Playing with CGPA dataset

Load required packages

```
library(readxl)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.5.2

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(psych) # describe()
```

```
## Warning: package 'psych' was built under R version 4.5.2
```

```
library(skimr) # skim()
```

```
## Warning: package 'skimr' was built under R version 4.5.2
```

```
library(pastecs) # stat.desc()
```

```
## Warning: package 'pastecs' was built under R version 4.5.2

##
## Attaching package: 'pastecs'

## The following objects are masked from 'package:dplyr':
##
##     first, last
```

Load and see the data

```r
df <- read_excel("C:\\Users\\Dell\\OneDrive\\Desktop\\STATISTICS\\2nd Year\\R programming\\varsity\\Pro
describe(df)
```

```
##                   vars  n       mean    sd     median    trimmed   mad
## ExamRoll             1 80 2316641.96 23.91 2316642.50 2316642.08 30.39
## Sex*                 2 80       1.62  0.49       2.00       1.66  0.00
## CGPA                 3 80       3.42  0.44       3.48       3.48  0.39
## Group                4 80       2.28  1.03       2.00       2.22  1.48
## AssignedTeacher*     5 80      11.18  6.64      10.50      11.02  8.15
##                          min        max range  skew kurtosis   se
## ExamRoll          2316601.00 2316682.00 81.00 -0.04    -1.24 2.67
## Sex*                    1.00       2.00  1.00 -0.51    -1.76 0.05
## CGPA                    2.05       3.99  1.94 -1.22     1.14 0.05
## Group                   1.00       4.00  3.00  0.19    -1.18 0.12
## AssignedTeacher*        1.00      23.00 22.00  0.18    -1.21 0.74
```

```r
skim(df)
```

Table 1: Data summary

Name	df
Number of rows	80
Number of columns	5
Column type frequency:	
character	2
numeric	3
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Sex	0	1	1	1	0	2	0
AssignedTeacher	0	1	1	2	0	23	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
ExamRoll	0	1	2316641.96	23.91	2316601.00	2316621.75	2316642.50	2316662.25	2316682.00	
CGPA	0	1	3.42	0.44	2.05	3.24	3.48	3.77	3.99	
Group	0	1	2.28	1.03	1.00	1.00	2.00	3.00	4.00	

```r
glimpse(df)
```

```
## Rows: 80
## Columns: 5
## $ ExamRoll        <dbl> 2316669, 2316632, 2316615, 2316643, 2316679, 2316667, ~
```

```
## $ Sex             <chr> "F", "M", "M", "M", "F", "F", "M", "M", "M", "F", "F",~
## $ CGPA            <dbl> 3.988, 3.953, 3.938, 3.938, 3.914, 3.910, 3.864, 3.857~
## $ Group           <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ AssignedTeacher <chr> "B", "O", "C", "FT", "MT", "T", "K", "I", "E", "G", "W~
```

```r
stat.desc(df)
```

```
##                  ExamRoll Sex         CGPA       Group AssignedTeacher
## nbr.val      8.000000e+01  NA  80.00000000  80.0000000              NA
## nbr.null     0.000000e+00  NA   0.00000000   0.0000000              NA
## nbr.na       0.000000e+00  NA   0.00000000   0.0000000              NA
## min          2.316601e+06  NA   2.05300000   1.0000000              NA
## max          2.316682e+06  NA   3.98800000   4.0000000              NA
## range        8.100000e+01  NA   1.93500000   3.0000000              NA
## sum          1.853314e+08  NA 273.21800000 182.0000000              NA
## median       2.316643e+06  NA   3.48400000   2.0000000              NA
## mean         2.316642e+06  NA   3.41522500   2.2750000              NA
## SE.mean      2.673324e+00  NA   0.04932226   0.1152529              NA
## CI.mean.0.95 5.321118e+00  NA   0.09817350   0.2294051              NA
## var          5.717328e+02  NA   0.19461483   1.0626582              NA
## std.dev      2.391093e+01  NA   0.44115171   1.0308532              NA
## coef.var     1.032138e-05  NA   0.12917208   0.4531223              NA
```

```r
summary(df) # Exam Roll numbers, Sex, CGPA, Group assignment, and Assigned Teacher
```

```
##     ExamRoll           Sex                 CGPA           Group
##  Min.   :2316601   Length:80          Min.   :2.053   Min.   :1.000
##  1st Qu.:2316622   Class :character   1st Qu.:3.242   1st Qu.:1.000
##  Median :2316643   Mode  :character   Median :3.484   Median :2.000
##  Mean   :2316642                      Mean   :3.415   Mean   :2.275
##  3rd Qu.:2316662                      3rd Qu.:3.769   3rd Qu.:3.000
##  Max.   :2316682                      Max.   :3.988   Max.   :4.000
##  AssignedTeacher
##  Length:80
##  Class :character
##  Mode  :character
##
##
##
```

```r
sum(is.na(df)) # no Missing values
```

```
## [1] 0
```

Convert variable types

```r
df$ExamRoll<-as.character(df$ExamRoll)
class(df$ExamRoll)
```

```
## [1] "character"
```

```r
df$Sex <- as.factor(df$Sex)
class(df$Sex)
```

```
## [1] "factor"
```

```r
df$Group <- as.factor(df$Group)
class(df$Group)
```

```
## [1] "factor"
```

```r
df$AssignedTeacher <- as.factor(df$AssignedTeacher)
class(df$AssignedTeacher)
```

```
## [1] "factor"
```

See the data again

```r
str(df)
```

```
## tibble [80 x 5] (S3: tbl_df/tbl/data.frame)
##  $ ExamRoll       : chr [1:80] "2316669" "2316632" "2316615" "2316643" ...
##  $ Sex            : Factor w/ 2 levels "F","M": 1 2 2 2 1 1 2 2 2 1 ...
##  $ CGPA           : num [1:80] 3.99 3.95 3.94 3.94 3.91 ...
##  $ Group          : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
##  $ AssignedTeacher: Factor w/ 23 levels "A","B","C","D",..: 2 15 3 6 13 20 11 9 5 7 ...
```

```r
summary(df)
```

```
##    ExamRoll          Sex         CGPA        Group  AssignedTeacher
##  Length:80          F:30   Min.   :2.053   1:23   A      : 4
##  Class :character   M:50   1st Qu.:3.242   2:23   B      : 4
##  Mode  :character          Median :3.484   3:23   C      : 4
##                            Mean   :3.415   4:11   D      : 4
##                            3rd Qu.:3.769          E      : 4
##                            Max.   :3.988          FT     : 4
##                                                   (Other):56
```

Descriptive Statistics

For CGPA (i) Mean (ii) Median (iii) Standard deviation

```r
describe(df$CGPA)
```

```
##    vars  n mean   sd median trimmed  mad  min  max range  skew kurtosis   se
## X1    1 80 3.42 0.44   3.48    3.48 0.39 2.05 3.99  1.94 -1.22     1.14 0.05
```

Frequency Distributions (i) Sex (ii) Group (iii) AssignedTeacher

```r
table(df$Sex)
```

```
##
##  F  M
## 30 50
```

```r
table(df$Group)
```

```
##
##  1  2  3  4
## 23 23 23 11
```

```r
table(df$AssignedTeacher)
```

```
##
##  A  B  C  D  E FT  G  H  I  J  K  L MT  N  O  P  Q  R  S  T  U  V  W
##  4  4  4  4  4  4  4  4  4  4  4  3  3  3  3  3  3  3  3  3  3  3  3
```

```r
df %>%
  count(df$Sex) %>%  # Frequency
  mutate(
    percent = n / sum(n) * 100,
    cum_n = cumsum(n),
    cum_percent = cumsum(percent),
    relative_freq = n / sum(n),
    cum_relative = cumsum(relative_freq)
  ) %>%
  rename(Frequency = n)
```

```
## # A tibble: 2 x 7
##   `df$Sex` Frequency percent cum_n cum_percent relative_freq cum_relative
##   <fct>        <int>   <dbl> <int>       <dbl>         <dbl>        <dbl>
## 1 F               30    37.5    30        37.5         0.375        0.375
## 2 M               50    62.5    80       100           0.625        1
```

```r
df %>%
  count(df$Group) %>%  # Frequency
  mutate(
    percent = n / sum(n) * 100,
    cum_n = cumsum(n),
    cum_percent = cumsum(percent),
    relative_freq = n / sum(n),
    cum_relative = cumsum(relative_freq)
  ) %>%
  rename(Frequency = n)
```

```
## # A tibble: 4 x 7
##   `df$Group` Frequency percent cum_n cum_percent relative_freq cum_relative
##   <fct>          <int>   <dbl> <int>       <dbl>         <dbl>        <dbl>
## 1 1                 23    28.7    23        28.7         0.288        0.288
## 2 2                 23    28.7    46        57.5         0.288        0.575
## 3 3                 23    28.7    69        86.2         0.288        0.862
## 4 4                 11    13.8    80       100           0.138        1
```

```r
df %>%
  count(df$AssignedTeacher) %>%  # Frequency
  mutate(
    percent = n / sum(n) * 100,
    cum_n = cumsum(n),
    cum_percent = cumsum(percent),
    relative_freq = n / sum(n),
    cum_relative = cumsum(relative_freq)
  ) %>%
  rename(Frequency = n)
```

```
## # A tibble: 23 x 7
##    `df$AssignedTeacher` Frequency percent cum_n cum_percent relative_freq
##    <fct>                    <int>   <dbl> <int>       <dbl>         <dbl>
##  1 A                            4       5     4           5          0.05
##  2 B                            4       5     8          10          0.05
##  3 C                            4       5    12          15          0.05
##  4 D                            4       5    16          20          0.05
##  5 E                            4       5    20          25          0.05
##  6 FT                           4       5    24          30          0.05
##  7 G                            4       5    28          35          0.05
##  8 H                            4       5    32          40          0.05
##  9 I                            4       5    36          45          0.05
## 10 J                            4       5    40          50          0.05
## # i 13 more rows
## # i 1 more variable: cum_relative <dbl>
```

Group Comparisons (i) CGPA by Sex (mean, SD, count) (ii) CGPA by Group (mean, SD, count)

```r
# Using dplyr (recommended)
result <- df %>%
  group_by(Sex) %>%
  summarise(
    Count = n(),
    Mean_CGPA = round(mean(CGPA, na.rm = TRUE), 2),
    SD_CGPA = round(sd(CGPA, na.rm = TRUE), 2),
    .groups = 'drop'
  )
result
```

```
## # A tibble: 2 x 4
##   Sex   Count Mean_CGPA SD_CGPA
##   <fct> <int>     <dbl>   <dbl>
## 1 F        30      3.52    0.38
## 2 M        50      3.35    0.47
```

```r
# knitr::kable(result, caption = "CGPA by Sex") # Display nicely

result <- df %>%
  group_by(Group) %>%
  summarise(
    Count = n(),
```

```
    Mean_CGPA = round(mean(CGPA, na.rm = TRUE), 2),
    SD_CGPA = round(sd(CGPA, na.rm = TRUE), 2),
    .groups = 'drop'
  )
result
```

```
## # A tibble: 4 x 4
##   Group Count Mean_CGPA SD_CGPA
##   <fct> <int>     <dbl>   <dbl>
## 1 1        23      3.84    0.07
## 2 2        23      3.57    0.09
## 3 3        23      3.26    0.13
## 4 4        11      2.54    0.32
```

```
# For everyday use: dplyr (most readable and flexible)

# For quick analysis: psych::describeBy()

# For large datasets: data.table

# For publication: gtsummary
```

Data Visualization

Distribution Plots (i) Histogram of CGPA (ii) Density plot of CGPA

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.5.2
```

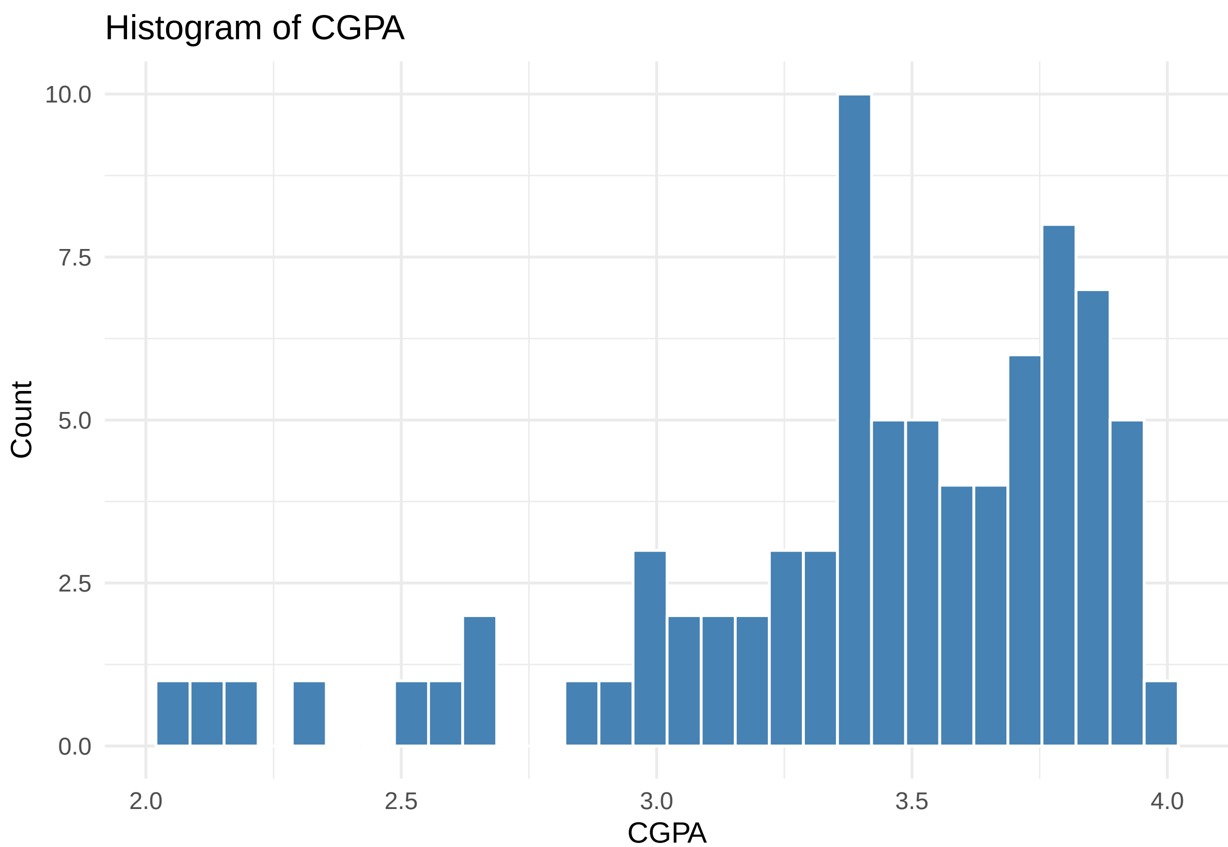
```
##
## Attaching package: 'ggplot2'
```

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

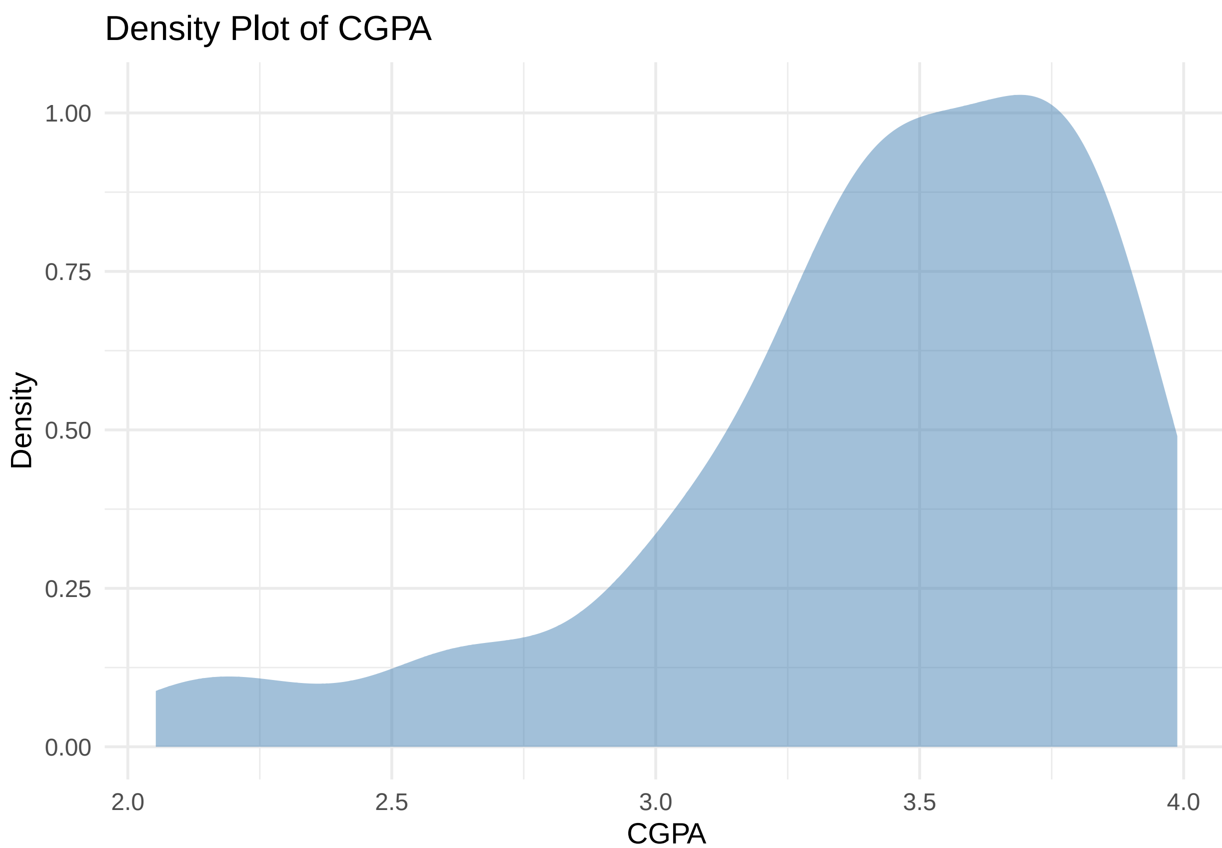
```
# install.packages("esquisse")
# library(esquisse) # Creates ggplot2 code automatically
# esquisser(df) # Opens drag-and-drop interface
```

```
# Template for any plot
#ggplot(data = df, aes(x = x_variable, y = y_variable)) +
#  geom_*(aes(color/fill = grouping_variable)) +  # Choose geom
#  labs(title = "Title", x = "X Label", y = "Y Label") +
#  theme_minimal() +
#  scale_fill_brewer(palette = "Set2")  # Color palette
```

```r
# Histogram of CGPA
ggplot(df, aes(x = CGPA)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "white") +
  labs(title = "Histogram of CGPA", x = "CGPA", y = "Count") +
  theme_minimal()
```



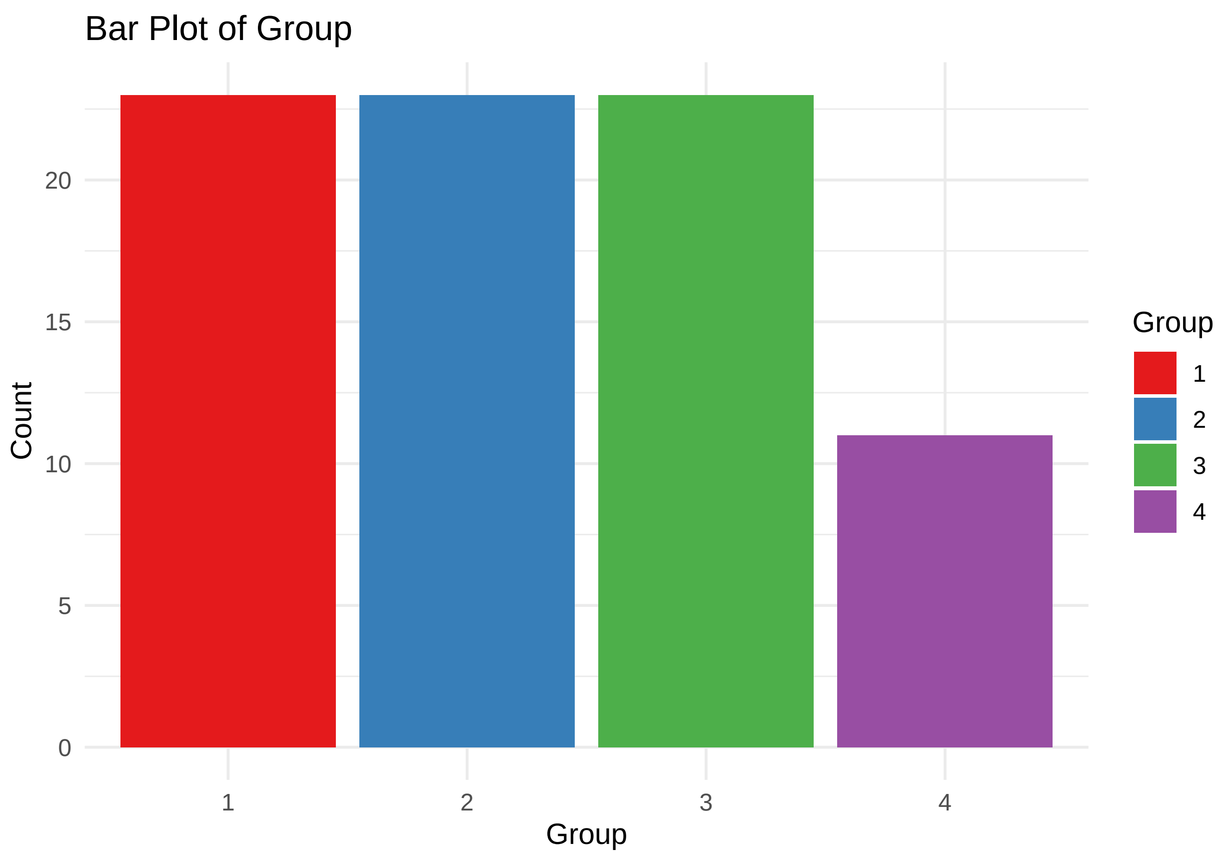

```r
# Density plot of CGPA
ggplot(df, aes(x = CGPA)) +
  geom_density(fill = "steelblue", alpha = 0.5, color = NA) +
  labs(title = "Density Plot of CGPA", x = "CGPA", y = "Density") +
  theme_minimal()
```



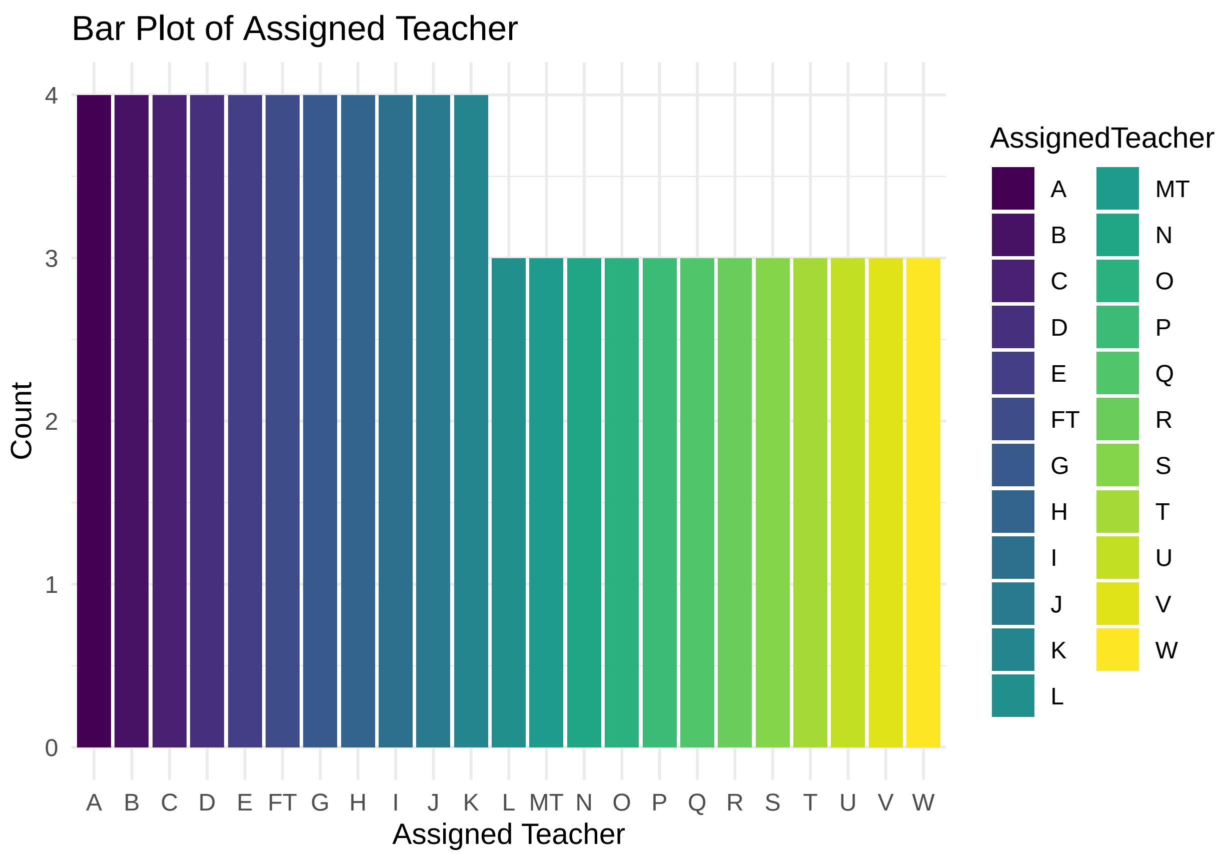
Categorical Data Plots: (i) Bar plot of Sex (ii) Bar plot of Group (iii) Bar plot of AssignedTeacher

```
# Bar plot of Sex
ggplot(df, aes(x = Sex, fill = Sex, group = Sex)) +
  geom_bar() +
  labs(title = "Bar Plot of Sex", x = "Sex", y = "Count") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")
```


```r
# Bar plot of Group
ggplot(df, aes(x = Group, fill = Group, group = Group)) +
  geom_bar() +
  labs(title = "Bar Plot of Group", x = "Group", y = "Count") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set1")
```

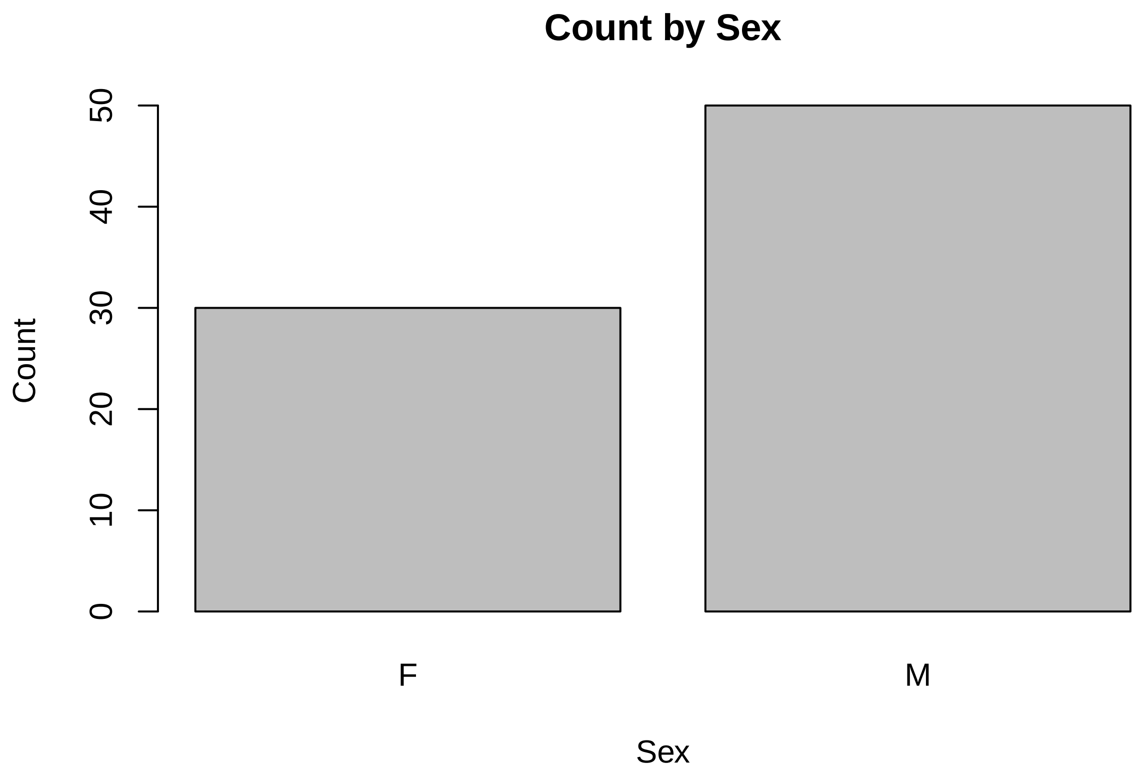



```r
# Bar plot of Assigned Teacher
ggplot(df, aes(x = AssignedTeacher, fill = AssignedTeacher, group = AssignedTeacher)) +
  geom_bar() +
  labs(title = "Bar Plot of Assigned Teacher",
       x = "Assigned Teacher", y = "Count") +
  theme_minimal() +
  scale_fill_viridis_d()        # many categories → no Brewer palette
```

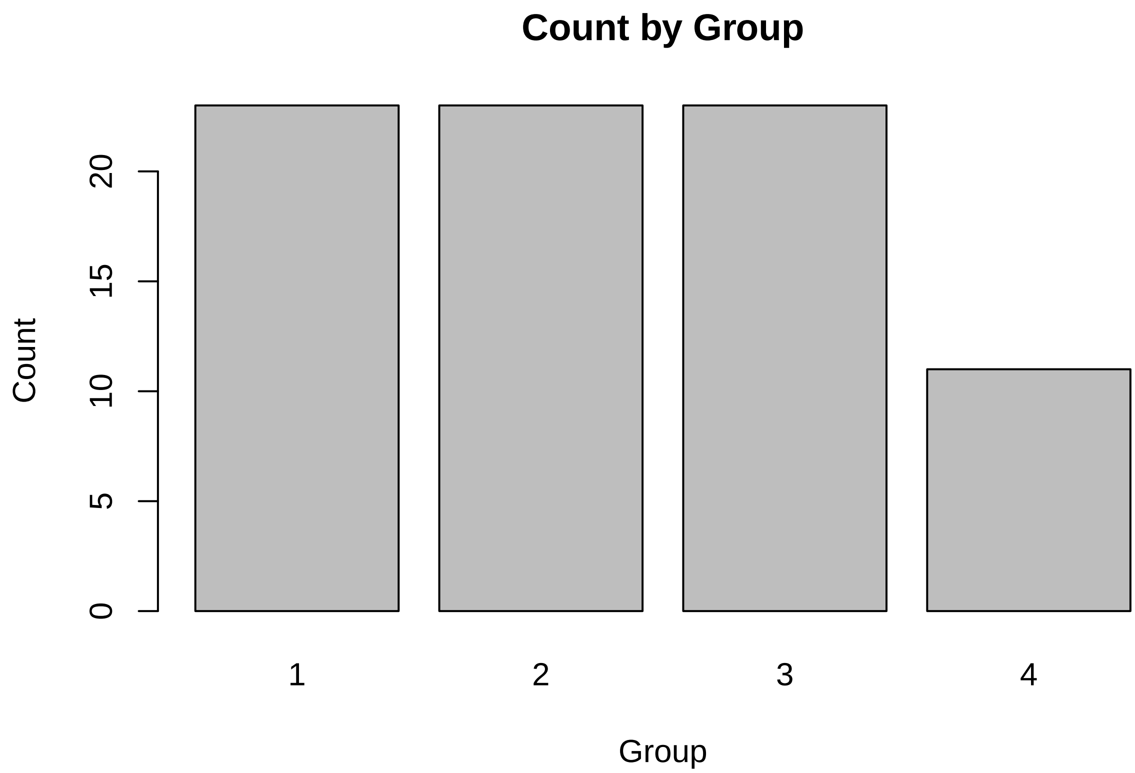


Easier Way

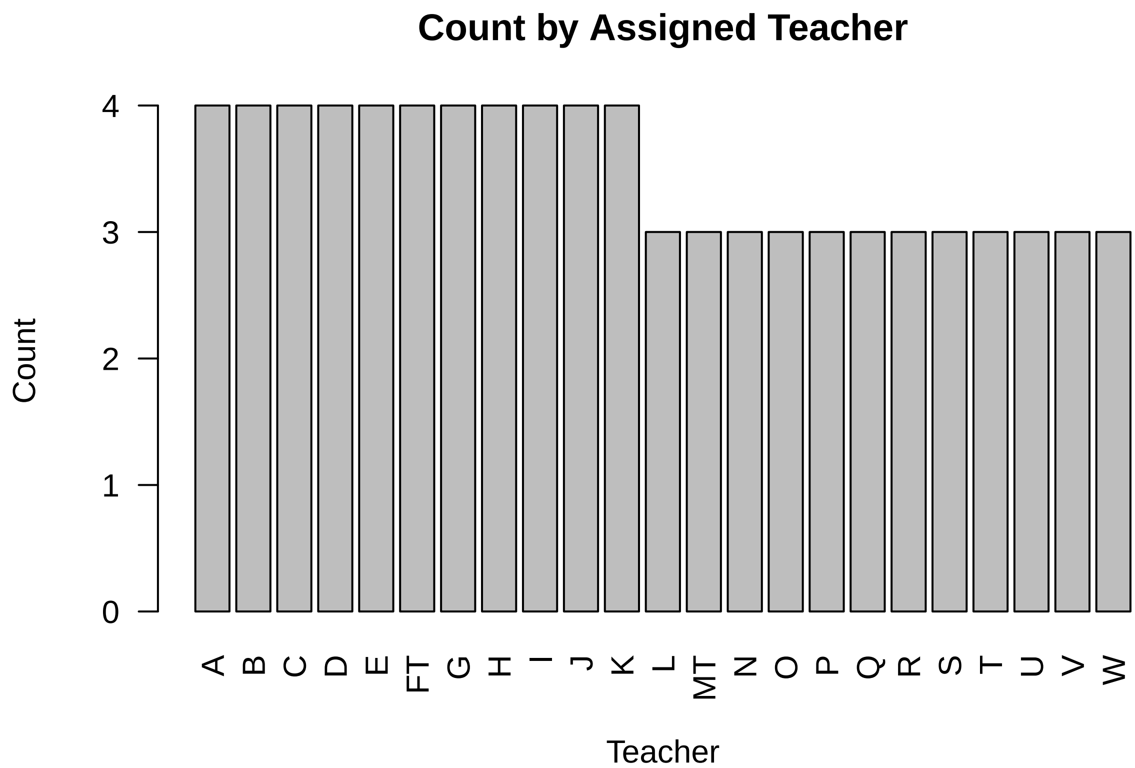
```
# Bar plot of Sex
barplot(table(df$Sex), main = "Count by Sex", xlab = "Sex", ylab = "Count")
```




```r
# Bar plot of Group
barplot(table(df$Group), main = "Count by Group", xlab = "Group", ylab = "Count")
```



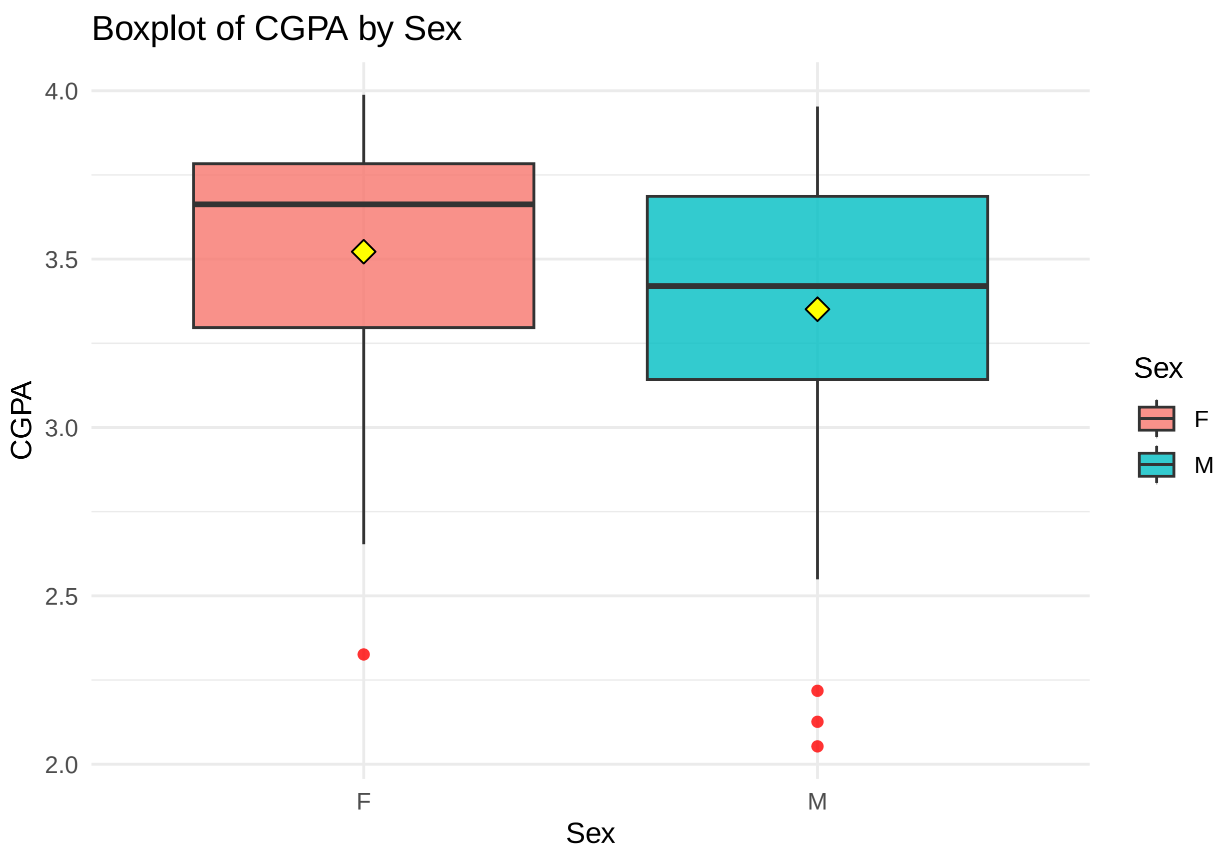

```r
# Bar plot of AssignedTeacher
barplot(table(df$AssignedTeacher), main = "Count by Assigned Teacher",
        xlab = "Teacher", ylab = "Count", las = 2)
```



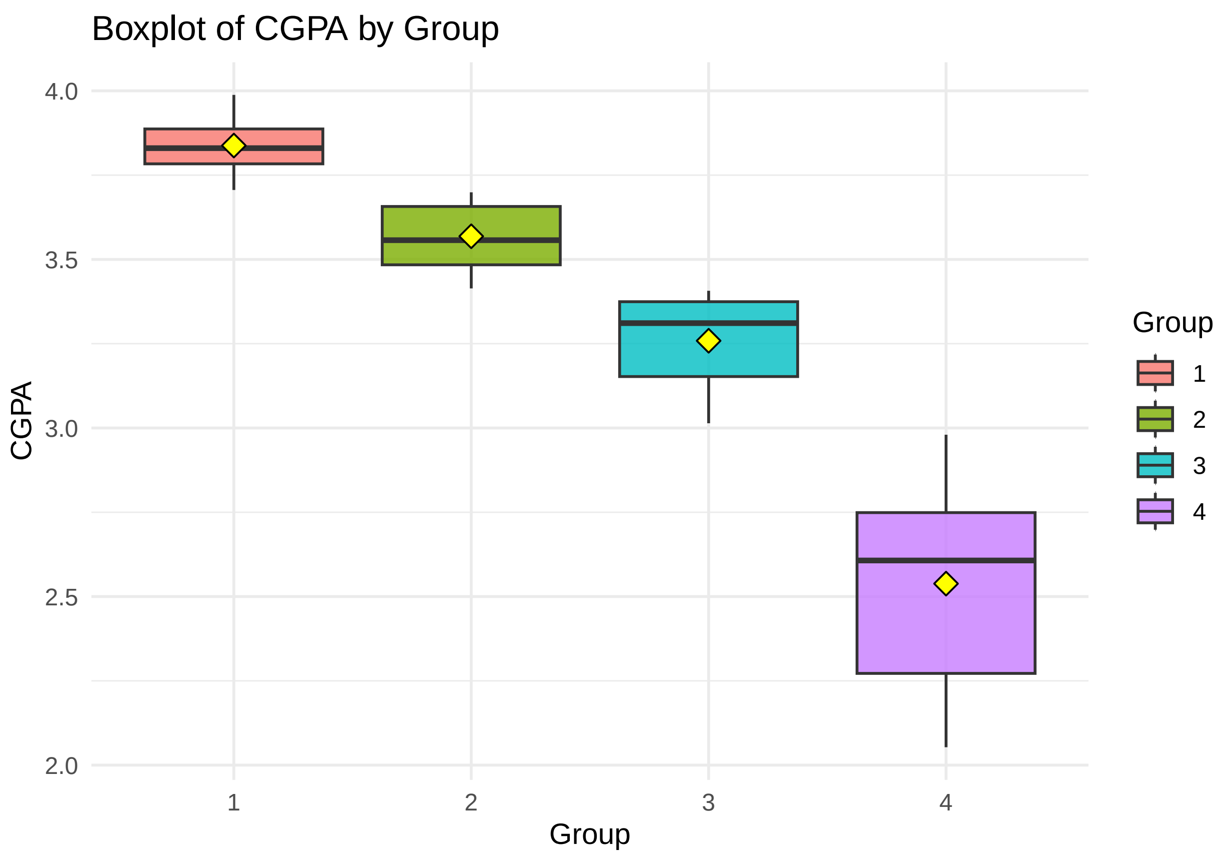
Comparative Plots: (i) Boxplot of CGPA by Sex (ii) Boxplot of CGPA by Group

```
library(ggplot2)

ggplot(df, aes(x = Sex, y = CGPA, fill = Sex)) +
  geom_boxplot(alpha = 0.8, outlier.color = "red") +
  stat_summary(fun = mean,
               geom = "point",
               shape = 23,
               size = 3,
               fill = "yellow") +
  labs(title = "Boxplot of CGPA by Sex",
       x = "Sex",
       y = "CGPA") +
  theme_minimal()
```

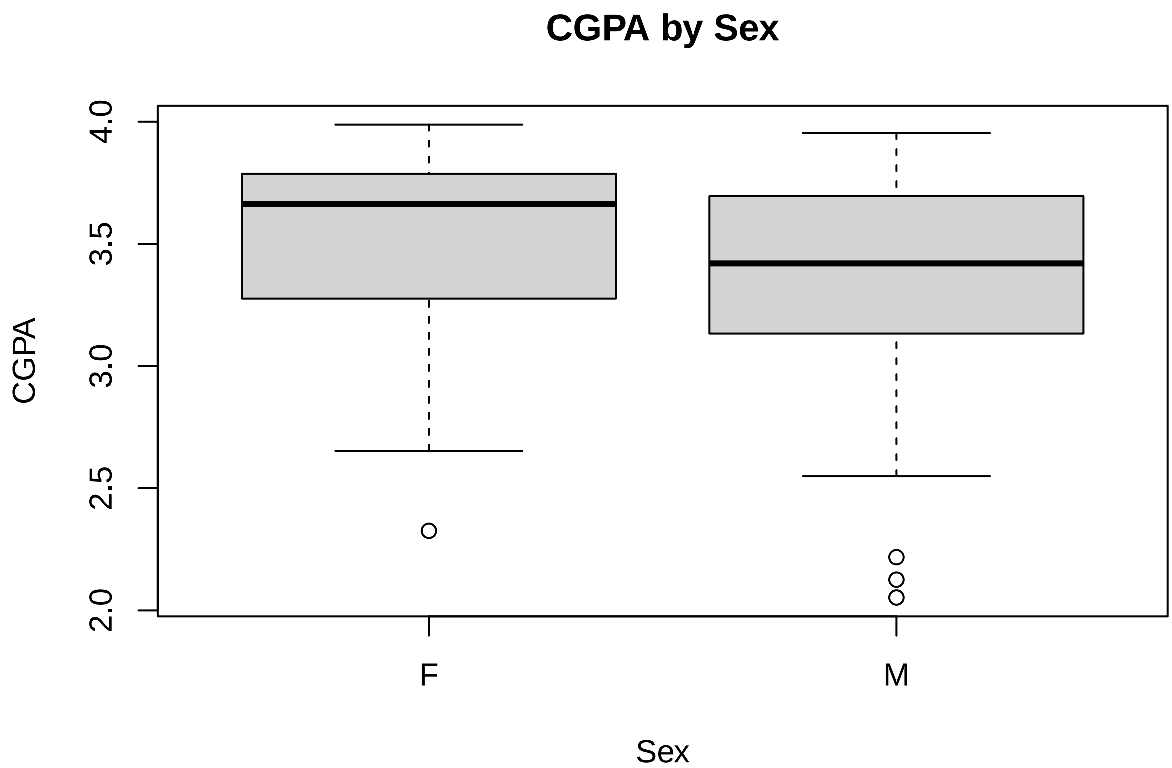



```r
ggplot(df, aes(x = Group, y = CGPA, fill = Group)) +
  geom_boxplot(alpha = 0.8, outlier.color = "red") +
  stat_summary(fun = mean,
               geom = "point",
               shape = 23,
               size = 3,
               fill = "yellow") +
  labs(title = "Boxplot of CGPA by Group",
       x = "Group",
       y = "CGPA") +
  theme_minimal()
```

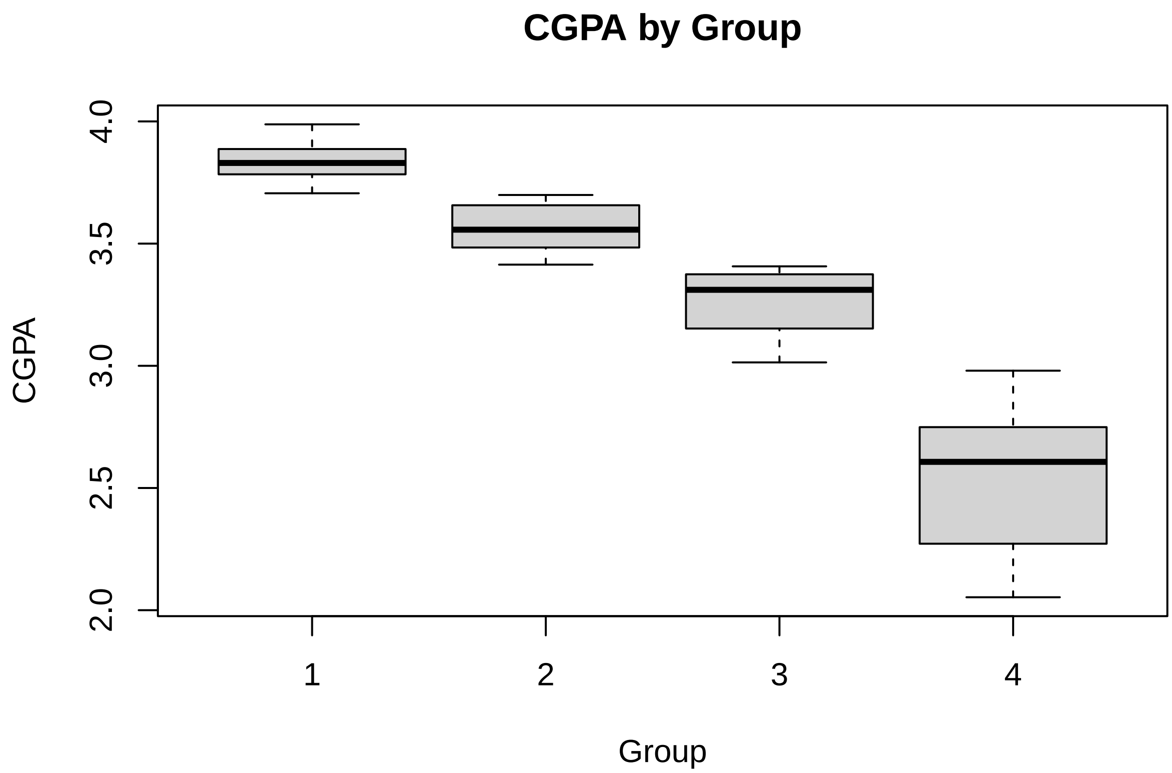


Easier way:

```
boxplot(CGPA ~ Sex, data = df, main = "CGPA by Sex", xlab = "Sex", ylab = "CGPA")
```




```r
boxplot(CGPA ~ Group, data = df, main = "CGPA by Group", xlab = "Group", ylab = "CGPA")
```



Inferential Statistics

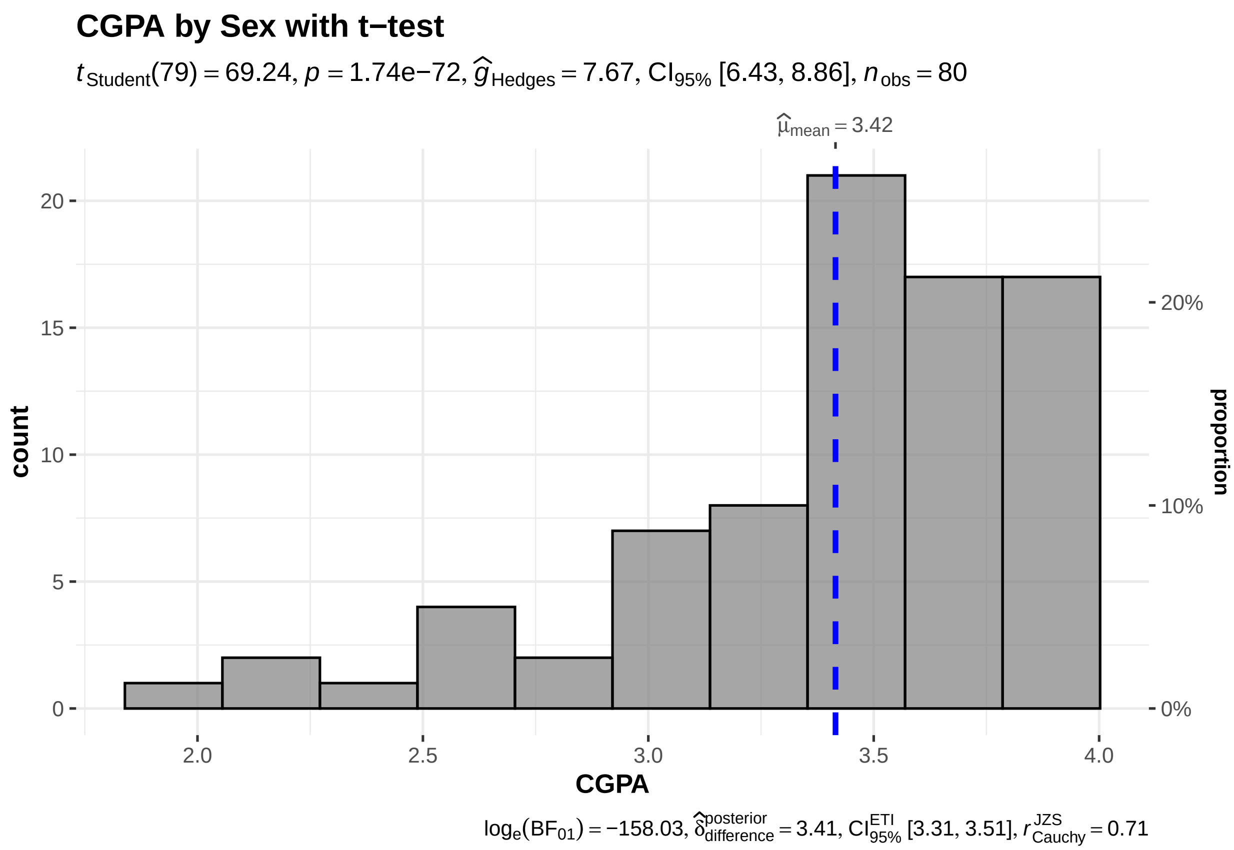
Mean Comparisons : 1) Independent samples t-test: CGPA by Sex 2) One-way ANOVA: CGPA by Group 3) One-way ANOVA: CGPA by AssignedTeacher

```
library(ggstatsplot)
```

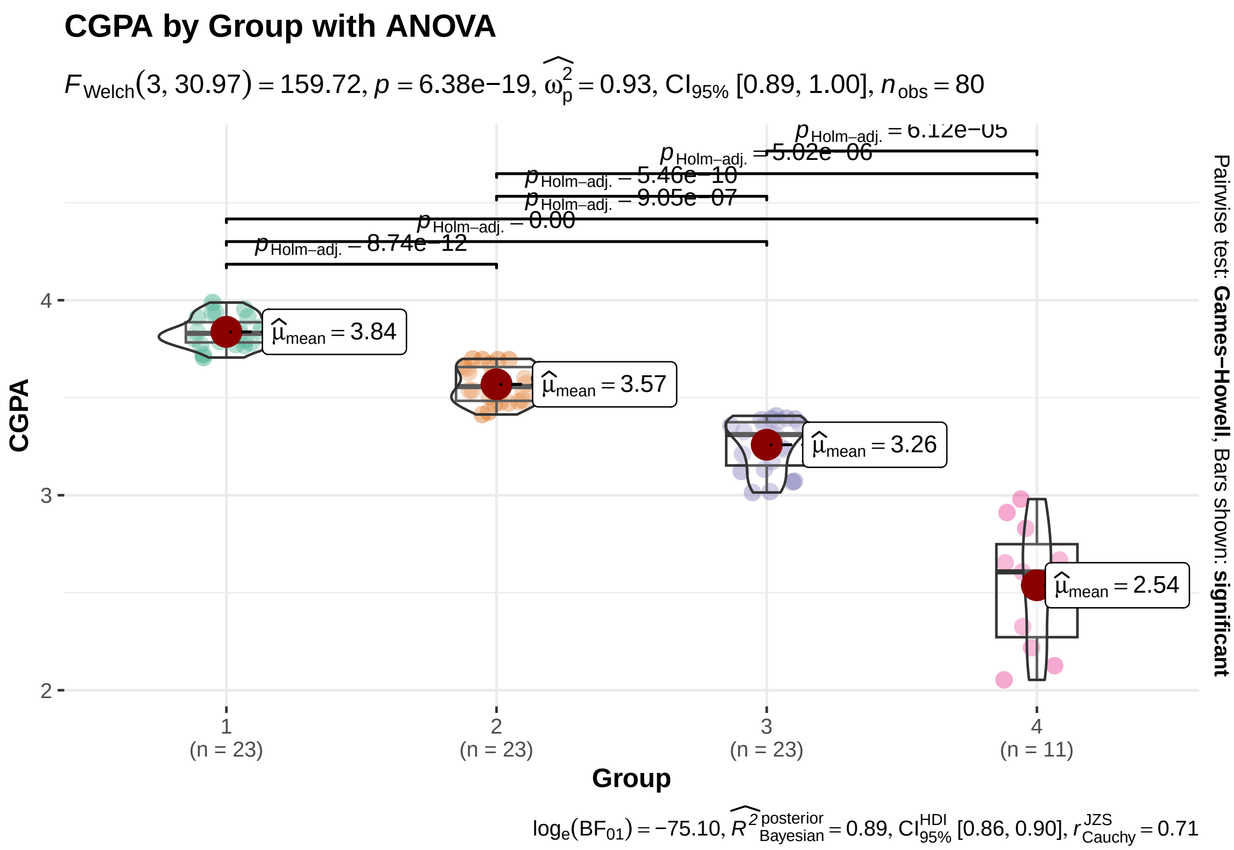
```
## Warning: package 'ggstatsplot' was built under R version 4.5.2
```

```
## You can cite this package as:
##      Patil, I. (2021). Visualizations with statistical details: The 'ggstatsplot' approach.
##      Journal of Open Source Software, 6(61), 3167, doi:10.21105/joss.03167
```

```
# t-test with plot
gghistostats(
  data = df,
  x = CGPA,
  y = Sex,
  type = "parametric",  # t-test
  title = "CGPA by Sex with t-test"
)
```




```r
# ANOVA with plot
ggbetweenstats(
  data = df,
  x = Group,
  y = CGPA,
  type = "parametric",  # ANOVA
  title = "CGPA by Group with ANOVA",
  pairwise.comparisons = TRUE,  # Shows post-hoc
  pairwise.display = "significant"
)
```



Easier Way:

```r
# (i) t-test
t.test(CGPA ~ Sex, data = df, var.equal = TRUE)   # Equal variances
```

```
##
##  Two Sample t-test
##
## data:  CGPA by Sex
## t = 1.6962, df = 78, p-value = 0.09384
## alternative hypothesis: true difference in means between group F and group M is not equal to 0
## 95 percent confidence interval:
##  -0.02967273  0.37124607
## sample estimates:
## mean in group F mean in group M
##        3.521967        3.351180
```

```r
t.test(CGPA ~ Sex, data = df, var.equal = FALSE) # Welch's t-test
```

```
##
##  Welch Two Sample t-test
##
## data:  CGPA by Sex
## t = 1.7854, df = 70.874, p-value = 0.07848
## alternative hypothesis: true difference in means between group F and group M is not equal to 0
```

```
## 95 percent confidence interval:
##  -0.01995516  0.36152849
## sample estimates:
## mean in group F mean in group M
##        3.521967        3.351180
```

```r
# (ii) ANOVA
aov_result <- aov(CGPA ~ Group, data = df)
summary(aov_result)  # ANOVA table
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Group        3 13.670   4.557   203.1 <2e-16 ***
## Residuals   76  1.705   0.022
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# (iii) Another ANOVA
aov_teacher <- aov(CGPA ~ AssignedTeacher, data = df)
summary(aov_teacher)
```

```
##                 Df Sum Sq Mean Sq F value Pr(>F)
## AssignedTeacher 22   1.78 0.08091   0.339  0.997
## Residuals       57  13.60 0.23850
```

```r
# Post-hoc tests
TukeyHSD(aov_result)  # Tukey's HSD
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = CGPA ~ Group, data = df)
##
## $Group
##           diff        lwr        upr p adj
## 2-1 -0.2691304 -0.3851445 -0.1531164 3e-07
## 3-1 -0.5790000 -0.6950141 -0.4629859 0e+00
## 4-1 -1.2994664 -1.4436908 -1.1552420 0e+00
## 3-2 -0.3098696 -0.4258836 -0.1938555 0e+00
## 4-2 -1.0303360 -1.1745604 -0.8861116 0e+00
## 4-3 -0.7204664 -0.8646908 -0.5762420 0e+00
```

```r
pairwise.t.test(df$CGPA, df$Group, p.adjust.method = "bonferroni")
```

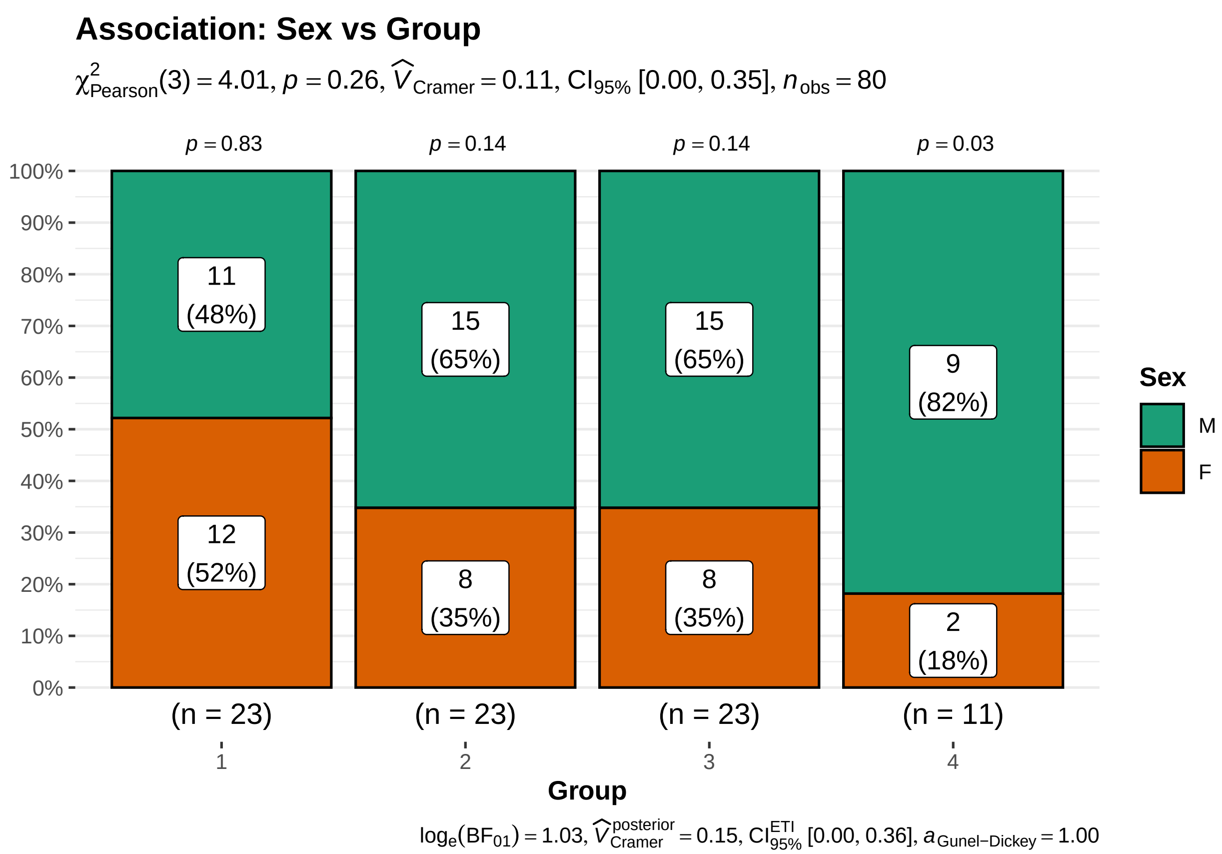
```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  df$CGPA and df$Group
##
##   1       2       3
## 2 2.5e-07 -       -
## 3 < 2e-16 4.9e-09 -
```

```
## 4 < 2e-16 < 2e-16 < 2e-16
##
## P value adjustment method: bonferroni
```

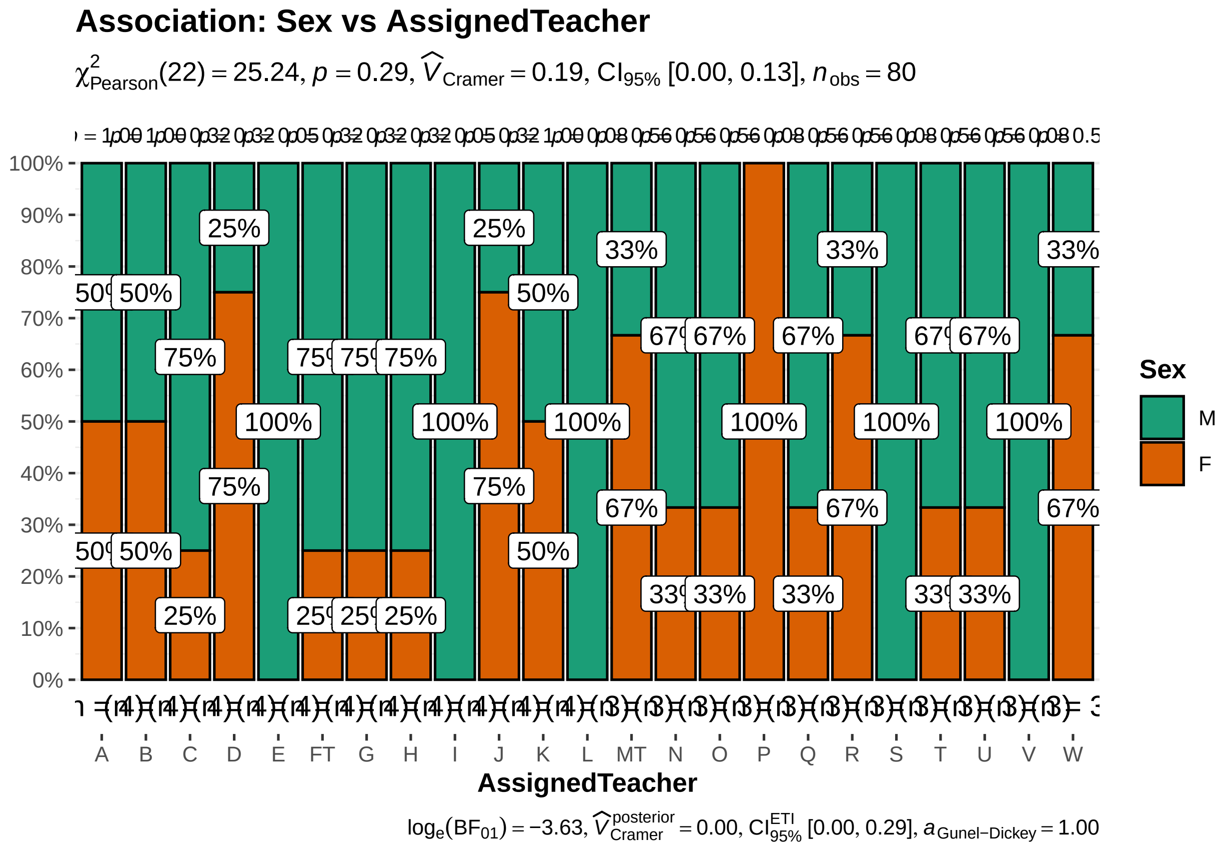
Association Tests : (i) Chi-square test: Sex vs Group (ii) Chi-square test: Sex vs AssignedTeacher

```r
library(ggstatsplot)

# Sex vs Group with chi-square
ggbarstats(
  data = df,
  x = Sex,
  y = Group,
  title = "Association: Sex vs Group",
  results.subtitle = TRUE,
  label = "both",   # Show counts and percentages
  perc.k = 1        # Decimal places for percentages
)
```




```r
# Sex vs AssignedTeacher
ggbarstats(
  data = df,
  x = Sex,
  y = AssignedTeacher,
  title = "Association: Sex vs AssignedTeacher"
)
```



Easier Way:

```r
# Sex vs Group with chi-square
chisq.test(table(df$Sex, df$Group))
```

```
## Warning in chisq.test(table(df$Sex, df$Group)): Chi-squared approximation may
## be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(df$Sex, df$Group)
## X-squared = 4.0095, df = 3, p-value = 0.2604
```

```r
# Sex vs AssignedTeacher
chisq.test(table(df$Sex, df$AssignedTeacher))
```

```
## Warning in chisq.test(table(df$Sex, df$AssignedTeacher)): Chi-squared
## approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(df$Sex, df$AssignedTeacher)
## X-squared = 25.244, df = 22, p-value = 0.2855
```

Regression Analysis

Simple Linear Regression : (i) Model 1: CGPA ~ Sex (ii) Model 2: CGPA ~ Group

```r
summary(lm(CGPA ~ Sex, data = df))
```

```
##
## Call:
## lm(formula = CGPA ~ Sex, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29818 -0.22513  0.09743  0.28478  0.60182
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.5220     0.0796  44.244   <2e-16 ***
## SexM         -0.1708     0.1007  -1.696   0.0938 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.436 on 78 degrees of freedom
## Multiple R-squared:  0.03557,    Adjusted R-squared:  0.02321
## F-statistic: 2.877 on 1 and 78 DF,  p-value: 0.09384
```

```r
summary(lm(CGPA ~ Group, data = df))
```

```
##
## Call:
## lm(formula = CGPA ~ Group, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.48527 -0.07246  0.00333  0.10026  0.44173
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.83774    0.03123 122.887  < 2e-16 ***
## Group2      -0.26913    0.04417  -6.094 4.23e-08 ***
## Group3      -0.57900    0.04417 -13.110  < 2e-16 ***
## Group4      -1.29947    0.05491 -23.668  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1498 on 76 degrees of freedom
## Multiple R-squared:  0.8891, Adjusted R-squared:  0.8847
## F-statistic: 203.1 on 3 and 76 DF,  p-value: < 2.2e-16
```

Multiple Regression : Full model: CGPA ~ Sex + Group + AssignedTeacher

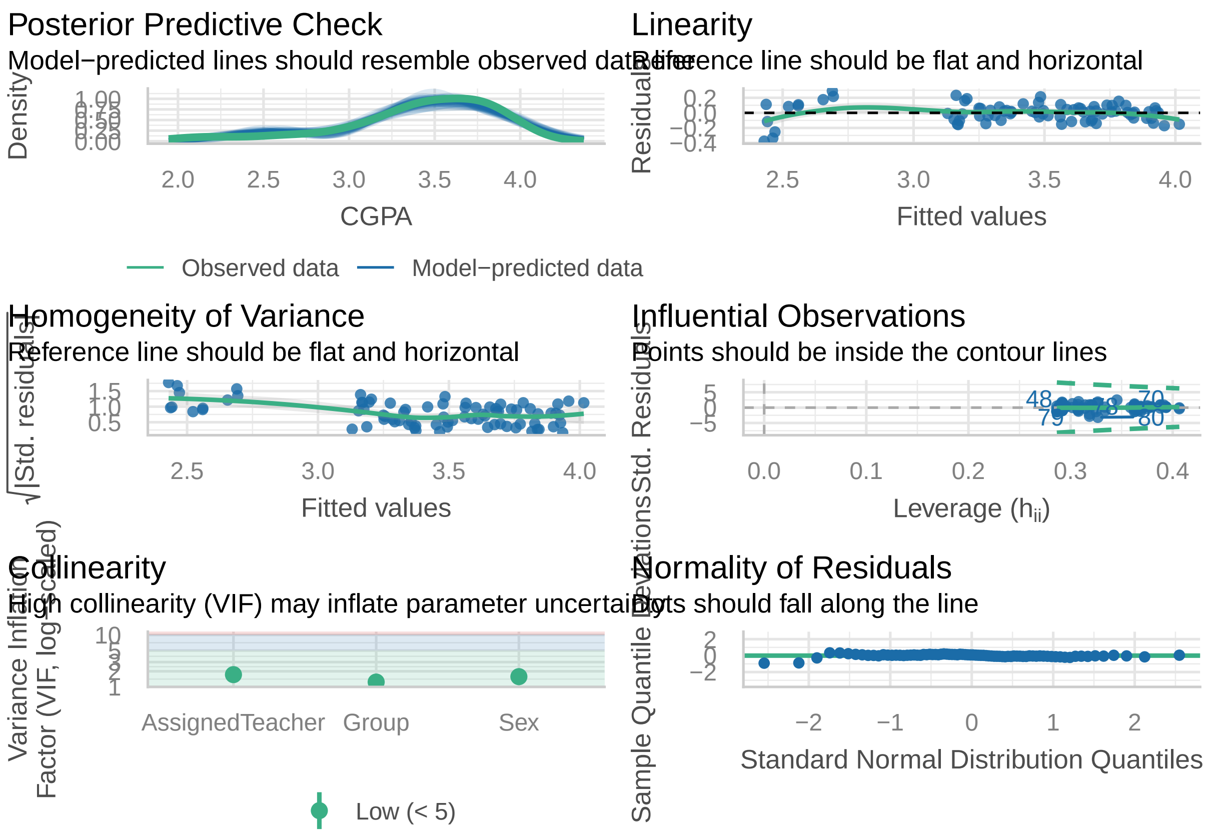
```r
# Full model
full_model <- lm(CGPA ~ Sex + Group + AssignedTeacher, data = df)
summary(full_model)
```

```
##
## Call:
## lm(formula = CGPA ~ Sex + Group + AssignedTeacher, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37629 -0.07189  0.01049  0.06907  0.29031
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3.697292   0.079024  46.787  < 2e-16 ***
## SexM                0.053618   0.042178   1.271   0.2092
## Group2             -0.278455   0.043500  -6.401 4.14e-08 ***
## Group3             -0.588325   0.043500 -13.525  < 2e-16 ***
## Group4             -1.321622   0.058344 -22.652  < 2e-16 ***
## AssignedTeacherB    0.225250   0.102817   2.191   0.0329 *
## AssignedTeacherC    0.183846   0.103356   1.779   0.0810 .
## AssignedTeacherD    0.066154   0.103356   0.640   0.5249
## AssignedTeacherE    0.093191   0.104957   0.888   0.3786
## AssignedTeacherFT   0.033096   0.103356   0.320   0.7501
## AssignedTeacherG    0.041346   0.103356   0.400   0.6907
## AssignedTeacherH    0.130846   0.103356   1.266   0.2111
## AssignedTeacherI    0.007691   0.104957   0.073   0.9419
## AssignedTeacherJ    0.260404   0.103356   2.519   0.0148 *
## AssignedTeacherK    0.264250   0.102817   2.570   0.0130 *
## AssignedTeacherL    0.005017   0.114247   0.044   0.9651
## AssignedTeacherMT   0.202096   0.111863   1.807   0.0765 .
## AssignedTeacherN   -0.007777   0.112184  -0.069   0.9450
## AssignedTeacherO    0.175557   0.112184   1.565   0.1236
## AssignedTeacherP    0.143302   0.113299   1.265   0.2115
## AssignedTeacherQ    0.021557   0.112184   0.192   0.8484
## AssignedTeacherR    0.193096   0.111863   1.726   0.0901 .
## AssignedTeacherS    0.087351   0.114247   0.765   0.4479
## AssignedTeacherT    0.113557   0.112184   1.012   0.3160
## AssignedTeacherU    0.023890   0.112184   0.213   0.8322
## AssignedTeacherV    0.165351   0.114247   1.447   0.1537
## AssignedTeacherW    0.211429   0.111863   1.890   0.0642 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1454 on 53 degrees of freedom
## Multiple R-squared:  0.9271, Adjusted R-squared:  0.8914
## F-statistic: 25.93 on 26 and 53 DF,  p-value: < 2.2e-16
```

```r
# All diagnostic plots
library(performance)
```

```
## Warning: package 'performance' was built under R version 4.5.2
```

```r
check_model(full_model)
```



Outlier Detection

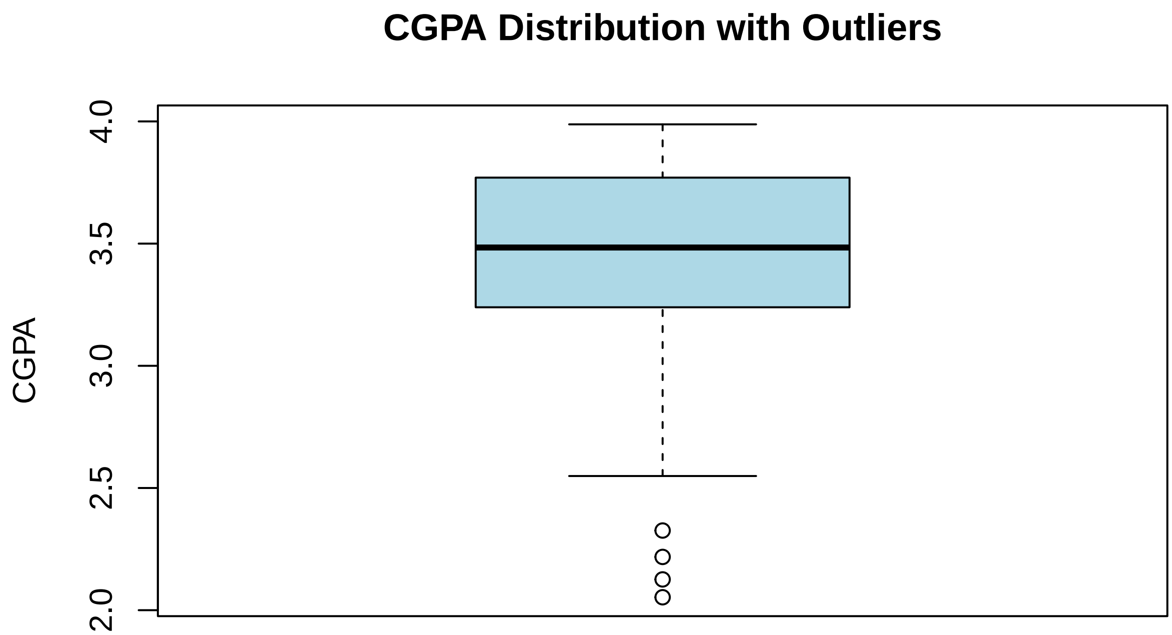
(i) Detect outliers in CGPA using the IQR rule
(ii) Visualize outliers using boxplots

```r
# Detect outliers using IQR rule
Q1 <- quantile(df$CGPA, 0.25)
Q3 <- quantile(df$CGPA, 0.75)
IQR_val <- Q3 - Q1
lower <- Q1 - 1.5 * IQR_val
upper <- Q3 + 1.5 * IQR_val

outliers <- df[df$CGPA < lower | df$CGPA > upper, ]
cat("\nOutliers detected:", nrow(outliers), "\n")
```

```
##
## Outliers detected: 4
```

```r
# Boxplot in one line
boxplot(df$CGPA, col = "lightblue", main = "CGPA Distribution with Outliers",
        ylab = "CGPA", outline = TRUE)
```



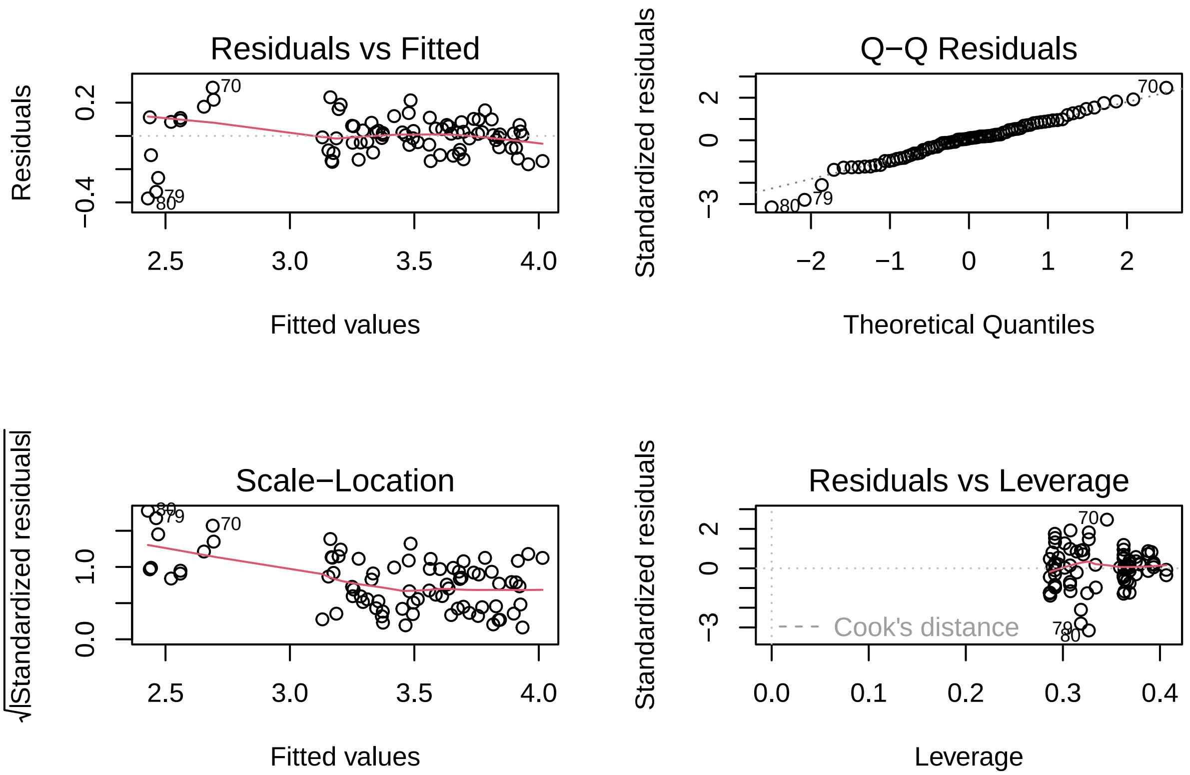
Regression Diagnostics

For the multiple regression model (CGPA ~ Sex + Group + AssignedTeacher), check:

i) Linearity
ii) Normality
iii) Homoscedasticity
iv) Multicollinearity

```
# Fit the model
full_model <- lm(CGPA ~ Sex + Group + AssignedTeacher, data = df)

# 1. Diagnostic plots (Linearity, Homoscedasticity, Normality of residuals)
par(mfrow = c(2, 2))   # 2x2 layout
plot(full_model)        # Generates:
```




```r
                                # 1. Residuals vs Fitted → checks linearity
                                # 2. Normal Q-Q → checks residual normality
                                # 3. Scale-Location → checks homoscedasticity
                                # 4. Residuals vs Leverage → checks influential points
par(mfrow = c(1, 1))   # Reset layout

# 2. Normality test of residuals
cat("\nShapiro-Wilk Test on Residuals:\n")
```

```
##
## Shapiro-Wilk Test on Residuals:
```

```r
shapiro.test(residuals(full_model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(full_model)
## W = 0.97797, p-value = 0.1824
```

```r
# 3. Multicollinearity (VIF)
library(car)
```

```
## Warning: package 'car' was built under R version 4.5.2
```

```
## Loading required package: carData

## Warning: package 'carData' was built under R version 4.5.2

##
## Attaching package: 'car'

## The following object is masked from 'package:psych':
##
##     logit

## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
cat("\nVariance Inflation Factors (VIF):\n")
```

```
##
## Variance Inflation Factors (VIF):
```

```r
vif(full_model)
```

```
##                     GVIF Df GVIF^(1/(2*Df))
## Sex             1.577656  1        1.256048
## Group           1.241791  3        1.036752
## AssignedTeacher 1.723375 22        1.012447
```