BIOSTATS 640 – Introduction to R
Fall 2023
https://people.umass.edu/biep640w/webpages/demonstrations.html

https://www.simplilearn.com/what-is-descriptive-statistics-article

02
Numerical Summarization & One and Two Sample Inference

September 15, 2023

Right click to download R dataset
sepsis.Rdata

Welcome to Lesson 02!
This lesson begins with a brief introduction to importing data, examining the structure of your data and how to work with factors in R. Tip. What we think of as categorical variables, R calls these factors. This is followed by an introduction to using R Studio for one and two sample inference. Have fun!

1. Highlights of Lesson 01 - Up and Running with R Studio

We will be doing all our work in R Studio	R Studio is an application that sits "on top" of R. R is under the hood". R Studio provides a very friendly environment for doing lots of things: writing and executing code, managing files and directories, and working with packages.
`>` **is the command prompt, located in the console pane** `#` **denotes a comment; R ignores the rest of the line**	
R is case sensitive	And is unforgiving!
Use the function `c(      )` to create vectors of data and separate arguments by commas	**For example:** `v1  <-  c(14,35,81,99)`

`dataframe`
A "dataset" (analogous to excel spreadsheet or SAS dataset or Stata dataset) is called a `dataframe` in R
	Examples:
	`Df1<-data.frame(A=c(1,2,3),B=c(2,3,4))`
	`dd<-as.data.frame(matrix(c(1,2,3,4),nrow = 2))`

R can work with more than one dataset at a time.	
To identify a variable in a dataframe, R utilizes a two-part naming convention: **dataframename$variablename**	**For example:** `arthritis$Age`

Statistical functions run on COMPLETE data only.
Tip: use option na.rm=T to remove missings (NA)
	For example:
	`mean(v1, na.rm=T)`
	`mean(v1, na.rm=TRUE)       # This also works.`

2. Set Your Working Directory

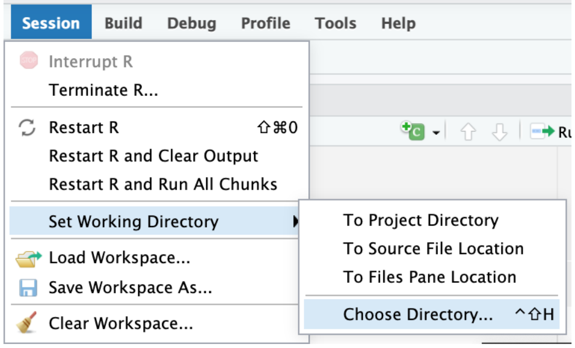
What is the working directory and why do I need to set it?

R needs to know where to find the files to **read from** and where to **write to.** This location is a directory with an associated path and is known as your **working directory**.

setwd( ) - Set your working directory

getwd( ) - Show current working directory

How to Set Your Working Directory Method I - Using the R Studio/Posit Menus
From the top menu bar, click Session > Set Working Directory > Choose Directory
Browse to navigate to your desired folder. Click **CHOOSE.**



How to Set Your Working Directory Method II - Using the `setwd( )` function in the console
IMPORTANT. The path name must be enclosed in quotes.
Example (Windows): `setwd("My Documents/BIOSTATS 640/homeworks")`
Example (Mac): `setwd("~Desktop/BIOSTATS 640/homeworks")`

How to Show Your Current Working Directory – Use the command `getwd( )`

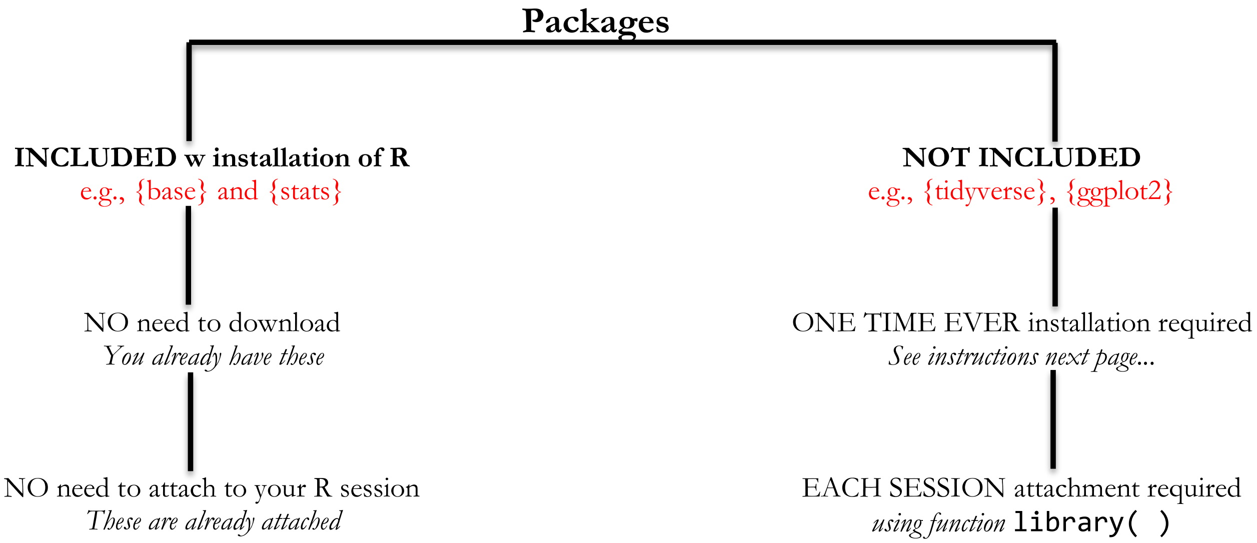
```
getwd( )                    # Yes.  The parentheses are left empty
```

#3. Good to Know!
Introduction to Packages

Why do I need to know about packages?
Your installation of R came with some pre-installed commands that are contained within pre-installed packages. Lots of them. However, very often (and I mean VERY OFTEN), you will want to make use of commands and features of packages that are not pre-installed. To do so, you need to: 1) download and install the package (one time); and 2) attach the package to your R Studio session (one time/session).

A **package** is a collection of functions and datasets

There are two types of packages, included with installation and not included with installation.



Packages

INCLUDED w installation of R
e.g., {base} and {stats}

NO need to download
You already have these

NO need to attach to your R session
These are already attached

NOT INCLUDED
e.g., {tidyverse}, {ggplot2}

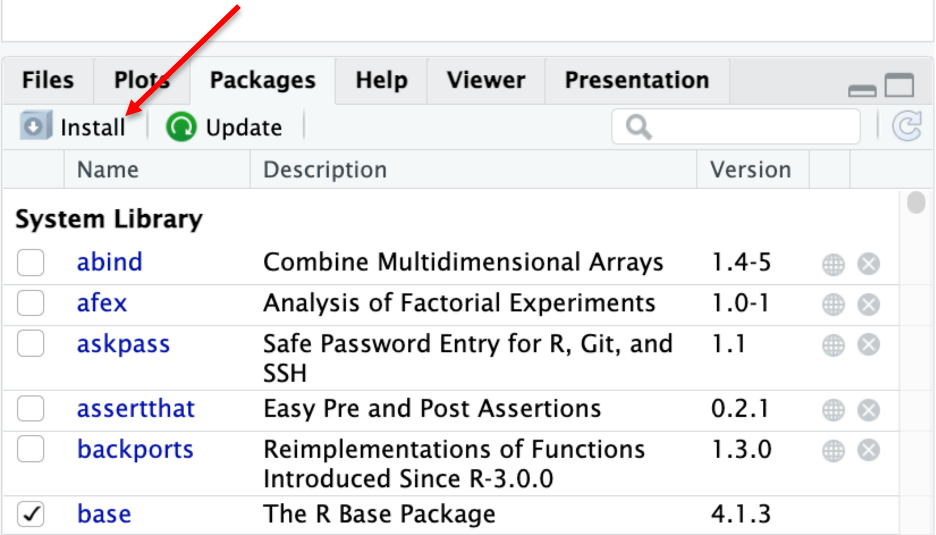
ONE TIME EVER installation required
See instructions next page...

EACH SESSION attachment required
using function `library( )`

How to Install a Package Using R Studio Menus (recommended)

Example: `{ swirl }`

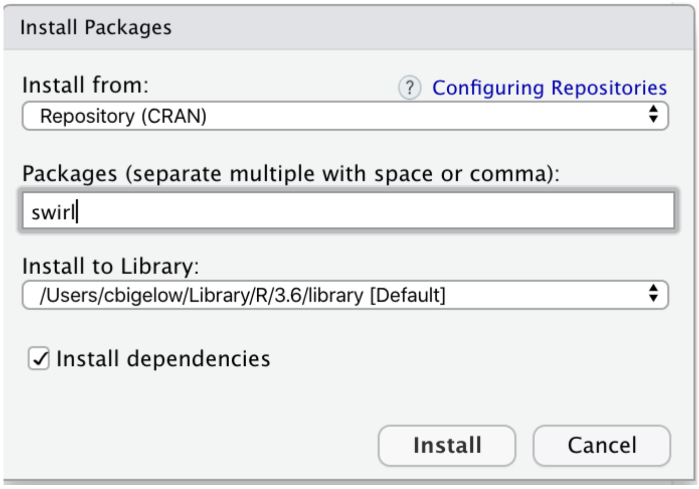
From the **FILES/PLOTS/PACKAGES** pane choose **PACKAGES** and click **Install**



System Library

	Name	Description	Version
□	abind	Combine Multidimensional Arrays	1.4-5
□	afex	Analysis of Factorial Experiments	1.0-1
□	askpass	Safe Password Entry for R, Git, and SSH	1.1
□	assertthat	Easy Pre and Post Assertions	0.2.1
□	backports	Reimplementations of Functions Introduced Since R-3.0.0	1.3.0
☑	base	The R Base Package	4.1.3

Key:


Install Packages
Install from: Repository (CRAN)
Packages (separate multiple with space or comma): swirl
Install to Library: /Users/cbigelow/Library/R/3.6/library [Default]
☑ Install dependencies
 	
	`In Install from:` **`default (Repository CRAN) is fine`**
	`In Packages (separate multiple with space or comma:)` **`swirl`**
	`In Install to Library:` **`leave as is`**
	`Check box for "Install dependencies":` **`check`**
	`At bottom, click` **`Install`**

** Note: Rtools (should be installed before installing some packages; after installing Rtools, you may need to do some extra settings, please GOOGLE '**how to install rtools**' to check the steps)

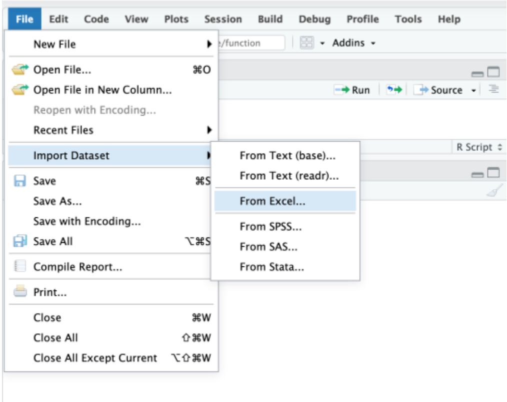
#4. How to Import an Excel Dataset into R

Preliminaries (Important):

(1) Make sure that you have downloaded from the course website the dataset arthritis.xlsx.
(2) Strongly encouraged: *(Source: marinstats lectures)* Importing Excel Data into R (video, 8:12)

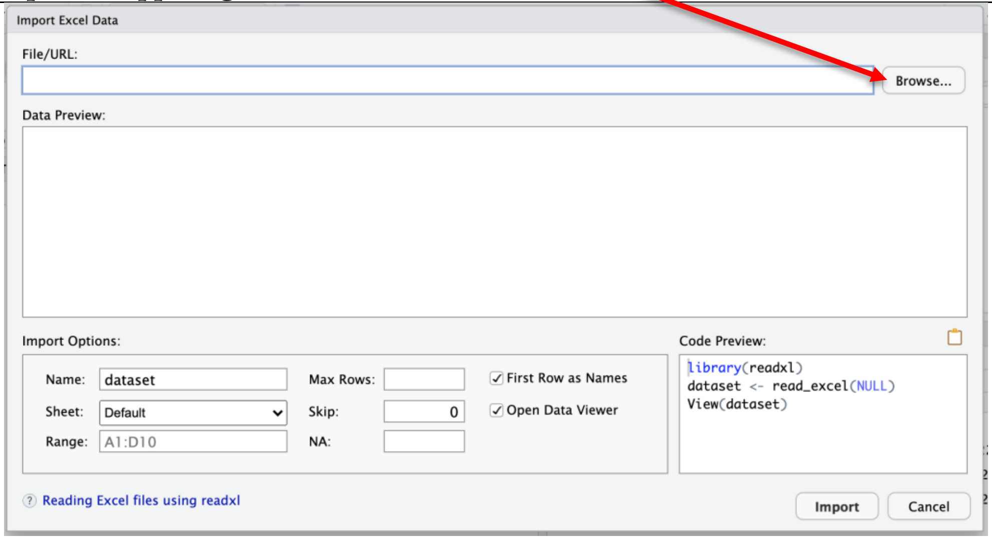
How to Import Excel Data Using R Studio Menus

Step 1: At upper left; FILE > IMPORT DATASET > FROM EXCEL



Note: R may return a message saying that you need to install readxl. Click YES. Then wait until you get a prompt.

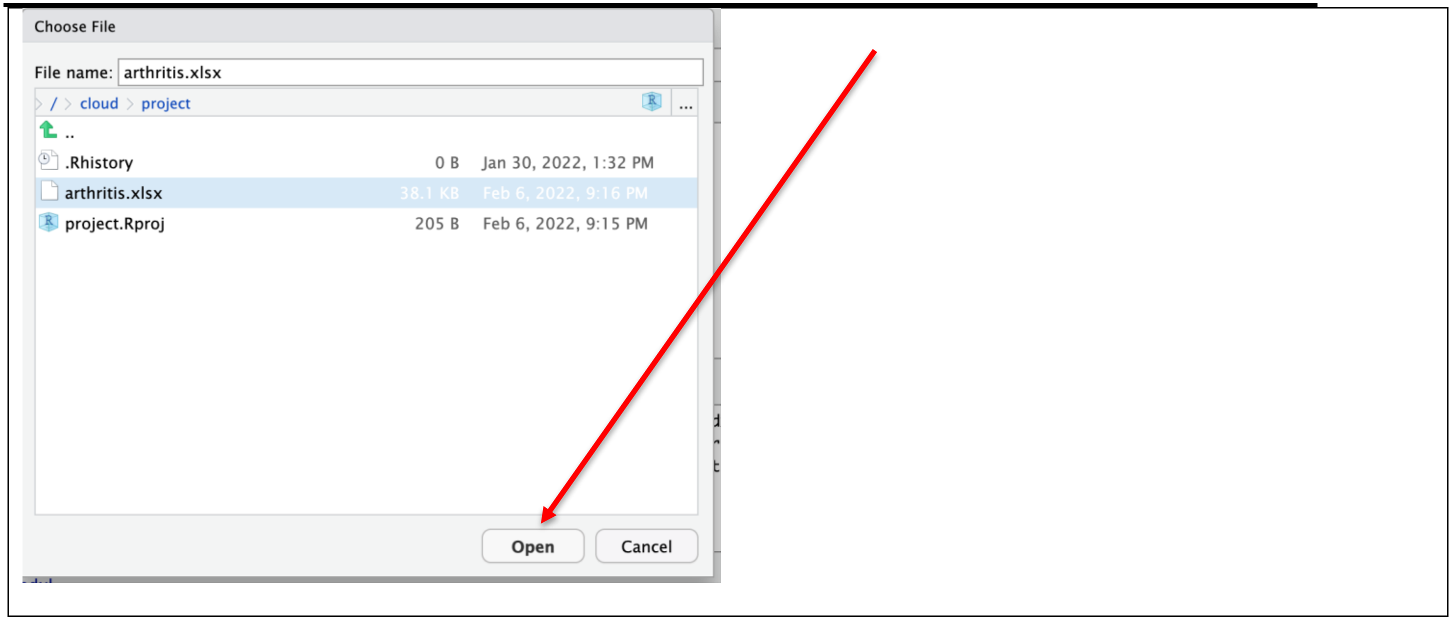
Step 2: At upper right, click on the icon BROWSE.



Name: dataset
Max Rows:
☑ First Row as Names
Sheet: Default
Skip: 0
☑ Open Data Viewer
Range: A1:D10
NA:

```
library(readxl)
dataset <- read_excel(NULL)
View(dataset)
```

Step 3: Navigate to choose arthritis.xlsx. At lower right, click OPEN

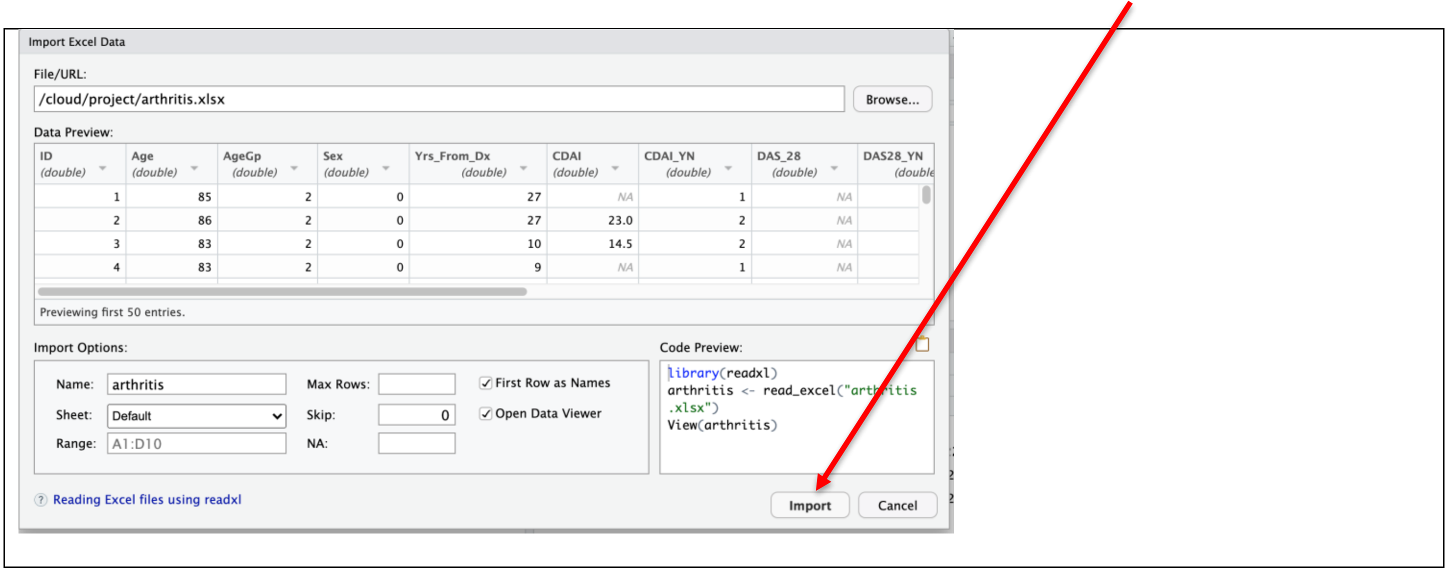


File name: arthritis.xlsx

/ > cloud > project

..		
.Rhistory	0 B	Jan 30, 2022, 1:32 PM
arthritis.xlsx	38.1 KB	Feb 6, 2022, 9:16 PM
project.Rproj	205 B	Feb 6, 2022, 9:15 PM

Step 4: Take your time here in making your selections. All set? At lower right, click IMPORT



File/URL: /cloud/project/arthritis.xlsx

Data Preview:

ID (double)	Age (double)	AgeGp (double)	Sex (double)	Yrs_From_Dx (double)	CDAI (double)	CDAI_YN (double)	DAS_28 (double)	DAS28_YN (double
1	85	2	0	27	NA	1	NA	
2	86	2	0	27	23.0	2	NA	
3	83	2	0	10	14.5	2	NA	
4	83	2	0	9	NA	1	NA	

Previewing first 50 entries.

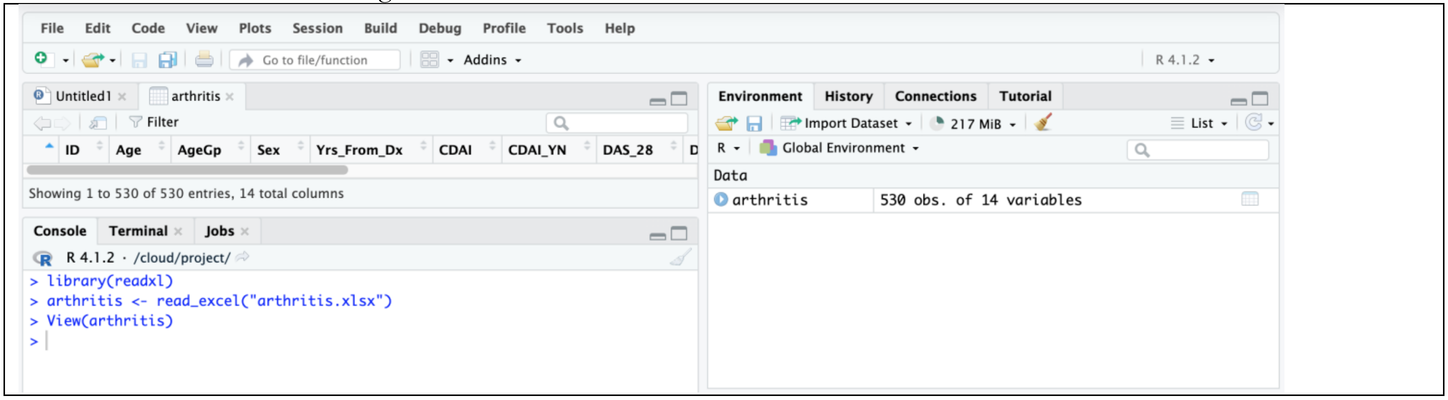
Import Options:

Name: arthritis
Sheet: Default
Range: A1:D10
Max Rows:
Skip: 0
NA:
☑ First Row as Names
☑ Open Data Viewer

Code Preview:

```
library(readxl)
arthritis <- read_excel("arthritis
.xlsx")
View(arthritis)
```

You should now see the following

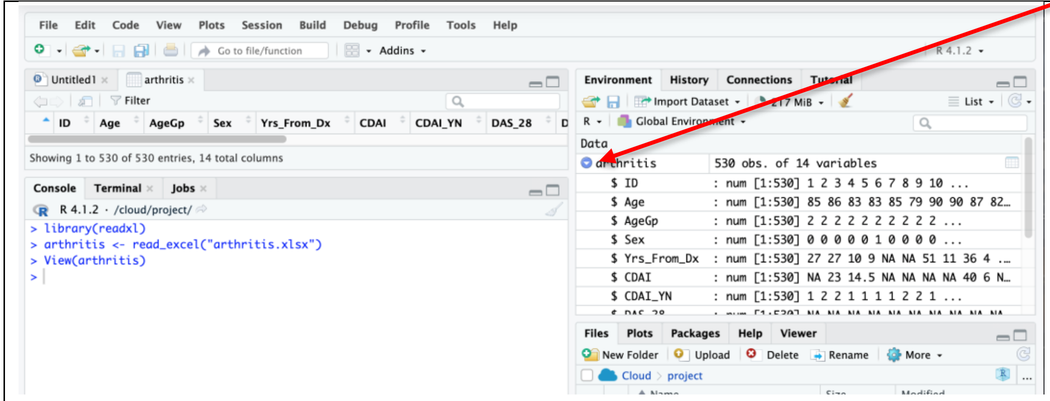


Showing 1 to 530 of 530 entries, 14 total columns

Data
arthritis 530 obs. of 14 variables

```
R 4.1.2 · /cloud/project/
> library(readxl)
> arthritis <- read_excel("arthritis.xlsx")
> View(arthritis)
>
```

Step 5: At right, in the Environment tab, click on the down arrow next to the dataset name arthritis.
You should then see the following information: sample size, # variables, data types, etc.



Showing 1 to 530 of 530 entries, 14 total columns

```
> library(readxl)
> arthritis <- read_excel("arthritis.xlsx")
> View(arthritis)
>
```

```
Data
arthritis         530 obs. of 14 variables
   $ ID          : num [1:530] 1 2 3 4 5 6 7 8 9 10 ...
   $ Age         : num [1:530] 85 86 83 83 85 79 90 90 87 82…
   $ AgeGp       : num [1:530] 2 2 2 2 2 2 2 2 2 2 ...
   $ Sex         : num [1:530] 0 0 0 0 0 1 0 0 0 0 ...
   $ Yrs_From_Dx : num [1:530] 27 27 10 9 NA NA 51 11 36 4 ..…
   $ CDAI        : num [1:530] NA 23 14.5 NA NA NA NA 40 6 N…
   $ CDAI_YN     : num [1:530] 1 2 2 1 1 1 1 2 2 1 ...
   $ DAS_28      : num [1:530] NA NA NA NA NA NA NA NA NA NA
```

#5. How to Create a Categorical Variable: Introduction to Factors in R

Exploration and analysis of categorical data in R involves working with R factor objects.

In statistical/biostatistical parlance, we talk about categorical variables. A categorical variable might be:

- Qualitative/nominal: e.g., state of residence (MA, NY, CT, etc)
- Qualitative/ordinal: e.g., level of pain ("low", "medium", "high")
- Quantitative discrete count: e.g., number of visits to the dentist (0, 1, 2, etc.)

R calls these factors. Factors are NOT THE SAME THING as R objects that are character objects. Typically (there might be exceptions), we want to use factors.

How to create a factor object (categorical variable) from a character object

```
# ----  Create a character object to play with -------------#
> character_v1 <- c("winter","winter","fall","spring", "spring", "summer", "fall")
> character_v1
[1] "winter" "winter" "fall"   "spring" "spring" "summer" "fall"

> # ---- default ordering is ALPHABETICAL ------#
> factor_v1 <- factor(character_v1)                                      # Default arrangement is ALPHABETICAL
> factor_v1
[1] winter winter fall   spring spring summer fall
Levels: fall spring summer winter

> factor_v2 <-factor(character_v1,
+                    levels=c("winter", "spring", "summer", "fall"))         # You can choose arrangement as you like
> factor_v2
[1] winter winter fall   spring spring summer fall
Levels: winter spring summer fall

> factor_v3 <-factor(character_v1,
+                    levels=c("winter", "spring", "summer", "fall"),
+                    ordered=TRUE)                                        # And you can declare as ORDINAL
> factor_v3
[1] winter winter fall   spring spring summer fall
 Levels: winter < spring < summer < fall
```

How to create a factor object (categorical variable) from a numeric object

```
> # ----  Create a numeric object to play with -------------#
> numeric_v1 <- c(1,1,2,1,1,2,2,1,1)
> numeric_v1
[1] 1 1 2 1 1 2 2 1 1
> factor_v4 <- factor(numeric_v1,
+                  levels = c(1,2),
+                  labels= c("Male","Female"))
> factor_v4
[1] Male   Male   Female Male   Male   Female Female Male   Male  
Levels: Male Female
> class(numeric_v1)
[1] "numeric"
> class(factor_v4)
[1] "factor"
```

Note: ‘as.factor()**’: as.factor coerces its argument to a factor. It is an abbreviated (sometimes faster) form of factor. **Usually, we use it to quickly generate new column in dataframe.**

6. One Sample Inference

Dataset (right click to download):

sepsis.Rdata

Packages used:
{DescTools}, {stargazer}, {summarytools} {tidyverse}

Tip for Hypothesis Testing

Alternative Hypothesis	R Code
Two sided	, alternative=”two.sided”
Right tail	, alternative=”greater”
Left tail	, alternative=”less”

Tip for Confidence Intervals

If you want …	R Code
95% CI	Nothing you need to do ... this is default
90% CI	, conf.level = .90
... and so on	, conf.level = .FILLIN

Load R dataset to session

Step 1: If you have not already done so, right click to download sepsis.Rdata from course website.

Step 2: R Studio/Posit in the Cloud Users Only) Upload sepsis.Rdata

Step 3: Put sepsis.Rdata into your working directory

Step 4: `load(file="sepsis.Rdata")`

6.1. One Sample – Continuous Outcome Normal Distribution Model

At a Glance

Numerical Summarization	summary(outcome) # Method 1
	
	library(summarytools) # Method 2
	descr(df$outcome,
	      stats=c(“n.valid”, “mean”, “sd”, “med”, “min”, “max”), # User chooses
	      transpose=TRUE)

Confidence Interval Estimation	# Confidence Interval for mean
	t.test(outcome ~ 1, data=df, conf.level=.90)$conf.int # Default is conf.level=.95
	
	# Confidence Interval for variance
	library(DescTools)
	VarTest(df$outcome, conf.level=.90)$conf.int # Default is conf.level=.95

Hypothesis Testing	# One Sample t-test of mean
	t.test(outcome ~ 1, data=df, mu=nullmean)
	
	# One Sample t-test of variance
	library(DescTools)
	VarTest(df$outcome, sigma.squared=nullvariance)

Examples.

```
# Z Test of mean:  Population variance/standard deviation are KNOWN

library(DescTools)
ZTest(sepsis$o2del,
        mu=1000,                                               # null hypothesis mean
        sd_pop=409,                                            # known population standard deviation sigma
        alternative="greater")                                 # alternative: true mean > null mean
```

```
          One Sample z-test

data:  sepsis$o2del
z = 0.75478, Std. Dev. Population = 409, p-value = 0.2252        # Null mu=1000 v mu > 1000 is NOT rejected
alternative hypothesis: true mean is greater than 1000
95 percent confidence interval:
 971.9137      Inf
sample estimates:
mean of x
 1023.817
```

```
# T-test of mean:  Population variance/standard deviation NOT known

t.test(o2del~1,                                                 # model formulation
       data=sepsis,                                             # data to use
       mu=1200,                                                 # null hypothesis mean
       alternative="two.sided",                                 # alternative: true mean ≠ null mean
       conf.level=.90,                                          # show 90% CI
       na.rm=TRUE)                                              # omit NA’s (missing values)
```

```
          One Sample t-test

data:  o2del
t = -5.5773, df = 167, p-value = 0.00000009658               # 2 sided p << .0001.  Reject null (mu=1200)
alternative hypothesis: true mean is not equal to 1200
90 percent confidence interval:
  971.5676 1076.0665                                         # 90% CI does NOT contain null mu=1200
sample estimates:
mean of x
 1023.817
```

```
# Test of Variance
library(DescTools)                                          # You could use var.test( ) in {base}. I like this

VarTest(sepsis$o2del,
      sigma.squared=1600)                                   # Null hypothesis variance (not SD!)
```

```
          One Sample Chi-Square test on variance

data:  sepsis$o2del
X-squared = 17498, df = 167, p-value < 0.00000000000000022     # 2 sided p << .0001 Reject null (sigma2 = 1600)
alternative hypothesis: true variance is not equal to 1600
95 percent confidence interval:
 136784.9 210324.6                                             # 95% CI does NOT contain null sigma2 = 1600
sample estimates:
variance of x
     167643.2
```

6.2. One Sample – Discrete Outcome

Binomial Distribution Model

At a Glance

Numerical Summarization	summary(outcome) # Method 1 library(summarytools) # Method 2 freq(df$outcome) # Outcome must be factor
Confidence Interval Estimation	# Confidence Interval for proportion - EXACT binom.test(x=#events,n=ntrials,conf.level=.90)$conf.int # Default is conf.level=.95 # Confidence Interval for proportion - NORMAL APPROXIMATION prop.test(x=#events,n=ntrials,conf.level=.90)$conf.int # Default is conf.level=.95
Hypothesis Testing	# Hypothesis Test for Binomial Proportion - EXACT binom.test(x=#events,n=ntrials,p=nullp, alternative=”less”) # “two.sided”, “greater”, “less” # Hypothesis Test for Binomial Proportion - NORMAL APPROXIMATION prop.test(x=#events,n=ntrials,p=nullp, alternative=”less”) # “two.sided”, “greater”, “less”

Examples.

```
# Binomial Proportion:  Exact Inference

library(tidyverse)


# For small to moderate sample size - For illustration I will obtain a small sample size = 25
temp <- sepsis %>%
       sample_n(25, na.rm=TRUE)

xevents <- sum(temp$treat, na.rm=TRUE)            # sum of 0/1 events gives x = xevents = # successes
ntrials <- sum(!is.na(temp$treat))                # sum of !is.na gives n = ntrials = # trials


binom.test(x=xevents,n=ntrials,p=.5)                          # Hypothesis Test (Null: p = .50)
```

```
         Exact binomial test

data:  xevents and ntrials
number of successes = 14, number of trials = 25, p-value = 0.69
alternative hypothesis: true probability of success is not equal to 0.5     # p=.69 do NOT reject null proportion =.50
95 percent confidence interval:
 0.3492816 0.7559763
sample estimates:
probability of success 
                  0.56
```

```
# Binomial Proportion:  Normal Approximation

library(tidyverse)

xevents <- sum(sepsis$treat, na.rm=TRUE)            # sum of 0/1 events gives x = xevents = # successes
ntrials <- sum(!is.na(sepsis$treat))                # sum of !is.na gives n = ntrials = # trials


prop.test(x=xevents,n=ntrials,p=.5, correct=FALSE)    # Hypothesis Test (Null: p = .50)
```

```
          1-sample proportions test without continuity correction

data:  xevents out of ntrials, null probability 0.5
X-squared = 0.10769, df = 1, p-value = 0.7428                # pvalue = .74 do NOT reject null proportion =.50
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.4466279 0.5381163                                         # 95% CI contains the null proportion = .50
sample estimates:
        p 
0.4923077
```

6.3. One Sample PAIRED – Continuous Outcome Normal Distribution Model

Preliminary – Is your paired data wide or long?

Wide Data

```
For each studyid, the pre and post data are in the SAME row (horizontal)
e.g., pre = sbp1 and post = sbp2
```

```
  studyid sbp1 sbp2
1       1  120  115
2       2  140  138
```

Long Data

```
For each studyid, the pre and post data are each in their OWN/SEPARATE rows (vertical)
In long data, you have a variable that tells you occasion (pre v post)
and another variable that is the outcome
```

```
  studyid    visit sbp
1       1      pre 120
2       1     post 115
3       2      pre 140
4       2     post 138
```

At a Glance

Numerical Summarization	`* WIDE:  Paired variables (e.g., pre and post) in WIDE format`
	`myvars <- c(“prevar”, “postvar”)`
	`descr(df[myvars],`
	`      stats=c(“n.valid”, “mean”, “sd”, “med”, “min”, “max”),        # User chooses`
	`      transpose=TRUE)`
	
	`* LONG:  Paired variables (e.g., pre and post) are in LONG FORMAT`
	`library(summarytools)`
	`with(df,`
	`   stby(data = outcomevar,`
	`        INDICES = timevar,                                        # timevar must be factor`
	`        FUN = descr, stats = c("mean", "sd", "min", "med","max"),    # User chooses`
	`        transpose=TRUE))`

Confidence Interval Estimation	`# Confidence Interval for mean`
	`t.test(outcome ~ 1, data=df, conf.level=.90)$conf.int         # Tip. Outcome = post - pre`
	
	`# Confidence Interval for variance`
	`library(DescTools)`
	`VarTest(df$outcome, conf.level=.90)$conf.int                 # Default is conf.level=.95`

Hypothesis Testing	`# One Sample t-test of mean`
	`t.test(outcome ~ 1, data=df, mu=nullmean)`
	
	`# One Sample t-test of variance`
	`library(DescTools)`
	`VarTest(df$outcome, sigma.squared=nullvariance)`

Examples.

```
# Paired Data Student t-Test:  WIDE

t.test(sepsis$temp0,sepsis$temp7, paired=TRUE,                  # data in WIDE
      var.equal=FALSE,
      na.rm=TRUE)
```

```
         Paired t-test

data:  sepsis$temp0 and sepsis$temp7
t = 13.144, df = 412, p-value < 0.00000000000000022         # p << .0001.  Null of equality pre/post is rejected
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.093632 1.478282                                          # 95% CI does NOT contain Null difference of 0
sample estimates:
mean of the differences
               1.285957
```

```
# Paired Data Student t-Test: LONG

library(tidyverse)

# paired t LONG requires sorted by id then by occasion nested in id
longdf <- longdf %>%
         arrange(id, hour)

# Now do paired t - LONG
t.test(temp ~ hour, data=longdf, paired=TRUE)
```

```
         Paired t-test

data:  temp by hour
t = 13.144, df = 412, p-value < 0.00000000000000022        # p << .0001.  Null of equality pre/post is rejected
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.093632 1.478282                                         # 95% CI does NOT contain Null difference of 0
sample estimates:
mean of the differences
               1.285957
```

7. Two Sample Inference

Dataset (right click to download):
`sepsis.Rdata`

Packages used:
`{DescTools}, {stargazer}, {summarytools} {tidyverse}`

Tip for Hypothesis Testing

Alternative Hypothesis	R Code
`Two sided`	`, alternative=”two.sided”`
`Right tail`	`, alternative=”greater”`
`Left tail`	`, alternative=”less”`

Tip for Confidence Intervals

If you want …	R Code
`95% CI`	`Nothing you need to do ... this is default`
`90% CI`	`, conf.level = .90`
`... and so on`	`, conf.level = .FILLIN`

Load R dataset to session

Step 1: If you have not already done so, right click to download `sepsis.Rdata` from course website.

Step 2: R Studio/Posit in the Cloud Users Only) Upload `sepsis.Rdata`

Step 3: Put `sepsis.Rdata` into your working directory

Step 4: `load(file="sepsis.Rdata")`

7.1. Two Independent Samples – Continuous Outcome

Normal Distribution Model

At a Glance

<table>
<tr><th></th><th></th></tr>
<tr><td><b>Numerical Summarization</b></td><td><pre>
* LONG: data are in LONG format
by(df[, c(“outcomevar”)],   # summarize only outcomevar
   df$groupvar,   # grouping variable
   summary)   # use function summary in {base}

library(summarytools)
with(df,
   stby(data = outcomevar,
        INDICES = groupvar,   # groupvar must be factor
        FUN = descr, stats = c("mean", "sd", "min", "med","max"),   # User chooses
        transpose=TRUE))
</pre></td></tr>
<tr><td><b>Confidence Interval Estimation</b></td><td><pre>
* LONG: data are in LONG format
# Confidence Interval for mean difference (group1 – group2)
t.test(outcome ~ groupvar, data=df, conf.level=.90)$conf.int
</pre></td></tr>
<tr><td><b>Hypothesis Testing</b></td><td><pre>
# Two Sample Test of Equality of Variances
var.test(outcome ~ groupvar, data=df,
      alternative = “two.sided”)   # “two.sided”, “greater”, “less”

# Two Sample Test of Equality of Means – UNEQUAL variances
t.test(outcome ~ groupvar, data=df,
    alternative=”two.sided”)   # “two.sided”, “greater”, “less”

# Two Sample Test of Equality of Means – EQUAL variances
t.test(outcome ~ groupvar, data=df, var.equal=TRUE,
    alternative=”two.sided”)   # “two.sided”, “greater”, “less”
</pre></td></tr>
<tr><td></td><td></td></tr>
</table>

Examples.

```
# Test of Equality of Variances

# REQUIRED:  group variable must be factor
sepsis$fatef <- factor(sepsis$fate,
                    levels=c(0,1),
                    labels=c("Alive", "Dead"))

var.test(o2del ~ fatef, data=sepsis)                        # Preliminary: test of vars

# Test of Equality of Means
t.test(o2del ~ fatef, data=sepsis,                          # t-test assuming equal var (provides CI, too)
        var.equal=TRUE)
```

```
         F test to compare two variances

data:  o2del by fatef
F = 0.91965, num df = 100, denom df = 66, p-value = 0.6975            # okay to assume equal variances
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.5846373 1.4175632
sample estimates:
ratio of variances
         0.9196542
```

```
         Two Sample t-test

data:  o2del by fatef
t = 2.5796, df = 166, p-value = 0.01076                                 # reject Null of equal means
alternative hypothesis: true difference in means between group Alive and group Dead is not equal to 0
95 percent confidence interval:
  38.40254 288.94124
sample estimates:
mean in group Alive  mean in group Dead
          1089.0910            925.4191
```

7.2. Two Independent Samples – Discrete Outcome

Binomial Distribution Model

At a Glance

Numerical Summarization	table(df$discrete1,df$discrete2, useNA=”always”) # Method 1
	
	library(summarytools) # Method 2
	with(df,
	     ctable(rowvar, colvar, # vars must be factor
	     prop="n"), # User chooses “n”, “r”, “c”
	     totals=TRUE) # use this if you want totals

Hypothesis Testing	# Fisher Exact Test of Equality of Proportions (NULL: Odds Ratio = 1)
	fisher.test(df$rowvar,df$colvar)
	
	# Chi Square Test of Equality of Proportions – WITH continuity correction (default)
	chisq.test(df$rowvar,df$colvar)
	
	# Chi Square Test of Equality of Proportions – WITHOUT continuity correction
	chisq.test(df$rowvar,df$colvar, correct=FALSE)

Example.

```
mytable <- table(sepsis$treat,sepsis$fate)                # Use table( ) to create table
dimnames(mytable) <- list(
    Treatment=c("Untreated","Treated"),
    Fate=c("Alive","Dead"))

mytable
chisq.test(mytable,correct=FALSE)                         # large n, no correction needed
```

```
          Fate
 Treatment  Alive Dead
  Untreated   139   92
  Treated     140   84

         Pearson's Chi-squared test

data:  mytable
X-squared = 0.25959, df = 1, p-value = 0.6104                # p-value = .61 Do NOT reject null of independence
```

8. Some Good Videos

__1. *(Source: MarinStats Lectures – R Tutorials)*
One Sample T-Test and Constructing One Sample Confidence Interval (video, 4:39)

__2. *(Source: MarinStats Lectures – R Tutorials)*
Wilcoxon Signed Rank Test (video, 3:38)

__3. *(Source: MarinStats Lectures – R Tutorials)*
Paired T Test in R (video, 4:19)

__4. *(Source: MarinStats Lectures – R Tutorials)*
Two Sample T-Test and Confidence Interval in R (video, 5:21)