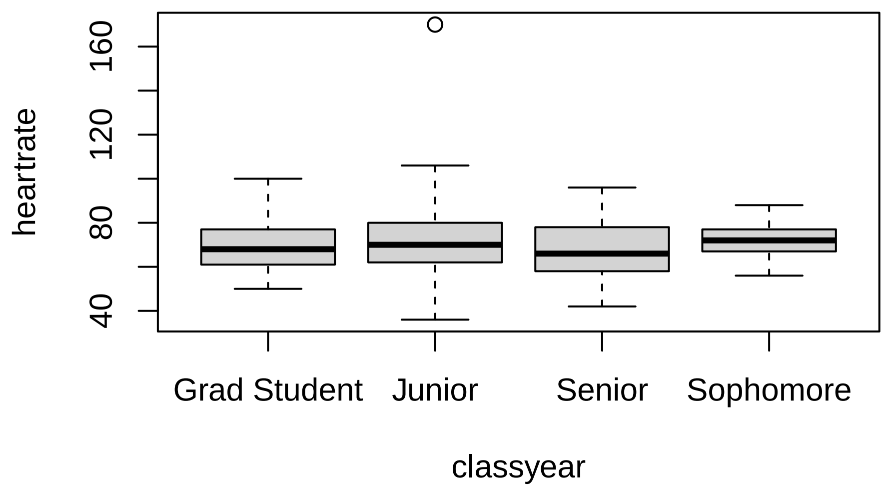
d) What assumptions go into the inference in the previous part? Check these assumptions using the data.

- ANOVA
- Assumptions
  - Independence of observations within each sample: people's heart rates are generally unaffected by each other
  - Independence between each sample
  - Observations are distributed normally
  - Constant variance between each sample

```
# run anova test
anova.1d <- aov(heartrate ~ classyear, data = survey)
summary(anova.1d)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## classyear     3    530   176.7   0.962  0.412
## Residuals   172  31570   183.6
```

```
# use boxplot to check for normality of within group distribution and check if
# the spreads are similar
boxplot(heartrate ~ classyear, data = survey)
```



e) Is the rate of coffee drinking different for grad students and undergrad students? Perform a formal hypothesis test to answer this question and provide a confidence interval to estimate the true difference.

```r
# two sample prop test
prop.test(table(survey$classyear=="Grad Student", survey$coffee))
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  table(survey$classyear == "Grad Student", survey$coffee)
## X-squared = 2.1494, df = 1, p-value = 0.1426
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.0476215  0.3299744
## sample estimates:
##    prop 1    prop 2
## 0.6911765 0.5500000
```

f) What assumptions go into the inference in the previous part? Check these assumptions using the data.

- Assumptions
  - Normal approximation to the underlying binomial distribution holds: three has to be enough successes and failures

Using `sample()`, `set.seed()` and `for` loops

Probabilities for events can be calculated (or really, estimated) via simulation by simply repeating an experiment a large number of times and counting the number of times the event of interest occurs. According to the Law of Large Numbers, as the number of repetitions increase, the proportion $\hat{p}_n$ of occurrences converge to the probability p of that event.

Problem 2: Basic Simulation

Suppose that a biased coin is tossed 5 times; the coin is weighted such that the probability of obtaining a heads is 0.6.

a) Calculate the probability of obtaining exactly 3 heads by hand (you can use `R` as a calculator)?

Your Answer Here

The following code illustrates the use of `sample()` to simulate the results for one set of 5 coin tosses.

```
#define parameters
prob.heads =
number.tosses =

#simulate the coin tosses
outcomes = sample(c(0, 1), size = number.tosses,
                  prob = c(1 - prob.heads, prob.heads), replace = TRUE)

#view the results
table(outcomes)

#store the results as a single number
total.heads = sum(outcomes)
total.heads
```

b) Using the information given about the experiment, set the parameters for `prob.heads` and `number.tosses` and run the code chunk.

i. To generate `outcomes`, the `sample()` command draws from the values `0` and `1` with probabilites corresponding to those specified by the argument `prob`. Which number corresponds to heads, and which corresponds to tails?

ii. Why is it necessary to specify `replace = TRUE`?

c) The following code uses a `for` loop to repeat (i.e., replicate) the experiment and record the results of each replicate. The term `k` is an index, used to keep track of each iteration of the loop; think of it as similar to the index of summation k (or i) in sigma notation ($\sum_{k=1}^{n}$).

The value `num.replicates` is set to 200, specifying that the experiment is to be repeated 50 times.

The command `set.seed()` is used to draw a reproducible random sample; re-running the code with the same seed (2020) will produce the same set of outcomes.

```
#define parameters
prob.heads = 0.6
number.tosses = 5
number.replicates = 200

#create empty vector to store outcomes
outcomes = vector("numeric", number.replicates)

#set the seed for a pseudo-random sample
set.seed(139)

#simulate the coin tosses
for(k in 1:number.replicates){

  outcomes.replicate = sample(c(0, 1), size = number.tosses,
                       prob = c(1 - prob.heads, prob.heads), replace = TRUE)

  outcomes[k] = sum(outcomes.replicate)
}

#view the results
outcomes
addmargins(table(outcomes))

heads.3 = (outcomes == 3)
table(heads.3)
```

d) Run the chunk. How many heads were observed in the fourth replicate of the experiment? Hint: look at `outcomes`. From the simulation results, calculate an estimate of the probability of observing exactly 3 heads when the biased coin is tossed 5 times.

e) Modify the simulation to estimate the probability of observing at most 4 heads when the biased coin is tossed 10 times. Use as many replicates as needed for a stable estimate.

f) Describe a more computationally efficient way to carry out the coin tossing simulations in this problem. Specifically, write a simulation that answers part (d) without using a `for` loop.

Problem 3: Using `if` statements

A bag contains 3 red and 3 white balls. Two balls are drawn from the bag, one at a time; the first ball is not replaced before the second ball is drawn.

a) What is the probability of drawing a white ball on the first pick and a red on the second?

Run the following code to simulate the results for 20 sets of two draws from the bag, where red and white balls are represented by `R` and `W`, respectively.

```
#define parameters
balls = rep(c("R", "W"), c(3,3))
number.draws = 2
replicates = 20

set.seed(139) #reset the seed

#create empty vector to store results
successes = vector("numeric", replicates)

#simulate the draws
for(k in 1:replicates){
  draw = sample(balls, size = number.draws, replace = FALSE)

  if(draw[1] == "W" & draw[2] == "R"){
    successes[k] = 1
  }
}

#view the results
successes
table(successes)
```

b) Explain the line of code used to generate `draw`.

An `if` statement has the basic structure `if( condition ) { statement }` ; if the condition is satisfied, then the statement will be carried out. The `if` statement in the loop records when a "success" occurs; if a particular replicate k is considered a success, then a `1` is recorded as the k^{th} element of the vector `successes`.

c) Examine the condition in the `if` statement and explain how the condition specifies when a success occurs. What is considered a success, in the context of this problem?

d) Set the number of replicates to 10,000 and re-run the simulation. What is the estimated probability of drawing a white ball on the first pick and a red on the second?

e) Using simulation, estimate the probability of drawing exactly one red ball. (Hint: The logical operator for "or" is the `|` symbol. Alternatively, think about using the `sum()` function.)

Prob 4: Simulating the Central Limit Theorem

The Youth Risk Behavioral Surveillance System (YRBSS) is a yearly survey conducted by the US Centers for Disease Control to measure health-related activity in high-school aged youth. The dataset contains responses from the 13,583 participants in 2013.

Suppose the individuals in `yrbss` are treated as a target population; the goal of the simulation is to visualize the sampling distribution of point estimates of mean weight, $\overline{x}_{weight}$.

The following code takes a random sample of 10 individuals from `yrbss` and stores the subset as `yrbss.sample`.

```
#load the data
yrbss = read.csv("data/yrbss.csv")

#set parameters
sample.size = 10

#obtain random sample of row numbers
set.seed(139)
sample.rows = sample(1:nrow(yrbss), sample.size)

#create yrbss.sample
yrbss.sample = yrbss[sample.rows, ]
mean(yrbss.sample$weight, na.rm=T)
```

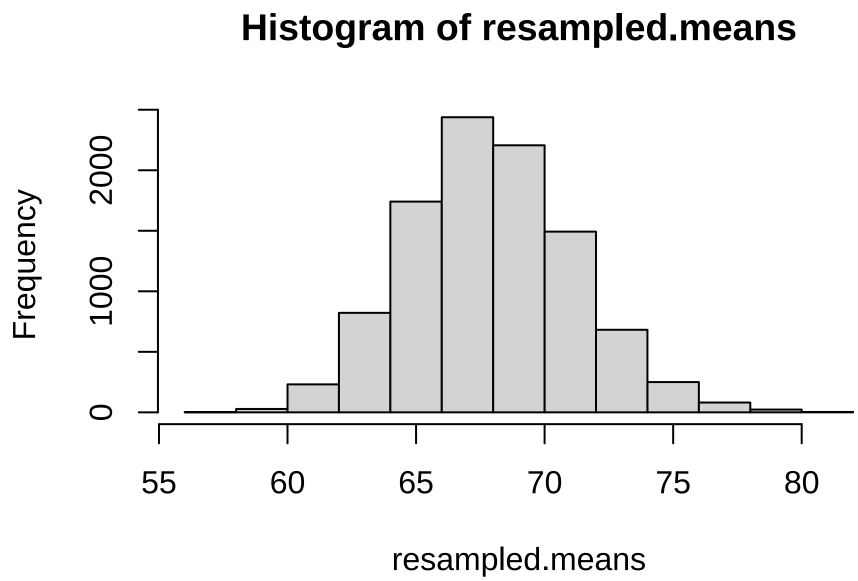
```
## [1] 80.51375
```

```
table(is.na(yrbss.sample$weight))
```

```
##
## FALSE  TRUE
##     8     2
```

Based on the code, write a simulation to take 1,000 random samples of size 10 from `yrbss` and calculate the sample mean of each sample. Afterwards, plot a histogram of the sample means. Draw a blue line at the mean of sample means and a red line at the mean in `yrbss` (which is being treated as the population mean weight).

```
#set parameters
nsims <- 10^4
sample.size <- 30

#set seed
set.seed(139)

#create empty vector to store results
resampled.means <- rep(NA, nsims)

#calculate sample means
for(llama in 1:nsims){

  sample.rows = sample(1:nrow(yrbss), sample.size)

  #create yrbss.sample
  yrbss.sample = yrbss[sample.rows, ]
  resampled.means[llama] = mean(yrbss.sample$weight, na.rm=T)
}

#create histogram of sample means
hist(resampled.means)
```




```
# hist(yrbss$weight)

#draw a blue line at the mean of sample means

#draw a red line at the population mean weight in yrbss
```

a) Explore the effect of larger sample sizes by re-running the code for sample sizes of 25, 100, and 300. How does the distribution of sample means change as sample size increases? (Hint: Use the argument `xlim = c(lb,ub)` in `hist()` to keep the axis scale fixed.)

b) With the goal of making inference about a population mean in mind, what is the advantage of a larger sample size?

Problem 5: Probability distributions in R

Detailed instructions for using the R functions for probability distributions are provided in the reference supplement, along with several examples.

Let $X_1, X_2, \ldots, X_{15}$ be i.i.d. Normal r.v.s. with mean $\mu = 1$ and variance $\sigma^2 = 3^2 = 9$. Let S^2 be the usual variance estimate: $S^2 = \sum(X_i - \bar{X})^2/(n-1)$, and let $\hat{\sigma}^2$ be the estimate using μ in the calculation instead: $\hat{\sigma}^2 = \sum(X_i - \mu)^2/n$. Write a simulation in R, using a `for` loop based on at least 10,000 iterations, to determine the following:

a) That both estimators (S^2 and $\hat{\sigma}^2$) are unbiased.

```
set.seed(139)
nsims=20000
mu=1
sigma=3
n=15
sigma2.hat=s2=rep(NA,nsims)

for(i in 1:nsims){
  # your code here: the function `rnorm` is needed
}

# your code here: determine empirical bias
```

b) Provide a separate histogram for each of the two sampling distributions. Which has lower spread?

```
# your code here
```

c) Which estimator is closer to the true value more often.

```
# your code here
```

d) Are you sure your answers above are correct? What could you do to be more certain?

e) Recall that the sampling distribution of S^2 is just a scaled χ^2_{n-1} (by a factor of $\sigma^2/(n-1)$. Show that the quantiles of a χ^2_{n-1} distribution (using `qchisq(ppoints(nsims),df)`) match the empirical quantiles of our observed S^2 using a quantile-quantile plot (`qqplot`). Interpret this plot (this reference guide might help).

```
# your code here
```