Unit 7

R for Analysis of One, Two and Three^{+} Samples

"Vive la difference!"

Statistical analysis often involves the fitting of sophisticated models (multiple predictor linear regression, logistic, survival, mixed models, etc). Among the limitations of these methods are: (1) it is difficult to appreciate the actual data; and (2) their validity rest on assumptions that may or may not hold.

Analyses of data should begin with simple approaches that are as close to the data and as ***"model-free"*** as possible. These have the advantage of being simple, relatively assumption free, and straightforward in their interpretation.

This unit describes the use of Stata for estimation and hypothesis tests of data in one, two and three plus n two samples.

Important! Be sure that you have already produced your data descriptions (See again, units 5 – ***R for Data Description*** and 6- ***R for Graphs***)!

Table of Contents

Learning Objectives

When you have finished this unit, you should be able to produce, using R:

- Confidence intervals and hypothesis tests for **one continuous variable** under the assumption of normality**;**
- A nonparametric hypothesis test for **one continuous or ordinal variable** in the small study setting where the assumption of normality is not appropriate;
- Confidence intervals and hypothesis tests for **one proportion** under the assumption of a binomial distribution;
- Confidence intervals and hypothesis for **paired continuous variables** under the assumption of normality;
- A nonparametric hypothesis test for **paired continuous or ordinal variables** in the small study setting where the assumption of normality is not appropriate;
- Confidence intervals and hypothesis tests for **two independent variables – continuous** under the assumption of normality;
- Confidence intervals and hypothesis tests for **two independent proportions** under the assumption of independent binomial distributions;
- A nonparametric hypothesis test for **two independent continuous or ordinal variables** in the small sample setting where the assumption of normality is not appropriate;
- A **one way analysis of variance** under the assumption of normality; ***and***
- A nonparametric hypothesis test for the comparison of three or more independent medians in the small sample setting where the assumption of normality is not appropriate.

Packages Used in These Notes **Be sure to have done a one-time installation:** **__1. tidyverse (note: ggplot2 is a componenet)** **__2. Hmisc** **__3. summarytools** **__4. aplpack**	**Data Used in These Notes** **Right click to download from the course website** **https://people.umass.edu/~biostat690c/** **__1. sepsis.Rdata** **__2. hers_640anova.Rdata** **__3. bpwide.Rdata** **__4. auto.Rdata**

Good to know

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

What could go wrong

#1. Comma: You forgot a comma

#2. Quotes: Unbalanced quotes

#3. Variable type: The variable is not the right type (eg – must be factor).
Suggestions: a) check the variable type using class( ); b) if need be, use the original variable to create a new variable that is the required type.

#4. "Data frame" is not actually a data frame: The analysis you are running produces no output or it produces only a title *Suggestions: a) it may be that your "dataframe" is not actually a dataframe and it needs to be a data frame; b) if need be, define it as a data frame using mydataframe <- as.data.frame(mydataframe)*

#5. Command is not working because of an incompatibility of multiple packages being attached at the same time: The analysis you are running produces no output or it produces only a title
Suggestions: a) it may be that you are using a function in a package that is interfering with a function by the same name in another package that happens to also be attached to your session; b) the solution is to re-write the command with the name of the package followed by two colons. For example:

```
summarytools::descr(sepsis$o2del, stats = c("n.valid","mean", "sd","med", "min", "max"))
```

#6. Missing values: Sometimes your output will be just NA. This may be because the function you are attempting to execute will not run if there are any missing values in the data you are using.
Suggestion: a) Just to be sure, tell R to work with non-missing data ONLY; b) tip – the way to do this may be different, depending on the package and the function you are using. *or example:*

```
Rmisc::CI(na.omit(df$x),ci = 0.90)
DescTools:: MeanCI(df$x,na.rm=TRUE,conf.level=0.90)
```

Sample Session

Data Used
Right click to download from the course website
https://people.umass.edu/~biostat690c/

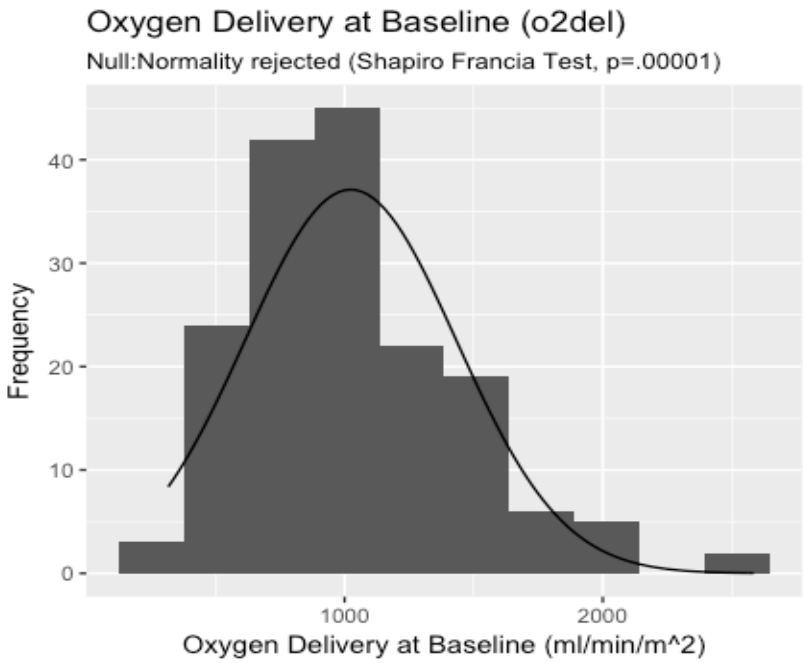
sepsis.Rdata

References:
Dupont WD Statistical Modeling for Biomedical Researchers, Second Edition. Cambridge University Press, 2008.

Benard GR, Wheeler AP et al (1997) The effects of ibuprofen on the physiology and survival of patients with sepsis. The Ibuprofen in Sepsis Study Group. NEJM 336: 912-8.

```
# PRELMINARIES

setwd("/Users/cbigelow/Desktop")          # set working directory
load(file="sepsis.Rdata")                 # load data (this assumes it is in your working directory)


library(dplyr)                                        # use package dplyr (component of tidyverse)
sepsis <- sepsis %>%                                  # begin with data frame sepsis THEN DO
   dplyr::select(temp0,temp7,treat,fate,apache,o2del,id)          # select variables of interest

str(sepsis)                                           # check dataframe structure

'data.frame':  455 obs. of  7 variables:
 $ temp0 : num  95.4 101.6 101 101 101.4 ...
 $ temp7 : num  96.2 99 NA 99.6 100.8 ...
 $ treat : num  0 1 0 1 0 1 1 0 1 0 ...
 $ fate  : num  1 0 1 0 0 0 1 0 0 0 ...
 $ apache: num  27 14 33 3 5 13 34 11 25 20 ...
 $ o2del : num  539 NA 551 1376 NA ...
 $ id    : num  1 2 3 4 5 6 7 8 9 10 ...
 - attr(*, "datalabel")= chr ""
 - attr(*, "time.stamp")= chr " 2 Sep 2008 10:17"
 - attr(*, "formats")= chr  "%9.0g" "%9.0g" "%9.0g" "%9.0g" ...
 - attr(*, "types")= int  254 254 254 254 254 254 254 254 254 254 ...
 - attr(*, "val.labels")= Named chr  "" "treatmnt" "race" "" ...
  ..- attr(*, "names")= chr  "" "treatmnt" "race" "" ...
 - attr(*, "var.labels")= chr  "Patient ID" "Treatment" "Race" "Baseline APACHE Score" ...
 - attr(*, "version")= int 114
 - attr(*, "label.table")=List of 3
  ..$ race    : Named int  0 1 2
  .. ..- attr(*, "names")= chr  "White" "Black" "Other"
  ..$ fate    : Named int  0 1
  .. ..- attr(*, "names")= chr  "Alive" "Dead"
  ..$ treatmnt: Named int  0 1
  .. ..- attr(*, "names")= chr  "Placebo" "Ibuprofen"
 - attr(*, "expansion.fields")= list()
 - attr(*, "byteorder")= int 2
```

```
attr(sepsis,"label.table")                 # check variable value labels
$race
White Black Other
    0     1     2

$fate
Alive  Dead
    0     1

$treatmnt
  Placebo Ibuprofen
        0         1

# ONE SAMPLE INFERENCE – Continuous variable

# descriptives
library(summarytools)
summarytools::descr(sepsis$o2del, stats = c("n.valid","mean", "sd","med", "min", "max"),
transpose = TRUE)

Descriptive Statistics
sepsis$o2del
N: 455

              N.Valid      Mean   Std.Dev   Median      Min       Max
----------- --------- --------- --------- -------- -------- ---------
      o2del    168.00   1023.82    409.44   947.20   316.88   2584.34

# test of normality
library(DescTools)
DescTools::ShapiroFranciaTest(sepsis$o2del)     # Test of Normality (null: normal)

       Shapiro-Francia normality test

data:  sepsis$o2del
W = 0.93575, p-value = 3.454e-06
Key:
Assumption of the null hypothesis of normality has led to an unlikely result (p-value <
.00001).  The null hypothesis is rejected.  Conclude there is statistically significant
evidence that o2del is not distributed normal.

# histogram with overlay normal
library(ggplot2)
ggplot(data=sepsis,na.rm=TRUE, aes(y=..count..,x=sepsis$o2del)) +

  geom_histogram(bins = 10,na.rm = TRUE)+

  labs(x="Oxygen Delivery at Baseline (ml/min/m^2)",
       y="Frequency",
       title="Oxygen Delivery at Baseline (o2del)",
       subtitle="Null:Normality rejected (Shapiro Francia Test, p=.00001)")+

  stat_function(
       fun = function(x, mean, sd, n, bw){
            dnorm(x = x, mean = mean, sd = sd) * n * bw},
       args = c(mean(sepsis$o2del,na.rm = TRUE),
             sd = sd(sepsis$o2del,na.rm = TRUE),
             n=sum(!is.na(sepsis$o2del)),
             bw = 226.74601))
```



Key:
While the Shapiro Francia test was statistically significant, the histogram suggests that the departure from normality is not so severe as to warrant a transformation or a non-parametric analysis.

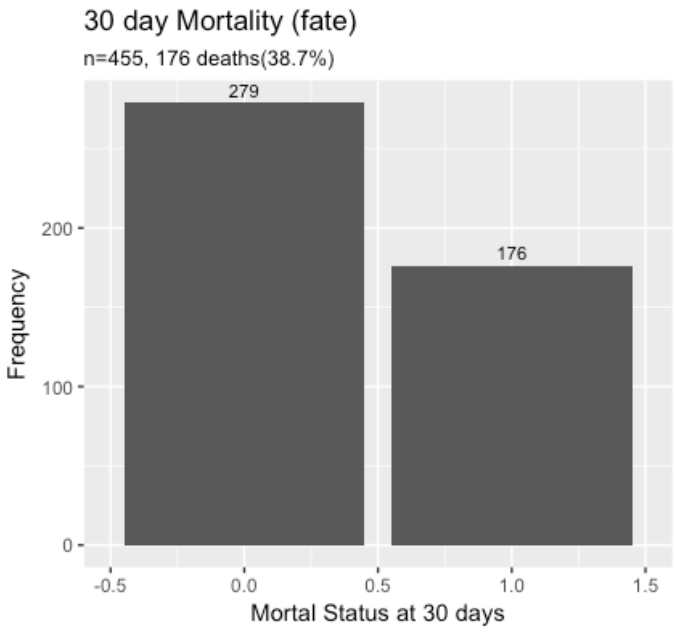
```
# one sample t-test (null: mean = 950)
t.test(sepsis$o2del, mu=950)

       One Sample t-test

data:  sepsis$o2del
t = 2.3368, df = 167, p-value = 0.02064
alternative hypothesis: true mean is not equal to 950
95 percent confidence interval:
  961.4515 1086.1827
sample estimates:
mean of x
 1023.817

# one sample CI for mean using percentiles of student-t with df = (n-1)
library(dplyr)
library(Rmisc)
sepsis %>%
  dplyr::select(o2del) %>%
  filter(!is.na(o2del))
  Rmisc::CI(data$o2del,ci = 0.99)

    upper      mean     lower
1106.1255 1023.8171  941.5086
```

```
#  PAIRED DATA – Continuous variable

# obtain change (post-pre) or (pre-post) as you like
sepsis$chg_24hrs<-sepsis$temp0-sepsis$temp7

# descriptives
library(summarytools)
summarytools::descr(sepsis$temp0, stats = c("n.valid","mean", "sd","med", "min", "max"),
transpose = TRUE)

Descriptive Statistics
sepsis$temp0
N: 455

             N.Valid     Mean   Std.Dev   Median     Min      Max
----------- --------- -------- --------- -------- ------- --------
      temp0    455.00   100.43      2.03   100.70   91.58   107.00

summarytools::descr(sepsis$temp7, stats = c("n.valid","mean", "sd","med", "min", "max"),
transpose = TRUE)

Descriptive Statistics
sepsis$temp7
N: 455

             N.Valid    Mean   Std.Dev   Median     Min      Max
----------- --------- ------- --------- -------- ------- --------
      temp7    413.00   99.19      1.84    99.14   88.70   104.18

summarytools::descr(sepsis$chg_24hrs, stats = c("n.valid","mean", "sd","med", "min", "max"),
transpose = TRUE)

Descriptive Statistics
sepsis$chg_24hrs
N: 455

                N.Valid   Mean   Std.Dev   Median     Min    Max
--------------- --------- ------ --------- -------- ------- ------
      chg_24hrs    413.00   1.29      1.99     1.22   -5.40   8.30


# t-test for paired data (data in WIDE format)
t.test(sepsis$temp0,sepsis$temp7, conf.level=0.95, paired=TRUE)

       Paired t-test

data:  sepsis$temp0 and sepsis$temp7
t = 13.144, df = 412, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.093632 1.478282
sample estimates:
mean of the differences
               1.285957
```

Key:
The 2-sided test is statistically significant (p << .0001). Reject the null hypothesis of means.

```
# ONE SAMPLE INFERENCE – Discrete variable (0/1 Bernoulli trials)

# descriptives
library(summarytools)
summarytools::freq(sepsis$fate)

Frequencies
sepsis$fate
Type: Numeric

             Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
----------- ------ --------- -------------- --------- --------------
          0    279     61.32          61.32     61.32          61.32
          1    176     38.68         100.00     38.68         100.00
       <NA>      0                               0.00         100.00
      Total    455    100.00         100.00    100.00         100.00



# ci for proportion using normal approximation method
library(Rmisc)
Rmisc::CI(sepsis$fate,ci = 0.95)

    upper      mean     lower
0.4317318 0.3868132 0.3418946


# ci for proportion using exact binomial method
library(binom)
binom::binom.confint(sum(sepsis$fate), nrow(sepsis),methods="exact")

  method   x   n      mean     lower     upper
    1   exact 176 455 0.3868132 0.3418278 0.4332801


# test of binomial proportion using normal approximation (null:  p=.30)
prop.test(sum(sepsis$fate), nrow(sepsis), p = 0.3,
          alternative = "two.sided", conf.level = 0.95, correct = TRUE)


1-sample proportions test with continuity correction

data:  sum(sepsis$fate) out of nrow(sepsis), null probability 0.3
X-squared = 15.918, df = 1, p-value = 6.613e-05
alternative hypothesis: true p is not equal to 0.3
95 percent confidence interval:
 0.3421224 0.4334455
sample estimates:
        p
0.3868132
```

```
# test of binomial proportion using exact binomial method (null:  p=.30)
binom.test(sum(sepsis$fate), nrow(sepsis), p = 0.3,
           alternative = "two.sided", conf.level = 0.95)

       Exact binomial test

data:  sum(sepsis$fate) and nrow(sepsis)
number of successes = 176, number of trials = 455, p-value = 7.854e-05
alternative hypothesis: true probability of success is not equal to 0.3
95 percent confidence interval:
 0.3418278 0.4332801
sample estimates:
probability of success
             0.3868132
Key:
The 2-sided test is statistically significant (p = .00008).  Reject the null.

# bar chart via histogram with option discrete
library(ggplot2)
ggplot2::ggplot(data=sepsis, aes(x=fate,y=..count..)) +
    geom_bar()+

    labs(y="Frequency",x="Mortal Status at 30 days",
       title="30 day Mortality (fate)",
       subtitle="n=455, 176 deaths(38.7%)") +

     stat_bin(aes(y=..count.., label=..count..,x=sepsis$fate),
           geom="text", size=3, vjust=-0.5,bins = 2)
```



```
# TWO INDEPENDENT SAMPLES INFERENCE – continuous variables
library(summarytools)

# frequency table of group variable
summarytools::freq(sepsis$treat)

Frequencies
sepsis$treat
Type: Numeric

             Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
---------- ------ --------- -------------- --------- --------------
         0    231     50.77          50.77     50.77          50.77
         1    224     49.23         100.00     49.23         100.00
      <NA>      0                               0.00         100.00
     Total    455    100.00         100.00    100.00         100.00


# descriptives of continuous outcome (apache) by group (treat)
library(psych)
psych::describeBy(sepsis$apache, sepsis$treat)

Descriptive statistics by group
group: 0
   vars   n  mean   sd median trimmed  mad min max range skew kurtosis   se
X1    1 230 15.19 6.92   14.5   14.78 6.67   0  41    41 0.61     0.35 0.46
------------------------------------------------------------------------------------------------
group: 1
   vars   n  mean   sd median trimmed  mad min max range skew kurtosis   se
X1    1 224 15.48 7.26     14   15.06 7.41   3  37    34 0.52    -0.36 0.49



# test of equality of variances (data in LONG format)
var.test(apache ~ treat, sepsis, alternative = "two.sided")


       F test to compare two variances

data:  apache by treat
F = 0.9088, num df = 229, denom df = 223, p-value = 0.4724
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.6995696 1.1800701
sample estimates:
ratio of variances
         0.9088018
```

Key:
Assumption of the null hypothesis has NOT let to an unlikely result (p-value = .47). The null hypothesis of equal variances is NOT rejected. Thus, these data provide NO statistically significant evidence that the variances are different.

```
# t test of equality of means
group0<-subset(sepsis,treat==0)
group1<-subset(sepsis,treat==1)
t.test(group0$apache,group1$apache)

Welch Two Sample t-test

data:  group0$apache and group1$apache
t = -0.4364, df = 449.52, p-value = 0.6628
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.599936  1.018492
sample estimates:
mean of x mean of y
 15.18696  15.47768

Key:
Assumption of the null hypothesis has NOT let to an unlikely result (p-value = .66).  The null
hypothesis of equal means is NOT rejected.  Thus, these data provide NO statistically significant
evidence that the means are different.
```

1. One Sample Inference

Good to know

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

1.1 Nonparametric Test of Median: Wilcoxon Signed Rank

Test	Assumptions	R Code
Test of median Wilcoxon Signed Rank Test	X **cannot** be assumed to be normally distributed	wilcox.test(x,mu=nullmedian, alternative="two.sided")

1.2 Continuous Outcome: Normal(μ, σ^2)

Test or CI	Assumptions	R Code
Test of Normality: Shapiro Francia Test	X is continuous	library(DescTools) DescTools::ShapiroFranciaTest(x)
Test of mean: Z-test	X is distributed normal and population standard deviation is known	z.test(x,mu=nullmean,stdev,n=length(x), alternative="two.sided", conf.level=0.95)
CI of mean: Assumes population standard deviation known	X is distributed normal and population standard deviation is known. Here, it is known to be = 15.	new <- source %> dplyr::summarise (n=sum(!is.na(x)), zcrit=qnorm(.975), ave=mean(x,na.rm=TRUE), sd=15, se=sd/sqrt(n), lower95 = ave - zcrit*se, upper95=ave + zcrit*se, varname="x") %>% select(varname,n,ave,lower95,upper95) new
Test of mean: t-test	X is distributed normal and population standard deviation is NOT known	t.test(x, mu=nullmean)

Test or CI	Assumptions	R Code

CI of mean:
Assumes population standard deviation NOT known
 	
2 way!
	X is distributed normal and population standard deviation is NOT known	Method I:
		`library(Rmisc)`
		`Rmisc::CI(na.omit(df$x),ci = 0.90)`
		
		Method II:
		`library(DescTools)`
		`DescTools::`
		`MeanCI(df$x,na.rm=TRUE,conf.level=0.90)`

CI of mean:
Assumes population standard deviation NOT known
 	
Direct coding - good to know
	X is distributed normal and population standard deviation is NOT known	`new <- source %>`
		`  dplyr::summarise (`
		`  n=sum(!is.na(x)),`
		`  tcrit=qt(.975, df=n-1),`
		`  ave=mean(x,na.rm=TRUE),`
		`  sd=sd(x,na.rm=TRUE),`
		`  se=sd/sqrt(n),`
		`  lower95 = ave-tcrit*se,`
		`  upper95=ave+tcrit*se,`
		`  varname="x") %>%`
		`select(varname,n,ave,lower95,upper95)`
		
		`new`

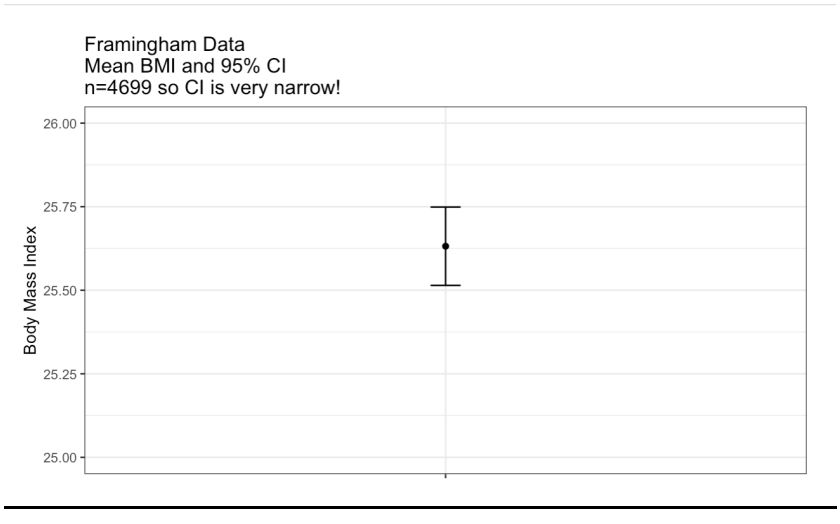
Test of variance with CI:
Chi Square test and CI estimate of SD
	X is distributed normal	`sigma.test(x,sigma=nullsd,`
		`  alternative="two.sided",`
		`  conf.level=0.95`

Plot of Mean ± 95% CI

Assumes population sigma is not known → uses percentiles of Student-t

```
library(Rmisc)
library(ggplot2)

# create dataset w mean and ci
plotdf <- summarySE(dataframe,measurevar="x", na.rm=TRUE)


# plot
ggplt(plotdf,aes(x=factor(""),y=x)) +
    geom_errorbar(aes(ymin=x-ci,ymax=x+ci,width=.1) +
    geom_point() +
    scale_x_discrete("")+
    scale_y_continuous(name="y-axislabel", limits=c(minytick,maxytick))
```

```
# Example -
# create dataset w mean and ci
framinghamdf <- as.data.frame(framinghamdf)
plotdf <- summarySE(framinghamdf,measurevar="bmi", na.rm=TRUE)

# plot
ggplot(plotdf,aes(x=factor(""),y=bmi)) +
     geom_errorbar(aes(ymin=bmi-ci,ymax=bmi+ci,width=.05)) +
     geom_point() +
     scale_x_discrete("")+
     scale_y_continuous(name="Body Mass Index", limits=c(25,26)) +
     ggtitle("Framingham Data\nMean BMI and 95% CI\nn=4699 so CI is very narrow!") +
     theme_bw()
```



1.3 Discrete – Binomial Proportion

Test or CI	Assumptions	R Code
Test and CI for binomial proportion: Exact	X is # events in n independent trials with same probability of event, p xvalue = # successes in n trials xvector = vector of 0 and 1's with 1=success.	Method I binom.test(xvalue,ntrials,p=nullp, alternative="two.sided", conf.level=0.95) Method II binom.test(sum(xvector), nrow(xvectpr), p = nullp, alternative = "two.sided", conf.level = 0.95)
CI for binomial proportion: Exact	X is # events in n independent trials with same prob of event, p	library(binom) binom::binom.confint(sum(xvector), nrow(xvector),methods="exact")
Test and CI for binomial proportion: Normal approximation	X is # events in n independent trials with same prob of event, p	Method I (with continuity correction) prop.test(xvalue, ntrials, p=nullp, alternative = "two.sided", conf.level = 0.95, correct = TRUE) Method II (with continuity correction) prop.test(sum(xvector), nrow(xvector), p=nullp, alternative = "two.sided", conf.level = 0.95, correct = TRUE)
CI for binomial proportion: Normal approximation	X is vector of 0's and 1's with 1="event" in n independent trials with same prob of event, p	library(Rmisc) Rmisc::CI(x,ci = 0.95) ci=" "could instead be: 0.90, 0.99, etc.

2. Paired Sample Inference

2.1 Nonparametric Test of Median: Wilcoxon Signed Rank

Test	Assumptions	R Code
Test of paired medians: Wilcoxon Signed Rank Test	x and y are paired and **cannot** be assumed to be normally distributed	wilcox.test(x,y,paired=TRUE)

2.2 Continuous Outcome: [Normal (μ_1, σ_1^2), Normal (μ_2, σ_2^2)]

Test	Assumptions	R Code
Test of paired means: T-test Data in WIDE format	y1 and y2 are normally distributed and are **paired** and population standard deviation of difference (y2-y1) is NOT known	t.test(y1,y2, var.equal=FALSE, paired=TRUE) Could also say: var.equal=TRUE
Test of paired means: T-test Data in LONG format Note: LONG means that the occasions of measurement are values of a separate, grouping, variable.	y is outcome measured on TWO occasions defined by a binary grouping variable group with two values	t.test(y ~ group, data=Data, paired=TRUE, conf.level=0.95) Note - group must be factor

3. Two Independent Samples Inference

3.1 Nonparametric Test of Two Medians: Wilcoxon Rank Sum Test

Test	Assumptions	R Code
Test of independent medians: Wilcoxon Rank Sum Test Data are in **WIDE** format. Note: An equivalent test is the Mann Whitney U test	y1 and y2 are **independent** and **cannot** be assumed to be distributed normal	wilcox.test(y1,y2)
Test of independent medians: Wilcoxon Rank Sum Test Data are in **LONG** format. Note: An equivalent test is the Mann Whitney U test	y is obtained from 2 independent groups defined by some group variable group. y **cannot** be assumed to be distributed normal	wilcox.test(y~group) Note - group must be factor

3.2 Continuous Outcome: [Normal (μ_1, σ_1^2), Normal (μ_2, σ_2^2)]

Test or CI	Assumptions	R Code
Test of independent variances Data are in **WIDE** format	y1 and y2 are independent and assumed distributed normal	`var.test(y1,y2)`
Test of independent variances: Data are in **LONG** format	y is obtained from 2 independent groups and is assumed distributed normal	`var.test(data=df,y~group)` Note - group must be factor
Test of independent means: t-test, **UNEQUAL** variances Data are in **WIDE** format. Note - this is Welsh t-test	y1 and y2 are independent and assumed distributed normal	`t.test(y1,y2)`
Test of independent means: t-test, **UNEQUAL** variances Data are in **LONG** format. Note - this is Welsh t-test	y is obtained from 2 independent groups and is assumed distributed normal	`t.test(data=df,y~group)` Note - group must be factor
Test of independent means: t-test, **EQUAL** variances Data are in **WIDE** format.	y1 and y2 are independent and assumed distributed normal	`t.test(y1,y2,var.equal=TRUE)`

Test of independent means: t-test, **EQUAL** variances
Data are in **LONG** format.
	y is obtained from 2 independent groups and is assumed distributed normal	`t.test(data=df,y~group,`
		`  var.equal=TRUE)`
		
		Note - group must be factor

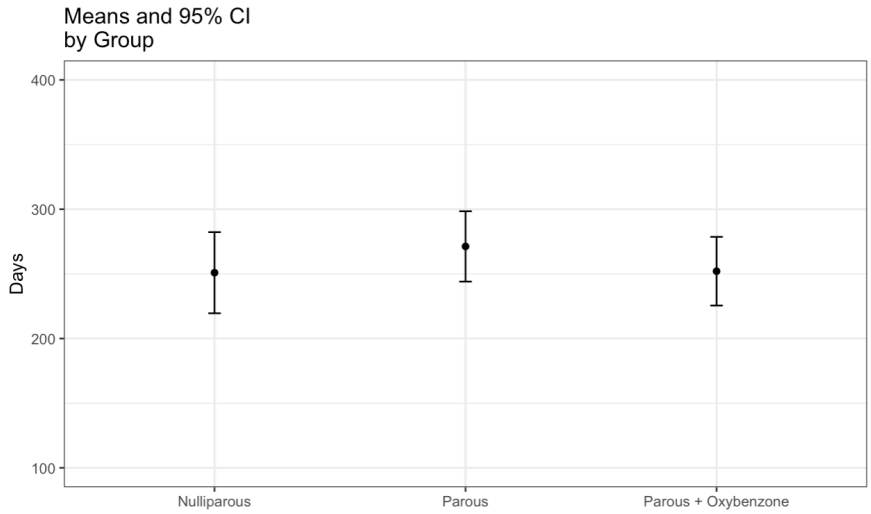
CI for mean, by group:	y is obtained from 2 independent groups and is assumed distributed normal, and sd's are assumed UNKNOWN	`library(dplyr)`
		
		`new <- source %>%`
		`  dplyr::group_by(group) %>%`
		`  dplyr::summarise(`
		`  n=sum(!is.na(y)),`
		`  tcrit=qt(.975,(n-1)),`
		`  ave=mean(y,na.rm=TRUE),`
		`  se=sd(y)/sqrt(n),`
		`  lower95 = ave-tcrit*se,`
		`  upper95=ave+tcrit*se) %>%`
		`select(group,n,ave,lower95,upper95)`
		
		`new`

Plot of Means ± 95% CI

Assumes population sigma is not known → uses percentiles of Student-t

```
library(Rmisc)
library(ggplot2)

# create dataset with mean and ci, by group
new <- Rmisc::summarySE(sourcedata,measurevar="yvar", groupvars="group",na.rm=TRUE)

# plot
ggplot2::ggplot(new,aes(x=group,y=yvar)) +
  geom_errorbar(aes(ymin=yvar-ci,ymax=yvar+ci), width=.05)+
  geom_point()  +  scale_x_discrete(name="") +
  scale_y_continuous(name="YAXISLABEL", limits=c(min_tick,max_tick))
```

```
# Example -
library(Rmisc)
library(ggplot2)

new <- Rmisc::summarySE(temp3,measurevar="fu_days",
             groupvars="treatf",na.rm=TRUE)

ggplot2::ggplot(new,aes(x=treatf,y=fu_days)) +
  geom_errorbar(aes(ymin=fu_days-ci,ymax=fu_days+ci), width=.05)+
  geom_point()  +
  scale_x_discrete(name="") +
  scale_y_continuous(name="Days",
                     limits=c(100,400)) +
  ggtitle("Means and 95% CI\nby Group") +
  theme_bw()
```



3.3 Discrete Outcome: Comparison of Two Binomial Proportions

Test or CI	Assumptions	R Code

Test of 2 binomial proportions:
 	
Data are values of
n = # trials
x = # events
	Two independent binomial distributions.	WITH continuity correction (default)
		prop.test(x=c(x1,x2), n=c(n1,n2))
		
		WITHOUT continuity correction
		prop.test(x=c(x1,x2), n=c(n1,n2),
		correct=FALSE)

Create 2x2 table: From counts "a", "b", "c" and "d"	-	table <- as.table(rbind(c(a,b),c(c,d))) dimnames(table) <- list(ROWVAR=c("label","label"), COLVAR=c("label","label"))
Create 2x2 table: From individual observations	-	table <- table(df$rowvar,df$colvar) dimnames(table) <- list(ROWVAR=c("label","label"), COLVAR=c("label","label"))
Chi Square Test: 2x2 table Null: independence, no association, p1=p2	Applicable to several designs: 2 group cohort, 2 group case-control, 4 group poisson, 1 group cross-sectional	**Without continuity correction:** chisq.test(table,correct=FALSE) **With continuity correction (default):** chisq.test(table)
Chi Square Test: Crosstab of individual observations Null: independence, no association, p1=p2 *Not recommended - the output looks cluttered. I suggest using table instead!*	Applicable to several designs: 2 group cohort, 2 group case-control, 4 group poisson, 1 group cross-sectional	**Without continuity correction:** chisq.test(df$rowvar,df$colvar, correct=FALSE) **With continuity correction (default):** chisq.test(df$rowvar,df$colvar)
Fisher Exact Test: Null: independence, no association, p1=p2	Applicable to several designs: 2 group cohort, 2 group case-control, 4 group poisson, 1 group cross-sectional	I suggest you specify HA fisher.test(table1,alternative="two.sided") fisher.test(table1,alternative="greater") fisher.test(table1,alternative="less") You can get just the OR if you like fisher.test(table1) or$estimate

4. ONE WAY Analysis of Variance

4.1 Nonparametric Test of Medians: Kruskal Wallis Test

Test	Assumptions	R Code
Kruskal-Wallis Test: Null – Equal medians	Observations y are independent but cannot be assumed distributed normal. Groups are defined by grouping variable group.	kruskal.test(y ~ group, data = df) df must be a dataframe

4.2 Continuous Outcome: One Way Analysis of Variance

Test	Assumptions	R Code
Test of Normality: Shapiro Francia Test	Y is continuous	library(DescTools) DescTools::ShapiroFranciaTest(y)
Test Homogeneity of Variances: Bartlett Test *Power is good when y is distributed normal, but is sensitive to violation of normality*	y is continuous and is distributed normal	library(car) car::bartlett.test(y~group, data=df)
Test Homogeneity of Variances: Levene Test *The Levene test is less sensitive to violation of normality but has lower power than Bartlett*	y is continuous and is distributed normal	Library(car) car::leveneTest(y~group, data=df)

- continued-

4.2 Continuous Outcome: One Way Analysis of Variance - *continued*

Test	Assumptions	R Code
Descriptives by group:	y is continuous and obtained from K independent groups defined by group	MethodI: library(summarytools) summarytools::with(df, stby(data = y, INDICES =group, FUN = descr, stats = c("mean", "sd", "min", "med", "max"))) Method II (my personal favorite): library(FSA) FSA::Summarize(y~group,data=df,na.rm=TRUE) Method III: library(Rmisc) Rmisc::summarySE(data=df,measurevar="y", groupvars=c("group"),na.rm=TRUE)
CI for mean, by group:	y is obtained from K independent groups and is assumed distributed normal, and sd's are assumed UNKNOWN	library(dplyr) new <- source %>% dplyr::group_by(group) %>% dplyr::summarise(n=sum(!is.na(y)), tcrit=qt(.975,(n-1)), ave=mean(y,na.rm=TRUE), se=sd(y)/sqrt(n), lower95 = ave-tcrit*se, upper95=ave+tcrit*se) %>% select(group,n,ave,lower95,upper95) new Want to plot? see page 20
One Way Analysis of Variance:	y is obtained from K independent groups, assumed distributed normal with constant variance	Method I: ANOVA (deviation from means codking) aov(y ~ group, data=df) Method II: Regression (reference cell coding) aov(y ~ 1 + G2 + …. + GK, data=df) where G2 … GK = 0/1 indicators of groups 2…K