Educational Technology Project - KBAI (Summer 2015 and Summer 2016) Data Analysis

Process Data

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

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

# Read in survey data sets
survey_sum15_soc = read.csv('Survey_CS7637_SUM15_SOC.csv')
survey_sum15_qc = read.csv('Survey_CS7637_SUM15_QC.csv')
survey_sum15_mc = read.csv('Survey_CS7637_SUM15_MC.csv')
survey_sum15_eoc = read.csv('Survey_CS7637_SUM15_EOC.csv')

survey_sum16_soc = read.csv('Survey_CS7637_SUM16_SOC.csv')
survey_sum16_qc = read.csv('Survey_CS7637_SUM16_QC.csv')
survey_sum16_mc = read.csv('Survey_CS7637_SUM16_MC.csv')
survey_sum16_eoc = read.csv('Survey_CS7637_SUM16_EOC.csv')

# Read in grade data sets
grades_sum15 = read.csv('Grades_CS7637_SUM15.csv', na.strings="")
grades_sum16 = read.csv('Grades_CS7637_SUM16.csv', na.strings="")
```

```
# Create data subsets containing information of interest and change names
survey_sum15_soc = survey_sum15_soc[, c(1, 2, 3, 4, 5, 7, 8, 16, 20)]
colnames(survey_sum15_soc) = c("student", "age", "gender", "birth", "residence",
                               "language", "english", "education", "programming")

survey_sum16_soc = survey_sum16_soc[, c(1, 2, 3, 4, 5, 7, 8, 11, 15)]
colnames(survey_sum16_soc) = c("student", "age", "gender", "birth", "residence",
                               "language", "english", "education", "programming")

survey_sum15_qc = survey_sum15_qc[, c(1, 4, 5)]
colnames(survey_sum15_qc) = c("student", "conf_p1_post", "conf_p2_pre")

survey_sum16_qc = survey_sum16_qc[, c(1, 3, 4)]
colnames(survey_sum16_qc) = c("student", "conf_p1_post", "conf_p2_pre")

survey_sum15_mc = survey_sum15_mc[, c(1, 4, 5)]
colnames(survey_sum15_mc) = c("student", "conf_p2_post", "conf_p3_pre")

survey_sum16_mc = survey_sum16_mc[, c(1, 3, 4)]
colnames(survey_sum16_mc) = c("student", "conf_p2_post", "conf_p3_pre")

survey_sum15_eoc = survey_sum15_eoc[, c(1, 2, 3)]
```

```
colnames(survey_sum15_eoc) = c("student", "conf_p3_post", "hours")

survey_sum16_eoc = survey_sum16_eoc[, c(1, 2, 3)]
colnames(survey_sum16_eoc) = c("student", "conf_p3_post", "hours")


colnames(grades_sum15) = c("student", "assign1", "assign2", "proj1", "assign3", "assign4",
                           "proj2", "assign5", "assign6", "proj3", "exam", "feedback")

colnames(grades_sum16) = c("student", "proj1", "proj2", "proj3", "assign1", "assign2",
                           "assign3", "assign4", "assign5", "assign6",  "exam", "feedback")

# Create grade summary variables
grades_sum15$assign_ave = 100*(grades_sum15$assign1 + grades_sum15$assign2 +
                           grades_sum15$assign3 + grades_sum15$assign4 +
                           grades_sum15$assign5 + grades_sum15$assign6)/120


grades_sum15$proj_ave = 100*(grades_sum15$proj1 + grades_sum15$proj2 +
                         grades_sum15$proj3)/300

grades_sum15$total = (grades_sum15$assign_ave*0.2 + grades_sum15$proj_ave*0.45 +
                      grades_sum15$exam*0.2 + (100*grades_sum15$feedback/15)*0.15)

grades_sum16$assign_ave = 100*(grades_sum16$assign1 + grades_sum16$assign2 +
                           grades_sum16$assign3 + grades_sum16$assign4 +
                           grades_sum16$assign5 + grades_sum16$assign6)/120


grades_sum16$proj_ave = 100*(grades_sum16$proj1 + grades_sum16$proj2 +
                         grades_sum16$proj3)/300

grades_sum16$total = (grades_sum16$assign_ave*0.2 + grades_sum16$proj_ave*0.45 +
                      grades_sum16$exam*0.2 + (100*grades_sum16$feedback/24)*0.15)

# Drop unnecessary fields from grades dataframes
grades_sum15 = grades_sum15[,c("student", "exam", "assign_ave", "proj_ave", "total")]
grades_sum16 = grades_sum16[,c("student", "exam", "assign_ave", "proj_ave", "total")]

# Merge datasets
kbai_data_sum15 = merge(x = survey_sum15_soc, y = survey_sum15_qc,
                        by = "student", all.x = TRUE)
kbai_data_sum15 = merge(x = kbai_data_sum15, y = survey_sum15_mc,
                        by = "student", all.x = TRUE)
kbai_data_sum15 = merge(x = kbai_data_sum15, y = survey_sum15_eoc,
                        by = "student", all.x = TRUE)
kbai_data_sum15 = merge(x = kbai_data_sum15, y = grades_sum15,
                        by = "student", all.x = TRUE)


kbai_data_sum16 = merge(x = survey_sum16_soc, y = survey_sum16_qc,
                        by = "student", all.x = TRUE)
kbai_data_sum16 = merge(x = kbai_data_sum16, y = survey_sum16_mc,
```

```
                        by = "student", all.x = TRUE)
kbai_data_sum16 = merge(x = kbai_data_sum16, y = survey_sum16_eoc,
                        by = "student", all.x = TRUE)
kbai_data_sum16 = merge(x = kbai_data_sum16, y = grades_sum16,
                        by = "student", all.x = TRUE)

kbai_data_sum15$semester = "Summer 2015"
kbai_data_sum16$semester = "Summer 2016"

kbai = rbind(kbai_data_sum15, kbai_data_sum16)

# Drop unneeded datasets
rm(grades_sum15, grades_sum16, kbai_data_sum15, kbai_data_sum16, survey_sum15_eoc,
   survey_sum15_mc, survey_sum15_qc, survey_sum15_soc, survey_sum16_eoc, survey_sum16_mc,
   survey_sum16_qc, survey_sum16_soc)

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

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

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

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

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

kbai$programming = revalue(kbai$programming, c("No Answer" = NA))

kbai$conf_p1_post = revalue(kbai$conf_p1_post, c("Very confident" = 5, "Somewhat confident"
                      = 4, "Neither confident nor unconfident" = 3, "Somewhat unconfident"
                      = 2, "Very unconfident" = 1, "No Answer" = NA))
```

```
kbai$conf_p2_pre = revalue(kbai$conf_p2_pre, c("Very confident" = 5, "Somewhat confident"
                      = 4, "Neither confident nor unconfident" = 3, "Somewhat unconfident"
                      = 2, "Very unconfident" = 1, "No Answer" = NA))

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

kbai$conf_p3_pre = revalue(kbai$conf_p3_pre, c("Very confident" = 5, "Somewhat confident"
                      = 4, "Neither confident nor unconfident" = 3, "Somewhat unconfident"
                      = 2, "Very unconfident" = 1, "No Answer" = NA))

kbai$conf_p3_post = revalue(kbai$conf_p3_post, c("Very confident" = 5, "Somewhat confident"
                      = 4, "Neither confident nor unconfident" = 3, "Somewhat unconfident"
                      = 2, "Very unconfident" = 1))

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

kbai$programming = factor(kbai$programming, levels = c("0", "1-3", "3-5", "5-10",
                      "10-15", "15-20", "20+"))

# 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
kbai$birth = sapply(kbai$birth, text_split)
kbai$residence = sapply(kbai$residence, text_split)
kbai$language = sapply(kbai$language, text_split)

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

# Clean birth country names
kbai$birth = ifelse(kbai$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"), "USA", kbai$birth)

kbai$birth = ifelse(kbai$birth %in% c("India", "INDIA"), "India", kbai$birth)
kbai$birth = ifelse(kbai$birth %in% c("China", "People's Republic of China", "P.R.CHINA",
                                      "Hong Kong, SAR", "Hong Kong"), "China", kbai$birth)
kbai$birth = ifelse(kbai$birth %in% c("South Korea", "Korea"), "Korea", kbai$birth)
kbai$birth = ifelse(kbai$birth %in% c("Addis Ababa", "Ethiopia"), "Ethiopia", kbai$birth)
```

```r
kbai$birth = ifelse(kbai$birth == "NA", NA, kbai$birth)

# Clean residence country names
kbai$residence = ifelse(kbai$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"), "USA", kbai$residence)

kbai$residence = ifelse(kbai$residence == "NA", NA, kbai$residence)
kbai$residence = ifelse(kbai$residence == "Myanmar, Hong Kong", "Myanmar", kbai$residence)
kbai$residence = ifelse(kbai$residence %in% c("China", "Hong Kong"), "China", kbai$residence)

# Clean language
kbai$language = ifelse(kbai$language %in% c("English", "American English", "ENGLISH",
                  "American", "English (US)", "First", "English Language",
                  "English and French", "English, Cantonese", "Java",
                  "Conative American Sign Language and English"), "English", kbai$language)
kbai$language = ifelse(kbai$language %in% c("Chinese", "Mandarin", "China",
                  "Mandarin Chinese", "Cantonese"), "Chinese", kbai$language)
kbai$language = ifelse(kbai$language %in% c("Principal", "Korean", "South Korean"),
                  "Korean", kbai$language)
kbai$language = ifelse(kbai$language %in% c("Swiss German", "German", "Germany"),
                  "German", kbai$language)
kbai$language = ifelse(kbai$language %in% c("Marathi", "Telugu", "Bengali", "Gujarati",
                  "Kannada", "Hindi", "Tamil"), "Indian", kbai$language)
kbai$language = ifelse(kbai$language %in% c("Thai", "ABAP"), "Thai",
                  kbai$language)
kbai$language = ifelse(kbai$language == "NA", NA, kbai$language)

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

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

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

kbai$conf_pre_ave = (kbai$conf_p2_pre + kbai$conf_p3_pre)/2

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

# Convert ranges to numeric values
kbai$age_num = revalue(kbai$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))
```

```
kbai$age_num = as.numeric(as.character(kbai$age_num))

kbai$prog_num = revalue(kbai$programming, c("0"=0, "1-3"=2, "3-5"=4, "5-10"=7.5,
                  "10-15"=12.5, "15-20"=17.5, "20+"=20))
kbai$prog_num = as.numeric(as.character(kbai$prog_num))

kbai$hours_num = revalue(kbai$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))
kbai$hours_num = as.numeric(as.character(kbai$hours_num))

# Create indicator for withdrawing from course
kbai$w_ind = ifelse(kbai$exam==0, 1, 0)

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

Explore Data

```
# Calculate summary statistics
summary(kbai)
```

```
##    student                 age         gender        birth
##  Length:586         18 to 24: 79   Female: 76   USA    :313
##  Class :character   25 to 34:327   Male  :501   China  : 58
##  Mode  :character   35 to 44:131   NA's  :  9   India  : 58
##                     45 to 54: 34                Canada : 10
##                     55 to 64:  6                Korea  :  8
##                     NA's    :  9                (Other):130
##                                                 NA's   :  9
##      residence      language      english          education
##  USA      :502   English:402   Fluent :196   Bachelors  :413
##  Canada   : 14   Chinese: 58   Native :368   Doctorate  : 44
##  India    : 10   Indian : 31   Partial: 12   High School:  1
##  China    :  5   Spanish: 20   NA's   : 10   Masters    :116
##  Singapore:  5   Korean :  6                 NA's       : 12
##  (Other)  : 40   (Other): 58
##  NA's     : 10   NA's   : 11
##   programming   conf_p1_post    conf_p2_pre     conf_p2_post
##  5-10   :142   Min.   :1.000   Min.   :1.000   Min.   :1.00
##  1-3    :139   1st Qu.:3.000   1st Qu.:3.000   1st Qu.:3.00
##  3-5    :121   Median :4.000   Median :4.000   Median :4.00
##  10-15  : 83   Mean   :3.761   Mean   :3.715   Mean   :3.87
##  15-20  : 39   3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:5.00
##  (Other): 52   Max.   :5.000   Max.   :5.000   Max.   :5.00
##  NA's   : 10   NA's   :88      NA's   :88      NA's   :109
##   conf_p3_pre     conf_p3_post       hours          exam
##  Min.   :1.000   Min.   :1.000   9-12   : 91   Min.   :  0.0
##  1st Qu.:3.000   1st Qu.:3.000   12-15  : 89   1st Qu.: 82.0
##  Median :4.000   Median :4.000   15-18  : 66   Median : 89.0
##  Mean   :3.446   Mean   :3.589   18-21  : 53   Mean   : 84.7
```

```
##  3rd Qu.:4.000   3rd Qu.:4.000   6-9    : 49   3rd Qu.: 94.0
##  Max.   :5.000   Max.   :5.000   (Other): 68   Max.   :100.0
##  NA's   :108     NA's   :170     NA's   :170   NA's   :1
##    assign_ave        proj_ave         total               semester
##  Min.   : 23.33   Min.   : 0.00   Min.   : 8.667   Summer 2015:287
##  1st Qu.: 75.62   1st Qu.:65.00   1st Qu.:75.025   Summer 2016:299
##  Median : 81.67   Median :74.33   Median :80.921
##  Mean   : 79.96   Mean   :71.03   Mean   :78.150
##  3rd Qu.: 87.50   3rd Qu.:81.67   3rd Qu.:84.877
##  Max.   :100.00   Max.   :96.33   Max.   :97.100
##  NA's   :2        NA's   :1       NA's   :2
##     conf_ave      conf_pre_ave   conf_post_ave      age_num
##  Min.   :1.400   Min.   :1.000   Min.   :1.000   Min.   :21.0
##  1st Qu.:3.200   1st Qu.:3.000   1st Qu.:3.333   1st Qu.:29.5
##  Median :3.800   Median :3.500   Median :4.000   Median :29.5
##  Mean   :3.693   Mean   :3.556   Mean   :3.761   Mean   :32.1
##  3rd Qu.:4.200   3rd Qu.:4.000   3rd Qu.:4.333   3rd Qu.:39.5
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :59.5
##  NA's   :221     NA's   :151     NA's   :218     NA's   :9
##     prog_num        hours_num         w_ind          native_ind
##  Min.   : 0.000   Min.   : 1.50   Min.   :0.0000   Min.   :0.0000
##  1st Qu.: 2.000   1st Qu.:10.50   1st Qu.:0.0000   1st Qu.:0.0000
##  Median : 7.500   Median :13.50   Median :0.0000   Median :1.0000
##  Mean   : 7.512   Mean   :13.51   Mean   :0.0359   Mean   :0.6389
##  3rd Qu.:12.500   3rd Qu.:16.50   3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :20.000   Max.   :21.00   Max.   :1.0000   Max.   :1.0000
##  NA's   :10       NA's   :170     NA's   :1        NA's   :10
##    higher_ind      gender_ind
##  Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:1.0000
##  Median :0.000   Median :1.0000
##  Mean   :0.273   Mean   :0.8683
##  3rd Qu.:1.000   3rd Qu.:1.0000
##  Max.   :1.000   Max.   :1.0000
##                  NA's   :9
```

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

```
##
##    Female      Male
## 0.1317158 0.8682842
```

Analyze Data by Gender

```
# Calculate age summary statistics
ddply(subset(kbai, !is.na(age_num)), "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 32.57237 8.191745   29.5    29.5    39.5
## 2   Male 32.02495 7.597820   29.5    29.5    39.5
```

```r
# Calculate overall grade summary statistics
ddply(subset(kbai, !is.na(gender)&!is.na(total)&w_ind==0), "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 80.53322 8.939572 82.11667 76.15833 86.86667
## 2   Male 80.00500 7.715237 81.20833 76.17083 84.92500
```

```r
# Calculate assignment summary statistics
ddply(subset(kbai, !is.na(gender)&!is.na(assign_ave)&w_ind==0), "gender", summarise,
            mean = mean(assign_ave), sd = sd(assign_ave), median = median(assign_ave),
            first_q = quantile(assign_ave, 0.25), third_q = quantile(assign_ave, 0.75))
```

```
##   gender     mean       sd   median  first_q third_q
## 1 Female 83.17778 9.518921 85.00000 77.08333    90.0
## 2   Male 81.05245 9.440888 81.66667 75.83333    87.5
```

```r
# Calculate project summary statistics
ddply(subset(kbai, !is.na(gender)&!is.na(proj_ave)&w_ind==0), "gender", summarise,
            mean = mean(proj_ave), sd = sd(proj_ave), median = median(proj_ave),
            first_q = quantile(proj_ave, 0.25), third_q = quantile(proj_ave, 0.75))
```

```
##   gender     mean       sd   median  first_q  third_q
## 1 Female 72.83111 14.02875 76.00000 64.33333 82.66667
## 2   Male 72.71970 12.79634 74.66667 66.33333 81.66667
```

```r
# Calculate final exam summary statistics
ddply(subset(kbai, !is.na(gender)&!is.na(exam)&w_ind==0), "gender", summarise,
            mean = mean(exam), sd = sd(exam), median = median(exam),
            first_q = quantile(exam, 0.25), third_q = quantile(exam, 0.75))
```

```
##   gender     mean       sd median first_q third_q
## 1 Female 88.62667 10.00375     90    84.5      96
## 2   Male 87.80165  9.27752     89    83.0      94
```

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

```
##   gender     mean       sd median first_q third_q
## 1 Female 5.407895 4.947532    4.0       2     7.5
## 2   Male 7.832000 5.708937    7.5       4    12.5
```

```r
# Calculate study hours summary statistics
ddply(subset(kbai, !is.na(gender)&!is.na(hours_num)&w_ind==0), "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 14.43443 5.058389   13.5    10.5    19.5
## 2   Male 13.38462 4.784522   13.5    10.5    16.5
```

```r
# Calculate confidence summary statistics
ddply(subset(kbai, !is.na(gender)&!is.na(conf_ave)&w_ind==0), "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 3.677966 0.7372053    3.8     3.2     4.2
## 2   Male 3.704290 0.6945016    3.8     3.2     4.2
```

```
# Calculate confidence summary statistics
ddply(subset(kbai, !is.na(gender)&!is.na(conf_pre_ave)&w_ind==0), "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 3.500000 0.8683135    3.5       3       4
## 2   Male 3.572603 0.8585799    4.0       3       4
```

```
ddply(subset(kbai, !is.na(gender)&!is.na(conf_post_ave)&w_ind==0), "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 3.751412 0.8522974      4 3.166667 4.333333
## 2   Male 3.776688 0.8001966      4 3.333333 4.333333
```

```
kbai_m = subset(kbai, gender == "Male")
kbai_f = subset(kbai, gender == "Female")

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

```
##
##    18 to 24    25 to 34    35 to 44    45 to 54    55 to 64
## 0.139720559 0.564870259 0.227544910 0.059880240 0.007984032
```

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

```
##
##   18 to 24   25 to 34   35 to 44   45 to 54   55 to 64
## 0.11842105 0.57894737 0.22368421 0.05263158 0.02631579
```

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

```
##
##        Afghanistan          Argentina          Australia
##        0.001996008        0.000000000        0.007984032
##            Bahamas             Brazil           Bulgaria
##        0.003992016        0.011976048        0.003992016
##             Canada              Chile              China
##        0.019960080        0.001996008        0.081836327
##           Colombia               Cuba     Czech Republic
##        0.001996008        0.001996008        0.001996008
##           Dominica Dominican Republic            Ecuador
##        0.001996008        0.001996008        0.001996008
##        El Salvador           Ethiopia            Germany
##        0.001996008        0.003992016        0.007984032
##          Guatemala              Haiti              India
##        0.001996008        0.001996008        0.095808383
##          Indonesia               Iran              Italy
##        0.001996008        0.005988024        0.003992016
```

```
##             Japan        Kazakhstan             Kenya
##       0.005988024       0.001996008       0.005988024
##             Korea            Kuwait           Lebanon
##       0.013972056       0.001996008       0.001996008
##            Mexico           Moldova           Myanmar
##       0.011976048       0.000000000       0.001996008
##             Nepal       New Zealand           Nigeria
##       0.005988024       0.001996008       0.003992016
##            Norway          Pakistan            Panama
##       0.003992016       0.011976048       0.005988024
##              Peru       Philippines            Poland
##       0.003992016       0.001996008       0.001996008
##       Puerto Rico            Russia            Serbia
##       0.001996008       0.007984032       0.001996008
##         Singapore      South Africa         Sri Lanka
##       0.003992016       0.001996008       0.001996008
##       Switzerland             Syria            Taiwan
##       0.001996008       0.001996008       0.013972056
##          Thailand           Tunisia            Turkey
##       0.003992016       0.001996008       0.007984032
##               UAE                UK           Ukraine
##       0.001996008       0.001996008       0.005988024
##               USA           Vietnam
##       0.566866267       0.011976048
```

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

```
##
##       Afghanistan         Argentina         Australia
##        0.00000000         0.01315789         0.01315789
##           Bahamas            Brazil           Bulgaria
##        0.00000000         0.00000000         0.00000000
##            Canada             Chile            China
##        0.00000000         0.00000000         0.22368421
##          Colombia               Cuba     Czech Republic
##        0.01315789         0.02631579         0.00000000
##          Dominica Dominican Republic            Ecuador
##        0.00000000         0.00000000         0.02631579
##       El Salvador           Ethiopia            Germany
##        0.00000000         0.00000000         0.00000000
##         Guatemala              Haiti              India
##        0.00000000         0.00000000         0.13157895
##         Indonesia               Iran              Italy
##        0.00000000         0.00000000         0.01315789
##             Japan         Kazakhstan              Kenya
##        0.00000000         0.00000000         0.02631579
##             Korea             Kuwait            Lebanon
##        0.01315789         0.00000000         0.00000000
##            Mexico            Moldova            Myanmar
##        0.00000000         0.01315789         0.00000000
##             Nepal        New Zealand            Nigeria
##        0.01315789         0.00000000         0.00000000
##            Norway           Pakistan             Panama
##        0.00000000         0.00000000         0.00000000
##              Peru        Philippines             Poland
```

```
##         0.00000000         0.02631579         0.00000000
##        Puerto Rico             Russia             Serbia
##         0.00000000         0.00000000         0.01315789
##          Singapore       South Africa          Sri Lanka
##         0.01315789         0.00000000         0.00000000
##        Switzerland              Syria             Taiwan
##         0.00000000         0.00000000         0.01315789
##           Thailand            Tunisia             Turkey
##         0.00000000         0.00000000         0.00000000
##                UAE                 UK            Ukraine
##         0.00000000         0.00000000         0.01315789
##                USA            Vietnam
##         0.38157895         0.01315789
```

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

```
##
##   Australia     Bahamas      Brazil      Canada       Chile       China
##       0.006       0.002       0.002       0.028       0.002       0.010
##    Colombia El Salvador     Germany       India   Indonesia     Ireland
##       0.002       0.002       0.004       0.018       0.002       0.004
##      Israel       Italy       Japan       Kenya     Myanmar Netherlands
##       0.000       0.000       0.002       0.002       0.002       0.004
## New Zealand    Pakistan      Panama        Peru   Singapore South Korea
##       0.002       0.004       0.002       0.002       0.008       0.006
##      Sweden Switzerland      Taiwan     Tunisia         UAE          UK
##       0.002       0.002       0.002       0.002       0.002       0.002
##     Ukraine         USA     Vietnam
##       0.002       0.868       0.002
```

```
prop.table(table(kbai_f$residence))
```

```
##
##   Australia     Bahamas      Brazil      Canada       Chile       China
##  0.01315789  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000
##    Colombia El Salvador     Germany       India   Indonesia     Ireland
##  0.00000000  0.00000000  0.00000000  0.01315789  0.00000000  0.00000000
##      Israel       Italy       Japan       Kenya     Myanmar Netherlands
##  0.01315789  0.01315789  0.01315789  0.02631579  0.00000000  0.00000000
## New Zealand    Pakistan      Panama        Peru   Singapore South Korea
##  0.00000000  0.00000000  0.00000000  0.00000000  0.01315789  0.00000000
##      Sweden Switzerland      Taiwan     Tunisia         UAE          UK
##  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000
##     Ukraine         USA     Vietnam
##  0.00000000  0.89473684  0.00000000
```

```
# Compare language background
prop.table(table(kbai_m$language))
```

```
##
##         Arabic      Bulgarian        Burmese        Chinese          Czech
##    0.006012024    0.002004008    0.002004008    0.086172345    0.002004008
##        English          Farsi       Filipino         French         German
##    0.713426854    0.006012024    0.000000000    0.002004008    0.006012024
## Haitian Creole         Indian     Indonesian        Italian       Japanese
```

```
##    0.002004008    0.054108216    0.002004008    0.000000000    0.002004008
##         Korean      Malayalam         Nepali      Norwegian        Persian
##    0.010020040    0.004008016    0.004008016    0.004008016    0.002004008
##         Polish     Portuguese        Russian        Serbian        Spanish
##    0.002004008    0.012024048    0.010020040    0.002004008    0.030060120
##        Swahili        Tagalog           Thai        Turkish      Ukrainian
##    0.002004008    0.000000000    0.004008016    0.008016032    0.002004008
##          Urdu     Vietnamese
##    0.008016032    0.010020040
```

```
prop.table(table(kbai_f$language))
```

```
##
##         Arabic      Bulgarian        Burmese        Chinese         Czech
##     0.00000000     0.00000000     0.00000000     0.19736842     0.00000000
##        English          Farsi       Filipino         French         German
##     0.60526316     0.00000000     0.01315789     0.00000000     0.00000000
## Haitian Creole         Indian     Indonesian        Italian       Japanese
##     0.00000000     0.05263158     0.00000000     0.01315789     0.00000000
##         Korean      Malayalam         Nepali      Norwegian        Persian
##     0.01315789     0.01315789     0.00000000     0.00000000     0.00000000
##         Polish     Portuguese        Russian        Serbian        Spanish
##     0.00000000     0.00000000     0.00000000     0.00000000     0.06578947
##        Swahili        Tagalog           Thai        Turkish      Ukrainian
##     0.00000000     0.01315789     0.00000000     0.00000000     0.00000000
##          Urdu     Vietnamese
##     0.00000000     0.01315789
```

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

```
##
##  Fluent  Native Partial
##   0.316   0.666   0.018
```

```
prop.table(table(kbai_f$english))
```

```
##
##     Fluent     Native    Partial
## 0.50000000 0.46052632 0.03947368
```

```
# Compare education
prop.table(table(kbai_m$education))
```

```
##
##   Bachelors   Doctorate High School     Masters
## 0.738955823 0.060240964 0.002008032 0.198795181
```

```
prop.table(table(kbai_f$education))
```

```
##
##   Bachelors   Doctorate High School     Masters
##   0.5921053   0.1842105   0.0000000   0.2236842
```

```
# Compare programming skills
prop.table(table(kbai_m$programming))
```

```
##
##     0   1-3   3-5  5-10 10-15 15-20   20+
```

```
## 0.026 0.214 0.196 0.266 0.154 0.072 0.072
```

```
prop.table(table(kbai_f$programming))
```

```
##
##          0        1-3        3-5       5-10      10-15      15-20
## 0.00000000 0.42105263 0.30263158 0.11842105 0.07894737 0.03947368
##        20+
## 0.03947368
```

```
# Compare hours
prop.table(table(kbai_m$hours))
```

```
##
##         0-3         3-6         6-9        9-12       12-15       15-18
## 0.008498584 0.056657224 0.124645892 0.218130312 0.220963173 0.164305949
##       18-21         21+
## 0.118980170 0.087818697
```

```
prop.table(table(kbai_f$hours))
```

```
##
##        0-3        3-6        6-9       9-12      12-15      15-18
## 0.00000000 0.06557377 0.06557377 0.22950820 0.18032787 0.13114754
##      18-21        21+
## 0.16393443 0.16393443
```

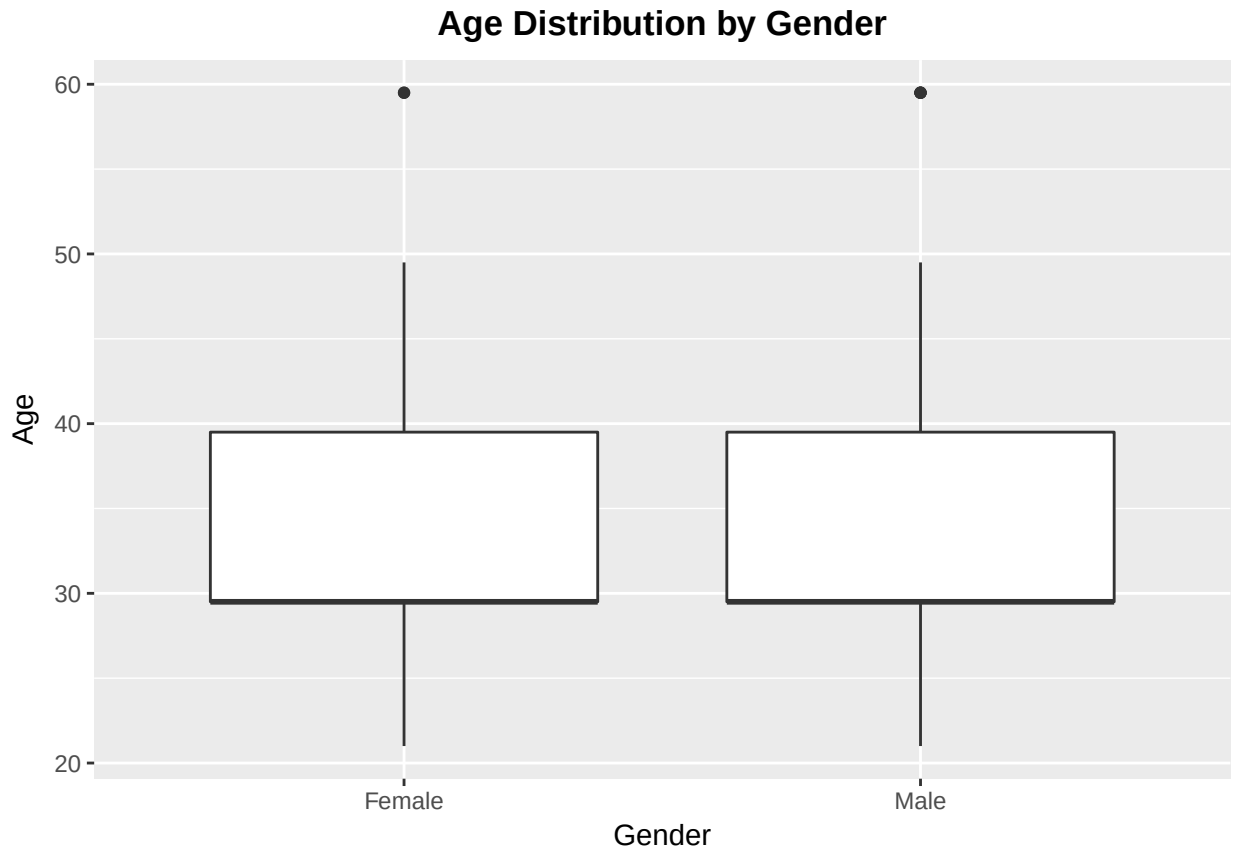
```
# Compare drop-out rates
prop.table(table(kbai_m$w_ind))
```

```
##
##          0          1
## 0.96606786 0.03393214
```

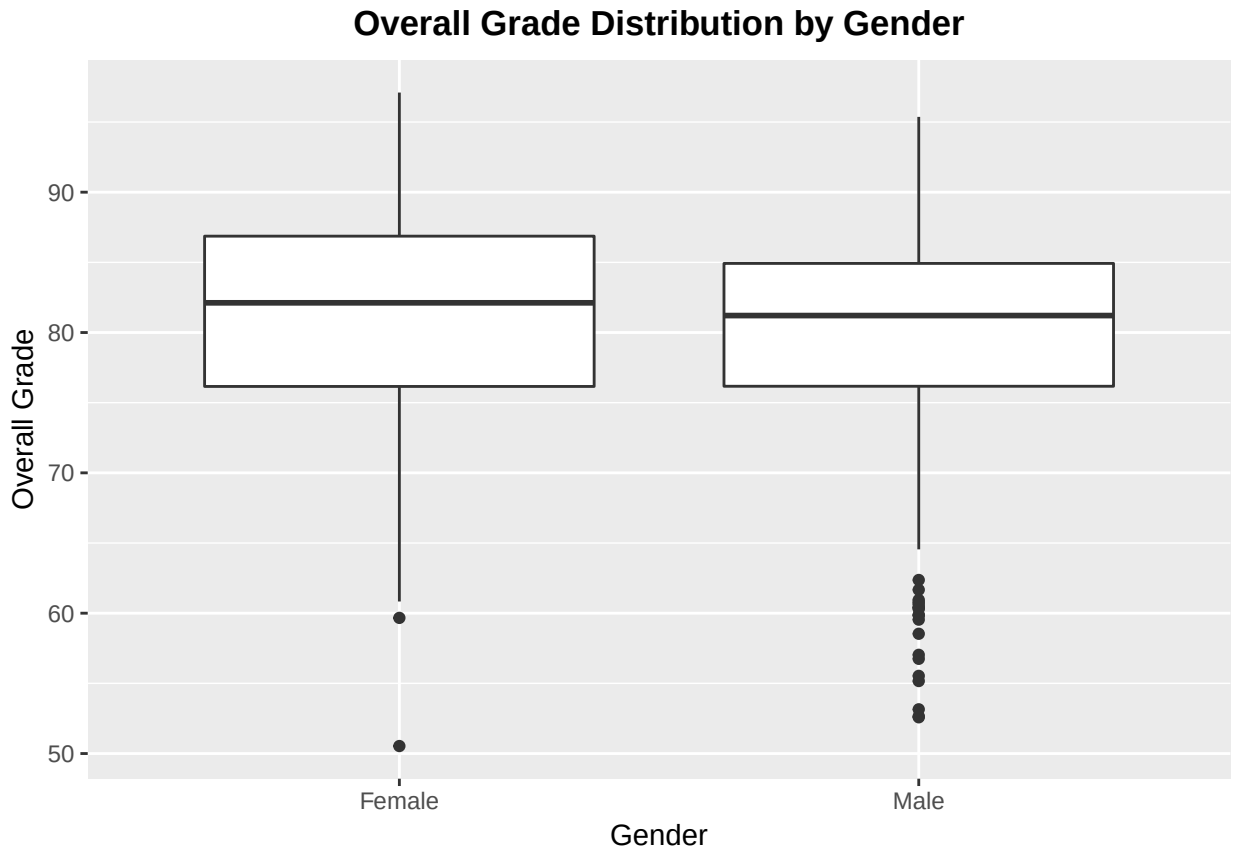
```
prop.table(table(kbai_f$w_ind))
```

```
##
##          0          1
## 0.98684211 0.01315789
```

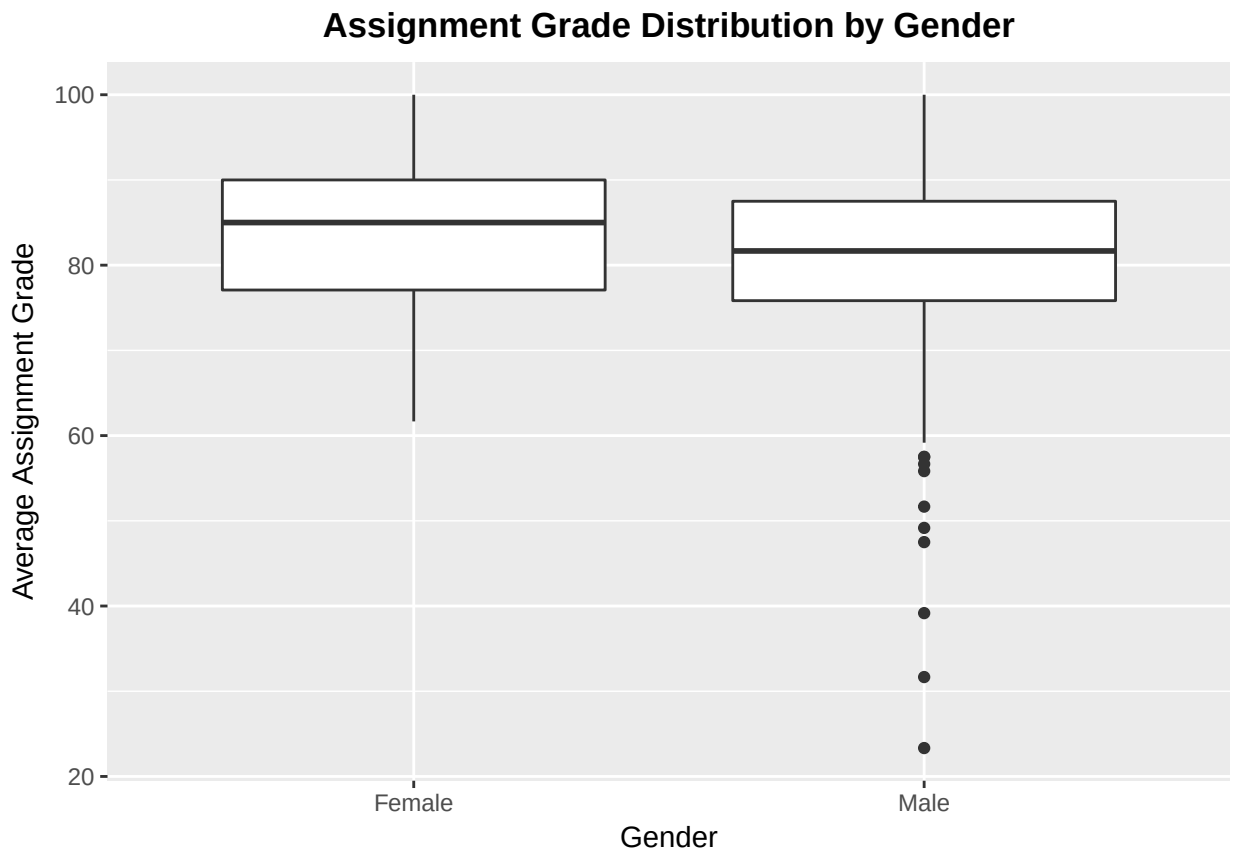
```
#Boxplot of age distribution by gender
ggplot(subset(kbai, !is.na(gender)), aes(gender, age_num)) +
 geom_boxplot() +
 labs(title = "Age Distribution by Gender",
      x = "Gender", y = "Age") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



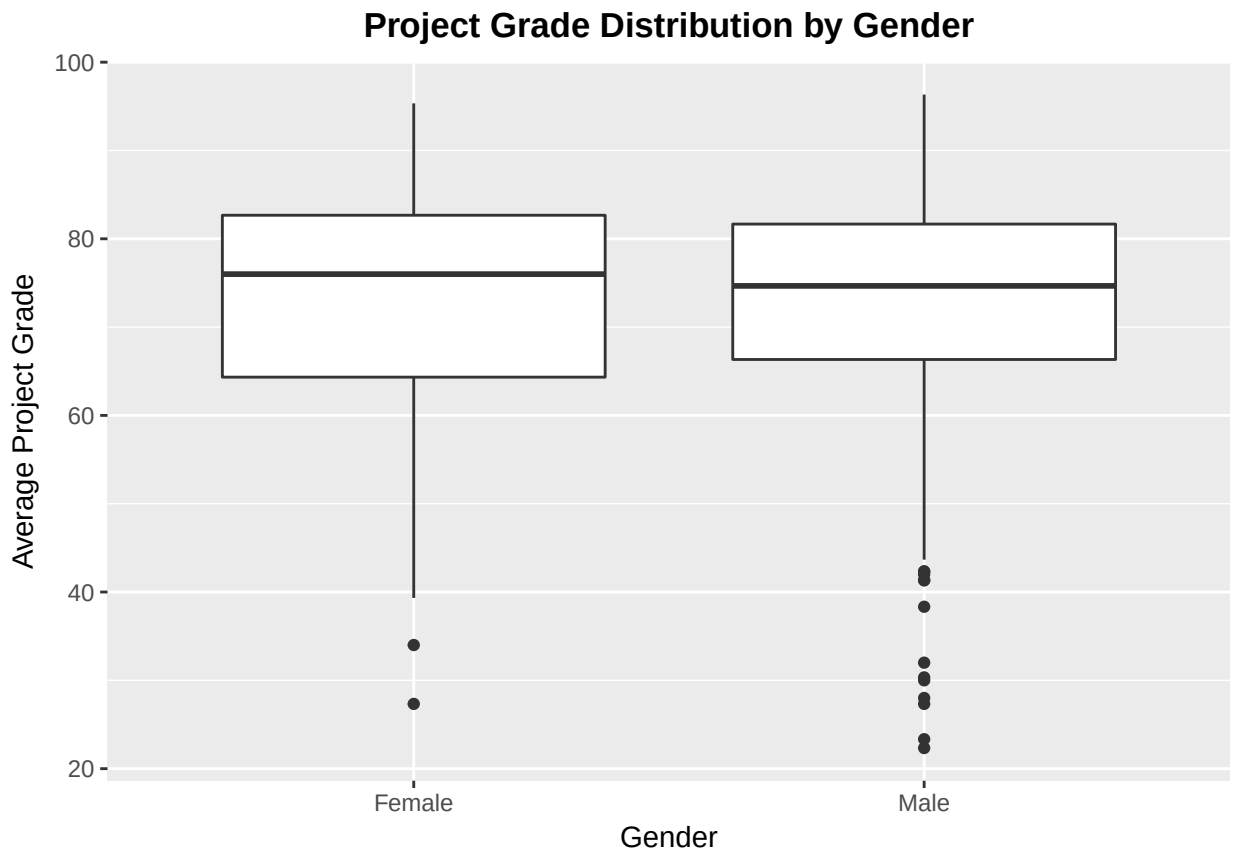

```
# Boxplot of overall grade distribution by gender
ggplot(subset(kbai, !is.na(total) & !is.na(gender) & w_ind==0), aes(gender, total)) +
 geom_boxplot() +
 labs(title = "Overall Grade Distribution by Gender",
      x = "Gender", y = "Overall Grade") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



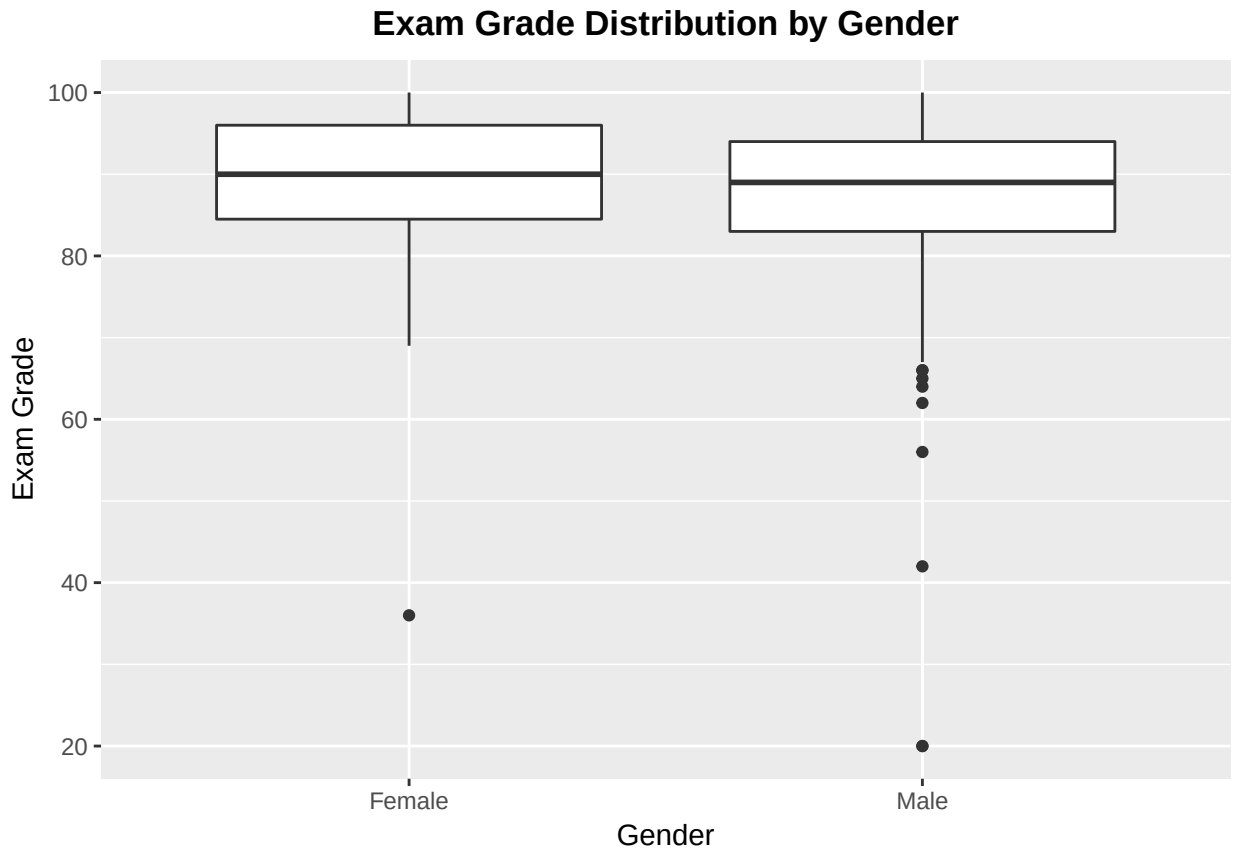

```
# Boxplot of assignment grade distribution by gender
ggplot(subset(kbai, !is.na(assign_ave) & !is.na(gender) & w_ind==0),
       aes(gender, assign_ave)) +
 geom_boxplot() +
 labs(title = "Assignment Grade Distribution by Gender",
      x = "Gender", y = "Average Assignment Grade") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



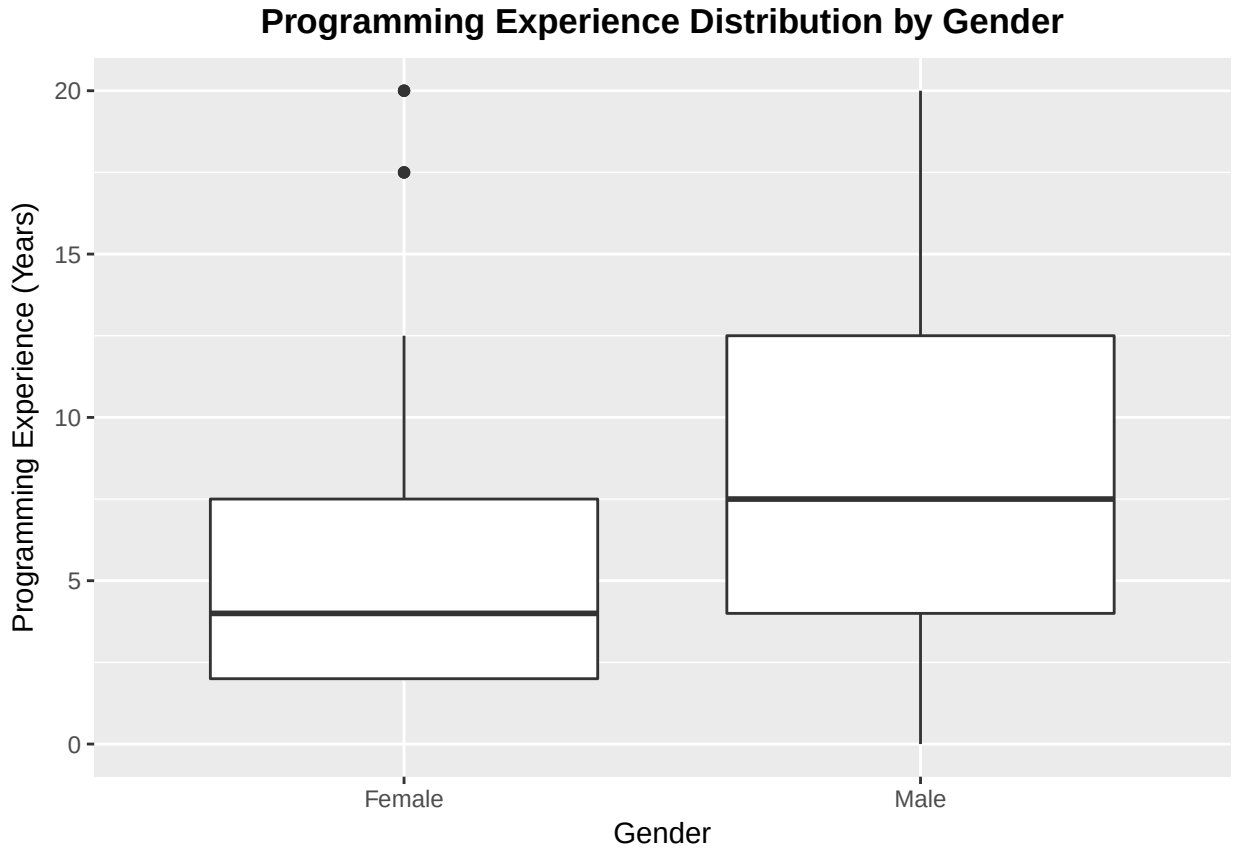

```
# Boxplot of project grade distribution by gender
ggplot(subset(kbai, !is.na(proj_ave) & !is.na(gender) & w_ind==0), aes(gender, proj_ave)) +
 geom_boxplot() +
 labs(title = "Project Grade Distribution by Gender",
      x = "Gender", y = "Average Project Grade") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



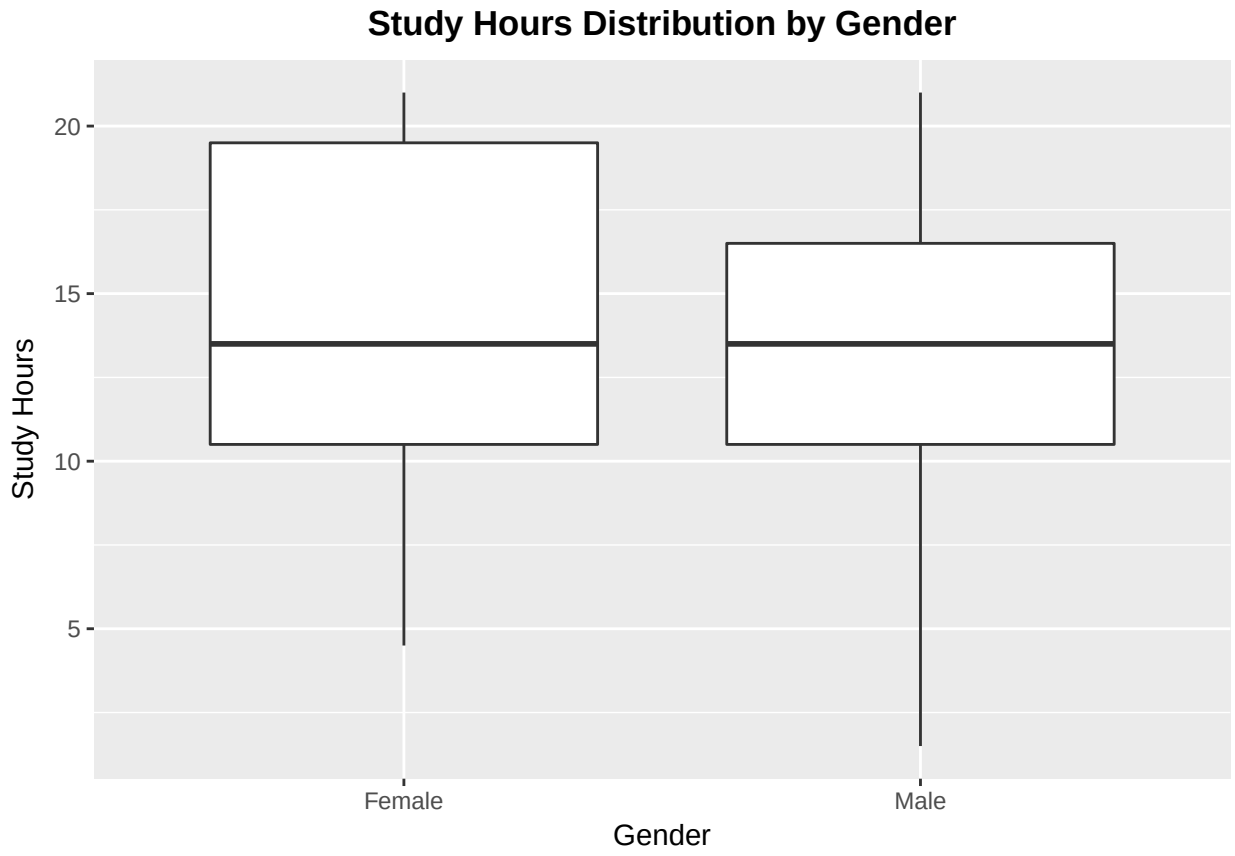

```
# Boxplot of exam grade distribution by gender
ggplot(subset(kbai, !is.na(exam) & !is.na(gender) & w_ind==0), aes(gender, exam)) +
 geom_boxplot() +
 labs(title = "Exam Grade Distribution by Gender",
      x = "Gender", y = "Exam Grade") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```




```
# Boxplot of programming experience by gender
ggplot(subset(kbai, !is.na(prog_num) & !is.na(gender) & w_ind==0), aes(gender, prog_num)) +
 geom_boxplot() +
 labs(title = "Programming Experience Distribution by Gender",
      x = "Gender", y = "Programming Experience (Years)") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```

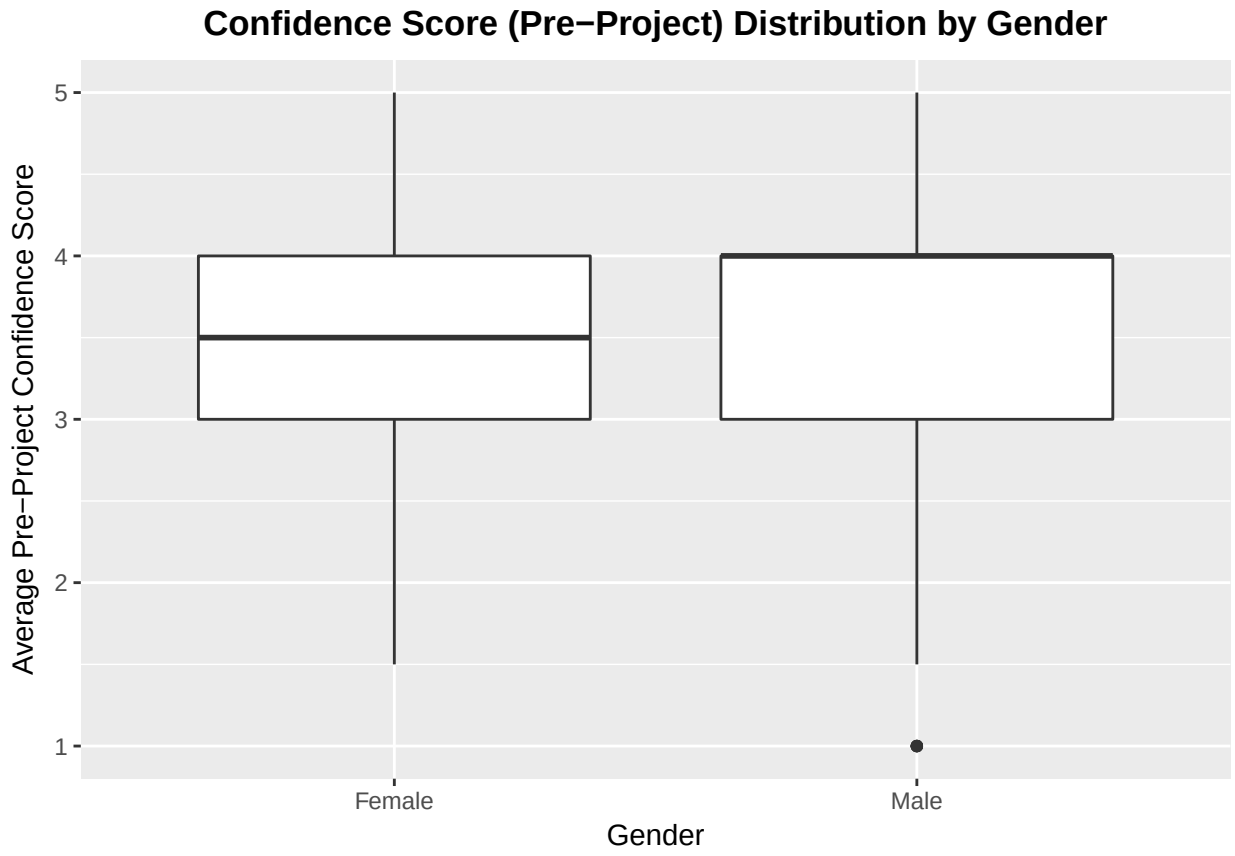



```
# Boxplot of hours spent studying by gender
ggplot(subset(kbai, !is.na(hours_num) & !is.na(gender) & w_ind==0), aes(gender, hours_num)) +
  geom_boxplot() +
 labs(title = "Study Hours Distribution by Gender",
      x = "Gender", y = "Study Hours") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```

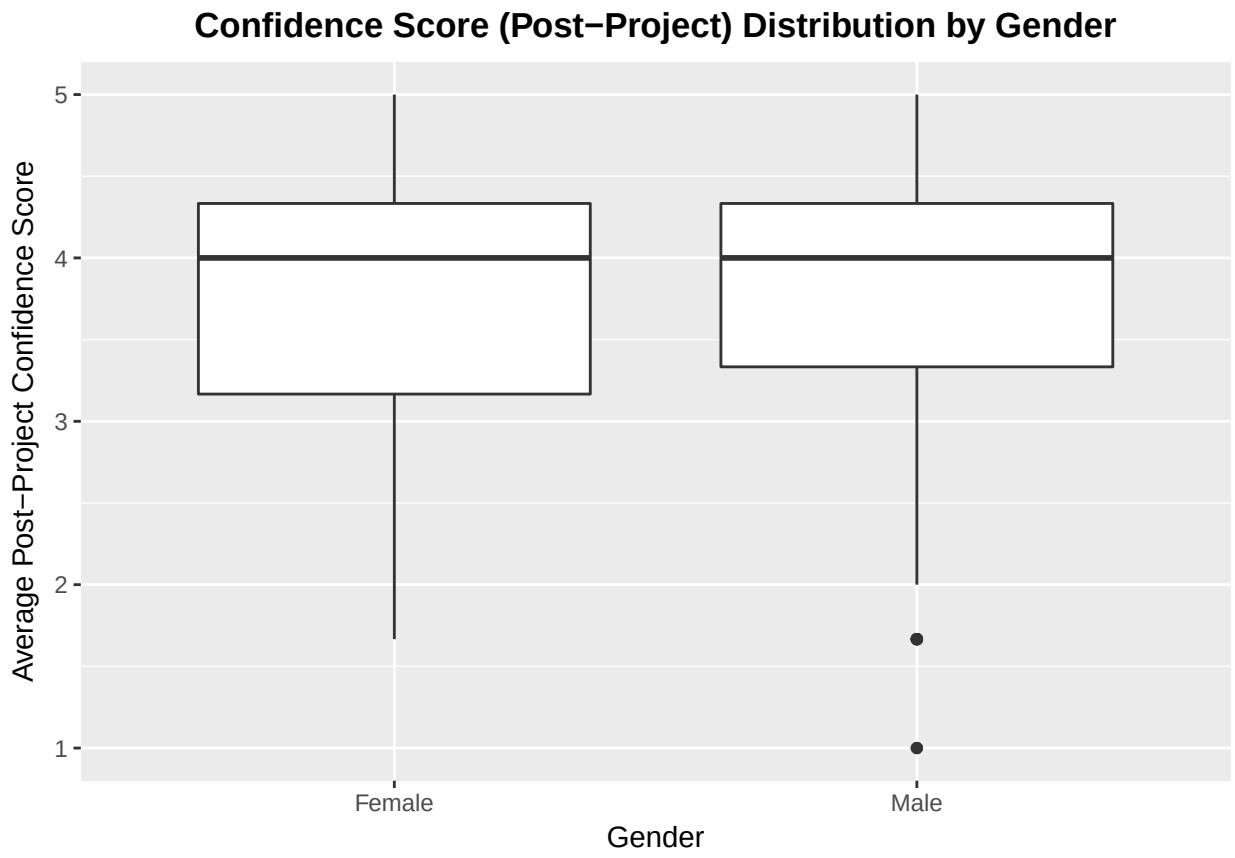



```
# Boxplot of confidence score by gender
ggplot(subset(kbai, !is.na(conf_ave) & !is.na(gender) & w_ind==0), aes(gender, conf_ave)) +
  geom_boxplot() +
 labs(title = "Confidence Score Distribution by Gender",
      x = "Gender", y = "Average Confidence Score") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```

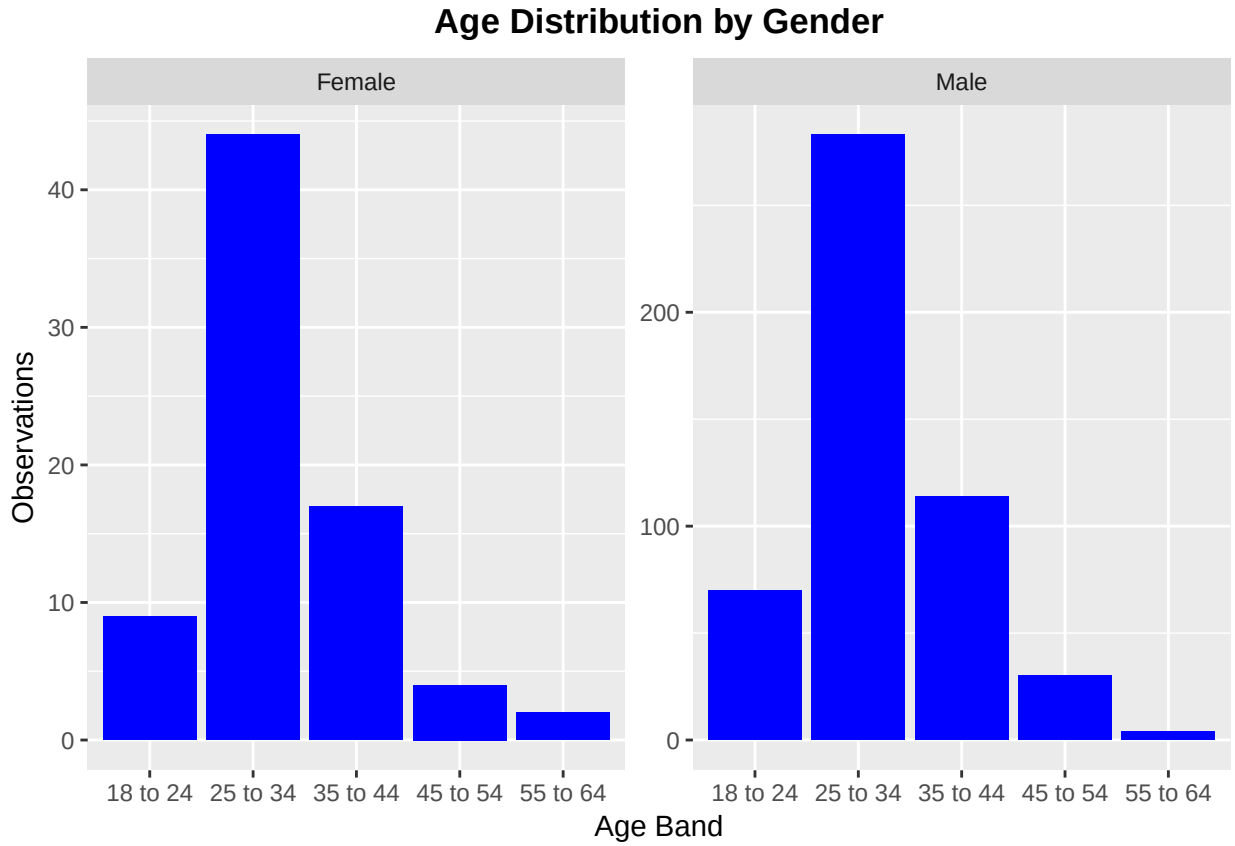

```
# Boxplot of confidence score (pre-project) by gender
ggplot(subset(kbai, !is.na(conf_pre_ave) & !is.na(gender) & w_ind==0), aes(gender,
    conf_pre_ave)) +  geom_boxplot() +
 labs(title = "Confidence Score (Pre-Project) Distribution by Gender",
      x = "Gender", y = "Average Pre-Project Confidence Score") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



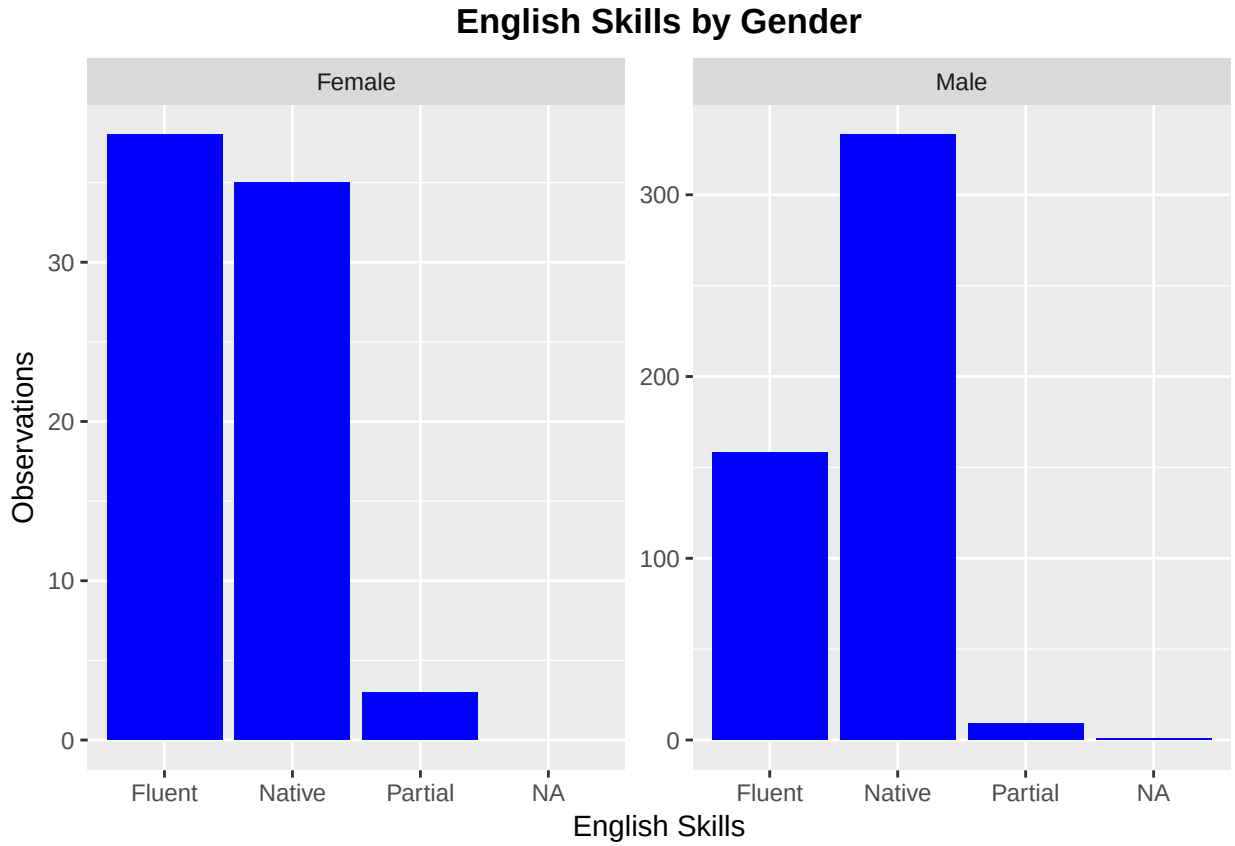

```
# Boxplot of confidence score (post-project) by gender
ggplot(subset(kbai, !is.na(conf_post_ave) & !is.na(gender) & w_ind==0), aes(gender,
    conf_post_ave)) +  geom_boxplot() +
 labs(title = "Confidence Score (Post-Project) Distribution by Gender",
      x = "Gender", y = "Average Post-Project Confidence Score") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



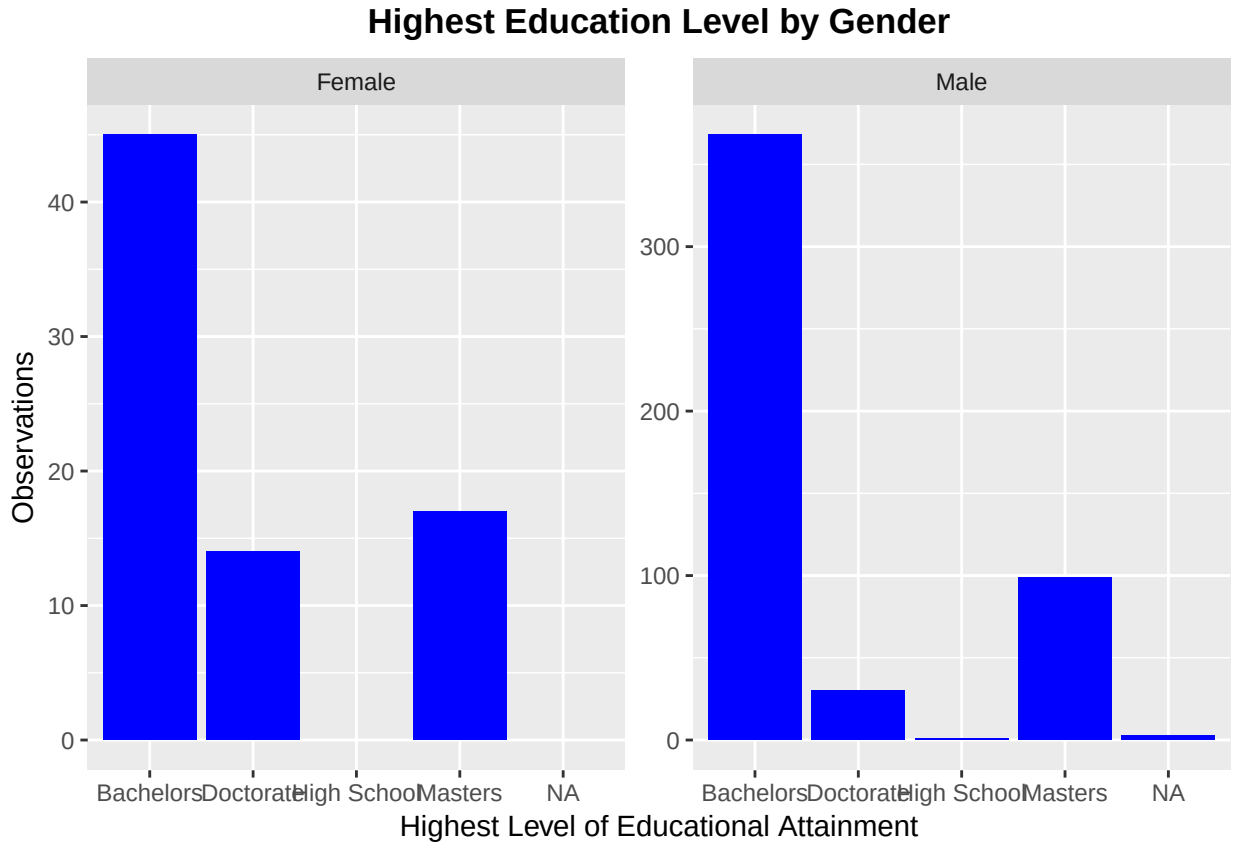

```
# Bar chart comparing age by gender
ggplot(subset(kbai, !is.na(gender)), aes(x = age)) +
    geom_bar(fill = "blue") +
    facet_wrap(~gender, scales = "free_y") +
    labs(title = "Age Distribution by Gender",
      x = "Age Band",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8,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



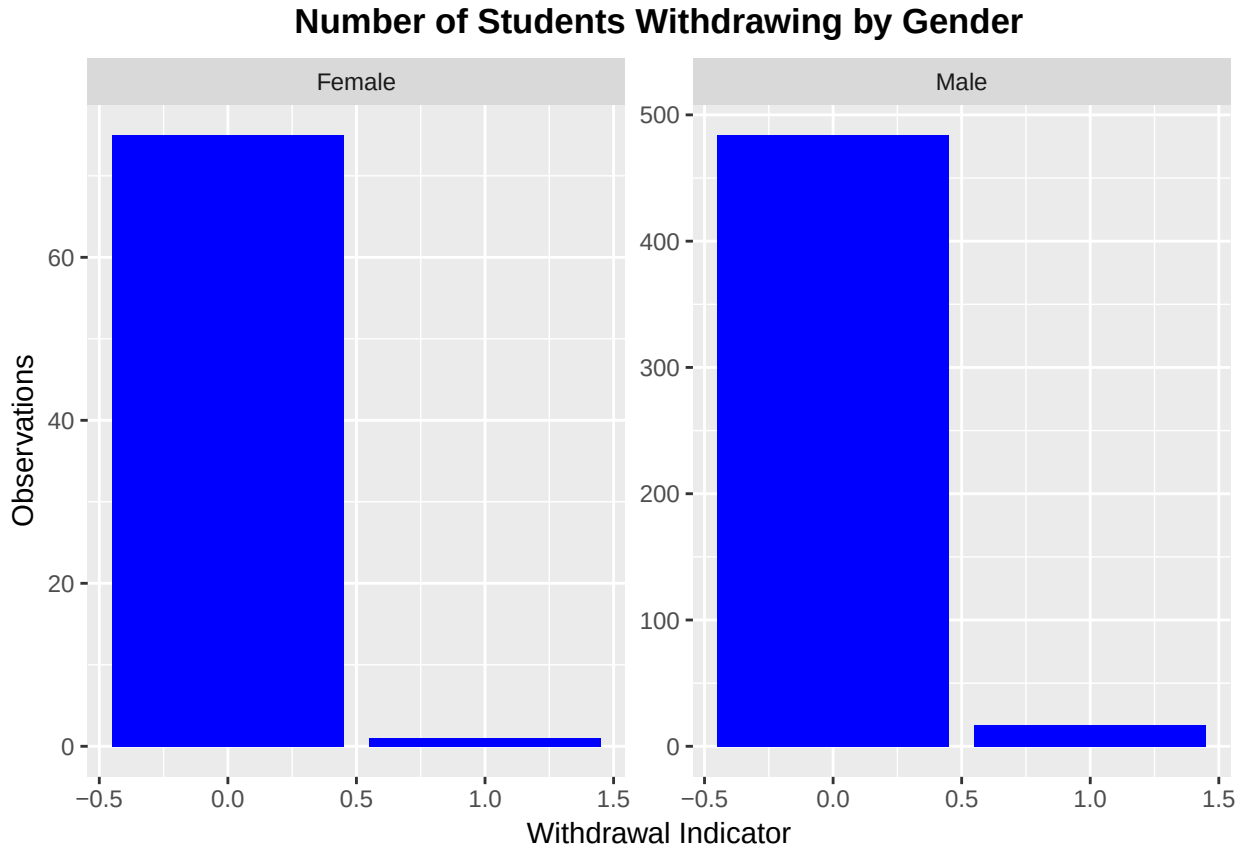

```
# Bar chart comparing English skills by gender
ggplot(subset(kbai, !is.na(gender)), aes(x = english)) +
    geom_bar(fill = "blue") +
    facet_wrap(~gender, scales = "free_y") +
    labs(title = "English Skills by Gender",
      x = "English Skills",
      y = "Observations") +
    theme(plot.title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




```
# Bar chart comparing education by gender
ggplot(subset(kbai, !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))
```

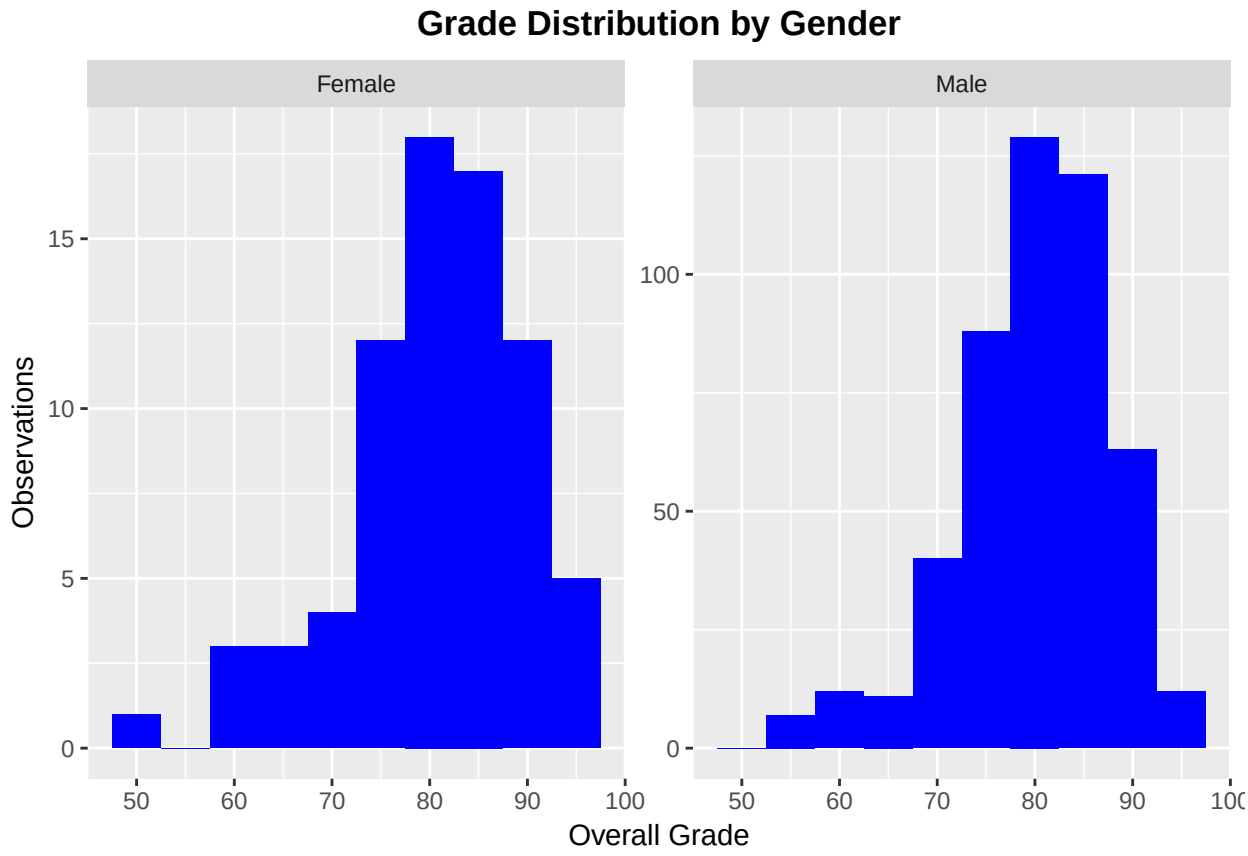



```
# Bar chart comparing w_ind by gender
ggplot(subset(kbai, !is.na(gender)), aes(x = w_ind)) +
    geom_bar(fill = "blue") +
    facet_wrap(~gender, scales = "free_y") +
    labs(title = "Number of Students Withdrawing by Gender",
      x = "Withdrawal Indicator",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



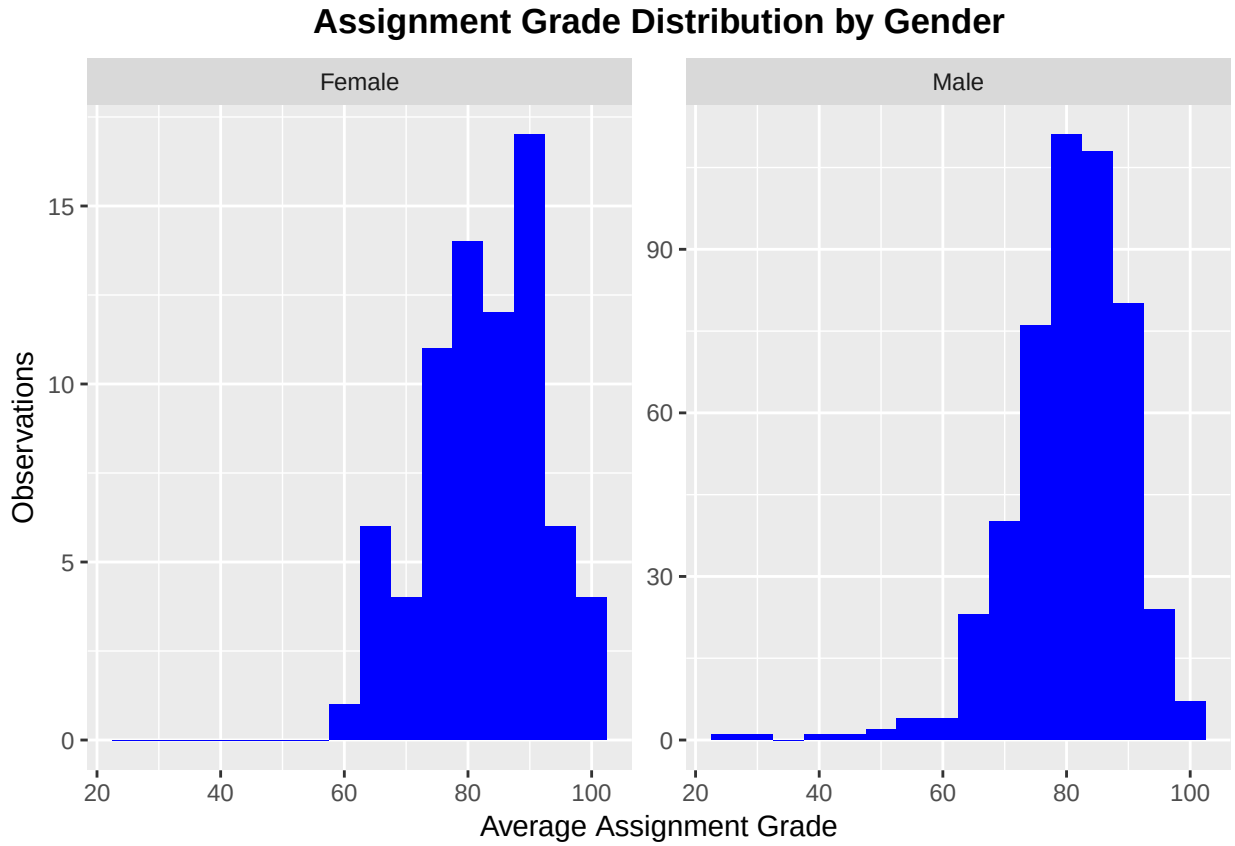

```
# Histogram of grades by gender
ggplot(subset(kbai, !is.na(gender) & w_ind==0), aes(x = total)) +
    geom_histogram(fill = "blue", binwidth = 5) +
    facet_wrap(~gender, scales = "free_y") +
    labs(title = "Grade Distribution by Gender",
      x = "Overall Grade",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```

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

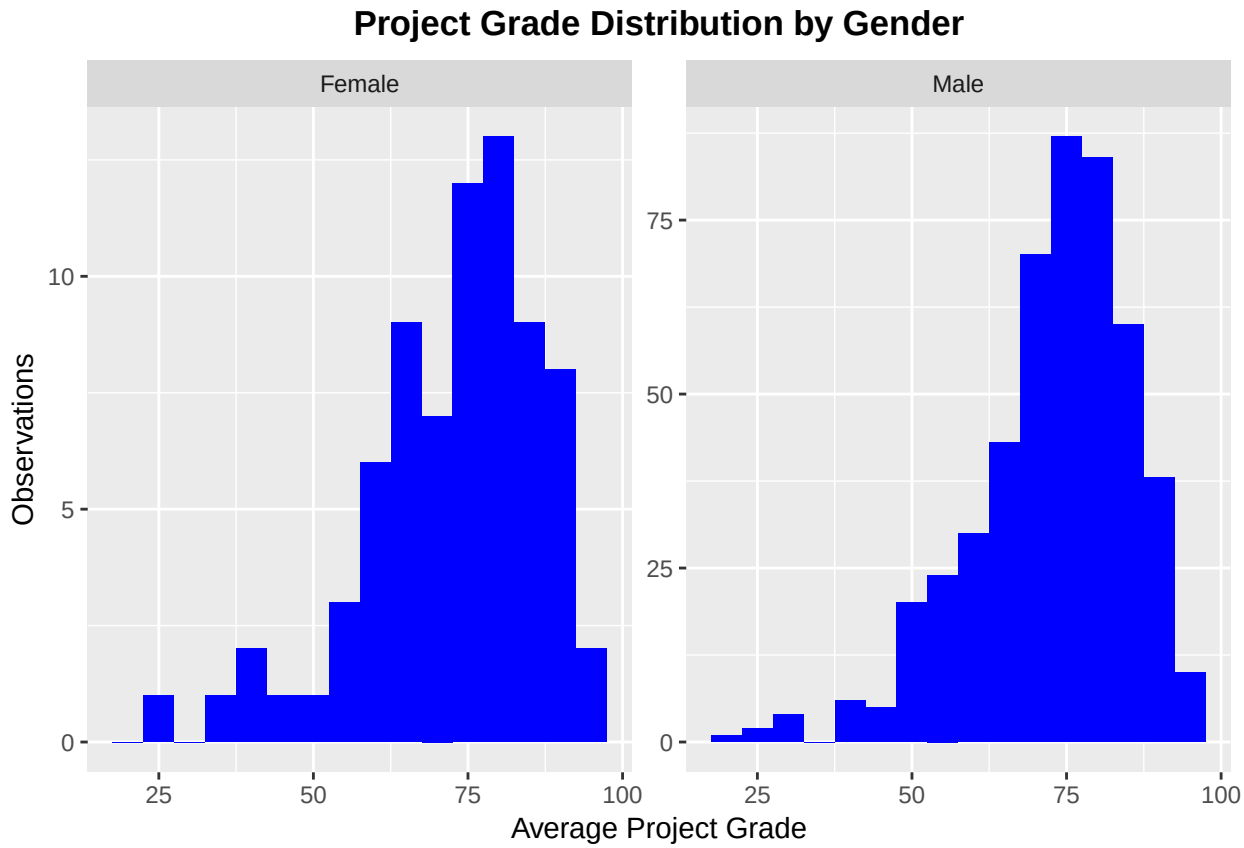



```
# Histogram of average assignment grade by gender
ggplot(subset(kbai, !is.na(gender) & w_ind==0), aes(x = assign_ave)) +
    geom_histogram(fill = "blue", binwidth = 5) +
    facet_wrap(~gender, scales = "free_y") +
    labs(title = "Assignment Grade Distribution by Gender",
      x = "Average Assignment Grade",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```

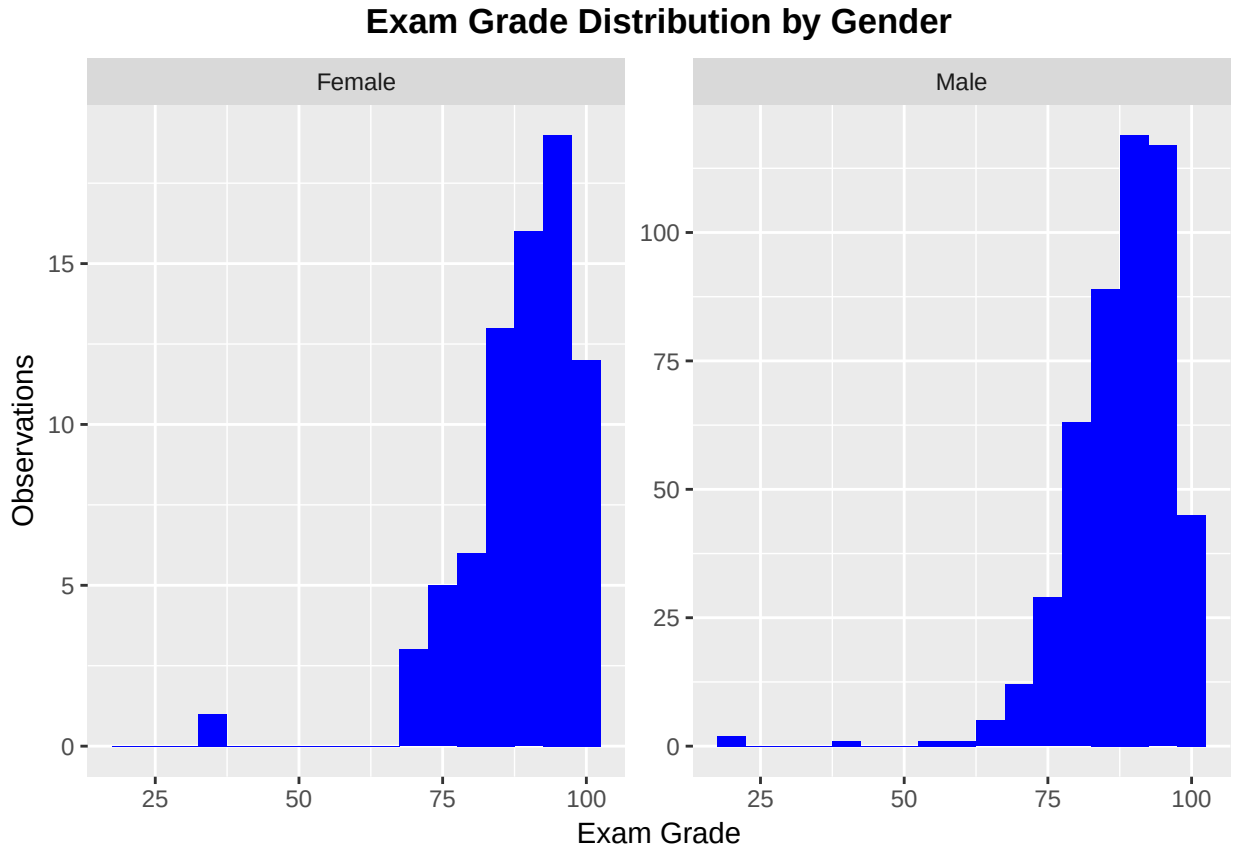
```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```




```
# Histogram of average project grade by gender
ggplot(subset(kbai, !is.na(gender) & w_ind==0), aes(x = proj_ave)) +
    geom_histogram(fill = "blue", binwidth = 5) +
    facet_wrap(~gender, scales = "free_y") +
    labs(title = "Project Grade Distribution by Gender",
      x = "Average Project Grade",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```

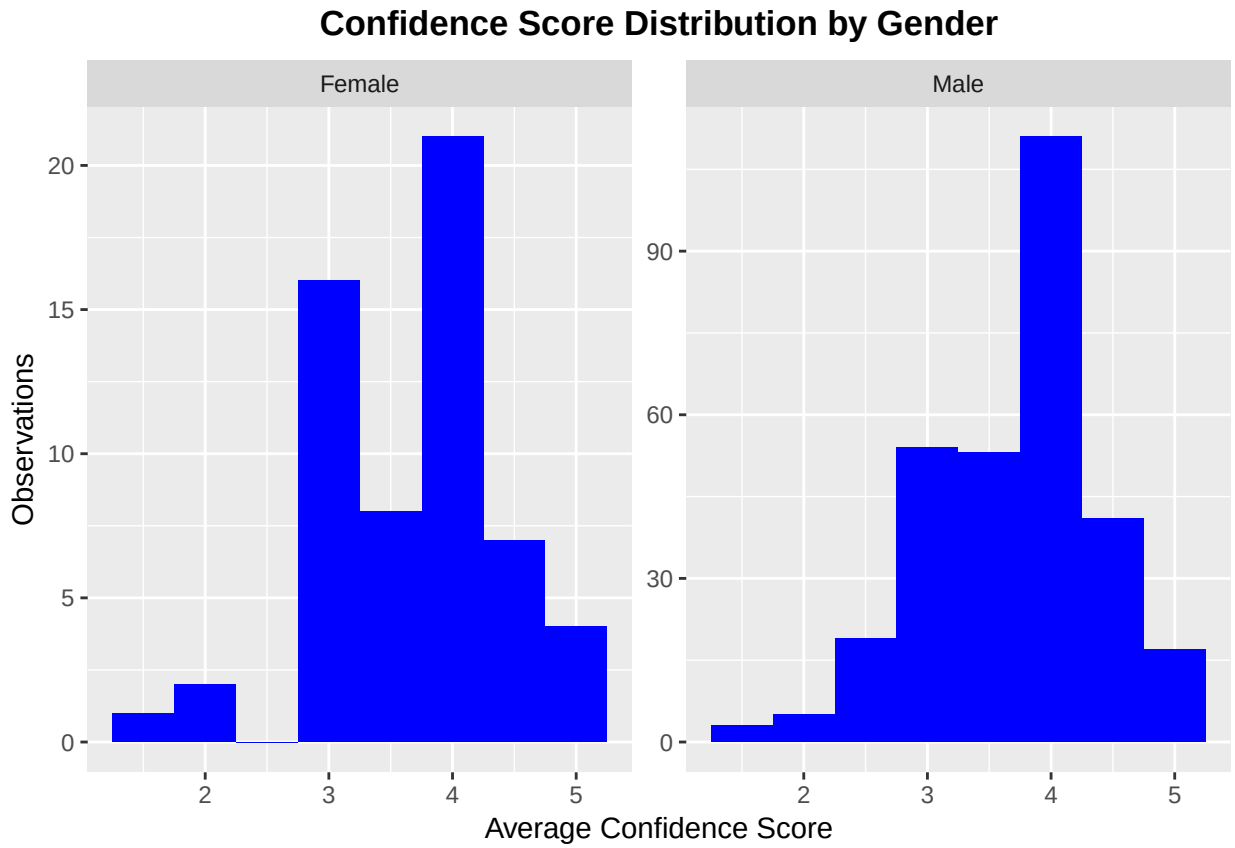



```
# Histogram of exam grade by gender
ggplot(subset(kbai, !is.na(gender) & w_ind==0), aes(x = exam)) +
    geom_histogram(fill = "blue", binwidth = 5) +
    facet_wrap(~gender, scales = "free_y") +
    labs(title = "Exam Grade Distribution by Gender",
      x = "Exam Grade",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



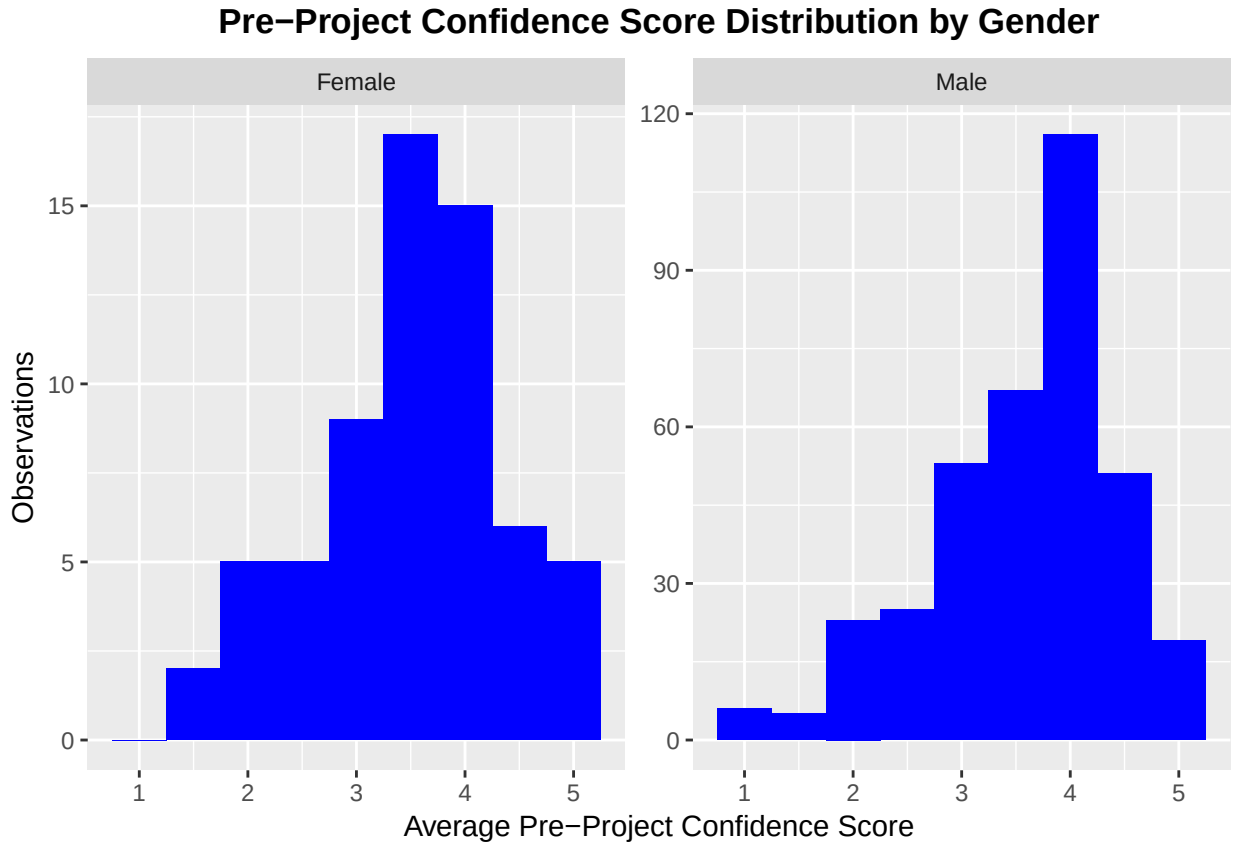

```
# Histogram of conf_ave by gender
ggplot(subset(kbai, !is.na(gender) & w_ind==0), aes(x = conf_ave)) +
    geom_histogram(fill = "blue", binwidth = 0.5) +
    facet_wrap(~gender, scale = "free_y") +
    labs(title = "Confidence Score Distribution by Gender",
      x = "Average Confidence Score",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```

```
## Warning: Removed 197 rows containing non-finite values (stat_bin).
```



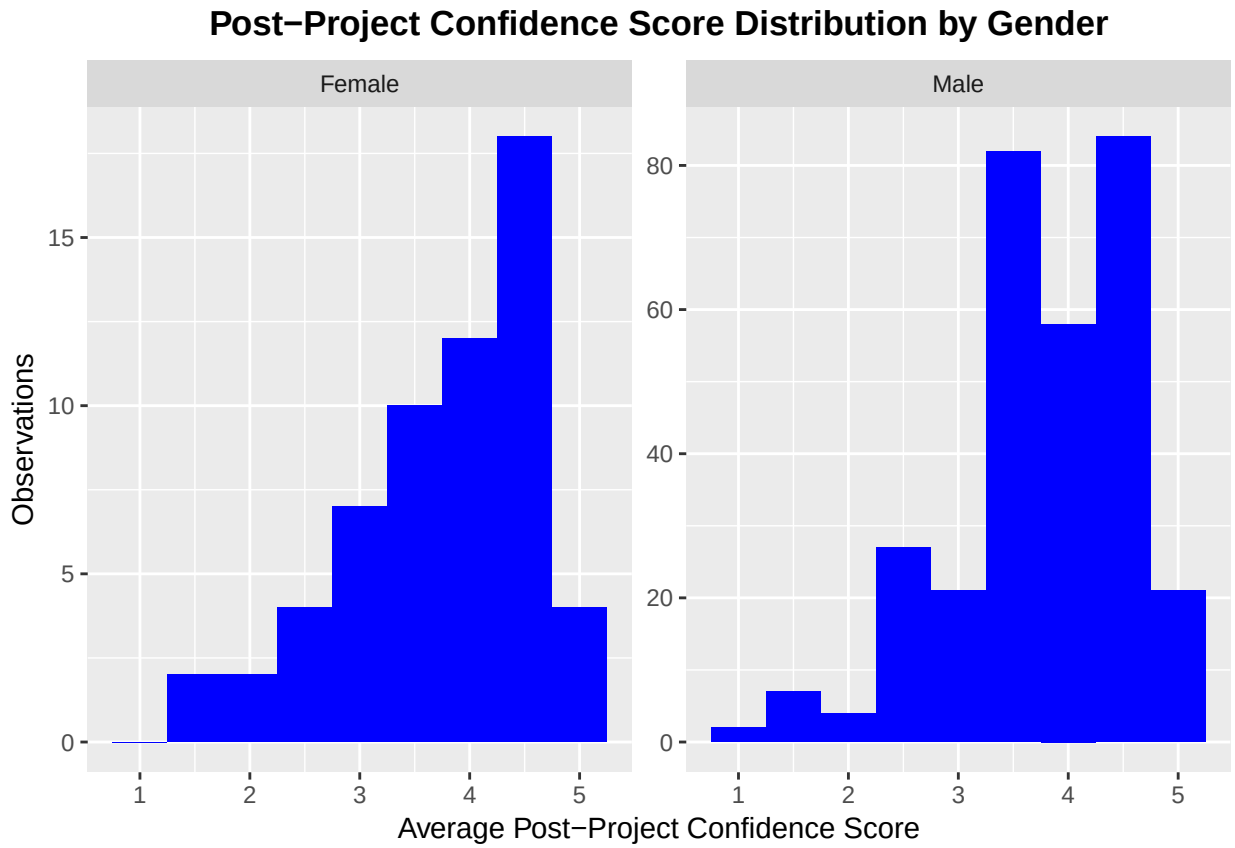

```
# Histogram of conf_pre_ave by gender
ggplot(subset(kbai, !is.na(gender) & w_ind==0), aes(x = conf_pre_ave)) +
    geom_histogram(fill = "blue", binwidth = 0.5) +
    facet_wrap(~gender, scale = "free_y") +
    labs(title = "Pre-Project Confidence Score Distribution by Gender",
      x = "Average Pre-Project Confidence Score",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```

```
## Warning: Removed 130 rows containing non-finite values (stat_bin).
```



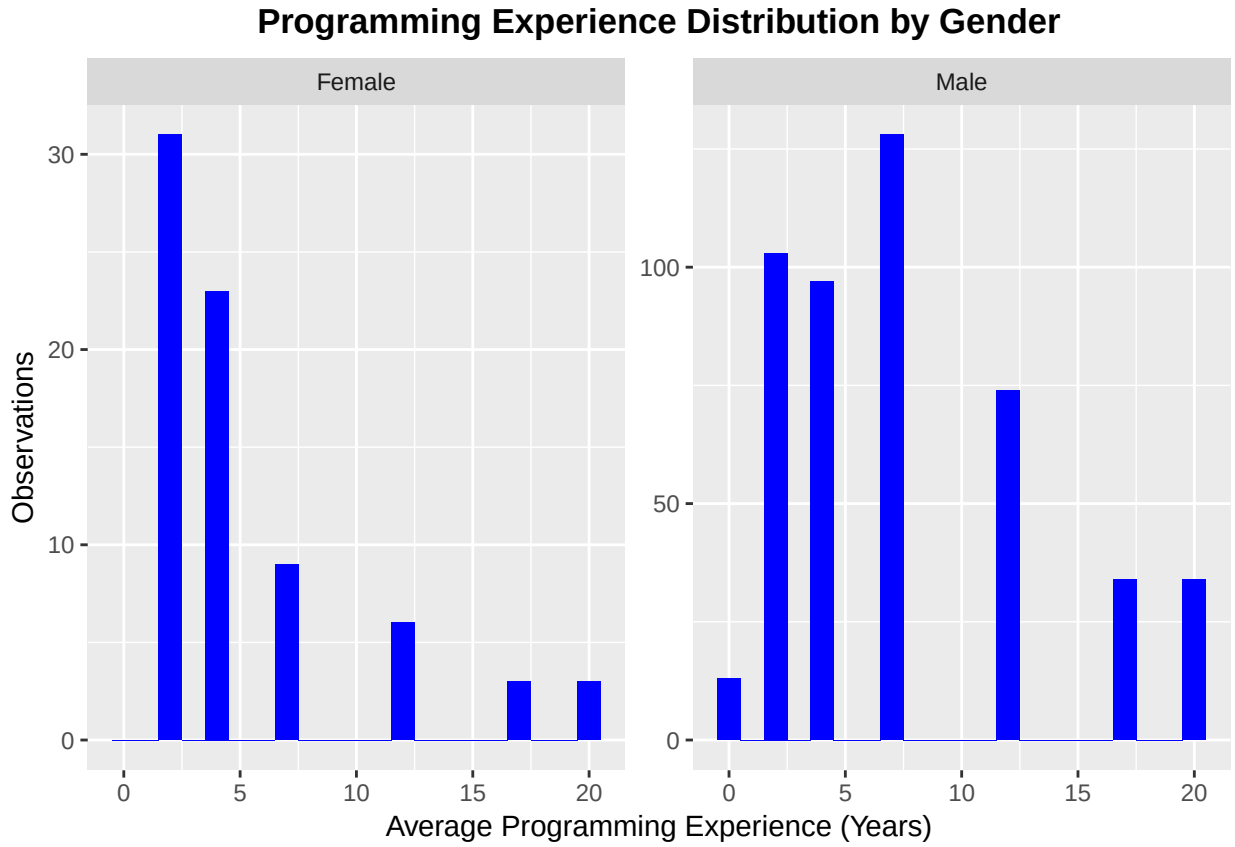

```
# Histogram of conf_post_ave by gender
ggplot(subset(kbai, !is.na(gender) & w_ind==0), aes(x = conf_post_ave)) +
    geom_histogram(fill = "blue", binwidth = 0.5) +
    facet_wrap(~gender, scale = "free_y") +
    labs(title = "Post-Project Confidence Score Distribution by Gender",
      x = "Average Post-Project Confidence Score",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```

```
## Warning: Removed 194 rows containing non-finite values (stat_bin).
```



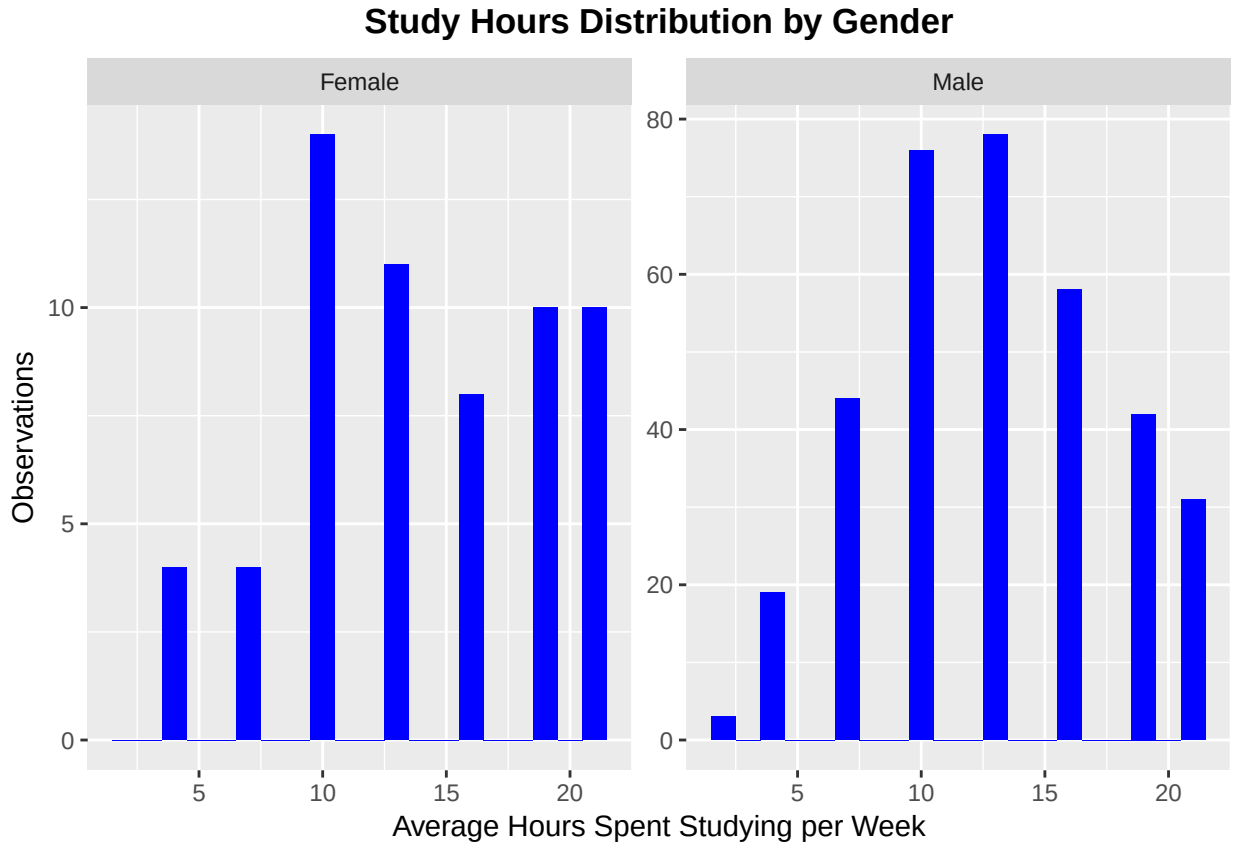

```
# Histogram of programming experience by gender
ggplot(subset(kbai, !is.na(gender) & w_ind==0), aes(x = prog_num)) +
    geom_histogram(fill = "blue", binwidth = 1) +
    facet_wrap(~gender, scale = "free_y") +
    labs(title = "Programming Experience Distribution by Gender",
      x = "Average Programming Experience (Years)",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```

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




```
# Histogram of study hours by gender
ggplot(subset(kbai, !is.na(gender) & w_ind==0), 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 147 rows containing non-finite values (stat_bin).
```




```
# Age tests
t.test(kbai_m$age_num, kbai_f$age_num)
```

```
##
##  Welch Two Sample t-test
##
## data:  kbai_m$age_num and kbai_f$age_num
## t = -0.54792, df = 95.608, p-value = 0.585
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.530699  1.435863
## sample estimates:
## mean of x mean of y
##  32.02495  32.57237
```

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

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  age_num by gender
## W = 19517, p-value = 0.6935
## alternative hypothesis: true location shift is not equal to 0
```

```
# Overall grade tests
t.test(kbai_m$total, kbai_f$total)
```

```
##
```

```
##  Welch Two Sample t-test
##
## data:  kbai_m$total and kbai_f$total
## t = -1.225, df = 108.74, p-value = 0.2232
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.394475  1.037280
## sample estimates:
## mean of x mean of y
##  78.13577  79.81436
```

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

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  total by gender
## W = 20428, p-value = 0.2911
## alternative hypothesis: true location shift is not equal to 0
```

```
# Assignment grade tests
t.test(kbai_m$assign_ave, kbai_f$assign_ave)
```

```
##
##  Welch Two Sample t-test
##
## data:  kbai_m$assign_ave and kbai_f$assign_ave
## t = -2.4643, df = 112.71, p-value = 0.01524
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -5.6530580 -0.6143104
## sample estimates:
## mean of x mean of y
##  79.67333  82.80702
```

```
wilcox.test(assign_ave ~ gender, data=kbai)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  assign_ave by gender
## W = 21668, p-value = 0.04834
## alternative hypothesis: true location shift is not equal to 0
```

```
# Project grade tests
t.test(kbai_m$proj_ave, kbai_f$proj_ave)
```

```
##
##  Welch Two Sample t-test
##
## data:  kbai_m$proj_ave and kbai_f$proj_ave
## t = -0.47069, df = 98.758, p-value = 0.6389
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.870007  3.002566
## sample estimates:
## mean of x mean of y
```

```
##  70.96540  71.89912
```

```
wilcox.test(proj_ave ~ gender, data=kbai)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  proj_ave by gender
## W = 19914, p-value = 0.5179
## alternative hypothesis: true location shift is not equal to 0
```

```
# Exam grade tests
t.test(kbai_m$exam, kbai_f$exam)
```

```
##
##  Welch Two Sample t-test
##
## data:  kbai_m$exam and kbai_f$exam
## t = -1.4456, df = 116.54, p-value = 0.151
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -6.2525607  0.9762186
## sample estimates:
## mean of x mean of y
##  84.82236  87.46053
```

```
wilcox.test(exam ~ gender, data=kbai)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  exam by gender
## W = 20983, p-value = 0.1506
## alternative hypothesis: true location shift is not equal to 0
```

```
# Withdrawal rate tests
t.test(kbai_m$w_ind, kbai_f$w_ind)
```

```
##
##  Welch Two Sample t-test
##
## data:  kbai_m$w_ind and kbai_f$w_ind
## t = 1.3446, df = 139.56, p-value = 0.1809
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.009771395  0.051319877
## sample estimates:
##  mean of x  mean of y
## 0.03393214 0.01315789
```

```
wilcox.test(w_ind ~ gender, data=kbai)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  w_ind by gender
## W = 18642, p-value = 0.3327
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Average confidence score tests
t.test(kbai_m$conf_ave, kbai_f$conf_ave)
```

```
##
##  Welch Two Sample t-test
##
## data:  kbai_m$conf_ave and kbai_f$conf_ave
## t = 0.1772, df = 79.474, p-value = 0.8598
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1885382  0.2253929
## sample estimates:
## mean of x mean of y
##  3.696393  3.677966
```

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

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  conf_ave by gender
## W = 8906.5, p-value = 0.9022
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Average pre-project confidence score tests
t.test(kbai_m$conf_pre_ave, kbai_f$conf_pre_ave)
```

```
##
##  Welch Two Sample t-test
##
## data:  kbai_m$conf_pre_ave and kbai_f$conf_pre_ave
## t = 0.56413, df = 85.746, p-value = 0.5741
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1671333  0.2995657
## sample estimates:
## mean of x mean of y
##  3.566216  3.500000
```

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

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  conf_pre_ave by gender
## W = 11062, p-value = 0.391
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Average post-project confidence score tests
t.test(kbai_m$conf_post_ave, kbai_f$conf_post_ave)
```

```
##
##  Welch Two Sample t-test
##
## data:  kbai_m$conf_post_ave and kbai_f$conf_post_ave
## t = 0.10526, df = 79.518, p-value = 0.9164
## alternative hypothesis: true difference in means is not equal to 0
```

```
## 95 percent confidence interval:
##  -0.2266485  0.2519622
## sample estimates:
## mean of x mean of y
##  3.764069  3.751412
```

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

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  conf_post_ave by gender
## W = 9124, p-value = 0.9596
## alternative hypothesis: true location shift is not equal to 0
```

```
# Programming experience tests
t.test(kbai_m$prog_num, kbai_f$prog_num)
```

```
##
##  Welch Two Sample t-test
##
## data:  kbai_m$prog_num and kbai_f$prog_num
## t = 3.8954, df = 107.77, p-value = 0.0001705
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.190558 3.657652
## sample estimates:
## mean of x mean of y
##  7.832000  5.407895
```

```
wilcox.test(prog_num ~ gender, data=kbai)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  prog_num by gender
## W = 13696, p-value = 6.122e-05
## alternative hypothesis: true location shift is not equal to 0
```

```
# Study hours
t.test(kbai_m$hours_num, kbai_f$hours_num)
```

```
##
##  Welch Two Sample t-test
##
## data:  kbai_m$hours_num and kbai_f$hours_num
## t = -1.5559, df = 79.767, p-value = 0.1237
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.4686227  0.3023198
## sample estimates:
## mean of x mean of y
##  13.35127  14.43443
```

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

```
##
##  Wilcoxon rank sum test with continuity correction
```

```
##
## data:  hours_num by gender
## W = 12115, p-value = 0.1129
## alternative hypothesis: true location shift is not equal to 0
```

```
# Native speaker
t.test(kbai_m$native_ind, kbai_f$native_ind)
```

```
##
##  Welch Two Sample t-test
##
## data:  kbai_m$native_ind and kbai_f$native_ind
## t = 3.3516, df = 96.282, p-value = 0.001149
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.08378822 0.32715915
## sample estimates:
## mean of x mean of y
## 0.6660000 0.4605263
```

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

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  native_ind by gender
## W = 15096, p-value = 0.0005183
## alternative hypothesis: true location shift is not equal to 0
```

```
# Higher education
t.test(kbai_m$higher_ind, kbai_f$higher_ind)
```

```
##
##  Welch Two Sample t-test
##
## data:  kbai_m$higher_ind and kbai_f$higher_ind
## t = -2.5059, df = 93.671, p-value = 0.01394
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.26958950 -0.03122991
## sample estimates:
## mean of x mean of y
## 0.2574850 0.4078947
```

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

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  higher_ind by gender
## W = 21902, p-value = 0.006401
## alternative hypothesis: true location shift is not equal to 0
```

Regression Analysis

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

```
##                 age_num   native_ind higher_ind  gender_ind
## age_num     1.000000000 -0.005179033  0.2244728 -0.02482279
## native_ind -0.005179033  1.000000000 -0.2358161  0.14477462
## higher_ind  0.224472823 -0.235816120  1.0000000 -0.11325869
## gender_ind -0.024822786  0.144774619 -0.1132587  1.00000000
```

```
# Fit regression to total grade data
total_lm = lm(total~gender + age_num + native_ind + higher_ind + semester,
              data=na.omit(kbai))

summary(total_lm)
```

```
##
## Call:
## lm(formula = total ~ gender + age_num + native_ind + higher_ind +
##     semester, data = na.omit(kbai))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.147  -3.706   0.841   5.320  16.715
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          80.75151    2.15159  37.531  < 2e-16 ***
## genderMale           -1.37057    1.19611  -1.146 0.252626
## age_num              -0.08533    0.05537  -1.541 0.124176
## native_ind            3.49499    0.95981   3.641 0.000311 ***
## higher_ind            2.65595    1.00694   2.638 0.008714 **
## semesterSummer 2016   1.88543    0.87867   2.146 0.032567 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.195 on 356 degrees of freedom
## Multiple R-squared:  0.0622, Adjusted R-squared:  0.04903
## F-statistic: 4.722 on 5 and 356 DF,  p-value: 0.0003439
```

```
# Fit regression to assignment grade data
assign_lm = lm(assign_ave~gender + age_num + native_ind + higher_ind + semester,
              data=na.omit(kbai))

summary(assign_lm)
```

```
##
## Call:
## lm(formula = assign_ave ~ gender + age_num + native_ind + higher_ind +
##     semester, data = na.omit(kbai))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -55.600  -4.767   0.324   5.715  22.187
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         76.96464    2.34636  32.802  < 2e-16 ***
## genderMale          -2.69234    1.30439  -2.064 0.039736 *
## age_num              0.02875    0.06038   0.476 0.634261
## native_ind           3.81328    1.04669   3.643 0.000309 ***
## higher_ind           2.00583    1.09809   1.827 0.068588 .
## semesterSummer 2016  6.72933    0.95821   7.023 1.11e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.937 on 356 degrees of freedom
## Multiple R-squared:  0.1624, Adjusted R-squared:  0.1507
## F-statistic: 13.81 on 5 and 356 DF,  p-value: 2.449e-12
```

```
# Fit regression to project grade data
project_lm = lm(proj_ave~gender + age_num + native_ind + higher_ind + semester,
              data=na.omit(kbai))

summary(project_lm)
```

```
##
## Call:
## lm(formula = proj_ave ~ gender + age_num + native_ind + higher_ind +
##     semester, data = na.omit(kbai))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.048  -6.611   1.714   8.664  24.423
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         75.84945    3.36778  22.522   <2e-16 ***
## genderMale          -1.53107    1.87222  -0.818   0.4140
## age_num             -0.19600    0.08667  -2.262   0.0243 *
## native_ind           3.41607    1.50234   2.274   0.0236 *
## higher_ind           3.53075    1.57611   2.240   0.0257 *
## semesterSummer 2016  3.81768    1.37534   2.776   0.0058 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.83 on 356 degrees of freedom
## Multiple R-squared:  0.05494,    Adjusted R-squared:  0.04167
## F-statistic: 4.139 on 5 and 356 DF,  p-value: 0.001146
```

```
# Fit regression to exam grade data
exam_lm = lm(exam~gender + age_num + native_ind + higher_ind + semester,
              data=na.omit(kbai))

summary(exam_lm)
```

```
##
## Call:
## lm(formula = exam ~ gender + age_num + native_ind + higher_ind +
```

```
##     semester, data = na.omit(kbai))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -89.271  -2.214   1.532   4.741  15.858
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         91.43769    2.62509  34.832  < 2e-16 ***
## genderMale          -0.54754    1.45934  -0.375  0.70774
## age_num             -0.05488    0.06755  -0.812  0.41709
## native_ind           3.19683    1.17103   2.730  0.00665 **
## higher_ind           3.47516    1.22853   2.829  0.00494 **
## semesterSummer 2016 -7.67964    1.07204  -7.164 4.55e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.999 on 356 degrees of freedom
## Multiple R-squared:  0.1502, Adjusted R-squared:  0.1382
## F-statistic: 12.58 on 5 and 356 DF,  p-value: 2.906e-11
```

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

summary(conf_lm)
```

```
##
## Call:
## lm(formula = conf_ave ~ gender + age_num + native_ind + higher_ind +
##     semester, data = na.omit(kbai))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4137 -0.4348  0.0893  0.4797  1.4246
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.745737   0.184657  20.285   <2e-16 ***
## genderMale           0.059488   0.102655   0.579   0.5626
## age_num             -0.004747   0.004752  -0.999   0.3185
## native_ind          -0.030360   0.082374  -0.369   0.7127
## higher_ind           0.148451   0.086419   1.718   0.0867 .
## semesterSummer 2016  0.085433   0.075411   1.133   0.2580
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7033 on 356 degrees of freedom
## Multiple R-squared:  0.01514,    Adjusted R-squared:  0.001312
## F-statistic: 1.095 on 5 and 356 DF,  p-value: 0.3628
```

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

```r
summary(conf_pre_lm)
```

```
##
## Call:
## lm(formula = conf_pre_ave ~ gender + age_num + native_ind + higher_ind +
##     semester, data = na.omit(kbai))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6689 -0.4911  0.0433  0.5089  1.6160
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.554530   0.222567  15.971   <2e-16 ***
## genderMale           0.060566   0.123729   0.490   0.6248
## age_num             -0.003445   0.005727  -0.601   0.5479
## native_ind          -0.022326   0.099285  -0.225   0.8222
## higher_ind           0.135047   0.104160   1.297   0.1956
## semesterSummer 2016  0.177750   0.090892   1.956   0.0513 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8477 on 356 degrees of freedom
## Multiple R-squared:  0.01664,    Adjusted R-squared:  0.002833
## F-statistic: 1.205 on 5 and 356 DF,  p-value: 0.3062
```

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

summary(conf_post_lm)
```

```
##
## Call:
## lm(formula = conf_post_ave ~ gender + age_num + native_ind +
##     higher_ind + semester, data = na.omit(kbai))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7996 -0.4212  0.1324  0.6012  1.3578
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.873208   0.215508  17.972   <2e-16 ***
## genderMale           0.058769   0.119805   0.491    0.624
## age_num             -0.005615   0.005546  -1.012    0.312
## native_ind          -0.035717   0.096136  -0.372    0.710
## higher_ind           0.157386   0.100857   1.560    0.120
## semesterSummer 2016  0.023889   0.088010   0.271    0.786
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8208 on 356 degrees of freedom
## Multiple R-squared:  0.01032,    Adjusted R-squared:  -0.003576
```

```
## F-statistic: 0.7427 on 5 and 356 DF,  p-value: 0.5919
```

```
# Fit regression to programming experience
prog_lm = lm(prog_num~gender + age_num + native_ind + higher_ind + semester,
             data=na.omit(kbai))

summary(prog_lm)
```

```
##
## Call:
## lm(formula = prog_num ~ gender + age_num + native_ind + higher_ind +
##     semester, data = na.omit(kbai))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2331  -3.2432  -0.6657   3.7543  13.7638
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -6.15959    1.30917  -4.705 3.64e-06 ***
## genderMale           2.15308    0.72779   2.958  0.00330 **
## age_num              0.38868    0.03369  11.537  < 2e-16 ***
## native_ind           1.08747    0.58401   1.862  0.06342 .
## higher_ind          -2.31079    0.61269  -3.772  0.00019 ***
## semesterSummer 2016 -0.35557    0.53464  -0.665  0.50644
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.986 on 356 degrees of freedom
## Multiple R-squared:  0.3156, Adjusted R-squared:  0.306
## F-statistic: 32.83 on 5 and 356 DF,  p-value: < 2.2e-16
```

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

summary(hours_lm)
```

```
##
## Call:
## lm(formula = hours_num ~ gender + age_num + native_ind + higher_ind +
##     semester, data = na.omit(kbai))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.5684  -3.5298   0.1562   3.9807   9.2107
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         10.53670    1.24618   8.455 7.27e-16 ***
## genderMale          -0.60948    0.69277  -0.880 0.379579
## age_num              0.12319    0.03207   3.841 0.000145 ***
## native_ind          -0.72495    0.55591  -1.304 0.193051
## higher_ind           0.55619    0.58321   0.954 0.340895
## semesterSummer 2016 -0.07551    0.50892  -0.148 0.882131
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.747 on 356 degrees of freedom
## Multiple R-squared:  0.05642,    Adjusted R-squared:  0.04317
## F-statistic: 4.258 on 5 and 356 DF,  p-value: 0.0008982
```