Educational Technology Project - Survey Data Analysis

Process Data

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

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

```
##
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```r
# Read in survey data set
survey = read.csv('20170414_Survey_Responses.csv')
```

```r
# Replace blanks with NA
is.na(survey) = (survey=="")

# Convert factors into character strings
survey$residence = as.character(survey$residence)
survey$birth = as.character(survey$birth)
survey$language = as.character(survey$language)
survey$education_field = as.character(survey$education_field)
survey$omscs_reason = as.character(survey$omscs_reason)
survey$specialization = as.character(survey$specialization)
survey$prog_languages = as.character(survey$prog_languages)

# Simplify level names
survey$education_level = revalue(survey$education_level,
                        c("Bachelor's degree (or equivalent)"="Bachelors",
                        "PhD degree (or equivalent)"="Doctorate",
                        "Master's degree (or equivalent)" = "Masters"))

course_names = colnames(survey)[12:39]

for(i in seq(1, length(course_names))){
    name = course_names[i]
    survey[, name] = as.character(survey[, name])
    survey[, name] = ifelse(survey[, name] == "Attempted but did not complete",
                            "Attempted", survey[, name])
    survey[, name] = ifelse(survey[, name] == "Currently undertaking (Spring 2017)",
                            "Current", survey[, name])
    survey[, name] = ifelse(is.na(survey[, name]), "Not Attempted", survey[, name])
    survey[,name] = factor(survey[,name], levels = c("Completed", "Current", "Attempted",
                                                     "Not Attempted"))
}

# Determine counts of courses completed, attempted and current for each respondent
# Initialize counter variables
survey$completed = 0
survey$attempted = 0
survey$current = 0

for(j in seq(1, dim(survey)[1])){
  for(i in seq(1, length(course_names))){
    name = course_names[i]
    if(survey[j, name]=="Completed"){
      survey$completed[j] = survey$completed[j] + 1
    } else if(survey[j, name]=="Attempted"){
      survey$attempted[j] = survey$attempted[j] + 1
    } else if(survey[j, name]=="Current"){
      survey$current[j] = survey$current[j] + 1
    }
  }
}

# Convert scales to numeric
survey$conf_post = revalue(survey$conf_post, c("Very confident" = 5, "Confident" = 4,
                      "Neutral" = 3, "Unconfident" = 2, "Very unconfident" = 1))

survey$conf_post = as.numeric(as.character(survey$conf_post))

survey$conf_prior = revalue(survey$conf_prior, c("Very confident" = 5, "Confident" = 4,
                      "Neutral" = 3, "Unconfident" = 2, "Very unconfident" = 1))

survey$conf_prior = as.numeric(as.character(survey$conf_prior))

statements_pos = colnames(survey)[c(49, 50, 51, 52, 53, 54, 56, 57, 58, 62, 64, 66,
                                    68, 69)]
statements_neg = colnames(survey)[c(55, 59, 60, 61, 63, 65, 67, 70)]

for(i in seq(1, length(statements_pos))){
    name = statements_pos[i]
    survey[, name] = revalue(survey[, name], c("Strongly Agree" = 5, "Agree" = 4,
                      "Neutral" = 3, "Disagree" = 2, "Strongly disagree" = 1))
    survey[, name] = as.numeric(as.character(survey[, name]))
}
```

```
## The following `from` values were not present in `x`: Strongly disagree

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

```
for(i in seq(1, length(statements_neg))){
    name = statements_neg[i]
    survey[, name] = revalue(survey[, name], c("Strongly Agree" = 1, "Agree" = 2,
                      "Neutral" = 3, "Disagree" = 4, "Strongly disagree" = 5))
    survey[, name] = as.numeric(as.character(survey[, name]))
}
```

```
# Create average confidence score
survey$conf_ave = (survey$conf_prior + survey$conf_post)/2

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

# Clean language
survey$language = ifelse(survey$language == "korean", "Korean", survey$language)
survey$language = ifelse(survey$language == "Bahasa Indonesia", "Indonesian",
                         survey$language)
survey$language = ifelse(survey$language %in% c("Telugu (Indian dialect)", "Marathi",
                  "Bengali", "Kannada", "Kannada ", "Hindi", "Tamil"), "Indian",
                  survey$language)
survey$language = ifelse(survey$language == "Tagalog (Filipino)", "Tagalog",
                         survey$language)

# Clean education field
survey$education_field = ifelse(survey$education_field == "Engineering and Theology (2)",
                                "Engineering", survey$education_field)
survey$education_field = ifelse(survey$education_field ==
                  "CS,  History, and Classical Studies", "Computer science",
                  survey$education_field)
survey$education_field = ifelse(survey$education_field == "Mathematics Education",
                                "Mathematics/statistics", survey$education_field)
survey$education_field = ifelse(survey$education_field == "biology",
                                "Physical/life sciences", survey$education_field)
survey$education_field = ifelse(survey$education_field == "Business Management",
                                "Economics/business", survey$education_field)
survey$education_field = ifelse(survey$education_field %in% c("Theology",
              "Social sciences", "Education"), "Humanities/arts", survey$education_field)
survey$education_field = ifelse(survey$education_field %in% c("Psychology",
                "Information Systems Management", "Information Technology"),
                                "Other sciences", survey$education_field)

# Create factors
survey$birth = factor(survey$birth)
survey$residence = factor(survey$residence)
survey$language = factor(survey$language)
survey$education_field = factor(survey$education_field)

# Reorder levels of factors where necessary
survey$omscs_semester = factor(survey$omscs_semester, levels = c("Spring 2014",
                          "Summer 2014", "Fall 2014", "Spring 2015", "Summer 2015",
                          "Fall 2015", "Spring 2016", "Summer 2016", "Fall 2016",
                          "Spring 2017"))

survey$hours = factor(survey$hours, levels = c("< 5", "5 - 9", "10 - 14", "15 - 19",
                              "20 - 24", "25 - 29", "30 - 34", "35 - 39", "40 +"))

survey$prog_years = factor(survey$prog_years, levels = c("0", "1", "2", "3 - 5", "6 - 8",
                                    "9 - 11", "12 - 14", "15+"))

survey$prior_cs_exp = factor(survey$prior_cs_exp, levels = c("0", "1", "2", "3 - 5",
                                                             "6 - 8", "9 - 11", "12+"))

# Convert ranges to numeric values
survey$age_num = revalue(survey$age, c("20 - 24"=22, "25 - 29"=27, "30 - 34"=32,
                                       "35 - 39"=37, "40 - 44"=42, "45 - 49"=47,
                                       "50 - 54"=52, "55 - 59"=57, "60 - 64"=62))
survey$age_num = as.numeric(as.character(survey$age_num))

survey$gpa_num = revalue(survey$gpa, c("2.5 - 2.9" = 2.7, "3.0 - 3.4" = 3.2,
                                       "3.5 - 3.9" = 3.7, "Don't Know" = NA))
survey$gpa_num = as.numeric(as.character(survey$gpa_num))

survey$hours_num = revalue(survey$hours, c("< 5"=2, "5 - 9"=7, "10 - 14"=12, "15 - 19"=17,
                              "20 - 24"=22, "25 - 29"=27, "30 - 34"=32, "35 - 39"=37,
                              "40 +"=42))
survey$hours_num = as.numeric(as.character(survey$hours_num))

survey$prog_num = revalue(survey$prog_years, c("3 - 5"= 4, "6 - 8"=7, "9 - 11"=10,
                                               "12 - 14"=13, "15+"=16))
survey$prog_num = as.numeric(as.character(survey$prog_num))

survey$prior_cs_num = revalue(survey$prior_cs_exp, c("3 - 5"=4, "6 - 8"=7, "9 - 11"=10,
                                                     "12+"=13))
survey$prior_cs_num = as.numeric(as.character(survey$prior_cs_num))

# Create programming language count variable
count_items = function(x){
  if(is.na(x)|x == "I had not previously had any programming experience"){
    num = 0
  }else{
    num = length(unlist(strsplit(x, ", ")))
  }
  return (num)
}

survey$prog_count = sapply(survey$prog_languages, count_items)

# Create specialization indicators
spec_list = c("Computational Perception and Robotics", "Computing Systems",
              "Interactive Intelligence", "Machine Learning")

for(i in seq(1, length(spec_list))){
    spec = spec_list[i]
    if(spec=="Computational Perception and Robotics"){
      name = "spec_robotics"
    } else if(spec=="Computing Systems"){
      name = "spec_systems"
    } else if(spec=="Interactive Intelligence"){
      name = "spec_intelligence"
    } else{
      name = "spec_ml"
    }
    survey[, name] = as.numeric(grepl(spec, survey$specialization))
}

# Create reason indicators
reason_list = c("To increase financial prospects", "To gain an extra qualification",
                "To gain promotion within your current industry of employment",
    "To switch to a career in a different industry from where you are currently employed",
                "To make connections", "For fun/challenge")

for(i in seq(1, length(reason_list))){
    reason = reason_list[i]
    if(reason=="To increase financial prospects"){
      name = "reason_financial"
    } else if(reason=="To gain an extra qualification"){
      name = "reason_quals"
    } else if(reason=="To gain promotion within your current industry of employment"){
      name = "reason_promotion"
    } else if(reason=="To switch to a career in a different industry from where you are currently employ
      name = "reason_switch"
    } else if(reason=="To make connections"){
      name = "reason_connections"
    } else{
      name = "reason_fun"
    }
    survey[, name] = as.numeric(grepl(reason, survey$omscs_reason))
}

# Create indicator variables
survey$cs_study_ind = ifelse(survey$prior_cs_study == "Yes", 1, 0)
survey$native_ind = ifelse(survey$language == "English", 1, 0)
survey$us_birth_ind = ifelse(survey$birth == "USA", 1, 0)
survey$us_res_ind = ifelse(survey$residence == "USA", 1, 0)
survey$higher_ind = ifelse(survey$education_level %in% c("Masters", "Doctorate"), 1, 0)
```

Create Score Variables

```
# Determine correlations between statement agreement scores by group
# Self-confidence
sc_corr = cor(survey[, c("selfconf1", "selfconf2", "selfconf3", "selfconf4", "selfconf5",
               "selfconf6", "selfconf7")])
sc_corr[sc_corr == 1] <- NA

colMeans(sc_corr, na.rm = TRUE)
```

```
## selfconf1 selfconf2 selfconf3 selfconf4 selfconf5 selfconf6 selfconf7
## 0.5660881 0.5261964 0.5557215 0.4976482 0.4393480 0.5442213 0.2952214
```

```
mean(sc_corr, na.rm = TRUE)
```

```
## [1] 0.4892064
```

```
# Equality
eq_corr = cor(survey[, c("equality1", "equality2", "equality3", "equality4", "equality5",
               "equality6")])
eq_corr[eq_corr == 1] <- NA

colMeans(eq_corr, na.rm = TRUE)
```

```
## equality1 equality2 equality3 equality4 equality5 equality6
## 0.5257517 0.5068182 0.5499074 0.5203307 0.2163926 0.4983357
```

```
mean(eq_corr, na.rm = TRUE)
```

```
## [1] 0.4695894
```

```
# Belonging
be_corr = cor(survey[, c("belonging1", "belonging2", "belonging3", "belonging4",
                         "belonging5", "belonging6", "belonging7", "belonging8",
                         "belonging9")])
be_corr[be_corr == 1] <- NA

colMeans(be_corr, na.rm = TRUE)
```

```
## belonging1 belonging2 belonging3 belonging4 belonging5 belonging6
##  0.3916325  0.3921835  0.4557640  0.1961097  0.3264607  0.4143715
## belonging7 belonging8 belonging9
##  0.3608049  0.4369175  0.2831651
```

```
mean(be_corr, na.rm = TRUE)
```

```
## [1] 0.3619344
```

```
# Check Cronbach's alpha for each group
psych::alpha(survey[, c("selfconf1", "selfconf2", "selfconf3", "selfconf4", "selfconf5",
               "selfconf6", "selfconf7")])
```

```
##
## Reliability analysis
## Call: psych::alpha(x = survey[, c("selfconf1", "selfconf2", "selfconf3",
##     "selfconf4", "selfconf5", "selfconf6", "selfconf7")])
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean   sd
##       0.86      0.87    0.87      0.49 6.7 0.016  4.1 0.65
##
##  lower alpha upper     95% confidence boundaries
## 0.83 0.86 0.89
##
##  Reliability if an item is dropped:
##           raw_alpha std.alpha G6(smc) average_r S/N alpha se
## selfconf1      0.82      0.84    0.83      0.46 5.1    0.021
## selfconf2      0.84      0.84    0.84      0.47 5.4    0.020
## selfconf3      0.83      0.84    0.83      0.46 5.2    0.021
## selfconf4      0.84      0.85    0.84      0.49 5.7    0.019
## selfconf5      0.85      0.86    0.86      0.51 6.2    0.018
## selfconf6      0.83      0.84    0.84      0.47 5.3    0.021
## selfconf7      0.88      0.89    0.88      0.57 7.9    0.014
##
##  Item statistics
##             n raw.r std.r r.cor r.drop mean   sd
## selfconf1 160  0.84  0.84  0.82   0.76  4.2 0.88
## selfconf2 160  0.77  0.79  0.75   0.70  4.5 0.63
## selfconf3 160  0.82  0.83  0.81   0.74  3.8 0.93
## selfconf4 160  0.76  0.76  0.72   0.66  3.9 0.85
## selfconf5 160  0.67  0.69  0.61   0.57  4.3 0.75
## selfconf6 160  0.83  0.81  0.78   0.73  3.8 1.03
## selfconf7 160  0.56  0.53  0.40   0.37  4.2 1.02
##
## Non missing response frequency for each item
##              1    2    3    4    5 miss
## selfconf1 0.01 0.04 0.13 0.42 0.39    0
## selfconf2 0.00 0.01 0.06 0.39 0.54    0
## selfconf3 0.02 0.06 0.25 0.44 0.23    0
## selfconf4 0.01 0.02 0.29 0.42 0.26    0
## selfconf5 0.01 0.01 0.11 0.45 0.42    0
## selfconf6 0.01 0.11 0.22 0.34 0.32    0
## selfconf7 0.02 0.08 0.11 0.31 0.48    0
```

```
psych::alpha(survey[, c("equality1", "equality2", "equality3", "equality4", "equality5",
               "equality6")])
```

```
##
## Reliability analysis
## Call: psych::alpha(x = survey[, c("equality1", "equality2", "equality3",
##     "equality4", "equality5", "equality6")])
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean   sd
##       0.79      0.84    0.87      0.47 5.3 0.028  4.4 0.62
##
##  lower alpha upper     95% confidence boundaries
## 0.73 0.79 0.84
##
##  Reliability if an item is dropped:
##           raw_alpha std.alpha G6(smc) average_r S/N alpha se
## equality1      0.73      0.80    0.83      0.44 4.0    0.036
## equality2      0.75      0.80    0.82      0.45 4.1    0.034
## equality3      0.74      0.79    0.80      0.43 3.8    0.035
## equality4      0.73      0.80    0.82      0.44 4.0    0.036
## equality5      0.88      0.88    0.89      0.60 7.4    0.016
## equality6      0.73      0.81    0.82      0.46 4.2    0.037
##
##  Item statistics
##             n raw.r std.r r.cor r.drop mean   sd
## equality1 160  0.78  0.81  0.76   0.68  4.5 0.77
## equality2 160  0.73  0.79  0.77   0.61  4.7 0.72
## equality3 160  0.77  0.84  0.84   0.68  4.7 0.68
## equality4 160  0.78  0.80  0.77   0.68  4.7 0.73
## equality5 160  0.59  0.46  0.29   0.27  3.3 1.34
## equality6 160  0.78  0.78  0.74   0.65  4.5 0.88
##
## Non missing response frequency for each item
##              1    2    3    4    5 miss
## equality1 0.01 0.01 0.08 0.26 0.64    0
## equality2 0.01 0.01 0.05 0.17 0.76    0
## equality3 0.02 0.00 0.02 0.16 0.81    0
## equality4 0.01 0.02 0.03 0.11 0.82    0
## equality5 0.07 0.29 0.19 0.16 0.29    0
## equality6 0.02 0.03 0.06 0.18 0.71    0
```

```
psych::alpha(survey[, c("belonging1", "belonging2", "belonging3", "belonging4",
                         "belonging5", "belonging6", "belonging7", "belonging8",
                         "belonging9")])
```

```
##
## Reliability analysis
## Call: psych::alpha(x = survey[, c("belonging1", "belonging2", "belonging3",
##     "belonging4", "belonging5", "belonging6", "belonging7", "belonging8",
##     "belonging9")])
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean  sd
##       0.82      0.84    0.86      0.36 5.1 0.021  3.9 0.7
##
##  lower alpha upper     95% confidence boundaries
## 0.78 0.82 0.87
##
##  Reliability if an item is dropped:
##            raw_alpha std.alpha G6(smc) average_r S/N alpha se
## belonging1      0.80      0.81    0.84      0.35 4.4    0.024
## belonging2      0.80      0.81    0.83      0.35 4.4    0.025
## belonging3      0.79      0.80    0.82      0.34 4.0    0.025
## belonging4      0.84      0.85    0.86      0.41 5.5    0.019
## belonging5      0.82      0.83    0.85      0.37 4.7    0.022
## belonging6      0.79      0.81    0.83      0.35 4.3    0.025
## belonging7      0.81      0.82    0.84      0.36 4.5    0.023
## belonging8      0.80      0.81    0.82      0.34 4.1    0.024
## belonging9      0.82      0.83    0.85      0.38 5.0    0.022
##
##  Item statistics
##              n raw.r std.r r.cor r.drop mean   sd
## belonging1 160  0.68  0.70  0.65   0.58  3.7 0.99
## belonging2 160  0.73  0.70  0.66   0.61  3.8 1.25
## belonging3 160  0.76  0.78  0.78   0.68  4.1 0.94
## belonging4 160  0.49  0.43  0.33   0.30  3.0 1.31
## belonging5 160  0.59  0.61  0.54   0.46  3.4 1.08
## belonging6 160  0.75  0.73  0.70   0.65  4.0 1.15
## belonging7 160  0.61  0.66  0.61   0.52  4.5 0.78
## belonging8 160  0.72  0.76  0.75   0.64  4.0 0.97
## belonging9 160  0.57  0.55  0.47   0.43  4.2 1.13
##
## Non missing response frequency for each item
##               1    2    3    4    5 miss
## belonging1 0.04 0.05 0.28 0.43 0.20    0
## belonging2 0.02 0.20 0.15 0.20 0.42    0
## belonging3 0.02 0.05 0.14 0.42 0.37    0
## belonging4 0.12 0.29 0.21 0.19 0.18    0
## belonging5 0.05 0.18 0.27 0.36 0.14    0
## belonging6 0.03 0.11 0.15 0.26 0.46    0
## belonging7 0.02 0.01 0.05 0.28 0.65    0
## belonging8 0.02 0.05 0.14 0.41 0.37    0
## belonging9 0.02 0.10 0.11 0.21 0.56    0
```

```
# Create average scores for each measure
survey$selfconf_score = rowMeans(survey[, c("selfconf1", "selfconf2", "selfconf3",
                              "selfconf4", "selfconf5", "selfconf6", "selfconf7")])

survey$equality_score = rowMeans(survey[, c("equality1", "equality2", "equality3",
                          "equality4", "equality5", "equality6")])

survey$belonging_score = rowMeans(survey[, c("belonging1", "belonging2", "belonging3",
                          "belonging4", "belonging5", "belonging6", "belonging7",
                          "belonging8", "belonging9")])
```

Explore Data

```
# Calculate summary statistics
summary(survey)
```

```
##                 timestamp      gender         age         residence
##  3-23-2017 20:03:44:  1   Female: 57   30 - 34:38   USA      :139
##  3-23-2017 20:05:41:  1   Male  :103   25 - 29:36   Canada   :  7
##  3-23-2017 20:28:09:  1                35 - 39:28   India    :  3
##  3-23-2017 20:36:25:  1                45 - 49:15   Singapore:  2
##  3-23-2017 20:44:03:  1                40 - 44:14   Australia:  1
##  3-23-2017 21:07:05:  1                20 - 24:11   Brazil   :  1
##  (Other)           :154                (Other):18   (Other)  :  7
##       birth          language    education_level
##  USA      :92   English   :104   Bachelors:113
##  India    :25   Chinese   : 17   Masters  : 38
##  China    :12   Indian    : 15   Doctorate:  9
##  Indonesia: 3   Spanish   :  8
##  Ecuador  : 2   Indonesian:  3
##  Mexico   : 2   Korean    :  2
##  (Other)  :24   (Other)   : 11
##                  education_field omscs_yn  omscs_reason
##  Computer science      :73     No :  7    Length:160
##  Economics/business    : 5     Yes:153    Class :character
##  Engineering           :50                Mode  :character
##  Humanities/arts       : 6
##  Mathematics/statistics:12
##  Other sciences        : 3
##  Physical/life sciences:11
##      omscs_semester           cs6035              cs6210
##  Spring 2015:42     Completed    : 47   Completed    : 11
##  Fall 2015  :26     Current      :  7   Current      :  3
##  Fall 2016  :25     Attempted    :  2   Attempted    :  3
##  Spring 2016:19     Not Attempted:104   Not Attempted:143
##  Fall 2014  :18
##  Spring 2014:17
##  (Other)    :13
##           cse6220             cse6242              cs6250
##  Completed    : 11   Completed    : 13   Completed    :77
##  Current      :  0   Current      : 35   Current      : 7
##  Attempted    :  2   Attempted    :  2   Attempted    : 2
##  Not Attempted:147   Not Attempted:110   Not Attempted:74
##
##
##
##            cs6262              cs6290              cs6300
##  Completed    : 10   Completed    :  8   Completed    :91
##  Current      :  6   Current      :  1   Current      :12
##  Attempted    :  0   Attempted    :  1   Attempted    : 1
##  Not Attempted:144   Not Attempted:150   Not Attempted:56
##
##
##
##            cs6310              cs6340              cs6400
##  Completed    : 22   Completed    : 11   Completed    : 30
##  Current      :  4   Current      :  1   Current      : 10
##  Attempted    :  1   Attempted    :  1   Attempted    :  0
##  Not Attempted:133   Not Attempted:147   Not Attempted:120
##
##
##
##            cs6440              cs6460             cs6475
##  Completed    : 50   Completed    : 7   Completed    : 48
##  Current      :  1   Current      :64   Current      :  6
##  Attempted    :  0   Attempted    : 0   Attempted    :  1
##  Not Attempted:109   Not Attempted:89   Not Attempted:105
##
##
##
##            cs6476              cs6505              cs6601
##  Completed    : 36   Completed    : 28   Completed    : 24
##  Current      :  9   Current      :  2   Current      :  8
##  Attempted    :  3   Attempted    : 14   Attempted    :  5
##  Not Attempted:112   Not Attempted:116   Not Attempted:123
##
##
##
##            cs6750              cs7637             cs7641
##  Completed    :  7   Completed    :89   Completed    :58
##  Current      : 10   Current      : 4   Current      :10
##  Attempted    :  1   Attempted    : 5   Attempted    :13
##  Not Attempted:142   Not Attempted:62   Not Attempted:79
##
##
##
##            cs7646            cse8803            cs8803_001
##  Completed    :50   Completed    :  6   Completed    : 45
##  Current      :12   Current      :  1   Current      :  2
##  Attempted    : 1   Attempted    :  4   Attempted    :  3
##  Not Attempted:97   Not Attempted:149   Not Attempted:110
##
##
##
##          cs8803_002          cs8803_003          cs8803_004
##  Completed    : 16   Completed    : 20   Completed    :  1
##  Current      :  0   Current      :  4   Current      :  0
##  Attempted    :  4   Attempted    :  2   Attempted    :  1
##  Not Attempted:140   Not Attempted:134   Not Attempted:158
##
##
##
##         cs8803_007          cs8803_008  specialization
##  Completed    :  3   Completed    :  1   Length:160
##  Current      :  3   Current      :  2   Class :character
##  Attempted    :  1   Attempted    :  1   Mode  :character
##  Not Attempted:153   Not Attempted:156
##
##
##
##         gpa         hours      prog_years prog_languages
##  2.5 - 2.9 : 2   10 - 14:41   15+    :47   Length:160
##  3.0 - 3.4 :24   15 - 19:36   3 - 5  :31   Class :character
##  3.5 - 3.9 :60   20 - 24:30   1      :21   Mode  :character
##  4         :66   25 - 29:17   2      :18
##  Don't Know: 8   30 - 34:13   12 - 14:14
##                  5 - 9  : 9   6 - 8  :12
##                  (Other):14   (Other):17
##  prior_cs_study prior_cs_exp   conf_prior      conf_post
##  No : 58        0     :33    Min.   :1.000   Min.   :1.000
##  Yes:102        1     :12    1st Qu.:3.000   1st Qu.:4.000
##                 2     :23    Median :4.000   Median :5.000
##                 3 - 5 :22    Mean   :3.844   Mean   :4.369
##                 6 - 8 :13    3rd Qu.:5.000   3rd Qu.:5.000
##                 9 - 11:17    Max.   :5.000   Max.   :5.000
##                 12+   :40
##    selfconf1      selfconf2       selfconf3     selfconf4
##  Min.   :1.00   Min.   :2.000   Min.   :1.0   Min.   :1.000
##  1st Qu.:4.00   1st Qu.:4.000   1st Qu.:3.0   1st Qu.:3.000
##  Median :4.00   Median :5.000   Median :4.0   Median :4.000
##  Mean   :4.15   Mean   :4.475   Mean   :3.8   Mean   :3.888
##  3rd Qu.:5.00   3rd Qu.:5.000   3rd Qu.:4.0   3rd Qu.:5.000
##  Max.   :5.00   Max.   :5.000   Max.   :5.0   Max.   :5.000
##
##    selfconf5       selfconf6       selfconf7       equality1
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:4.000   1st Qu.:3.000   1st Qu.:4.000   1st Qu.:4.000
##  Median :4.000   Median :4.000   Median :4.000   Median :5.000
##  Mean   :4.275   Mean   :3.844   Mean   :4.162   Mean   :4.519
##  3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:5.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##
##    equality2       equality3       equality4       equality5
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:5.000   1st Qu.:5.000   1st Qu.:5.000   1st Qu.:2.000
##  Median :5.000   Median :5.000   Median :5.000   Median :3.000
##  Mean   :4.662   Mean   :4.731   Mean   :4.719   Mean   :3.306
##  3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:5.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
```

```
##
##    equality6       belonging1      belonging2    belonging3
##  Min.   :1.000   Min.   :1.000   Min.   :1.0   Min.   :1.000
##  1st Qu.:4.000   1st Qu.:3.000   1st Qu.:3.0   1st Qu.:4.000
##  Median :5.000   Median :4.000   Median :4.0   Median :4.000
##  Mean   :4.537   Mean   :3.694   Mean   :3.8   Mean   :4.069
##  3rd Qu.:5.000   3rd Qu.:4.000   3rd Qu.:5.0   3rd Qu.:5.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.0   Max.   :5.000
##
##    belonging4      belonging5      belonging6   belonging7
##  Min.   :1.000   Min.   :1.000   Min.   :1    Min.   :1.000
##  1st Qu.:2.000   1st Qu.:3.000   1st Qu.:3    1st Qu.:4.000
##  Median :3.000   Median :3.500   Median :4    Median :5.000
##  Mean   :3.006   Mean   :3.356   Mean   :4    Mean   :4.531
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:5    3rd Qu.:5.000
##  Max.   :5.000   Max.   :5.000   Max.   :5    Max.   :5.000
##
##    belonging8     belonging9      completed        attempted
##  Min.   :1.00   Min.   :1.000   Min.   : 0.000   Min.   : 0.0000
##  1st Qu.:4.00   1st Qu.:4.000   1st Qu.: 3.000   1st Qu.: 0.0000
##  Median :4.00   Median :5.000   Median : 5.000   Median : 0.0000
##  Mean   :4.05   Mean   :4.181   Mean   : 5.125   Mean   : 0.4625
##  3rd Qu.:5.00   3rd Qu.:5.000   3rd Qu.: 7.000   3rd Qu.: 1.0000
##  Max.   :5.00   Max.   :5.000   Max.   :11.000   Max.   :14.0000
##
##     current       conf_ave        age_num         gpa_num
##  Min.   :0.0   Min.   :1.500   Min.   :22.00   Min.   :2.700
##  1st Qu.:1.0   1st Qu.:3.500   1st Qu.:27.00   1st Qu.:3.700
##  Median :1.0   Median :4.250   Median :32.00   Median :3.700
##  Mean   :1.4   Mean   :4.106   Mean   :35.88   Mean   :3.738
##  3rd Qu.:2.0   3rd Qu.:4.500   3rd Qu.:42.00   3rd Qu.:4.000
##  Max.   :7.0   Max.   :5.000   Max.   :62.00   Max.   :4.000
##                                                NA's   :8
##    hours_num        prog_num       prior_cs_num      prog_count
##  Min.   : 2.00   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:12.00   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 3.000
##  Median :17.00   Median : 7.000   Median : 4.000   Median : 5.000
##  Mean   :20.19   Mean   : 8.244   Mean   : 5.794   Mean   : 4.694
##  3rd Qu.:27.00   3rd Qu.:16.000   3rd Qu.:10.750   3rd Qu.: 6.000
##  Max.   :42.00   Max.   :16.000   Max.   :13.000   Max.   :14.000
##
##  spec_robotics     spec_systems    spec_intelligence    spec_ml
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000    Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000    1st Qu.:0.000
##  Median :0.0000   Median :0.0000   Median :0.0000    Median :0.000
##  Mean   :0.1062   Mean   :0.1562   Mean   :0.4875    Mean   :0.275
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:1.0000    3rd Qu.:1.000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000    Max.   :1.000
##
##  reason_financial  reason_quals    reason_promotion reason_switch
##  Min.   :0.000    Min.   :0.0000   Min.   :0.0000   Min.   :0.0
##  1st Qu.:0.000    1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0
##  Median :0.000    Median :1.0000   Median :0.0000   Median :0.0
##  Mean   :0.375    Mean   :0.6562   Mean   :0.3063   Mean   :0.3
##  3rd Qu.:1.000    3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0  
##  Max.   :1.000    Max.   :1.0000   Max.   :1.0000   Max.   :1.0  
##                                                                  
##  reason_connections   reason_fun      cs_study_ind      native_ind  
##  Min.   :0.0000     Min.   :0.0000   Min.   :0.0000   Min.   :0.00  
##  1st Qu.:0.0000     1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00  
##  Median :0.0000     Median :0.0000   Median :1.0000   Median :1.00  
##  Mean   :0.1187     Mean   :0.4062   Mean   :0.6375   Mean   :0.65  
##  3rd Qu.:0.0000     3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.00  
##  Max.   :1.0000     Max.   :1.0000   Max.   :1.0000   Max.   :1.00  
##                                                                     
##   us_birth_ind     us_res_ind       higher_ind     selfconf_score 
##  Min.   :0.000   Min.   :0.0000   Min.   :0.0000   Min.   :1.857  
##  1st Qu.:0.000   1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:3.714  
##  Median :1.000   Median :1.0000   Median :0.0000   Median :4.143  
##  Mean   :0.575   Mean   :0.8688   Mean   :0.2938   Mean   :4.085  
##  3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:4.607  
##  Max.   :1.000   Max.   :1.0000   Max.   :1.0000   Max.   :5.000  
##                                                                   
##  equality_score  belonging_score
##  Min.   :1.000   Min.   :1.000  
##  1st Qu.:4.167   1st Qu.:3.333  
##  Median :4.500   Median :3.944  
##  Mean   :4.412   Mean   :3.854  
##  3rd Qu.:4.833   3rd Qu.:4.333  
##  Max.   :5.000   Max.   :5.000  
## 
```

```r
# Explore outlier cases for completed, current and attempted
subset(survey[,c("completed", "current", "attempted")], completed > 10)
```

```
##     completed current attempted
## 37         11       0         0
## 154        11       2         0
```

```r
subset(survey[,c("completed", "current", "attempted")], current > 3)
```

```
##     completed current attempted
## 114         5       4         0
## 137         7       7        14
```

```r
subset(survey[,c("completed", "current", "attempted")], attempted > 5)
```

```
##     completed current attempted
## 137         7       7        14
```

```r
# Change outlier values to medians
survey$current = ifelse(survey$current > 3, 1, survey$current)
survey$attempted = ifelse(survey$attempted == 14, 0, survey$attempted)

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

```
## 
##  Female    Male 
## 0.35625 0.64375
```

```
# Calculate proportion of respondent who are currently enrolled in CS6460 (EduTech)
count(subset(survey, cs6460 == "Current"), "cs6460")$freq/dim(survey)[1]
```

```
## [1] 0.4
```

Analyze Data by Gender

```
# Calculate confidence summary statistics
ddply(survey, "gender", summarise,
    mean = mean(conf_prior), sd = sd(conf_prior), median = median(conf_prior),
    first_q = quantile(conf_prior, 0.25), third_q = quantile(conf_prior, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 3.315789 1.1363773      3       2       4
## 2   Male 4.135922 0.8289912      4       4       5
```

```
ddply(survey, "gender", summarise,
    mean = mean(conf_post), sd = sd(conf_post), median = median(conf_post),
    first_q = quantile(conf_post, 0.25), third_q = quantile(conf_post, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 4.175439 0.9281858      4       4       5
## 2   Male 4.475728 0.7776871      5       4       5
```

```
ddply(survey, "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 3.745614 0.8080113    4.0       3     4.5
## 2   Male 4.305825 0.6868264    4.5       4     5.0
```

```
# Calculate summary stats for self confidence statements
ddply(survey, "gender", summarise,
    mean = mean(selfconf1), sd = sd(selfconf1), median = median(selfconf1),
    first_q = quantile(selfconf1, 0.25), third_q = quantile(selfconf1, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 3.877193 0.9077086      4       3       4
## 2   Male 4.300971 0.8264613      4       4       5
```

```
ddply(survey, "gender", summarise,
    mean = mean(selfconf2), sd = sd(selfconf2), median = median(selfconf2),
    first_q = quantile(selfconf2, 0.25), third_q = quantile(selfconf2, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 4.315789 0.6855106      4       4       5
## 2   Male 4.563107 0.5886159      5       4       5
```

```
ddply(survey, "gender", summarise,
    mean = mean(selfconf3), sd = sd(selfconf3), median = median(selfconf3),
    first_q = quantile(selfconf3, 0.25), third_q = quantile(selfconf3, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 3.578947 0.9626483      4       3       4
## 2   Male 3.922330 0.8932132      4       3       5
```

```r
ddply(survey, "gender", summarise,
    mean = mean(selfconf4), sd = sd(selfconf4), median = median(selfconf4),
    first_q = quantile(selfconf4, 0.25), third_q = quantile(selfconf4, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 3.771930 0.8241347      4       3       4
## 2   Male 3.951456 0.8674805      4       3       5
```

```r
ddply(survey, "gender", summarise,
    mean = mean(selfconf5), sd = sd(selfconf5), median = median(selfconf5),
    first_q = quantile(selfconf5, 0.25), third_q = quantile(selfconf5, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 4.105263 0.7484319      4       4       5
## 2   Male 4.368932 0.7408301      4       4       5
```

```r
ddply(survey, "gender", summarise,
    mean = mean(selfconf6), sd = sd(selfconf6), median = median(selfconf6),
    first_q = quantile(selfconf6, 0.25), third_q = quantile(selfconf6, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 3.280702 1.0308524      3       2       4
## 2   Male 4.155340 0.8939588      4       4       5
```

```r
ddply(survey, "gender", summarise,
    mean = mean(selfconf7), sd = sd(selfconf7), median = median(selfconf7),
    first_q = quantile(selfconf7, 0.25), third_q = quantile(selfconf7, 0.75))
```

```
##   gender     mean       sd median first_q third_q
## 1 Female 4.017544 1.008734      4       3       5
## 2   Male 4.242718 1.023891      5       4       5
```

```r
ddply(survey, "gender", summarise,
    mean = mean(selfconf_score), sd = sd(selfconf_score), median = median(selfconf_score),
    first_q = quantile(selfconf_score, 0.25), third_q = quantile(selfconf_score, 0.75))
```

```
##   gender     mean        sd   median  first_q  third_q
## 1 Female 3.849624 0.6609809 3.857143 3.428571 4.428571
## 2   Male 4.214979 0.6109819 4.285714 3.857143 4.714286
```

```r
# Calculate summary statistics for equality statements
ddply(survey, "gender", summarise,
    mean = mean(equality1), sd = sd(equality1), median = median(equality1),
    first_q = quantile(equality1, 0.25), third_q = quantile(equality1, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 4.649123 0.6121166      5       4       5
## 2   Male 4.446602 0.8369899      5       4       5
```

```r
ddply(survey, "gender", summarise,
    mean = mean(equality2), sd = sd(equality2), median = median(equality2),
    first_q = quantile(equality2, 0.25), third_q = quantile(equality2, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 4.684211 0.5718983      5       4       5
## 2   Male 4.650485 0.7885049      5       5       5
```

```r
ddply(survey, "gender", summarise,
    mean = mean(equality3), sd = sd(equality3), median = median(equality3),
    first_q = quantile(equality3, 0.25), third_q = quantile(equality3, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 4.842105 0.4135851      5       5       5
## 2   Male 4.669903 0.7845112      5       5       5
```

```
ddply(survey, "gender", summarise,
    mean = mean(equality4), sd = sd(equality4), median = median(equality4),
    first_q = quantile(equality4, 0.25), third_q = quantile(equality4, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 4.842105 0.4135851      5       5       5
## 2   Male 4.650485 0.8483980      5       5       5
```

```
ddply(survey, "gender", summarise,
    mean = mean(equality5), sd = sd(equality5), median = median(equality5),
    first_q = quantile(equality5, 0.25), third_q = quantile(equality5, 0.75))
```

```
##   gender     mean       sd median first_q third_q
## 1 Female 3.438596 1.414435      4       2     5.0
## 2   Male 3.233010 1.300105      3       2     4.5
```

```
ddply(survey, "gender", summarise,
    mean = mean(equality6), sd = sd(equality6), median = median(equality6),
    first_q = quantile(equality6, 0.25), third_q = quantile(equality6, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 4.649123 0.6941394      5       4       5
## 2   Male 4.475728 0.9685714      5       4       5
```

```
ddply(survey, "gender", summarise,
    mean = mean(equality_score), sd = sd(equality_score), median = median(equality_score),
    first_q = quantile(equality_score, 0.25), third_q = quantile(equality_score, 0.75))
```

```
##   gender     mean        sd   median  first_q  third_q
## 1 Female 4.517544 0.4721239 4.666667 4.333333 5.000000
## 2   Male 4.354369 0.6774775 4.500000 4.083333 4.833333
```

```
# Calculate summary statistics for belonging statements
ddply(survey, "gender", summarise,
    mean = mean(belonging1), sd = sd(belonging1), median = median(belonging1),
    first_q = quantile(belonging1, 0.25), third_q = quantile(belonging1, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 3.333333 1.0745985      3       3       4
## 2   Male 3.893204 0.8846471      4       3       4
```

```
ddply(survey, "gender", summarise,
    mean = mean(belonging2), sd = sd(belonging2), median = median(belonging2),
    first_q = quantile(belonging2, 0.25), third_q = quantile(belonging2, 0.75))
```

```
##   gender     mean       sd median first_q third_q
## 1 Female 3.456140 1.310264      4       2       5
## 2   Male 3.990291 1.184004      4       3       5
```

```
ddply(survey, "gender", summarise,
    mean = mean(belonging3), sd = sd(belonging3), median = median(belonging3),
    first_q = quantile(belonging3, 0.25), third_q = quantile(belonging3, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 3.684211 1.0716792      4       3       4
## 2   Male 4.281553 0.7848751      4       4       5
```

```
ddply(survey, "gender", summarise,
    mean = mean(belonging4), sd = sd(belonging4), median = median(belonging4),
    first_q = quantile(belonging4, 0.25), third_q = quantile(belonging4, 0.75))
```

```
##   gender     mean       sd median first_q third_q
## 1 Female 2.842105 1.264762      3       2       4
## 2   Male 3.097087 1.332214      3       2       4
```

```
ddply(survey, "gender", summarise,
    mean = mean(belonging5), sd = sd(belonging5), median = median(belonging5),
    first_q = quantile(belonging5, 0.25), third_q = quantile(belonging5, 0.75))
```

```
##   gender     mean       sd median first_q third_q
## 1 Female 3.157895 1.177004      3       2       4
## 2   Male 3.466019 1.017644      4       3       4
```

```
ddply(survey, "gender", summarise,
    mean = mean(belonging6), sd = sd(belonging6), median = median(belonging6),
    first_q = quantile(belonging6, 0.25), third_q = quantile(belonging6, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 3.543860 1.2687195      4       2       5
## 2   Male 4.252427 0.9972359      5       4       5
```

```
ddply(survey, "gender", summarise,
    mean = mean(belonging7), sd = sd(belonging7), median = median(belonging7),
    first_q = quantile(belonging7, 0.25), third_q = quantile(belonging7, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 4.473684 0.8885233      5       4       5
## 2   Male 4.563107 0.7231454      5       4       5
```

```
ddply(survey, "gender", summarise,
    mean = mean(belonging8), sd = sd(belonging8), median = median(belonging8),
    first_q = quantile(belonging8, 0.25), third_q = quantile(belonging8, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 3.842105 1.0314600      4       3       5
## 2   Male 4.165049 0.9192611      4       4       5
```

```
ddply(survey, "gender", summarise,
    mean = mean(belonging9), sd = sd(belonging9), median = median(belonging9),
    first_q = quantile(belonging9, 0.25), third_q = quantile(belonging9, 0.75))
```

```
##   gender     mean       sd median first_q third_q
## 1 Female 3.947368 1.216274      4       3       5
## 2   Male 4.310680 1.057458      5       4       5
```

```
ddply(survey, "gender", summarise,
    mean = mean(belonging_score), sd = sd(belonging_score),
    median = median(belonging_score),
    first_q = quantile(belonging_score, 0.25), third_q = quantile(belonging_score, 0.75))
```

```
##   gender     mean        sd   median  first_q  third_q
## 1 Female 3.586745 0.7794071 3.555556 3.111111 4.111111
## 2   Male 4.002157 0.5997620 4.111111 3.555556 4.444444
```

```r
# Calculate age summary statistics
ddply(survey, "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 34.71930 9.593067     32      27      37
## 2   Male 36.51456 9.432801     37      27      42
```

```r
# Calculate gpa summary statistics
ddply(subset(survey, !is.na(gpa_num)), "gender", summarise, mean = mean(gpa_num),
        sd = sd(gpa_num), median = median(gpa_num), first_q =
        quantile(gpa_num, 0.25), third_q = quantile(gpa_num, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 3.772222 0.3217972    4.0     3.7       4
## 2   Male 3.719388 0.2895635    3.7     3.7       4
```

```r
# Calculate study hours summary statistics
ddply(survey, "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 21.03509 9.084848     22      12      27
## 2   Male 19.71845 9.095805     17      12      22
```

```r
# Calculate programming years summary statistics
ddply(survey, "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 6.192982 5.648763      4       2      10
## 2   Male 9.378641 6.057108     10       4      16
```

```r
# Calculate prior cs experience summary statistics
ddply(survey, "gender", summarise, mean = mean(prior_cs_num), sd = sd(prior_cs_num),
      median = median(prior_cs_num), first_q = quantile(prior_cs_num, 0.25),
      third_q = quantile(prior_cs_num, 0.75))
```

```
##   gender     mean       sd median first_q third_q
## 1 Female 4.210526 4.857751      2       0       7
## 2   Male 6.669903 5.086291      4       2      13
```

```r
# Calculate programming language count summary statistics
ddply(survey, "gender", summarise, mean = mean(prog_count), sd = sd(prog_count),
      median = median(prog_count), first_q = quantile(prog_count, 0.25),
      third_q = quantile(prog_count, 0.75))
```

```
##   gender     mean       sd median first_q third_q
## 1 Female 4.105263 2.335168      4       2     5.0
## 2   Male 5.019417 2.585710      5       3     6.5
```

```r
# Calculate course completion/attempt summary statistics
ddply(survey, "gender", summarise, mean = mean(completed), sd = sd(completed),
      median = median(completed), first_q = quantile(completed, 0.25),
      third_q = quantile(completed, 0.75))
```

```
##   gender     mean       sd median first_q third_q
## 1 Female 4.649123 2.831527      4       3       7
## 2   Male 5.388350 2.690699      6       4       7
```

```r
ddply(survey, "gender", summarise, mean = mean(attempted), sd = sd(attempted),
      median = median(attempted), first_q = quantile(attempted, 0.25),
      third_q = quantile(attempted, 0.75))
```

```
##   gender      mean        sd median first_q third_q
## 1 Female 0.3333333 0.7867958      0       0       0
## 2   Male 0.3980583 0.7963127      0       0       1
```

```r
ddply(survey, "gender", summarise, mean = mean(current), sd = sd(current),
      median = median(current), first_q = quantile(current, 0.25),
      third_q = quantile(current, 0.75))
```

```
##   gender     mean        sd median first_q third_q
## 1 Female 1.228070 0.8241347      1       1       2
## 2   Male 1.407767 0.7199795      1       1       2
```

```r
survey_m = subset(survey, gender == "Male")
survey_f = subset(survey, gender == "Female")

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

```
## 
##    20 - 24    25 - 29    30 - 34    35 - 39    40 - 44    45 - 49 
## 0.06796117 0.19417476 0.23300971 0.16504854 0.11650485 0.09708738 
##    50 - 54    55 - 59    60 - 64 
## 0.08737864 0.03883495 0.00000000
```

```r
prop.table(table(survey_f$age))
```

```
## 
##    20 - 24    25 - 29    30 - 34    35 - 39    40 - 44    45 - 49 
## 0.07017544 0.28070175 0.24561404 0.19298246 0.03508772 0.08771930 
##    50 - 54    55 - 59    60 - 64 
## 0.01754386 0.05263158 0.01754386
```

```r
# Compare birth country
prop.table(table(survey_m$birth))
```

```
## 
##      Australia         Brazil         Canada          Chile          China 
##    0.000000000    0.009708738    0.009708738    0.009708738    0.067961165 
##       Colombia        Ecuador         Greece         Guyana        Holland 
##    0.000000000    0.009708738    0.000000000    0.009708738    0.009708738 
##      Hong Kong          India      Indonesia        Ireland          Kenya 
##    0.009708738    0.145631068    0.029126214    0.009708738    0.009708738 
##       Malaysia         Mexico        Myanmar       Pakistan           Peru 
##    0.009708738    0.009708738    0.009708738    0.009708738    0.009708738 
##    Philippines         Russia      Singapore    South Korea Southeast Asia 
##    0.000000000    0.009708738    0.009708738    0.009708738    0.000000000 
##       Thailand        Ukraine United Kingdom            USA     Yugoslavia 
##    0.000000000    0.009708738    0.009708738    0.563106796    0.009708738
```

```r
prop.table(table(survey_f$birth))
```

```
##
##      Australia         Brazil         Canada          Chile          China
##     0.01754386     0.00000000     0.00000000     0.00000000     0.08771930
##       Colombia        Ecuador         Greece         Guyana        Holland
##     0.01754386     0.01754386     0.01754386     0.00000000     0.00000000
##      Hong Kong          India      Indonesia        Ireland          Kenya
##     0.00000000     0.17543860     0.00000000     0.00000000     0.00000000
##       Malaysia         Mexico        Myanmar       Pakistan           Peru
##     0.00000000     0.01754386     0.00000000     0.00000000     0.00000000
##    Philippines         Russia      Singapore    South Korea Southeast Asia
##     0.01754386     0.00000000     0.00000000     0.00000000     0.01754386
##       Thailand        Ukraine United Kingdom            USA     Yugoslavia
##     0.01754386     0.00000000     0.00000000     0.59649123     0.00000000
```

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

```
##
##   Australia      Brazil      Canada       China     Ecuador       India
## 0.000000000 0.009708738 0.058252427 0.009708738 0.009708738 0.009708738
##      Israel       Japan       Kenya    Paraguay   Singapore    Thailand
## 0.000000000 0.000000000 0.009708738 0.009708738 0.019417476 0.000000000
##         USA
## 0.864077670
```

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

```
##
##  Australia     Brazil     Canada      China    Ecuador      India
## 0.01754386 0.00000000 0.01754386 0.00000000 0.00000000 0.03508772
##     Israel      Japan      Kenya   Paraguay  Singapore   Thailand
## 0.01754386 0.01754386 0.00000000 0.00000000 0.00000000 0.01754386
##        USA
## 0.87719298
```

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

```
##
##        Burmese        Chinese        English         French         Indian
##    0.009708738    0.087378641    0.660194175    0.009708738    0.077669903
##     Indonesian         Korean      Malayalam     Portuguese        Russian
##    0.029126214    0.019417476    0.000000000    0.009708738    0.019417476
## Serbo-Croatian        Spanish        Swahili        Tagalog           Thai
##    0.009708738    0.048543689    0.009708738    0.000000000    0.000000000
##           Urdu
##    0.009708738
```

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

```
##
##        Burmese        Chinese        English         French         Indian
##     0.00000000     0.14035088     0.63157895     0.00000000     0.12280702
##     Indonesian         Korean      Malayalam     Portuguese        Russian
##     0.00000000     0.00000000     0.01754386     0.00000000     0.00000000
## Serbo-Croatian        Spanish        Swahili        Tagalog           Thai
##     0.00000000     0.05263158     0.00000000     0.01754386     0.01754386
##           Urdu
##     0.00000000
```

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

```
## numeric(0)
```

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

```
## numeric(0)
```

```
# Compare education level
prop.table(table(survey_m$education_level))
```

```
##
##  Bachelors    Masters  Doctorate
## 0.72815534 0.22330097 0.04854369
```

```
prop.table(table(survey_f$education_level))
```

```
##
##  Bachelors    Masters  Doctorate
## 0.66666667 0.26315789 0.07017544
```

```
# Compare education field
prop.table(table(survey_m$education_field))
```

```
##
##       Computer science     Economics/business            Engineering
##            0.485436893            0.038834951            0.320388350
##        Humanities/arts Mathematics/statistics         Other sciences
##            0.029126214            0.038834951            0.009708738
## Physical/life sciences
##            0.077669903
```

```
prop.table(table(survey_f$education_field))
```

```
##
##       Computer science     Economics/business            Engineering
##             0.40350877             0.01754386             0.29824561
##        Humanities/arts Mathematics/statistics         Other sciences
##             0.05263158             0.14035088             0.03508772
## Physical/life sciences
##             0.05263158
```

```
# Compare omscs_yn
prop.table(table(survey_m$omscs_yn))
```

```
##
##          No         Yes
## 0.009708738 0.990291262
```

```
prop.table(table(survey_f$omscs_yn))
```

```
##
##        No       Yes
## 0.1052632 0.8947368
```

```
# Compare OMSCS commencement semester
prop.table(table(survey_m$omscs_semester))
```

```
##
## Spring 2014 Summer 2014   Fall 2014 Spring 2015 Summer 2015   Fall 2015
##  0.12621359  0.03883495  0.09708738  0.25242718  0.00000000  0.20388350
## Spring 2016 Summer 2016   Fall 2016 Spring 2017
##  0.10679612  0.00000000  0.13592233  0.03883495
```

```
prop.table(table(survey_f$omscs_semester))
```

```
##
## Spring 2014 Summer 2014   Fall 2014 Spring 2015 Summer 2015   Fall 2015
##  0.07017544  0.05263158  0.14035088  0.28070175  0.00000000  0.08771930
## Spring 2016 Summer 2016   Fall 2016 Spring 2017
##  0.14035088  0.00000000  0.19298246  0.03508772
```

```
# Compare prior cs study ind
prop.table(table(survey_m$prior_cs_study))
```

```
##
##        No       Yes
## 0.2815534 0.7184466
```

```
prop.table(table(survey_f$prior_cs_study))
```

```
##
##        No       Yes
## 0.5087719 0.4912281
```

```
# Compare prior cs experience
prop.table(table(survey_m$prior_cs_exp))
```

```
##
##          0          1          2      3 - 5      6 - 8     9 - 11
## 0.12621359 0.06796117 0.14563107 0.16504854 0.06796117 0.12621359
##        12+
## 0.30097087
```

```
prop.table(table(survey_f$prior_cs_exp))
```

```
##
##          0          1          2      3 - 5      6 - 8     9 - 11
## 0.35087719 0.08771930 0.14035088 0.08771930 0.10526316 0.07017544
##        12+
## 0.15789474
```

```
# Compare specializations (including current EduTech students)
prop.table(table(survey_m$spec_robotics))
```

```
##
##         0         1
## 0.8834951 0.1165049
```

```
prop.table(table(survey_f$spec_robotics))
```

```
##
##         0         1
## 0.9122807 0.0877193
```

```r
prop.table(table(survey_m$spec_systems))
```

```
##
##         0         1
## 0.8446602 0.1553398
```

```r
prop.table(table(survey_f$spec_systems))
```

```
##
##         0         1
## 0.8421053 0.1578947
```

```r
prop.table(table(survey_m$spec_intelligence))
```

```
##
##         0         1
## 0.4854369 0.5145631
```

```r
prop.table(table(survey_f$spec_intelligence))
```

```
##
##         0         1
## 0.5614035 0.4385965
```

```r
prop.table(table(survey_m$spec_ml))
```

```
##
##        0        1
## 0.776699 0.223301
```

```r
prop.table(table(survey_f$spec_ml))
```

```
##
##         0         1
## 0.6315789 0.3684211
```

```r
# Compare specializations (excluding current EduTech students)
prop.table(table(subset(survey_m, cs6460 != "Current")$spec_robotics))
```

```
##
##         0         1
## 0.7884615 0.2115385
```

```r
prop.table(table(subset(survey_f, cs6460 != "Current")$spec_robotics))
```

```
##
##         0         1
## 0.8863636 0.1136364
```

```r
prop.table(table(subset(survey_m, cs6460 != "Current")$spec_systems))
```

```
##
##         0         1
## 0.8076923 0.1923077
```

```r
prop.table(table(subset(survey_f, cs6460 != "Current")$spec_systems))
```

```
##
##         0         1
## 0.7954545 0.2045455
```

```
prop.table(table(subset(survey_m, cs6460 != "Current")$spec_intelligence))
```

```
##
##         0         1
## 0.8076923 0.1923077
```

```
prop.table(table(subset(survey_f, cs6460 != "Current")$spec_intelligence))
```

```
##
##         0         1
## 0.7045455 0.2954545
```

```
prop.table(table(subset(survey_m, cs6460 != "Current")$spec_ml))
```

```
##
##         0         1
## 0.5769231 0.4230769
```

```
prop.table(table(subset(survey_f, cs6460 != "Current")$spec_ml))
```

```
##
##         0         1
## 0.5454545 0.4545455
```

```
# Compare reasons
prop.table(table(survey_m$reason_financial))
```

```
##
##         0         1
## 0.6019417 0.3980583
```

```
prop.table(table(survey_f$reason_financial))
```

```
##
##         0         1
## 0.6666667 0.3333333
```

```
prop.table(table(survey_m$reason_quals))
```

```
##
##         0         1
## 0.3592233 0.6407767
```

```
prop.table(table(survey_f$reason_quals))
```

```
##
##         0         1
## 0.3157895 0.6842105
```

```
prop.table(table(survey_m$reason_promotion))
```

```
##
##         0         1
## 0.6699029 0.3300971
```

```
prop.table(table(survey_f$reason_promotion))
```

```
##
##         0         1
## 0.7368421 0.2631579
```

```r
prop.table(table(survey_m$reason_switch))
```

```
##
##         0         1
## 0.6893204 0.3106796
```

```r
prop.table(table(survey_f$reason_switch))
```

```
##
##         0         1
## 0.7192982 0.2807018
```

```r
prop.table(table(survey_m$reason_connections))
```

```
##
##         0         1
## 0.8932039 0.1067961
```

```r
prop.table(table(survey_f$reason_connections))
```

```
##
##         0         1
## 0.8596491 0.1403509
```

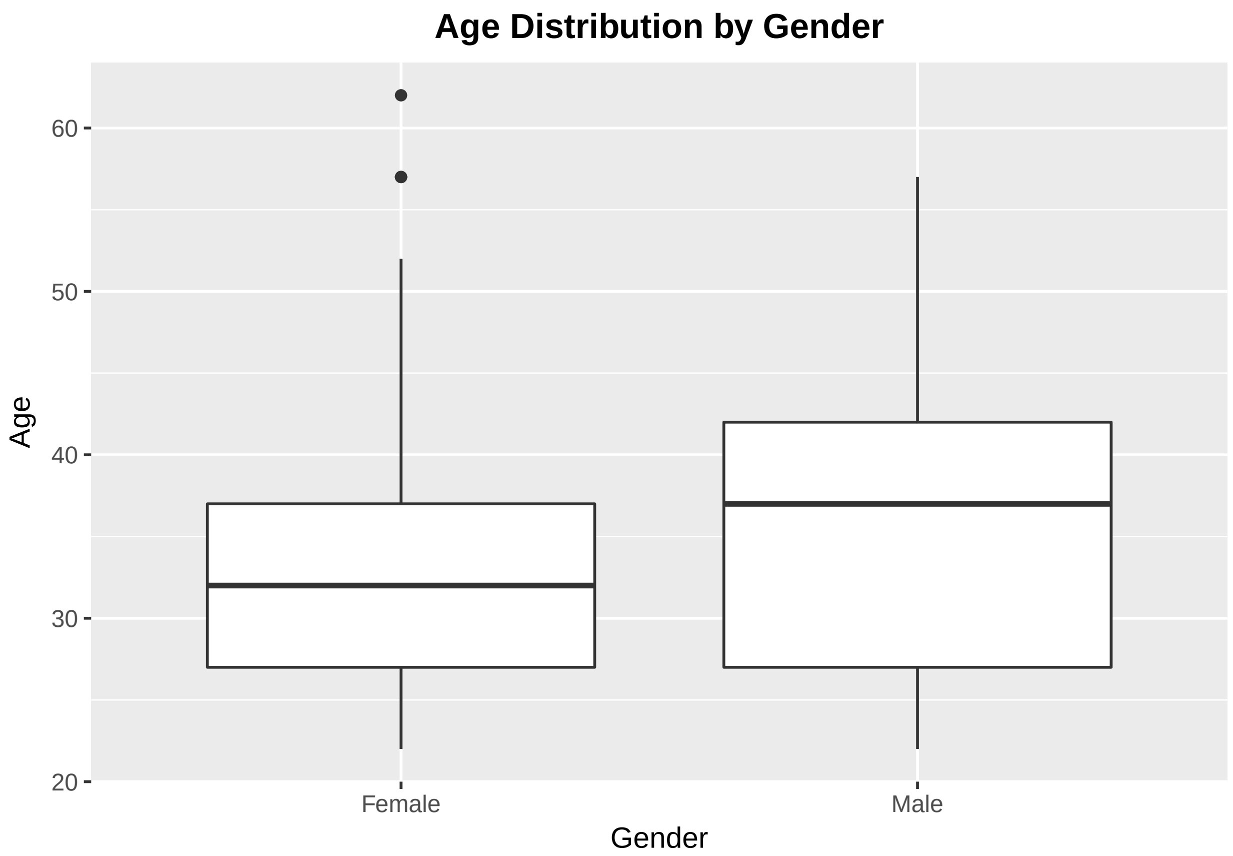
```r
prop.table(table(survey_m$reason_fun))
```

```
##
##         0         1
## 0.6213592 0.3786408
```

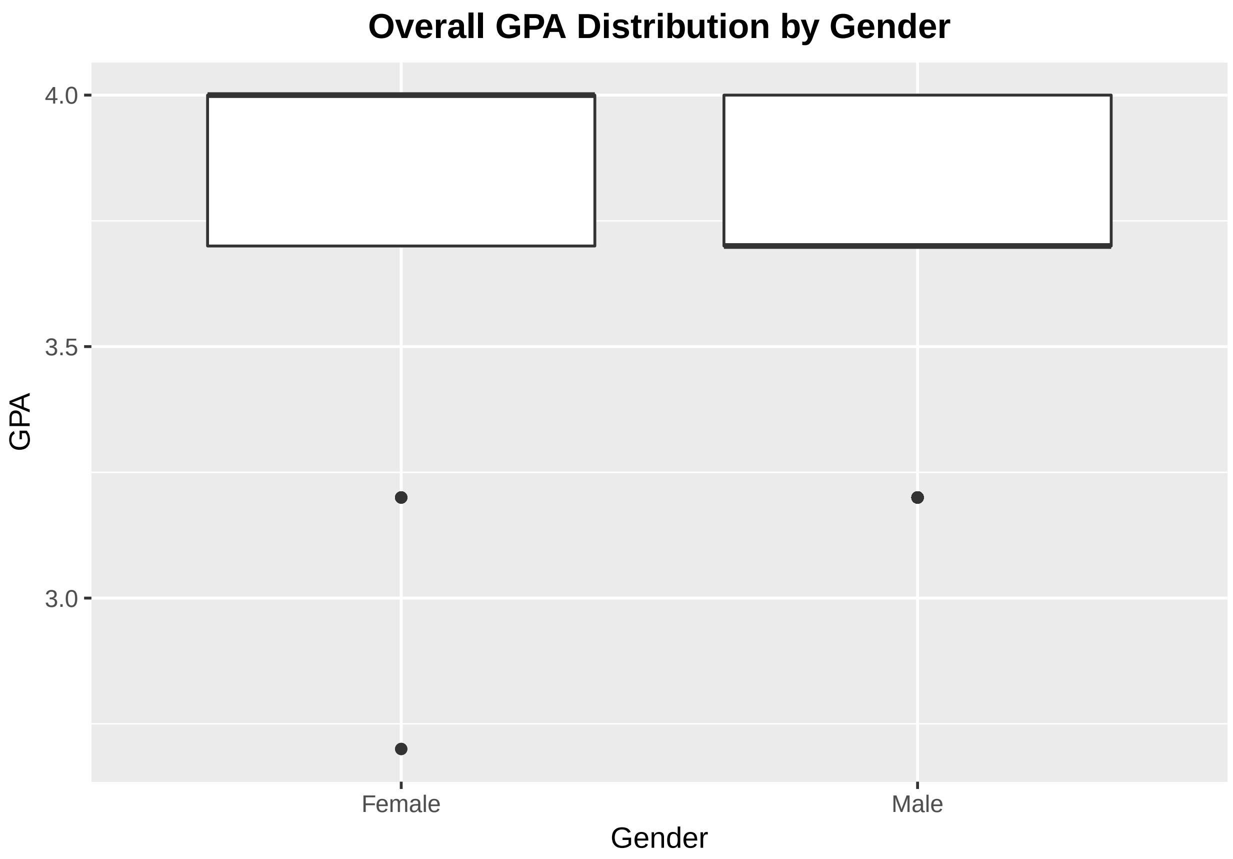
```r
prop.table(table(survey_f$reason_fun))
```

```
##
##         0         1
## 0.5438596 0.4561404
```

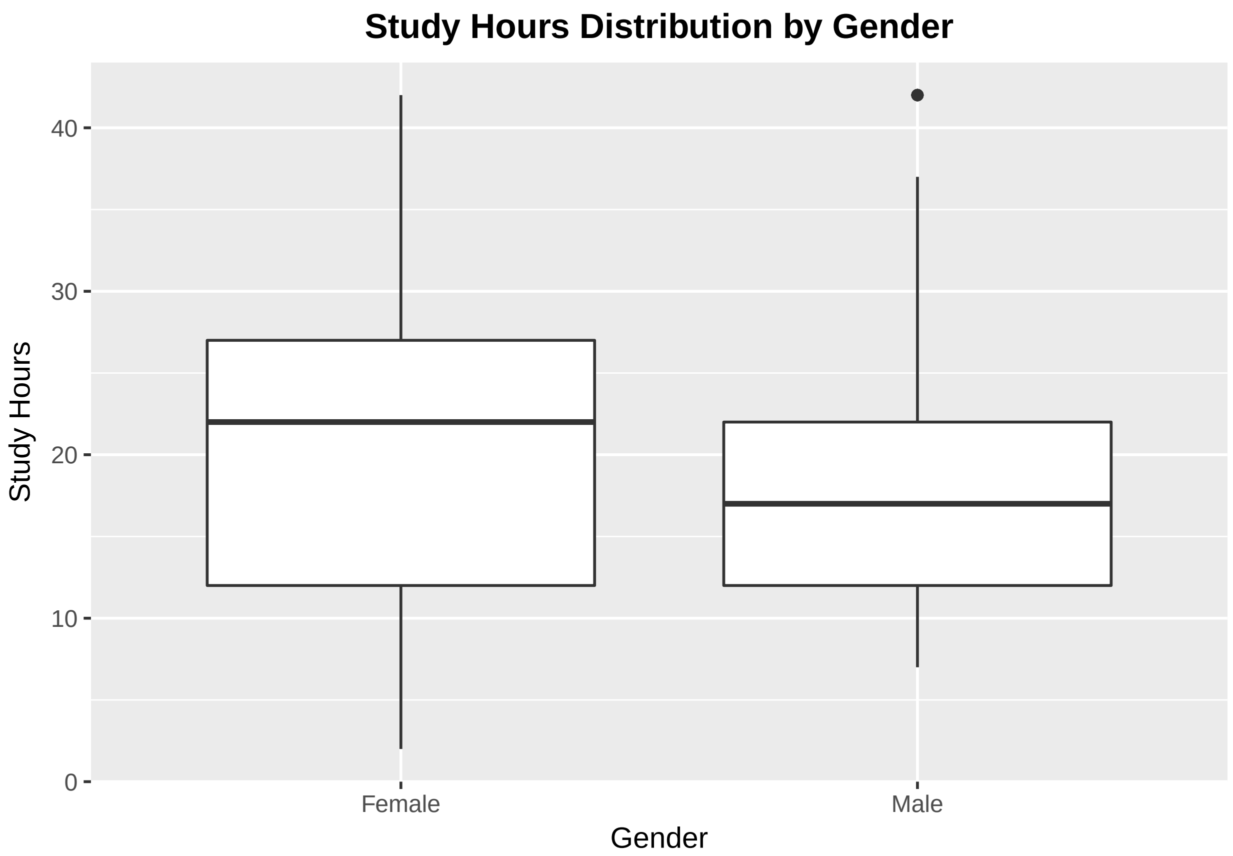
```r
#Boxplot of age distribution by gender
ggplot(survey, aes(gender, age_num)) +
 geom_boxplot() +
 labs(title = "Age Distribution by Gender",
      x = "Gender", y = "Age") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



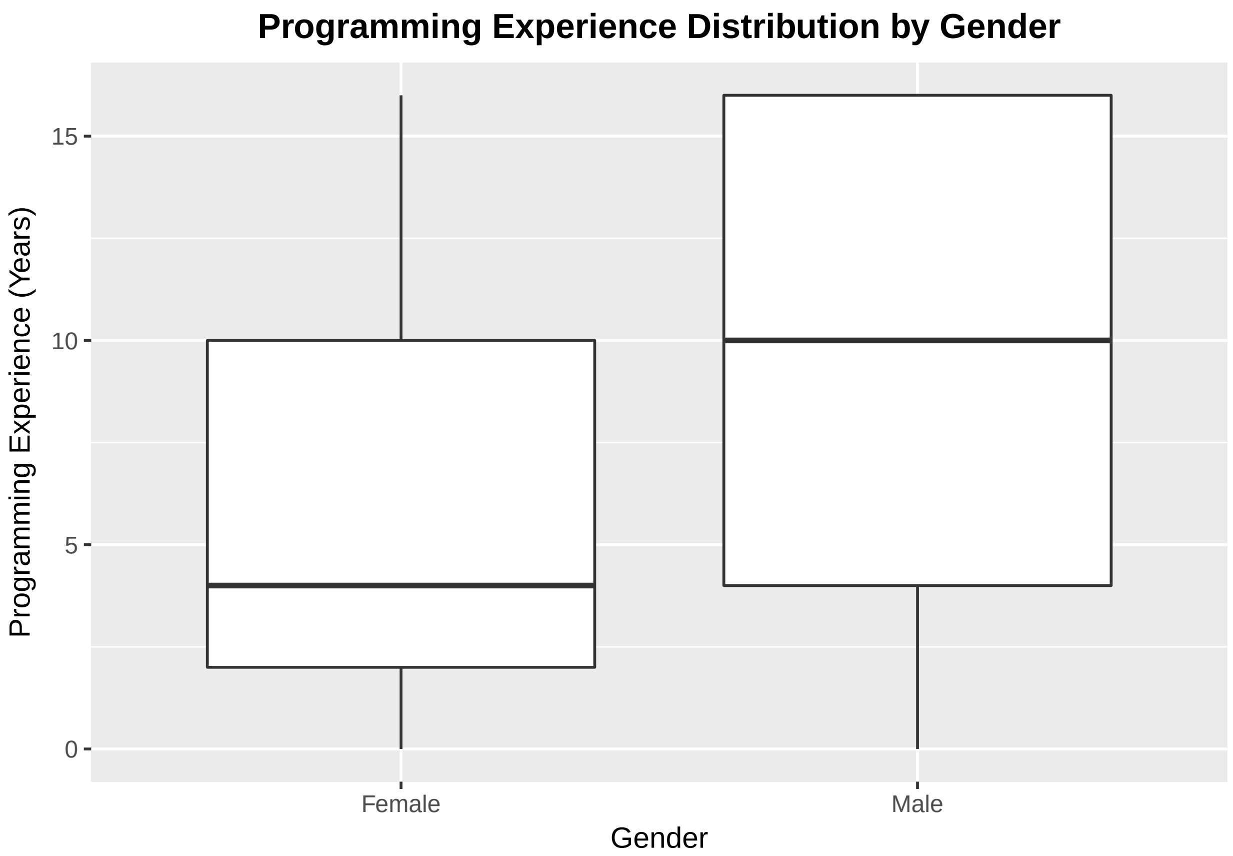
```r
# Boxplot of gpa distribution by gender
ggplot(subset(survey, !is.na(gpa_num)), aes(gender, gpa_num)) +
 geom_boxplot() +
 labs(title = "Overall GPA Distribution by Gender",
      x = "Gender", y = "GPA") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



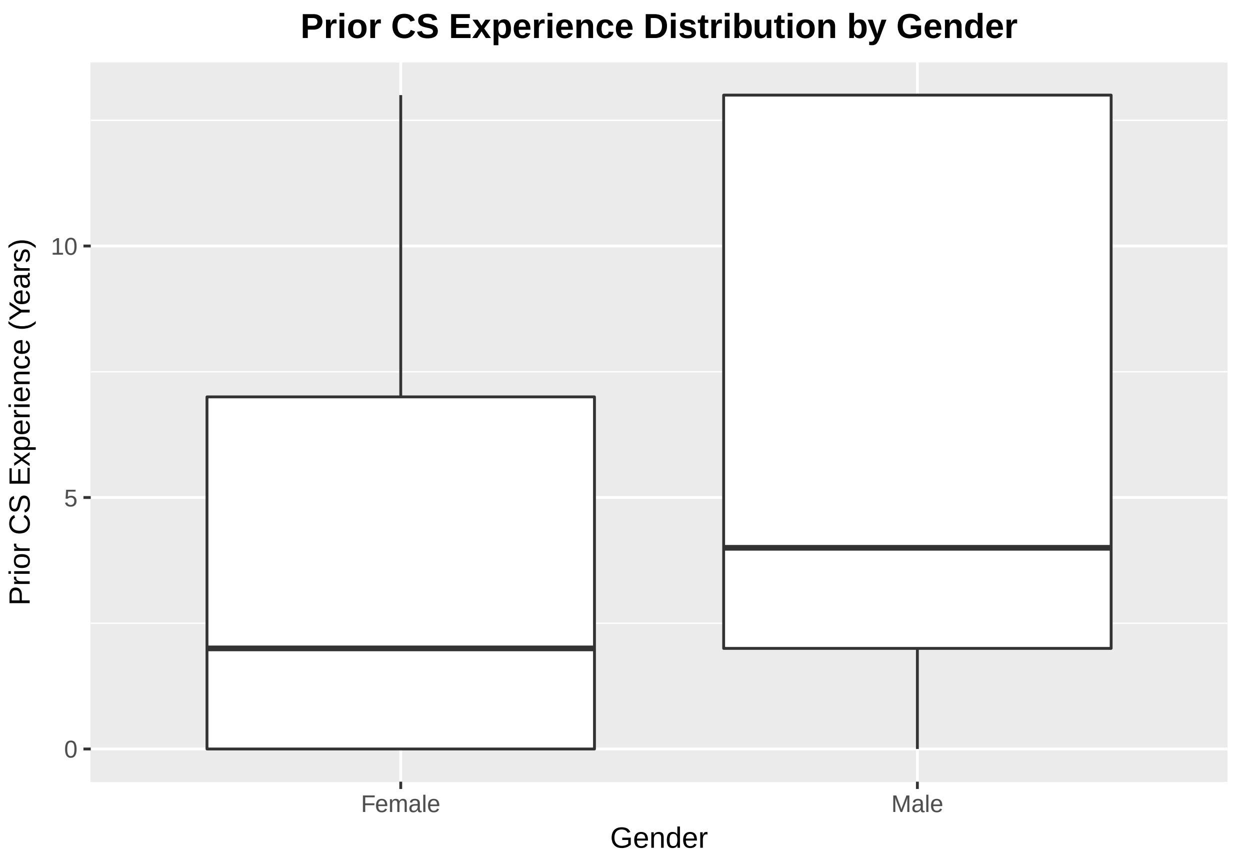
```r
# Boxplot of hours spent studying by gender
ggplot(survey, aes(gender, hours_num)) +  geom_boxplot() +
 labs(title = "Study Hours Distribution by Gender",
      x = "Gender", y = "Study Hours") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



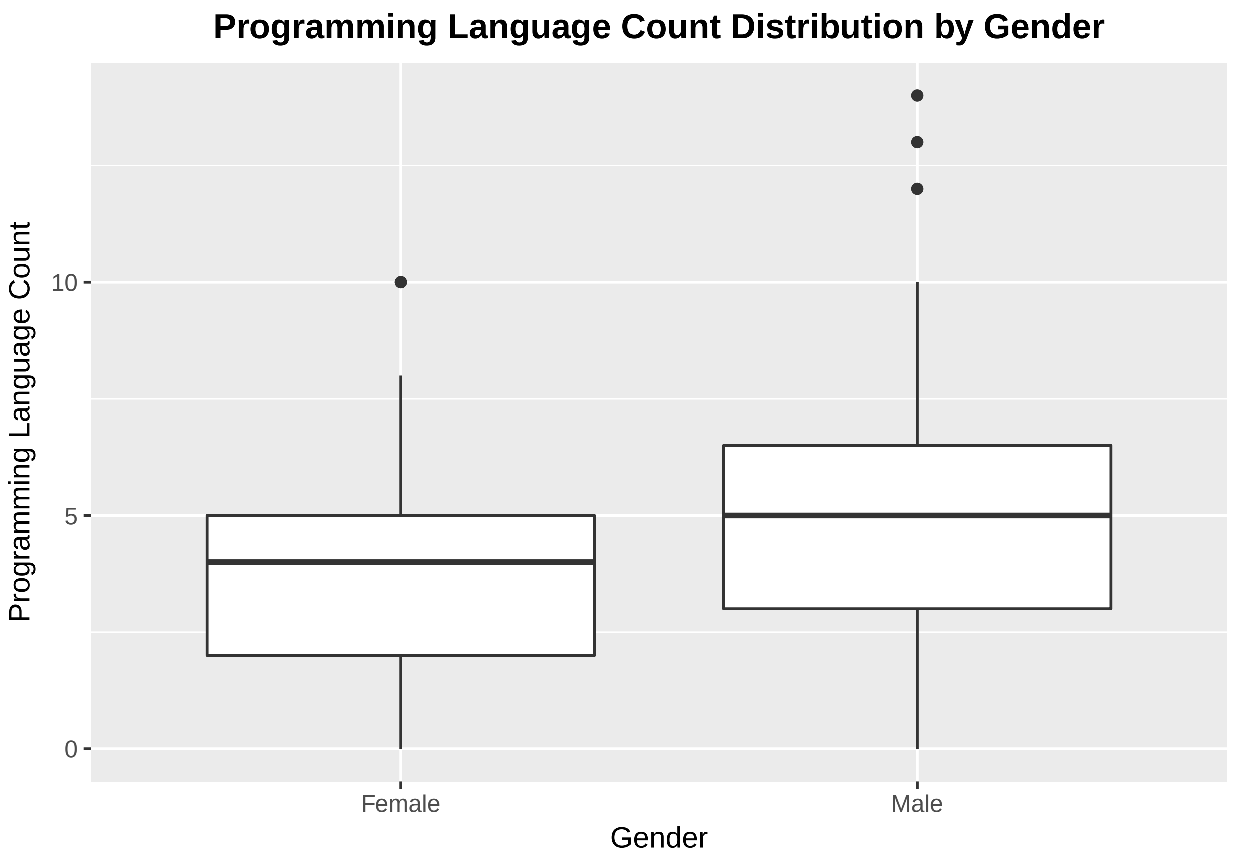
```r
# Boxplot of programming experience by gender
ggplot(survey, aes(gender, prog_num)) +
 geom_boxplot() +
 labs(title = "Programming Experience Distribution by Gender",
      x = "Gender", y = "Programming Experience (Years)") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



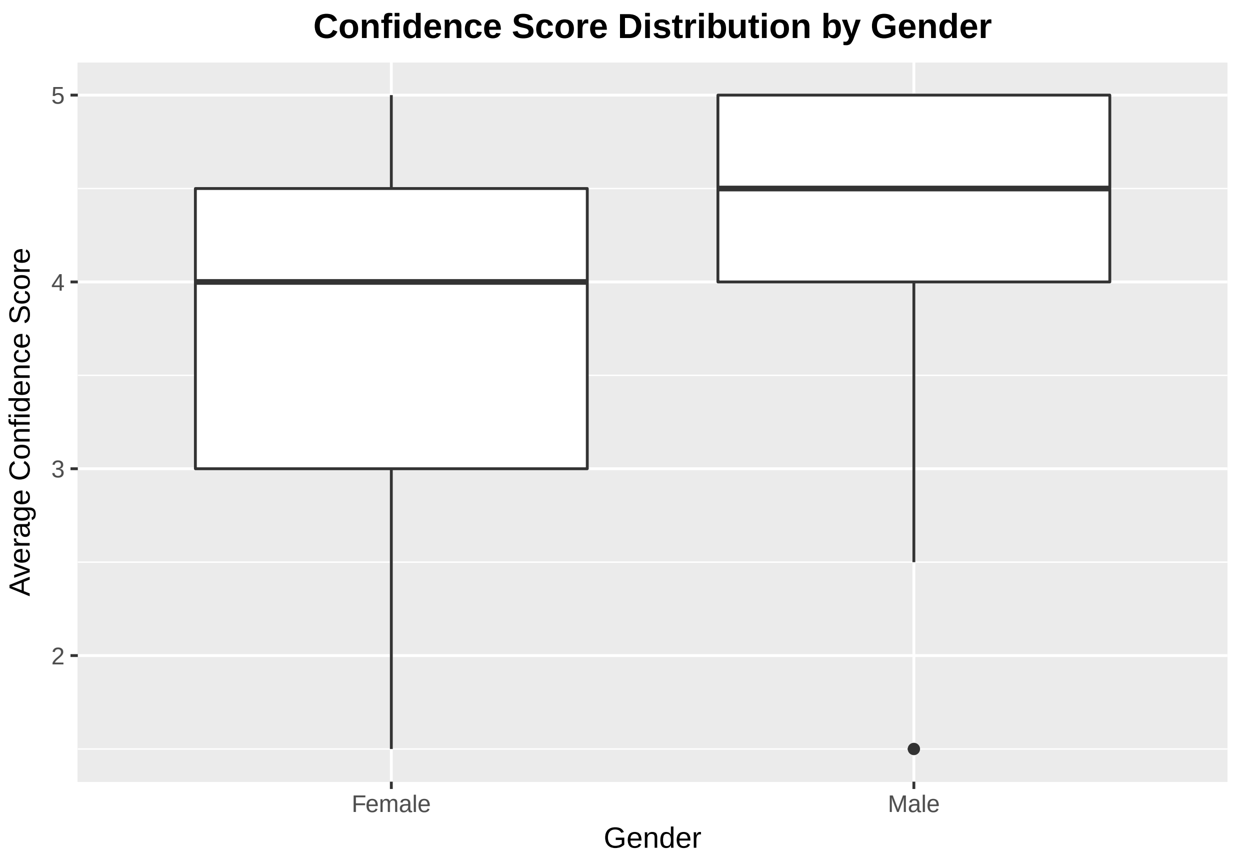
```
# Boxplot of prior cs experience by gender
ggplot(survey, aes(gender, prior_cs_num)) +
 geom_boxplot() +
 labs(title = "Prior CS Experience Distribution by Gender",
      x = "Gender", y = "Prior CS Experience (Years)") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



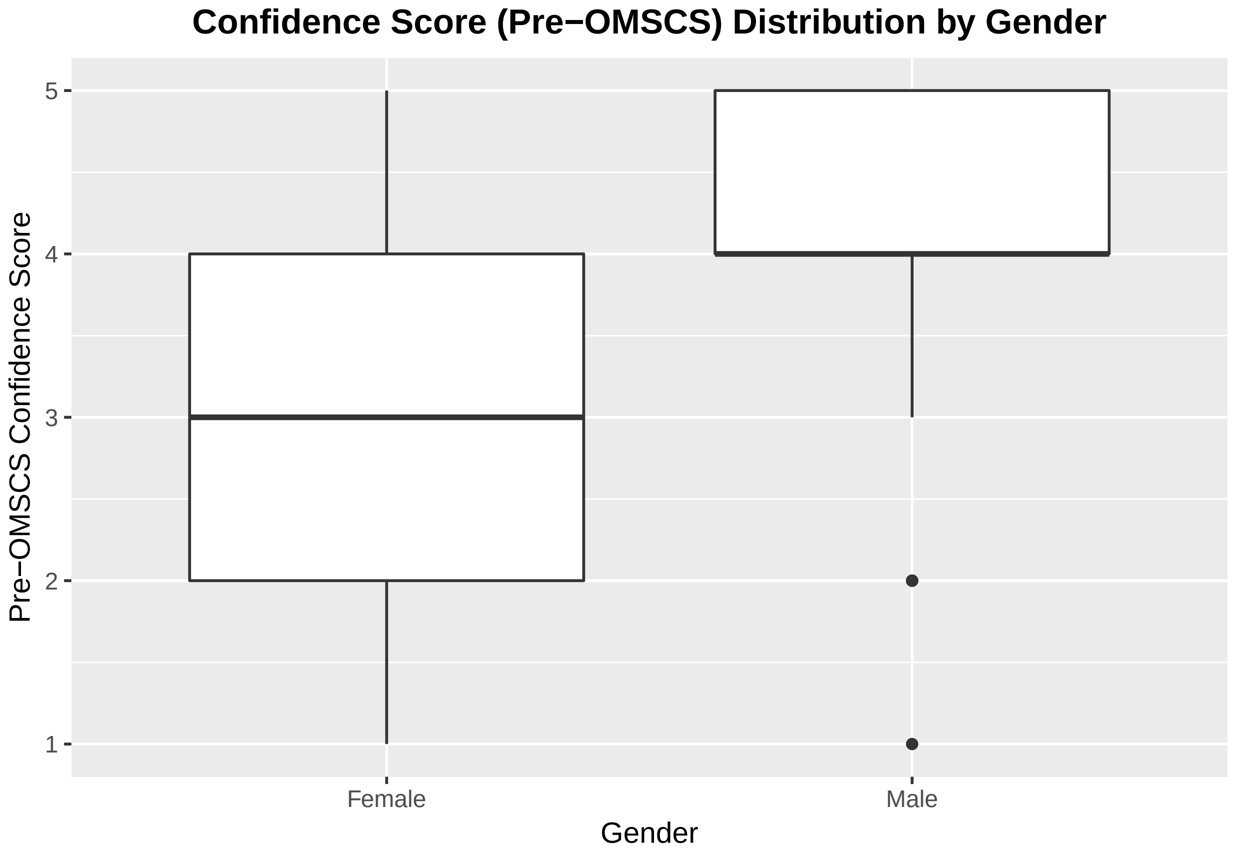
```
# Boxplot of programming language count by gender
ggplot(survey, aes(gender, prog_count)) +
 geom_boxplot() +
 labs(title = "Programming Language Count Distribution by Gender",
      x = "Gender", y = "Programming Language Count") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



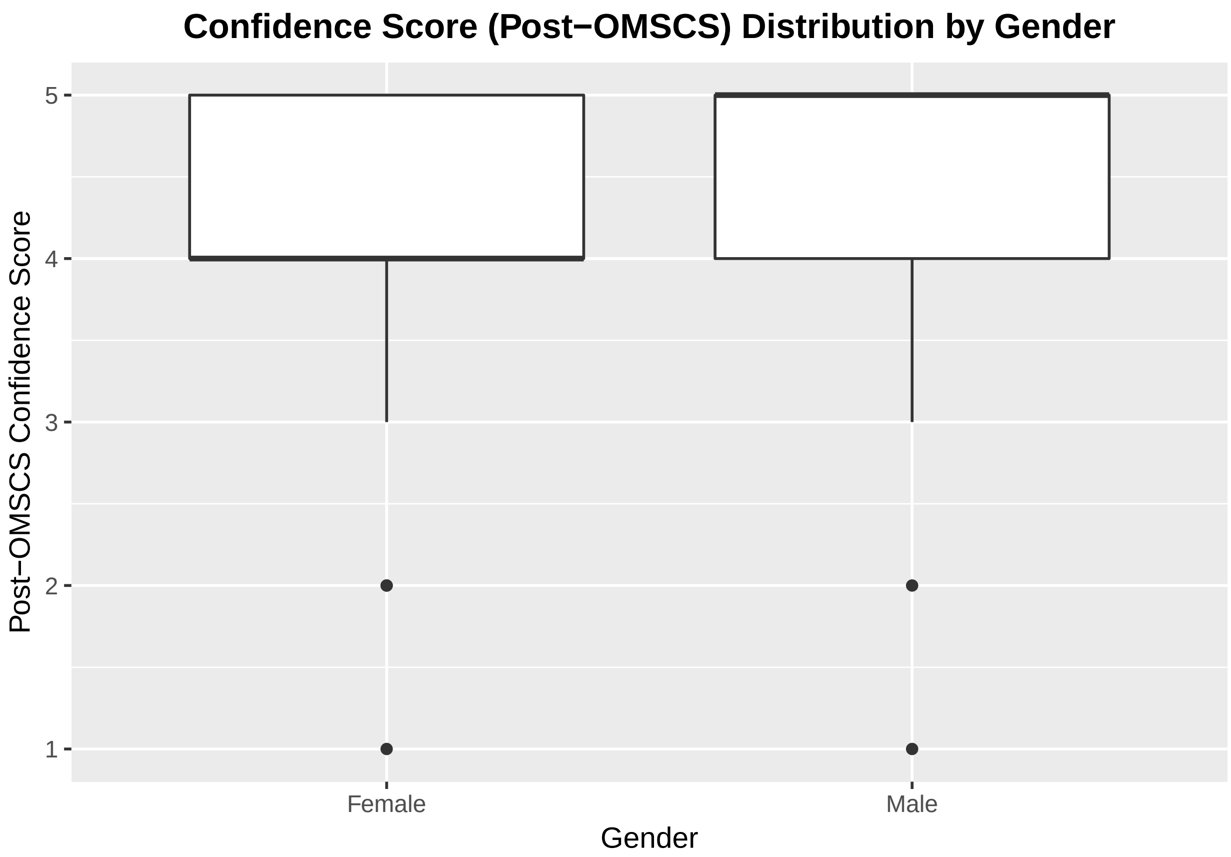
```
# Boxplot of confidence score by gender
ggplot(survey, aes(gender, conf_ave)) +  geom_boxplot() +
 labs(title = "Confidence Score Distribution by Gender",
      x = "Gender", y = "Average Confidence Score") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



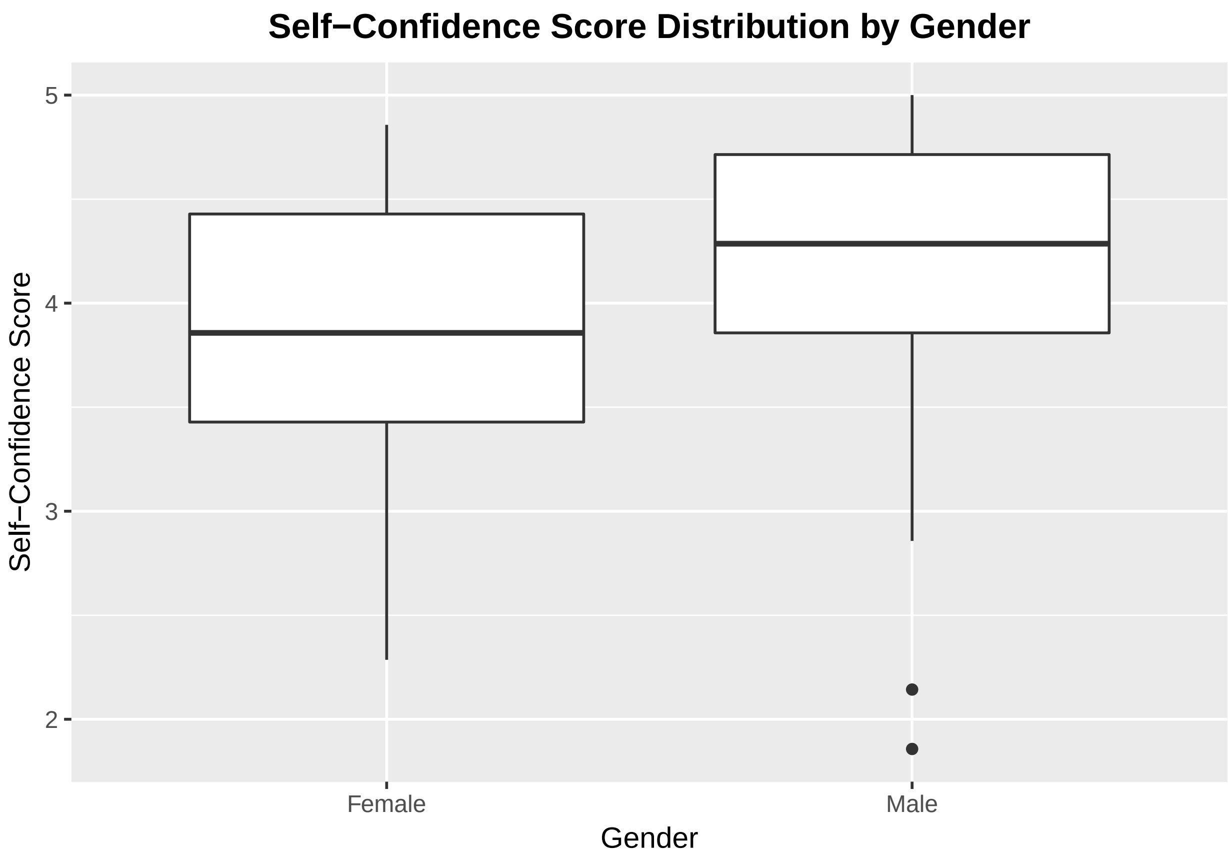
```
# Boxplot of confidence score (pre-OMSCS) by gender
ggplot(survey, aes(gender, conf_prior)) +  geom_boxplot() +
 labs(title = "Confidence Score (Pre-OMSCS) Distribution by Gender",
      x = "Gender", y = "Pre-OMSCS Confidence Score") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



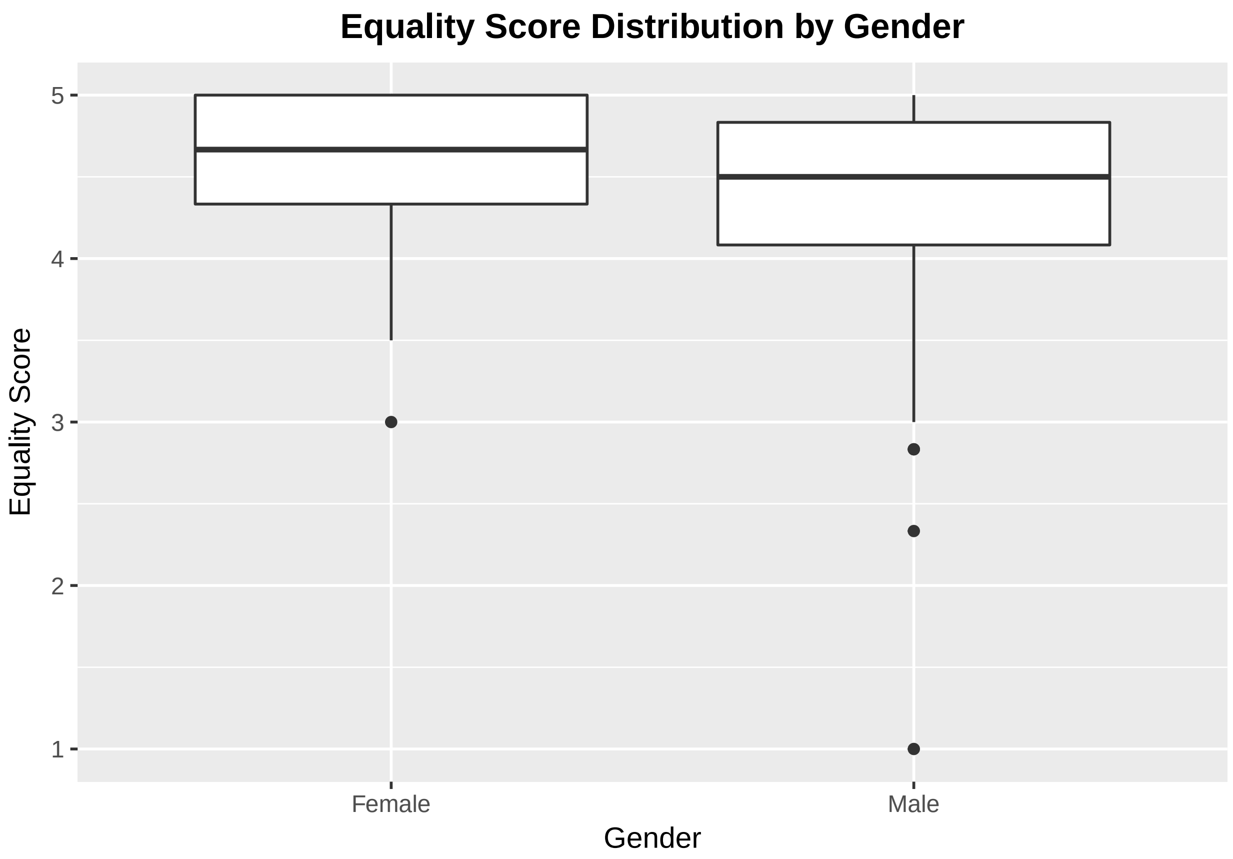
```r
# Boxplot of confidence score (post-OMSCS) by gender
ggplot(survey, aes(gender, conf_post)) +  geom_boxplot() +
 labs(title = "Confidence Score (Post-OMSCS) Distribution by Gender",
      x = "Gender", y = "Post-OMSCS Confidence Score") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



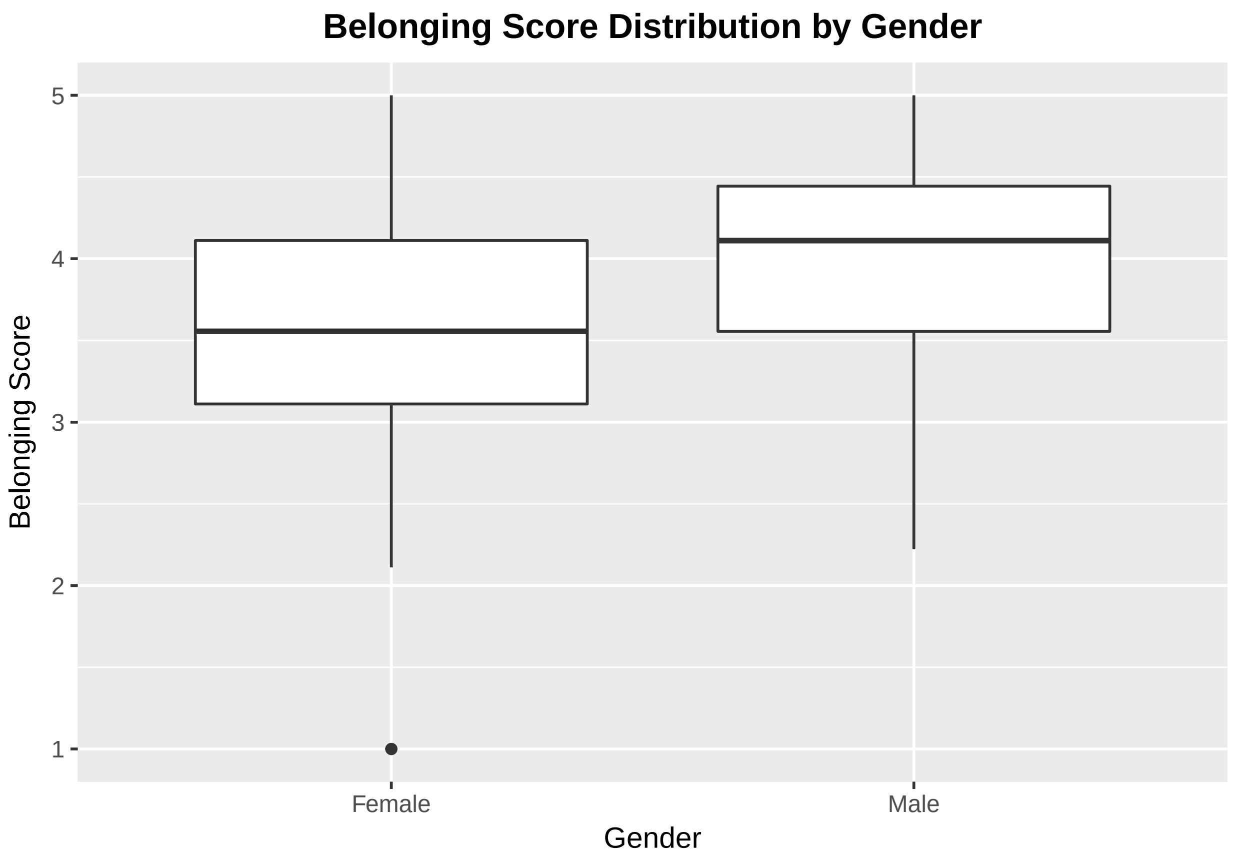
```
# Boxplot of self-confidence score by gender
ggplot(survey, aes(gender, selfconf_score)) +  geom_boxplot() +
 labs(title = "Self-Confidence Score Distribution by Gender",
      x = "Gender", y = "Self-Confidence Score") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



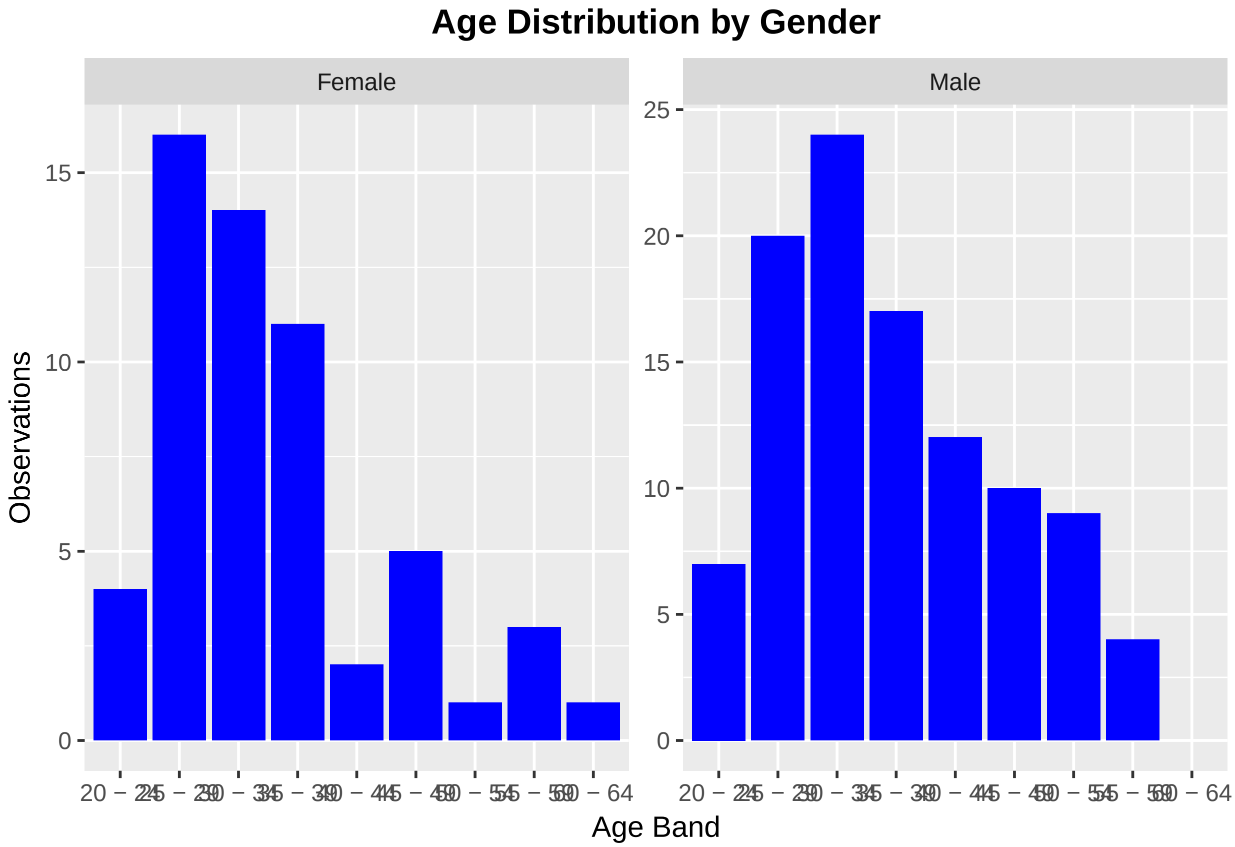
```
# Boxplot of equality score by gender
ggplot(survey, aes(gender, equality_score)) +  geom_boxplot() +
 labs(title = "Equality Score Distribution by Gender",
      x = "Gender", y = "Equality Score") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



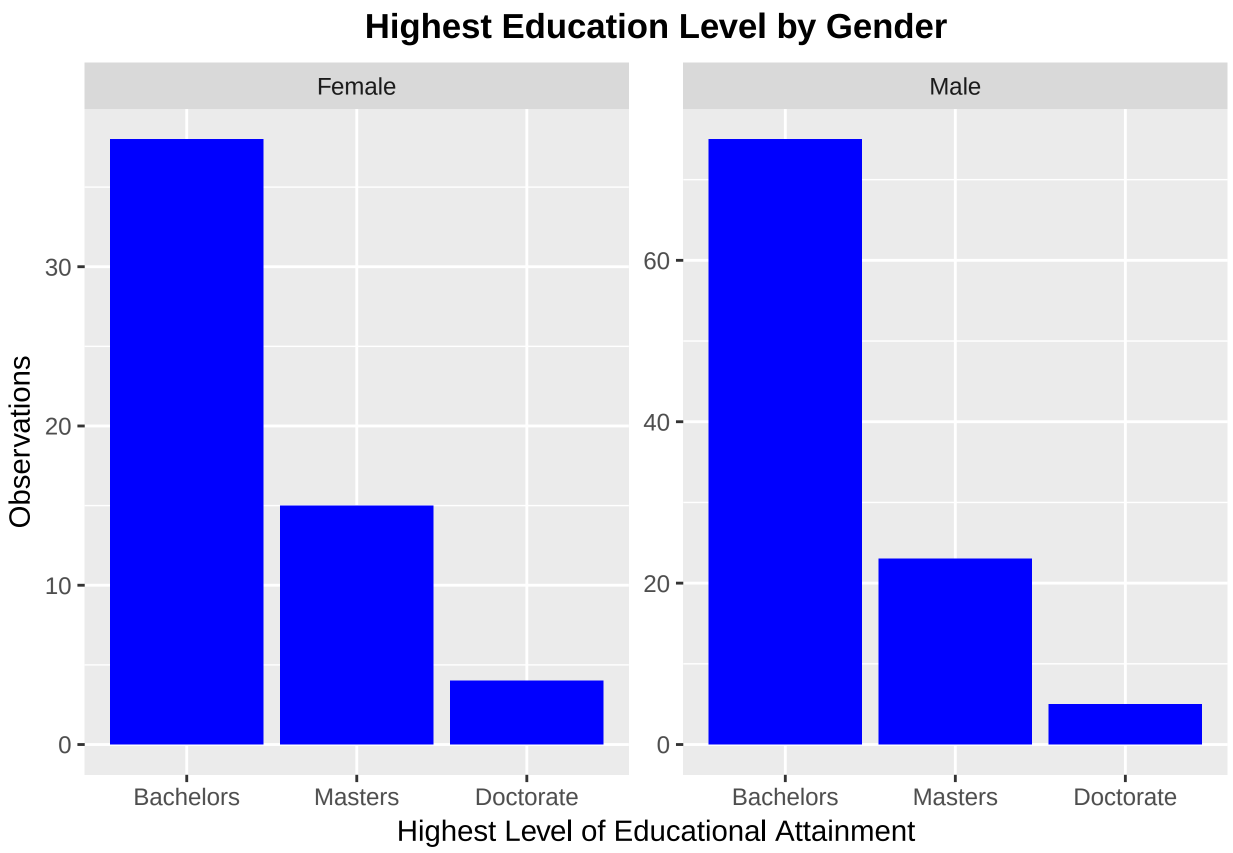
```
# Boxplot of belonging score by gender
ggplot(survey, aes(gender, belonging_score)) +  geom_boxplot() +
 labs(title = "Belonging Score Distribution by Gender",
      x = "Gender", y = "Belonging Score") +
 theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



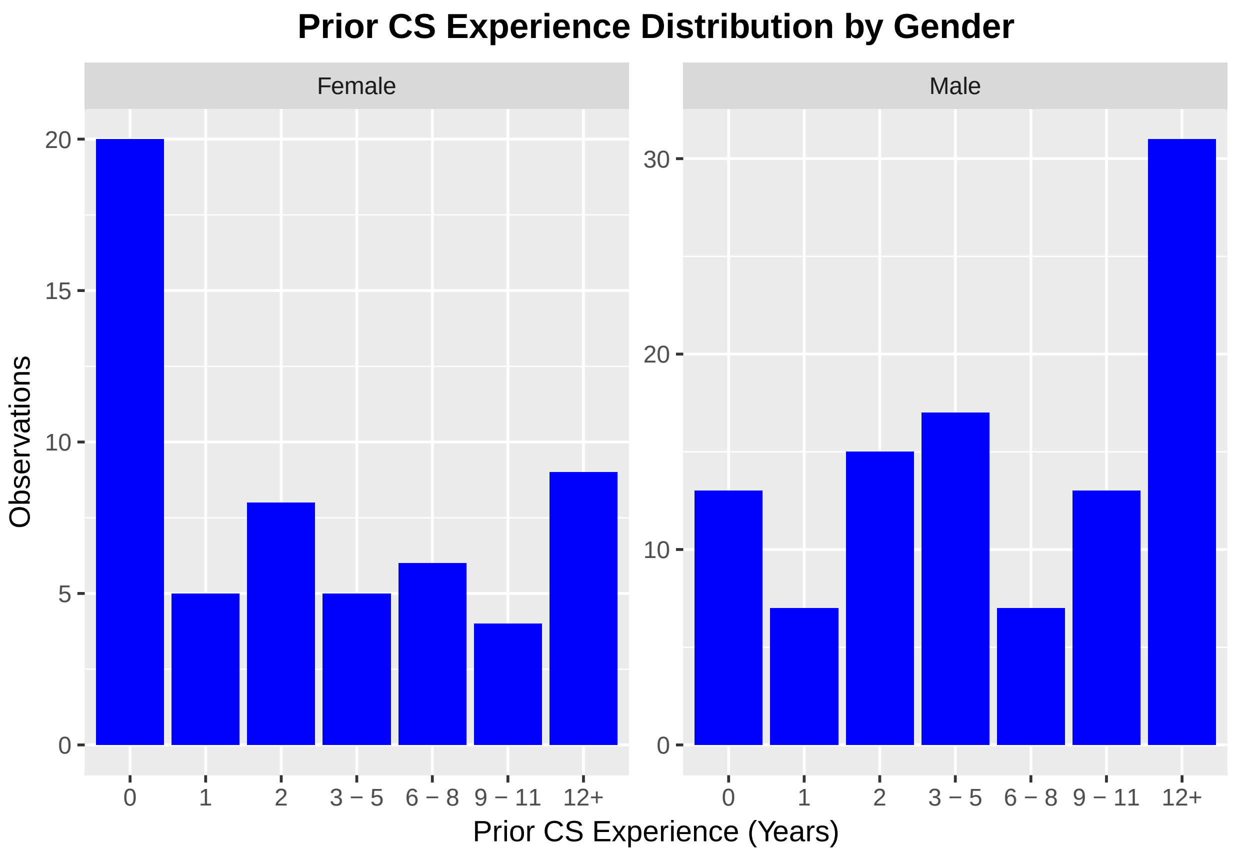
```
# Bar chart comparing age by gender
ggplot(survey, aes(x = age)) +
    geom_bar(fill = "blue") +
    facet_wrap(~gender, scales = "free_y") +
    labs(title = "Age Distribution by Gender",
      x = "Age Band",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold",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



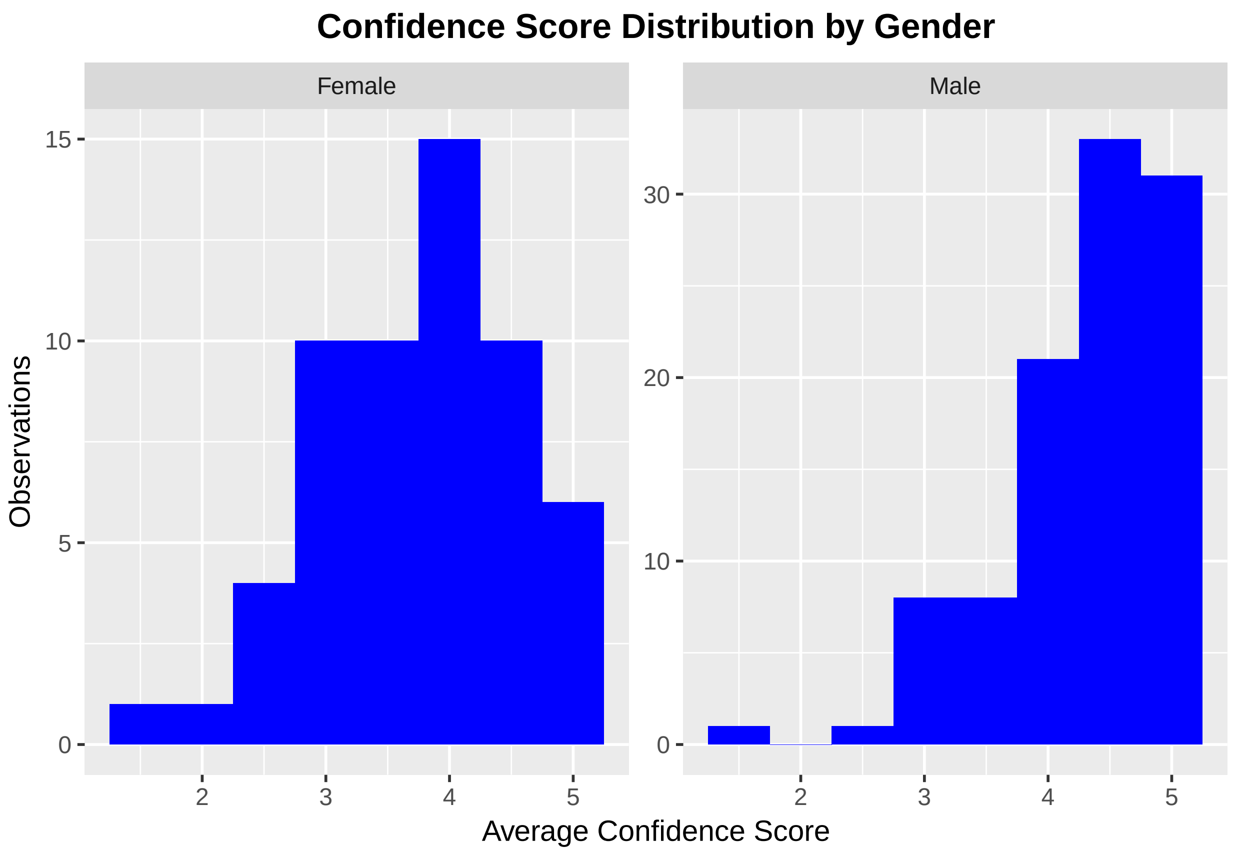
```r
# Bar chart comparing education level by gender
ggplot(survey, aes(x = education_level)) +
    geom_bar(fill = "blue") +
    facet_wrap(~gender, scales = "free_y") +
    labs(title = "Highest Education Level by Gender",
      x = "Highest Level of Educational Attainment",
      y = "Observations")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



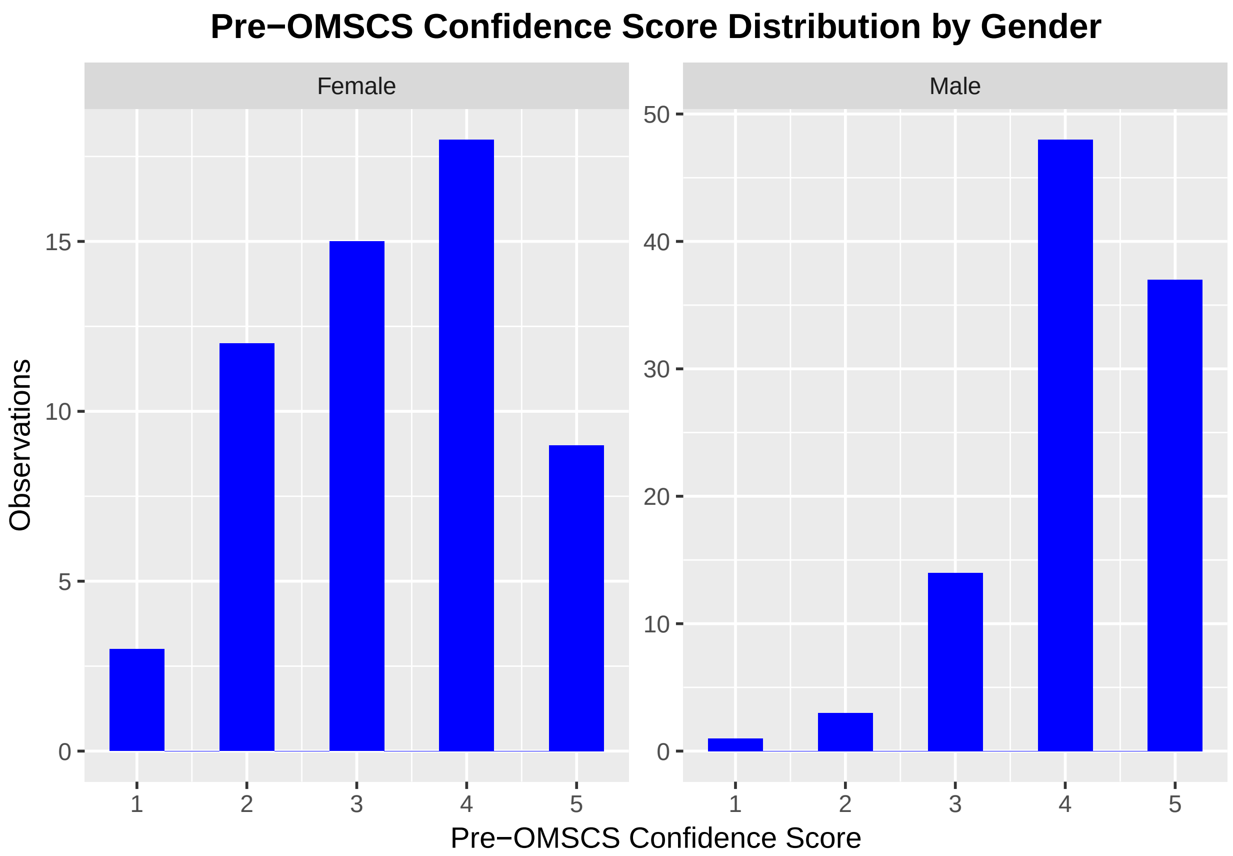
```r
# Bar chart comparing prior cs experience by gender
ggplot(survey, aes(x = prior_cs_exp)) +
    geom_bar(fill = "blue") +
    facet_wrap(~gender, scales = "free_y") +
    labs(title = "Prior CS Experience Distribution by Gender",
      x = "Prior CS Experience (Years)",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



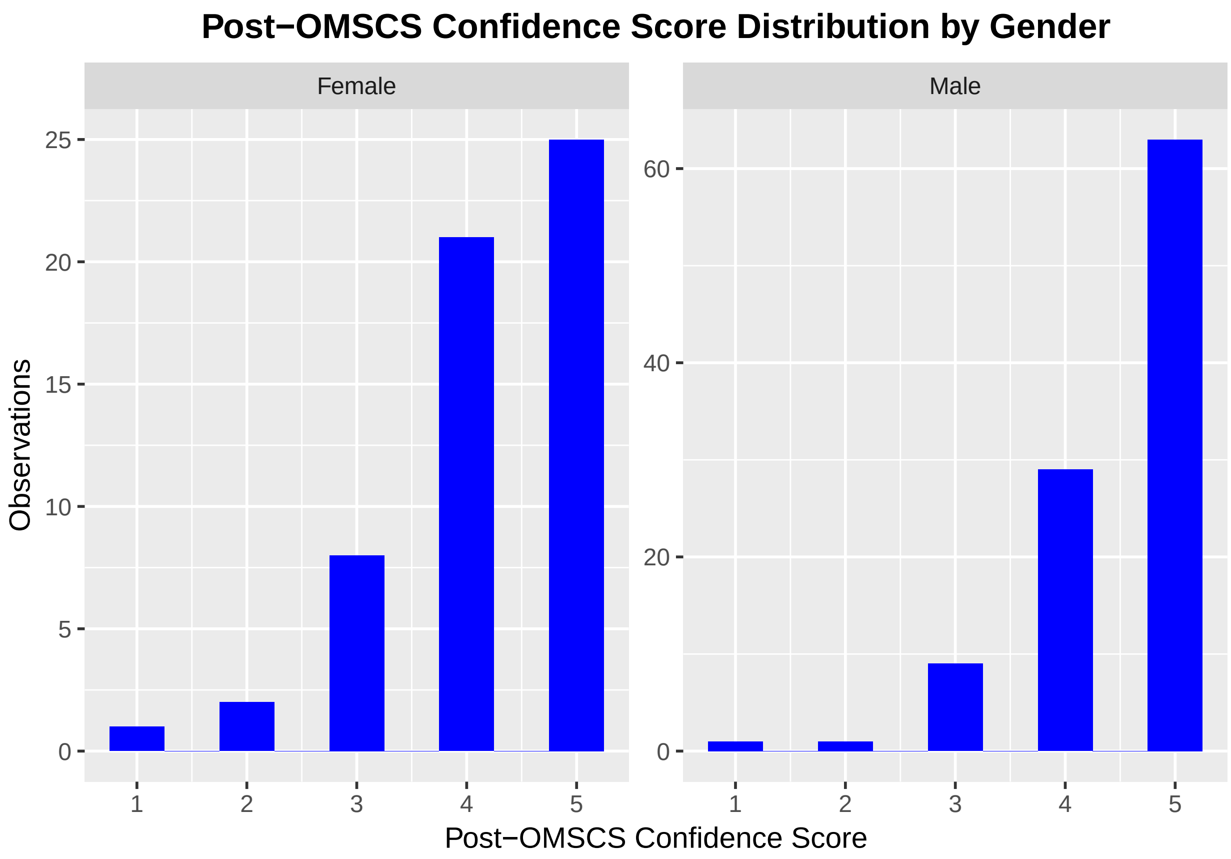
```r
# Histogram of conf_ave by gender
ggplot(survey, aes(x = conf_ave)) +
    geom_histogram(fill = "blue", binwidth = 0.5) +
    facet_wrap(~gender, scale = "free_y") +
    labs(title = "Confidence Score Distribution by Gender",
      x = "Average Confidence Score",
      y = "Observations") +
   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



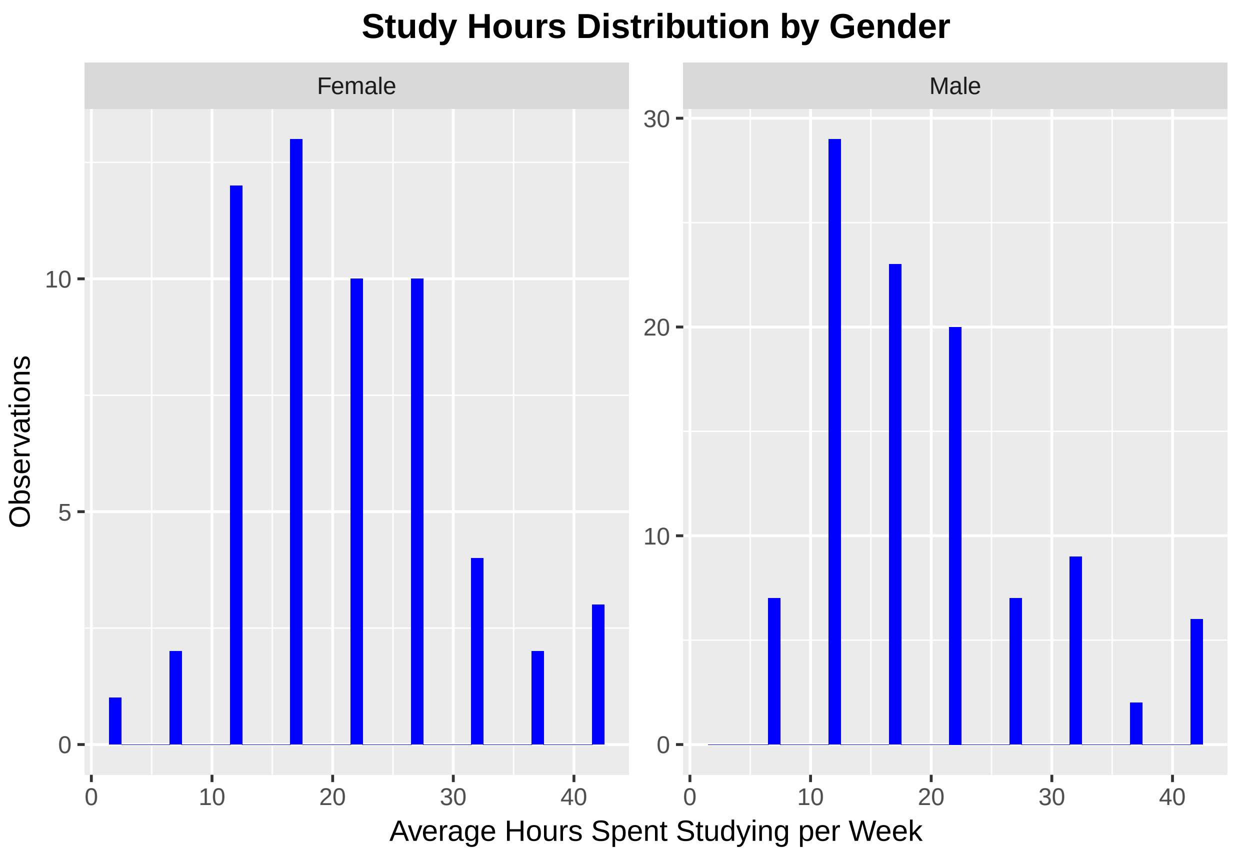
```r
# Histogram of conf_prior by gender
ggplot(survey, aes(x = conf_prior)) +
    geom_histogram(fill = "blue", binwidth = 0.5) +
    facet_wrap(~gender, scale = "free_y") +
    labs(title = "Pre-OMSCS Confidence Score Distribution by Gender",
      x = "Pre-OMSCS Confidence Score",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



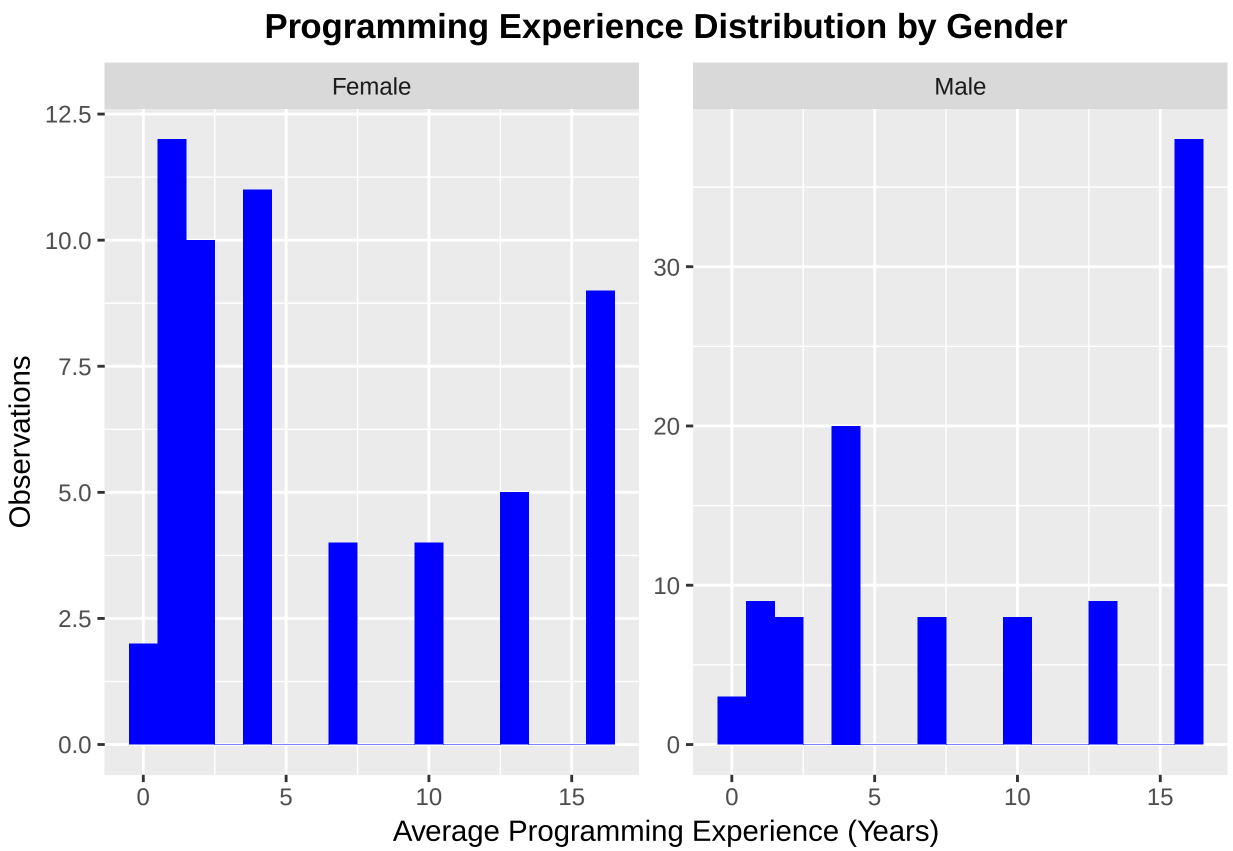
```
# Histogram of conf_post by gender
ggplot(survey, aes(x = conf_post)) +
    geom_histogram(fill = "blue", binwidth = 0.5) +
    facet_wrap(~gender, scale = "free_y") +
    labs(title = "Post-OMSCS Confidence Score Distribution by Gender",
      x = "Post-OMSCS Confidence Score",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



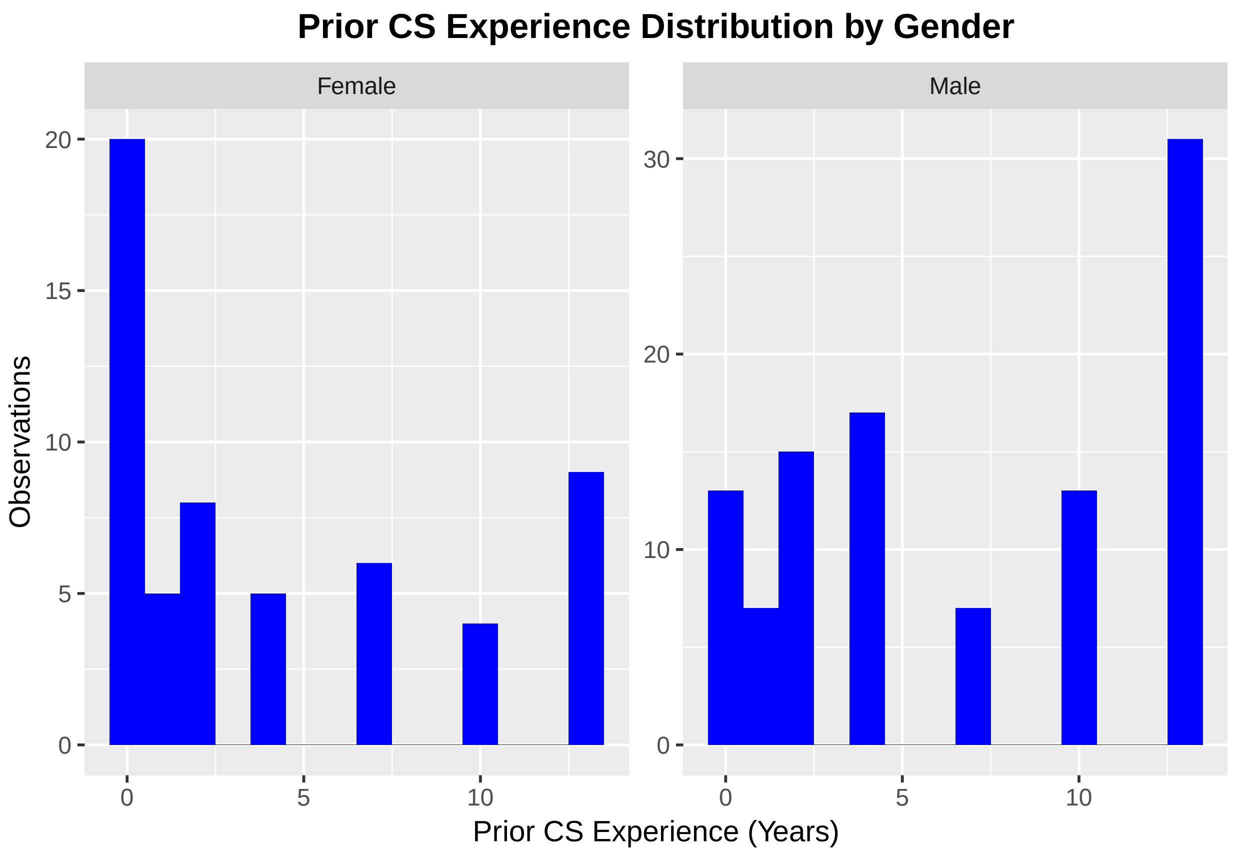
```
# Histogram of study hours by gender
ggplot(survey, aes(x = hours_num)) +
    geom_histogram(fill = "blue", binwidth = 1) +
    facet_wrap(~gender, scale = "free_y") +
    labs(title = "Study Hours Distribution by Gender",
      x = "Average Hours Spent Studying per Week",
      y = "Observations")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



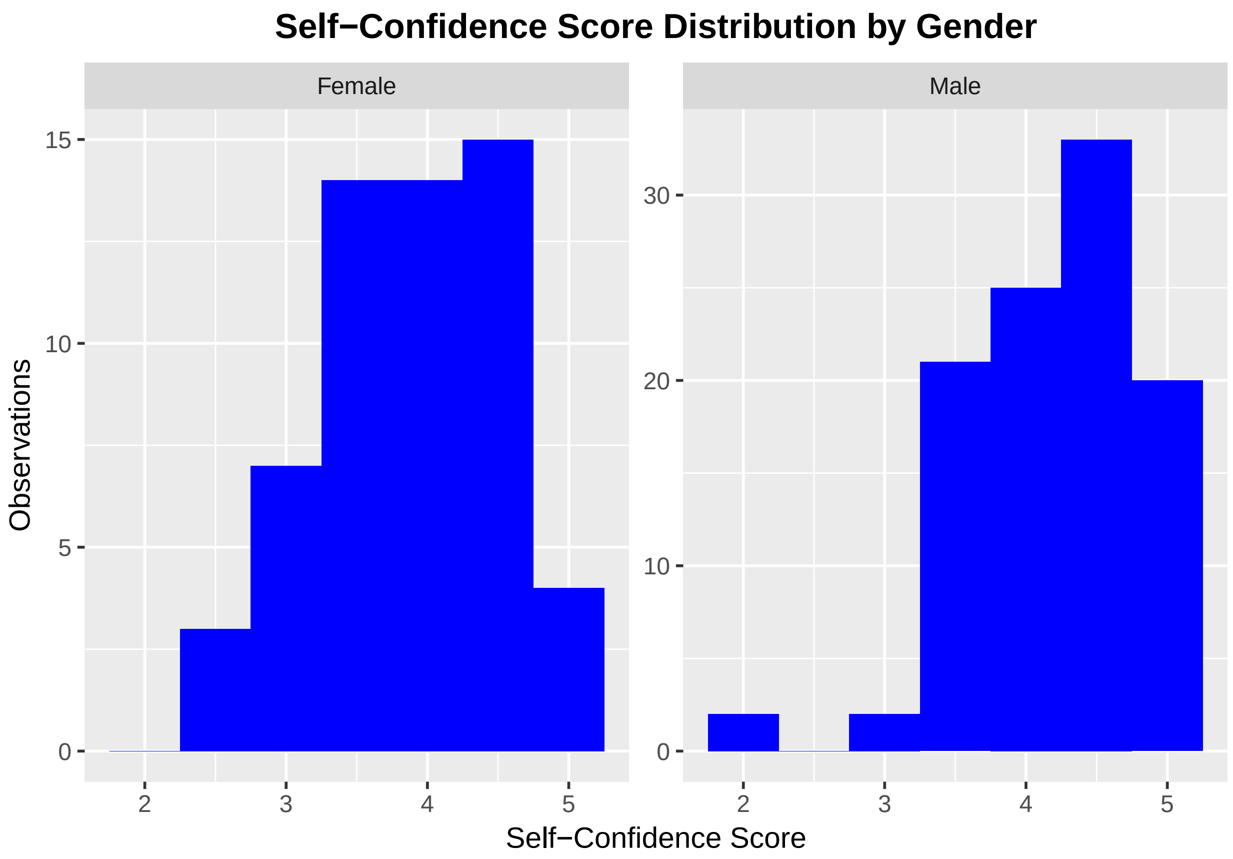
```r
# Histogram of programming experience by gender
ggplot(survey, aes(x = prog_num)) +
    geom_histogram(fill = "blue", binwidth = 1) +
    facet_wrap(~gender, scale = "free_y") +
    labs(title = "Programming Experience Distribution by Gender",
      x = "Average Programming Experience (Years)",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



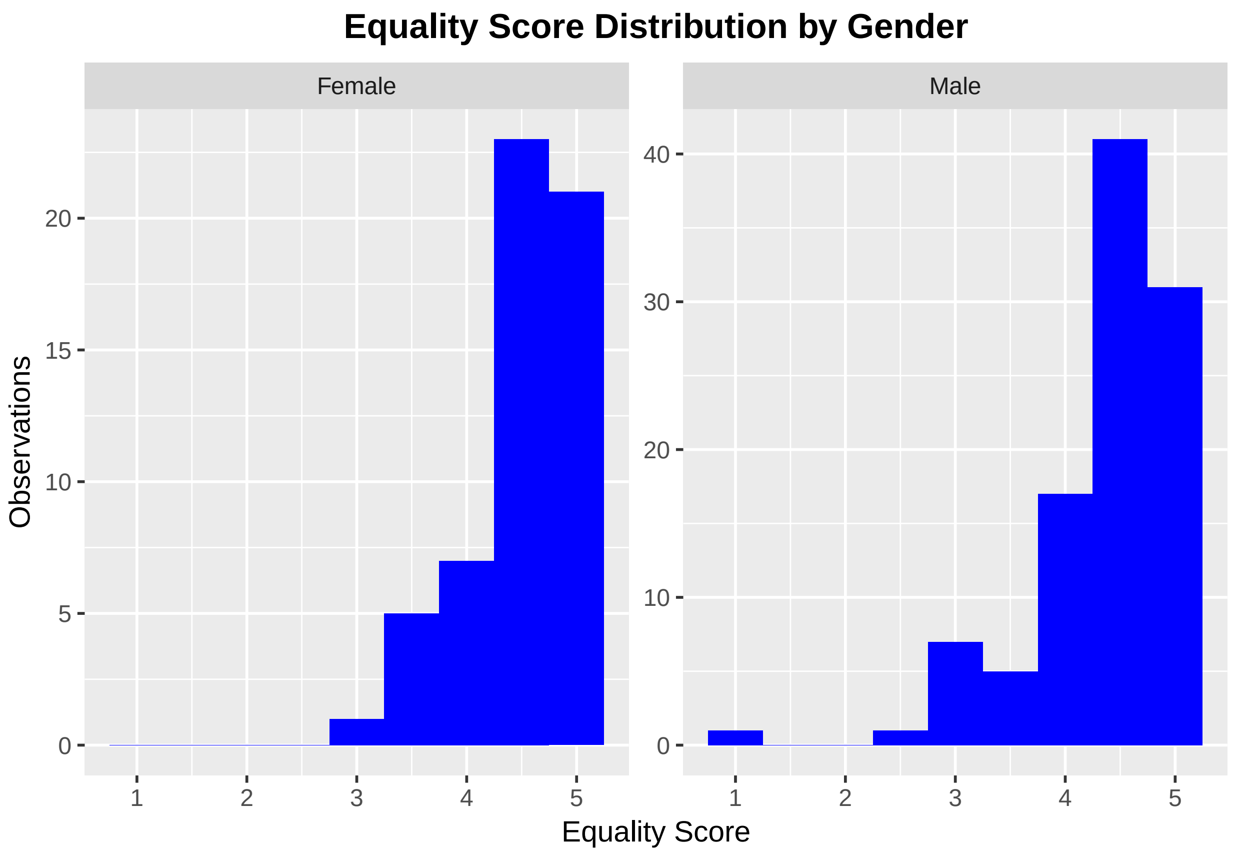
```r
# Histogram of prior cs experience by gender
ggplot(survey, aes(x = prior_cs_num)) +
    geom_histogram(fill = "blue", binwidth = 1) +
    facet_wrap(~gender, scale = "free_y") +
    labs(title = "Prior CS Experience Distribution by Gender",
      x = "Prior CS Experience (Years)",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



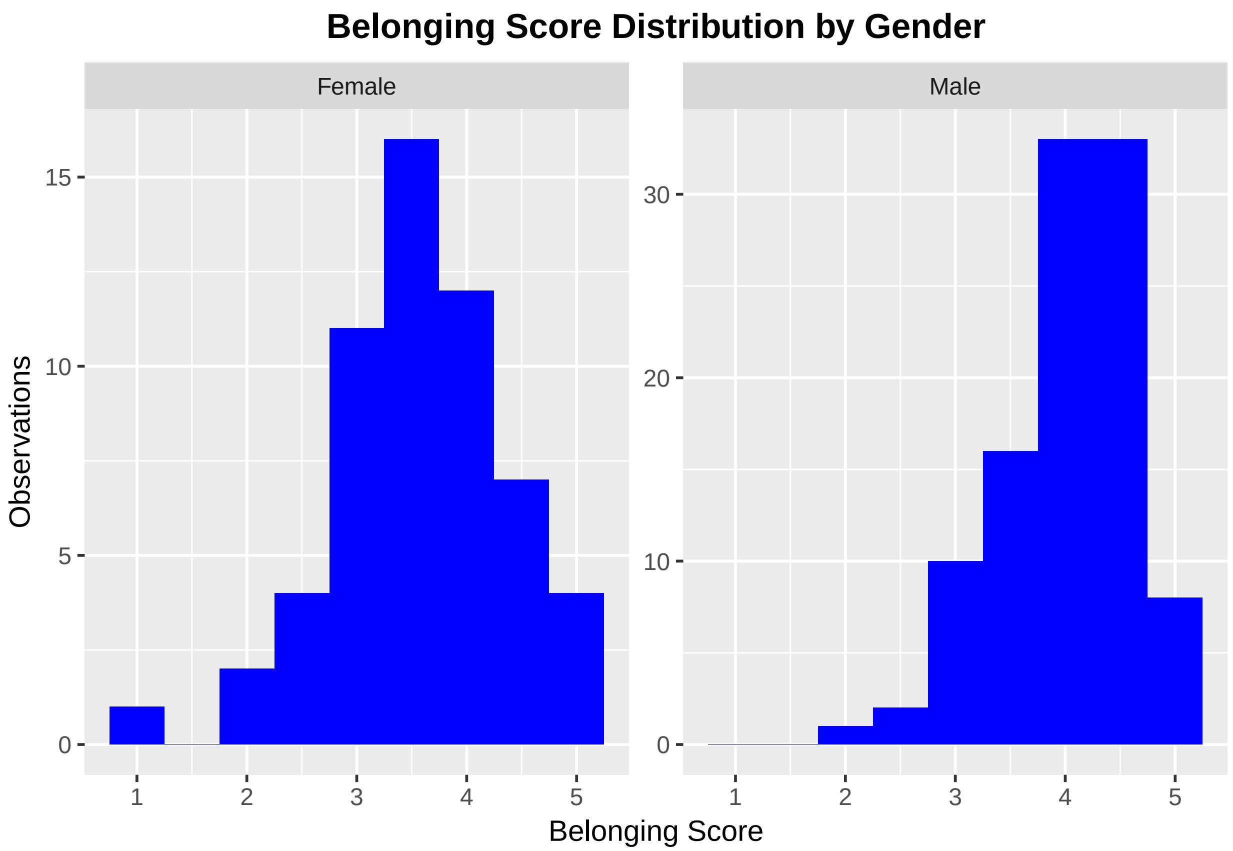
```r
# Histogram of self-confidence score by gender
ggplot(survey, aes(x = selfconf_score)) +
    geom_histogram(fill = "blue", binwidth = 0.5) +
    facet_wrap(~gender, scale = "free_y") +
    labs(title = "Self-Confidence Score Distribution by Gender",
      x = "Self-Confidence Score",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



```r
# Histogram of equality score by gender
ggplot(survey, aes(x = equality_score)) +
    geom_histogram(fill = "blue", binwidth = 0.5) +
    facet_wrap(~gender, scale = "free_y") +
    labs(title = "Equality Score Distribution by Gender",
      x = "Equality Score",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



```
# Histogram of belonging score by gender
ggplot(survey, aes(x = belonging_score)) +
    geom_histogram(fill = "blue", binwidth = 0.5) +
    facet_wrap(~gender, scale = "free_y") +
    labs(title = "Belonging Score Distribution by Gender",
      x = "Belonging Score",
      y = "Observations") +
    theme(plot.title = element_text(lineheight=.8, face="bold", hjust=0.5))
```



```
# Age tests
t.test(survey_m$age_num, survey_f$age_num)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$age_num and survey_f$age_num
## t = 1.1404, df = 114.04, p-value = 0.2565
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.323369  4.913898
## sample estimates:
## mean of x mean of y
##  36.51456  34.71930
```

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

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  age_num by gender
## W = 2560, p-value = 0.1744
## alternative hypothesis: true location shift is not equal to 0
```

```
# GPA tests
t.test(survey_m$gpa_num, survey_f$gpa_num)
```

```
##
```

```
##  Welch Two Sample t-test
##
## data:  survey_m$gpa_num and survey_f$gpa_num
## t = -1.0033, df = 99.97, p-value = 0.3181
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.15731386  0.05164493
## sample estimates:
## mean of x mean of y
##  3.719388  3.772222
```

```
wilcox.test(gpa_num ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  gpa_num by gender
## W = 3010, p-value = 0.1297
## alternative hypothesis: true location shift is not equal to 0
```

```
# Average confidence score tests
t.test(survey_m$conf_ave, survey_f$conf_ave)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$conf_ave and survey_f$conf_ave
## t = 4.4242, df = 100.88, p-value = 2.451e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.3090170 0.8114054
## sample estimates:
## mean of x mean of y
##  4.305825  3.745614
```

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

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  conf_ave by gender
## W = 1715, p-value = 8.567e-06
## alternative hypothesis: true location shift is not equal to 0
```

```
# Pre-OMSCS confidence score tests
t.test(survey_m$conf_prior, survey_f$conf_prior)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$conf_prior and survey_f$conf_prior
## t = 4.789, df = 89.576, p-value = 6.591e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.4798879 1.1603778
## sample estimates:
## mean of x mean of y
##  4.135922  3.315789
```

```r
wilcox.test(conf_prior ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  conf_prior by gender
## W = 1713, p-value = 4.403e-06
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Post-OMSCS confidence score tests
t.test(survey_m$conf_post, survey_f$conf_post)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$conf_post and survey_f$conf_post
## t = 2.0729, df = 99.701, p-value = 0.04076
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.01286762 0.58771150
## sample estimates:
## mean of x mean of y
##  4.475728  4.175439
```

```r
wilcox.test(conf_post ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  conf_post by gender
## W = 2378.5, p-value = 0.02681
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Study hours
t.test(survey_m$hours_num, survey_f$hours_num)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$hours_num and survey_f$hours_num
## t = -0.87752, df = 115.8, p-value = 0.382
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.288435  1.655153
## sample estimates:
## mean of x mean of y
##  19.71845  21.03509
```

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

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  hours_num by gender
## W = 3251.5, p-value = 0.252
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Programming experience tests
t.test(survey_m$prog_num, survey_f$prog_num)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$prog_num and survey_f$prog_num
## t = 3.3285, df = 122.67, p-value = 0.001153
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.291131 5.080186
## sample estimates:
## mean of x mean of y
##  9.378641  6.192982
```

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

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  prog_num by gender
## W = 2025.5, p-value = 0.0009543
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Prior CS experience tests
t.test(survey_m$prior_cs_num, survey_f$prior_cs_num)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$prior_cs_num and survey_f$prior_cs_num
## t = 3.0155, df = 120.26, p-value = 0.003131
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.8446286 4.0741246
## sample estimates:
## mean of x mean of y
##  6.669903  4.210526
```

```r
wilcox.test(prior_cs_num ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  prior_cs_num by gender
## W = 2032.5, p-value = 0.001081
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Programming language count tests
t.test(survey_m$prog_count, survey_f$prog_count)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$prog_count and survey_f$prog_count
## t = 2.2813, df = 125.94, p-value = 0.02421
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1211331 1.7071756
## sample estimates:
## mean of x mean of y
##  5.019417  4.105263
```

```
wilcox.test(prog_count ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  prog_count by gender
## W = 2323.5, p-value = 0.0281
## alternative hypothesis: true location shift is not equal to 0
```

```
# Self-confidence score tests
t.test(survey_m$selfconf_score, survey_f$selfconf_score)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$selfconf_score and survey_f$selfconf_score
## t = 3.4386, df = 108.2, p-value = 0.0008312
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1547532 0.5759571
## sample estimates:
## mean of x mean of y
##  4.214979  3.849624
```

```
wilcox.test(selfconf_score ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  selfconf_score by gender
## W = 1970.5, p-value = 0.0005691
## alternative hypothesis: true location shift is not equal to 0
```

```
t.test(survey_m$selfconf1, survey_f$selfconf1)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$selfconf1 and survey_f$selfconf1
## t = 2.9183, df = 106.82, p-value = 0.004291
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1359072 0.7116486
## sample estimates:
## mean of x mean of y
##  4.300971  3.877193
```

```
wilcox.test(selfconf1 ~ gender, data=survey)
```

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

```
## data:  selfconf1 by gender
## W = 2113.5, p-value = 0.001597
## alternative hypothesis: true location shift is not equal to 0
```

```
t.test(survey_m$selfconf2, survey_f$selfconf2)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$selfconf2 and survey_f$selfconf2
## t = 2.2955, df = 101.72, p-value = 0.02376
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.03360707 0.46102758
## sample estimates:
## mean of x mean of y
##  4.563107  4.315789
```

```
wilcox.test(selfconf2 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  selfconf2 by gender
## W = 2356, p-value = 0.01934
## alternative hypothesis: true location shift is not equal to 0
```

```
t.test(survey_m$selfconf3, survey_f$selfconf3)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$selfconf3 and survey_f$selfconf3
## t = 2.2164, df = 108.55, p-value = 0.02875
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.03629975 0.65046570
## sample estimates:
## mean of x mean of y
##  3.922330  3.578947
```

```
wilcox.test(selfconf3 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  selfconf3 by gender
## W = 2315, p-value = 0.01907
## alternative hypothesis: true location shift is not equal to 0
```

```
t.test(survey_m$selfconf4, survey_f$selfconf4)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$selfconf4 and survey_f$selfconf4
## t = 1.2949, df = 120.79, p-value = 0.1978
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.09495805  0.45401102
## sample estimates:
## mean of x mean of y
##  3.951456  3.771930
```

```r
wilcox.test(selfconf4 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  selfconf4 by gender
## W = 2536, p-value = 0.1306
## alternative hypothesis: true location shift is not equal to 0
```

```r
t.test(survey_m$selfconf5, survey_f$selfconf5)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$selfconf5 and survey_f$selfconf5
## t = 2.1418, df = 114.68, p-value = 0.03433
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.01980794 0.50752982
## sample estimates:
## mean of x mean of y
##  4.368932  4.105263
```

```r
wilcox.test(selfconf5 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  selfconf5 by gender
## W = 2328.5, p-value = 0.01776
## alternative hypothesis: true location shift is not equal to 0
```

```r
t.test(survey_m$selfconf6, survey_f$selfconf6)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$selfconf6 and survey_f$selfconf6
## t = 5.3828, df = 102.56, p-value = 4.665e-07
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.5523674 1.1969088
## sample estimates:
## mean of x mean of y
##  4.155340  3.280702
```

```r
wilcox.test(selfconf6 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  selfconf6 by gender
## W = 1570, p-value = 3.775e-07
## alternative hypothesis: true location shift is not equal to 0
```

```r
t.test(survey_m$selfconf7, survey_f$selfconf7)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$selfconf7 and survey_f$selfconf7
## t = 1.345, df = 117.15, p-value = 0.1812
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1063897  0.5567388
## sample estimates:
## mean of x mean of y
##  4.242718  4.017544
```

```r
wilcox.test(selfconf7 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  selfconf7 by gender
## W = 2486, p-value = 0.08382
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Equality score tests
t.test(survey_m$equality_score, survey_f$equality_score)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$equality_score and survey_f$equality_score
## t = -1.7839, df = 149.65, p-value = 0.07646
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.34391287  0.01756302
## sample estimates:
## mean of x mean of y
##  4.354369  4.517544
```

```r
wilcox.test(equality_score ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  equality_score by gender
## W = 3284.5, p-value = 0.2088
## alternative hypothesis: true location shift is not equal to 0
```

```r
t.test(survey_m$equality1, survey_f$equality1)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$equality1 and survey_f$equality1
## t = -1.7512, df = 146.01, p-value = 0.08202
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.43108509  0.02604336
## sample estimates:
## mean of x mean of y
##  4.446602  4.649123
```

```r
wilcox.test(equality1 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  equality1 by gender
## W = 3292, p-value = 0.1335
## alternative hypothesis: true location shift is not equal to 0
```

```r
t.test(survey_m$equality2, survey_f$equality2)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$equality2 and survey_f$equality2
## t = -0.3108, df = 146.68, p-value = 0.7564
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.2481694  0.1807192
## sample estimates:
## mean of x mean of y
##  4.650485  4.684211
```

```r
wilcox.test(equality2 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  equality2 by gender
## W = 2848.5, p-value = 0.6782
## alternative hypothesis: true location shift is not equal to 0
```

```r
t.test(survey_m$equality3, survey_f$equality3)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$equality3 and survey_f$equality3
## t = -1.8176, df = 157.72, p-value = 0.07103
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.35933112  0.01492641
## sample estimates:
## mean of x mean of y
##  4.669903  4.842105
```

```r
wilcox.test(equality3 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  equality3 by gender
## W = 3189, p-value = 0.1895
## alternative hypothesis: true location shift is not equal to 0
```

```r
t.test(survey_m$equality4, survey_f$equality4)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$equality4 and survey_f$equality4
## t = -1.9172, df = 156.01, p-value = 0.05703
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.389040496  0.005800843
## sample estimates:
## mean of x mean of y
##  4.650485  4.842105
```

```r
wilcox.test(equality4 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  equality4 by gender
## W = 3122, p-value = 0.3161
## alternative hypothesis: true location shift is not equal to 0
```

```r
t.test(survey_m$equality5, survey_f$equality5)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$equality5 and survey_f$equality5
## t = -0.90584, df = 107.68, p-value = 0.367
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.6554683  0.2442948
## sample estimates:
## mean of x mean of y
##  3.233010  3.438596
```

```r
wilcox.test(equality5 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  equality5 by gender
## W = 3192, p-value = 0.3467
## alternative hypothesis: true location shift is not equal to 0
```

```r
t.test(survey_m$equality6, survey_f$equality6)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$equality6 and survey_f$equality6
## t = -1.3085, df = 147.61, p-value = 0.1928
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  -0.43527364  0.08848433
## sample estimates:
## mean of x mean of y
##  4.475728  4.649123
```

```r
wilcox.test(equality6 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  equality6 by gender
## W = 3103, p-value = 0.4542
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Belonging score tests
t.test(survey_m$belonging_score, survey_f$belonging_score)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$belonging_score and survey_f$belonging_score
## t = 3.4922, df = 93.219, p-value = 0.0007342
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1792029 0.6516228
## sample estimates:
## mean of x mean of y
##  4.002157  3.586745
```

```r
wilcox.test(belonging_score ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  belonging_score by gender
## W = 1966, p-value = 0.0005437
## alternative hypothesis: true location shift is not equal to 0
```

```r
t.test(survey_m$belonging1, survey_f$belonging1)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$belonging1 and survey_f$belonging1
## t = 3.3544, df = 98.292, p-value = 0.001131
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2286667 0.8910744
## sample estimates:
## mean of x mean of y
##  3.893204  3.333333
```

```r
wilcox.test(belonging1 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  belonging1 by gender
## W = 2047.5, p-value = 0.000807
## alternative hypothesis: true location shift is not equal to 0
```

```
t.test(survey_m$belonging2, survey_f$belonging2)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$belonging2 and survey_f$belonging2
## t = 2.5543, df = 106.15, p-value = 0.01206
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1195647 0.9487371
## sample estimates:
## mean of x mean of y
##  3.990291  3.456140
```

```
wilcox.test(belonging2 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  belonging2 by gender
## W = 2255.5, p-value = 0.01088
## alternative hypothesis: true location shift is not equal to 0
```

```
t.test(survey_m$belonging3, survey_f$belonging3)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$belonging3 and survey_f$belonging3
## t = 3.6953, df = 89.834, p-value = 0.0003771
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2761938 0.9184920
## sample estimates:
## mean of x mean of y
##  4.281553  3.684211
```

```
wilcox.test(belonging3 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  belonging3 by gender
## W = 1980, p-value = 0.0002715
## alternative hypothesis: true location shift is not equal to 0
```

```
t.test(survey_m$belonging4, survey_f$belonging4)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$belonging4 and survey_f$belonging4
## t = 1.1981, df = 120.86, p-value = 0.2332
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  -0.1663665  0.6763307
## sample estimates:
## mean of x mean of y
##  3.097087  2.842105
```

```r
wilcox.test(belonging4 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  belonging4 by gender
## W = 2603.5, p-value = 0.2257
## alternative hypothesis: true location shift is not equal to 0
```

```r
t.test(survey_m$belonging5, survey_f$belonging5)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$belonging5 and survey_f$belonging5
## t = 1.6623, df = 102.3, p-value = 0.09951
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.05952398  0.67577334
## sample estimates:
## mean of x mean of y
##  3.466019  3.157895
```

```r
wilcox.test(belonging5 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  belonging5 by gender
## W = 2512, p-value = 0.117
## alternative hypothesis: true location shift is not equal to 0
```

```r
t.test(survey_m$belonging6, survey_f$belonging6)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$belonging6 and survey_f$belonging6
## t = 3.6399, df = 94.758, p-value = 0.0004442
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.3220952 1.0950398
## sample estimates:
## mean of x mean of y
##  4.252427  3.543860
```

```r
wilcox.test(belonging6 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  belonging6 by gender
## W = 1982, p-value = 0.0003035
## alternative hypothesis: true location shift is not equal to 0
```

```r
t.test(survey_m$belonging7, survey_f$belonging7)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$belonging7 and survey_f$belonging7
## t = 0.64998, df = 97.395, p-value = 0.5172
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1836162  0.3624613
## sample estimates:
## mean of x mean of y
##  4.563107  4.473684
```

```r
wilcox.test(belonging7 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  belonging7 by gender
## W = 2834.5, p-value = 0.6696
## alternative hypothesis: true location shift is not equal to 0
```

```r
t.test(survey_m$belonging8, survey_f$belonging8)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$belonging8 and survey_f$belonging8
## t = 1.9701, df = 104.92, p-value = 0.05146
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.002080254  0.647966815
## sample estimates:
## mean of x mean of y
##  4.165049  3.842105
```

```r
wilcox.test(belonging8 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  belonging8 by gender
## W = 2401, p-value = 0.04213
## alternative hypothesis: true location shift is not equal to 0
```

```r
t.test(survey_m$belonging9, survey_f$belonging9)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$belonging9 and survey_f$belonging9
## t = 1.8936, df = 102.78, p-value = 0.06108
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.01720392  0.74382630
## sample estimates:
## mean of x mean of y
##  4.310680  3.947368
```

```r
wilcox.test(belonging9 ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  belonging9 by gender
## W = 2442, p-value = 0.05112
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Prior CS ind test
t.test(survey_m$cs_study_ind, survey_f$cs_study_ind)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$cs_study_ind and survey_f$cs_study_ind
## t = 2.8301, df = 105.4, p-value = 0.005574
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.06803058 0.38640649
## sample estimates:
## mean of x mean of y
## 0.7184466 0.4912281
```

```r
wilcox.test(cs_study_ind ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  cs_study_ind by gender
## W = 2268.5, p-value = 0.004344
## alternative hypothesis: true location shift is not equal to 0
```

```r
# English ind test
t.test(survey_m$native_ind, survey_f$native_ind)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$native_ind and survey_f$native_ind
## t = 0.35897, df = 113.51, p-value = 0.7203
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1293073  0.1865377
## sample estimates:
## mean of x mean of y
## 0.6601942 0.6315789
```

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

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  native_ind by gender
## W = 2851.5, p-value = 0.7188
## alternative hypothesis: true location shift is not equal to 0
```

```r
# US birth ind test
t.test(survey_m$us_birth_ind, survey_f$us_birth_ind)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$us_birth_ind and survey_f$us_birth_ind
## t = -0.40755, df = 116.37, p-value = 0.6844
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1956207  0.1288519
## sample estimates:
## mean of x mean of y
## 0.5631068 0.5964912
```

```r
wilcox.test(us_birth_ind ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  us_birth_ind by gender
## W = 3033.5, p-value = 0.6849
## alternative hypothesis: true location shift is not equal to 0
```

```r
# US residence ind test
t.test(survey_m$us_res_ind, survey_f$us_res_ind)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$us_res_ind and survey_f$us_res_ind
## t = -0.23651, df = 119.58, p-value = 0.8134
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1229137  0.0966831
## sample estimates:
## mean of x mean of y
## 0.8640777 0.8771930
```

```r
wilcox.test(us_res_ind ~ gender, data=survey)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  us_res_ind by gender
## W = 2974, p-value = 0.8169
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Higher education ind test
t.test(survey_m$higher_ind, survey_f$higher_ind)
```

```
##
##  Welch Two Sample t-test
##
## data:  survey_m$higher_ind and survey_f$higher_ind
## t = -0.79991, df = 109.75, p-value = 0.4255
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.21382942  0.09085207
## sample estimates:
## mean of x mean of y 
## 0.2718447 0.3333333
```

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

```
## 
##  Wilcoxon rank sum test with continuity correction
## 
## data:  higher_ind by gender
## W = 3116, p-value = 0.4163
## alternative hypothesis: true location shift is not equal to 0
```

Regression Analysis

```r
# Check for multicollinearity
cor_subset = survey[, c("age_num", "prior_cs_num", "native_ind", "higher_ind",
                        "cs_study_ind", "completed")]
cor(na.omit(cor_subset))
```

```
##                  age_num prior_cs_num   native_ind  higher_ind
## age_num       1.00000000   0.53998873  0.251844563  0.16356165
## prior_cs_num  0.53998873   1.00000000  0.198503723 -0.06523606
## native_ind    0.25184456   0.19850372  1.000000000  0.01294594
## higher_ind    0.16356165  -0.06523606  0.012945942  1.00000000
## cs_study_ind -0.03466723   0.17810433 -0.008177423 -0.11310636
## completed     0.07868277   0.02230533 -0.023849306  0.14048204
##              cs_study_ind    completed
## age_num      -0.034667233  0.078682766
## prior_cs_num  0.178104327  0.022305335
## native_ind   -0.008177423 -0.023849306
## higher_ind   -0.113106360  0.140482039
## cs_study_ind  1.000000000 -0.008282098
## completed    -0.008282098  1.000000000
```

```r
# Fit regression to conf_ave
conf_lm = lm(conf_ave~gender + age_num + cs_study_ind + native_ind +  higher_ind +
               completed, data=survey)

summary(conf_lm)
```

```
## 
## Call:
## lm(formula = conf_ave ~ gender + age_num + cs_study_ind + native_ind + 
##     higher_ind + completed, data = survey)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.76895 -0.42732  0.09353  0.50667  1.37922 
## 
## Coefficients:
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.187252   0.255174  12.491  < 2e-16 ***
## genderMale     0.461281   0.122933   3.752 0.000248 ***
## age_num        0.004830   0.006286   0.768 0.443435
## cs_study_ind   0.141303   0.121246   1.165 0.245658
## native_ind     0.041680   0.122395   0.341 0.733918
## higher_ind    -0.087922   0.127791  -0.688 0.492486
## completed      0.069739   0.020992   3.322 0.001118 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7137 on 153 degrees of freedom
## Multiple R-squared:  0.1899, Adjusted R-squared:  0.1582
## F-statistic: 5.978 on 6 and 153 DF,  p-value: 1.231e-05
```

```
# Fit regression to conf_prior
conf_prior_lm = lm(conf_prior~gender + age_num + cs_study_ind + native_ind +  higher_ind +
               completed, data=survey)

summary(conf_prior_lm)
```

```
##
## Call:
## lm(formula = conf_prior ~ gender + age_num + cs_study_ind + native_ind +
##     higher_ind + completed, data = survey)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4295 -0.4496  0.0261  0.6611  1.9662
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.559206   0.332179   7.704 1.54e-12 ***
## genderMale     0.665501   0.160031   4.159 5.32e-05 ***
## age_num        0.012419   0.008183   1.518  0.13117
## cs_study_ind   0.443838   0.157834   2.812  0.00557 **
## native_ind    -0.064494   0.159331  -0.405  0.68620
## higher_ind    -0.085471   0.166355  -0.514  0.60814
## completed      0.037983   0.027327   1.390  0.16658
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.929 on 153 degrees of freedom
## Multiple R-squared:  0.2097, Adjusted R-squared:  0.1787
## F-statistic: 6.767 on 6 and 153 DF,  p-value: 2.235e-06
```

```
# Fit regression to conf_post
conf_post_lm = lm(conf_post~gender + age_num + cs_study_ind + native_ind +  higher_ind +
               completed, data=survey)

summary(conf_post_lm)
```

```
##
## Call:
## lm(formula = conf_post ~ gender + age_num + cs_study_ind + native_ind +
##     higher_ind + completed, data = survey)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3441 -0.3493  0.2026  0.4908  1.3189
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.815297   0.283886  13.440  < 2e-16 ***
## genderMale    0.257061   0.136766   1.880   0.0621 .
## age_num      -0.002759   0.006994  -0.394   0.6938
## cs_study_ind -0.161232   0.134888  -1.195   0.2338
## native_ind    0.147855   0.136167   1.086   0.2793
## higher_ind   -0.090373   0.142170  -0.636   0.5259
## completed     0.101494   0.023355   4.346 2.52e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.794 on 153 degrees of freedom
## Multiple R-squared:  0.1483, Adjusted R-squared:  0.1149
## F-statistic: 4.441 on 6 and 153 DF,  p-value: 0.0003596
```

```r
# Fit regression to selfconf_score
selfconf_lm = lm(selfconf_score~gender + age_num + cs_study_ind + native_ind +
                   higher_ind + completed, data=survey)

summary(selfconf_lm)
```

```
##
## Call:
## lm(formula = selfconf_score ~ gender + age_num + cs_study_ind +
##     native_ind + higher_ind + completed, data = survey)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.35767 -0.42014  0.02469  0.47805  1.24051
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.282873   0.221743  14.805  < 2e-16 ***
## genderMale    0.285781   0.106828   2.675  0.00828 **
## age_num       0.012052   0.005463   2.206  0.02886 *
## cs_study_ind  0.126062   0.105361   1.196  0.23336
## native_ind   -0.009847   0.106360  -0.093  0.92636
## higher_ind   -0.129715   0.111049  -1.168  0.24459
## completed     0.029219   0.018242   1.602  0.11129
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6202 on 153 degrees of freedom
## Multiple R-squared:  0.1275, Adjusted R-squared:  0.09325
## F-statistic: 3.725 on 6 and 153 DF,  p-value: 0.001738
```

```r
# Fit regression to equality_score
equality_lm = lm(equality_score~gender + age_num + cs_study_ind + native_ind +
                   higher_ind + completed, data=survey)
```

```
summary(equality_lm)
```

```
##
## Call:
## lm(formula = equality_score ~ gender + age_num + cs_study_ind +
##     native_ind + higher_ind + completed, data = survey)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.93547 -0.24829  0.09629  0.39943  0.98015
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.225450   0.207542  20.360  < 2e-16 ***
## genderMale   -0.202636   0.099986  -2.027 0.044436 *
## age_num       0.009158   0.005113   1.791 0.075242 .
## cs_study_ind -0.012696   0.098613  -0.129 0.897725
## native_ind    0.213989   0.099548   2.150 0.033159 *
## higher_ind   -0.387740   0.103937  -3.731 0.000269 ***
## completed    -0.005493   0.017074  -0.322 0.748088
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5804 on 153 degrees of freedom
## Multiple R-squared:  0.1448, Adjusted R-squared:  0.1112
## F-statistic: 4.317 on 6 and 153 DF,  p-value: 0.0004725
```

```
# Fit regression to belonging_score
belonging_lm = lm(belonging_score~gender + age_num + cs_study_ind + native_ind +
                    higher_ind + completed, data=survey)

summary(belonging_lm)
```

```
##
## Call:
## lm(formula = belonging_score ~ gender + age_num + cs_study_ind +
##     native_ind + higher_ind + completed, data = survey)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.53731 -0.40095  0.05514  0.46888  1.43403
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.642155   0.240301  15.157  < 2e-16 ***
## genderMale    0.425493   0.115768   3.675 0.000328 ***
## age_num       0.004062   0.005920   0.686 0.493619
## cs_study_ind -0.062131   0.114179  -0.544 0.587125
## native_ind   -0.047027   0.115261  -0.408 0.683840
## higher_ind   -0.172517   0.120343  -1.434 0.153743
## completed    -0.016933   0.019769  -0.857 0.393025
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.6721 on 153 degrees of freedom
## Multiple R-squared:  0.103,  Adjusted R-squared:  0.06783
## F-statistic: 2.928 on 6 and 153 DF,  p-value: 0.009864
```

```
# Fit regression to study hours
hours_lm = lm(hours_num~gender + age_num + cs_study_ind + native_ind +  higher_ind +
               completed, data=survey)

summary(hours_lm)
```

```
##
## Call:
## lm(formula = hours_num ~ gender + age_num + cs_study_ind + native_ind +
##     higher_ind + completed, data = survey)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.278  -7.547  -1.897   4.795  22.243
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  18.545459   3.281607   5.651 7.57e-08 ***
## genderMale   -1.713133   1.580958  -1.084    0.280
## age_num       0.003752   0.080845   0.046    0.963
## cs_study_ind  1.201520   1.559254   0.771    0.442
## native_ind    0.429752   1.574035   0.273    0.785
## higher_ind    1.167708   1.643429   0.711    0.478
## completed     0.238426   0.269969   0.883    0.379
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.178 on 153 degrees of freedom
## Multiple R-squared:  0.01803,    Adjusted R-squared:  -0.02048
## F-statistic: 0.4682 on 6 and 153 DF,  p-value: 0.8311
```

```
# Fit regression to number of years' programming experience
prog_lm = lm(prog_num~gender + age_num + cs_study_ind + native_ind +  higher_ind +
               completed, data=survey)

summary(prog_lm)
```

```
##
## Call:
## lm(formula = prog_num ~ gender + age_num + cs_study_ind + native_ind +
##     higher_ind + completed, data = survey)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.783  -3.540  -0.084   3.716  10.009
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -5.394168   1.827030  -2.952  0.00365 **
## genderMale    2.039854   0.880196   2.317  0.02180 *
## age_num       0.286103   0.045010   6.356 2.26e-09 ***
## cs_study_ind  2.168805   0.868112   2.498  0.01354 * 
## native_ind    1.641806   0.876342   1.873  0.06291 . 
## higher_ind   -1.429879   0.914977  -1.563  0.12018   
## completed     0.006058   0.150305   0.040  0.96790   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 5.11 on 153 degrees of freedom
## Multiple R-squared:  0.3231, Adjusted R-squared:  0.2965 
## F-statistic: 12.17 on 6 and 153 DF,  p-value: 3.639e-11
```

```
# Fit regression to number of years' cs experience
prior_cs_lm = lm(prior_cs_num~gender + age_num + cs_study_ind + native_ind +  higher_ind +
               completed, data=survey)

summary(prior_cs_lm)
```

```
## 
## Call:
## lm(formula = prior_cs_num ~ gender + age_num + cs_study_ind + 
##     native_ind + higher_ind + completed, data = survey)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -9.7387 -2.6242 -0.1886  2.9593  8.8372 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  -6.47620    1.48361  -4.365 2.33e-05 ***
## genderMale    1.49010    0.71475   2.085   0.0387 *  
## age_num       0.29106    0.03655   7.963 3.54e-13 ***
## cs_study_ind  1.60965    0.70493   2.283   0.0238 *  
## native_ind    0.65563    0.71162   0.921   0.3583    
## higher_ind   -1.40946    0.74299  -1.897   0.0597 .  
## completed    -0.03303    0.12205  -0.271   0.7871    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 4.149 on 153 degrees of freedom
## Multiple R-squared:  0.3702, Adjusted R-squared:  0.3455 
## F-statistic: 14.99 on 6 and 153 DF,  p-value: 1.908e-13
```

```
# Fit regression to gpa
gpa_lm = lm(gpa_num~gender + age_num + cs_study_ind + native_ind +  higher_ind +
               completed, data=survey)

summary(gpa_lm)
```

```
## 
## Call:
## lm(formula = gpa_num ~ gender + age_num + cs_study_ind + native_ind + 
##     higher_ind + completed, data = survey)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.20517 -0.06386  0.04777  0.20549  0.39627
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7806436  0.1113890  33.941   <2e-16 ***
## genderMale   -0.0537681  0.0527781  -1.019   0.3100
## age_num       0.0005714  0.0027637   0.207   0.8365
## cs_study_ind  0.0195166  0.0516386   0.378   0.7060
## native_ind    0.0895858  0.0528644   1.695   0.0923 .
## higher_ind   -0.0223444  0.0544762  -0.410   0.6823
## completed    -0.0174206  0.0093688  -1.859   0.0650 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2991 on 145 degrees of freedom
##   (8 observations deleted due to missingness)
## Multiple R-squared:  0.05422,    Adjusted R-squared:  0.01509
## F-statistic: 1.386 on 6 and 145 DF,  p-value: 0.2242
```

```
# Fit logistic regression to prior cs study indicator
cs_study_lm = glm(cs_study_ind~gender + age_num + cs_study_ind + native_ind +
                higher_ind + completed, data=survey, family=binomial())
```

```
## Warning in model.matrix.default(mt, mf, contrasts): the response appeared
## on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(mt, mf, contrasts): problem with term 3 in
## model.matrix: no columns are assigned
```

```
summary(cs_study_lm)
```

```
##
## Call:
## glm(formula = cs_study_ind ~ gender + age_num + cs_study_ind +
##     native_ind + higher_ind + completed, family = binomial(),
##     data = survey)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7166  -1.2108   0.7541   0.9207   1.4221
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.501061   0.727559   0.689  0.49102
## genderMale   0.987078   0.352579   2.800  0.00512 **
## age_num     -0.008923   0.018938  -0.471  0.63753
## native_ind  -0.014307   0.368752  -0.039  0.96905
## higher_ind  -0.417569   0.374678  -1.114  0.26508
## completed   -0.017076   0.063235  -0.270  0.78713
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 209.55  on 159  degrees of freedom
## Residual deviance: 199.51  on 154  degrees of freedom
## AIC: 211.51
##
## Number of Fisher Scoring iterations: 4
```