Introduction to R

Day 1 –

Introduction to R

-What is R?

-Advantages of R

-Disadvantages of R

-The four windows

-multiply 2 x 2 in console

-multiply 2 x 2 in script

-Hashtags

-Install tidyverse

-Load tidyverse

**\*\*Independent exercise 1\*\***

Install and load ‘visreg’

Basic functions

-mtcars

-Calculate mean, standard deviation, n; store as objects

-Use objects to calculate SE

-Overwriting R objects

**\*\*Independent exercise 2\*\***

Calculate mean, SD, and n of mtcars$wt. Calculate SE

-Write your own R function

-Datum classes: numeric, character, date

-Create a data frame

-Vector classes: numeric, factor, logical, date

-Getting help

Import/export data

-Organizing data in spreadsheet

-Each column is a different variable

-No caps, no spaces, no symbols

-Do not mix text with numbers

-Save ‘example.xlsx’ as csv file

-Import example csv

-str()

-Export csv

**\*\*Independent exercise 3\*\***

Create csv in Excel, import csv, create object using csv, export csv

Subsetting data and operators

-brackets

-subset function with ==, <, <=, &, |

**\*\*Independent exercise 4\*\***

Remove row that is missing data for ‘letter’ from d1

Hint: Google what you don’t know!

Day 2 –

dplyr

-filter()

-arrange()

-select()

-mutate()

-pipes (%>%)

**\*\*Independent exercise 5\*\***

Create dataframe of cars with <5 gears

Sort by ascending gears

Exclude all variables but wt, qsec, and vs

Create new variable ‘weight in kg’ (hint: 2.20462 lbs/kg)

Export csv

Merge dataframes (join dataframes by a vector)

-left\_join()

-right\_join()

-inner\_join()

Join dataframes

-bind\_rows

-bind\_cols

Aggregate data

-summarize

-group\_by() with summarize()

-aggregate function

**\*\*Independent exercise 6\*\***

Calculate mean ‘disp’ in mtcars without using pipes

Calculate mean ‘disp’ in mtcars by number of gears

-multiple groups and functions

Plotting data

-plot()

**\*\*Independent exercise 7\*\***

Plot horsepower (hp) as a function of displacement (disp) in mtcars.

Make the points red.

-ggplot()

scatterplot

**\*\*Independent exercise 8\*\***

Turn points blue, reduce axis title size, remove axis title bolding, increase axis scale font

barplot

Fit some models

Model comparison

**\*\*Independent exercise 9: sort AIC results so that the smallest values are first\*\***

Effect size calculation

**\*\*Independent exercise 10, calculate the percent change associated with 'Girth'\*\***

Fancy ggplot plot

**\*\*Independent exercise 12\*\***

change the x-axis label to 'CO2 concentration', increase font size of x and y axis labels, increase font size of axis scales

# R code for Introduction to R by Bruce Hammock, UC Davis, 14, 16 Feb 2023

#open an R script

2\*2#R is a programmable calculator

#the four windows

#install.packages("tidyverse")#installs the tidyverse package

library(tidyverse)#load all packages at top of code

#Independent exercise #1 install and load the R package 'visreg'

mtcars

mtcars$mpg#the dollar sign selects a column, or 'vector' within a dataframe

#Basic functions

mean(mtcars$mpg)

mean(mtcars$mpg,na.rm=TRUE)#na.rm is called an 'argument'

m<-mean(mtcars$mpg,na.rm=TRUE)#'m' is now an 'object'

s<-sd(mtcars$mpg,na.rm=TRUE)

l<-length(mtcars$mpg)

l=length(mtcars$mpg)#'=' is equivalent to '<-'.

s/sqrt(l)#calculate se

6/sqrt(32)#trust but verify since it calls l '32L'

l<-5#overwrites what we had previously defined 'l' to be

l=5# =is equivalent to <-

#Independent exercise #2 Calculate mean, standard deviation, sample size (n), and standard error of wt in mtcars

m<-mean(mtcars$wt,na.rm=TRUE)#'m' is now an 'object'

s<-sd(mtcars$wt,na.rm=TRUE)

l<-length(mtcars$wt)

s/sqrt(l)#calculate se

#write your own function

sum.of.squares<-function(x,y,z) {x^2 + y^2 + z^2}

#defined in the environment

sum.of.squares(2,4,6)#spit out in console

ss<-sum.of.squares(2,4,6)#stored in global environment

#data types

data\_example\_one<-1

class(data\_example\_one) #'class' is used to determine data class of a vector

mean(data\_example\_one)

data\_example\_two<-"red"

class(data\_example\_two)

mean(data\_example\_two)

data\_example\_three<-"TRUE"#stores 'TRUE' as text

class(data\_example\_three)

data\_example\_four<-TRUE#stores 'TRUE' as 'logical data'

class(data\_example\_four)

data\_example\_five<-as.Date("2021/4/1")#Dates are represented as the number of days since 1970-01-01, with negative values for earlier dates.

#as.Date stores as dates rather than factor

class(data\_example\_five)

class(l)#'L' means 'integer'

#make our own data frame. This is essential, nobody should miss getting this working

id <- c(1,2,3,4,6,4,6)#vector. differs from a 'list' because it is not a mixture of data classes

#getting help

?c

color <- c("red", "white", "red", "green","blue","biege","seagreen")#vector. quotes tell R that red is text, not an R object

outcome <- c(TRUE,TRUE,TRUE,FALSE,FALSE,FALSE,TRUE)#vector

letter<- c("a","b","c","NA","d","e","f")#vector

date<-as.Date(c("2021/4/1","1986/10/2","2004/8/7","1996/6/8","1958/8/1","2017/9/10","1980/8/5"))#second parentheses runs 'as.Date' on everything within ()

d1 <- data.frame(id,color,outcome,letter,date)#dataframe, or group of vectors of the same length

str(d1)#str means structure

id<-factor(id)#change numeric variable to factor

class(id)

id<-as.numeric(id)

d1 <- data.frame(id,color,outcome,letter,date)

str(d1)

#import/export data

#organizing data in spreadsheet

#no caps, no spaces, no symbols

#do not mix text with numbers

#save 'example.xlsx' as csv file

setwd("C:/Users/bruce/Desktop/R intro/R intro class 3")#set working directory

d2<-read.csv("example.csv", check.names = FALSE) #check.names checks column name syntax

str(d2)

mean(d2$fork\_length)

write.csv("fish.csv",x=d2)#looks different now!

#Independent exercise #3 Create csv in Excel, import csv into R, create object of csv, export csv

#subsetting a vector or dataframe with []

v<-c(1,6,3,4,5,6)

v[2]

v[2:6]

d1[3,5]#intersection of third row and 5th column

d1[,3]#all rows, third column

d1[4,]#all columns, fourth row

mean(d1$id) #the same as mean(d1[,1])

mean(d1[,1])#the same as mean(d1$id)

#subsetting data and operators

d2<-subset(d1,outcome=="TRUE")#one equals is for assigning, two equals for logical testd3<-subset(d1,id<3)

d4<-subset(d1,id<=3)

d5<-subset(d1,id<3 & color =="red")

d6<-subset(d1,id<3 | color =="green")

#Independent exercise 4, remove row from d1 that is missing data for 'letter'. HINT: google what you don't know!

#d7<-subset(d1,letter!="NA")

d7

#dplyr

hum<-filter(mtcars,cyl==4)#removes cars without 4 cylinders

hum<-subset(mtcars,cyl==4)#equivalent to the line above

sorted<-arrange(hum,desc(mpg))#sort by descending mpg

three\_variables<-select(sorted,mpg,cyl,hp)#select mpg, cyl, hp

km\_l<-mutate(three\_variables,km\_per\_l=mpg\*0.425144)#add a column showing km/l

#can do it all at once, but easy to make parentheses error

km\_l<-filter(arrange(select(mutate(mtcars,km\_per\_l=mpg\*0.425144),mpg,cyl,hp,km\_per\_l),desc(mpg)),cyl==4)

km\_l\_suggestion<-mutate(select(arrange(filter(mtcars,cyl==4),desc(mpg)),mpg,cyl,hp),km\_per\_l=mpg\*0.425144)#demonstrate going through functions sequentially

#exercise #5 create dataframe of cars with less than 5 gears, sort by ascending gears, exclude

#all variables but wt, qsec, and vs, create new variable 'weight\_in\_kg' (hint: 2.20462 lbs/kg)

#export csv

d10<-filter(mtcars,gear<5)#select only gears <5

d11<-arrange(d10,gear)

d12<-select(d11,wt,qsec,vs)#exclude all variables but wt, qsec, and vs

d13<-mutate(d12,weight\_in\_kg=d12$wt\*2000\*0.453592)#convert tons to pounds, and then convert to kg.

write.csv("exercise\_5.csv",x=d13)

#note, can introduce error for class to find by changing this 'mutate(d12,weight\_in\_kg=d12$wt' to mutate(d12,weight\_in\_kg=d10$wt

#pipes

mtcars %>%

filter(cyl==4) %>%

arrange(desc(mpg)) %>%

select(mpg,cyl,hp) %>%

mutate(km\_per\_l=mpg\*0.425144)

#make new data frame

id <- c(1,4,3,2,5,10,2)

plant <- c("grass", "redwood", "rose", "pea","maple","cactus","tulip")

d2 <- data.frame(id,plant)

#Merge dataframes (merge dataframes by vector)

d1

left\_join(d1,d2)#merges d1 and d2 by left column (like vlookup in excel). d1 rows are kept if incomplete rows exist

right\_join(d1,d2)#d2 rows are kept if incomplete rows exist

inner\_join(d1,d2)#only complete rows kept

#join dataframes

bind\_rows(d1,d2)#stacks the two dataframes together, rather than merging them

bind\_cols(d1,d2)#joins the two dataframes next to one another

naming\_example<-bind\_cols(d1,d2)#joint d1 and d2 beside one another

colnames(naming\_example)<-c("id\_1","color","outcome","letter","date","id\_2","plant")#renaming columns

#aggregate data

mtcars %>%

summarize(mean(mpg))

mtcars %>%

summarize(n())

mtcars %>% #another option

nrow()# equivalent to nrow(mtcars)

mtcars %>% #number of columns, not rows. excludes the 'cars' column because it lacks a header

length()

#group\_by combined with summarize

mtcars %>%

group\_by(cyl) %>%

summarize(mean(mpg))

#Independent exercise 6

# Calculate mean 'disp' in mtcars without using pipes

#Calculate mean 'disp' in mtcars by number of gears

mean(mtcars$disp)

mtcars %>%

group\_by(gear) %>%

summarize(mean(disp))

#group\_by combined with summarize:multiple variables, functions

mtcars %>%

group\_by(cyl,gear) %>%

summarize(mean(mpg),max(qsec),sd(hp))#add n()

#storing results and exporting as csv

result<-mtcars %>%

group\_by(cyl,gear) %>%

summarize(mean(mpg),max(qsec),sd(hp))

write.csv("result.csv",x=result)

getwd()#get working directory

#aggregate function

aggregate(mpg~gear+cyl,data=mtcars,FUN=mean)#my go to option. other functions work, like sd, min, max

# the squiggle means 'as a function of'

#Plotting data

#Base plot function

plot(qsec~hp,data=mtcars)

#Base plot function: endless arguments

plot(qsec~hp,data=mtcars,xlab="Horsepower",ylab="Seconds per quarter mile",pch=2)#relabel x and y axes, pch dictates point shape

plot(qsec~hp,data=mtcars,xlab="Horsepower",ylab="Seconds per quarter mile",pch=2,ylim=c(10,25))#set y-axis min and max

plot(qsec~hp,data=mtcars,xlab="Horsepower",ylab="Seconds per quarter mile",pch=mtcars$cyl)#point shape according to cylinder

#show that \* = 8, triangles = 6, and x = 4

#show how to export plot

#Independent exercise 7: plot hp as a function of disp in mtcars, make the points red

plot(hp~disp,data=mtcars,col="red")

#ggplot2

#simple scatterplot

ggplot(data=BOD, aes(x=Time,y=demand))+#creates the canvas, or the top layer

geom\_point()+#specifies scatterplot

geom\_line()#adds a line

#try colour="green", size = 4

#scatterplot with points categorized by cyl

df<-mtcars

df$cyl <- as.factor(df$cyl)

str(df)

ggplot(data=df, aes(x = hp, y = mpg, shape=cyl)) +

geom\_point(size=3)+#defines type of graph (e.g., scatter, line, bar)

xlab("Horsepower")+

ylab("Miles per gallon")+

theme(axis.title = element\_text(size=14, face="bold"))

#Independent exercise #8

#Turn points blue, reduce axis title size, remove axis title bolding, increase axis scale font

#ggplot2

df<-mtcars

df$cyl <- as.factor(df$cyl)

str(df)

ggplot(df, aes(x = hp, y = mpg, shape=cyl)) +

geom\_point(size=3,colour="blue")+#here we've turned the points blue

xlab("Horsepower")+

ylab("Miles per gallon")+

theme(axis.title = element\_text(size=10))+

theme(axis.text = element\_text(size = 20))

#barplot

make<-c("Mazda", "Mazda","Datsun","Hornet","Hornet","Valiant","Duster","Merc")

hp<-c(110, 110, 93, 110, 175, 105,62,95)

new\_df<- data.frame(make,hp)

ggplot(data=new\_df, aes(x = make, y = hp)) +

geom\_col(alpha=0.5)+#defines type of graph (e.g., scatter, line, bar), alpha = 0.5 makes the bars transparent

xlab("")+

ylab("Horsepower")+

theme(axis.title = element\_text(size=14))+

theme(axis.text = element\_text(size = 14))

#fit some models!

trees

plot(Height~Volume,data=trees)

plot(Height~Girth,data=trees)

plot(Girth~Volume,data=trees)

pairs(trees)

m1<-lm(Height~1,data=trees)#mean(trees$Height) is a check of y-intercept

m2<-lm(Height~Volume,data=trees)

m3<-lm(Height~Girth,data=trees)

m4<-lm(Height~Volume+Girth,data=trees)

AIC(m1,m2,m3,m4)

#independent exercise 9: sort AIC results so that the smallest values are first

to\_be\_sorted<-AIC(m1,m2,m3,m4)

arrange(to\_be\_sorted,AIC)

#or alternatively...

AIC(m1,m2,m3,m4) %>%

arrange(AIC)

summary(m4)#y intercept, slope, p-value

visreg(m2)#view model plotted over data

visreg(m4)#view partial residuals

plot(m2)#diagnostic plots

plot(m4)

anova(m4)

#effect size calculations using m4

#height~83.2958+0.5756\*Vol - 1.8615\*Girth the model

predicted\_ht\_Vol\_min<-83.2958+0.5756\*min(trees$Volume) - 1.8615\*mean(trees$Girth)#model predicted height at min volume

predicted\_ht\_Vol\_max<-83.2958+0.5756\*max(trees$Volume) - 1.8615\*mean(trees$Girth)#model predicted height at max volume

(predicted\_ht\_Vol\_max-predicted\_ht\_Vol\_min)/predicted\_ht\_Vol\_min\*100

#Independent exercise 10, calculate the percent change associated with 'Girth'

predicted\_ht\_Girth\_min<-83.2958+0.5756\*mean(trees$Volume) - 1.8615\*min(trees$Girth)#model predicted height at min girth

predicted\_ht\_Girth\_max<-83.2958+0.5756\*mean(trees$Volume) - 1.8615\*max(trees$Girth)#model predicted height at max girth

(predicted\_ht\_Girth\_min-predicted\_ht\_Girth\_max)/predicted\_ht\_Girth\_min\*100

#Additional ggplot example, borrowed from 'R programming 101'

?CO2

ggplot(data=CO2, aes(x=conc,y=uptake,colour=Treatment))+#because colour = Treatment is on the top row, it applies to all layers underneath

geom\_point(size=4,alpha=0.5)+#alpha makes the points transparent

geom\_smooth()+#adds smoothed line

facet\_wrap(~Type)+#divides plots by a category, in this case 'Type'

theme\_bw()

#Independent exercise 12: change the x-axis label to 'CO2 concentration', increase font size of x and y axis labels, increase font size of axis scales

ggplot(data=CO2, mapping= aes(x=conc,y=uptake,colour=Treatment))+#because colour = Treatment is on the top row, it applies to all layers underneath

geom\_point(size=4,alpha=0.5)+#alpha makes the points transparent

geom\_smooth()+#adds smoothed line

facet\_wrap(~Type)+#divides plots by a category, in this case 'Type'

theme\_bw()+

xlab("CO2 concentration")+

theme(axis.title = element\_text(size=14))+

theme(axis.text = element\_text(size = 14))

