Correlation Matrices, Reliability Tests, and t-Tests

Contents

```
library(mosaic)
library(ggplot2)
library(dplyr)

#install.packages("psych")
library(psych)
```

Data inside R packages

To demonstrate the use of the `alpha()` function in the `psych` package from computing Cronbach's alpha, we will need a data set with scale items. Many packages have example datasets pre-loaded. The `psych` package has a dataset called `bfi` that has n = 2800 participants responding to 25 personality items for the Big Five Inventory.

```
#We can see a list of ALL the datasets current avaliable
data()
```

Use the `?` to get more information.

```
?bfi
```

If we see a dataset we like we can load it into our environment

```r
data(bfi)
glimpse(bfi)
```

```
## Observations: 2,800
## Variables: 28
## $ A1        <int> 2, 2, 5, 4, 2, 6, 2, 4, 4, 2, 4, 2, 5, 5, 4, 4, 4, 5...
## $ A2        <int> 4, 4, 4, 4, 3, 6, 5, 3, 3, 5, 4, 5, 5, 5, 5, 3, 6, 5...
## $ A3        <int> 3, 5, 5, 6, 3, 5, 5, 1, 6, 6, 5, 5, 5, 5, 2, 6, 6, 5...
## $ A4        <int> 4, 2, 4, 5, 4, 6, 3, 5, 3, 6, 6, 5, 6, 6, 2, 6, 2, 4...
## $ A5        <int> 4, 5, 4, 5, 5, 5, 5, 1, 3, 5, 5, 5, 4, 6, 1, 3, 5, 5...
## $ C1        <int> 2, 5, 4, 4, 4, 6, 5, 3, 6, 6, 4, 5, 5, 4, 5, 5, 4, 5...
## $ C2        <int> 3, 4, 5, 4, 4, 6, 4, 2, 6, 5, 3, 4, 4, 4, 5, 5, 4, 5...
## $ C3        <int> 3, 4, 4, 3, 5, 6, 4, 4, 3, 6, 5, 5, 3, 4, 5, 5, 4, 5...
## $ C4        <int> 4, 3, 2, 5, 3, 1, 2, 2, 4, 2, 3, 4, 2, 2, 2, 3, 4, 4...
## $ C5        <int> 4, 4, 5, 5, 2, 3, 3, 4, 5, 1, 2, 5, 2, 1, 2, 5, 4, 3...
## $ E1        <int> 3, 1, 2, 5, 2, 2, 4, 3, 5, 2, 1, 3, 3, 2, 3, 1, 1, 2...
## $ E2        <int> 3, 1, 4, 3, 2, 1, 3, 6, 3, 2, 3, 3, 3, 2, 4, 1, 2, 2...
## $ E3        <int> 3, 6, 4, 4, 5, 6, 4, 4, NA, 4, 2, 4, 3, 4, 3, 6, 5, ...
## $ E4        <int> 4, 4, 4, 4, 4, 5, 5, 2, 4, 5, 5, 5, 2, 6, 6, 6, 5, 6...
## $ E5        <int> 4, 3, 5, 4, 5, 6, 5, 1, 3, 5, 4, 4, 4, 5, 5, 4, 5, 6...
## $ N1        <int> 3, 3, 4, 2, 2, 3, 1, 6, 5, 5, 3, 4, 1, 1, 2, 4, 4, 6...
## $ N2        <int> 4, 3, 5, 5, 3, 5, 2, 3, 5, 5, 3, 5, 2, 1, 4, 5, 4, 5...
## $ N3        <int> 2, 3, 4, 2, 4, 2, 2, 2, 2, 5, 4, 3, 2, 1, 2, 4, 4, 5...
## $ N4        <int> 2, 5, 2, 4, 4, 2, 1, 6, 3, 2, 2, 2, 2, 2, 2, 5, 4, 4...
## $ N5        <int> 3, 5, 3, 1, 3, 3, 1, 4, 3, 4, 3, NA, 2, 1, 3, 5, 5, ...
## $ O1        <int> 3, 4, 4, 3, 3, 4, 5, 3, 6, 5, 5, 4, 4, 5, 5, 6, 5, 5...
## $ O2        <int> 6, 2, 2, 3, 3, 3, 2, 2, 6, 1, 3, 6, 2, 3, 2, 6, 1, 1...
## $ O3        <int> 3, 4, 5, 4, 4, 5, 5, 4, 6, 5, 5, 4, 4, 4, 5, 6, 5, 4...
## $ O4        <int> 4, 3, 5, 3, 3, 6, 6, 5, 6, 5, 6, 5, 5, 4, 5, 3, 6, 5...
## $ O5        <int> 3, 3, 2, 5, 3, 1, 1, 3, 1, 2, 3, 4, 2, 4, 5, 2, 3, 4...
## $ gender    <int> 1, 2, 2, 2, 1, 2, 1, 1, 1, 2, 1, 1, 2, 1, 1, 1, 2, 1...
## $ education <int> NA, NA, NA, NA, NA, 3, NA, 2, 1, NA, 1, NA, NA, NA, ...
## $ age       <int> 16, 18, 17, 17, 17, 21, 18, 19, 19, 17, 21, 16, 16, ...
```

First, we can get descriptive statistics for individual personality scale items. Let's do the agreeableness scale first.

```r
#Get descriptive statistics with mosaic or dplyr here. Get them for Age too.

#Also, get the frequencies for gender and education
```

Bivariate Correlation Matrix

In addition to descriptive statistics, we would probably also like to get a correlation matrix for items on a scale. I like the function `corr.test()` in the `psych` package. Notice that we are using the `dplyr` `select()` function inside of `corr.test()`.

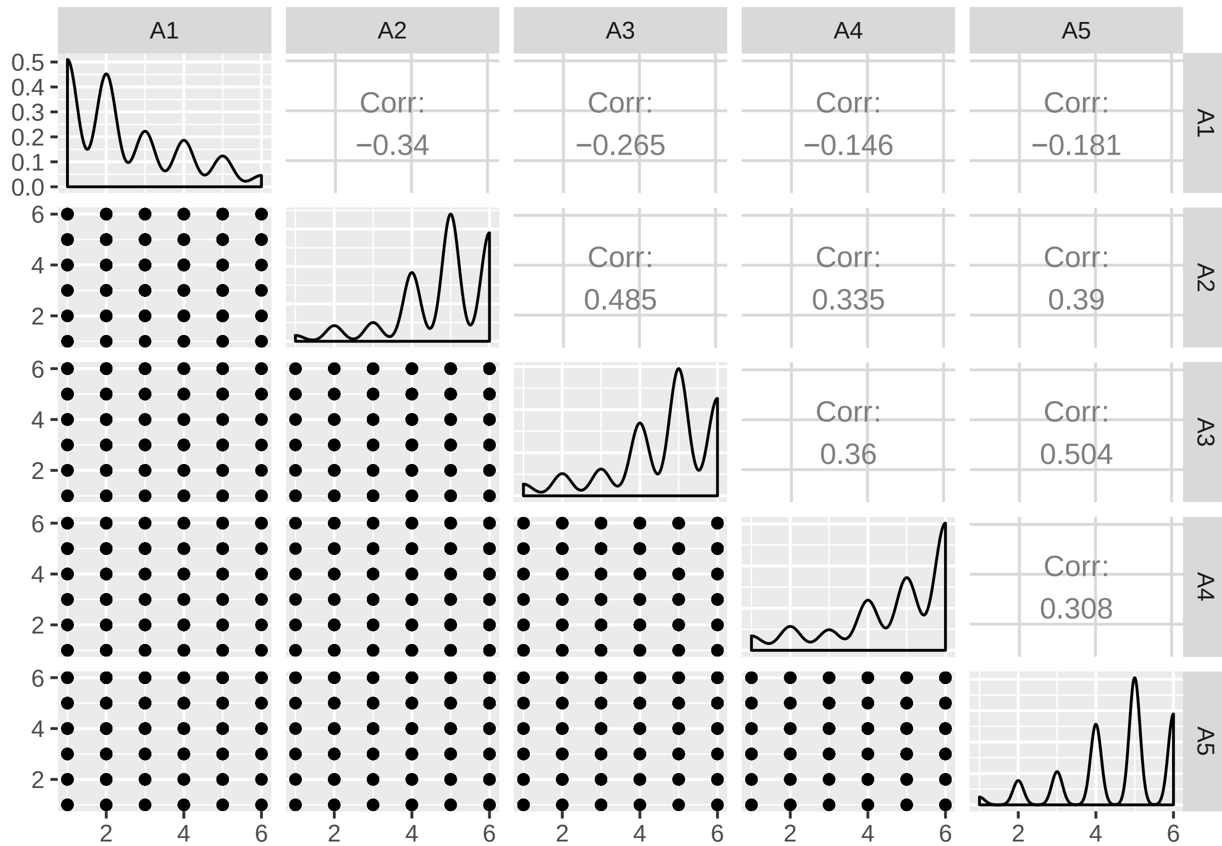
```
corr.test(select(bfi, A1, A2, A3, A4, A5))
```

```
## Call:corr.test(x = select(bfi, A1, A2, A3, A4, A5))
## Correlation matrix
##       A1    A2    A3    A4    A5
## A1  1.00 -0.34 -0.27 -0.15 -0.18
## A2 -0.34  1.00  0.49  0.34  0.39
## A3 -0.27  0.49  1.00  0.36  0.50
## A4 -0.15  0.34  0.36  1.00  0.31
## A5 -0.18  0.39  0.50  0.31  1.00
## Sample Size
##      A1   A2   A3   A4   A5
## A1 2784 2757 2759 2767 2769
## A2 2757 2773 2751 2758 2757
## A3 2759 2751 2774 2759 2758
## A4 2767 2758 2759 2781 2765
## A5 2769 2757 2758 2765 2784
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##    A1 A2 A3 A4 A5
## A1  0  0  0  0  0
## A2  0  0  0  0  0
## A3  0  0  0  0  0
## A4  0  0  0  0  0
## A5  0  0  0  0  0
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

There is also a `pairs()` function in Base R and the `ggpairs()` function in the `GGally` package that is a ggplot correlation matrix/matrix scatter plot provider.

```
#install.packages("GGally")
library(GGally)

ggpairs(select(bfi, A1, A2, A3, A4, A5))
```



It is clear that we will need to reverse score A1 to assess reliability and to create our agreeableness scale scores. Hint: to reverse score an item you might use this formula: `item.r = (min(item, na.rm = TRUE) + max(item, na.rm = TRUE)) - item`. Although, beware that if no one in your sample used the min and/or max of your scale, this code will NOT work.

```
bfi <- bfi %>%
  mutate(A1.r = (min(A1, na.rm = TRUE) + max(A1, na.rm = TRUE)) - A1)
```

Now compute correlation matrices for the other 4 personality sub scales and create any reverse scored items the need to be created.

```
corr.test(select(bfi, A1.r, A2, A3, A4, A5))
```

```
## Call:corr.test(x = select(bfi, A1.r, A2, A3, A4, A5))
## Correlation matrix
##      A1.r   A2   A3   A4   A5
## A1.r 1.00 0.34 0.27 0.15 0.18
## A2   0.34 1.00 0.49 0.34 0.39
## A3   0.27 0.49 1.00 0.36 0.50
## A4   0.15 0.34 0.36 1.00 0.31
## A5   0.18 0.39 0.50 0.31 1.00
## Sample Size
##      A1.r   A2   A3   A4   A5
```

```
## A1.r 2784 2757 2759 2767 2769
## A2   2757 2773 2751 2758 2757
## A3   2759 2751 2774 2759 2758
## A4   2767 2758 2759 2781 2765
## A5   2769 2757 2758 2765 2784
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##      A1.r A2 A3 A4 A5
## A1.r    0  0  0  0  0
## A2      0  0  0  0  0
## A3      0  0  0  0  0
## A4      0  0  0  0  0
## A5      0  0  0  0  0
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

Or, you can do it all at once with the pipe!

```
bfi %>%
  mutate(A1.r = (min(A1, na.rm = TRUE) + max(A1, na.rm = TRUE)) - A1) %>%
  select(A1.r, A2, A3, A4, A5) %>%
  corr.test()
```

```
## Call:corr.test(x = .)
## Correlation matrix
##      A1.r   A2   A3   A4   A5
## A1.r 1.00 0.34 0.27 0.15 0.18
## A2   0.34 1.00 0.49 0.34 0.39
## A3   0.27 0.49 1.00 0.36 0.50
## A4   0.15 0.34 0.36 1.00 0.31
## A5   0.18 0.39 0.50 0.31 1.00
## Sample Size
##      A1.r   A2   A3   A4   A5
## A1.r 2784 2757 2759 2767 2769
## A2   2757 2773 2751 2758 2757
## A3   2759 2751 2774 2759 2758
## A4   2767 2758 2759 2781 2765
## A5   2769 2757 2758 2765 2784
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##      A1.r A2 A3 A4 A5
## A1.r    0  0  0  0  0
## A2      0  0  0  0  0
## A3      0  0  0  0  0
## A4      0  0  0  0  0
## A5      0  0  0  0  0
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

For more practice, find another dataset that sounds interesting and run some correlation matrices.

```
#data()
```

Cronbach's alpha

Now that we have our reverse scored items and we have a sense of how the items relate, the next step is to test the scale reliability. We do this with Cronbach's alpha. There is a function called `alpha()` in the `psych` package.

```
alpha(select(bfi, A1.r, A2, A3, A4, A5))
```

```
##
## Reliability analysis
## Call: alpha(x = select(bfi, A1.r, A2, A3, A4, A5))
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean  sd
##        0.7      0.71    0.68      0.33 2.5 0.009  4.7 0.9
##
##  lower alpha upper     95% confidence boundaries
## 0.69 0.7 0.72
##
##  Reliability if an item is dropped:
##      raw_alpha std.alpha G6(smc) average_r S/N alpha se     NA
## A1.r      0.72      0.73    0.67      0.40 2.6   0.0087 0.0059
## A2        0.62      0.63    0.58      0.29 1.7   0.0119 0.0154
## A3        0.60      0.61    0.56      0.28 1.6   0.0124 0.0086
## A4        0.69      0.69    0.65      0.36 2.3   0.0098 0.0144
## A5        0.64      0.66    0.61      0.32 1.9   0.0111 0.0115
##
##  Item statistics
##         n raw.r std.r r.cor r.drop mean  sd
## A1.r 2784  0.58  0.57  0.38   0.31  4.6 1.4
## A2   2773  0.73  0.75  0.67   0.56  4.8 1.2
## A3   2774  0.76  0.77  0.71   0.59  4.6 1.3
## A4   2781  0.65  0.63  0.47   0.39  4.7 1.5
## A5   2784  0.69  0.70  0.60   0.49  4.6 1.3
##
## Non missing response frequency for each item
##         1    2    3    4    5    6 miss
## A1.r 0.03 0.08 0.12 0.14 0.29 0.33 0.01
## A2   0.02 0.05 0.05 0.20 0.37 0.31 0.01
## A3   0.03 0.06 0.07 0.20 0.36 0.27 0.01
```

```
## A4   0.05 0.08 0.07 0.16 0.24 0.41 0.01
## A5   0.02 0.07 0.09 0.22 0.35 0.25 0.01
```

How would you use the pipe?

```
#use the pipe
```

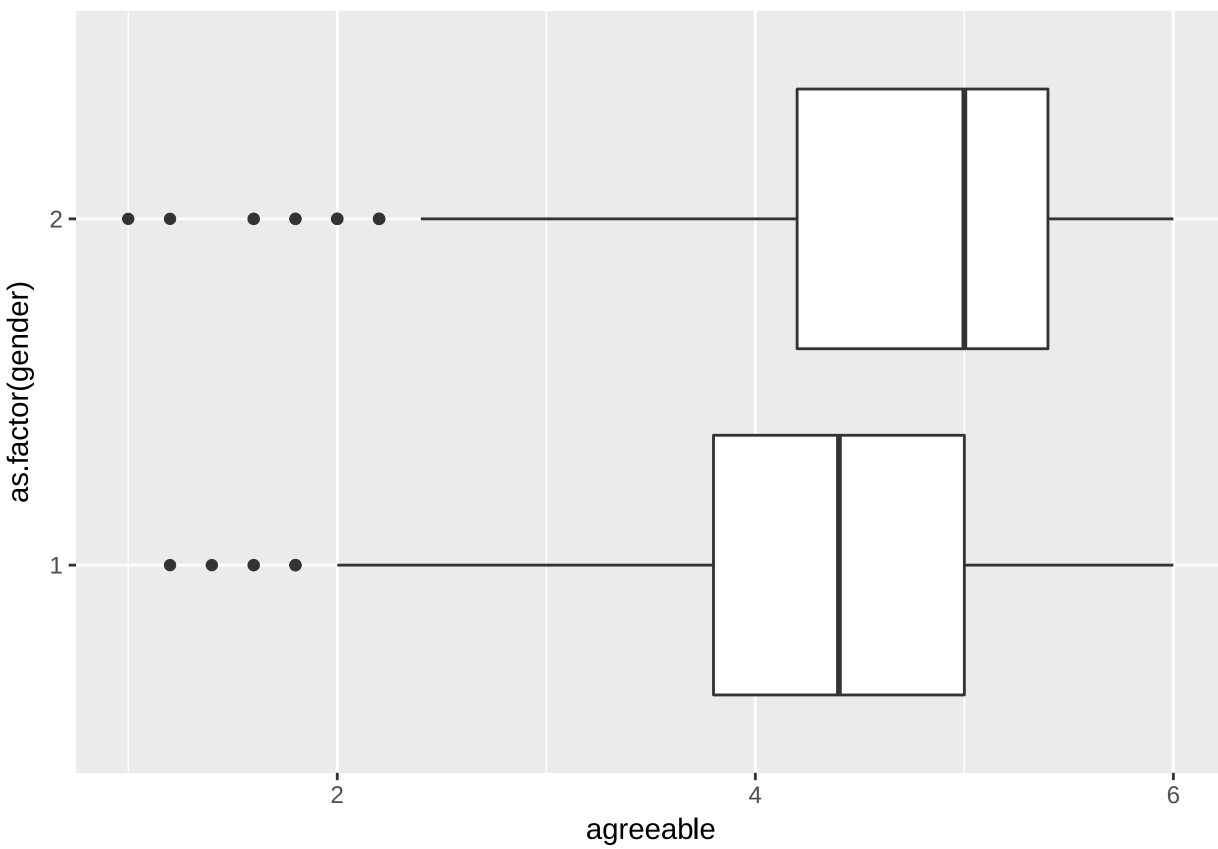
With an alphas of .70 we have a reliable agreeableness scale. The next step would be to create agreeableness scale scores for participants, the average of these 5 items. We can do this with `mutate()`. Note that we cannot use `mean()` here because `mean()` is going to get the mean of a column (a variable). What we want it the mean across columns.

```
bfi <- bfi %>%
  mutate(agreeable = (A1.r + A2 + A3 + A4 + A5)/5)

#If a participant is missing data on any item, we should use the base R function rowMe
bfi <- bfi %>%
  mutate(agreeable = rowMeans(select(bfi, A1.r, A2, A3, A4, A5), na.rm = TRUE))
```

Let's take a look at it with a visualization.

```
qplot(y = agreeable, x = as.factor(gender), data = bfi, bins = 10, geom = "boxplot") +
  coord_flip()
```



Also try making histogram, faceted by gender, and perhaps turning the `qplot()` code above into `ggplot()` code.

```r
#ggplot practice
```

Now calculate the reliabilities for each of the other 4 scales. Be sure to include the reverse scored items you created where relevant.

```r
#calculate reliabilities
```

Create scale scores and make visualizations of those scale scores.

```r
#scale scores and visualizations
```

t-Tests

Independent Samples t-Test

Next we can ask, are men and women different on agreeableness? To answer this question we need to run an independent samples t-test. We'll use the function `t.test()` in the `mosaic` package. Recall that `mosaic` first needs the formula, `y ~ x`, then the data, `data = dataName`.

```r
library(mosaic) #reminder!

t.test(~agreeable, mu = 3.5, data = bfi)
```

```
## ~agreeable

##
##  One Sample t-test
##
## data:  agreeable
## t = 67.857, df = 2799, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 3.5
## 95 percent confidence interval:
##  4.618804 4.685386
## sample estimates:
## mean of x
##  4.652095
```

```r
t.test(agreeable ~ gender, data = bfi)
```

```
## agreeable ~ gender

##
##  Welch Two Sample t-test
##
## data:  agreeable by gender
```

```
## t = -10.892, df = 1690, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.4682270 -0.3253277
## sample estimates:
## mean in group 1 mean in group 2
##        4.385546        4.782323
```

There is a statistically significant difference between women and men on agreeableness, $t(1654.50)$ = -10.73, $p < .001$, with women ($M = 4.77$, $SD = 0.86$) scoring higher than men ($M = 4.38$, $SD = 0.93$). It is possible to code in these numbers such that if the data were updated, the text would update as well.

We can also ask for equal variance assumed.

```r
t.test(agreeable ~ gender, data = bfi, var.equal = TRUE)
```

```
## agreeable ~ gender
```

```
##
##  Two Sample t-test
##
## data:  agreeable by gender
## t = -11.216, df = 2798, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.4661458 -0.3274089
## sample estimates:
## mean in group 1 mean in group 2
##        4.385546        4.782323
```

Mini-Lesson: Embedding Statistics in Text

```r
tmod <- t.test(agreeable ~ gender, data = bfi)
```

```
## agreeable ~ gender
```

```r
names(tmod)
```

```
## [1] "statistic"   "parameter"   "p.value"     "conf.int"    "estimate"
## [6] "null.value"  "alternative" "method"      "data.name"
```

```r
tmod$statistic
```

```
##         t
## -10.89195
```

```r
tmod$parameter
```

```
##       df
## 1689.967
```

```r
tmod$p.value
```

```
## [1] 9.522067e-27
```

```r
ds <- favstats(agreeable ~ gender, data = bfi)
```

There is a statistically significant difference between women and men on agreeableness, $t(1689.97)$ = -10.89, $p < 0.001$, with women ($M = 4.78$, $SD = 0.85$) scoring higher than men ($M = 4.39$, $SD = 0.93$).

Test for gender differences for the other 4 scales. Then test differences between those who graduated college, and those who did not. Hint: use `mutate()` to create a dichotomous variable as we did for `wise_hus`.

```r
#more t-Tests
```

Paired Samples t-Test

To demonstrate paired samples t-test I'd like to use yet another built in dataset. The `sat.act` dataset has information for 700 people on their SAT verbal, SAT quantitative, and ACT scores.

```r
data(sat.act)
#?sat.act
glimpse(sat.act)
```

```
## Observations: 700
## Variables: 6
## $ gender    <int> 2, 2, 2, 1, 1, 1, 2, 1, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2...
## $ education <int> 3, 3, 3, 4, 2, 5, 5, 3, 4, 5, 3, 4, 4, 4, 3, 4, 3, 4...
## $ age       <int> 19, 23, 20, 27, 33, 26, 30, 19, 23, 40, 23, 34, 32, ...
## $ ACT       <int> 24, 35, 21, 26, 31, 28, 36, 22, 22, 35, 32, 29, 21, ...
## $ SATV      <int> 500, 600, 480, 550, 600, 640, 610, 520, 400, 730, 76...
## $ SATQ      <int> 500, 500, 470, 520, 550, 640, 500, 560, 600, 800, 71...
```

Let's compare people's scores on the SAT verbal to their scores on the SAT quantitative using the `t.test()` function in Base R. This time instead of using the formula in the beginning, we add the two variables to be compared separated by a `,`. We also need to add `paired = TRUE`.

```r
t.test(~(SATV-SATQ), data = sat.act)
```

```
## ~(SATV - SATQ)
```

```
##
##  One Sample t-test
##
## data:  SATV
## t = 0.57483, df = 686, p-value = 0.5656
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -5.116109  9.351917
## sample estimates:
## mean of x
##  2.117904
```

There is no statistically significant difference between SAT verbal and SAT quantitative scores, $t(686) = 0.57$, $p = .566$. But perhaps there is a difference for boys only, or for girls only. We can create a dataset of just boys using `filter()` and then another one of just girls. This would be completely fine, but we could also pipe directly into the `t.test()` function. There is a catch: The pipe, `%>%`, puts the resulting dataframe into the first position in the destination function, but we want it in the third position. We can use the `.` symbol to pipe the dataframe into whichever position we'd like.

```r
sat.act %>%
  filter(gender == 1) %>%
  t.test(~(SATV-SATQ), data = .)
```

```
## ~(SATV - SATQ)
## <environment: 0x7ff9ec97cc68>
```

```
##
##  One Sample t-test
##
## data:  SATV
## t = -3.4934, df = 244, p-value = 0.0005661
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -32.08114  -8.94743
## sample estimates:
## mean of x
## -20.51429
```

```r
sat.act %>%
  filter(gender == 2) %>%
  t.test(~(SATV-SATQ), data = .)
```

```
## ~(SATV - SATQ)
## <environment: 0x7ff9ec298bf8>
```

```
##
##  One Sample t-test
```

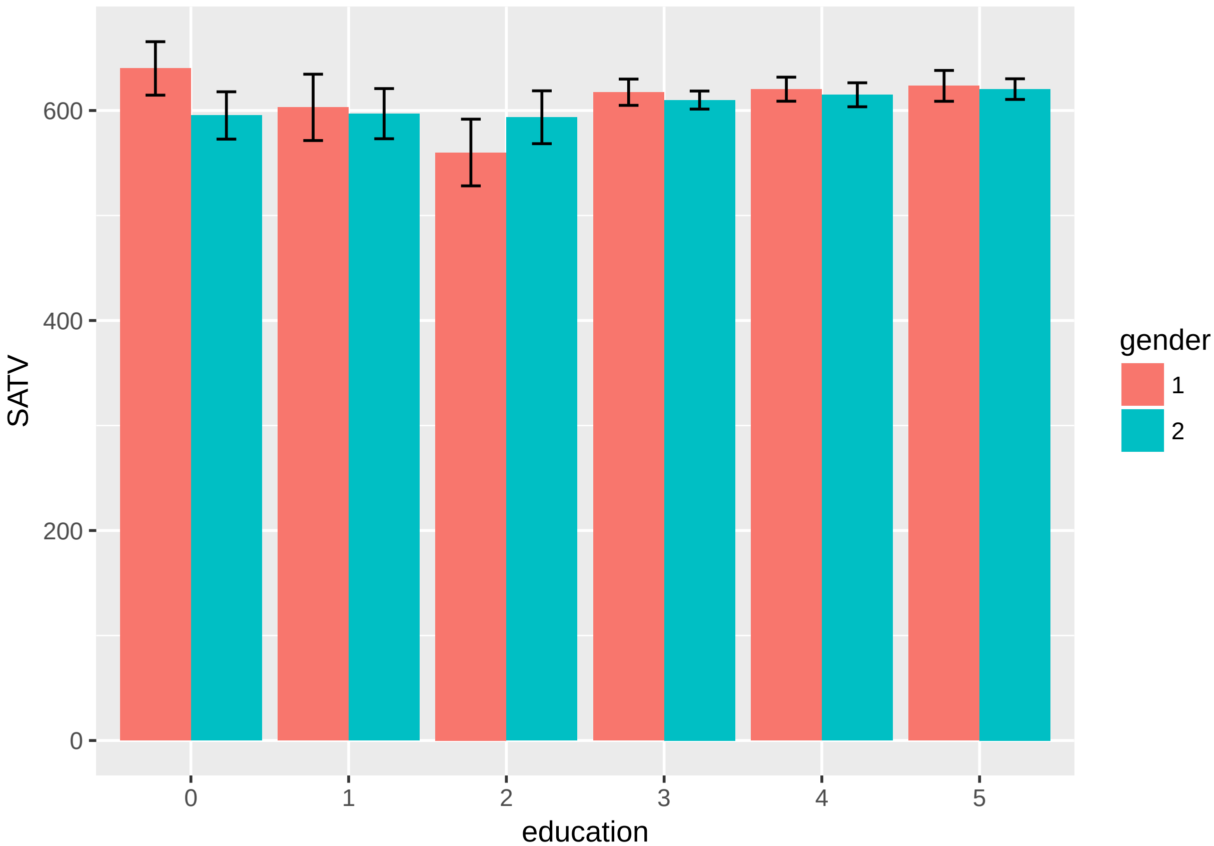
```
##
## data:  SATV
## t = 3.1813, df = 441, p-value = 0.00157
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##   5.604249 23.721543
## sample estimates:
## mean of x
##   14.6629
```

Explore the differences in `SATV` and `SATQ` for each level of the `education` variable.

```r
#differences for each level of education
?sat.act
```

Side-by-Side Bar Chart

```r
sat.act %>%
  mutate(education = as.factor(education),
         gender = as.factor(gender)) %>%
ggplot(aes(x = education, y = SATV, fill = gender)) +
  stat_summary(fun.y = "mean", geom = "bar",
               position = position_dodge(width = 0.9)) +
  stat_summary(fun.data = "mean_se", geom = "errorbar",
               width = 0.25,
               position = position_dodge(width = 0.9))
```



Bonus: Chi-Square Test

Here is a bonus chi-square test to test if education is associated with gender. Question: Is degree attainment related to gender?

```
counts <- tally(gender~education, data = filter(bfi, !is.na(education)))
```

```
chisq.test(counts)
```

```
##
##  Pearson's Chi-squared test
##
## data:  counts
## X-squared = 21.672, df = 4, p-value = 0.0002329
```