Game Day

Contents

Introduction

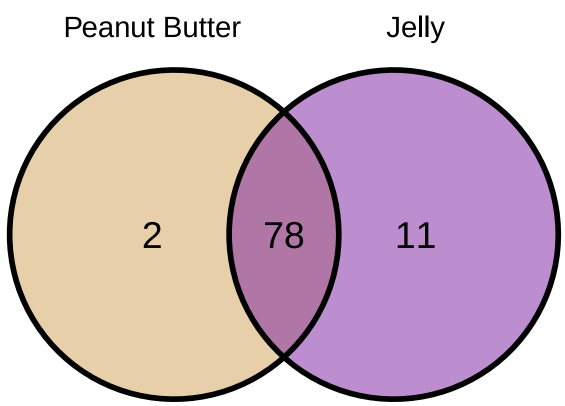
In addition to functions already included in R, below contains the code you will need to load to be prepared to answer questions for Jeopardy

```r
library(dplyr)
library(ggplot2)

theme_set(theme_bw(base_size = 16))

## Bootstrap Function
bootstrap <- function(x, statistic, n = 1000L) {
  bs <- replicate(n, {
    sb <- sample(x, replace = TRUE)
    statistic(sb)
  })
  data.frame(Sample = seq_len(n),
             Statistic = bs)
}

se <- function(x) sd(x) / sqrt(length(x))

source("https://collinn.github.io/f24/labs/clt_lab_functions.R")

## College data
college <- read.csv("https://collinn.github.io/data/college2019.csv")

## Hawks data
hawks <- read.csv("https://collinn.github.io/data/hawks.csv")

## Fix penguin data
penguins <- read.csv("https://collinn.github.io/data/penguins.csv")
penguins <- filter(penguins, !is.na(bill_length_mm))
```

Solutions

Probability

Probability 1



Probability 2

$$\left(\frac{3}{12}\right)\left(\frac{4}{11}\right)\left(\frac{5}{10}\right) = 0.045$$

Probability 3

1. No

$$0.15 = P(A \text{ and } B) = P(A|B) \times P(B) \neq P(A)P(B) = 0.2 \times 0.8 = 0.16$$

2.

$$0.15 = P(A \text{ and } B) = P(A|B) \times P(B) = P(A|B) \times 0.8 \quad \Rightarrow \quad P(A|B) = 0.1875$$

Probability 4

$P(Box) = 0.8$, $P(Pass|Box) = 0.86$ $P(Pass|\text{No Box}) = 0.65$

So

$$
\begin{aligned}
P(Box|Pass) &= \frac{P(Pass|Box) \times P(Box)}{P(Pass)} \\
&= \frac{P(Pass|Box) \times P(Box)}{P(Pass|Box)P(Box) + P(Pass|NoBox)P(NoBox)} \\
&= \frac{(0.86)(0.8)}{(0.86)(0.8) + (0.65)(0.2)} \\
&= 0.841
\end{aligned}
$$

Probability 5

$$
\begin{aligned}
P(M|V) &= \frac{P(V|M)P(M)}{P(V)} \\
&= \frac{P(V|M)P(M)}{P(V|M)P(M) + P(V|I)P(I)} \\
&= 0.7959
\end{aligned}
$$

Sampling Distributions

Sampling Dist 1

Normal:

- Mean
- Standard Deviation/Standard Error

t-distribution

- Degrees of freedom (n-1)

Sampling Dist 2

Sample size and population standard deviation (σ)

Sampling Dist 3

1. You would need a large n
2. You can bootstrap and see if it looks normally distributed

Sampling Dist 4

```r
## Collect 100,000
samp <- rt(n = 1e5, df = 15)

## Proportion absolute greater than 1
mean(abs(samp) > 1) # 0.33
```

```
## [1] 0.33221
```

```r
mean(abs(samp) <= 1) # 0.66
```

```
## [1] 0.66779
```

```r
## Finding the middle 50% tells you most likely range
## Since middle 50 so far less than 1, this is best bet since over
## 50% will be less than 1
qt(c(0.25, 0.75), df = 15)
```

```
## [1] -0.6912  0.6912
```

Sampling Dist 5

```r
peng <- filter(penguins, species == "Adelie")

## Have to bootstrap to get sampling distribution
median_boot <- bootstrap(peng$flipper_length_mm, median)

## Standard deviation of samp dist is std. error
sd(median_boot$Statistic)
```

```
## [1] 0.382
```

Confidence Intervals

Conf Int 1

A has more variability – you can tell by the distance of the points from the mean (black line). The bars are shorter than in B, but this is because B is using a larger critical value, as is indicated by the fact that it has much higher proportion of coverage

Conf Int 2

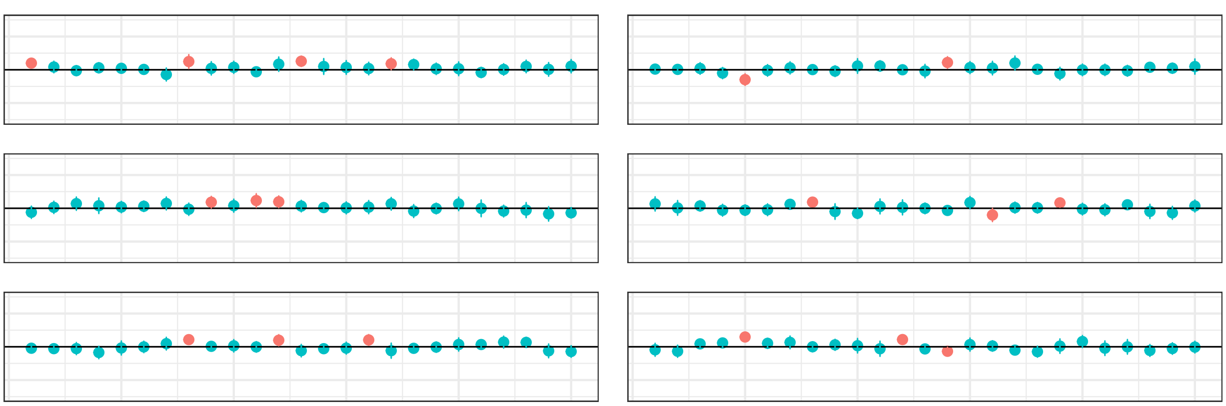
This was just random, but one way to play the odds is to determine that < 4 corresponds to a failure rate of $3/25 = 0.12$ or lower. Finding quantiles, we see

```r
qnorm(c(0.06, 0.94))
```

```
## [1] -1.5548  1.5548
```

Since 1.6 is *wider* than this by just a tiny bit, the odds are slightly better if you choose < 4

```r
## Simulate doing this 6 times
p <- lapply(1:6, function(x) simulateConfInt(m = 1.6))
do.call(gridExtra::grid.arrange, p)
```



Conf Int 3

```r
## Original
qt(0.995, df = 38) * 25 / sqrt(100)
```

```
## [1] 6.7789
```

```r
qt(0.995, df = 38) * 25 / sqrt(150)
```

```
## [1] 5.5349
```

```r
qt(0.995, df = 38) * 20 / sqrt(100)
```

```
## [1] 5.4231
```

```r
qt(0.975, df = 38) * 25 / sqrt(100) # largest change
```

```
## [1] 5.061
```

Conf Int 4

```r
qnorm(c(0.15, 0.85), mean = 75, sd = 10)
```

```
## [1] 64.636 85.364
```

```r
## Could also use critical values and Point estimate +/- method
qnorm(c(0.15, 0.85))
```

```
## [1] -1.0364  1.0364
```

Conf Int 5

```r
(cv <- qnorm(c(0.05, 0.95)))
```

```
## [1] -1.6449  1.6449
```

```r
## Does not contain 20
22.5 + cv * (6.4 / sqrt(25))
```

```
## [1] 20.395 24.605
```

General Topics

General 1

Since this is just the t statistic, we can compare it directly to critical values

```r
## This would contain our t-statistic
qt(c(0.025, 0.975), df = 14)
```

```
## [1] -2.1448  2.1448
```

General 2

```r
group_by(hawks, Species) %>%
  summarize(mean(Wing))
```

```
## # A tibble: 3 x 2
##   Species `mean(Wing)`
##   <chr>          <dbl>
```

```
## 1 CH               244.
## 2 RT               384.
## 3 SS               185.
```

General 3

This calculation can be easily performed by hand

```r
## Generate Sample
p <- c(rep(1, 24), rep(0, 16))

## Compute standard error
(vv <- se(p))
```

```
## [1] 0.078446
```

```r
## t dist with df = 39
mean(p) + qt(c(0.05, 0.95), df = 39) * vv
```

```
## [1] 0.46783 0.73217
```

General 4

```r
boot <- bootstrap(college$Adm_Rate, median)
quantile(boot$Statistic, c(0.1, 0.9))
```

```
##    10%    90%
## 0.6838 0.7000
```

General 5

```r
samp <- rbinom(100, 1, p = 0.5)
table(samp)
```

```
## samp
##  0  1
## 45 55
```