Educational Technology Project - HCI (Fall 2016) Data Analysis

Process Data

```
# Set cwd
setwd("D:/Documents/Data Science/Educational Technology/R/hci")
#setwd("E:/Educational Technology/R/HCI")
getwd()

# Load libraries
library(plyr)
library(tools)
library(ggplot2)

# Read in survey data sets
CS6750_fall16_soc = read.csv('Survey_CS6750_FALL16_SOC.csv')
CS6750_fall16_qc = read.csv('Survey_CS6750_FALL16_QC.csv')
CS6750_fall16_mc = read.csv('Survey_CS6750_FALL16_MC.csv')
CS6750_fall16_eoc = read.csv('Survey_CS6750_FALL16_EOC.csv')

# Read in grade data sets
grades = read.csv('Grades_CS6750_FALL16.csv', na.strings="")
```

```
# Create data subsets containing information of interest and change names
# CS6750 - HCI
CS6750_fall16_soc = CS6750_fall16_soc[, c(1, 2, 3, 4, 5, 7, 8, 11)]
colnames(CS6750_fall16_soc) = c("student", "age", "gender", "birth", "residence",
                                "language", "english", "education")

CS6750_fall16_qc = CS6750_fall16_qc[, c(1, 2, 3)]
colnames(CS6750_fall16_qc) = c("student", "conf_p1_post", "conf_p2_pre")

CS6750_fall16_mc = CS6750_fall16_mc[, c(1, 2, 3)]
colnames(CS6750_fall16_mc) = c("student", "conf_p2_post", "conf_p3_pre")

CS6750_fall16_eoc = CS6750_fall16_eoc[, c(1, 3, 2)]
colnames(CS6750_fall16_eoc) = c("student", "hours", "conf_p3_post")

colnames(grades) = c("student", "assign_p1", "assign_p2", "assign_p3", "assign_p4",
                     "assign_p5", "assign_m1", "assign_m2", "assign_m3", "extra",
                     "assign_m4", "assign_m5", "project", "feedback", "test1", "test2")

# Create grade summary variables
grades$assign_ave = 100*(grades$assign_p1 + grades$assign_p2 +
      grades$assign_p3 + grades$assign_p4 + grades$assign_p5 +
      grades$assign_m1 + grades$assign_m2 + grades$assign_m3 +
      grades$assign_m4 + grades$assign_m5)/200

grades$test_ave = 100*(grades$test1 + grades$test2)/200
```

```r
grades$total = (grades$assign_ave*0.4 + grades$test_ave*0.3 +
                  grades$project*0.2 + grades$feedback*0.1)

# Drop unnecessary fields from grades dataframe
grades = grades[,c("student", "assign_ave", "test_ave", "project", "feedback",
                   "total")]

# Merge HCI datasets
hci = merge(x = CS6750_fall16_soc, y = CS6750_fall16_qc, by = "student", all.x = TRUE)
hci = merge(x = hci, y = CS6750_fall16_mc, by = "student", all.x = TRUE)
hci = merge(x = hci, y = CS6750_fall16_eoc, by = "student", all.x = TRUE)
hci = merge(x = hci, y = grades, by = "student", all.x = TRUE)

hci$semester = "Fall 2016"

hci$course = "HCI"

# Drop unneeded datasets
rm(CS6750_fall16_soc, CS6750_fall16_qc, CS6750_fall16_mc, CS6750_fall16_eoc, grades)

# Replace blanks with NA
is.na(hci) = (hci=="")

# Convert factors into character strings
hci$student = as.character(hci$student)
hci$birth = as.character(hci$birth)
hci$residence = as.character(hci$residence)
hci$language = as.character(hci$language)

# Drop blank factor levels
hci$age = factor(hci$age)
hci$gender = factor(hci$gender)
hci$english = factor(hci$english)
hci$education = factor(hci$education)
hci$conf_p1_post = factor(hci$conf_p1_post)
hci$conf_p2_pre = factor(hci$conf_p2_pre)
hci$conf_p2_post = factor(hci$conf_p2_post)
hci$conf_p3_pre = factor(hci$conf_p3_pre)
hci$conf_p3_post = factor(hci$conf_p3_post)
hci$hours = factor(hci$hours)

# Simplify level names
hci$age = revalue(hci$age, c("No Answer" = NA))
hci$gender = revalue(hci$gender, c("No Answer" = NA))
hci$english = revalue(hci$english, c("Native speaker"="Native",
                          "Fully fluent (non-native speaker)"="Fluent",
                          "Partially fluent" = "Partial", "No Answer" = NA))

hci$education = revalue(hci$education, c("Bachelors Degree"="Bachelors",
                         "Doctoral Degree"="Doctorate",
                         "High School (or international equivalent)"="High School",
                         "Masters Degree" = "Masters", "No Answer" = NA))
```

```
## The following `from` values were not present in `x`: High School (or international equivalent)
```

```
hci$conf_p1_post = revalue(hci$conf_p1_post, c("Very confident" = 5, "Somewhat confident"
                      = 4, "Neither confident nor unconfident" = 3, "Somewhat unconfident"
                      = 2, "Very unconfident" = 1))
```

```
## The following `from` values were not present in `x`: Very unconfident
```

```
hci$conf_p2_pre = revalue(hci$conf_p2_pre, c("Very confident" = 5, "Somewhat confident"
                      = 4, "Neither confident nor unconfident" = 3, "Somewhat unconfident"
                      = 2, "Very unconfident" = 1))

hci$conf_p2_post = revalue(hci$conf_p2_post, c("Very confident" = 5, "Somewhat confident"
                      = 4, "Neither confident nor unconfident" = 3, "Somewhat unconfident"
                      = 2, "Very unconfident" = 1, "No Answer" = NA))
```

```
## The following `from` values were not present in `x`: Very unconfident, No Answer
```

```
hci$conf_p3_pre = revalue(hci$conf_p3_pre, c("Very confident" = 5, "Somewhat confident"
                      = 4, "Neither confident nor unconfident" = 3, "Somewhat unconfident"
                      = 2, "Very unconfident" = 1))
```

```
## The following `from` values were not present in `x`: Very unconfident
```

```
hci$conf_p3_post = revalue(hci$conf_p3_post, c("Very confident" = 5, "Somewhat confident"
                      = 4, "Neither confident nor unconfident" = 3, "Somewhat unconfident"
                      = 2, "Very unconfident" = 1, "No Answer" = NA))
```

```
## The following `from` values were not present in `x`: Somewhat unconfident, No Answer
```

```
hci$hours = revalue(hci$hours, c("No Answer" = NA))
```

```
## The following `from` values were not present in `x`: No Answer
```

```
hci$hours = revalue(hci$hours, c("<3 hours per week" = "0-3",
               "3 - 6 hours per week" = "3-6", "6 - 9 hours per week" = "6-9",
               "9 - 12 hours per week" = "9-12", "12 - 15 hours per week" = "12-15",
               "15 - 18 hours per week" = "15-18", "18 - 21 hours per week" = "18-21",
               "21 or more hours per week" = "21+"))
```

```
## The following `from` values were not present in `x`: <3 hours per week, 21 or more hours per week
```

```
hci$hours = factor(hci$hours, levels = c("0-3", "3-6", "6-9", "9-12", "12-15",
                          "15-18", "18-21", "21+"))

hci$age = factor(hci$age, levels = c("Under 18", "18 to 24", "25 to 34", "35 to 44",
                                     "45 to 54", "55 to 64"))

hci$course = factor(hci$course, levels = c("KBAI", "HCI", "EduTech"))
hci$semester = factor(hci$semester, levels = c("Fall 2016", "Summer 2016",
                                               "Spring 2016", "Fall 2015", "Summer 2015"))

# Create function for removing "1:" from text fields and convert to title case
text_split = function(x){
  x = unlist(strsplit(x, ": "))[2]
  return(toTitleCase(x))
}

# Remove "1:" from text fields
hci$birth = sapply(hci$birth, text_split)
```

```r
hci$residence = sapply(hci$residence, text_split)
hci$language = sapply(hci$language, text_split)

# Get lists of unique values
#unique(hci$birth)
#unique(hci$residence)
#unique(hci$language)

# Clean birth country names
hci$birth = ifelse(hci$birth %in% c("United States", "USA", "U.S.A.", "US", "Usa",
                    "Us", "The United States of America", "uSA", "United States of America",
                    "U.S.", "U.S", "Denver City, Tx", "Ethiopia - US Army Base"), "USA",
                    hci$birth)

hci$birth = ifelse(hci$birth %in% c("India", "INDIA"), "India", hci$birth)
hci$birth = ifelse(hci$birth %in% c("China", "People's Republic of China",
                    "P.R.CHINA", "Hong Kong, SAR", "Hong Kong", "CHINA", "China P.R."),
                    "China", hci$birth)
hci$birth = ifelse(hci$birth %in% c("South Korea", "Korea"), "Korea", hci$birth)
hci$birth = ifelse(hci$birth %in% c("Addis Ababa", "Ethiopia"), "Ethiopia",
                       hci$birth)
hci$birth = ifelse(hci$birth %in% c("United Kingdom", "England"), "UK",
                       hci$birth)
hci$birth = ifelse(hci$birth == "NA", NA, hci$birth)

# Create alternative birth groupings
hci$birth2 = hci$birth
hci$birth2 = ifelse(hci$birth %in% c("Syria", "Taiwan", "Vietnam",
    "Pakistan", "Japan", "Korea", "Kuwait", "Philippines", "Indonesia",
    "Sri Lanka", "Singapore", "Nepal", "Turkey", "Kazakhstan", "Iran",
    "Afghanistan", "Thailand", "Myanmar", "Lebanon", "Tunisia", "UAE",
    "Bangladesh", "Qatar", "Malaysia"), "Other Asia", hci$birth2)
hci$birth2 = ifelse(hci$birth %in% c("Ukraine", "Italy", "Norway",
    "Serbia", "Moldova", "Czech Republic", "Poland", "Russia", "Switzerland",
    "Germany", "Bulgaria", "UK", "Finland", "Romania", "Lithuania",
    "Luxembourg"), "Europe", hci$birth2)
hci$birth2 = ifelse(hci$birth %in% c("Puerto Rico", "Canada",
    "Dominican Republic", "Mexico", "Dominica", "El Salvador", "Cuba",
    "Haiti", "Bahamas", "Guatemala", "Panama", "Grenada", "Honduras",
    "Nicaragua", "The Bahamas", "Trinidad and Tobago"), "Other Nth America",
    hci$birth2)
hci$birth2 = ifelse(hci$birth %in% c("Peru", "Ecuador", "Colombia",
    "Brazil", "Argentina", "Chile"), "Sth America", hci$birth2)
hci$birth2 = ifelse(hci$birth %in% c("Nigeria", "Kenya",
    "South Africa", "Ethiopia", "Ghana", "Rwanda"), "Africa", hci$birth2)

hci$birth2 = ifelse(hci$birth %in% c("Australia", "New Zealand"),
    "Other", hci$birth2)

unique(hci$birth2)

# Clean residence country names
hci$residence = ifelse(hci$residence %in% c("United States", "USA", "U.S.A.", "US", "Usa",
```

```
                    "The United States of America", "uSA", "United States of America",
                    "United State", "USa", "Los Angeles", "Houston", "U.S", "U.S.", "YSA",
                    "Us", "United STates", "America", "JS"), "USA", hci$residence)

hci$residence = ifelse(hci$residence == "NA", NA, hci$residence)
hci$residence = ifelse(hci$residence == "Myanmar, Hong Kong", "Myanmar", hci$residence)
hci$residence = ifelse(hci$residence %in% c("China", "Hong Kong"), "China", hci$residence)
hci$residence = ifelse(hci$residence == "United Kingdom", "UK", hci$residence)

# Clean language
hci$language = ifelse(hci$language %in% c("English", "American English", "ENGLISH",
                   "American", "English (US)", "English Language", "Englist",
                   "C++, but you Probably Mean \"English\"", "ENGLISH", "En", "JavaScript",
                   "Elijah", "Dallas", "First",
                   "English and French", "English, Cantonese", "Java",
                   "Conative American Sign Language and English"), "English",
                   hci$language)
hci$language = ifelse(hci$language %in% c("Chinese", "Mandarin", "China",
                   "Mandarin Chinese", "Cantonese", "Chiinese", "CHINESE", "Manderin",
                   "Java", "Python"), "Chinese", hci$language)
hci$language = ifelse(hci$language %in% c("Marathi", "Telugu", "Bengali", "Gujarati",
                   "Kannada", "Hindi", "Tamil", "Odiya", "TAMIL", "Punjabi", "Hindo",
                   "Indian Language"), "Indian", hci$language)
hci$language = ifelse(hci$language %in% c("Principal", "Korean", "South Korean"),
                   "Korean", hci$language)
hci$language = ifelse(hci$language == "Farsi/English", "Farsi", hci$language)
hci$language = ifelse(hci$language == "Spanish/English", "Spanish", hci$language)
hci$language = ifelse(hci$language %in% c("Swiss German", "German", "Germany"),
                   "German", hci$language)
hci$language = ifelse(hci$language %in% c("Persian", "Persian (Farsi)"), "Farsi",
                   hci$language)
hci$language = ifelse(hci$language %in% c("Thai", "ABAP"), "Thai",
                   hci$language)
hci$language = ifelse(hci$language == "NA", NA, hci$language)


# Create factors
hci$birth = factor(hci$birth)
hci$birth2 = factor(hci$birth2)
hci$residence = factor(hci$residence)
hci$language = factor(hci$language)
hci$semester = factor(hci$semester)

# Convert confidence scores to numeric
hci$conf_p1_post = as.numeric(as.character(hci$conf_p1_post))
hci$conf_p2_pre = as.numeric(as.character(hci$conf_p2_pre))
hci$conf_p2_post = as.numeric(as.character(hci$conf_p2_post))
hci$conf_p3_pre = as.numeric(as.character(hci$conf_p3_pre))
hci$conf_p3_post = as.numeric(as.character(hci$conf_p3_post))

# Calculate average confidence scores
hci$conf_ave = (hci$conf_p1_post + hci$conf_p2_pre + hci$conf_p2_post +
                    hci$conf_p3_pre + hci$conf_p3_post)/5
```

```
hci$conf_pre_ave = (hci$conf_p2_pre + hci$conf_p3_pre)/2

hci$conf_post_ave = (hci$conf_p1_post + hci$conf_p2_post + hci$conf_p3_post)/3

# Convert ranges to numeric values
hci$age_num = revalue(hci$age, c("18 to 24"=21, "25 to 34"=29.5, "35 to 44"=39.5,
                                 "45 to 54"=49.5, "55 to 64"=59.5, "Under 18" = 18))
hci$age_num = as.numeric(as.character(hci$age_num))

hci$hours_num = revalue(hci$hours, c("0-3"=1.5, "3-6"=4.5, "6-9"=7.5, "9-12"=10.5,
                "12-15"=13.5, "15-18"=16.5, "18-21"=19.5, "21+"=21))
hci$hours_num = as.numeric(as.character(hci$hours_num))

# Create indicator variables
hci$native_ind = ifelse(hci$english == "Native", 1, 0)
hci$higher_ind = ifelse(hci$education %in% c("Masters", "Doctorate"), 1, 0)
hci$gender_ind = ifelse(hci$gender == "Male", 1, 0)

# Drop NA values
hci = subset(hci, !is.na(student))
```

Explore Data

```
# Calculate summary statistics
summary(hci)
```

```
##    student                age        gender       birth       residence
##  Length:83          Under 18: 0   Female:20   USA    :56   USA     :70
##  Class :character   18 to 24: 5   Male  :60   India  : 7   Canada  : 3
##  Mode  :character   25 to 34:45   NA's  : 3   Canada : 2   Kenya   : 3
##                     35 to 44:17               China  : 2   India   : 1
##                     45 to 54:10               Kenya  : 2   Malaysia: 1
##                     55 to 64: 3               (Other):12   (Other) : 3
##                     NA's    : 3               NA's   : 2   NA's    : 2
##     language     english       education   conf_p1_post    conf_p2_pre
##  English:69   Fluent :13   Bachelors:64   Min.   :2.000   Min.   :1.000
##  Indian : 3   Native :65   Doctorate: 2   1st Qu.:4.000   1st Qu.:4.000
##  Chinese: 2   Partial: 1   Masters  :14   Median :4.000   Median :4.000
##  Spanish: 2   NA's   : 4   NA's     : 3   Mean   :4.145   Mean   :4.289
##  Arabic : 1                               3rd Qu.:5.000   3rd Qu.:5.000
##  (Other): 4                               Max.   :5.000   Max.   :5.000
##  NA's   : 2                               NA's   :7       NA's   :7
##   conf_p2_post    conf_p3_pre        hours     conf_p3_post
##  Min.   :2.000   Min.   :2.000   6-9    :18   Min.   :1.000
##  1st Qu.:4.000   1st Qu.:4.000   9-12   :16   1st Qu.:4.000
##  Median :4.000   Median :4.000   3-6    : 8   Median :4.000
##  Mean   :4.219   Mean   :4.301   12-15  : 8   Mean   :4.327
##  3rd Qu.:5.000   3rd Qu.:5.000   15-18  : 1   3rd Qu.:5.000
##  Max.   :5.000   Max.   :5.000   (Other): 1   Max.   :5.000
##  NA's   :10      NA's   :10      NA's   :31   NA's   :31
##    assign_ave       test_ave        project          feedback
##  Min.   :56.40   Min.   :68.00   Min.   : 84.00   Min.   : 1.00
```

```
##  1st Qu.:86.45   1st Qu.:84.00   1st Qu.: 89.00   1st Qu.:28.00
##  Median :90.70   Median :87.75   Median : 91.50   Median :31.00
##  Mean   :88.73   Mean   :87.07   Mean   : 92.67   Mean   :29.14
##  3rd Qu.:92.88   3rd Qu.:91.38   3rd Qu.: 98.00   3rd Qu.:33.00
##  Max.   :98.10   Max.   :96.00   Max.   :100.00   Max.   :33.00
##                  NA's   :1
##      total           semester      course                birth2
##  Min.   :69.06   Fall 2016:83   KBAI   : 0   USA              :56
##  1st Qu.:81.08                  HCI    :83   India            : 7
##  Median :83.74                  EduTech: 0   Other Asia       : 6
##  Mean   :83.24                               Other Nth America: 5
##  3rd Qu.:86.09                               Africa           : 3
##  Max.   :90.08                               (Other)          : 4
##  NA's   :1                                   NA's             : 2
##     conf_ave     conf_pre_ave   conf_post_ave      age_num
##  Min.   :2.00   Min.   :1.500   Min.   :2.333   Min.   :21.00
##  1st Qu.:4.00   1st Qu.:4.000   1st Qu.:4.000   1st Qu.:29.50
##  Median :4.30   Median :4.500   Median :4.333   Median :29.50
##  Mean   :4.32   Mean   :4.304   Mean   :4.273   Mean   :34.72
##  3rd Qu.:4.80   3rd Qu.:5.000   3rd Qu.:4.667   3rd Qu.:39.50
##  Max.   :5.00   Max.   :5.000   Max.   :5.000   Max.   :59.50
##  NA's   :33     NA's   :14      NA's   :33      NA's   :3
##    hours_num        native_ind       higher_ind       gender_ind
##  Min.   : 4.500   Min.   :0.0000   Min.   :0.0000   Min.   :0.00
##  1st Qu.: 7.500   1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:0.75
##  Median : 9.000   Median :1.0000   Median :0.0000   Median :1.00
##  Mean   : 9.288   Mean   :0.8228   Mean   :0.1928   Mean   :0.75
##  3rd Qu.:10.500   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:1.00
##  Max.   :19.500   Max.   :1.0000   Max.   :1.0000   Max.   :1.00
##  NA's   :31       NA's   :4                         NA's   :3
```

```r
# Calculate proportion of class by gender
prop.table(table(hci$gender))
```

```
##
## Female   Male
##   0.25   0.75
```

Analyze Data by Gender

```r
# Calculate age summary statistics
ddply(subset(hci, !is.na(age_num) & !is.na(gender)), "gender", summarise,
      mean = mean(age_num),
      sd = sd(age_num), median = median(age_num), first_q = quantile(age_num, 0.25),
      third_q = quantile(age_num, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 32.72500 10.534998   29.5    29.5    32.0
## 2   Male 35.38333  8.603983   29.5    29.5    39.5
```

```r
# Calculate study hours summary statistics
ddply(subset(hci, !is.na(gender)&!is.na(hours_num)), "gender", summarise,
             mean = mean(hours_num), sd = sd(hours_num), median = median(hours_num),
             first_q = quantile(hours_num, 0.25), third_q = quantile(hours_num, 0.75))
```

```
##   gender mean       sd median first_q third_q
## 1 Female 9.00 3.240370      9     7.5    10.5
## 2   Male 9.15 3.034418      9     7.5    10.5
```

```
# Calculate confidence summary statistics
ddply(subset(hci, !is.na(gender)&!is.na(conf_ave)), "gender", summarise,
             mean = mean(conf_ave), sd = sd(conf_ave), median = median(conf_ave),
             first_q = quantile(conf_ave, 0.25), third_q = quantile(conf_ave, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 4.260000 0.3657564    4.3       4     4.4
## 2   Male 4.321053 0.5686199    4.3       4     4.8
```

```
# Calculate confidence summary statistics
ddply(subset(hci, !is.na(gender)&!is.na(conf_pre_ave)), "gender", summarise,
             mean = mean(conf_pre_ave), sd = sd(conf_pre_ave), median = median(conf_pre_ave),
             first_q = quantile(conf_pre_ave, 0.25), third_q = quantile(conf_pre_ave, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 3.928571 0.6753103    4.0   3.625       4
## 2   Male 4.386792 0.7248442    4.5   4.000       5
```

```
ddply(subset(hci, !is.na(gender)&!is.na(conf_post_ave)), "gender", summarise,
             mean = mean(conf_post_ave), sd = sd(conf_post_ave),
             median = median(conf_post_ave), first_q = quantile(conf_post_ave, 0.25),
             third_q = quantile(conf_post_ave, 0.75))
```

```
##   gender     mean        sd   median first_q  third_q
## 1 Female 4.333333 0.3849002 4.500000       4 4.666667
## 2   Male 4.245614 0.5407381 4.333333       4 4.666667
```

```
# Calculate grade summary statistics
ddply(subset(hci, !is.na(gender)&!is.na(total)), "gender", summarise,
             mean = mean(total), sd = sd(total),
             median = median(total), first_q = quantile(total, 0.25),
             third_q = quantile(total, 0.75))
```

```
##   gender     mean       sd median first_q third_q
## 1 Female 82.90474 5.347154  83.90 81.1500  85.925
## 2   Male 83.46467 3.840930  83.62 81.1675  86.090
```

```
hci_m = subset(hci, gender == "Male")
hci_f = subset(hci, gender == "Female")

# Compare age
prop.table(table(hci_m$age))
```

```
##
##   Under 18   18 to 24   25 to 34   35 to 44   45 to 54   55 to 64
## 0.00000000 0.03333333 0.55000000 0.23333333 0.16666667 0.01666667
```

```
prop.table(table(hci_f$age))
```

```
##
## Under 18 18 to 24 25 to 34 35 to 44 45 to 54 55 to 64
##     0.00     0.15     0.60     0.15     0.00     0.10
```

```
# Compare birth country
prop.table(table(hci_m$birth))
```

```
##
##              Canada               China             Ecuador
##          0.03333333          0.00000000          0.00000000
##               India                Iran               Italy
##          0.08333333          0.01666667          0.01666667
##               Kenya               Korea             Lebanon
##          0.00000000          0.00000000          0.01666667
##             Malaysia              Mexico             Myanmar
##          0.01666667          0.01666667          0.01666667
##             Pakistan              Panama              Rwanda
##          0.01666667          0.01666667          0.01666667
## Trinidad and Tobago                 USA
##          0.01666667          0.71666667
```

```
prop.table(table(hci_f$birth))
```

```
##
##              Canada               China             Ecuador
##                0.00                0.10                0.05
##               India                Iran               Italy
##                0.10                0.00                0.00
##               Kenya               Korea             Lebanon
##                0.10                0.05                0.00
##             Malaysia              Mexico             Myanmar
##                0.00                0.00                0.00
##             Pakistan              Panama              Rwanda
##                0.00                0.00                0.00
## Trinidad and Tobago                 USA
##                0.00                0.60
```

```
# Compare birth country2
prop.table(table(hci_m$birth2))
```

```
##
##            Africa             China            Europe             India
##        0.01666667        0.00000000        0.01666667        0.08333333
##        Other Asia Other Nth America       Sth America               USA
##        0.08333333        0.08333333        0.00000000        0.71666667
```

```
prop.table(table(hci_f$birth2))
```

```
##
##            Africa             China            Europe             India
##              0.10              0.10              0.00              0.10
##        Other Asia Other Nth America       Sth America               USA
##              0.05              0.00              0.05              0.60
```

```
# Compare country of residence
prop.table(table(hci_m$residence))
```

```
##
##     Canada      India      Kenya   Malaysia     Sweden     Taiwan
## 0.05000000 0.01666667 0.01666667 0.01666667 0.01666667 0.01666667
##         UK        USA
## 0.01666667 0.85000000
```

```r
prop.table(table(hci_f$residence))
```

```
##
##   Canada    India    Kenya Malaysia   Sweden   Taiwan       UK      USA
##      0.0      0.0      0.1      0.0      0.0      0.0      0.0      0.9
```

```r
# Compare language background
prop.table(table(hci_m$language))
```

```
##
##     Arabic    Burmese    Chinese    English      Farsi     Indian
## 0.00000000 0.01666667 0.01666667 0.88333333 0.01666667 0.03333333
##     Korean    Spanish       Urdu
## 0.00000000 0.01666667 0.01666667
```

```r
prop.table(table(hci_f$language))
```

```
##
##  Arabic Burmese Chinese English   Farsi  Indian  Korean Spanish    Urdu
##    0.05    0.00    0.05    0.75    0.00    0.05    0.05    0.05    0.00
```

```r
# Compare English skills
prop.table(table(hci_m$english))
```

```
##
##     Fluent     Native    Partial
## 0.13793103 0.84482759 0.01724138
```

```r
prop.table(table(hci_f$english))
```

```
##
##  Fluent  Native Partial
##    0.25    0.75    0.00
```

```r
# Compare education
prop.table(table(hci_m$education))
```

```
##
##  Bachelors  Doctorate    Masters
## 0.76271186 0.01694915 0.22033898
```

```r
prop.table(table(hci_f$education))
```

```
##
## Bachelors Doctorate   Masters
##      0.90      0.05      0.05
```

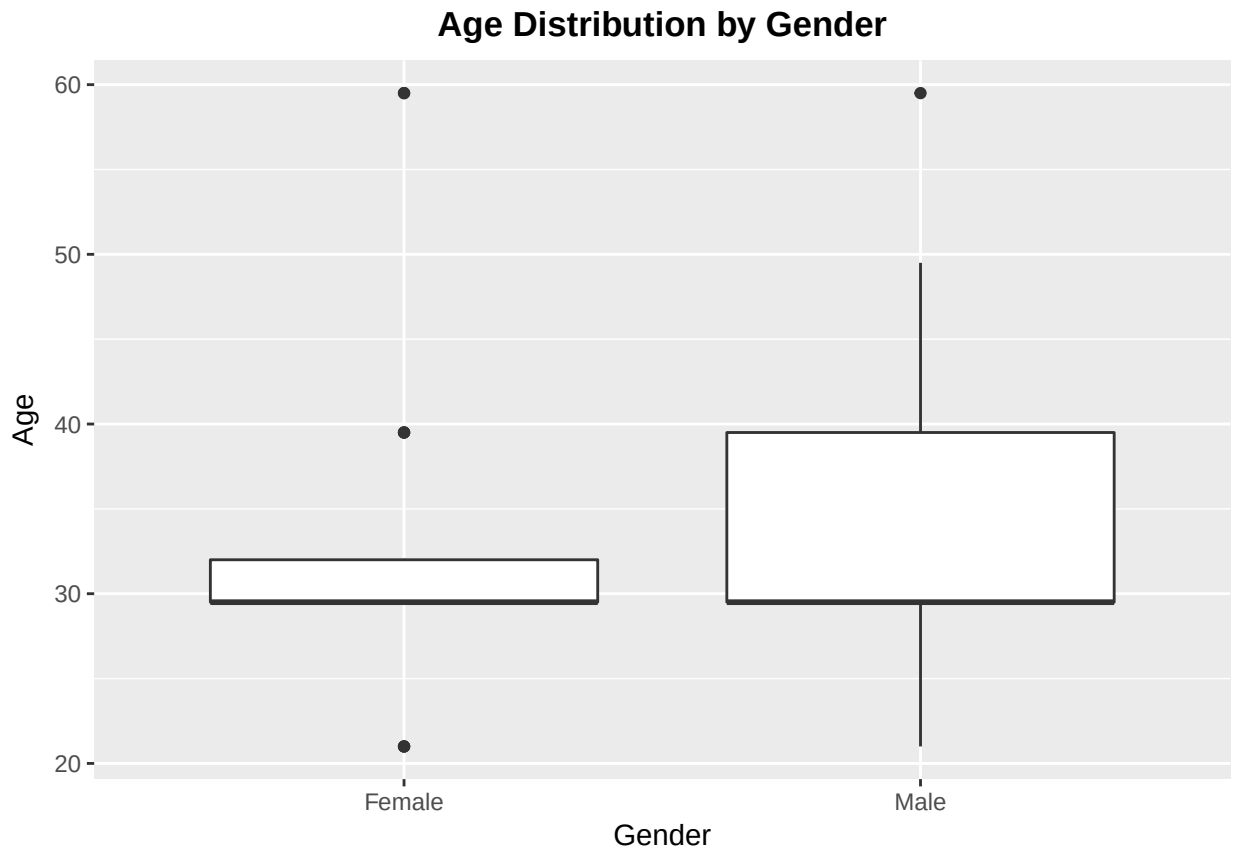
```r
# Compare hours
prop.table(table(hci_m$hours))
```

```
##
##   0-3   3-6   6-9  9-12 12-15 15-18 18-21   21+
## 0.000 0.150 0.350 0.325 0.150 0.025 0.000 0.000
```

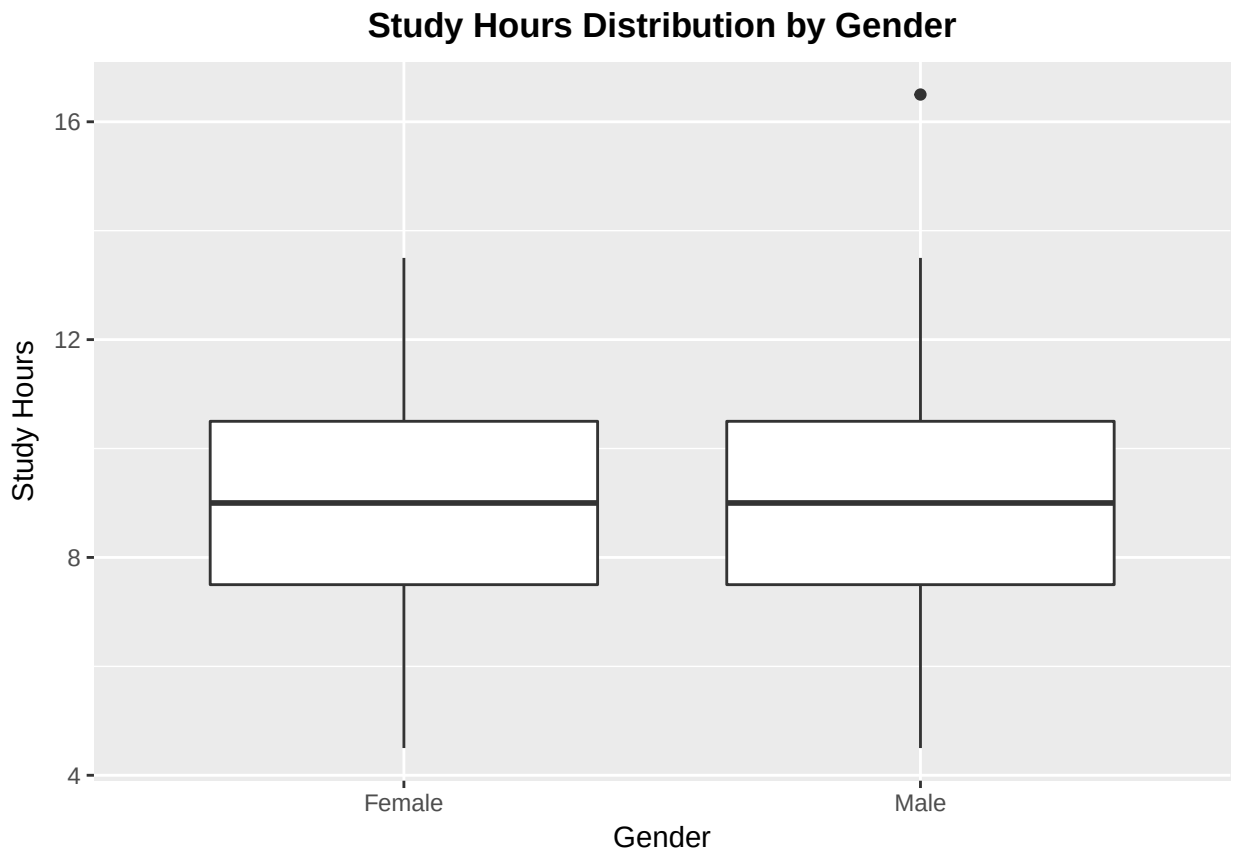
```r
prop.table(table(hci_f$hours))
```

```
##
##   0-3   3-6   6-9  9-12 12-15 15-18 18-21   21+
##   0.0   0.2   0.3   0.3   0.2   0.0   0.0   0.0
```

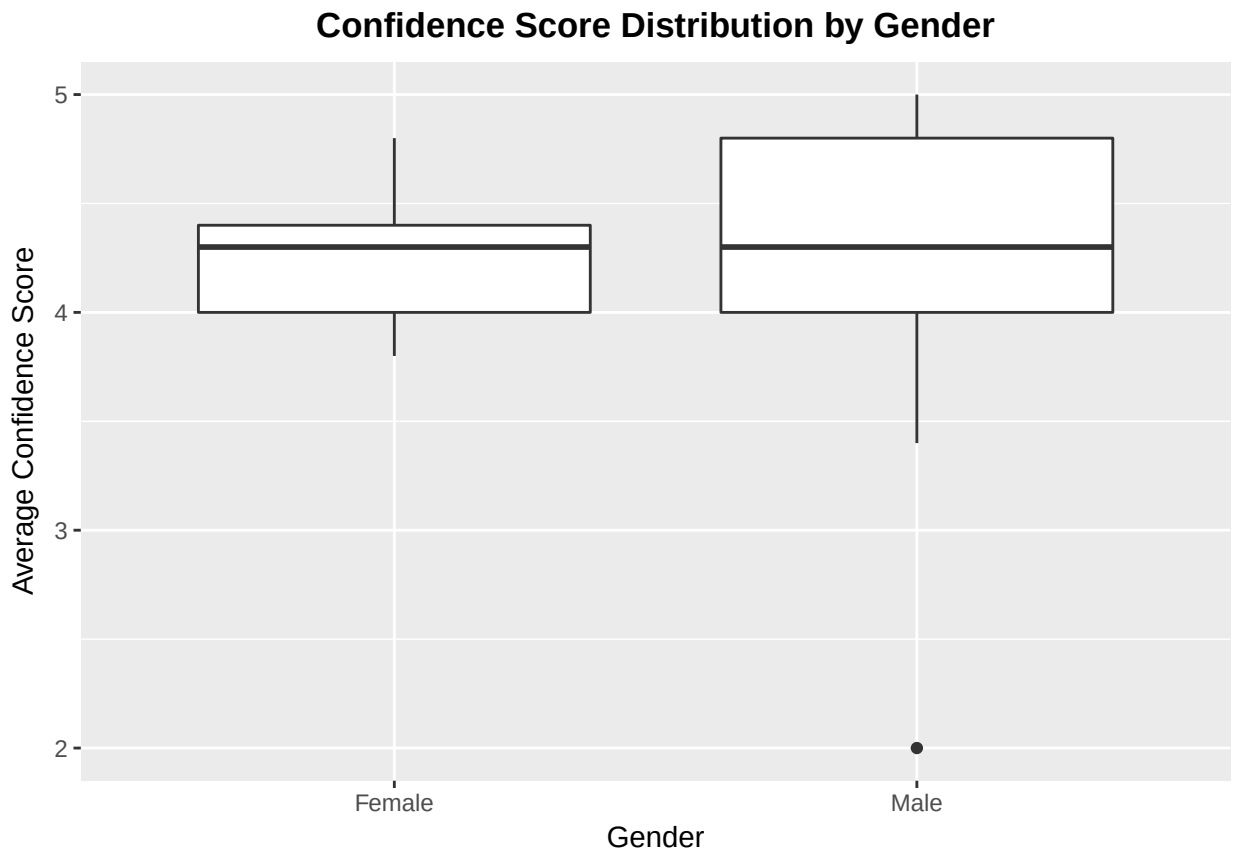
```r
#Boxplot of age distribution by gender
ggplot(subset(hci, !is.na(gender)), aes(gender, age_num)) +
 geom_boxplot() +
 labs(title = "Age Distribution by Gender",
      x = "Gender", y = "Age") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



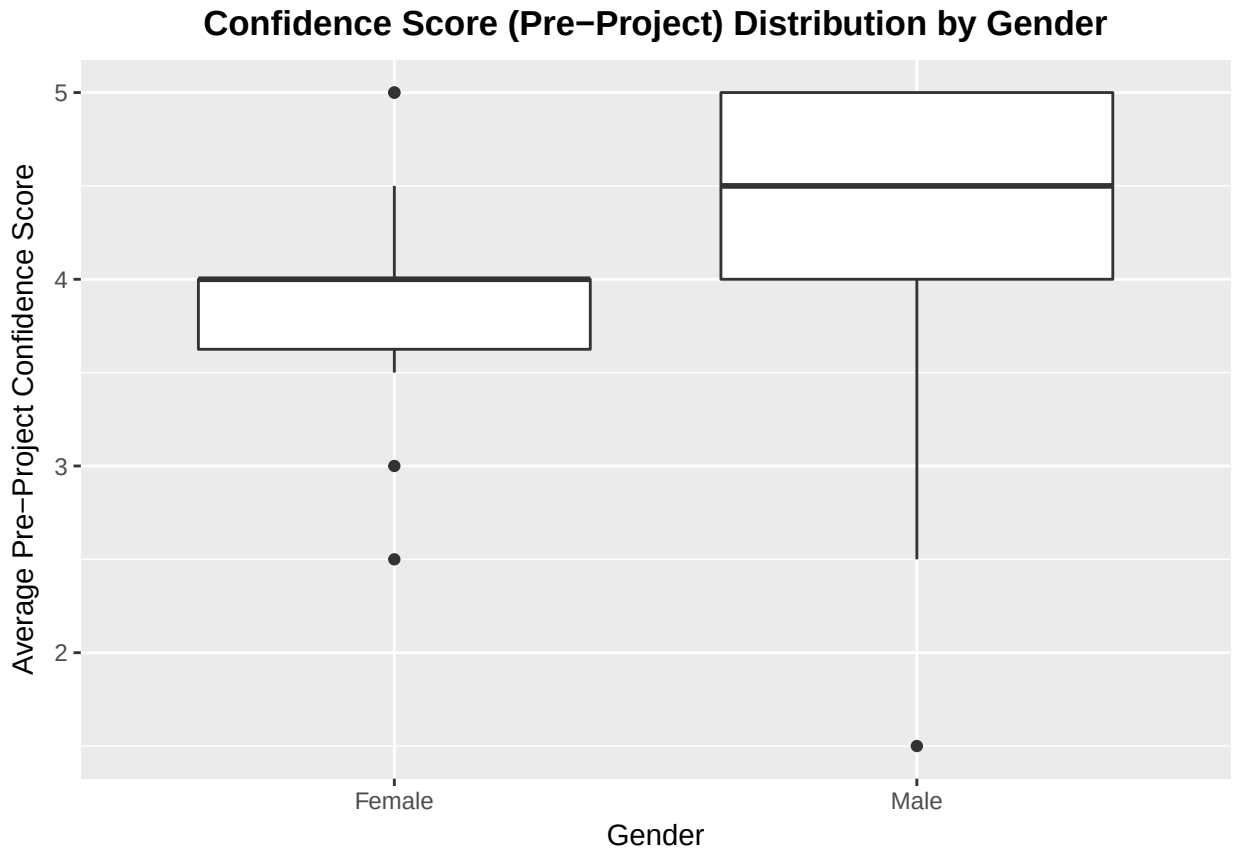

```r
# Boxplot of hours spent studying by gender
ggplot(subset(hci, !is.na(hours_num) & !is.na(gender)), aes(gender, hours_num)) +
  geom_boxplot() +
 labs(title = "Study Hours Distribution by Gender",
      x = "Gender", y = "Study Hours") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```




```
# Boxplot of confidence score by gender
ggplot(subset(hci, !is.na(conf_ave) & !is.na(gender)), aes(gender, conf_ave)) +
  geom_boxplot() +
 labs(title = "Confidence Score Distribution by Gender",
      x = "Gender", y = "Average Confidence Score") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```

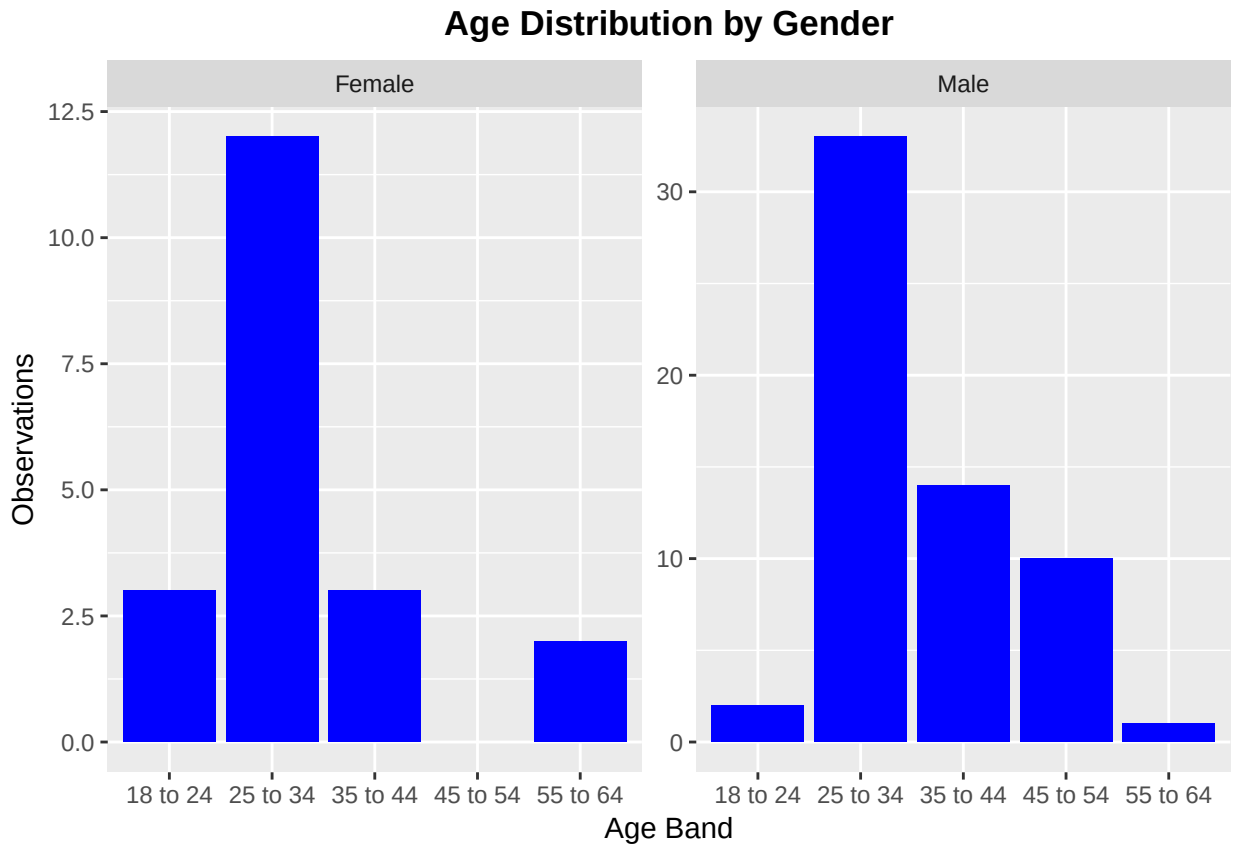



```
# Boxplot of confidence score (pre-project) by gender
ggplot(subset(hci, !is.na(conf_pre_ave) & !is.na(gender)), aes(gender,
    conf_pre_ave)) +  geom_boxplot() +
 labs(title = "Confidence Score (Pre-Project) Distribution by Gender",
      x = "Gender", y = "Average Pre-Project Confidence Score") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```

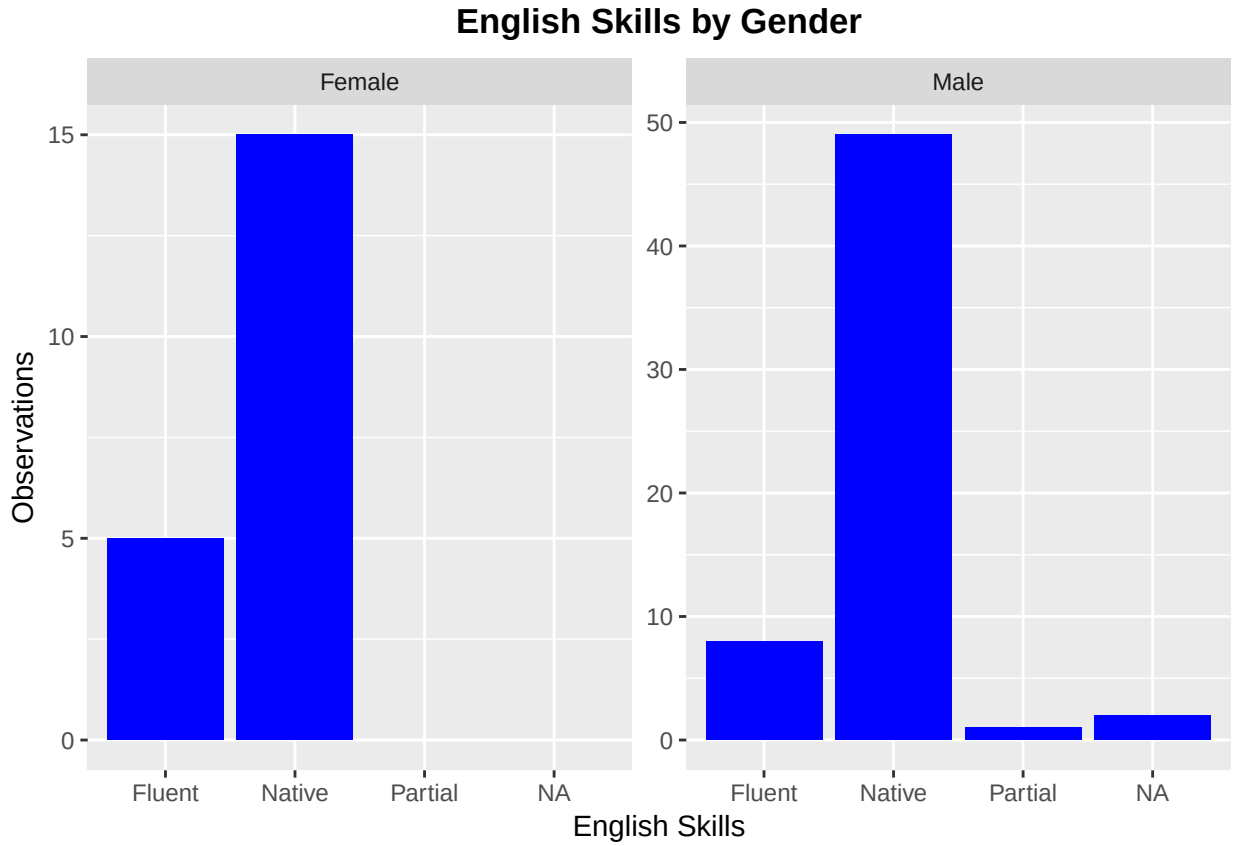



```
# Boxplot of confidence score (post-project) by gender
ggplot(subset(hci, !is.na(conf_post_ave) & !is.na(gender)), aes(gender,
    conf_post_ave)) +  geom_boxplot() +
 labs(title = "Confidence Score (Post-Project) Distribution by Gender",
      x = "Gender", y = "Average Post-Project Confidence Score") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```

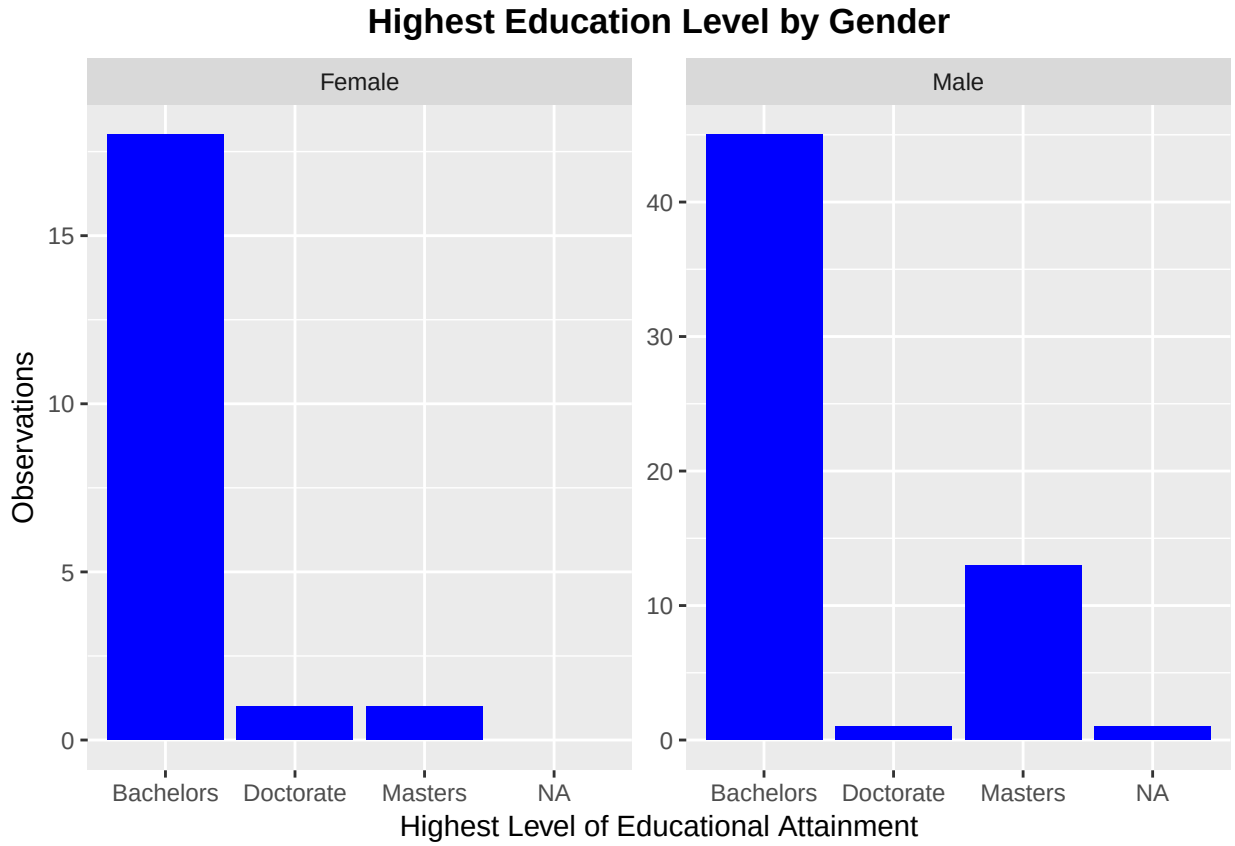

```
# Bar chart comparing age by gender
ggplot(subset(hci, !is.na(gender)), aes(x = age)) +
    geom_bar(fill = "blue") +
    facet_wrap(~gender, scales = "free_y") +
    labs(title = "Age Distribution by Gender",
      x = "Age Band",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```




```
# Bar chart comparing English skills by gender
ggplot(subset(hci, !is.na(gender)), aes(x = english)) +
    geom_bar(fill = "blue") +
    facet_wrap(~gender, scales = "free_y") +
    labs(title = "English Skills by Gender",
      x = "English Skills",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```

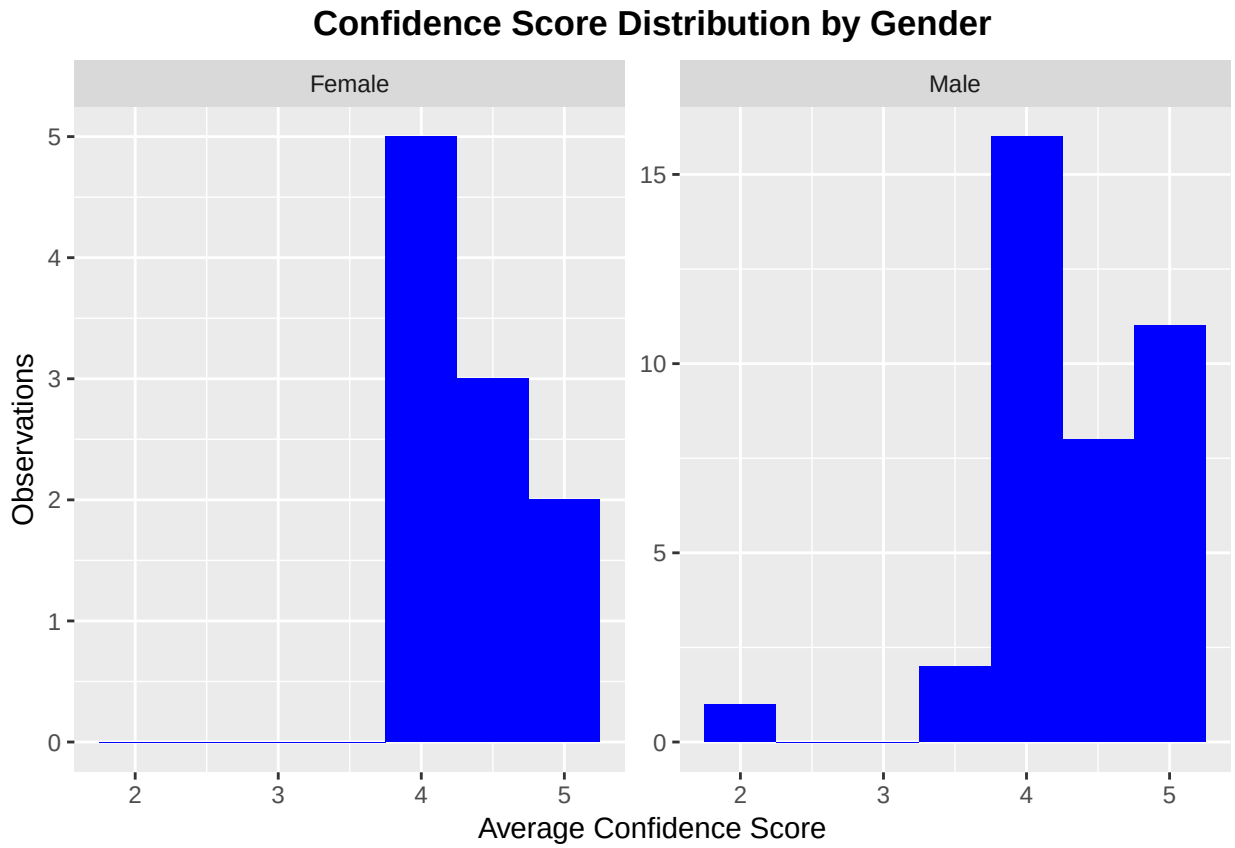



```
# Bar chart comparing education by gender
ggplot(subset(hci, !is.na(gender)), aes(x = education)) +
    geom_bar(fill = "blue") +
    facet_wrap(~gender, scales = "free_y") +
    labs(title = "Highest Education Level by Gender",
      x = "Highest Level of Educational Attainment",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



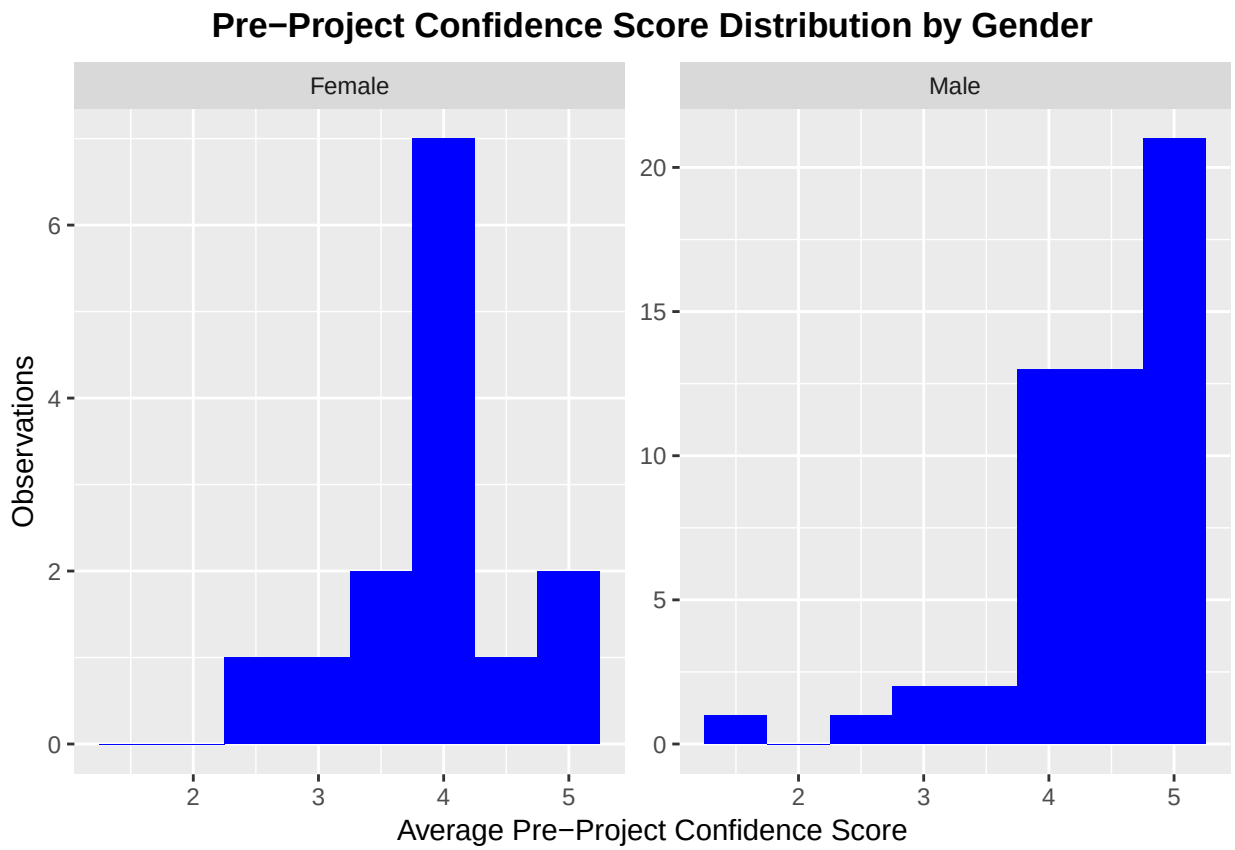

```
# Histogram of conf_ave by gender
ggplot(subset(hci, !is.na(gender)), aes(x = conf_ave)) +
    geom_histogram(fill = "blue", binwidth = 0.5) +
    facet_wrap(~gender, scale = "free_y") +
    labs(title = "Confidence Score Distribution by Gender",
      x = "Average Confidence Score",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```

```
## Warning: Removed 32 rows containing non-finite values (stat_bin).
```



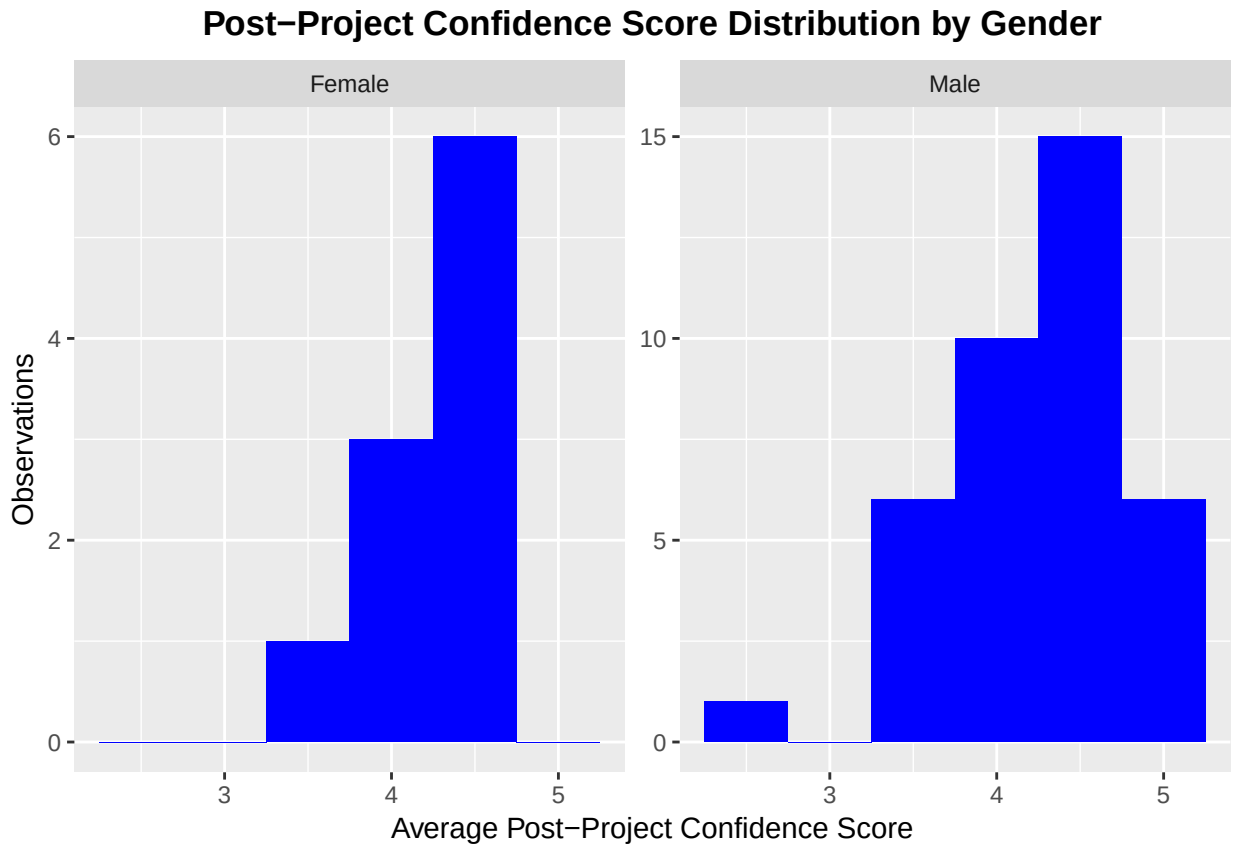

```
# Histogram of conf_pre_ave by gender
ggplot(subset(hci, !is.na(gender)), aes(x = conf_pre_ave)) +
    geom_histogram(fill = "blue", binwidth = 0.5) +
    facet_wrap(~gender, scale = "free_y") +
    labs(title = "Pre-Project Confidence Score Distribution by Gender",
      x = "Average Pre-Project Confidence Score",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```

```
## Warning: Removed 13 rows containing non-finite values (stat_bin).
```



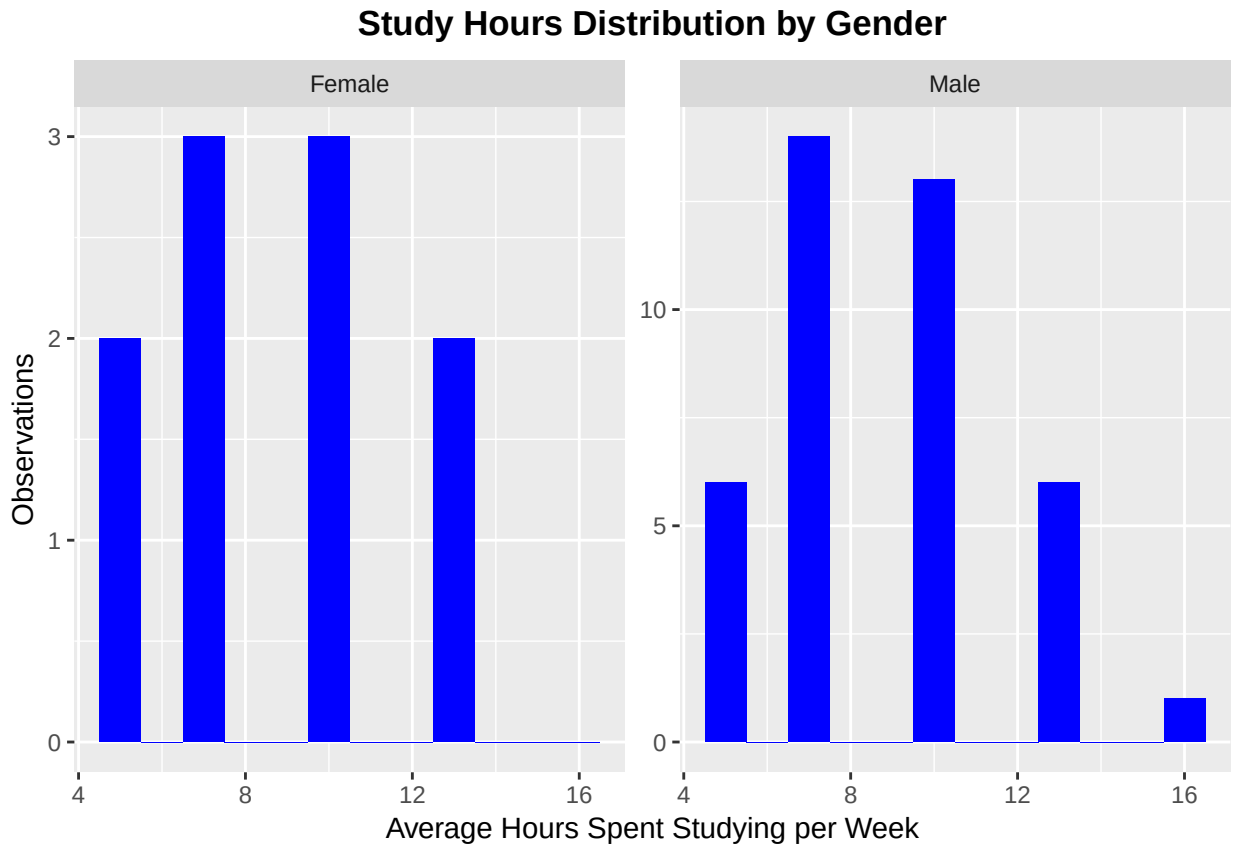

```
# Histogram of conf_post_ave by gender
ggplot(subset(hci, !is.na(gender)), aes(x = conf_post_ave)) +
    geom_histogram(fill = "blue", binwidth = 0.5) +
    facet_wrap(~gender, scale = "free_y") +
    labs(title = "Post-Project Confidence Score Distribution by Gender",
      x = "Average Post-Project Confidence Score",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```

```
## Warning: Removed 32 rows containing non-finite values (stat_bin).
```



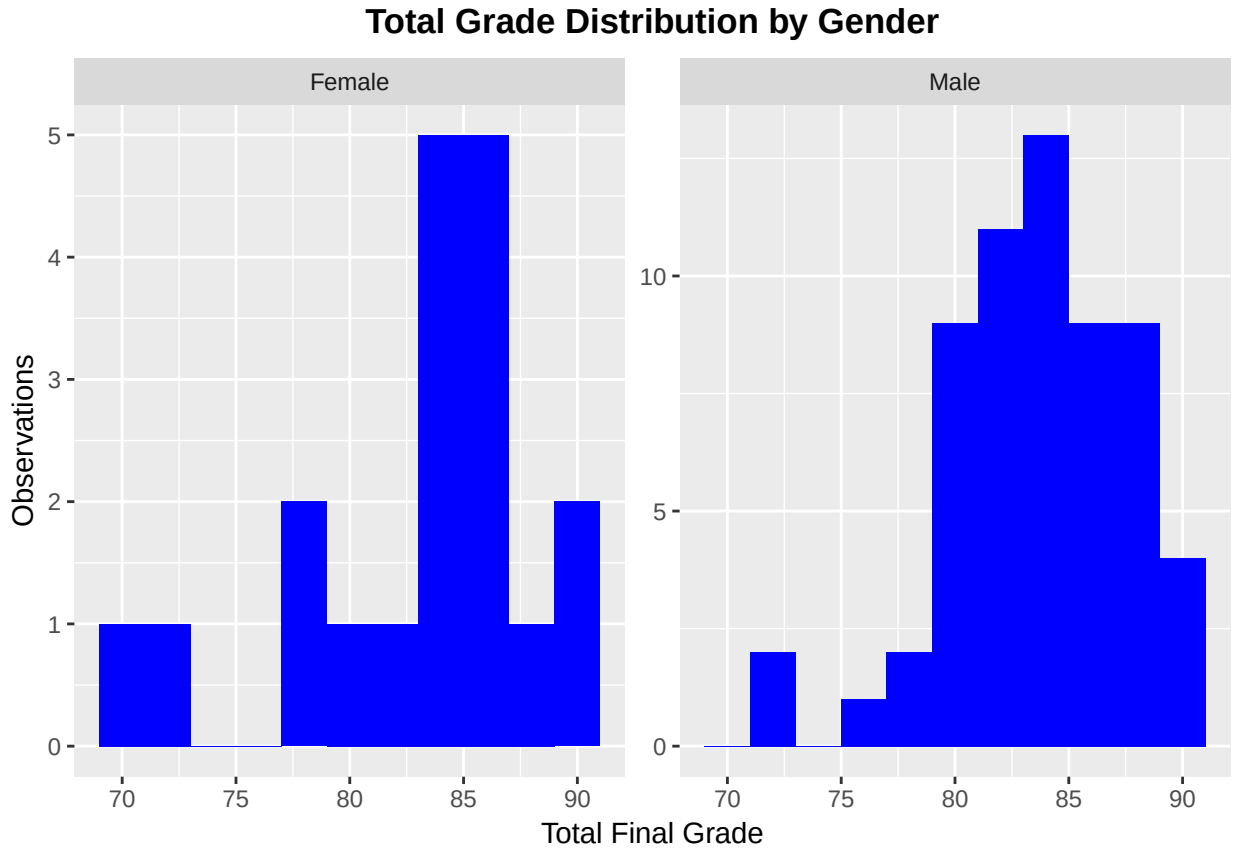

```
# Histogram of study hours by gender
ggplot(subset(hci, !is.na(gender)), aes(x = hours_num)) +
    geom_histogram(fill = "blue", binwidth = 1) +
    facet_wrap(~gender, scale = "free_y") +
    labs(title = "Study Hours Distribution by Gender",
      x = "Average Hours Spent Studying per Week",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```

```
## Warning: Removed 30 rows containing non-finite values (stat_bin).
```




```
# Histogram of grades by gender
ggplot(subset(hci, !is.na(gender)), aes(x = total)) +
    geom_histogram(fill = "blue", binwidth = 2) +
    facet_wrap(~gender, scale = "free_y") +
    labs(title = "Total Grade Distribution by Gender",
      x = "Total Final Grade",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```

```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```




```
# Age tests
t.test(hci_m$age_num, hci_f$age_num)
```

```
##
##  Welch Two Sample t-test
##
## data:  hci_m$age_num and hci_f$age_num
## t = 1.0207, df = 27.943, p-value = 0.3162
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.677113  7.993780
## sample estimates:
## mean of x mean of y
##  35.38333  32.72500
```

```
wilcox.test(age_num ~ gender, data=hci)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  age_num by gender
## W = 470, p-value = 0.1099
## alternative hypothesis: true location shift is not equal to 0
```

```
# Higher ed tests
t.test(hci_m$higher_ind, hci_f$higher_ind)
```

```
##
```

```
##  Welch Two Sample t-test
##
## data:  hci_m$higher_ind and hci_f$higher_ind
## t = 1.5127, df = 45.151, p-value = 0.1373
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.04417563  0.31084230
## sample estimates:
## mean of x mean of y
## 0.2333333 0.1000000
```

```
wilcox.test(higher_ind ~ gender, data=hci)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  higher_ind by gender
## W = 520, p-value = 0.2024
## alternative hypothesis: true location shift is not equal to 0
```

```
# Native speaker test
t.test(hci_m$native_ind, hci_f$native_ind)
```

```
##
##  Welch Two Sample t-test
##
## data:  hci_m$native_ind and hci_f$native_ind
## t = 0.85965, df = 28.374, p-value = 0.3972
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1309981  0.3206533
## sample estimates:
## mean of x mean of y
## 0.8448276 0.7500000
```

```
wilcox.test(native_ind ~ gender, data=hci)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  native_ind by gender
## W = 525, p-value = 0.3481
## alternative hypothesis: true location shift is not equal to 0
```

```
# Average confidence score tests
t.test(hci_m$conf_ave, hci_f$conf_ave)
```

```
##
##  Welch Two Sample t-test
##
## data:  hci_m$conf_ave and hci_f$conf_ave
## t = 0.41268, df = 21.931, p-value = 0.6838
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.2458132  0.3679184
## sample estimates:
## mean of x mean of y
```

```
##  4.321053  4.260000
```

```
wilcox.test(conf_ave ~ gender, data=hci)
```

```
## Warning in wilcox.test.default(x = c(3.8, 4, 4.8, 4.8, 4.4, 3.8, 4, 4.4, :
## cannot compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  conf_ave by gender
## W = 160, p-value = 0.4479
## alternative hypothesis: true location shift is not equal to 0
```

```
# Average pre-project confidence score tests
t.test(hci_m$conf_pre_ave, hci_f$conf_pre_ave)
```

```
##
##  Welch Two Sample t-test
##
## data:  hci_m$conf_pre_ave and hci_f$conf_pre_ave
## t = 2.223, df = 21.616, p-value = 0.037
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.03030185 0.88614019
## sample estimates:
## mean of x mean of y
##  4.386792  3.928571
```

```
wilcox.test(conf_pre_ave ~ gender, data=hci)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  conf_pre_ave by gender
## W = 212.5, p-value = 0.01124
## alternative hypothesis: true location shift is not equal to 0
```

```
# Average post-project confidence score tests
t.test(hci_m$conf_post_ave, hci_f$conf_post_ave)
```

```
##
##  Welch Two Sample t-test
##
## data:  hci_m$conf_post_ave and hci_f$conf_post_ave
## t = -0.58467, df = 19.498, p-value = 0.5655
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.4011977  0.2257591
## sample estimates:
## mean of x mean of y
##  4.245614  4.333333
```

```
wilcox.test(conf_post_ave ~ gender, data=hci)
```

```
## Warning in wilcox.test.default(x = c(4, 4, 4.66666666666667,
## 4.66666666666667, : cannot compute exact p-value with ties
```

```
##
```

```
##  Wilcoxon rank sum test with continuity correction
##
## data:  conf_post_ave by gender
## W = 206.5, p-value = 0.6776
## alternative hypothesis: true location shift is not equal to 0
```

```
# Study hours
t.test(hci_m$hours_num, hci_f$hours_num)
```

```
##
##  Welch Two Sample t-test
##
## data:  hci_m$hours_num and hci_f$hours_num
## t = 0.13257, df = 13.232, p-value = 0.8965
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.290013  2.590013
## sample estimates:
## mean of x mean of y
##      9.15      9.00
```

```
wilcox.test(hours_num ~ gender, data=hci)
```

```
## Warning in wilcox.test.default(x = c(13.5, 10.5, 4.5, 4.5, 10.5, 10.5,
## 7.5, : cannot compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  hours_num by gender
## W = 196.5, p-value = 0.9395
## alternative hypothesis: true location shift is not equal to 0
```

```
# Total grade
t.test(hci_m$total, hci_f$total)
```

```
##
##  Welch Two Sample t-test
##
## data:  hci_m$total and hci_f$total
## t = 0.42318, df = 24.166, p-value = 0.6759
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.169925  3.289785
## sample estimates:
## mean of x mean of y
##  83.46467  82.90474
```

```
wilcox.test(total ~ gender, data=hci)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  total by gender
## W = 581, p-value = 0.9041
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Check for multicollinearity
cor_subset = hci[, c("age_num", "native_ind", "higher_ind", "gender_ind")]
cor(na.omit(cor_subset))
```

```
##              age_num  native_ind  higher_ind gender_ind
## age_num    1.0000000  0.10507418  0.41732788  0.1219176
## native_ind 0.1050742  1.00000000 -0.01060694  0.1078971
## higher_ind 0.4173279 -0.01060694  1.00000000  0.1528829
## gender_ind 0.1219176  0.10789708  0.15288294  1.0000000
```

```r
# Fit regression to confidence score
conf_lm = lm(conf_ave~gender + age_num + native_ind + higher_ind,
             data=na.omit(hci))

summary(conf_lm)
```

```
##
## Call:
## lm(formula = conf_ave ~ gender + age_num + native_ind + higher_ind,
##     data = na.omit(hci))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.18756 -0.30418 -0.04504  0.40641  0.81244
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.657331   0.471268   9.883 1.59e-12 ***
## genderMale   0.018034   0.200772   0.090    0.929
## age_num     -0.005831   0.008927  -0.653    0.517
## native_ind  -0.199164   0.272603  -0.731    0.469
## higher_ind   0.077707   0.197039   0.394    0.695
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5518 on 42 degrees of freedom
## Multiple R-squared:  0.02438,    Adjusted R-squared:  -0.06854
## F-statistic: 0.2624 on 4 and 42 DF,  p-value: 0.9004
```

```r
# Fit regression to pre-project confidence score
conf_pre_lm = lm(conf_pre_ave~gender + age_num + native_ind + higher_ind,
             data=na.omit(hci))

summary(conf_pre_lm)
```

```
##
## Call:
## lm(formula = conf_pre_ave ~ gender + age_num + native_ind + higher_ind,
##     data = na.omit(hci))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7000 -0.3912  0.1088  0.5681  0.8039
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.03476    0.58168   8.656 6.94e-11 ***
## genderMale   0.19515    0.24781   0.788    0.435
## age_num     -0.00956    0.01102  -0.868    0.391
## native_ind  -0.55665    0.33647  -1.654    0.106
## higher_ind   0.11140    0.24320   0.458    0.649
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6811 on 42 degrees of freedom
## Multiple R-squared:  0.1035, Adjusted R-squared:  0.01808
## F-statistic: 1.212 on 4 and 42 DF,  p-value: 0.3201
```

```
# Fit regression to post-project confidence score
conf_post_lm = lm(conf_post_ave~gender + age_num + native_ind + higher_ind,
             data=na.omit(hci))

summary(conf_post_lm)
```

```
##
## Call:
## lm(formula = conf_post_ave ~ gender + age_num + native_ind +
##     higher_ind, data = na.omit(hci))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.84592 -0.26795  0.06538  0.32047  0.82074
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.405712   0.454840   9.686 2.88e-12 ***
## genderMale  -0.100047   0.193774  -0.516    0.608
## age_num     -0.003345   0.008616  -0.388    0.700
## native_ind   0.039162   0.263100   0.149    0.882
## higher_ind   0.055246   0.190171   0.291    0.773
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5326 on 42 degrees of freedom
## Multiple R-squared:  0.01106,    Adjusted R-squared:  -0.08312
## F-statistic: 0.1174 on 4 and 42 DF,  p-value: 0.9756
```

```
# Fit regression to study hours
hours_lm = lm(hours_num~gender + age_num + native_ind + higher_ind,
             data=na.omit(hci))

summary(hours_lm)
```

```
##
## Call:
## lm(formula = hours_num ~ gender + age_num + native_ind + higher_ind,
##     data = na.omit(hci))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.215 -2.163 -0.760  2.240  6.008
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.61912    2.39835   2.760  0.00853 **
## genderMale  -0.04761    1.02176  -0.047  0.96306
## age_num      0.12225    0.04543   2.691  0.01018 *
## native_ind  -1.91801    1.38731  -1.383  0.17412
## higher_ind  -0.90833    1.00276  -0.906  0.37019
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.808 on 42 degrees of freedom
## Multiple R-squared:  0.1946, Adjusted R-squared:  0.1179
## F-statistic: 2.538 on 4 and 42 DF,  p-value: 0.05398
```

```
# Fit regression to grades
grades_lm = lm(total~gender + age_num + native_ind + higher_ind,
            data=na.omit(hci))

summary(grades_lm)
```

```
##
## Call:
## lm(formula = total ~ gender + age_num + native_ind + higher_ind,
##     data = na.omit(hci))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.090 -2.469  0.427  2.327  7.922
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 82.257186   3.167026  25.973   <2e-16 ***
## genderMale  -1.502252   1.349235  -1.113   0.2719
## age_num      0.000182   0.059993   0.003   0.9976
## native_ind   3.404973   1.831954   1.859   0.0701 .
## higher_ind   0.605856   1.324148   0.458   0.6496
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.708 on 42 degrees of freedom
## Multiple R-squared:  0.1163, Adjusted R-squared:  0.03216
## F-statistic: 1.382 on 4 and 42 DF,  p-value: 0.2565
```