Lecture1

Uzair Mirza

06/05/2021

tidyverse Package used

Simulation Example slide 26/36

a) Topic: Height x Sprint time(100m)
b) Independent variable(x) = Height, dependent variable(y) = Sprint time(100m)

```
## set the seed to have reusable results
set.seed(6969)

## now run simulation and generate the samples
    # for both height and Sprint time(100m) we can use runif() set a min/max
    # height and Sprint time(100m) with round()
# height in format XXX.XX
height <- runif(10, 145, 185) %>% round(digits = 2)

# sprint times in format XX.XX
sprint_time <- runif(10, 20, 40) %>% round(digits = 2)

# now make a data frame of the results
data.1 <- tibble(height = height, sprint_time = sprint_time)
data.1
```

```
## # A tibble: 10 x 2
##    height sprint_time
##     <dbl>       <dbl>
##  1   169.        27.6
##  2   158.        38.0
##  3   174.        30
##  4   152.        28.5
##  5   156.        37.5
##  6   152.        32.4
##  7   146.        20.4
##  8   176.        22.8
##  9   163.        35.3
## 10   182.        36.6
```

To simulate a diff sample, change the seed or remove the set.seed() to get a new sample on each run

Breakout Group Example slide 30/36

Declare and assign individuals to one of the 2 groups. I used sample() with replacement because I made the Placebo and Treatment into a binary group and made them in a vector, to keep it fair sample() by default assigns equal prob to each outcome hence chances for choosing treatment and placebo for each member will be equal.

```
## set the seed to have reusable results
set.seed(6969)

# assign RV to the 4 ppl in the group from 1-4
group <- c(1:4)

# declaring the treatment and placebo group.
  # Treatment = 1, Placebo = 0
test.1 <- c(0,1)

# generate samples to assign to each member in the group
pre.data1 <- sample(test.1, 4, replace = TRUE)

# now make a data frame
data.2 <- tibble(Subject = group, Treatment_Recieved = pre.data1)
data.2
```

```
## # A tibble: 4 x 2
##   Subject Treatment_Recieved
##     <int>              <dbl>
## 1       1                  1
## 2       2                  1
## 3       3                  0
## 4       4                  0
```

Using runif() to generate the similar outcome. I *expect* to not get the same result as the sample() but lets see.

```
## set the seed to have reusable results
set.seed(6969)

# generate samples to assign to each member in the group.
  # here would have to use round to get the result in binary as runif() would not
  # include the extremes ie. min, max
pre.data2 <- runif(4, min = 0, max = 1) %>% round(digits = 0)

# now make a data frame
data.3 <- tibble(Subject = group, Treatment_Recieved = pre.data2)
data.3
```

```
## # A tibble: 4 x 2
##   Subject Treatment_Recieved
##     <int>              <dbl>
## 1       1                  1
## 2       2                  0
## 3       3                  1
## 4       4                  0
```

as expected we did not recieve the same result as sample() is distributed discretely with each outcome given equal prob and runif() is distributed continuously(uniform distribution).

Example slide 34/36

find sd ### R

```
ages <- c(20,21,19,20,20,20,20)
sd(ages)
```

```
## [1] 0.5773503
```

Manually $\sigma_y^2 = \frac{1}{N-1}\Sigma(y_i - \mu_y)^2$
$N = 7$
$\mu_y = \frac{1}{N}\Sigma y_i = \frac{1}{7} * 140 = 20$
$\sigma_y^2 = \frac{1}{6}\Sigma(y_i - 20)^2 = \frac{1}{6} * 2 = \frac{1}{3} = 0.3333$

find sd ### R

```
GTA <- c(1,1,1,1,0,0,0,0,0,0)
sd(GTA)
```

```
## [1] 0.5163978
```

Manually $\sigma_y^2 = \frac{N}{N-1}P(1-P)$
$N = 10, P = \frac{4}{10} \implies \sigma_y^2 = \frac{10}{9}0.4(0.6) = 0.2667$

Reason for difference in R vs Manual calculations

The reason for difference is because in R the sd() is the sample sd not the population sd.