MTH142 Intro to Stats:: CHEAT SHEET

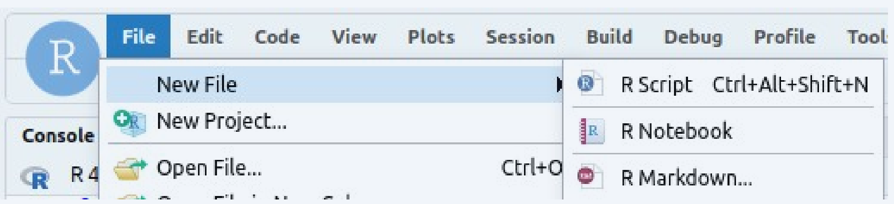
Getting started

Signing into RStudio

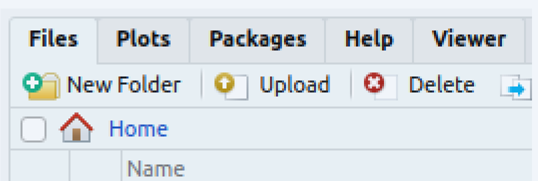
http://hcc-rstudiosrv1.hcc.edu:8787/

Username: Your HCC email with @hcc.edu
Password: Your HCC password

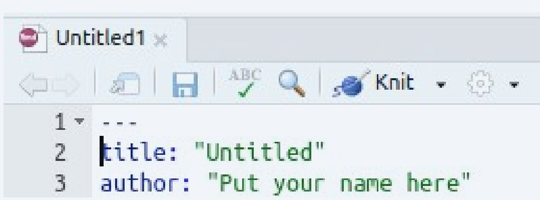
.Rmd files



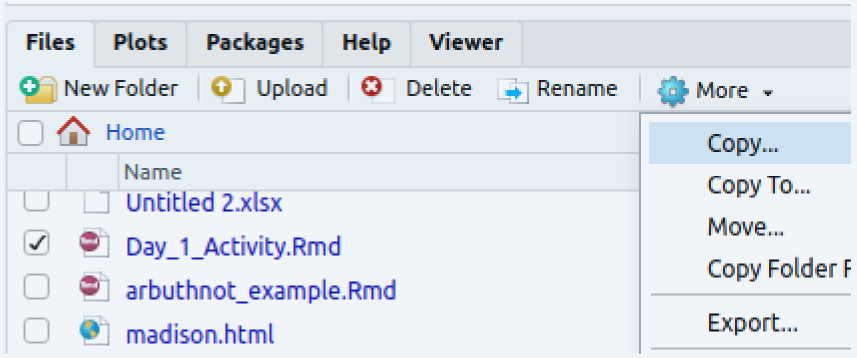
Uploading a file Lower Right quadrant click upload



Knitting a file click the knit button and save.



Exporting a file select file in lower right quadrant click **more** then **Export...**



Screen Shots

Mac: Shift + Command + 3

Windows: Shift + S

Chromebook: Shift + Ctrl +

Libraries

library(openintro)
library(tidyverse)
library(broom)

To get more info on a dataset, load a library then use ?
library(openintro)
?cars93

View Data Sets

cars93 is the data set
glimpse(data = cars93)
head(cars93)
First six rows **(**tail() works too)
view(cars93)
makes window for all data
names(cars93)
show variable names only

Summary Statistics

fivenum(cars93$price)
Just the five number summary
summary(cars93)
summarizes all the variables
tally(cars93)
counts the observations
mean(cars93$weight, na.rm = TRUE)
calculates the mean of the weights and removes missing values.
sd(cars93$weight)
calculates the standard deviation

Variables

a_variable <- c(1,2,3)
takes the vector 1,2,3 and saves it as a variable with the assignment operator

Math Type

Use **$$** to enclose math type.

μ	σ
$\bar{x}$	$\hat{p}$
H_o	α
$\ne$	$\approx.$
$\sim$	

Graphs

box plots
ggplot(data = cars93,
mapping = aes(x = weight, fill = type)) +
geom_boxplot()

histograms
ggplot(cars93, aes(x = weight)) +
geom_histogram()

bar
ggplot(cars93, aex(x= type)) +
geom_bar()

scatter plots
ggplot(data=cars93, aes(x = weight, y = mpg_city)) +
geom_point()+
geom_smooth(method = "lm", se=FALSE)

titles and labels
ggplot(cars93, aes(x= type)) +
geom_bar()+
labs(x= 'WEIGHTS', title = 'Cars from 93')

Confidence Intervals

For a mean from a vector of data
t.test(cars93$weight)
produces a 95% confidence interval for the mean weights of cars sold in 1993.

For a proportion from Statistics
prop.test(x = 20, n = 60, conf.level=0.90,correct=FALSE)
produces a 90% confidence interval for the population proportion given 20 successes in 60 trials.

For mean use formula below
For the confidence interval of the mean from statistics we use the following formula:

$$\bar{x} \pm t^* SE$$

List of Standard Error formulas

$$SE = \sqrt{\frac{p(1-p)}{n}}$$
SE for a single proportion

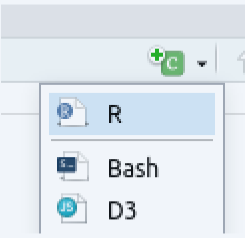
$$SE = \frac{\sigma}{\sqrt{n}} \approx \frac{s}{\sqrt{n}}$$
SE for a single mean or paired means.

$$SE = \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}$$
SE for two means (not paired)

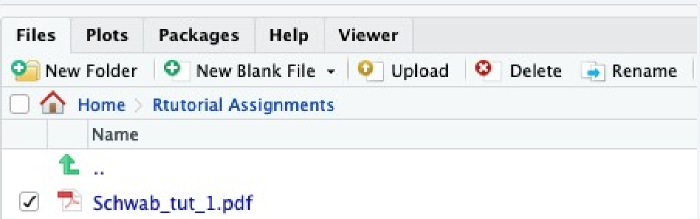
MTH142 Intro to Stats:: CHEAT SHEET

R Markdown cont.

New chunk click the green c+ and select r



Rename a file
Click the box next to the file and chose rename.



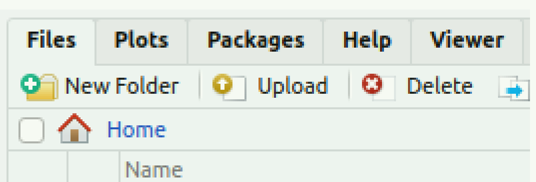
Uploading Data

1. Copy and paste data into a spreadsheet.

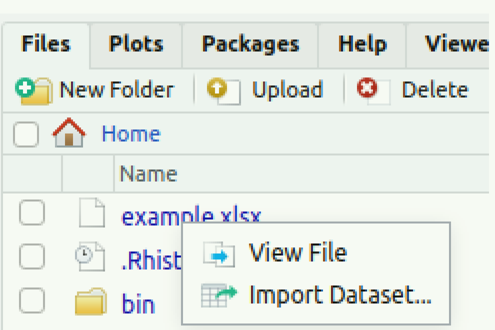
2. Variables are column headers (no spaces in variable name)

3. Save as an .xls file to your computer (no spaces in file names)

4. Upload .xls file



5. Click on the file you uploaded > Import Dataset



normal distribution

Pnorm(q= 1, mean=0 ,sd=1)
Outputs the probability if z=1 from a standard normal dist.

qnorm(p = 0.01,mean=0, sd=1)
Outputs the z-score if probability=0.01

$$z^* = \frac{\hat{p} - p}{(SE)}$$

This is the formula for a z*-score for a single proportion.

student t distribution

pt(1,df=11, lower.tail = FALSE)
outputs the p-value if t = 1 and 11 degrees of freedom on the upper tail of the distribution

qt(0.01,df=11)
outputs the t-score if probability = 0.01

$$t^* = \frac{\bar{x} - \mu}{SE}$$

This is the formula for a t*-score for a single mean.

simulation

```
library(infer)

# 1. Make a vector to mimic data

more_than_1_relationship <- c(
  rep(TRUE, 152),
  rep(FALSE, 51)
)
# 2. change the vector to a data frame

more_than_1_relationship <- as.data.frame(more_than_1_relationship)

# 3 run infer chain

set.seed(2024)

null_distn_one_prop <- more_than_1_relationship  |>
specify(response = more_than_1_relationship, success = "TRUE") |>
hypothesize(null = "point", p = 0.5) |>
generate(reps = 10000, type = "draw") |>
calculate(stat = "prop")

# 4 Take a look at the distribution

visualise(data = null_distn_one_prop)

get_p_value(x = null_distn_one_prop,
          obs_stat = 0.749,
          direction = 'greater')
```

Hypothesis Tests

__Proportions__
prop.test(x=20, n=60, p=0.5, correct=FALSE)
Outputs the results of hypothesis test with 20 successes in 60 trials

__Single Mean__
t.test(cars93$weight, mu=2000, alternative="g"**)**
Outputs of a right tailed hypothesis test with parameter mu=2000.

__Difference of Means__
t.test(time_hrs~division,data=nyc_marathon, alternative="t")
Computes the difference of men and women winning times in NYC marathon

__ANOVA__
cars_anova<-aov(weight~type,data=cars93)
performs the analysis of variance and saves it as a variable “cars_anova”
anova(cars_anova)
outputs the summary of the analysis

__Linear Regression__
cars_linear<-lm(price~weight,data=cars93)
performs the linear regression and saves it as a variable “cars_linear”
summary(cars_linear)
outputs the linear summary of the regression