PSY.308d.DA4

Daniel Pinedo

** IMPORTANT NOTES: PLEASE READ BEFORE STARTING ** The structure of your write-up should be: - *Study 1 Results & Discussion:* typical results section + one paragraph discussion interpreting the results of your analyses - *Study 2 Results & Discussion:* typical results section + one paragraph discussion interpreting the results of your analyses - *Conclusion:* an INTEGRATED discussion of the two studies findings (do not simply repeat what has already been reported), implications, limitations, and future directions.

Additionally, all write-ups should be in APA format and your write-up should NOT exceed 4 pages.

Study 1: Pilot Study

```r
dat <- read.csv("https://www.dropbox.com/s/qhx6mb45e1njvk4/PSY.308d.DA4_1.csv?dl=1")

#load libraries used
library(pacman)
p_load(jmv, psych, car, effsize)

#make dataset more readable
dat$GroupNum <- factor(dat$GroupNum,
  levels = c(1,2),
  labels = c("Not Hungry", "Very Hungry"))
```

McCormick & Co. is proposing a new chartiable foundation, FeedForward, which hopes to set-up meal program interventions in small towns to fight hunger amongst under-resourced citizens. While McCormick is a large company and rather successful, they want evidence before investing more money into the programs and other ideas. They recently hired you as an (sadly unpaid) intern to generate some data and give them the scoop and you have to do it on a budget (A.K.A. no funding for participants - just for supplies). You decide to put up some flyers in a couple nearby apartment buildings to recruit volunteers to join your pilot study for a meal program - luckily you got 9 people to participate!

Research Question(s): 1. Does the meal program significantly decrease the hunger compared to those on the waitlist? 2. Does participants' weight significantly increase from baseline to the end of the program?

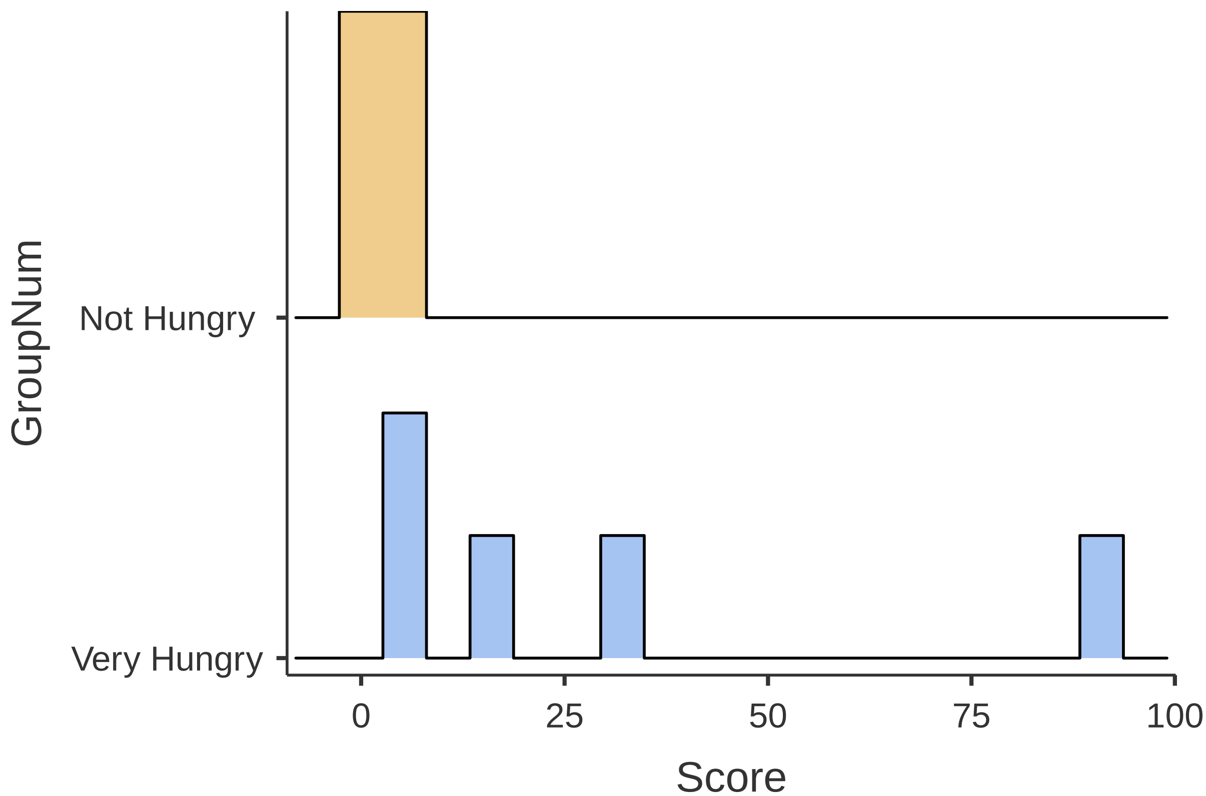
Variables: Score: self-reported hunger at the end of the program (1 = Not hungry, 100 = very hungry). Group: program assignment (GroupNum; 1 = Program, 2 = Waitlist). Pre: weight in pounds prior at baseline. Post: weight in pounds at the end of two weeks.

Question #1 Assumptions: 1. Independence of observations 2. Normal distribution of dependent variable by condition 3. Homogeneity of variance

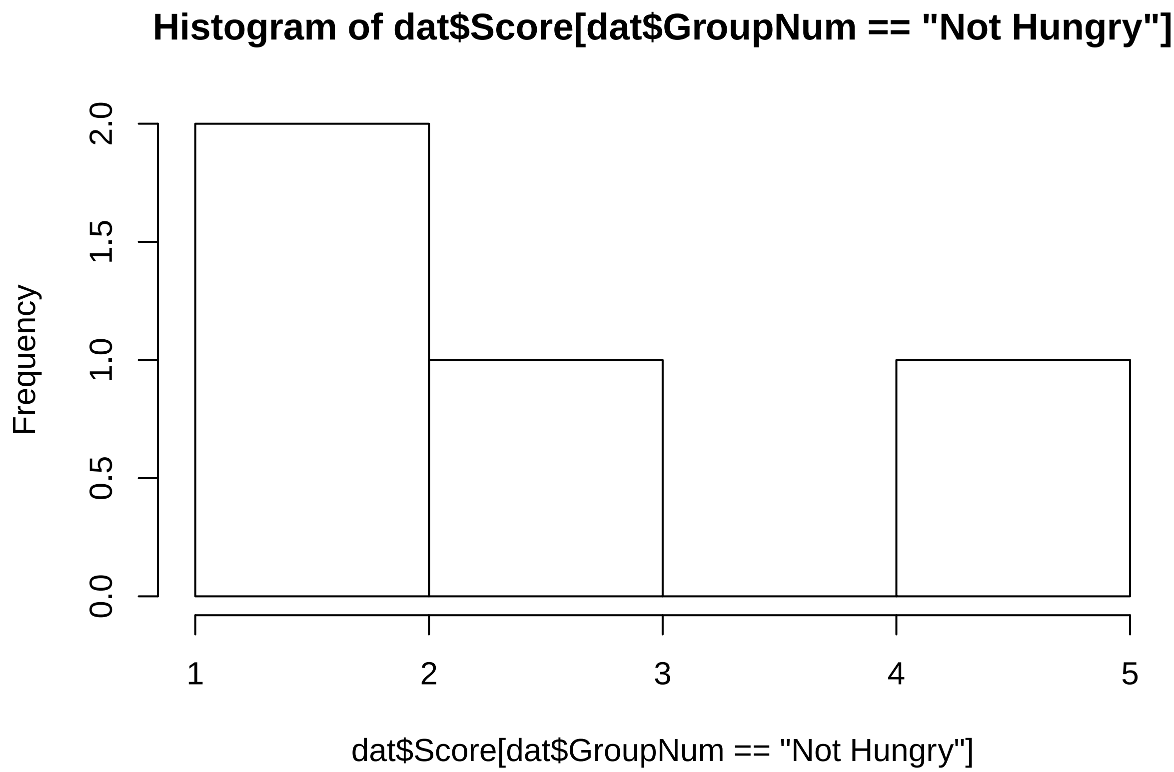
```r
# Descriptives by Condition
descriptives(data = dat,
             vars = 'Score',
             splitBy = 'GroupNum',
             sd = TRUE,
             min = TRUE,
             max = TRUE,
             skew = TRUE,
             kurt = TRUE,
             hist = TRUE)
```

```
##
##  DESCRIPTIVES
```

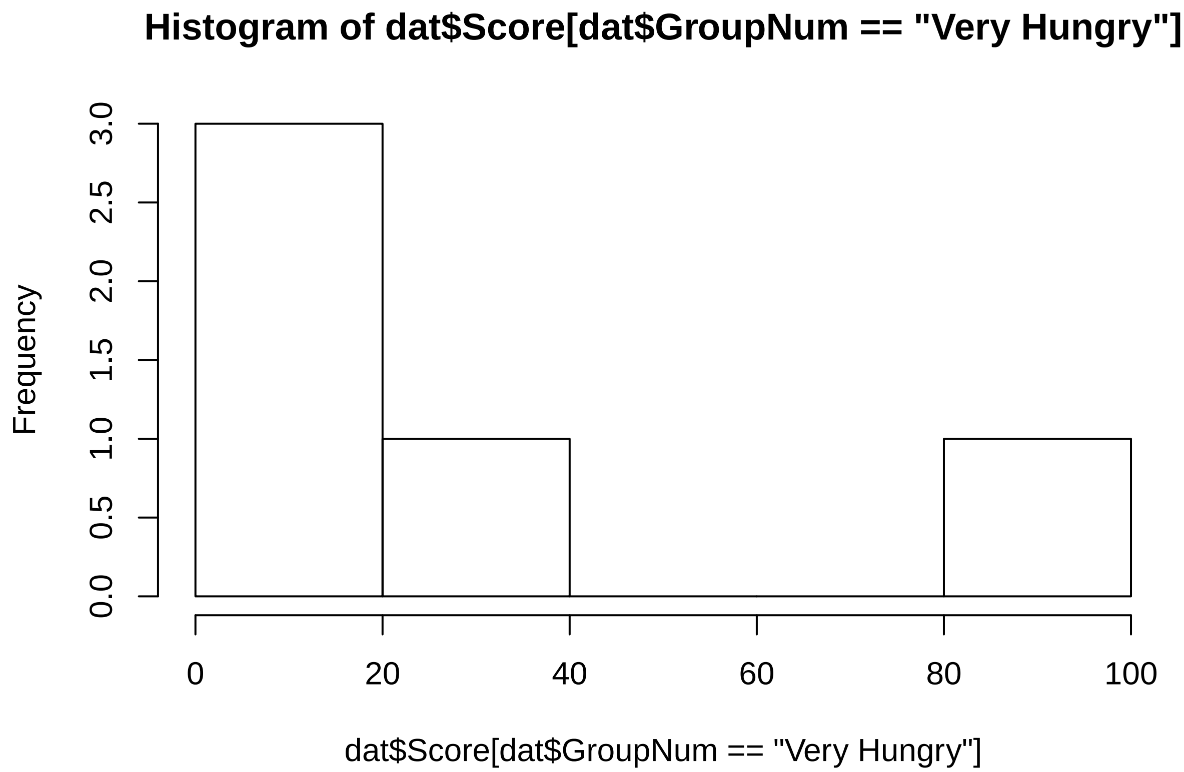
```
##
## Descriptives
## ----------------------------------------------
##                          GroupNum       Score
## ----------------------------------------------
##   N                      Not Hungry         4
##                          Very Hungry        5
##   Missing                Not Hungry         0
##                          Very Hungry        0
##   Mean                   Not Hungry      2.75
##                          Very Hungry     29.6
##   Median                 Not Hungry      2.50
##                          Very Hungry       15
##   Standard deviation     Not Hungry      1.71
##                          Very Hungry     36.5
##   Minimum                Not Hungry         1
##                          Very Hungry        4
##   Maximum                Not Hungry         5
##                          Very Hungry       92
##   Skewness               Not Hungry     0.753
##                          Very Hungry     1.80
##   Std. error skewness    Not Hungry      1.01
##                          Very Hungry    0.913
##   Kurtosis               Not Hungry     0.343
##                          Very Hungry     3.26
##   Std. error kurtosis    Not Hungry      2.62
##                          Very Hungry     2.00
## ----------------------------------------------
```




```
hist(dat$Score [dat$GroupNum== 'Not Hungry'])
```




```r
hist(dat$Score [dat$GroupNum== 'Very Hungry'])
```




```r
# Mann-Whitney test (with Levene's)
ttestIS(data = dat,
        vars = 'Score',       # DV
        group = 'GroupNum',   # IV
        desc = TRUE,
        mann = TRUE,           # Mann-Whitney U
        eqv = TRUE,            # Levene's test
        meanDiff = FALSE,
        ci = FALSE,
        effectSize = FALSE)    # Cohen's d
```

```
##
##  INDEPENDENT SAMPLES T-TEST
##
##  Independent Samples T-Test
##  ─────────────────────────────────────────────────────────
##                              statistic    df      p
##  ─────────────────────────────────────────────────────────
##    Score    Student's t         -1.45    7.00    0.190
##             Mann-Whitney U       1.00            0.032
##  ─────────────────────────────────────────────────────────
##
##
##  ASSUMPTIONS
##
##  Test of Equality of Variances (Levene's)
```

```
## ----------------------------------------
##            F      df    p
## ----------------------------------------
##   Score    4.43     1    0.073
## ----------------------------------------
##   Note. A low p-value suggests a
##   violation of the assumption of
##   equal variances
##
##
## Group Descriptives
## -----------------------------------------------------------------
##            Group          N    Mean    Median    SD      SE
## -----------------------------------------------------------------
##   Score    Not Hungry     4    2.75      2.50    1.71    0.854
##            Very Hungry    5    29.6      15.0    36.5     16.3
## -----------------------------------------------------------------
```

Effect Size

```
# Cliff's Delta (non-parametric effect size) - a more robust version of Cohen's d which considers the o

cliff.delta(Score ~ GroupNum, data = dat, conf.level = .95, magnitude = TRUE, method = "Cliff's Delta")
```

```
##
## Cliff's Delta
##
## delta estimate: -0.9 (large)
## 95 percent confidence interval:
##      lower      upper
## -0.9870433 -0.4036840
```

```
# Because this package is nice - it gives you the Delta estimate and an interpretation.
```

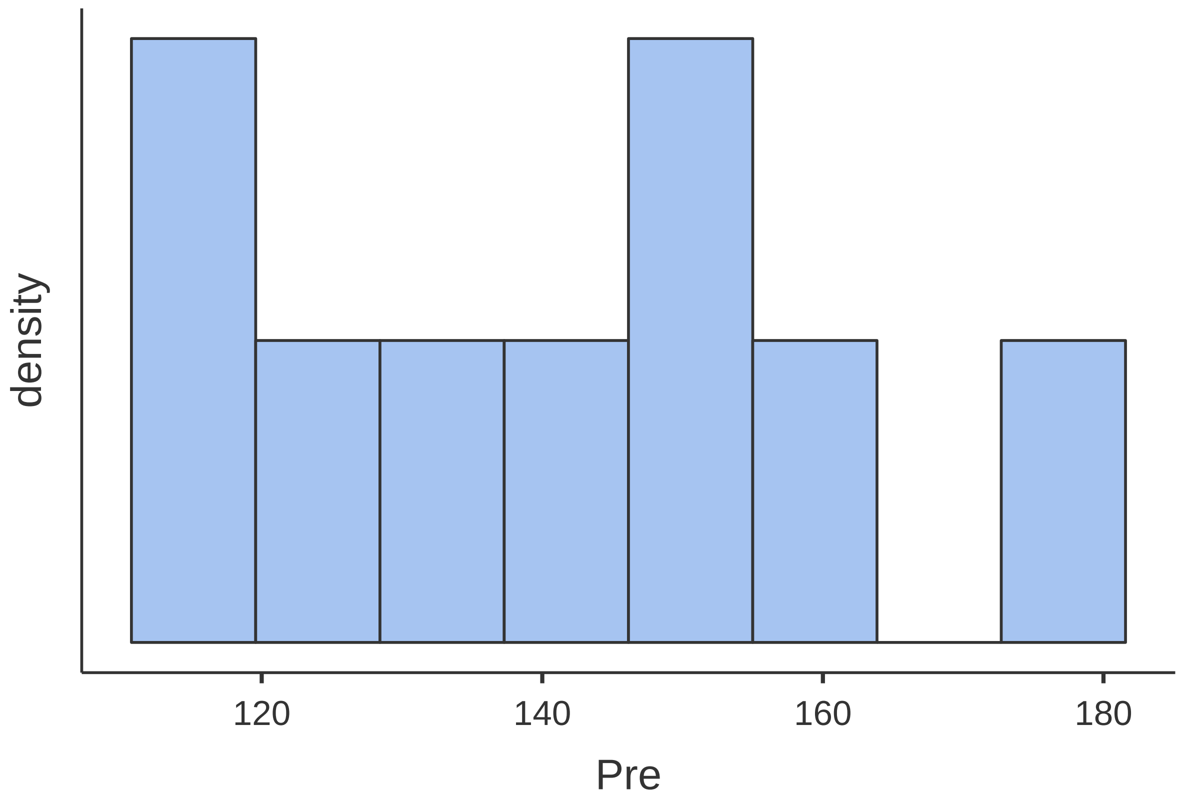
Question #2 Assumptions: 1. Independence of observations 2. Normal distribution of dependent variable by time 3. Normal distribution of difference scores (Shapiro-Wilk)

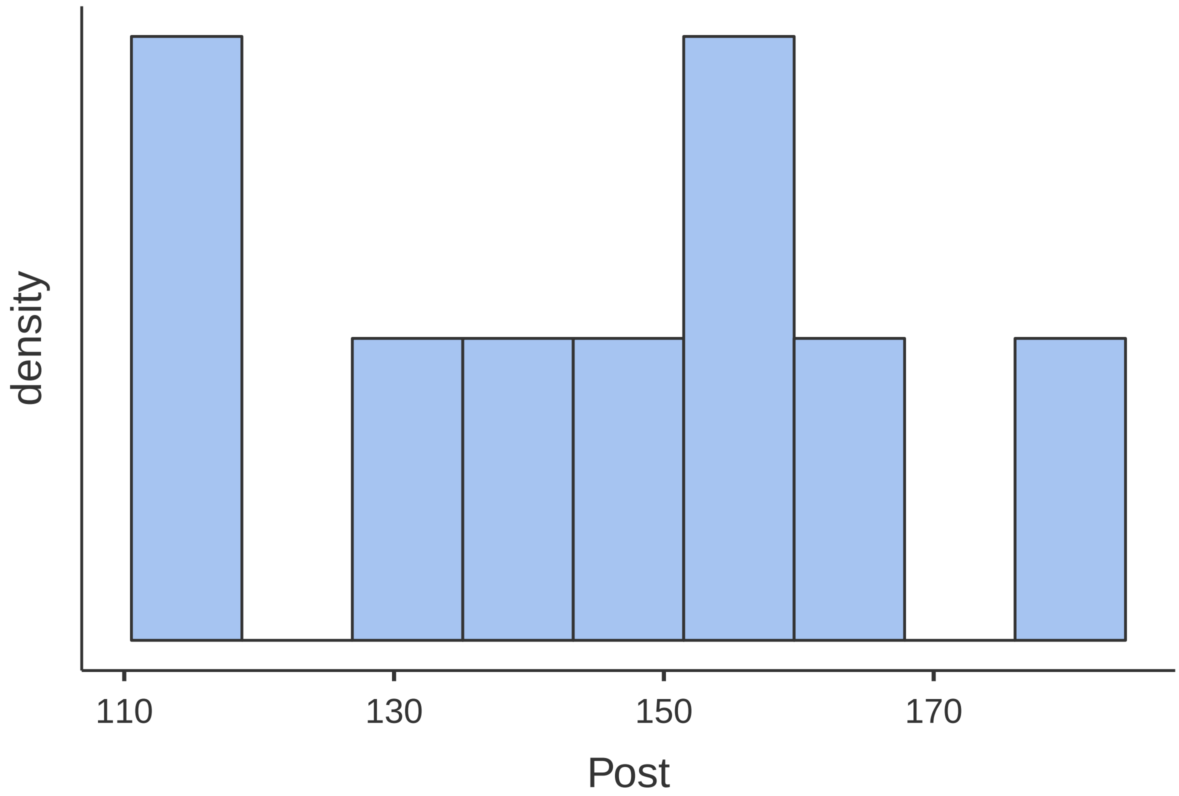
```
# Compute difference scores
dat$diff <- (dat$Post - dat$Pre)

# Descriptives
descriptives(data = dat,
             vars = c('Pre', 'Post', 'diff'),
             sd = TRUE,
             range = TRUE,
             min = TRUE,
             max = TRUE,
             skew = TRUE,
             kurt = TRUE,
             hist = TRUE)
```

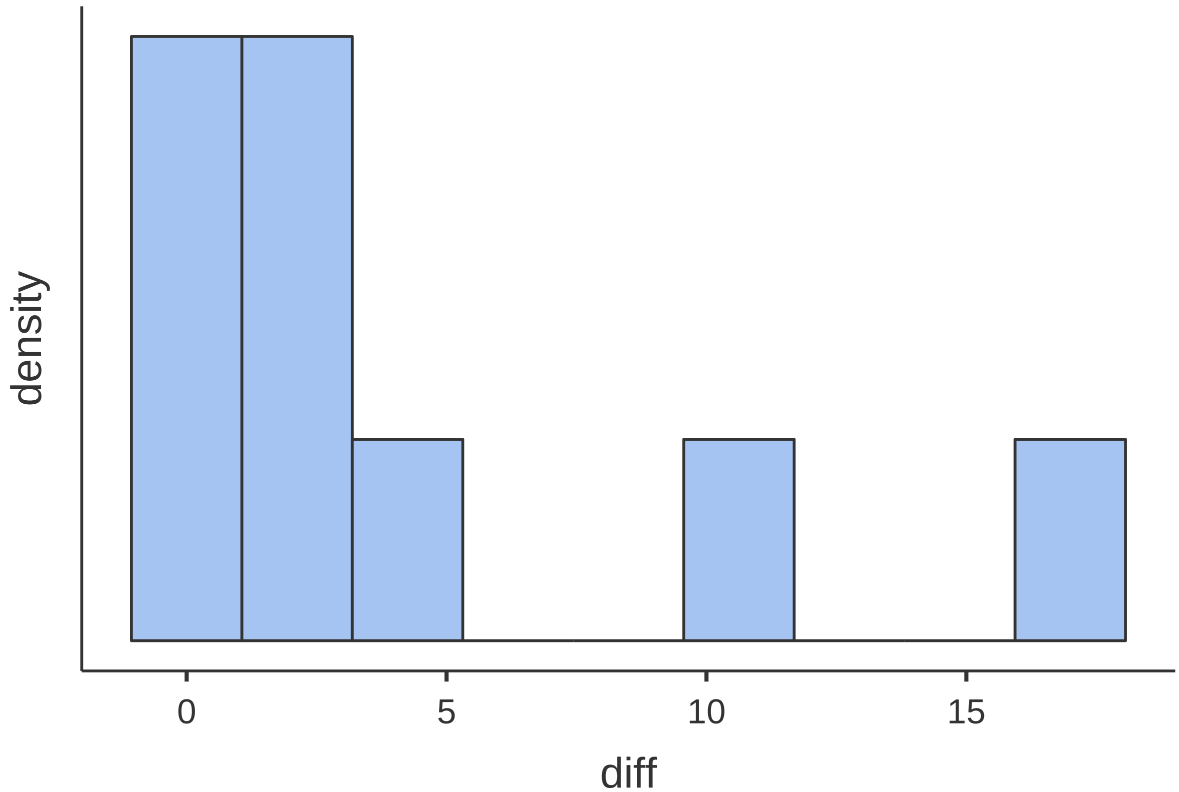
```
##
## DESCRIPTIVES
##
## Descriptives
## ----------------------------------------------------
##                          Pre      Post       diff
```

```
## ----------------------------------------------------
##    N                          9          9         9
##    Missing                    0          0         0
##    Mean                     140        144      4.12
##    Median                   141        148      2.00
##    Standard deviation      21.9       21.5      5.66
##    Range                     62       65.5      17.0
##    Minimum                  113        114     -1.00
##    Maximum                  175        179      16.0
##    Skewness               0.131    -0.0414      1.53
##    Std. error skewness    0.717      0.717     0.717
##    Kurtosis               -1.14     -0.552      1.50
##    Std. error kurtosis     1.40       1.40      1.40
## ----------------------------------------------------
```





density
110
130
150
170
Post




```r
# Wilcoxan Signed Rank Test (with Shapiro-Wilk)

#Pre/Post
ttestPS(dat,
        pairs = list(
                list(i1 = 'Pre', i2 = 'Post')),
        wilcoxon = TRUE,
        norm = TRUE,
        meanDiff = FALSE)
```

```
##
##  PAIRED SAMPLES T-TEST
##
##  Paired Samples T-Test
##  ────────────────────────────────────────────────────────────
##                                  statistic    df      p
##  ────────────────────────────────────────────────────────────
##    Pre      Post     Student's t        -2.18    8.00    0.061
##                      Wilcoxon W          3.00            0.020
##  ────────────────────────────────────────────────────────────
##
##
##  Test of Normality (Shapiro-Wilk)
##  ──────────────────────────────────────
##                        W        p
##  ──────────────────────────────────────
```

```
##    Pre    -    Post    0.806    0.024
##  --------------------------------------
##    Note. A low p-value suggests a
##    violation of the assumption of
##    normality
```

```r
#Post/Pre
ttestPS(dat,
        pairs = list(
                list(i1 = 'Post', i2 = 'Pre')),
        wilcoxon = TRUE,
        norm = FALSE,
        meanDiff = FALSE)
```

```
##
##  PAIRED SAMPLES T-TEST
##
##  Paired Samples T-Test
##  ------------------------------------------------------------
##                                 statistic    df      p
##  ------------------------------------------------------------
##    Post    Pre    Student's t       2.18    8.00    0.061
##                   Wilcoxon W        42.0            0.020
##  ------------------------------------------------------------
```

Effect Size

```r
# Effect size for non-parametric related-samples t-test
# http://yatani.jp/teaching/doku.php?id=hcistats:wilcoxonsigned
# r = z/sqrt(N) where n is the total number of observations (cases x 2)

# To get number of cases(n):
n <- dim(dat)[1] # number of cases/rows
print(paste("Number of cases/rows(n):", n))
```

```
## [1] "Number of cases/rows(n): 9"
```

```r
# To get sum of ranks(W)
W <- n*(n+1) / 2 # you can verify this by running Wilcoxon W in both directions and summing both scores
print(paste("Sum of ranks(W):", W))
```

```
## [1] "Sum of ranks(W): 45"
```

```r
# z = (W - mW + .5)/o
# https://www.statisticssolutions.com/how-to-conduct-the-wilcox-sign-test/
mW <- 0 #the null that there is no difference between T1 and T2

  #first we need to find sd
  cat("\n")
```

```r
  sd = sqrt((n*(n+1)*((2*n)+1))/6)
  print("sd = sqrt((n*(n+1)*((2*n)+1))/6)")
```

```
## [1] "sd = sqrt((n*(n+1)*((2*n)+1))/6)"
```

```r
  print(paste("sd = sqrt((", n, "*(", n, "+1)*((2*", n, ")+1))/6)", sep = ""))
```

```
## [1] "sd = sqrt((9*(9+1)*((2*9)+1))/6)"
```

```r
print(paste("sd =", round(sd,2)))
```

```
## [1] "sd = 16.88"
```

```r
# now to find z:
cat("\n")
```

```r
z = (W - mW + .5)/sd # technically this should be +- .5 dependening on directionality of W-score.
print("z = (W - mW + .5)/sd")
```

```
## [1] "z = (W - mW + .5)/sd"
```

```r
print("mW = the null that there is no difference between time 1 and time 2")
```

```
## [1] "mW = the null that there is no difference between time 1 and time 2"
```

```r
print(paste( "z = (", round(W,2), " - ", mW, " + .5)/", round(sd,2), sep = ""))
```

```
## [1] "z = (45 - 0 + .5)/16.88"
```

```r
print(paste("z =", round(z,2)))
```

```
## [1] "z = 2.7"
```

```r
#r is the Pearson value reported for the non-parametric paired-sample t-test
cat("\n")
```

```r
N = n*2 #total number of scores (2 x number of cases)
print(paste("Total number of scores(N):", N))
```

```
## [1] "Total number of scores(N): 18"
```

```r
r = z/sqrt(N)
print("r = z/sqrt(n) = Effect size = the probability of a time 2 score being greater than a time 1 scor
```

```
## [1] "r = z/sqrt(n) = Effect size = the probability of a time 2 score being greater than a time 1 sco
```

```r
print(paste("r = ", round(z,2), "/sqrt(", N, ")", sep = ""))
```

```
## [1] "r = 2.7/sqrt(18)"
```

```r
print(paste("r =", round(r,2)))
```

```
## [1] "r = 0.64"
```

```r
# Based on Cohen's criteria of evaluating effect size: .10 = small, .30 = medium, and .50 = large

# This z score above is iffy but was provided in full. R also calculates z scores for this test automag

# Wilcoxon signed rank test with z-score
wilcoxon <- stats::wilcox.test(dat$Post, dat$Pre, paired = TRUE, exact = TRUE)
wilcoxon.z <- stats::qnorm(wilcoxon$p.value) # z score estimate
cat("\n")
```

```r
print("Wilcoxon test run from library(stats):")
```

```
## [1] "Wilcoxon test run from library(stats):"
```

```r
wilcoxon
```

```
##
##  Wilcoxon signed rank test
```

```
##
## data:  dat$Post and dat$Pre
## V = 42, p-value = 0.01953
## alternative hypothesis: true location shift is not equal to 0
```

```
print(paste("Wilcoxon z-score run from library(stats):", round(wilcoxon.z, 2)))
```

```
## [1] "Wilcoxon z-score run from library(stats): -2.06"
```

```
wilcoxon.r <- abs(wilcoxon.z/sqrt(N))
print(paste("library(stats) effect size calculated as abs(", round(wilcoxon.z, 2), "/sqrt(", N, ")):
```

```
## [1] "library(stats) effect size calculated as abs(-2.06/sqrt(18)): 0.49"
```

```
# This runs what is called Exact Wilcoxon-Pratt Signed-Rank Test with a z-score
library(coin)
```

```
## Loading required package: survival
```

```
cat("\n")
```

```
wilcoxon.coin <- coin::wilcoxsign_test(dat$Post ~ dat$Pre, distribution = "exact")
wilcoxon.coin.z = as.numeric(statistic(wilcoxon.coin, type="standardized"))
names(wilcoxon.coin.z) = "Z"
print("Wilcoxon test run from library(coin):")
```

```
## [1] "Wilcoxon test run from library(coin):"
```

```
wilcoxon.coin
```

```
##
##  Exact Wilcoxon-Pratt Signed-Rank Test
##
## data:  y by x (pos, neg)
##   stratified by block
## Z = 2.3102, p-value = 0.01953
## alternative hypothesis: true mu is not equal to 0
```

```
print(paste("Wilcoxon z-score run from library(coin):", round(wilcoxon.coin.z, 2)))
```

```
## [1] "Wilcoxon z-score run from library(coin): 2.31"
```

```
wilcoxon.coin.r <- abs(wilcoxon.coin.z/sqrt(N))
print(paste("library(coin) effect size calculated as abs(", round(wilcoxon.coin.z, 2), "/sqrt(", N, "
```

```
## [1] "library(coin) effect size calculated as abs(2.31/sqrt(18)): 0.54"
```

Study 2: Intervention

```
dat2 <- read.csv("https://www.dropbox.com/s/78onrh7icd92cs4/PSY.308d.DA4_2.csv?dl=1")

library(pacman)
p_load(jmv, psych, car)
```

McCormick & Co. was impressed with your pilot study - meal programs seem to be worthwhile. They have invested $10,000 in your research account to do a bigger study where you can recruit and pay participants, gain more supplies, and extend your findings. You decide to set-up three different meal plan programs: Full Day (3 meals per day), Two-A-Day (2 meals per day), and One-A-Day (1 meal per day), and measure the average calorie intake above the noted minimum of 1,200 calories per day for each participant at the end of a two-week period. Given that these are under-resourced participants, they are encouraged to consume above the minimum calorie intake recommended by the health board.

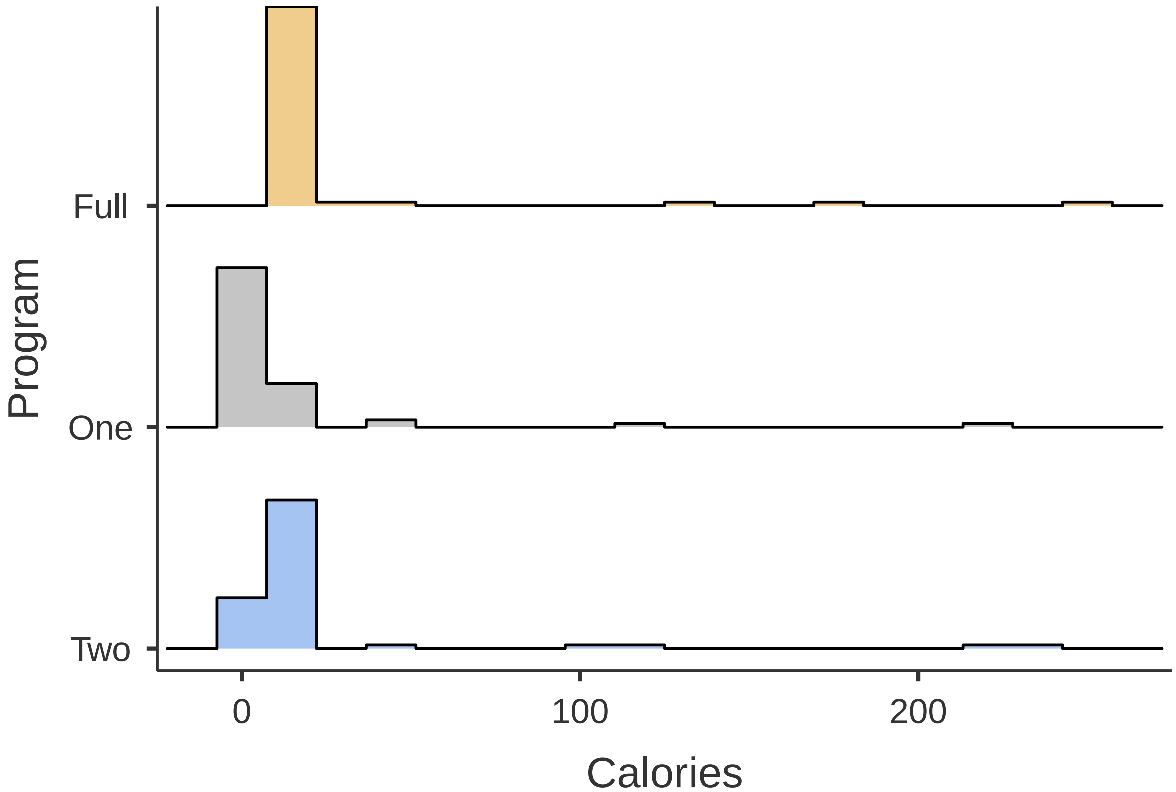
Research Question(s): 1. Is there a difference in the number of calories above the minimum consumed between the meal program conditions?

Assumptions: 1. Independence of observations - pass 2. Normal distribution of dependent variable by condition - fail 3. Homogeneity of variance - pass

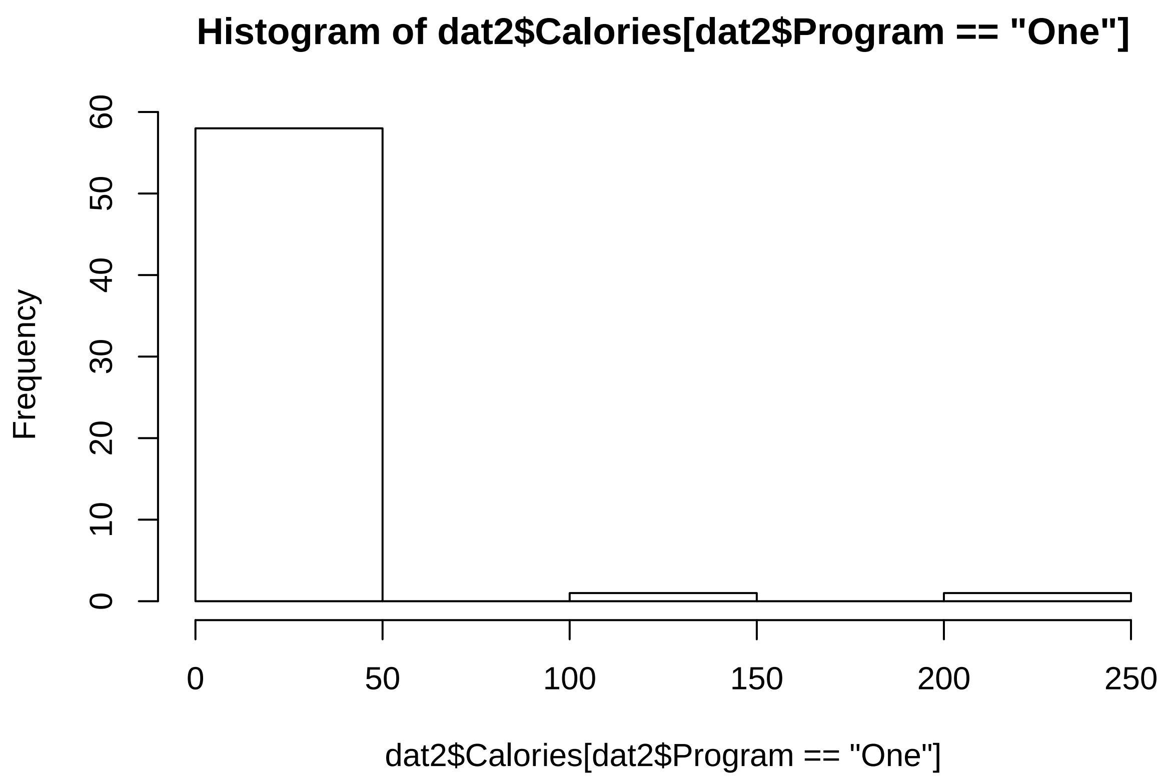
```r
# Descriptives by Team
descriptives(data = dat2,
             vars = 'Calories',
             splitBy = 'Program',
             sd = TRUE,
             min = TRUE,
             max = TRUE,
             skew = TRUE,
             kurt = TRUE,
             hist = TRUE)
```

```
##
##  DESCRIPTIVES
##
##  Descriptives
##  ----------------------------------------------
##                           Program    Calories
##  ----------------------------------------------
##    N                      Full             60
##                           One              60
##                           Two              60
##    Missing                Full              0
##                           One               0
##                           Two               0
##    Mean                   Full           25.7
##                           One            12.0
##                           Two            20.9
##    Median                 Full           17.0
##                           One            6.00
##                           Two            10.0
##    Standard deviation     Full           38.9
##                           One            31.9
##                           Two            42.9
##    Minimum                Full             12
##                           One               1
##                           Two               5
##    Maximum                Full            251
##                           One             219
##                           Two             229
##    Skewness               Full           4.70
##                           One            5.54
##                           Two            4.16
##    Std. error skewness    Full          0.309
##                           One           0.309
##                           Two           0.309
##    Kurtosis               Full           22.8
##                           One            32.9
##                           Two            17.3
##    Std. error kurtosis    Full          0.608
```

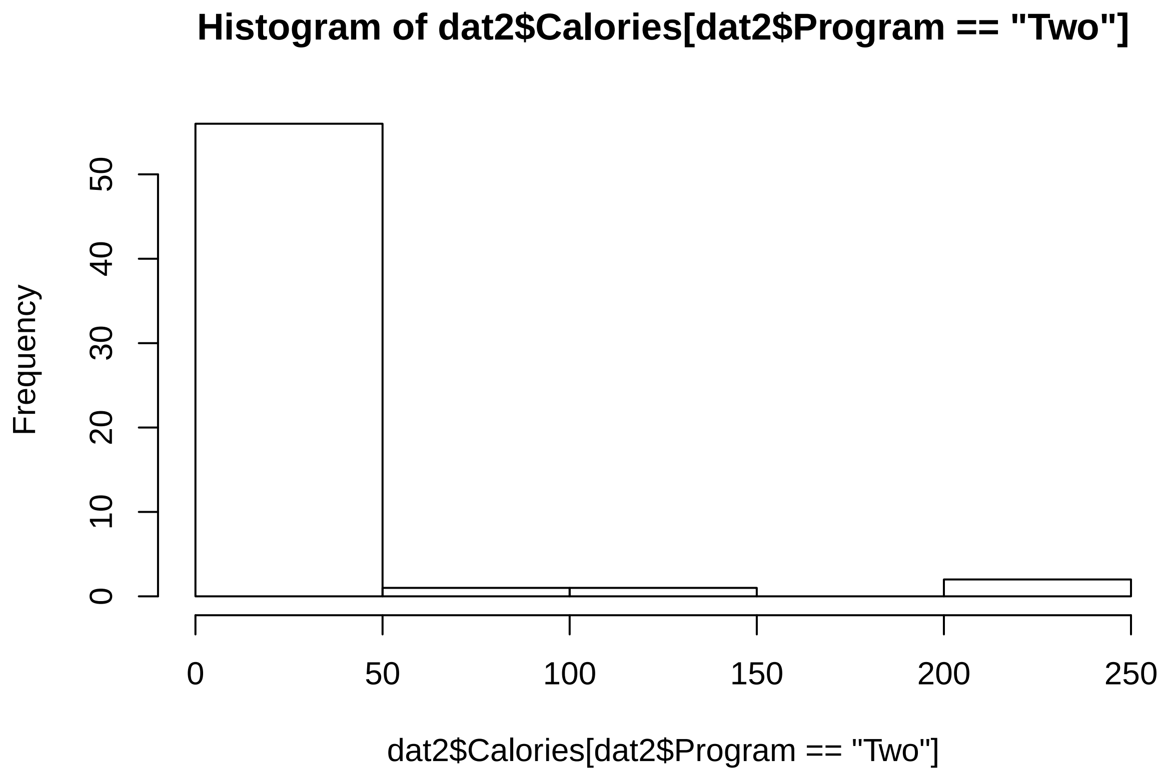
```
##                          One           0.608
##                          Two           0.608
##   ---------------------------------------------
```



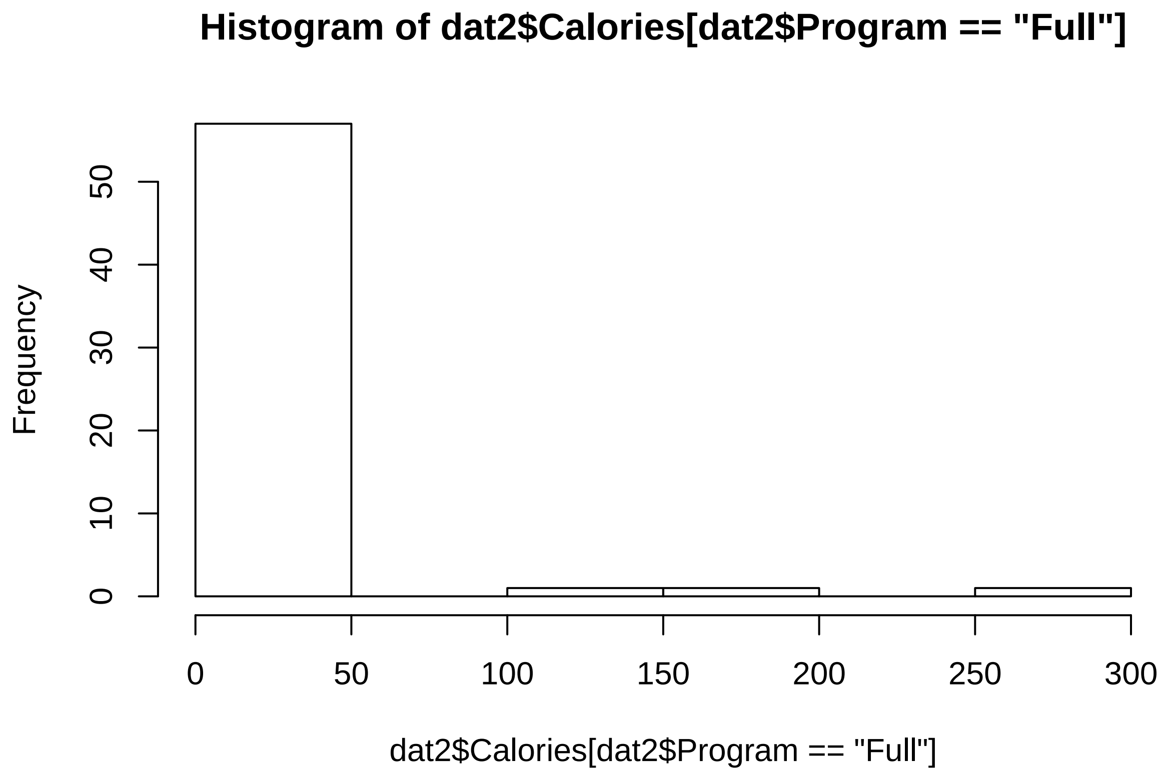

```
hist(dat2$Calories [dat2$Program== 'One'])
```




```
hist(dat2$Calories [dat2$Program== 'Two'])
```




```
hist(dat2$Calories [dat2$Program== 'Full'])
```




```r
# ANOVA with each Program on Calories (with Levene's Test)
INTaov <- jmv::ANOVA(data = dat2,
                     dep = 'Calories',
                     factors = c('Program'),
                     effectSize = 'partEta',
                     postHoc = c('Program'),
                     postHocCorr = 'tukey',
                     homo = TRUE)
INTaov
```

```
##
##  ANOVA
##
##  ANOVA
##  ------------------------------------------------------------------------------
##                  Sum of Squares    df     Mean Square    F       p         <U+03B7>²p
##  ------------------------------------------------------------------------------
##    Program                 5796      2            2898    1.99    0.140     0.022
##    Residuals             258073    177            1458
##  ------------------------------------------------------------------------------
##
##
##  ASSUMPTION CHECKS
##
##  Test for Homogeneity of Variances (Levene's)
##  --------------------------------------------
##    F        df1    df2    p
```

```
## --------------------------------------------
##    0.774      2    177    0.463
## --------------------------------------------
##
##
##  POST HOC TESTS
##
##  Post Hoc Comparisons - Program
## ----------------------------------------------------------------------------------
##    Program         Program    Mean Difference    SE       df      t          p-tukey
## ----------------------------------------------------------------------------------
##    Full       -    One                  13.70    6.97     177      1.965       0.124
##               -    Two                   4.82    6.97     177      0.691       0.769
##    One        -    Two                  -8.88    6.97     177     -1.274       0.412
## ----------------------------------------------------------------------------------
```

```r
# Kruskal-Wallis including pairwise comparisons (Dwass-Steel-Critchlow-Fligner)
KW_INT_aov <- anovaNP(data = dat2,
                      dep = 'Calories',
                      group = c('Program'),
                      pairs = TRUE)
KW_INT_aov
```

```
##
##  ONE-WAY ANOVA (NON-PARAMETRIC)
##
##  Kruskal-Wallis
## ------------------------------------
##               <U+03C7>²      df    p
## ------------------------------------
##    Calories    98.7     2     < .001
## ------------------------------------
##
##
##  DWASS-STEEL-CRITCHLOW-FLIGNER PAIRWISE COMPARISONS
##
##  Pairwise comparisons - Calories
## -----------------------------------
##                   W         p
## -----------------------------------
##    Full    One    -11.68    < .001
##    Full    Two    -10.07    < .001
##    One     Two      8.61    < .001
## -----------------------------------
```

```r
# Epsilon-squared estimate of effect size
# .1 .3 .5 --- .2 .4 .6 are good estimates for small/med/large

# EpiSq = (H)/((n^2-1)/(n+1))

H <- KW_INT_aov$table$asDF[ , "chiSq"]
print(paste("H =", round(H, 2)))
```

```
## [1] "H = 98.7"
```

```r
N2 <- dim(dat2)[1]
print(paste("N =", N2))
```

```
## [1] "N = 180"
```

```r
EpiSq = (H)/((N2^2-1)/(N2+1))
print(paste("EpiSq = ", round(H, 2), "/((", N2, "^2-1)/(", N2, "+1)) = ", round(EpiSq, 2), sep = ""))
```

```
## [1] "EpiSq = 98.7/((180^2-1)/(180+1)) = 0.55"
```