# BDHS2010_Midterm

Jonathan Rascon

2025-10-25

```r
#MIDTERM 1--------
  #Notes on logistics: Each task assigned in the midterm document (a thru g)
  #is denoted by <a>, <b>, <c>, etc. and once that task is completed, I denote that
  # with "<a> COMPLETED", for example. I alo used the "-------" pancake method
  #to create an outline for my code.

#The given data set is psychiatric symptoms, secondary to an acute spinal injury
#and associated to age. The columns are: Record ID- the unique identifier, Age
#Group (two- 18-35, 65-80), BSI total (brief symptom indicator), and Sig.Scale
#which is a score associated with clinical significannce. Record ID is nominal,
#Age Group is categorical, and BSI and Sig.Scale are ordinal--for our purposes,
#these two should be stored as numeric.

#State the Null and Alternative Hypotheses------
  #NULL hypothesis- there is no difference between the mean scores in number of
    #symptoms and degree of symptoms between age-groups.
  #Alternative hypothesis- there is significant difference between the mean
    #scores in number of symptoms and degree of symptoms between age groups.

#set working directory and create Git repository------
  #I created a repo on the github as "Midterm.1" and created a git project in R
  #ands connected it to my online repo. Just to check the directory, I ran:
getwd() #which returns my repo directory
```

```
## [1] "/Users/jonathanrascon/R projects/BHDS2010/Git Repos/Midterm.1"
```

```r
#libraries-------
  #For convenience, I'll add all libraries here:

library(readxl)
library(tidyverse)
library(pastecs)
library(reshape2)
library(car)

#<a>Correctly create a data frame-------
  #First, I copied and pasted the data set into an excel file before
  #upload. I also will store this in github.

BSI.sig.data <- read_xlsx("midterm.dataset.xlsx")

#Check data types
```

```r
names(BSI.sig.data)
```

```
## [1] " Record "   "Age.Group " "BSI.Total " "Sig.Scale "
```

```r
  #After a quick check, I found that all my columns were pulled with a weird
  #naming format; they all had a space after them, giving them the weird naming
  #convention you see below. I renamed them all.
BSI.sig.data <- BSI.sig.data %>% rename(Age.Group=`Age.Group `,
    Record = ` Record `, Sig.Scale=`Sig.Scale `, BSI.Total=`BSI.Total `)

  #I want to see if my data is stored as the type of data I want:
  #numeric, character, etc.
typeof(BSI.sig.data$Record) ; typeof(BSI.sig.data$Age.Group)
```

```
## [1] "double"

## [1] "character"
```

```r
typeof(BSI.sig.data$BSI.Total) ; typeof(BSI.sig.data$Sig.Scale)
```

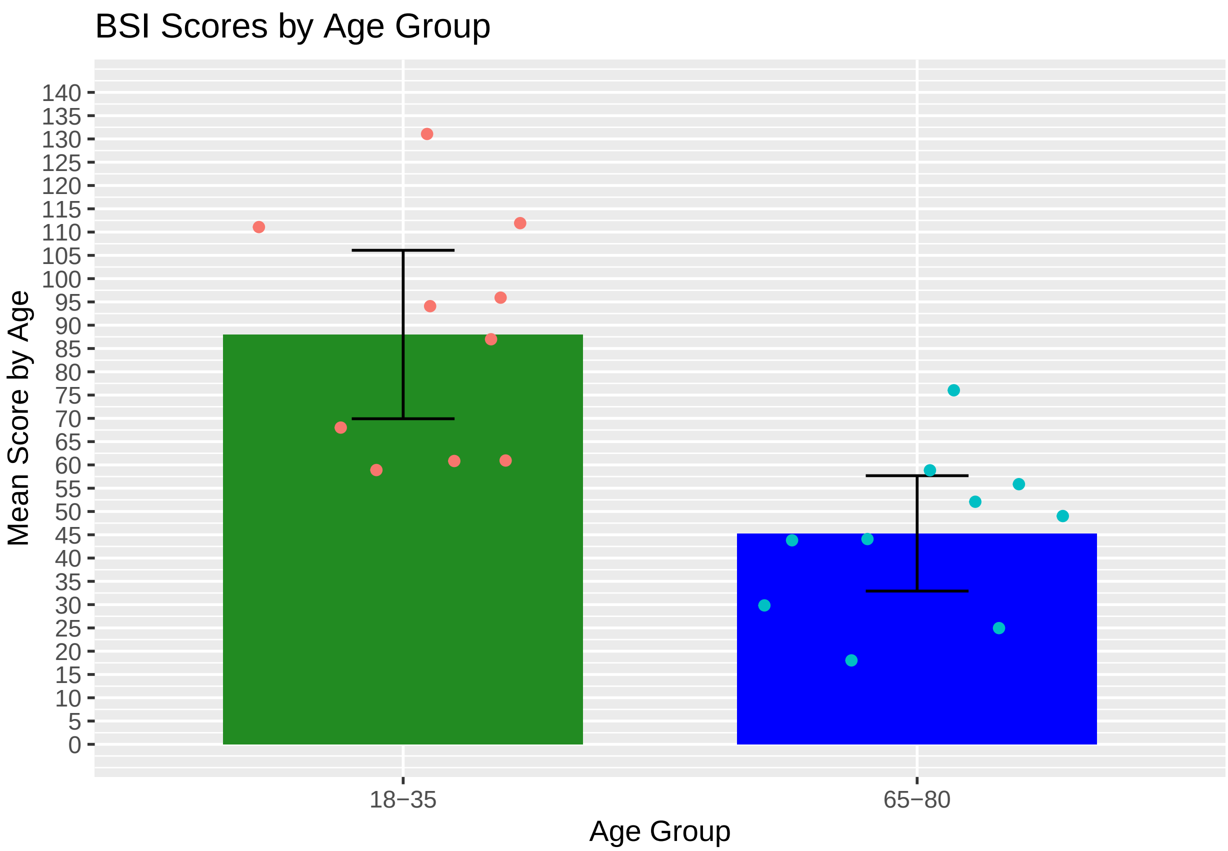
```
## [1] "double"

## [1] "double"
```

```r
  #After checking all data types, I see that ID and Age are stored as "double". I will
  #change Record and Age to character and Age as a factor.

BSI.sig.data <- BSI.sig.data %>% mutate_at(c("Record", "Age.Group"), as.character)
BSI.sig.data <- BSI.sig.data %>% mutate_at("Age.Group", factor)

#using the mutate_at function with concatenate, I change both to character strings
#and the age group to a factor. <a> COMPLETED.

#<b>Calculate the mean, variance and standard deviation for each age group-------
  #By nesting the stat.desc function inside both the round and by functions, I can
  #get a rounded, stratified statistical summary of the data. I also included the
  #norm argument test to true to return a normality test.

by(data = BSI.sig.data$BSI.Total, BSI.sig.data$Age.Group,
   FUN = function(x) round(stat.desc(x, norm = TRUE), 3))
```

```
## BSI.sig.data$Age.Group: 18-35
##      nbr.val     nbr.null       nbr.na          min          max        range
##       10.000        0.000        0.000       59.000      131.000       72.000
##          sum       median         mean      SE.mean CI.mean.0.95          var
##      880.000       90.500       88.000        7.996       18.088      639.333
##      std.dev     coef.var     skewness     skew.2SE     kurtosis     kurt.2SE
##       25.285        0.287        0.213        0.155       -1.540       -0.577
##   normtest.W   normtest.p
##        0.914        0.310
## ------------------------------------------------------------
## BSI.sig.data$Age.Group: 65-80
##      nbr.val     nbr.null       nbr.na          min          max        range
##       10.000        0.000        0.000       18.000       76.000       58.000
##          sum       median         mean      SE.mean CI.mean.0.95          var
##      453.000       46.500       45.300        5.475       12.386      299.789
##      std.dev     coef.var     skewness     skew.2SE     kurtosis     kurt.2SE
```

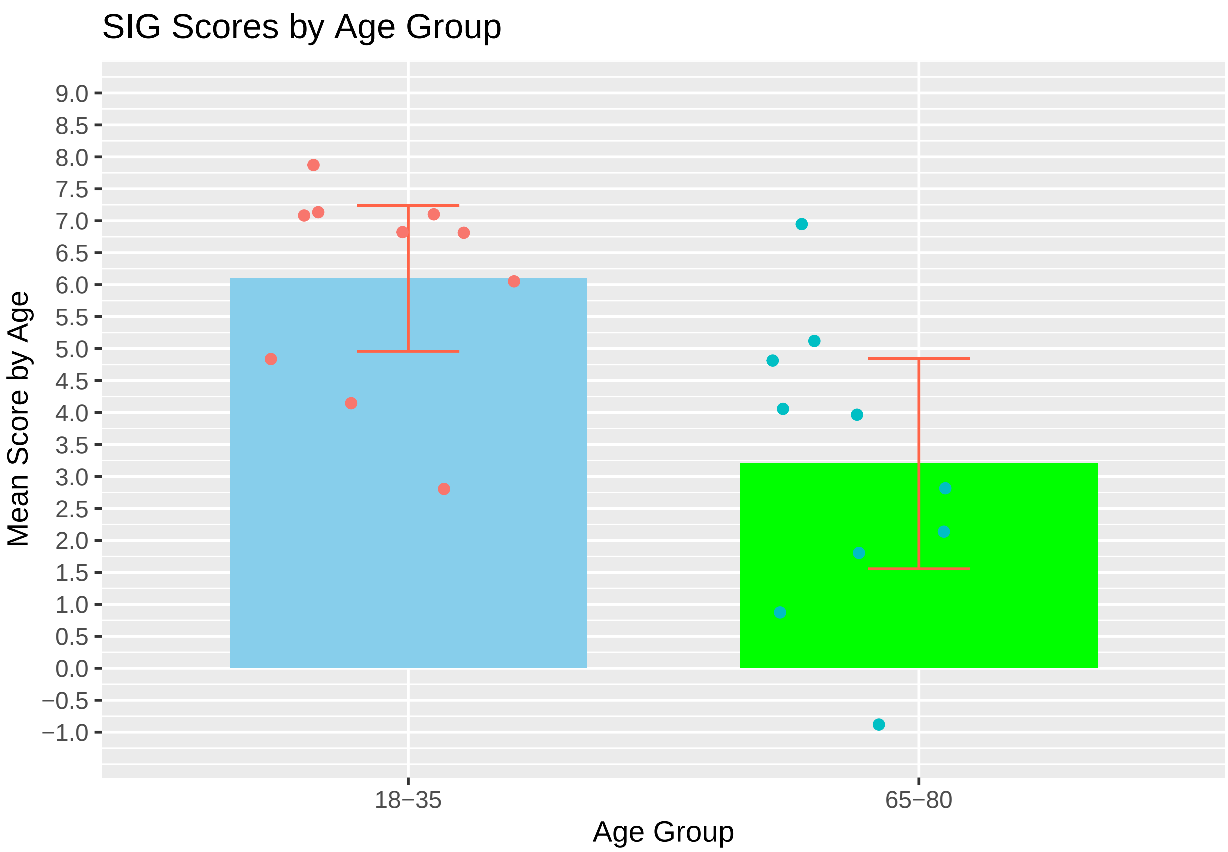
```
##       17.314        0.382        0.015        0.011       -1.087       -0.408
##   normtest.W   normtest.p
##        0.972        0.911
  #The first thing to notice is the extreme difference in mean! The BSI mean
  #is much higher for the 18-35 group. Next, we can notice that, for both groups,
  #the coefficient of variation is relatively small, suggesting that the values
  #cluster fairly close to the mean. The next thing I notice is that
  #both groups are likely normally distributed, because both groups show a
  #normtest.p value greater than alpha=.05, therefore in both cases (where the
  #NULL is that the data is normally distributed) we fail to reject the NULL.
  #Complimenting this, is the normtest.W values for both groups being relatively
  #close to 1. I conclude that this data is likely normally distributed.
  #The sample means along with their confidence intervals do not overlap,
  #suggest a difference in the population means of the age groups.

by(data = BSI.sig.data$Sig.Scale, BSI.sig.data$Age.Group,
   FUN = function(x) round(stat.desc(x, norm = TRUE), 3))

## BSI.sig.data$Age.Group: 18-35
##      nbr.val     nbr.null       nbr.na          min          max        range
##       10.000        0.000        0.000        3.000        8.000        5.000
##          sum       median         mean      SE.mean CI.mean.0.95          var
##       61.000        7.000        6.100        0.504        1.141        2.544
##      std.dev     coef.var     skewness     skew.2SE     kurtosis     kurt.2SE
##        1.595        0.261       -0.736       -0.536       -0.999       -0.374
##   normtest.W   normtest.p
##        0.848        0.055
## ------------------------------------------------------------
## BSI.sig.data$Age.Group: 65-80
##      nbr.val     nbr.null       nbr.na          min          max        range
##       10.000        0.000        0.000       -1.000        7.000        8.000
##          sum       median         mean      SE.mean CI.mean.0.95          var
##       32.000        3.500        3.200        0.727        1.645        5.289
##      std.dev     coef.var     skewness     skew.2SE     kurtosis     kurt.2SE
##        2.300        0.719       -0.170       -0.123       -0.966       -0.362
##   normtest.W   normtest.p
##        0.981        0.972
#In general, we can notice almost all of the same trends for the number of
  #subscales that have a significant t-score (sig.scale). The mean in 18-35 is
  #much higher than in 65-80. The coeifficient of variation is higher in 65-80
  #suggesting that the data points are more spread out in this group. Both groups
  #show a high probability for normality and symmetry. And, the confidence
  #intervals do not overlap (but are VERY close), suggesting again that there is
  #a difference in the underlying population means.

#Therefore, for both dependent variables, I suspect that there is a significant
  #difference in means between groups.  <b> COMPLETED.

#<c>Create bar charts displaying difference in means for BSI Total-------
  #(1)assign an object, (2) pipe the data into ggplot, (3)assign aesthetics,
  #(4)use stat_summary to create bars set to the means, (5)create errorbar geom
  #using mean_cl_normal to create 95% confidence interval based on the normal
  #distribution, (6)use scale_y_continuous to set breaks on the y-axis-- I did
```

```r
  #by 5 to more easily read the mean and CI values. (7)used scale_fill_manual to
  #color the age groups, (8)set geom_jitter (this wasn't required, I just liked the
  #idea of superposing the raw data points on top of the bars) to display raw data-
  #the width and height arguments in geom_jitter tell r how much each point "jitters"
  #away from the others, (9) and added labels and removed the legend.

BSI.plot <- BSI.sig.data %>%
  ggplot(aes(x = Age.Group, y = BSI.Total, fill = Age.Group)) +
  stat_summary(fun = mean, geom = "bar", width = .7) +
  geom_errorbar(stat = "summary", fun.data = "mean_cl_normal", width = 0.2,
              color = "black") +
  scale_y_continuous(limits = c(0, 140), breaks = seq(from = 0, to = 140, by = 5))+
  scale_fill_manual(values = c("forestgreen", "blue")) +
  geom_jitter(aes(color = Age.Group), width = .3, height = .2, stat = "identity") +
  labs(title = "BSI Scores by Age Group", x = "Age Group" ,
       y = "Mean Score by Age") +
  theme(legend.position = "none")

BSI.plot
```



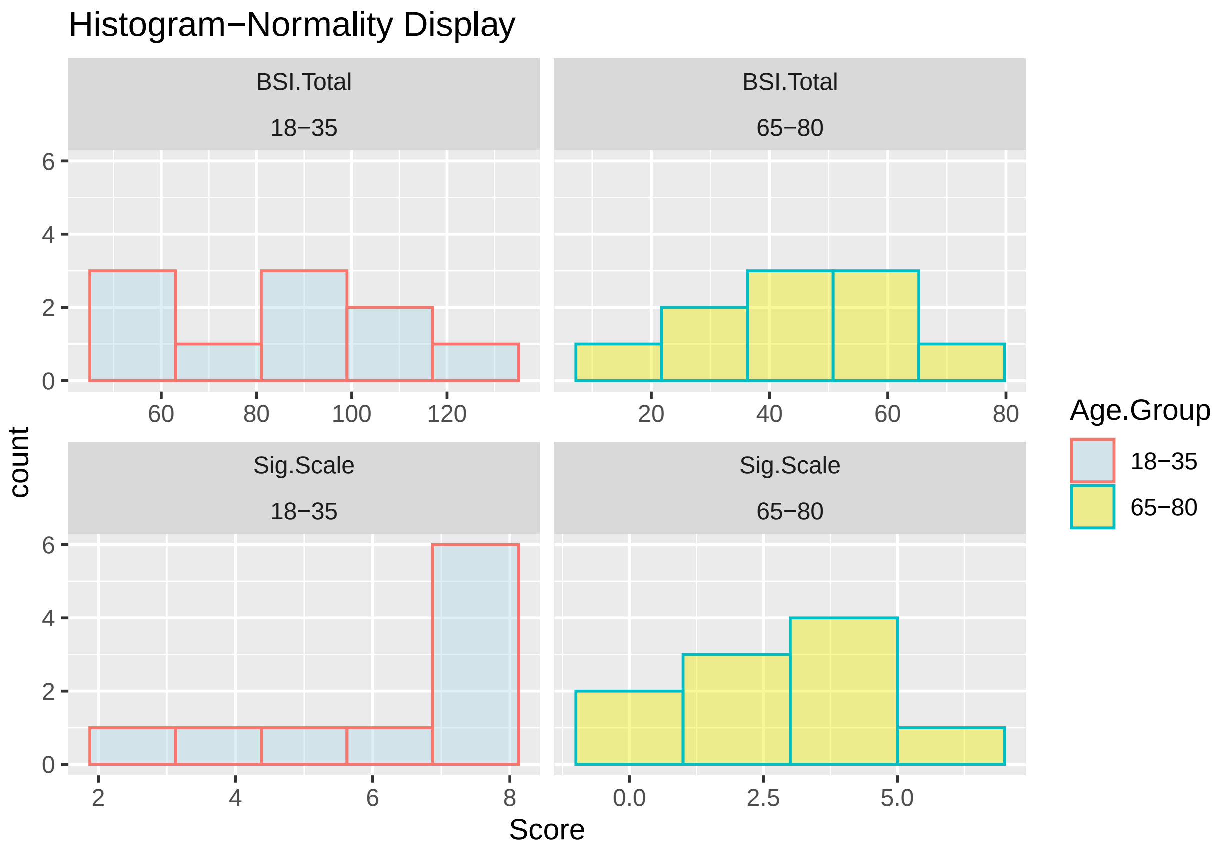

```r
#From this plot, we can easily see the very large difference in means, and that the
  #confidence intervals do not overlap, supporting the idea that the underlying means
  #of these two groups are different. With the addition of geom_jitter we can even
  #see that the range of the spread of data points are similar between groups,
  #which reflects similarity of coefficients of variation. <c> COMPLETED

#<d>Create bar charts displaying difference in means for Sig Scale-------
```

```
  #Almost all methods for building this graph were the same as the previous, with
  #the exception of setting the limits within the scale_y_continuous function from
  #-1.2. If I did not set the limits to less than -1, the minimum jitter-point value
  #-1 would be removed from the graph and I would receive an error (and of course,
  #the point would not appear on the graph). My guess is that I need to set the limit
  #to the min value less by the height of the jitter value in order for the point to
  #consistently show.

Sig.Scale.plot <- BSI.sig.data %>%
  ggplot(aes(x = Age.Group, y = Sig.Scale, fill = Age.Group)) +
  stat_summary(fun = mean, geom = "bar", width = .7) +
  geom_errorbar(stat = "summary", fun.data = "mean_cl_normal",
                width = 0.2, color = "tomato")+
  scale_y_continuous(limits = c(-1.2, 9), breaks = seq(from = -1, to = 9, by = .5))+
  scale_fill_manual(values = c("skyblue", "green")) +
  geom_jitter(aes(color = Age.Group), width = .3, height = .2, stat = "identity") +
  labs(title = "SIG Scores by Age Group", x = "Age Group" , y = "Mean Score by Age") +
  theme(legend.position = "none")

Sig.Scale.plot
```




```
#From this plot, again, we can easily see the very large difference in means. It is
  #not quite as clear that the confidence intervals do not overlap (they don't), but
  #at worst, someone might conclude that they come very close (which they do). Again,
  #it is fairly clear from the graph that the underlying means of the two groups
  #differ significantly. Also, from the geom_jitter overlay, we can see that the
  #variation in the 65-80 group is much larger, which is also reflected in the
```

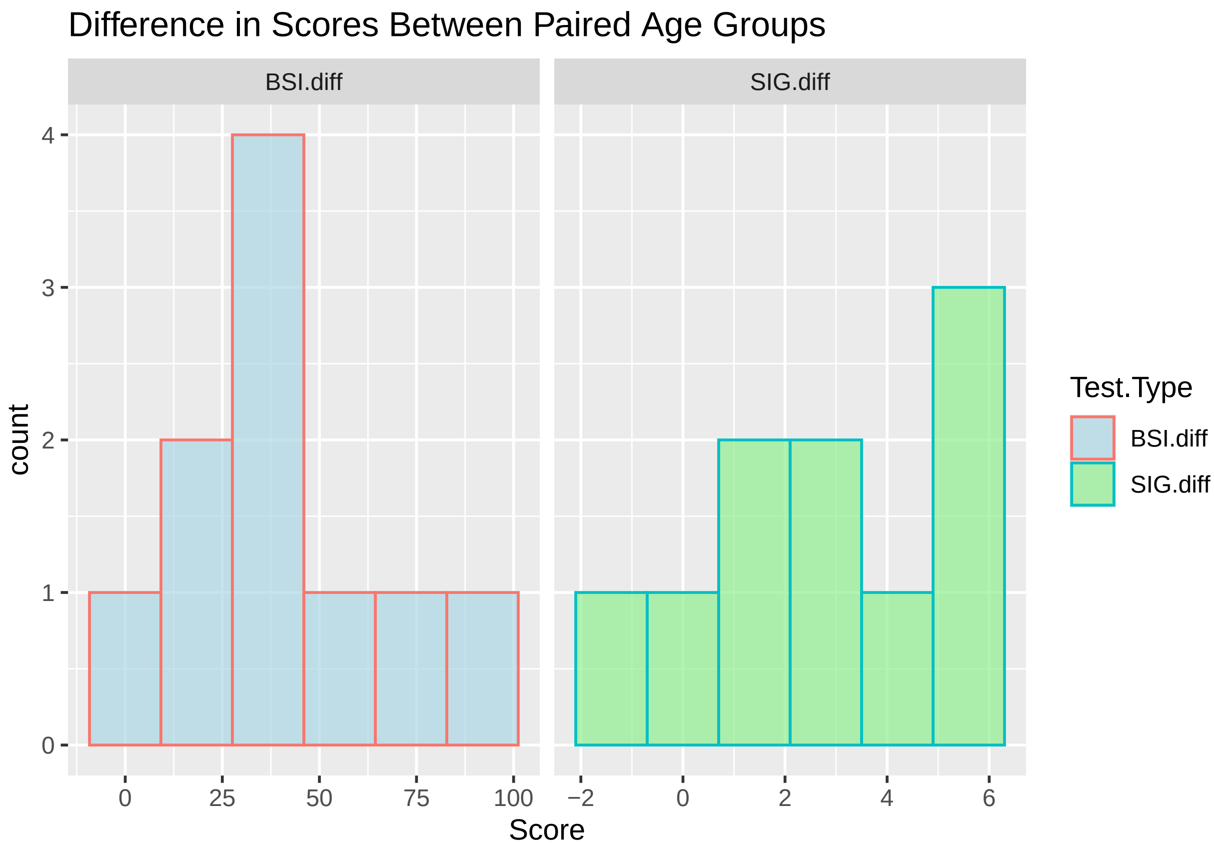
```r
  #respective coefficients of variation.
  #<d> COMPLETED.

#Melt data into long(tidy) format---------
  #The goal here is to create a faceted set of histograms to display the normality
  #suggested by the Shapiro-Wilk scores we saw in the stat.desc outputs. To create
  #a long, or tidy, data set, I assign unique identifiers with the id.vars argument.
  #This tells r to keep all identifiers from these columns; the other columns will
  #be "melted" together. the argument variable.name takes the column names(the ones
  #not preserved in the id.vars argument) and creates a new column with the old
  #column names as entries, and the assigned name as the name of the new column.
  #All of the entries from those columns are stored under the column defined by the
  #value.name argument.

BSI.sig.data.long <- melt(BSI.sig.data, id.vars = c("Record", "Age.Group"),
          variable.name = "Test.Type", value.name = "Score")

#Create histograms facted by test type and age group------
  #For a histogram in ggplot, we only assign an x component into aes. This is
  #a good reason to use long data; I can assign all my scores to the x-component
  #and facet them into the appropriate groups with facet_wrap. Thus, I use faceting
  #to create histograms by test type and age group. I use the scales = "free_x"
  #argument in facet_wrap because the two test types have a different scoring system.
  #This tells r to assign x valuees appropriate to the specific data. I didn't use
  #the "free" argument because I want to keep my counts along the y-axis consistent.

data.histogram <- BSI.sig.data.long %>% ggplot(aes(Score)) +
  geom_histogram(aes(color = Age.Group, fill = Age.Group),
  position = "identity", bins = 5, alpha = .4) +
  scale_fill_manual(values = c("lightblue", "yellow2")) +
  facet_wrap(~Test.Type + Age.Group, scales = "free_x") +
  labs(title = "Histogram-Normality Display")

data.histogram
```




```
#I think these graphs do a good job displaying the "normality" of the data, with
#the exception of the Sig.Scale for 18-35 age group. One question I have for these
#graphs is how to choose the appropriate number of bins.


#<e>Pair rows 1-10 to 11-20. and build histograms of differences,--------
#and use Shapiro-Wilks test to test normality.
#the difficulty was in pairing 1 to 11, 2 to 12, etc. and I could not see a way
#to do this simply with pivot_wider or dcast (because the Record numbers do not
#match up). Therefore, I chose to build a new data frame using cbind nested inside
#data.frame; (1) I attached the data set to my workspace, then (2)pulled each cell
#from its place in the original data.frame.
#(3) I ran a summary to see the new names of the wide data frame (X1-X6)
# then (4) overwrote the new data frame by piping it into the rename function to give
#the columns appropriate names, (5) piped into mutate_at to assign my numeric columns
#as numeric (they were not), and (6) piped into mutate to create my difference
#columns. This "brute force" method may not have been the most efficient, but it
#certainly worked.

#Note that going forward, I had to put the column names between two tick `` marks.
#This, I suspect, is because the names have spaces in them.

attach(BSI.sig.data)
BSI.sig.data.wide <- data.frame(cbind(Record[1:10], Record[11:20], BSI.Total[1:10],
                         BSI.Total[11:20], Sig.Scale[1:10], Sig.Scale[11:20]))
detach(BSI.sig.data)
summary(BSI.sig.data.wide)
```

```
##       X1                 X2                 X3                 X4
##  Length:10          Length:10          Length:10          Length:10
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##       X5                 X6
##  Length:10          Length:10
##  Class :character   Class :character
##  Mode  :character   Mode  :character
```

```
BSI.sig.data.wide <- BSI.sig.data.wide %>% rename("ID 18-35" = X1, "ID 65-80" = X2,
      "BSI 18-35" = X3, "BSI 65-80" = X4, "Sig Scale 18-35" = X5,
      "Sig Scale 65-80" = X6) %>% mutate_at(c("BSI 18-35", "BSI 65-80",
            "Sig Scale 18-35", "Sig Scale 65-80"), as.numeric) %>%
            mutate(BSI.diff = `BSI 18-35`- `BSI 65-80`,
             SIG.diff = `Sig Scale 18-35` - `Sig Scale 65-80`)

#To create histograms of the differences fo the two test types, I (1) created
#a new object to store the graph, (2) piped the wide data frame into select, (3)
#selected only the new difference columns, (4) melted those columns by test type
#and score (I left out the id.vars argument because I did not keep any for the graph)
#and (5) piped that into ggplot with only Score assigned to the aesthetics, (6)
#add a histogram layer, added a manual color scheme by test type, (7) and faceted
#by test type, using the free_x argument, and (8) finally giving each facet n.breaks
#in the x-axis using the scale_x_continuous function (I don't know why one facet
#only got 5)

diff.plot <- BSI.sig.data.wide %>% select(BSI.diff, SIG.diff) %>%
  melt(variable.name = "Test.Type", value.name = "Score") %>% ggplot(aes(Score)) +
  geom_histogram(aes(color = Test.Type, fill = Test.Type),
                 position = "identity", bins = 6, alpha = .7) +
  scale_fill_manual(values = c("lightblue", "lightgreen")) +
  facet_wrap(~Test.Type, scales = "free_x") +
  scale_x_continuous(n.breaks = 6) +
  labs(title = "Difference in Scores Between Paired Age Groups")

diff.plot
```




```
#The data looks relatively normal, as the Shapiro-Wilks test will bear out, but
#I still wonder the best way to assign number of bins.

#Run shapiro-Wilks test to test for normailty---------
#NULL hypothesis for BSI.diff: data is distributed normally; ALT. hypothesis:
#data is not distributed normally.
shapiro.test(BSI.sig.data.wide$BSI.diff)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  BSI.sig.data.wide$BSI.diff
## W = 0.92598, p-value = 0.4095
```

```
shapiro.test(BSI.sig.data.wide$SIG.diff)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  BSI.sig.data.wide$SIG.diff
## W = 0.951, p-value = 0.6803
```

```
#For both test, we "fail to reject the null", that is to say, we conclude that
#the data is likely normal with p-values > alpha=.05. I suspected this going in
#because the data sets we combined by subtraction were also normally distributed.
#<e> COMPLETED.

#<f>Test for significant differences between age groups of each test-----
  #Test all assumptions necessary to run each test.
```

```r
  #(1)Normality: I already tested this back in <b> for all groups and tests
  #using the stat.decs function; all groups tested as normal by the Shapiro-Wilks
  #test
  #(2)Test variances for equality:
  #F-test to compare variances between age groups:
  #NULL hypotheses: underlying population variances of the two groups are equal;
  #ALT. hypotheses: underlying populaion variances are different.
attach(BSI.sig.data.wide)

var.test(`BSI 18-35`,`BSI 65-80`)
```

```
##
##  F test to compare two variances
##
## data:  BSI 18-35 and BSI 65-80
## F = 2.1326, num df = 9, denom df = 9, p-value = 0.2746
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.5297106 8.5858828
## sample estimates:
## ratio of variances
##           2.132612
```

```r
#I ran an f-test to compare variances because we assume that the samples are
#independent, and because we saw that the data is normally distributed; therefore
#an f-test is appropriate to compare variances between age groups.
  #The p-value is greater than alpha=.05, therefore we fail to reject the null
  #and conclude that the variances between age groups for BSI total are equal.

var.test(`Sig Scale 18-35`, `Sig Scale 65-80`)
```

```
##
##  F test to compare two variances
##
## data:  Sig Scale 18-35 and Sig Scale 65-80
## F = 0.48109, num df = 9, denom df = 9, p-value = 0.2908
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.1194966 1.9368753
## sample estimates:
## ratio of variances
##          0.4810924
```

```r
  #The p-value is greater than alpha=.05, therfore we fail to reject the null
  #and conclude that the variances between age groups for Sig Scale are equal.
  #I will note that this result is not well demostrated by my jitter overlay in the
  #Sig Scale bar graph.

#Run t-tests to compare by age group each test type---------
  #Because each test groups tests as normal, they are presumed to be independent,
  #and that the variances are equal, we'll run each test with paired set to
  #false and var.equal set to true.
#NULL hypothesis: True mean is equal between age groups
#ALT. hypothesis: True means are not equal.
```

```r
t.test(`BSI 18-35`,`BSI 65-80`, paired = FALSE, var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  BSI 18-35 and BSI 65-80
## t = 4.4062, df = 18, p-value = 0.0003407
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  22.34032 63.05968
## sample estimates:
## mean of x mean of y
##      88.0      45.3
```

```r
t.test(`Sig Scale 18-35`, `Sig Scale 65-80`, paired = FALSE, var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  Sig Scale 18-35 and Sig Scale 65-80
## t = 3.2766, df = 18, p-value = 0.004192
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.040555 4.759445
## sample estimates:
## mean of x mean of y
##       6.1       3.2
```

```r
detach(BSI.sig.data.wide)
#For both tests, the p-value is significantly less than alpha= .05, therefore
#we fail to reject either null and conclude that the the true means between
#age groups are different, and that the confidence interval tell how much larger
#the means of the 18-35 group are compared to the 65-80 group. <f> COMPLETED.

#This result is well reflected in the bar graphs shown above, given that the means
#for both graphs appear so far apart, and that the confidence intervals do not overlap.

#<g>Retest by age group with the new assumption that the data is paired--------
# i.e.each person was followed over a long period of time and we tested them at
#different timepoints: ONLY FOR THE BSI data!!
  #NULL hypothesis: The difference of the means from the two timepoints is zero
  #ALT. hypothesis: The difference of the means from the two timepoints is not zero
  #(1)Normality: because I ran a Shapiro-Wilks test above on the normality
  #of the differences, and it was shown to be normally distributed, I conclude
  #that the differences data is normal for the paired samples.
  #(2)Variance: I will test the variance at the two timepoints.
    #Using Levene's test, we calculate the difference in variance of the BSI Total
    #score by age. Because the data is normally distibuted, I set the center to mean.
leveneTest(BSI.Total~Age.Group, data = BSI.sig.data, center = mean)
```

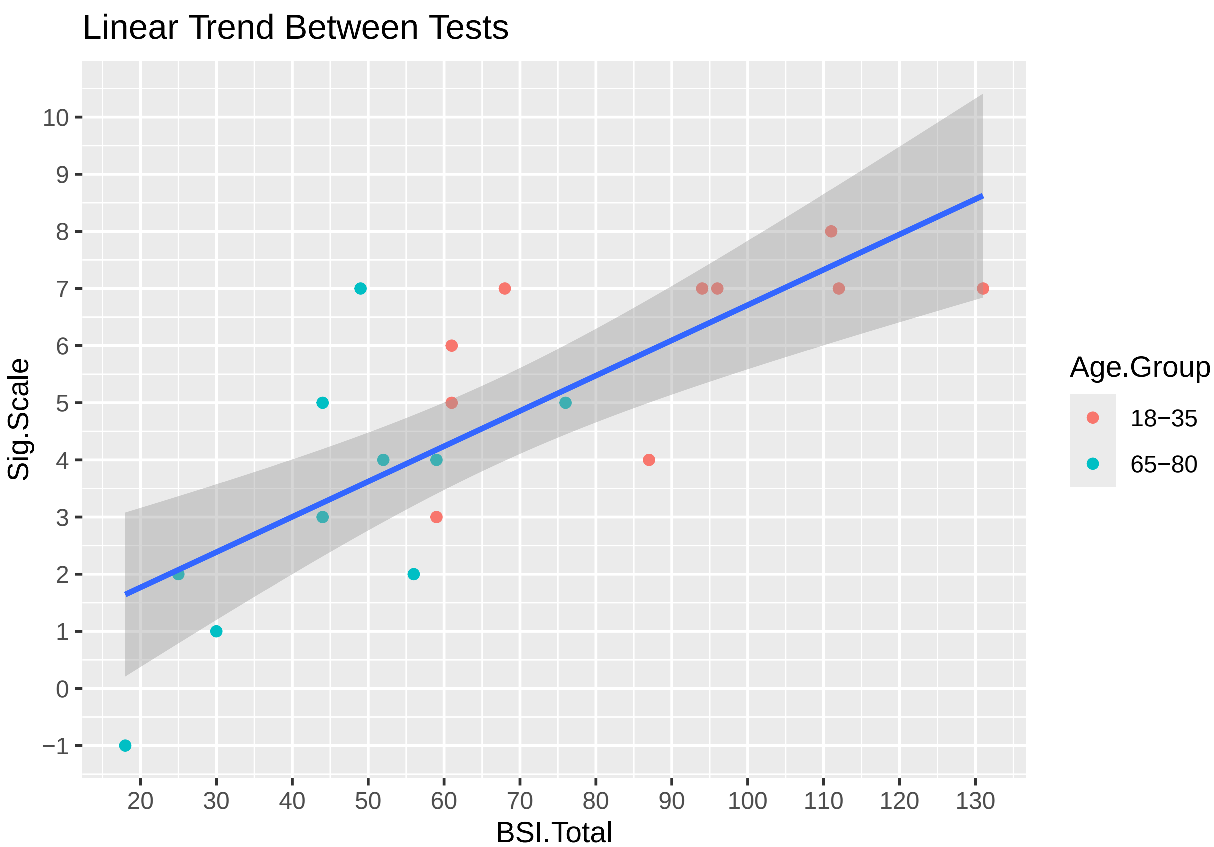
```
## Levene's Test for Homogeneity of Variance (center = mean)
##       Df F value Pr(>F)
## group  1  2.2145  0.154
##       18
```

```
  #fail to reject the null; we assume variances are equal.

#We accept the NULL, i.e. the variances for the different timepoints
#for each test are equal. Therefore, I will run a t-test with paired set to true
#and var.equal set to true.

#Create a data frame with only ID numbers 1-10---------

paired.data.wide <- BSI.sig.data.wide %>% select(-`ID 65-80`) %>%
  rename(ID = `ID 18-35`)

attach(paired.data.wide)

t.test(`BSI 18-35`, `BSI 65-80`, paired = TRUE, var.equal = TRUE)
```

```
##
##  Paired t-test
##
## data:  BSI 18-35 and BSI 65-80
## t = 4.9066, df = 9, p-value = 0.00084
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  23.01346 62.38654
## sample estimates:
## mean difference
##            42.7
```

```
#We fail to reject the null, and conclude that the mean difference is not equal to
#zero. <g> COMPLETED.

detach(paired.data.wide)

#Conclusions--------
  #(1)Treated as independent samples, that is, samples taken as a snapshot of a moment
  #in time, the means of the two age groups of the Brief Symptom Inventory (BSI)
  #total score show a statistically significant difference in mean scores, with
  #the 18-35 group showing a higher score on average. The variances, on average,
  #tested as equal, both tested as normally distributed, with similar coefficients
  #of variation. We would conclude that, on average, older patients who suffer an
  #acute spinal injury, suffer few psychiatric symptoms as a co-morbid effect
  # of that injury on the BSI scale.

  #We also can conclude that there is a statistically significant difference of the
  #means between the two age groups on the Significance Score, with the 18-35 group
  #showing a higher score on average. The variances, on average, tested as equal,
  #both tested as normally distributed, but the coefficients of variation differed,
  #showing a higher deviation from the mean in the older group. We would conclude
  #that, on average, the older patients show a lesser number of "clinically significant"
  #t-scores on the symptom subscales.

  #A final note for the independent samples is that there is a great deal of parity
  #between the statistical description and conclusions of the BSI total data and
  #the Sig Scores data leading me to believe there is a correlation between a person's
  #score on the BSI and the Sig Scale.
```

```r
point.graph <- BSI.sig.data %>% ggplot(aes(x = BSI.Total, y = Sig.Scale)) +
    geom_point(aes(group = Age.Group, color = Age.Group)) +
    geom_smooth(method = "lm") + scale_x_continuous(n.breaks = 14) +
    scale_y_continuous(n.breaks = 10) +labs(title = "Linear Trend Between Tests")

point.graph
```




```r
cor(BSI.sig.data$BSI.Total, BSI.sig.data$Sig.Scale)
```

```
## [1] 0.7719855
```

```r
#Both the graph and the pearson correlation test suggest a strong positive
#linear relationship between scores of the two tests.

#(2)Treated as a paired sample, i.e. individuals tested at different timepoints,
#the data shows a statistically significant difference in means for the average
#BSI total score at different ages. For a paired sample, this suggests that,
#over time, a person's score on the BSI scale decreases, or that the number of
#psychiatric symptoms a person suffers decreases as the time since the acute
#spinal injury is greater.
```