Test1Markdown

Josh Kostak

9/26/2018

Notes

DATACAMP NOTES

Intro to R

```
class(var) -- checks the class of var

num_vec <- c(1,2,3)
names(num_vec)<-c("One", "Two", "Three") -- assigning name to elements of vector

mid <- vector[c(3,4,5)] -- makes a new vector of the the values in vector and positions 3,4, & 5

greater0 <- vector > 0 -- makes a new vector of the values in vector greater then 0

myMatrix <- matrix(1:9, byrow = TRUE, nrow = 3)
-- makes a matrix w/3rows that contain numbers 1-9 in row major order

colnames(matrix)/rownames(matrix) -- name the rows or cols in a matrix

rowSums(matrix) -- sums up rows

plus5 <- vector + 5
-- makes a vector of the values in vector plus 5 to each element

factor() -- encode vector as a factor

levels(vector)<- c("Male", "Female") -- creates levels for the vector

summary() -- gives overview of contents

str(dataframe) -- gives you structure of dataframe

dataframe[vector,TRUE] -- selects elements that are true in vector

for(var in vector) -- one way to do for loop
{
  print(var)
}

next -- shifts to the next loop iteration

for(i in 1:length(var)) -- another way to do for loop
{
  print(var[i])
}

print(paste("on row", i, "and col", j)) -- concatination
```

```
help(____) / ?_____ -- brings up info on fuctions

na.rm -- argument when true removes all empty elements

args(___) -- shows arguments of a function

my_fun <- function(arg1, arg2, ...) -- create your own function
{
  //body
  return(var)
}

my_fun <- fuction(a, b=1) -- sets the default of b to be 1
```

Intro to Data

```
dataframe$catagoryname <- droplevels(dataframe$catagoryname)
-- get rid of whole category in dataframe

email_mutated <- email %>%
    mutate(num_char_cat = ifelse(num_char < med_num_char, "below", "at or below"))
    --creates new col that uses num_char var and simplifies it into
    "below median" or "at or above median"

ggplot(data = DF, aes(x=science, y = math, color = subject)) +
  geom_point()
  --makes a scatterplot and automatically colors based on subject
```

Intro to Tidyverse

```
gapminder %>%
  filter(year == 2007, country = "United States")
  -- filter gapminder by year = 2007 and country =united states

gapminder %>%
  arrange(country) -- arrange by country

arrange(desc(country)) -- same as above but in descending order

gapminder %>%
  mutate(pop = pop/10000) -- modify existing variable

gapminder %>%
  mutate(gdp = gdppercap * pop) -- makes a new variable by multiplying two existing ones together

ggplot(gapminder, aes(x= year, y = lifeExp)) +
  geom_point() +
    scale_x_log10()    --adds log scale to a graph on the x axis
```

```
ggplot(gapminder, aes(x= year, y = lifeExp, size = pop)) +
  geom_point()    --makes size of dots represent population variable

facet_wrap(~var) -- add to end of a plot to divide data by var name and display multiple plots

gapminder %>%
  sumarize(meanLifeExp = mean(lifeExp),
            totalpop = sum(pop))      --collapses data down into these vars

gapminder %>%
  group_by(year) %>%
    sumarize(meanLifeExp = mean(lifeExp),
              totalpop = sum(pop))     --same as above but grouped by year

group_by(year, pop)   -- you can group by two groups

geom_line(), geom_bar(), geom_histogram(binwidth = 5), geom_boxplot(), geom_density()
--other plots, hist only needs x val

geom_bar(position = "dodge") -- makes side by side bar chart
```

R Markdown

```
number sign before a word makes it a heading
spat makes bullet
word surrounded by two splats makes it bold
one splt on either side makes it italics
back ticks means it code
"[word](url)" -- makes word a link to the url
dollar signs means its an equation
back tick r folowed by code and another back tick lets you insert r code
"```{r chained}" -- name the code chunck chained
"```{r ref.label = 'chained'}" -- use chained label
 Exploratory Data Analysis

levels(comics$align) -- will show different levelsof align variable

facet_grid(var ~ othervar) -- put first val in rows of grid and 2nd one in cols, add to ggplot

facet_grid(...., labeller = label_both) -- labels vars
```

CLASS NOTES

head(________) – first 6 elements

tail(________) – last 6 elements

ctrl+alt+i – inserts a code chunk

echo = FALSE – in markdown – wont display commands on page

dim(________) – dimensions of object

glimpse(_____) – view structure of object

labs(x = “”, y = “# of something”, title = “Title”, subtitle = “subtitle”) – add on end of ggplot

3 s’s – shape, center, spread

facet_grid(.~group) – another way to use facet, add onto a ggplot

CLASS ASSIGNMENT

1. In the Flight Delays Case Study in Section 1.1,

    a. The data contain flight delays for two airlines, American Airlines and United Airlines. Conduct a two-sided permutation test to see if the mean delay times between the two carriers are statistically significant.

    Null Hypothesis: $H_0 : \mu_{AA} - \mu_{UA} = 0$ Verses: $H_A : \mu_{AA} - \mu_{UA} \neq 0$

    b. The flight delays occured in May and June of 2009. Conduct a two-sided permutation test to see if the difference in mean delay times between the 2 months is statistically significant.

```
#Null Hypothesis:
#    $H_{0}: \mu_{AA}-\mu_{UA} = 0$
#    Verses:
#    $H_{A}: \mu_{AA}-\mu_{UA} \neq 0$

FD <- FlightDelays
glimpse(FD)

#find shape of data
ggplot(data=FD, aes(x=Delay)) +
    geom_histogram(color = "black", fill = "purple") +
      labs(title = "BIG TITLE")+
      facet_grid(Carrier~.)

FD %>%
    group_by(Carrier) %>%
      summarize(MeanDelay = mean(Delay), IQRDelay = IQR(Delay),
                MedianDelay = median(Delay), SDDelay = sd(Delay),
                N = n())

delays <- FD$Delay
#median(delays)
#IQR(delays)
sims <- 10^4 -1

answer <- numeric(sims)

for (i in 1:sims)
{
  #2906 is amount of AA delays and 4029 is the total number of delays
  index <- sample(4029, 2906, replace = FALSE)
  answer[i] <- mean(delays[index]) - mean(delays[-index])
}

obs <-tapply(FD$Delay, FD$Carrier, mean)
obs
obs_diff <- obs[1] - obs[2]
#obs[1] is the first carriers mean delay (AA) and obs[2] is the second one (UA)
obs_diff

pval <- (sum(answer <= obs_diff)+1)/(sims+1)
pval
```

SOLUTION:

```
FD <- FlightDelays
glimpse(FD)
FD %>%
    group_by(Month) %>%
      summarize(m = n(), MeanDelay = mean(Delay))

delays <- FD$Delay
sims <- 10^4 -1

answer <- numeric(sims)

#mixes data up
for (i in 1:sims)
{
  index <- sample(4029, 2030, replace = FALSE)
  answer[i] <- mean(delays[index]) - mean(delays[-index])
}

#applies mean of both months
obs <-tapply(FD$Delay, FD$Month, mean)
obs
#finds difference in means
obs_diff <- obs[1] - obs[2]
obs_diff

#if the val in answer is less then the diff then add one and
#divide by number of simulation element
#idk where the < comes from...
pval <- (sum(answer < obs_diff)+1)/(sims+1)
pval
```

SOLUTION:

2. In the Flight Delays Case Study in Section 1.1, the data contain flight delays for two airlines, American Airlines and United Airlines.

    a. Compute the proportion of times that each carrier's flights was delayed more than 20 minutes. Conduct a two-sided test to see if the difference in these proportions is statistically significant.

    b. Compute the variance in the flight delay lengths for each carrier. Conduct a test to see if the variance for United Airlines is greater than that of American Airlines.

        the null hypo is (signma of UA squared)/(sigma of AA) = 1

```
# a. Your code here
FD <- FlightDelays
glimpse(FD)
FD %>%
    group_by(Carrier) %>%
      summarize(m = n(), MeanDelay = mean(Delay > 20))

delays <- FD$Delay
sims <- 10^4 -1

answer <- numeric(sims)
```

```r
#mixes data up
for (i in 1:sims)
{
  index <- sample(4029, 2906, replace = FALSE)
  answer[i] <- mean(delays[index] > 20) - mean(delays[-index] > 20)
}

#applies mean of both months
obs <-tapply(FD$Delay > 20, FD$Carrier, mean)
obs
#finds difference in means
obs_diff <- obs[1] - obs[2]
obs_diff

#if the val in answer is less then the diff then add
#one and divide by number of simulation element
#idk where the < comes from...
pval <- (sum(answer < obs_diff)+1)/(sims+1)
pval
```

SOLUTION:

```r
#this is the one you must divide the stuff instead of subtract
# b. Your code here
FD <- FlightDelays
glimpse(FD)
FD %>%
    group_by(Carrier) %>%
      summarize(m = n(), VarienceDelay = var(Delay))

delays <- FD$Delay
sims <- 10^4 -1

answer <- numeric(sims)

#mixes data up
for (i in 1:sims)
{
  index <- sample(4029, 2906, replace = FALSE)
  answer[i] <- var(delays[index]) / var(delays[-index])
}

#applies mean of both months
obs <-tapply(FD$Delay, FD$Carrier, var)
obs
#finds difference in means
obs_diff <- obs[1] / obs[2]
obs_diff

#if the val in answer is less then the diff then add one and
#divide by number of simulation element
#idk where the < comes from...
pval <- (sum(answer < obs_diff)+1)/(sims+1)
pval
```

Class assignment from tuesday:

```
library(readxl)
library(dplyr)
library(ggplot2)


DF <- read_excel("TMP.xlsx")

#fix errors in the data set created by excel
DF <- DF %>%
      mutate(Age_Cohort = gsub("12-Jun", "6-12", Age_Cohort))

DF <- DF %>%
  mutate(Age_Cohort = gsub("42898", "6-12", Age_Cohort))

DF <- DF %>%
  mutate(Age_Cohort = gsub("0 - 5", "0-5", Age_Cohort))

DF


#start finding answers for the questions on the assignment
DF %>%
  filter(Gender == "Male") %>%
  summarize(MeanMaleExpenditures = mean(Expenditures))

male <- 18001

DF %>%
  filter(Ethnicity == "Hispanic") %>%
    summarize(MeanHispanicExpenditures = mean(Expenditures))

hispanic <- 11066

DF %>%
  filter(Age_Cohort == "22-50") %>%
  summarize(Mean22to50Expenditures = mean(Expenditures))

twentytwotofifty <- 40209

DF %>%
  filter(Age_Cohort == "22-50", Ethnicity == "White not Hispanic", Gender == "Male") %>%
  summarize(MeanMW22to50Expenditures = mean(Expenditures))

whiteMale22to50 <- 38604

DF %>%
  filter(Age_Cohort == "22-50", Ethnicity == "Asian") %>%
  summarize(MeanAsian22to50Expenditures = mean(Expenditures))

asian22tofifty <- 39581
```

```
#make a dataframe from the results and turn dataframe into a bar chart
bars <- data.frame(Catagory = c("Male","Hispanic","22-50", "White Male 22-50", "Asian 22-50"),
                  values = c(male, hispanic, twentytwotofifty, whiteMale22to50, asian22tofifty))

ggplot(bars, aes(x=Catagory, weight=values)) +
  geom_bar() +
  labs(x = "Catagory", y = "Mean Expenditures", title = "Average Expenditures")




#instructor example of grouping by gender and piping into ggplot
DF %>%
  group_by(Gender) %>%
  summarize(ME = mean(Expenditures), MDE = median(Expenditures), n= n()) %>%
  ggplot(aes(x = Gender, y= ME, fill = Gender)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Expenditure by Gender", y = "Mean Expenditure") +
  theme_bw() +
  scale_fill_manual(values = c("pink", "blue"))

#instructor example of grouping by ethnicity and piping into ggplot
DF %>%
  group_by(Ethnicity) %>%
  summarize(ME = mean(Expenditures), MDE = median(Expenditures), n= n()) %>%
  ggplot(aes(x = reorder(Ethnicity, ME), y = ME)) +
  geom_bar(stat="identity", fill = "red") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 50, hjust = 1)) +
  labs(x = "", y = "Mean Expenditure", title = "Average Expenditure by Ethnicity")
```