Wk 5 Notes

D. ODay

2022-07-12

```
require(lolcat)
```

```
## Loading required package: lolcat
```

```
## lolcat 2.0.0
```

Hypothesis Testing

- What is a hypothesis?
  - An assumption related to a process or population
- Hypothesis Testing
  - is a procedure which uses sample statistic(s) to make inference about a population
- Statistical Significance
  - Refers to the assumption that the observed difference or association/phenomenon represents a significant departure from what might be expected by chance alone

Significance Level and Risk

- α is a selection of risk that you are willing to take
- Given a true null hypothesis, α is the probability the null hypothesis could be rejected (Type I Error)
  - The smaller the selected level of α, the smaller the probability of rejecting a true null hypothesis
  - The researcher selects this value
- **p-value**
  - The probability that an observed statistic, or one that is more extreme, could have occurred by chance, given a true null hypothesis
  - reject a null hypothesis of the p-value is less than or equal to the selected level of α
- Test Statistics
  - In hypothesis testing we
    - take samples
    - Calculate sample statistics
    - Calculate test statistics
    - Calculate Probabilities

Type 1 and 2 Error

- Type I and II Error
  - When testing the hypotheses, we can make errors with respect to our conclusions
  - these errors are referred to as Type I and Type II errors
- Type I error
  - symbol: α
  - The probability of rejecting a true null hypothesis
  - Also referred to as a false positive, or producer's risk
- Type II error
  - "The ability to detect something" symbol: 1 - β
  - The probability of rejecting a false null hypothesis
  - The ability of the test to correctly reject a false null hypothesis
- Confidence
  - Symbol: 1 - α
  - The probability of accepting a true null hypothesis

Expermental Outcomes The top row represents the True outcome that should happen. The column represents what we do according to our test

	TRUE	FALSE
Accept H_0	$1-\alpha$ (Confidence)	β (Type II Error)
Reject H_0	α (Type I Error)	$1-\beta$ (Power)

Figure 1: Error table

Observations

- $\alpha + \beta$ will never equal 1. They are conditional probabilities based upon different conditions
- Specifically, α is based upon the premise that H_0 is true, β is predicated on the assumption that H_0 is false
- Both α and β represent risk
  - the expression of the researcher's willingness to commit an error in inference
- Power, the ability of the test to correctly reject a false H_0, must be "purchased" with sample size and with the selection of an appropriate experimental design
- α is not always more important than β

Beta, Power and Sample Size

Beta and Power

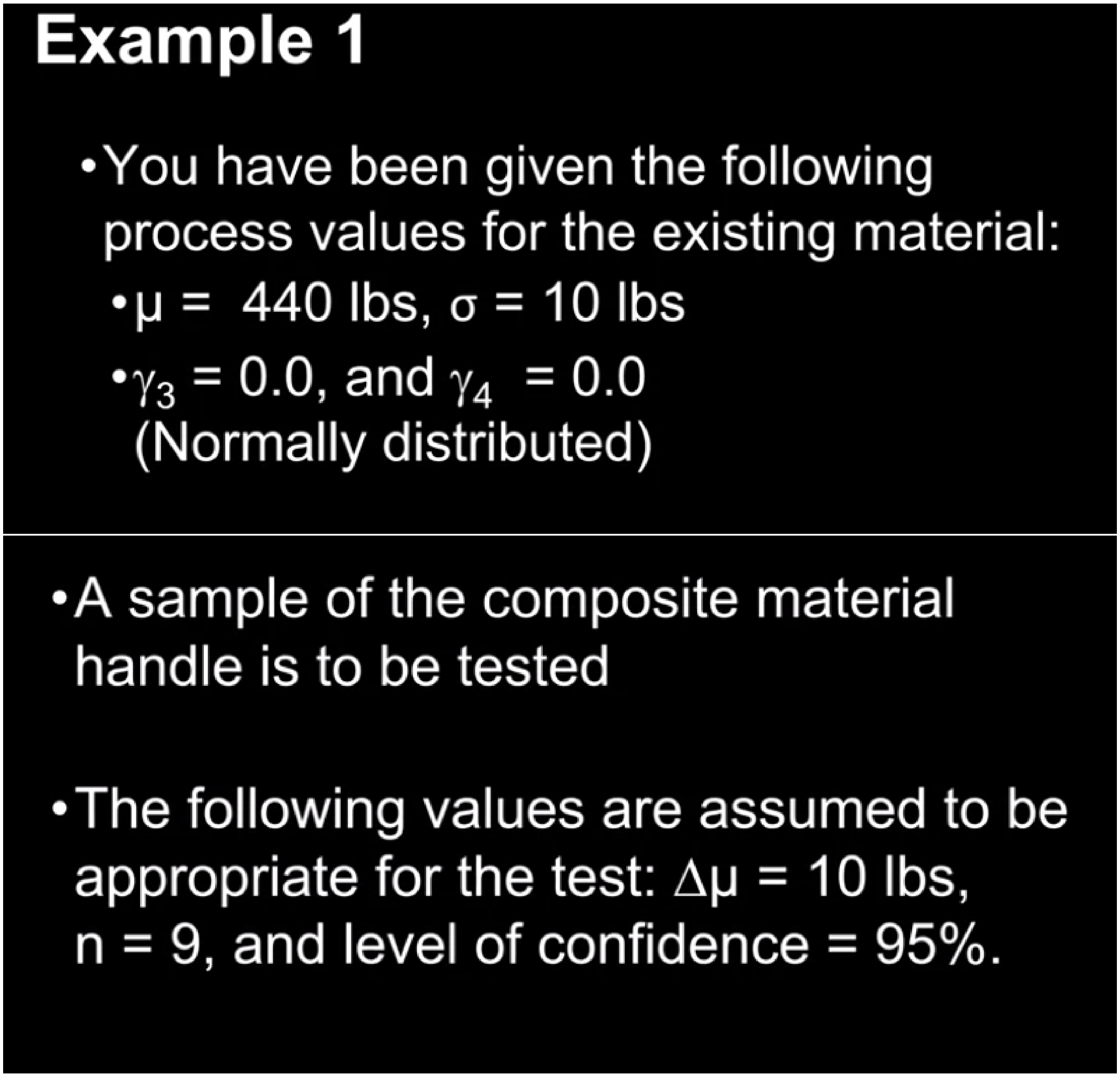
- Power is the prob of correctly reject a False H_0
  - Correct decision when H_0 is false
  - Power is designated as 1 - β
  - Used to determine how well a test is working and likely to detect a true effect (or difference)
  - Affected by
    * True value of population parameters
    * Significance level
    * Standard Deviation
    * Sample Size
- Strategies for Considering Power in Experiments
  - Strategy 1
    * Fix the probability of committing a Type I error
    * Then select a sample size large enough so that β is acceptably small and testing is not too expensive or time consuming to conduct
  - Strategy 2
    * Consider the Null and Research Hypothesis and select the α and β pair which best represent your wishes related to the research
    * Then, calculate the sample size required to maintain the selected risk levels

-Calculating β and Power for means - Determine the critical value in the H_0 corresponding the the z-value for the given value of α - Calculate the z-value and area corresponding to the calculated on the "H_{1} is true" (in other words the $\bar{X}$ value given in the alternative hypothesis) curve

Other things to keep in mind

- The large the value od $\mu_0 - \mu_0$, also known as delta, the larger power (1 - β) will become
- Generally, both α and β should be small. In industry, studies planned without an initial regard for β generally result in low power or high β values
- It's not possible to commit a Type I and Type II error at the same time
- Had a two-tailed test been employed, the power of the test would have been the sum of the two areas falling beyond α on the H_1 distribution
- Increasing α will generally reduce β
- Increasing n will generally increase power
- Increasing power of the test can be accomplished by reducing the standard error through design modifications (for example, matched groups and stratified sampling)

Calculating Power



Beta and Power for Changes in Means

```
# Example 1 - Calculating Power for Changes in Means
power.mean.t.onesample(sample.size = 9
                       ,effect.size = 10
                       ,variance.est = 10^2
                       ,alpha = 0.05
                       ,alternative = "two.sided")
```

```
##   test       type alternative sample.size actual df effect.size variance alpha
## 1    t one.sample   two.sided           9      9  8          10      100  0.05
##   conf.level      beta     power
## 1       0.95 0.2519845 0.7480155
```

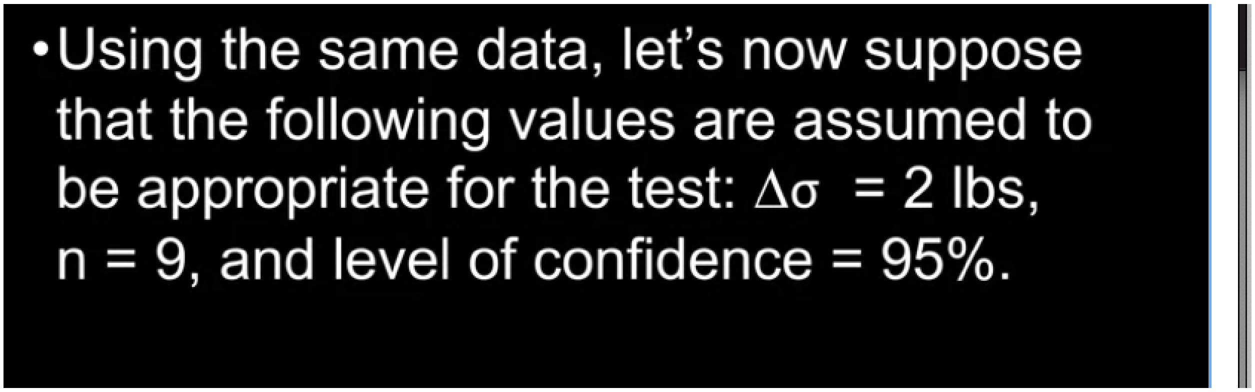


Figure 2: Ex_var

```r
# Power to detect an increase in variance
power.variance.onesample(sample.size = 9
                         ,null.hypothesis.variance = 10^2
                         ,alternative.hypothesis.variance = 12^2
                         ,alpha = 0.05
                         ,alternative = "two.sided")
```

Beta and Power for changes Variance

```
##         test       type alternative sample.size df ratio alpha conf.level
## 1 chi-square one.sample   two.sided           9  8  1.44  0.05       0.95
##        beta     power
## 1 0.8565105 0.1434895
```

```r
# Power to detect a decrease in variance
power.variance.onesample(sample.size = 9
                         ,null.hypothesis.variance = 10^2
                         ,alternative.hypothesis.variance = 8^2
                         ,alpha = 0.05
                         ,alternative = "two.sided")
```

```
##         test       type alternative sample.size df ratio alpha conf.level
## 1 chi-square one.sample   two.sided           9  8  0.64  0.05       0.95
##        beta      power
## 1 0.9063741 0.09362591
```

Calculating Sample Size

- The proper sample size is not an opinion
- The minimum effect size (δ) to be detected
- That is, the smallest degree of shift in the parameter that the researcher wishes to identify
- The number of treatment levels (groups)
- Population variance
- Probability of committing a Type I and Type II error

Sample size for two sample tests of Means

- Assumptions
  - σ is unknown
  - Continuous data, independent samples
  - 2 normal distributions
  - Non-directional tests
- we have to solve iteratively to find the smallest value for a formula

Example

- $\alpha = 0.05$
- $\beta = 0.02$
- $\delta = 1$
- $sigma = 2$
- What is the appropriate sample size?

```
alpha<-0.05
beta<-0.02
deltamu<-1
sd<-2
sample.size.mean.t.onesample(effect.size = 1
                             , variance.est = 2^2
                             , alpha = 0.05
                             , beta = 0.02
                             , alternative = "two.sided")
```

```
##   test       type alternative sample.size actual df effect.size variance alpha
## 1    t one.sample   two.sided          67     67 66           1        4  0.05
##   conf.level beta     power
## 1       0.95 0.02 0.9808893
```

Sample Size for test of Variance

- For non directional test, we must consider two cases
  - One in which the variance increases
  - One in which the variance decreases

```
# Sample Size Calculations for Changes in Variance ---------------------------
# Sample Size, one sample
sigma0<-2 # 4 variance
sigma1<-3 # 9 variance
sigma2<-1 # 1 variance
alpha<-0.05
beta<-0.02

# Two sided, one sample (testing for a change in either direction)
# In this situation, I use the two.sided test for both the larger and smaller variance,
# and take the larger sample size of the two.
sample.size.variance.onesample(null.hypothesis.variance = sigma0^2
```

```r
                              ,alternative.hypothesis.variance = sigma1^2
                              ,alpha = alpha
                              ,beta = beta
                              ,alternative = "two.sided")
```

```
##         test       type alternative sample.size df ratio alpha conf.level
## 1 chi-square one.sample   two.sided          52 51  2.25  0.05       0.95
##        beta     power
## 1 0.0188222 0.9811778
```

```r
sample.size.variance.onesample(null.hypothesis.variance = sigma0^2
                              ,alternative.hypothesis.variance = sigma2^2
                              ,alpha = alpha
                              ,beta = beta
                              ,alternative = "two.sided")
```

```
##         test       type alternative sample.size df ratio alpha conf.level
## 1 chi-square one.sample   two.sided          19 18  0.25  0.05       0.95
##         beta     power
## 1 0.01705209 0.9829479
```

Testing for Differences / Changes in Means

Independent vs Dependent Samples

- How to select the Appropriate Test for Two Samples
  - Identify the type of data associates with the Criterion Measure of interest
    * Nominal
    * Ordinal
    * Continuous
- Independent Samples
  - There are no linkage between any items in each of the two samples
  - Example
    * An admissions officer of a small college wants to compare the mean standardized test scores of applicants educated in rural high schools versus urban high schools.
- Dependent Samples
  - Each of the items within each sample are independent of every other item in the sample
  - Each item (specimen) in one group is linked or related to a corresponding item in the other sample
  - Can be due to
    * repeated measures
      · two sets of data represent repeated measures (pairs of observations) from a single sample (dependent by nature)
    * Matching / Pairing
      · The two samples are dependent by design, based on paired or grouped testing through time, or upon a pretest or covariate
  - Example
    * an analyst for an educational testing service wants to compare the mean GMAT scores of students before and after taking a GMAT review course.

Two Independent Sample Tests for Means

- Unknown variances (t test)
  - When they are both equal
  - When they are unequal

When we assume equal variance

- $H_0 : \mu_1 = \mu_2$
- $H_0 : \mu_1 \neq \mu_2$

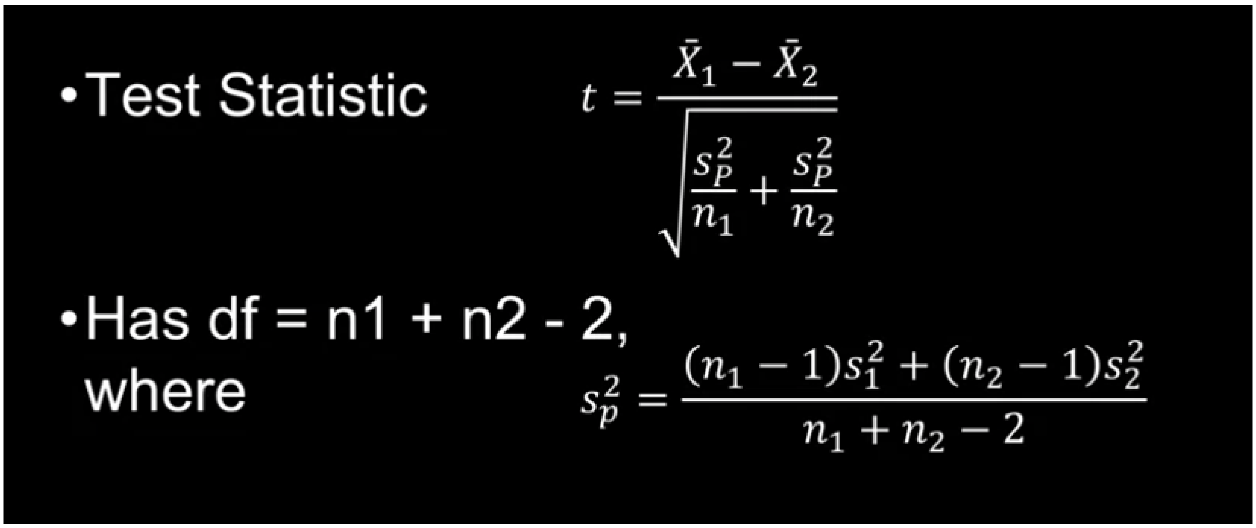


Figure 3: Equal Variance Formula

```
CapPull2 <- read.delim("~/Documents/GitHub/school_cu/school_cu/methods for quality improvement/DTSA5704_
```

```
ro<-round.object
nqtr<-function(x,d){noquote(t(round.object(x, d)))}
options(scipen=999)

# Two Sample t Test, Equal Variance ---------------------------------------

# Use simple when all parameter estimates are given
ro(t.test.twosample.independent.simple(sample.mean.g1 = 0.0060
                                    ,sample.variance.g1 = 0.0015^2
                                    ,sample.size.g1 = 25
                                    ,sample.mean.g2 = 0.0090
                                    ,sample.variance.g2 = 0.0013^2
                                    ,sample.size.g2 = 30
                                    ,conf.level = 0.95),6)
```

```
##
##  Two-Sample t Test For Means (Equal Variances)
##
## data:  input sample means and variances
```

```
## t statistic = -7.9464, null hypothesis difference = 0, p-value <
## 0.00000000000000022
## alternative hypothesis: true difference of means is not equal to 0
## 95 percent confidence interval:
##  -0.003757 -0.002243
## sample estimates:
##                diff              se.est                  df             g1.mean
##           -0.003000            0.000378           53.000000            0.006000
##     g1.mean.lowerci     g1.mean.upperci      g1.sample.size              g1.var
##            0.005381            0.006619           25.000000            0.000002
##      g1.var.lowerci      g1.var.upperci               g1.sd       g1.sd.lowerci
##            0.000001            0.000004            0.001500            0.001171
##       g1.sd.upperci             g2.mean     g2.mean.lowerci     g2.mean.upperci
##            0.002087            0.009000            0.008515            0.009485
##      g2.sample.size              g2.var      g2.var.lowerci      g2.var.upperci
##           30.000000            0.000002            0.000001            0.000003
##               g2.sd       g2.sd.lowerci       g2.sd.upperci var.test.conf.level
##            0.001300            0.001035            0.001748            0.950000
##          var.test.F      var.test.df.g1      var.test.df.g2          var.test.p
##            1.331361           24.000000           29.000000            0.458792
```

```r
# CapPull2.dat file

# Use when you have a data file available
t.test.twosample.independent(g1 = CapPull2$pull[CapPull2$mold==1]
                             ,g2 = CapPull2$pull[CapPull2$mold==2])
```

```
##
##  Two-Sample t Test For Means (Equal Variances)
##
## data:  input sample means and variances
## t statistic = -1.8865, null hypothesis difference = 0, p-value =
## 0.06529
## alternative hypothesis: true difference of means is not equal to 0
## 95 percent confidence interval:
##  -0.38010991  0.01210991
## sample estimates:
##                diff              se.est                  df             g1.mean
##         -0.18400000          0.09753632         48.00000000          1.73200000
##     g1.mean.lowerci     g1.mean.upperci      g1.sample.size              g1.var
##          1.59524174          1.86875826         25.00000000          0.10976667
##      g1.var.lowerci      g1.var.upperci               g1.sd       g1.sd.lowerci
##          0.06692396          0.21243191          0.33131053          0.25869666
##       g1.sd.upperci             g2.mean     g2.mean.lowerci     g2.mean.upperci
##          0.46090336          1.91600000          1.76828099          2.06371901
##      g2.sample.size              g2.var      g2.var.lowerci      g2.var.upperci
##         25.00000000          0.12806667          0.07808134          0.24784798
##               g2.sd       g2.sd.lowerci       g2.sd.upperci var.test.conf.level
##          0.35786403          0.27943039          0.49784333          0.95000000
##          var.test.F      var.test.df.g1      var.test.df.g2          var.test.p
##          0.85710567         24.00000000         24.00000000          0.70870116
```

```r
# Make factors
str(CapPull2)
```

```
## 'data.frame':    50 obs. of  2 variables:
##  $ mold: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pull: num  2.5 2 1.6 1.2 1.4 1.3 1.7 1.9 1.8 2.3 ...
```

```r
CapPull2$mold<-as.factor(CapPull2$mold)
str(CapPull2)
```

```
## 'data.frame':    50 obs. of  2 variables:
##  $ mold: Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
##  $ pull: num  2.5 2 1.6 1.2 1.4 1.3 1.7 1.9 1.8 2.3 ...
```

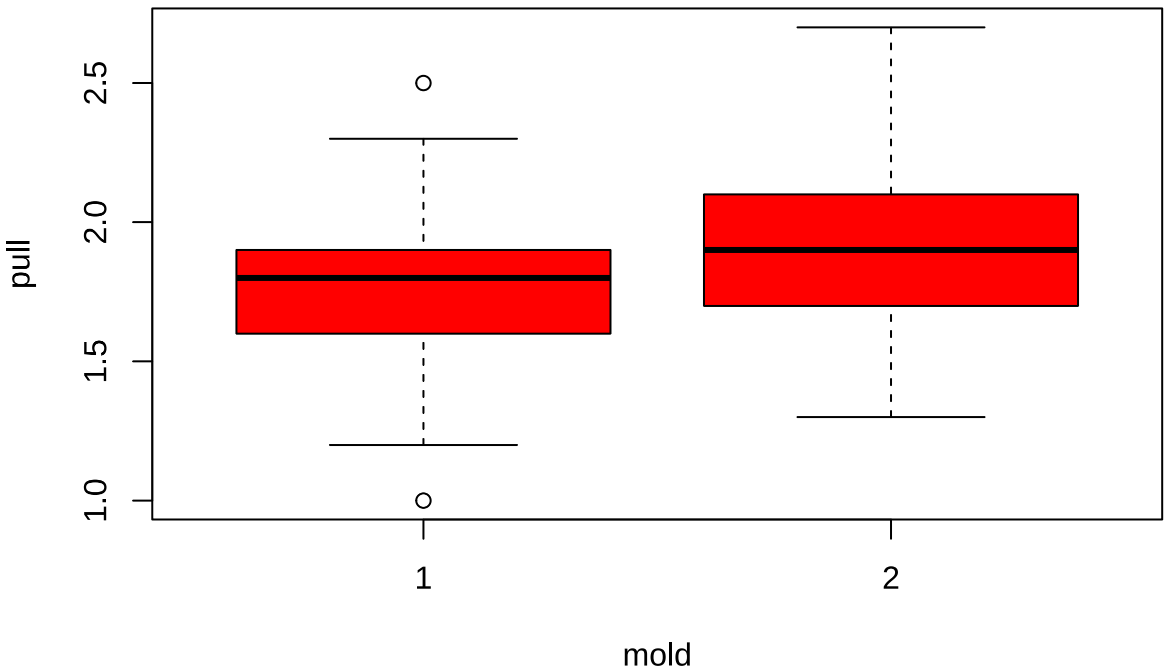
```r
# Descriptive Summary
summary.continuous(fx = pull~mold, data = CapPull2)
```

```
##   mold  n missing  mean       var g3.skewness  g3test.p g4.kurtosis  g4test.p
## 1    1 25       0 1.732 0.1097667  0.03023537 0.9447432   0.6420715 0.3709109
## 2    2 25       0 1.916 0.1280667  0.31460949 0.4752471  -0.1107518 0.9235752
```

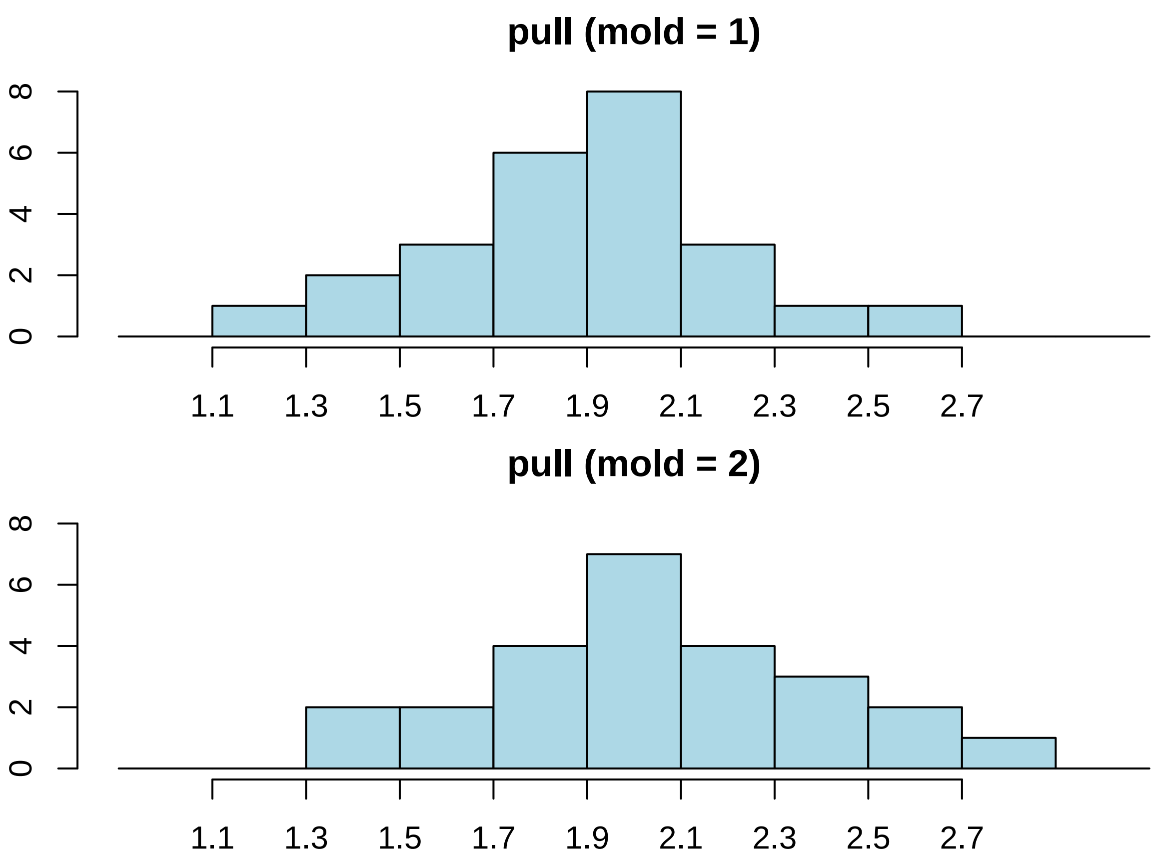
```r
# Use when you want to use a formula of y~x and have
# a data file with factors
t.test.twosample.independent.fx(fx = pull~mold
                                ,data = CapPull2)
```

```
##
##  Two-Sample t Test For Means (Equal Variances)
##
## data:  input sample means and variances
## t statistic = -1.8865, null hypothesis difference = 0, p-value =
## 0.06529
## alternative hypothesis: true difference of means is not equal to 0
## 95 percent confidence interval:
##  -0.38010991  0.01210991
## sample estimates:
##                diff              se.est                  df             g1.mean
##         -0.18400000          0.09753632         48.00000000          1.73200000
##     g1.mean.lowerci     g1.mean.upperci      g1.sample.size              g1.var
##          1.59524174          1.86875826         25.00000000          0.10976667
##      g1.var.lowerci      g1.var.upperci               g1.sd       g1.sd.lowerci
##          0.06692396          0.21243191          0.33131053          0.25869666
##       g1.sd.upperci             g2.mean     g2.mean.lowerci     g2.mean.upperci
##          0.46090336          1.91600000          1.76828099          2.06371901
##      g2.sample.size              g2.var      g2.var.lowerci      g2.var.upperci
##         25.00000000          0.12806667          0.07808134          0.24784798
##               g2.sd       g2.sd.lowerci       g2.sd.upperci var.test.conf.level
##          0.35786403          0.27943039          0.49784333          0.95000000
##          var.test.F      var.test.df.g1      var.test.df.g2          var.test.p
##          0.85710567         24.00000000         24.00000000          0.70870116
```

```r
# Data visualization
boxplot(pull~mold, data = CapPull2, col="red")
```




```r
process.group.plot(fx = pull~mold,data = CapPull2)
```



##

Two Sample of Unequal Variance t Test for Means

- The samples are randomly selected from two independent population or processed
- The underlying processes are normally distributed -The population or process variances are not equal

```
ro(t.test.twosample.independent.simple(sample.mean.g1 = 75
                                      ,sample.variance.g1 = 20^2
                                         ,sample.size.g1 = 12
                                         ,sample.mean.g2 = 82
                                         ,sample.variance.g2 = 9^2
                                         ,sample.size.g2 = 12
                                         ,conf.level = 0.90),4)
```

```
##
##  Two-Sample t Test For Means (Unequal Variances)
##
## data:  input sample means and variances
## t statistic = -1.1056, null hypothesis difference = 0, p-value = 0.286
## alternative hypothesis: true difference of means is not equal to 0
## 90 percent confidence interval:
##  -18.0855   4.0855
## sample estimates:
##                diff              se.est                  df             g1.mean
##             -7.0000              6.3311             15.2795             75.0000
##     g1.mean.lowerci     g1.mean.upperci      g1.sample.size              g1.var
##             64.6315             85.3685             12.0000            400.0000
##      g1.var.lowerci      g1.var.upperci               g1.sd      g1.sd.lowerci
##            223.6325            961.7879             20.0000             14.9543
##       g1.sd.upperci             g2.mean     g2.mean.lowerci     g2.mean.upperci
##             31.0127             82.0000             77.3342             86.6658
##      g2.sample.size              g2.var      g2.var.lowerci      g2.var.upperci
##             12.0000             81.0000             45.2856            194.7621
##               g2.sd      g2.sd.lowerci       g2.sd.upperci var.test.conf.level
##              9.0000              6.7295             13.9557              0.9000
##          var.test.F     var.test.df.g1      var.test.df.g2          var.test.p
##              4.9383             11.0000             11.0000              0.0135
```

Two Dependent Sample Tests for Means

Repeated Measures t test for Means

- Used to compare means of repeated measures or paired groups
- Tests tests the following hypotheses

$H_0 : \mu_1 = \mu_2$ or $H_0 : \mu_d = 0$

$H_0 : \mu_1 \neq \mu_2$ or $H_0 : \mu_d \neq 0$

Assumptions

Repeated Measures

- the n pairs of scores are independent of one another
- The population for difference scores is normally distributed, as are the populations for each group

Matched Pairs

- The specimens in the two samples are independent (by nature)
- The population for difference scores is normally distributed, as are the populations for each group
- Homogenity of varaince is assumed (Not critical if sample sizes are equal)
- The units or specimens in the two samples are dependent by design

```
Noise <- read.delim("~/Documents/GitHub/school_cu/school_cu/methods for quality improvement/DTSA5704_De
```

```
# Paired t test for Means (Dependent by Nature) -------------------------------------------------
summary.continuous(Noise)[-1,]
```

```
##   dv.name  n missing mean       var adtest.AA  adtest.p  swtest.W  swtest.p
## 2     Old 10       0 17.3  76.67778 0.4022606 0.3581241 0.9146568 0.3145334
## 3     New 10       0 20.9 101.65556 0.4821423 0.2308047 0.8970475 0.2032742
```

```
# Calculate difference between Old and New
Noise$Diff<-Noise$Old-Noise$New

summary.continuous(Noise)[-1,]
```

```
##   dv.name  n missing mean       var adtest.AA  adtest.p  swtest.W  swtest.p
## 2     Old 10       0 17.3  76.67778 0.4022606 0.3581241 0.9146568 0.3145334
## 3     New 10       0 20.9 101.65556 0.4821423 0.2308047 0.8970475 0.2032742
## 4    Diff 10       0 -3.6  24.48889 0.4696272 0.2475374 0.9003229 0.2208915
```

```
# Drop first and fourth column
Noise<-Noise[-c(1,4)]

# Transpose data
Noise.I<-transform.dependent.format.to.independent.format(data = Noise)
Noise.I
```

```
##    cell measure
## 1   Old      24
## 2   Old      22
## 3   Old      28
## 4   Old       8
## 5   Old       7
## 6   Old      23
## 7   Old      14
## 8   Old      27
## 9   Old       4
## 10  Old      16
## 11  New      27
## 12  New      18
## 13  New      36
## 14  New      12
## 15  New       9
## 16  New      26
```

```
## 17  New      18
## 18  New      37
## 19  New      14
## 20  New      12
```

```
str(Noise.I)
```

```
## 'data.frame':    20 obs. of  2 variables:
##  $ cell   : chr  "Old" "Old" "Old" "Old" ...
##  $ measure: int  24 22 28 8 7 23 14 27 4 16 ...
```

```
Noise.I$cell<-as.factor(Noise.I$cell)

# Paired t test
t.test.twosample.dependent(x1 = Noise$Old
                           ,x2 = Noise$New)
```

```
##
##  Dependent Samples t Test for Means (D-bar method)
##
## data:  sample mean, sample size, and estimated variance
## t statistic = -2.3005, null hypothesis mean = 0, p-value = 0.04696
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -7.14003303 -0.05996697
## sample estimates:
## sample.mean      se.est          df var.lowerci         var var.upperci
##  -3.6000000   1.5648926   9.0000000  11.5861163  24.4888889  81.6178555
##  sd.lowerci          sd  sd.upperci       power
##   3.4038385   4.9486249   9.0342601   0.5370023
```

```
# Calculate difference between Old and New
Noise$Diff<-Noise$Old-Noise$New

# Dbar method
t.test.twosample.dependent.simple.dbar(pair.differences.mean = mean(Noise$Diff)
                                       ,pair.differences.variance = var(Noise$Diff)
                                       ,sample.size = 10)
```


```
##
##  Dependent Samples t Test for Means (D-bar method)
##
## data:  sample mean, sample size, and estimated variance
## t statistic = -2.3005, null hypothesis mean = 0, p-value = 0.04696
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -7.14003303 -0.05996697
## sample estimates:
## sample.mean      se.est          df var.lowerci         var var.upperci
##  -3.6000000   1.5648926   9.0000000  11.5861163  24.4888889  81.6178555
##  sd.lowerci          sd  sd.upperci       power
##   3.4038385   4.9486249   9.0342601   0.5370023
```

```r
# Calculate the Pearson Product Moment Correlation Coefficient
cor(Noise$Old, Noise$New)
```

```
## [1] 0.8712675
```

```r
cor.pearson.r.onesample(x = Noise$Old, y = Noise$New)
```

```
##
##  One-Sample Test for Pearson Product Moment Correlation
##
## data:  sample r and sample size
## t.statistic = 5.0209, null hypothesis correlation = 0, p-value =
## 0.001026
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5352868 0.9692160
## sample estimates:
##    sample.r          df sample.size   r.squared z_r.lowerci         z_r
##   0.8712675   8.0000000  10.0000000   0.7591070   0.5975205   1.3383173
## z_r.upperci       power
##   2.0791140   0.9430485
```

```r
# Matched Pairs t test (Dependendent by Design) ---------------------------
cor.pearson.r.onesample.simple(sample.r = 0.60, sample.size = 30)
```

```
##
##  One-Sample Test for Pearson Product Moment Correlation
##
## data:  sample r and sample size
## t.statistic = 3.9686, null hypothesis correlation = 0, p-value =
## 0.0004571
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.305840 0.789587
## sample estimates:
##    sample.r          df sample.size   r.squared z_r.lowerci         z_r
##   0.6000000  28.0000000  30.0000000   0.3600000   0.3159519   0.6931472
## z_r.upperci       power
##   1.0703424   0.9496775
```

```r
ro(t.test.twosample.dependent.simple.meandiff(sample.mean.g1 = 35.24
                                              ,sample.mean.g2 = 38.02
                                              ,sample.variance.g1 = 5.18^2
                                              ,sample.variance.g2 = 5.63^2
                                              ,sample.size = 30
                                              ,rho.estimate = 0.60),4)
```

```
##
##  Dependent Samples t Test For Means - Difference of Means Method (Equal
##  Variances)
##
```

```
## data:  input sample means, variances, and correlation estimate
## t statistic = -3.1388, null hypothesis difference = 0, p-value = 0.0039
## alternative hypothesis: true difference of means is not equal to 0
## 95 percent confidence interval:
##  -4.5914 -0.9686
## sample estimates:
##               diff             se.est                 df        sample.size
##            -2.7800             0.8857            29.0000            30.0000
##            g1.mean     g1.mean.lowerci    g1.mean.upperci             g1.var
##            35.2400            33.3058            37.1742            26.8324
##     g1.var.lowerci     g1.var.upperci              g1.sd      g1.sd.lowerci
##            17.0188            48.4911             5.1800             4.1254
##      g1.sd.upperci            g2.mean    g2.mean.lowerci    g2.mean.upperci
##             6.9636            38.0200            35.9177            40.1223
##             g2.var     g2.var.lowerci     g2.var.upperci              g2.sd
##            31.6969            20.1042            57.2821             5.6300
##      g2.sd.lowerci      g2.sd.upperci var.test.conf.level         var.test.t
##             4.4838             7.5685             0.9500            -0.5516
##        var.test.df         var.test.p
##            28.0000             0.5856
```

Testing for Differences / Changes in Variance

Two Sample Tests for Variances

Assumptions: F Test for Variances

- The samples are randomly selected from two independent populations or processes
- The underlying processes are normally distributed
- The F Test for Variances is not performed when using a t test for means

```
# Two Independent Sample F Test for Variances -----------------------------------------

ro(variance.test.twosample.independent.simple(sample.variance.g1 = 0.0015^2
                                              ,sample.size.g1 = 25
                                              ,sample.variance.g2 = 0.0013^2
                                              ,sample.size.g2 = 30),7)
```

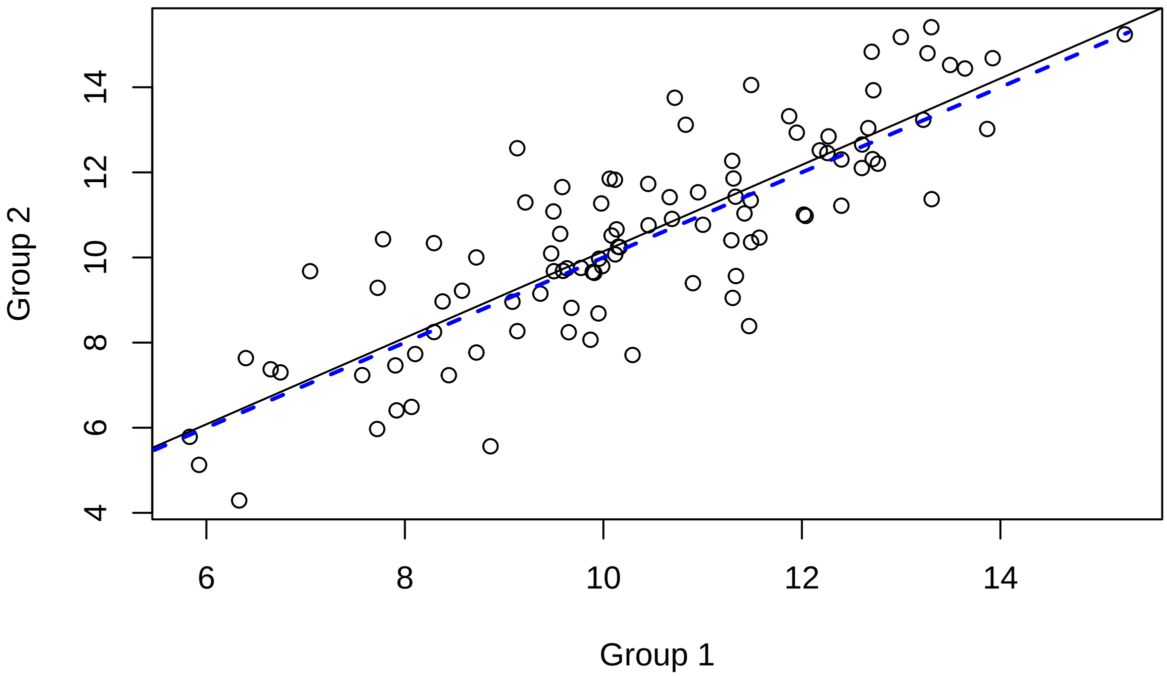
```
##
##  Two-Sample F Test For Variance
##
## data:  input variances and sample sizes
## F statistic = 1.3314, variance ratio = 1, p-value = 0.4588
## alternative hypothesis: true variance ratio is not equal to 1
## sample estimates:
## sample.variance.g1              df.g1     sample.size.g1 sample.variance.g2
##          0.0000022         24.0000000         25.0000000          0.0000017
##              df.g2     sample.size.g2              power
##         29.0000000         30.0000000          0.1080484
```

```r
# Compare results to t test
ro(t.test.twosample.independent.simple(sample.mean.g1 = 0.0060
                                      ,sample.variance.g1 = 0.0015^2
                                      ,sample.size.g1 = 25
                                      ,sample.mean.g2 = 0.0090
                                      ,sample.variance.g2 = 0.0013^2
                                      ,sample.size.g2 = 30
                                      ,conf.level = 0.95),7)
```

```
##
##  Two-Sample t Test For Means (Equal Variances)
##
## data:  input sample means and variances
## t statistic = -7.9464, null hypothesis difference = 0, p-value <
## 0.000000000000000022
## alternative hypothesis: true difference of means is not equal to 0
## 95 percent confidence interval:
##  -0.0037572 -0.0022428
## sample estimates:
##                 diff              se.est                  df            g1.mean
##           -0.0030000           0.0003775          53.0000000          0.0060000
##      g1.mean.lowerci     g1.mean.upperci      g1.sample.size             g1.var
##            0.0053808           0.0066192          25.0000000          0.0000022
##       g1.var.lowerci      g1.var.upperci               g1.sd      g1.sd.lowerci
##            0.0000014           0.0000044           0.0015000          0.0011712
##        g1.sd.upperci             g2.mean     g2.mean.lowerci    g2.mean.upperci
##            0.0020867           0.0090000           0.0085146          0.0094854
##       g2.sample.size              g2.var      g2.var.lowerci     g2.var.upperci
##           30.0000000           0.0000017           0.0000011          0.0000031
##                g2.sd       g2.sd.lowerci       g2.sd.upperci var.test.conf.level
##            0.0013000           0.0010353           0.0017476          0.9500000
##           var.test.F      var.test.df.g1      var.test.df.g2         var.test.p
##            1.3313609          24.0000000          29.0000000          0.4587921
```

```r
#ISO Plot to Compare Variances
# ISO Plot Example for Variances
g1<-rnorm(n = 100, mean = 10, sd = 2)
g2<-g1+rnorm(n = 100, mean = 0, sd = runif(1,0,2))
plot(g1, g2, xlab = "Group 1", ylab = "Group 2")
abline(lm(g2~g1))

# Add ISO line
min <- min(range(g1), range(g2))
max <- max(range(g1), range(g2))
lines(x = min:max, y = min:max, lwd = 2, lty = 2, col = "blue")
```



The Dependent Sample T-test for Variances #### Assumptions - The pairs of scores are independent of another - The sample data are either dependent by nature, or dependent by design (cirtical, therefore you may be required to test correlation) - The underlying process distributions are normally distributed (not critical if n is large)

```
# Matched Pairs t test for Variances --------------------------------------
cor.pearson.r.onesample.simple(sample.r = 0.60, sample.size = 30)
```

```
##
##  One-Sample Test for Pearson Product Moment Correlation
##
## data:  sample r and sample size
## t.statistic = 3.9686, null hypothesis correlation = 0, p-value =
## 0.0004571
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.305840 0.789587
## sample estimates:
##    sample.r          df sample.size   r.squared z_r.lowerci         z_r
##   0.6000000  28.0000000  30.0000000   0.3600000   0.3159519   0.6931472
## z_r.upperci       power
##   1.0703424   0.9496775
```

```
variance.test.twosample.dependent.simple(sample.variance.g1 = 5.18^2
                                         ,sample.variance.g2 = 5.63^2
                                         ,sample.size = 30
                                         ,rho.estimate = 0.60
                                         ,conf.level = 0.95)
```

```
##
##  Two Dependent Sample t Test For Variance
##
```

```
## data:  dependent sample variances, sample size, and r
## t statistic = -0.55164, variance difference = 0, p-value = 0.5856
## alternative hypothesis: true variance difference is not equal to 0
## 95 percent confidence interval:
##  -22.92773  13.19873
## sample estimates:
## sample.variance.g1 sample.variance.g2   pearson.estimate        sample.size
##            26.8324            31.6969             0.6000            30.0000
##                 df
##            28.0000
```

Testing for differences / changes in Proportions

Fisher's Exact Test

Assumptions

- The two processes from which the sample data are drawn are inherently independent in nature, and are both based upon the Bernoullis process
- The samples are randomly selected from the underlying processes being investigated

```
# Two Sample Tests for Proportions ----------------------------------------

# Fisher's Exact Test
proportion.test.twosample.exact.simple(sample.proportion.g1 = 0.18
                                       ,sample.size.g1 = 750
                                       ,sample.proportion.g2 = 0.12
                                       ,sample.size.g2 = 750
                                       ,conf.level = 0.99)
```

```
##
##  Two-Sample Proportion Test - Fisher Exact Test
##
## data:  sample proportions and sample sizes
## null hypothesis odds ratio = 1, p-value = 0.001425
## alternative hypothesis: true odds ratio is not equal to 1
## 99 percent confidence interval:
##  1.195212 2.174273
## sample estimates:
## sample.prop.g1 sample.size.g1    n1.times.p1    n1.times.q1   p.g1.lowerci
##      0.1800000    750.0000000    135.0000000    615.0000000      0.1453363
##   p.g1.upperci sample.prop.g2 sample.size.g2    n2.times.p2    n2.times.q2
##      0.2188118      0.1200000    750.0000000     90.0000000    660.0000000
##   p.g2.lowerci   p.g2.upperci
##      0.0912707      0.1536761
```

McNemar's Test for Change

```
# McNemar's Test for Change - Dependent Proportions -----------------------
# Contingency table format = ct<-(a,c,b,d)
```

```r
ct<-c(56,56,4,4)

# Create Contingency Table
(ct.new<-matrix(ct,nrow = 2
                , dimnames = list("Before Maint" = c("Pass", "Fail"),
                                  "After Maint" = c("Pass", "Fail"))))
```

```
##             After Maint
## Before Maint Pass Fail
##         Pass   56    4
##         Fail   56    4
```

```r
# Perform McNemar's Test
(mcnemar.out<-proportion.test.mcnemar.simple(b = 4, c = 56))
```

```
##
##  McNemar's Test for Dependent Proportions (Exact)
##
## data:  off-diagonal 2x2 elements
## p = 0.066667, null hypothesis proportion = 0.5, p-value =
## 0.0000000000009085
## alternative hypothesis: true proportion is not equal to 0.5
## sample estimates:
##          b        p_b          c        p_c
##  4.00000000  0.06666667 56.00000000  0.93333333
```

```r
mcnemar.test(ct.new)
```

```
##
##  McNemar's Chi-squared test with continuity correction
##
## data:  ct.new
## McNemar's chi-squared = 43.35, df = 1, p-value = 0.00000000004577
```

Two Sample Independent Tests for Poisson Counts

```r
Eddycur <- read.delim("~/Documents/GitHub/school_cu/school_cu/methods for quality improvement/DTSA5704_I
```

```r
# Two Sample Independent Tests for Poisson Rates (Counts) -----------------

# Descriptive Summary
summary.impl(Eddycur$Before, stat.n = T, stat.mean = T)
```

```
##   dv.name   n      mean
## 1      fx 130 0.8615385
```

```r
summary.impl(Eddycur$After, stat.n = T, stat.mean = T)
```

```
##   dv.name   n mean
## 1      fx 130    2
```

```r
# Test for Poisson distribution
poisson.dist.test(Eddycur$Before)
```

```
##
##  Poisson Distribution Fit Test Using Variance and Mean
##
## data:  input data
## chi.square = 122.46, degrees of freedom = 129, p-value = 0.7097
## alternative hypothesis: true chi.square is not equal to 129
## sample estimates:
##      chi.square sample variance     sample mean
##     122.4642857       0.8178891       0.8615385
```

```r
poisson.dist.test(Eddycur$After)
```

```
##
##  Poisson Distribution Fit Test Using Variance and Mean
##
## data:  input data
## chi.square = 147, degrees of freedom = 129, p-value = 0.2655
## alternative hypothesis: true chi.square is not equal to 129
## sample estimates:
##      chi.square sample variance     sample mean
##       147.00000         2.27907         2.00000
```

```r
# Poisson test
# Remember that sample count has to be
# n times lambda
(count_before<-sum(Eddycur$Before))
```

```
## [1] 112
```

```r
(n_before<-length(Eddycur$Before))
```

```
## [1] 130
```

```r
(count_after<-sum(Eddycur$After))
```

```
## [1] 260
```

```r
(n_after<-length(Eddycur$After))
```

```
## [1] 130
```

```r
poisson.test.twosample.simple(sample.count.g1 = 112
                              ,sample.size.g1 = 130
                              ,sample.count.g2 = 260
                              ,sample.size.g2 = 130)
```

```
##
##  Two-Sample Poisson Test
##
## data:  sample counts and sample sizes
## z = -7.6734, difference in rates = 0, p-value = 0.00000000000001674
## alternative hypothesis: true difference in rates is not equal to 0
## 95 percent confidence interval:
##  NA NA
## sample estimates:
##           rate.g1           rate.g2        lambda.hat g1.lambda.lowerci
##         0.8615385         2.0000000         1.4307692         0.7093881
## g1.lambda.upperci g2.lambda.lowerci g2.lambda.upperci
##         1.0366546         1.7642621         2.2584649
```