

BIOSTATS 640 – Introduction to R  
Spring 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  
*February 17, 2023*

Right click to download R dataset  
[sepsis.Rdata](#)

**Welcome to Lesson 02!**

In this lesson, we begin with some nuts and bolts tools of R that ... well ... you just need to know! Next, you'll learn how to create a categorical variable in R (R calls this a factor and it is not the same thing as a character/string variable). The remainder of this lesson will be an introduction to using R Studio for one and two sample inference. Enjoy!

		Page
1	Highlight of Lesson 01 – Up and Running with R Studio .....	2
2	Set Your Working Directory .....	3
3	Good to Know: Introduction to Packages .....	4
4	How to Import an Excel Dataset into R Studio .....	6
5	How to Create a Categorical Variable: Introduction to Factors in R. ....	9
6	One Sample Inference .....	11
7	Two Sample Inference .....	18
8	Some Good Videos .....	22

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

<b>We will be doing all our work in R Studio</b>	R Studio is an application that sits “on top” of R. R is then “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 <code>c( )</code> to create vectors of data and separate arguments by commas	<b>For example:</b> <code>v1 &lt;- c(14, 35, 81, 99)</code>
<b>dataframe</b> A “dataset” (analogous to excel spreadsheet or SAS dataset or Stata dataset) is called a <b>dataframe</b> in R	
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	<b>For example:</b>  <code>arthritis\$Age</code>
Statistical functions run on <b>COMPLETE</b> data only. <b>Tip:</b> use option <code>na.rm=T</code> to remove missings (NA)	<b>For example:</b> <code>mean(v1, na.rm=T)</code> <code>mean(v1, na.rm=TRUE) # This also works.</code>

## 2. Set Your Working Directory

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

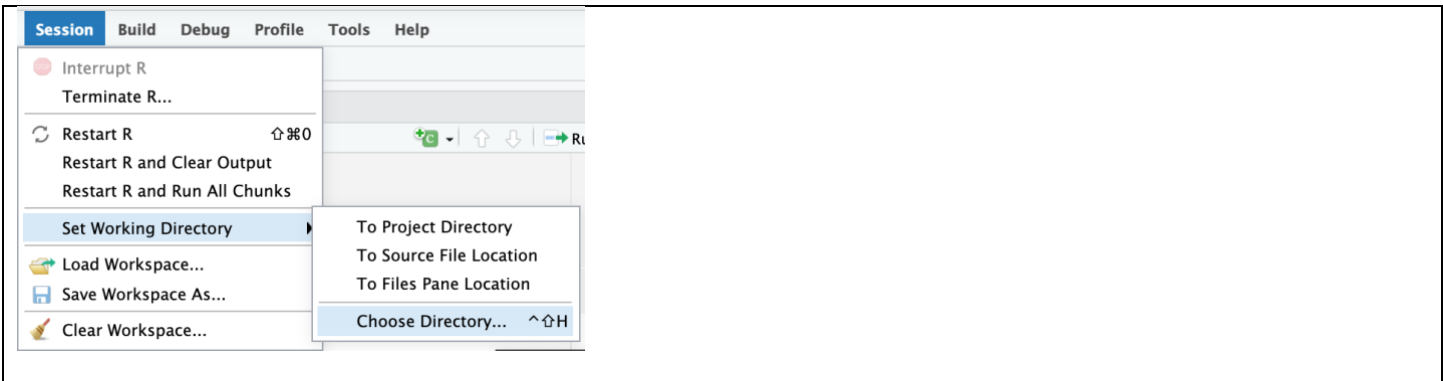
R needs to know where to find the files to **read from** and where to **write to**. This location is a directory with an associated path and is known as your **working directory**.

**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

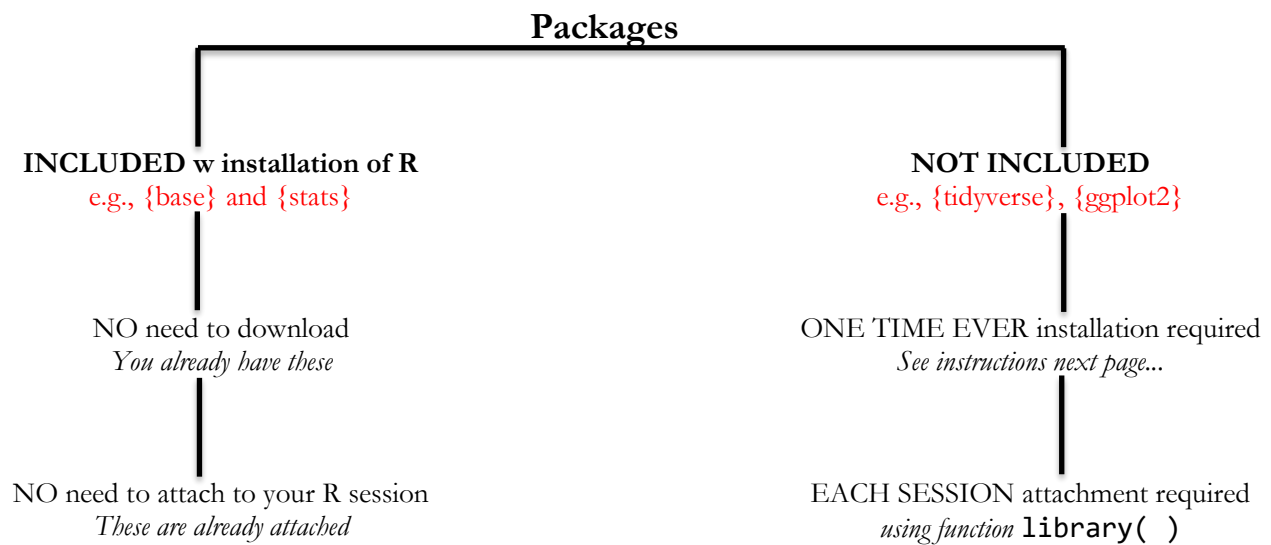
#3. Good to Know!  
Introduction to Packages

**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

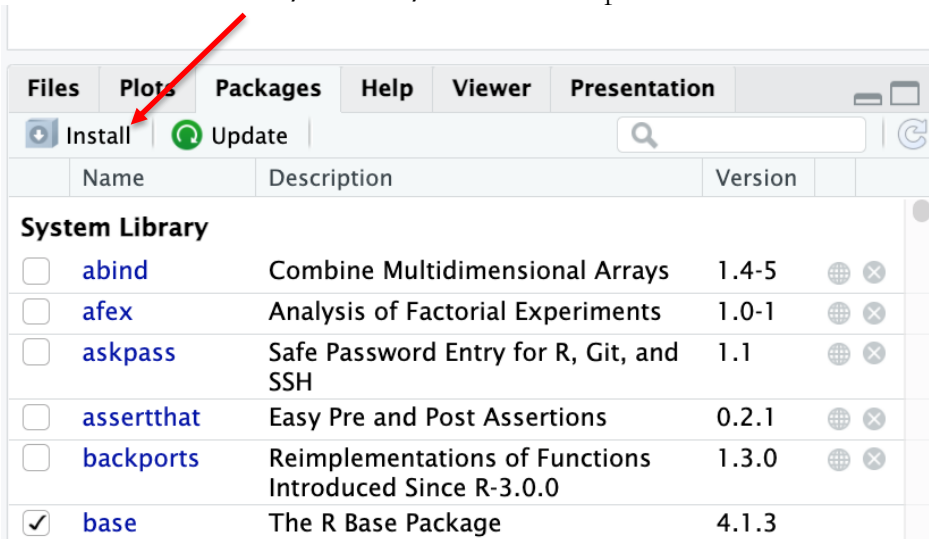
There are two types of packages, included with installation and not included with installation.



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

Example: `{ swirl }`

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



**Key:**

	<p>In <b>Install from:</b> <b>default (Repository CRAN)</b> is fine</p> <p>In <b>Packages (separate multiple with space or comma:)</b> <b>swirl</b></p> <p>In <b>Install to Library:</b> <b>leave as is</b></p> <p>Check box for <b>“Install dependencies”:</b> <b>check</b></p> <p>At bottom, click <b>Install</b></p>
--	---

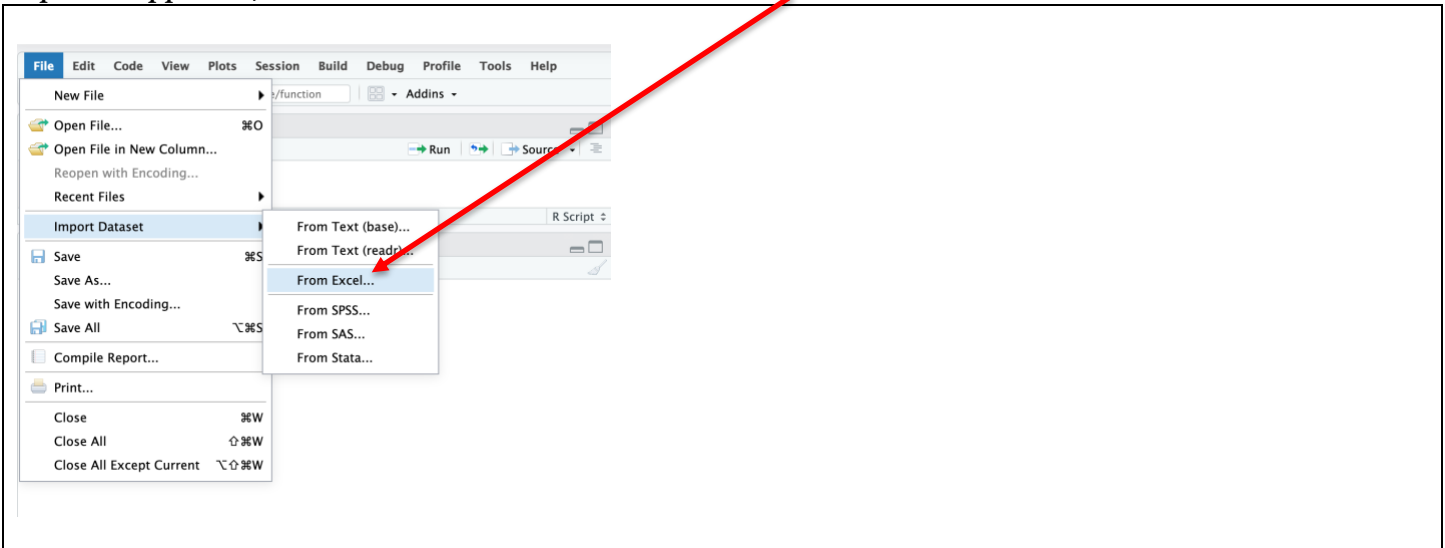
## #4. How to Import an Excel Dataset into R

### 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](#))

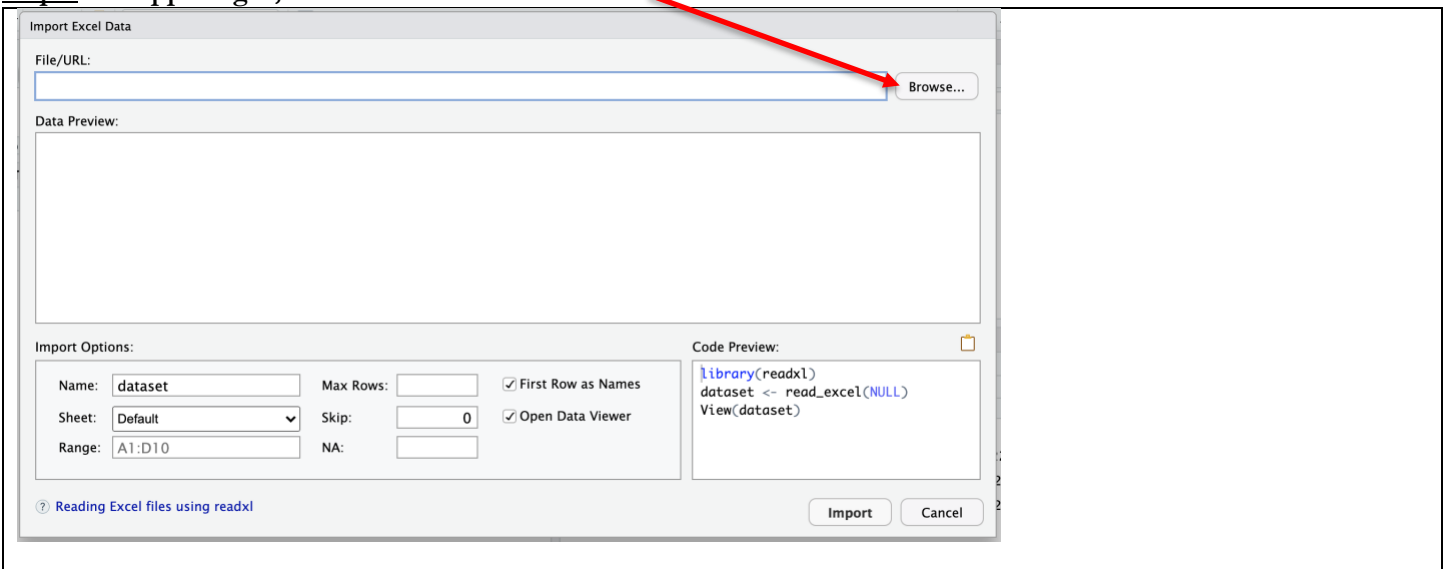
### How to Import Excel Data Using R Studio Menus

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

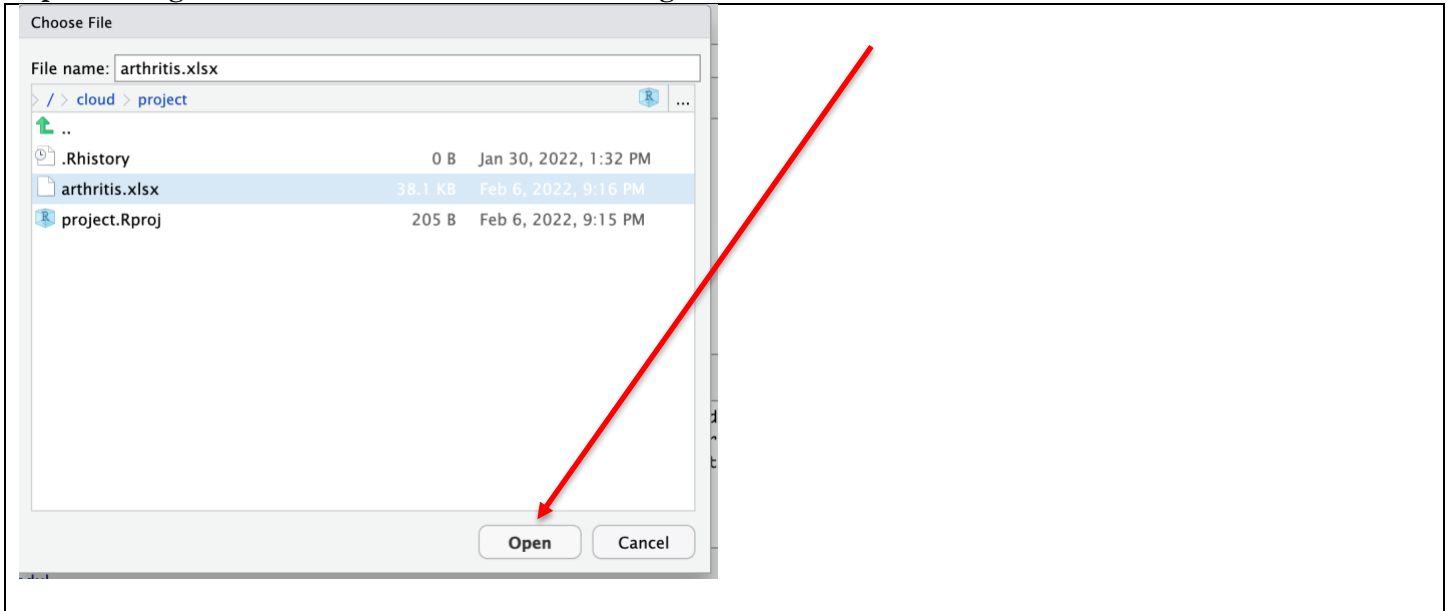


Note: R may return a message saying that you need to install readxl. Click YES. Then wait until you get a prompt.

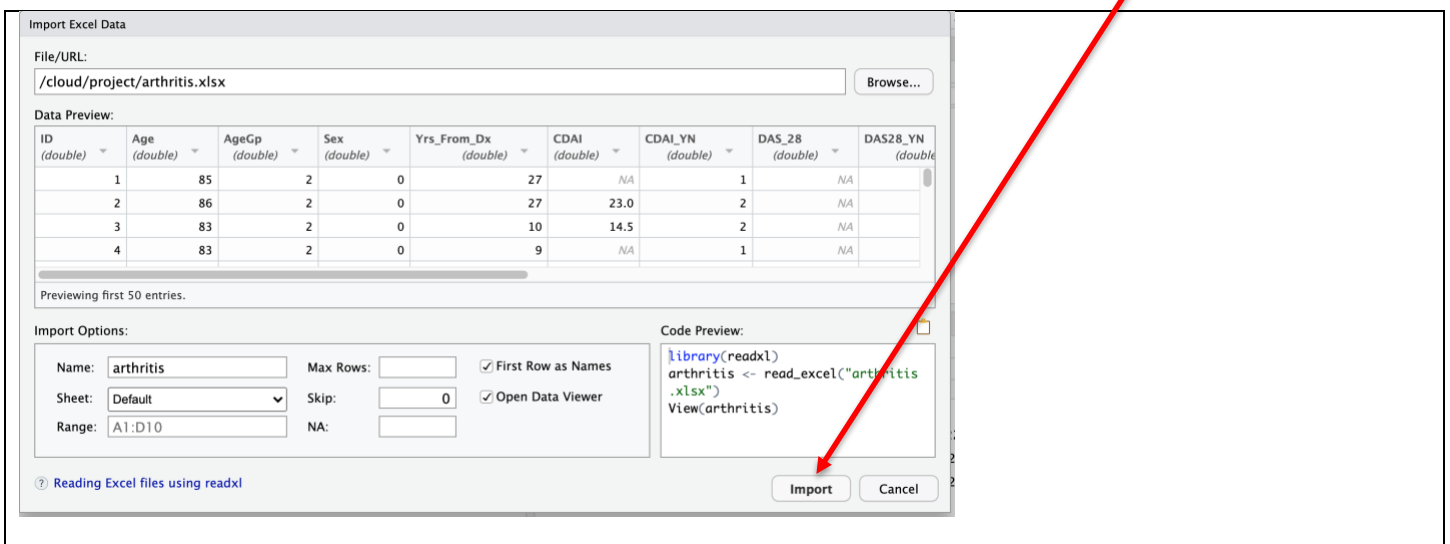
**Step 2:** At upper right, click on the icon BROWSE.



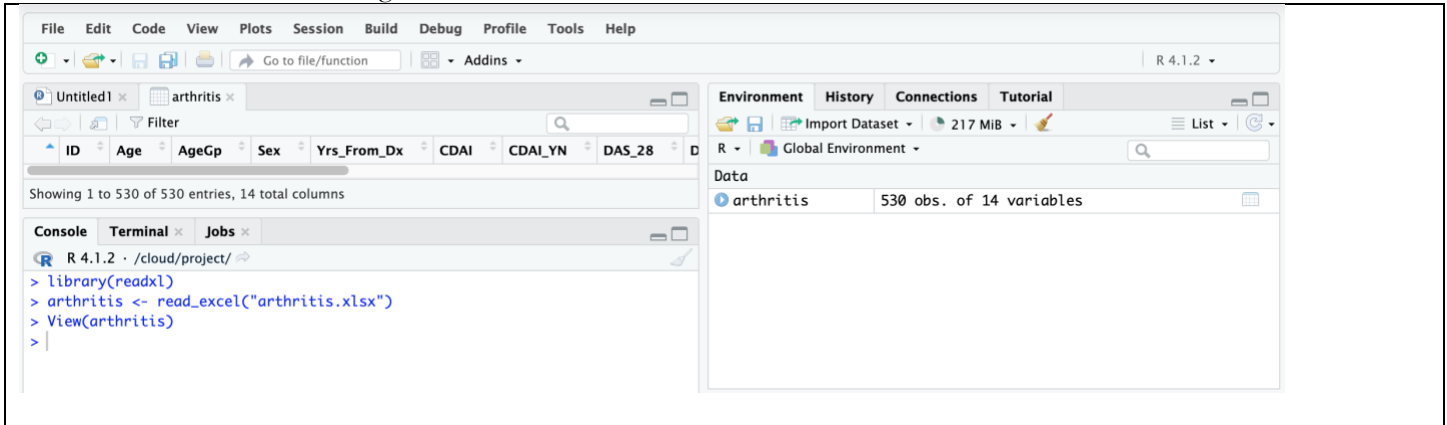
**Step 3:** Navigate to choose arthritis.xlsx. At lower right, click OPEN



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

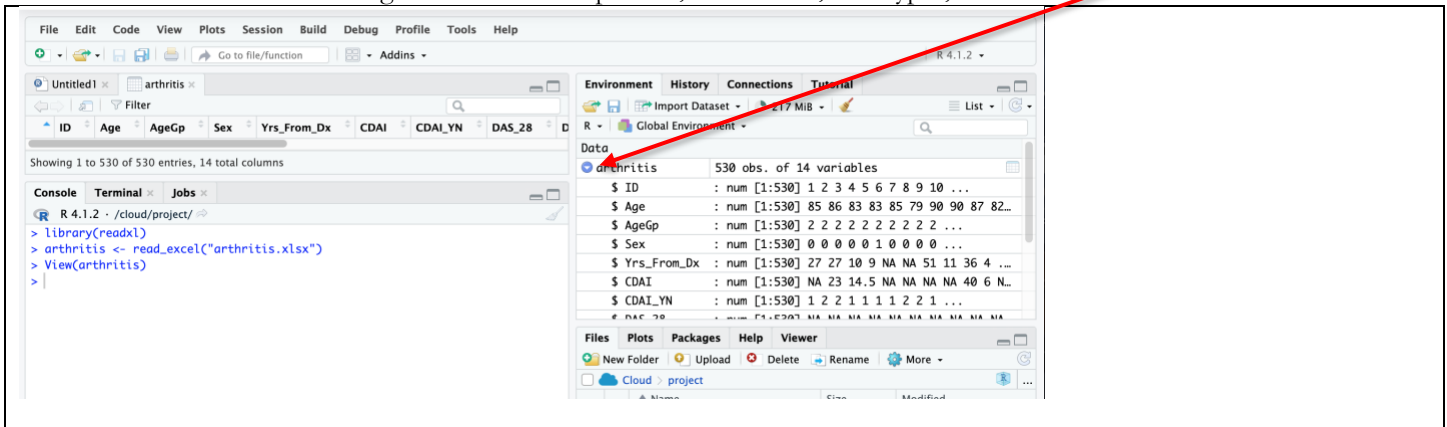


You should now see the following



**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.





#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"
```

## 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	<code>, alternative="two.sided"</code>
Right tail	<code>, alternative="greater"</code>
Left tail	<code>, alternative="less"</code>

### Tip for Confidence Intervals

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

### 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**

<p><b>Numerical Summarization</b></p>	<pre>summary(outcome) # Method 1 library(summarytools) # Method 2 descr(df\$outcome, # User chooses       stats=c("n.valid", "mean", "sd", "med", "min", "max"),       transpose=TRUE)</pre>
<p><b>Confidence Interval Estimation</b></p>	<pre># 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</pre>
<p><b>Hypothesis Testing</b></p>	<pre># 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)</pre>

Examples.

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

library(DescTools)
ZTest(sepsis$o2del,
      mu=1000,
      sd_pop=409,
      alternative="greater")
```

# null hypothesis mean  
# known population standard deviation sigma  
# alternative: true mean > null mean

One Sample z-test

```
data: sepsis$o2del
z = 0.75478, Std. Dev. Population = 409, p-value = 0.2252
alternative hypothesis: true mean is greater than 1000
95 percent confidence interval:
 971.9137      Inf
sample estimates:
mean of x
 1023.817
```

# Null  $\mu=1000$  v  $\mu > 1000$  is NOT rejected

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

t.test(o2del~1,
      data=sepsis,
      mu=1200,
      alternative="two.sided",
      conf.level=.90,
      na.rm=TRUE)
```

# model formulation  
# data to use  
# null hypothesis mean  
# alternative: true mean  $\neq$  null mean  
# show 90% CI  
# omit NA's (missing values)

One Sample t-test

```
data: o2del
t = -5.5773, df = 167, p-value = 0.00000009658
alternative hypothesis: true mean is not equal to 1200
90 percent confidence interval:
 971.5676 1076.0665
sample estimates:
mean of x
 1023.817
```

# 2 sided  $p << .0001$ . Reject null ( $\mu=1200$ )  
# 90% CI does NOT contain null  $\mu=1200$

```
# Test of Variance
library(DescTools)
VarTest(sepsis$o2del,
      sigma.squared=1600)
```

# You could use var.test( ) in {base}. I like this  
# Null hypothesis variance (not SD!)

One Sample Chi-Square test on variance

```
data: sepsis$o2del
X-squared = 17498, df = 167, p-value < 0.0000000000000022
alternative hypothesis: true variance is not equal to 1600
95 percent confidence interval:
 136784.9 210324.6
sample estimates:
variance of x
 167643.2
```

# 2 sided  $p << .0001$  Reject null ( $\sigma^2 = 1600$ )  
# 95% CI does NOT contain null  $\sigma^2 = 1600$

**6.2. One Sample – Discrete Outcome  
Binomial Distribution Model**

**At a Glance**

<b>Numerical Summarization</b>	<pre>summary(outcome) # Method 1 library(summarytools) # Method 2 freq(df\$outcome) # Outcome must be factor</pre>
<b>Confidence Interval Estimation</b>	<pre># 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</pre>
<b>Hypothesis Testing</b>	<pre># Hypothesis Test for Binomial Proportion - EXACT binom.test(x=#events,n=ntrials,p=nullp, # "two.sided", "greater", "less"            alternative="less") # Hypothesis Test for Binomial Proportion - NORMAL APPROXIMATION prop.test(x=#events,n=ntrials,p=nullp, # "two.sided", "greater", "less"           alternative="less")</pre>

**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	120	115
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**

<b>Numerical Summarization</b>	<pre> * WIDE: Paired variables (e.g., pre and post) in WIDE format myvars &lt;- 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))                     </pre>
<b>Confidence Interval Estimation</b>	<pre> # 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                     </pre>
<b>Hypothesis Testing</b>	<pre> # 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)                     </pre>



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 = <span style="background-color: yellow;">.FILLIN</span>

### 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**

<p><b>Numerical Summarization</b></p>	<pre>* LONG: data are in LONG format by(df[, c("outcomevar")],    df\$groupvar,    summary) # summarize only outcomevar # grouping variable # use function summary in {base}  library(summarytools) with(df,   stby(data = outcomevar,         INDICES = groupvar,         FUN = descr, stats = c("mean", "sd", "min", "med", "max"),         transpose=TRUE)) # groupvar must be factor # User chooses</pre>
<p><b>Confidence Interval Estimation</b></p>	<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>
<p><b>Hypothesis Testing</b></p>	<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>

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,
       var.equal=TRUE)                       # t-test assuming equal var (provides CI, too)
```

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**

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

**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](#))