Real-world Data Analysis Using R: Subset of Framingham Data

Mina Peyton

2024-09-20

```
# if needed, install required packages for this script
# install.packages(c("tidyverse", "readr", "psych", "car", "moments", "ggplot2",
# "dunn.test", "rstatix")
```

```
# attach libraries that are need to run the script
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

```
library(readr)
library(psych)
```

```
##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:psych':
##
##     logit
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```
library(moments)
library(ggplot2)
library(dunn.test)
library(rstatix)
```

```
##
## Attaching package: 'rstatix'
##
## The following object is masked from 'package:stats':
##
##     filter
```

Read in framingham.csv data file and store it as a data frame object name “df”.
The “df” object is now listed in the “Environment” df with 4240 obs of 16 variables.

You can click on “df” to open in a new tab to view the data.

```
df = read_csv("framingham.csv")
```

```
## Rows: 4240 Columns: 16
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (16): male, age, education, currentSmoker, cigsPerDay, BPMeds, prevalent...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
str(df)
```

```
## spc_tbl_ [4,240 × 16] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ male           : num [1:4240] 1 0 1 0 0 0 0 0 1 1 ...
##  $ age            : num [1:4240] 39 46 48 61 46 43 63 45 52 43 ...
##  $ education      : num [1:4240] 4 2 1 3 3 2 1 2 1 1 ...
##  $ currentSmoker  : num [1:4240] 0 0 1 1 1 0 0 1 0 1 ...
##  $ cigsPerDay     : num [1:4240] 0 0 20 30 23 0 0 20 0 30 ...
##  $ BPMeds         : num [1:4240] 0 0 0 0 0 0 0 0 0 0 ...
##  $ prevalentStroke: num [1:4240] 0 0 0 0 0 0 0 0 0 0 ...
##  $ prevalentHyp   : num [1:4240] 0 0 0 1 0 1 0 0 1 1 ...
##  $ diabetes       : num [1:4240] 0 0 0 0 0 0 0 0 0 0 ...
##  $ totChol        : num [1:4240] 195 250 245 225 285 228 205 313 260 225 ...
##  $ sysBP          : num [1:4240] 106 121 128 150 130 ...
##  $ diaBP          : num [1:4240] 70 81 80 95 84 110 71 71 89 107 ...
##  $ BMI            : num [1:4240] 27 28.7 25.3 28.6 23.1 ...
##  $ heartRate      : num [1:4240] 80 95 75 65 85 77 60 79 76 93 ...
##  $ glucose        : num [1:4240] 77 76 70 103 85 99 85 78 79 88 ...
##  $ TenYearCHD     : num [1:4240] 0 0 0 1 0 0 1 0 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   male = col_double(),
##   ..   age = col_double(),
##   ..   education = col_double(),
##   ..   currentSmoker = col_double(),
##   ..   cigsPerDay = col_double(),
##   ..   BPMeds = col_double(),
##   ..   prevalentStroke = col_double(),
##   ..   prevalentHyp = col_double(),
##   ..   diabetes = col_double(),
##   ..   totChol = col_double(),
##   ..   sysBP = col_double(),
##   ..   diaBP = col_double(),
##   ..   BMI = col_double(),
##   ..   heartRate = col_double(),
##   ..   glucose = col_double(),
##   ..   TenYearCHD = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Objective 1:

Determine if there is a difference in average BMI for individuals with high cholesterol (cholesterol ≥ 240 mm/L) compared to individuals without high cholesterol (i.e., with normal/borderline cholesterol levels).

Total cholesterol: Less than 200 mg/dL is normal, 200–239 mg/dL is borderline high, and 240 mg/dL or higher is high
(https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/lipid-panel
(https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/lipid-panel))

What is your research question?
Is there a difference in average BMI in individuals with high cholesterol compared to those that do not have high cholesterol?

What is the study design?
What is the population of interest?

Create two new variables from the total cholesterol level info above.
Chol_group to define the three levels above (normal, borderline, high)
Chol_bin to define two levels as binary (0 - do not have high cholesterol, 1 - have high cholesterol)

```
df <- df %>%
  mutate(Chol_group = case_when(
    totChol <= 200 ~ "normal",
    totChol > 200 & totChol <= 239 ~ "borderline",
    totChol >= 240 ~ "high")) %>%
  mutate(Chol_bin = ifelse(Chol_group == "high", 1, 0))

str(df)
```

```
## tibble [4,240 × 18] (S3: tbl_df/tbl/data.frame)
##  $ male           : num [1:4240] 1 0 1 0 0 0 0 0 1 1 ...
##  $ age            : num [1:4240] 39 46 48 61 46 43 63 45 52 43 ...
##  $ education      : num [1:4240] 4 2 1 3 3 2 1 2 1 1 ...
##  $ currentSmoker  : num [1:4240] 0 0 1 1 1 0 0 1 0 1 ...
##  $ cigsPerDay     : num [1:4240] 0 0 20 30 23 0 0 20 0 30 ...
##  $ BPMeds         : num [1:4240] 0 0 0 0 0 0 0 0 0 0 ...
##  $ prevalentStroke: num [1:4240] 0 0 0 0 0 0 0 0 0 0 ...
##  $ prevalentHyp   : num [1:4240] 0 0 0 1 0 1 0 0 1 1 ...
##  $ diabetes       : num [1:4240] 0 0 0 0 0 0 0 0 0 0 ...
##  $ totChol        : num [1:4240] 195 250 245 225 285 228 205 313 260 225 ...
##  $ sysBP          : num [1:4240] 106 121 128 150 130 ...
##  $ diaBP          : num [1:4240] 70 81 80 95 84 110 71 71 89 107 ...
##  $ BMI            : num [1:4240] 27 28.7 25.3 28.6 23.1 ...
##  $ heartRate      : num [1:4240] 80 95 75 65 85 77 60 79 76 93 ...
##  $ glucose        : num [1:4240] 77 76 70 103 85 99 85 78 79 88 ...
##  $ TenYearCHD     : num [1:4240] 0 0 0 1 0 0 1 0 0 0 ...
##  $ Chol_group     : chr [1:4240] "normal" "high" "high" "borderline" ...
##  $ Chol_bin       : num [1:4240] 0 1 1 0 1 0 0 1 1 0 ...
```

Set the new variables as factors

```
df <- df %>%
  mutate(Chol_group = as.factor(case_when(
    totChol <= 200 ~ "normal",
    totChol > 200 & totChol <= 239 ~ "borderline",
    totChol >= 240 ~ "high"))) %>%
  mutate(Chol_bin = as.factor(ifelse(Chol_group == "high", 1, 0)))

str(df)
```

```
## tibble [4,240 × 18] (S3: tbl_df/tbl/data.frame)
##  $ male           : num [1:4240] 1 0 1 0 0 0 0 0 1 1 ...
##  $ age            : num [1:4240] 39 46 48 61 46 43 63 45 52 43 ...
##  $ education      : num [1:4240] 4 2 1 3 3 2 1 2 1 1 ...
##  $ currentSmoker  : num [1:4240] 0 0 1 1 1 0 0 1 0 1 ...
##  $ cigsPerDay     : num [1:4240] 0 0 20 30 23 0 0 20 0 30 ...
##  $ BPMeds         : num [1:4240] 0 0 0 0 0 0 0 0 0 0 ...
##  $ prevalentStroke: num [1:4240] 0 0 0 0 0 0 0 0 0 0 ...
##  $ prevalentHyp   : num [1:4240] 0 0 0 1 0 1 0 0 1 1 ...
##  $ diabetes       : num [1:4240] 0 0 0 0 0 0 0 0 0 0 ...
##  $ totChol        : num [1:4240] 195 250 245 225 285 228 205 313 260 225 ...
##  $ sysBP          : num [1:4240] 106 121 128 150 130 ...
##  $ diaBP          : num [1:4240] 70 81 80 95 84 110 71 71 89 107 ...
##  $ BMI            : num [1:4240] 27 28.7 25.3 28.6 23.1 ...
##  $ heartRate      : num [1:4240] 80 95 75 65 85 77 60 79 76 93 ...
##  $ glucose        : num [1:4240] 77 76 70 103 85 99 85 78 79 88 ...
##  $ TenYearCHD     : num [1:4240] 0 0 0 1 0 0 1 0 0 0 ...
##  $ Chol_group     : Factor w/ 3 levels "borderline","high",..: 3 2 2 1 2 1 1 2 2 1
...
##  $ Chol_bin       : Factor w/ 2 levels "0","1": 1 2 2 1 2 1 1 2 2 1 ...
```

Re-order the levels of the Chol_group

```
df$Chol_group = factor(df$Chol_group, levels = c("normal", "borderline", "high"))
str(df)
```

```
## tibble [4,240 × 18] (S3: tbl_df/tbl/data.frame)
##  $ male           : num [1:4240] 1 0 1 0 0 0 0 0 1 1 ...
##  $ age            : num [1:4240] 39 46 48 61 46 43 63 45 52 43 ...
##  $ education      : num [1:4240] 4 2 1 3 3 2 1 2 1 1 ...
##  $ currentSmoker  : num [1:4240] 0 0 1 1 1 0 0 1 0 1 ...
##  $ cigsPerDay     : num [1:4240] 0 0 20 30 23 0 0 20 0 30 ...
##  $ BPMeds         : num [1:4240] 0 0 0 0 0 0 0 0 0 0 ...
##  $ prevalentStroke: num [1:4240] 0 0 0 0 0 0 0 0 0 0 ...
##  $ prevalentHyp   : num [1:4240] 0 0 0 1 0 1 0 0 1 1 ...
##  $ diabetes       : num [1:4240] 0 0 0 0 0 0 0 0 0 0 ...
##  $ totChol        : num [1:4240] 195 250 245 225 285 228 205 313 260 225 ...
##  $ sysBP          : num [1:4240] 106 121 128 150 130 ...
##  $ diaBP          : num [1:4240] 70 81 80 95 84 110 71 71 89 107 ...
##  $ BMI            : num [1:4240] 27 28.7 25.3 28.6 23.1 ...
##  $ heartRate      : num [1:4240] 80 95 75 65 85 77 60 79 76 93 ...
##  $ glucose        : num [1:4240] 77 76 70 103 85 99 85 78 79 88 ...
##  $ TenYearCHD     : num [1:4240] 0 0 0 1 0 0 1 0 0 0 ...
##  $ Chol_group     : Factor w/ 3 levels "normal","borderline",..: 1 3 3 2 3 2 2 3 3 2
...
##  $ Chol_bin       : Factor w/ 2 levels "0","1": 1 2 2 1 2 1 1 2 2 1 ...
```

Exploratory data analysis: Explore the variables of interest

```
table(df$Chol_group)
```

```
##
##     normal borderline       high
##        887       1422       1881
```

```
table(df$Chol_bin)
```

```
##
##    0    1
## 2309 1881
```

Find and remove NA values

```
sum(is.na(df$BMI)) # 19
```

```
## [1] 19
```

```
sum(is.na(df$totChol)) # 50
```

```
## [1] 50
```

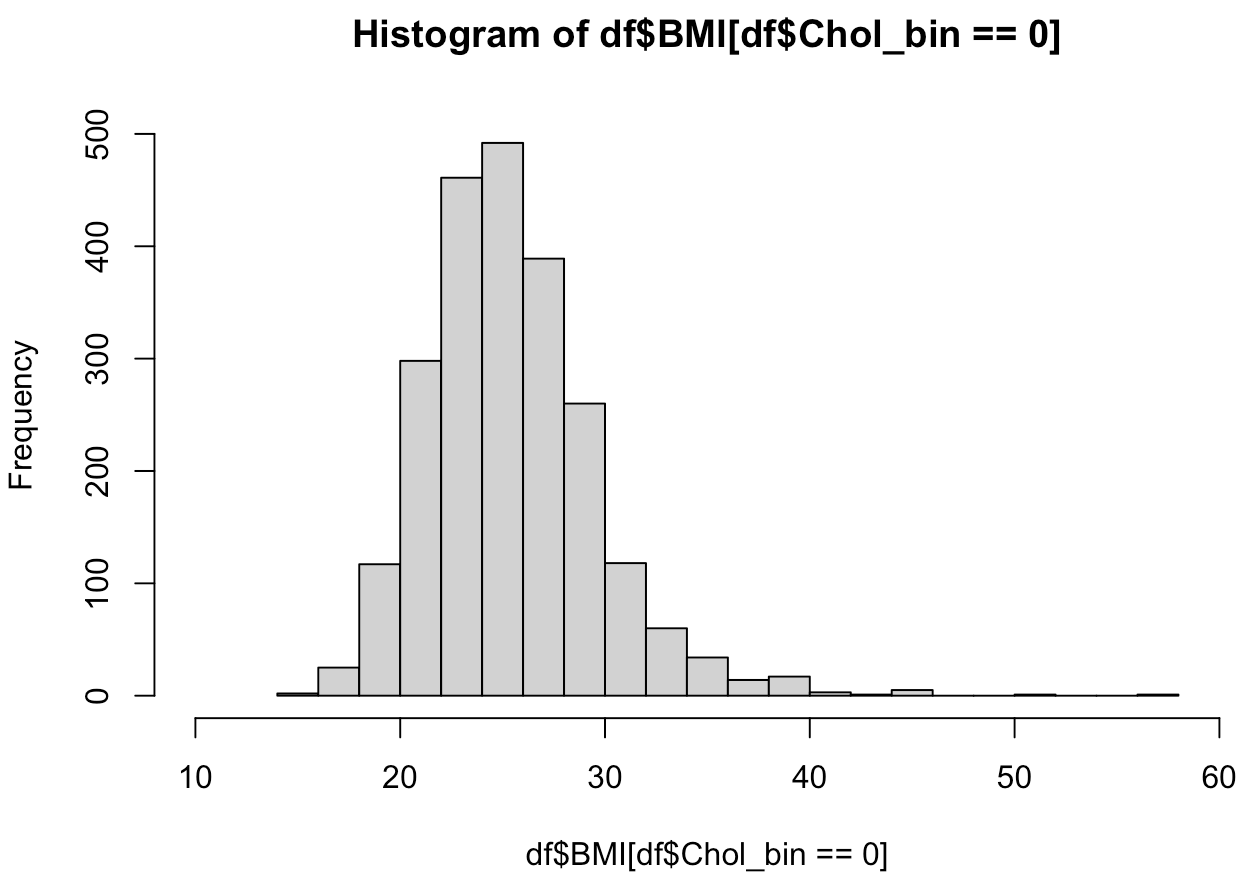
```
remove = c(which(is.na(df$BMI)),which(is.na(df$totChol)))
```

```
df = df[-remove,] # [rows, columns]
dim(df)
```

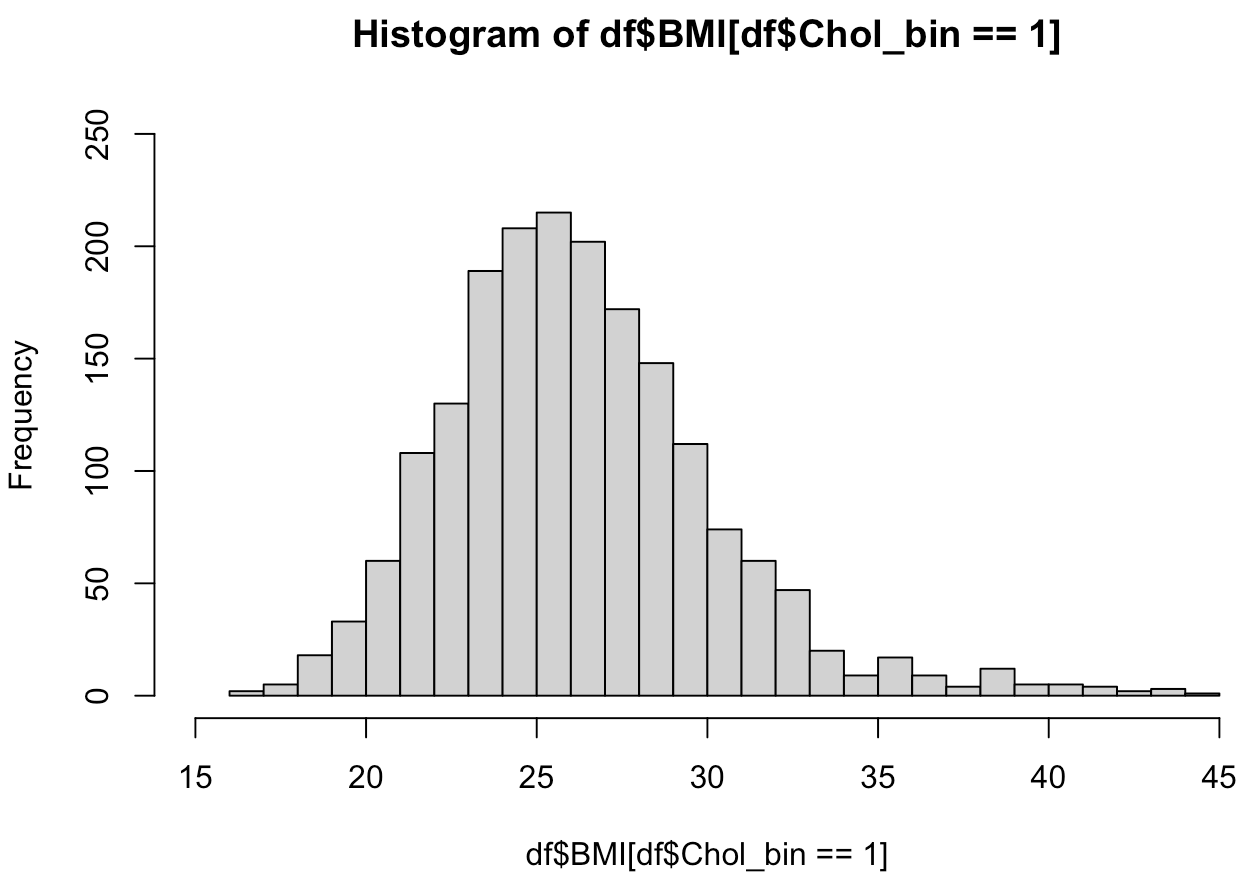
```
## [1] 4172   18
```

Create graphical summaries that visualizes BMI for individuals with high cholesterol compared to individuals without high cholesterol

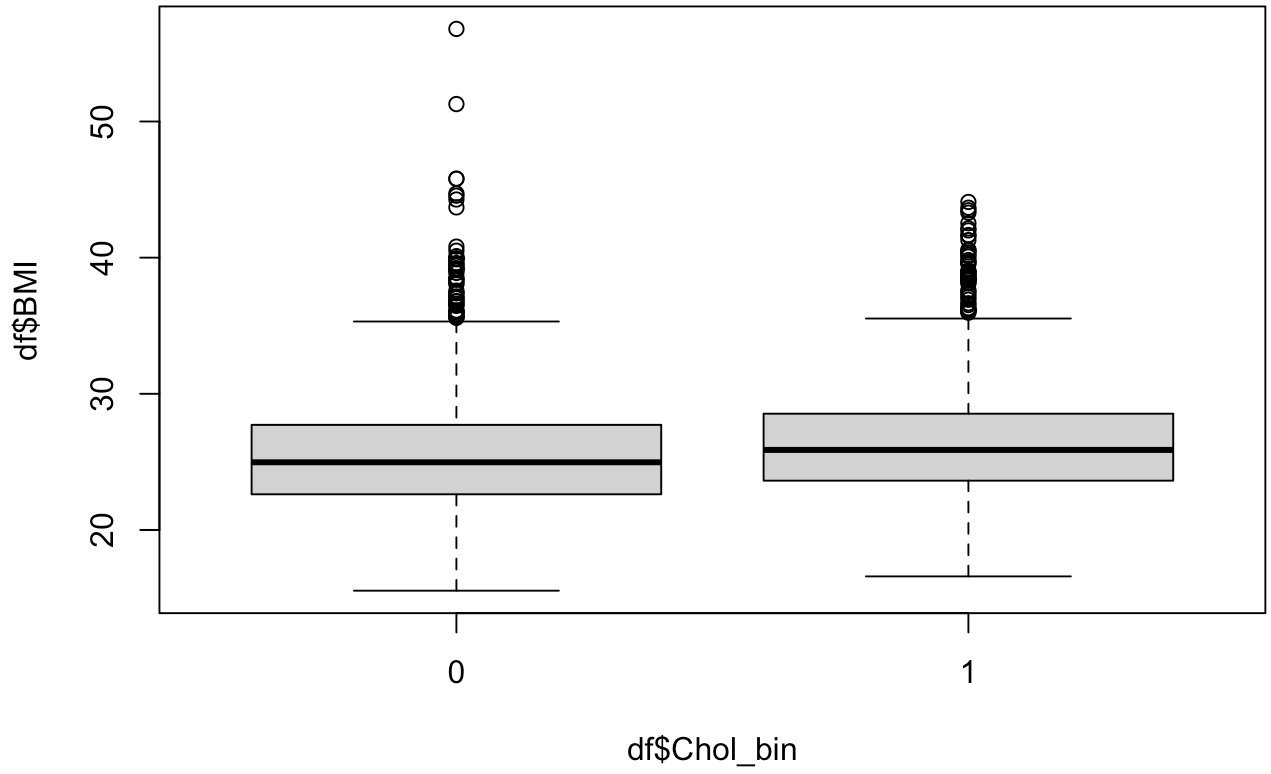
```
hist(df$BMI[df$Chol_bin == 0], breaks = 20, xlim = c(10, 60), ylim = c(0,500))
```




```
hist(df$BMI[df$Chol_bin ==1], breaks = 20, xlim = c(15, 45), ylim = c(0,250))
```




```
boxplot(df$BMI ~ df$Chol_bin, data = df)
```



Calculate summary statistics for BMI for both group (with and without high cholesterol)

```
#library(psych)
describeBy(df$BMI, df$Chol_bin)
```

```
##
##  Descriptive statistics by group
## group: 0
##    vars    n  mean   sd median trimmed  mad   min  max range skew kurtosis   se
## X1    1 2298 25.41 4.13  24.97   25.11 3.72 15.54 56.8 41.26 1.11     3.54 0.09
## ------------------------------------------------------------
## group: 1
##    vars    n  mean   sd median trimmed mad   min   max range skew kurtosis   se
## X1    1 1874 26.27 3.96  25.88      26 3.6 16.59 44.09  27.5  0.9     1.77 0.09
```

```
sum_stats = df %>% group_by(Chol_bin) %>%
  summarise(
    n = n(),
    mean = mean(BMI),
    sd = sd(BMI),
    se = sd/sqrt(n),
    median = median(BMI))
```

Determine statistical test that would be most appropriate for answering the research question: one-sample t-test, paired t-test, two-sample t-test (assuming unequal variances), or two-sample t-test (assuming equal variances).

Check assumptions for your test.

```
# Check normality
shapiro.test(df$BMI[df$Chol_bin==0])
```

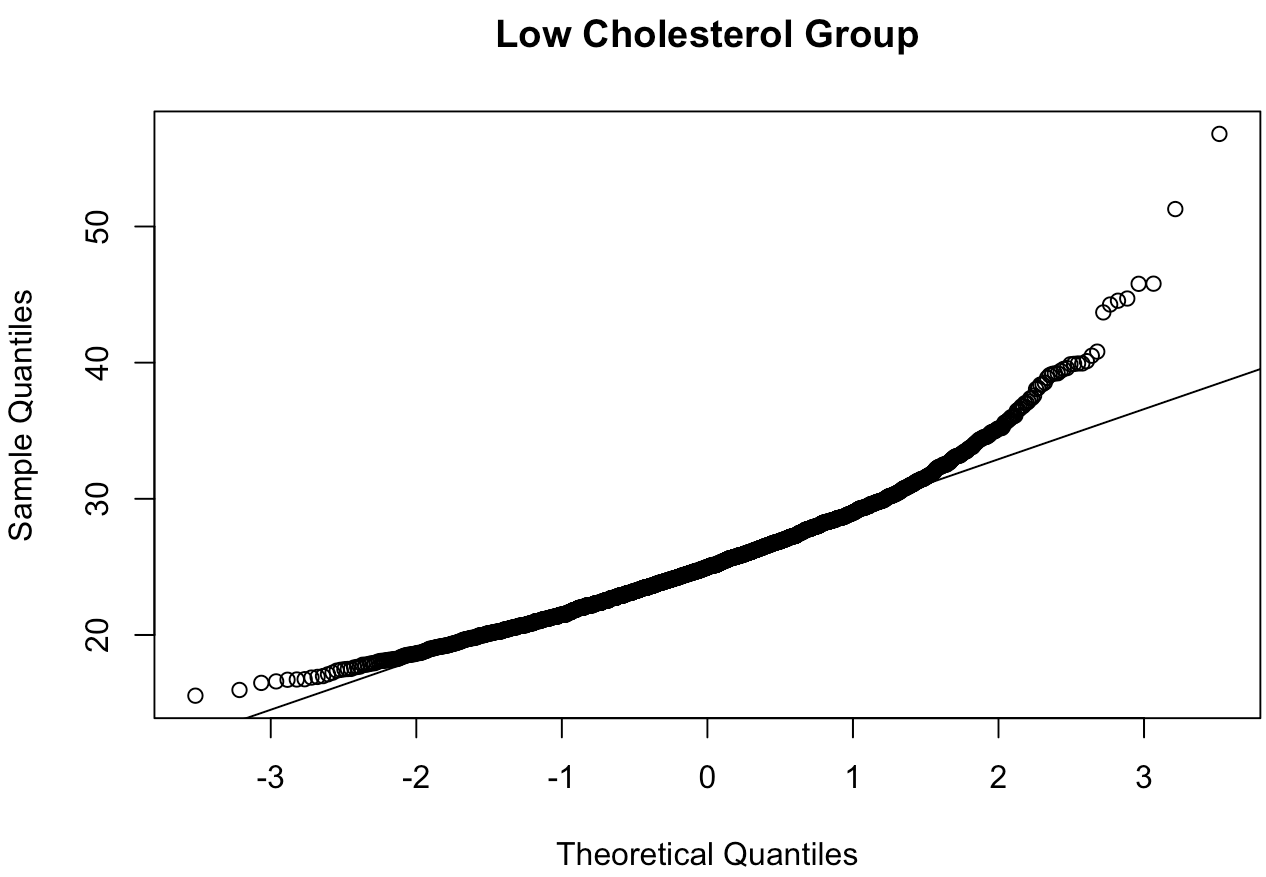
```
##
##  Shapiro-Wilk normality test
##
## data:  df$BMI[df$Chol_bin == 0]
## W = 0.94947, p-value < 2.2e-16
```

```
shapiro.test(df$BMI[df$Chol_bin==1])
```

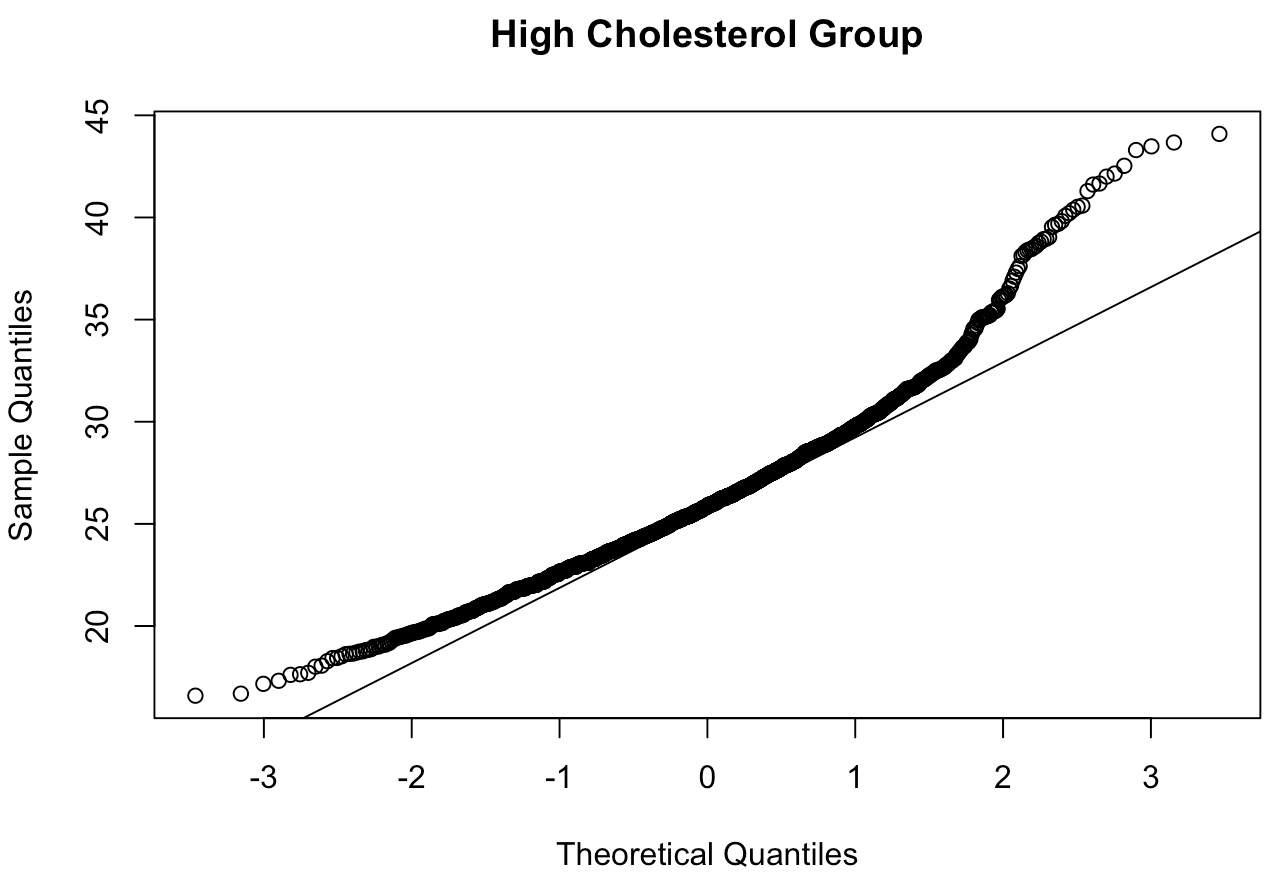
```
##
##  Shapiro-Wilk normality test
##
## data:  df$BMI[df$Chol_bin == 1]
## W = 0.96025, p-value < 2.2e-16
```

```
# Sensitive to large sample sizes
```

```
qqnorm(df$BMI[df$Chol_bin==0], main = "Low Cholesterol Group")
qqline(df$BMI)
```




```
qqnorm(df$BMI[df$Chol_bin==1], main = "High Cholesterol Group")
qqline(df$BMI)
```




```
# check equal variance

# Levene's Test
leveneTest(BMI ~ Chol_bin, data = df)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    1  1.5417 0.2144
##       4170
```

```
# null hypothesis = variances of the groups are equal
# alternative hypothesis = variances of the groups are unequal
# p = 0.21, fail to reject the null hypothesis, variance of the groups are equal
```

Statistical Inference: Carry out the hypothesis test.

- What are the hypotheses?

null hypothesis = no difference in mean BMI between both groups
alternative hypothesis = there is a difference in mean BMI between both groups

- Your p-value?
- Make a conclusion using α = 0.05

```
# Two-sample t-test with equal variance
t.test(df$BMI~df$Chol_bin, alternative="two.sided", var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  df$BMI by df$Chol_bin
## t = -6.8379, df = 4170, p-value = 9.201e-12
## alternative hypothesis: true difference in means between group 0 and group 1 is not e
qual to 0
## 95 percent confidence interval:
##  -1.1105608 -0.6156364
## sample estimates:
## mean in group 0 mean in group 1
##        25.40683        26.26993
```

```
# 9.201e-12
```

```
# Wilcoxon Rank-Sum Test
wilcox.test(df$BMI[df$Chol_bin==0], df$BMI[df$Chol_bin==1])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  df$BMI[df$Chol_bin == 0] and df$BMI[df$Chol_bin == 1]
## W = 1859111, p-value = 2.957e-14
## alternative hypothesis: true location shift is not equal to 0
```

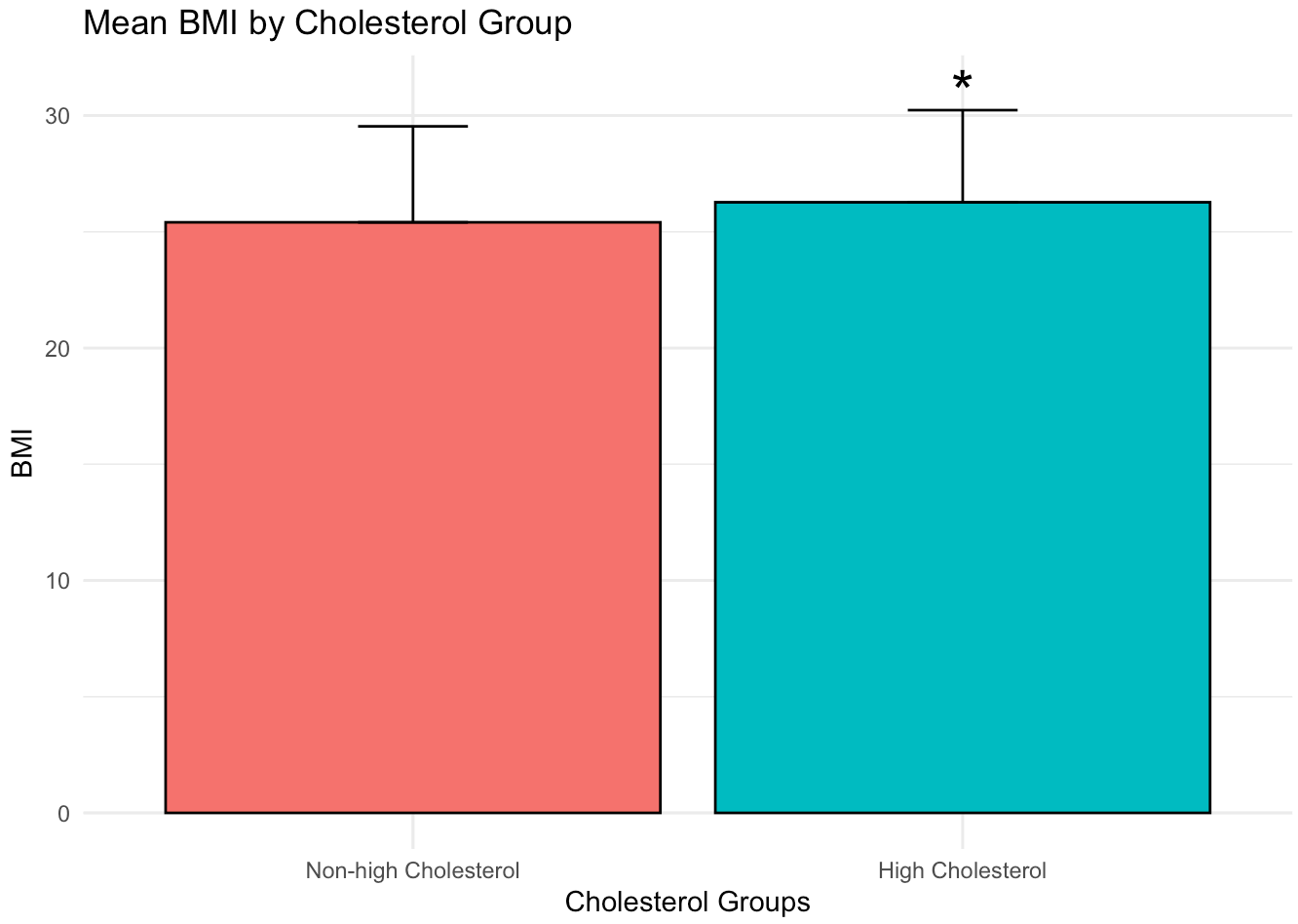
```
# p-value = 2.957e-14
```

Conclusion: Provide an answer to your research question.
There is a significant difference (p-value = 9.201e-12) in mean BMI between individuals with high cholesterol compared to those without high cholesterol. The high cholesterol group had higher BMI (26.3 +- 3.96) compared to non-high cholesterol group (25.4 +- 4.13).

Data visualization
(https://ggplot2-book.org/ (https://ggplot2-book.org/))

*statistically different from non-high cholesterol

```
p <- ggplot(sum_stats, aes(x = Chol_bin, y = mean, fill = Chol_bin)) +
  geom_bar(stat = "identity", color = "black") +
  geom_errorbar(aes(ymin = mean, ymax = mean + sd), width = 0.2) +
  scale_x_discrete(labels = c("0" = "Non-high Cholesterol", "1" = "High Cholesterol")) +
  theme_minimal() +
  labs(title = "Mean BMI by Cholesterol Group",
       x = "Cholesterol Groups",
       y = "BMI") +
  theme(legend.position = "none")  # Hide the legend

# Add annotations
p + annotate("text", x = 2, y = (sum_stats$mean[1] + sum_stats$sd[1]) + 1.5, label =
"*", size = 8, color = "black")
```



Pratice Exercise

Determine if there is a difference in totChol between males and females
males = 1
females = 0

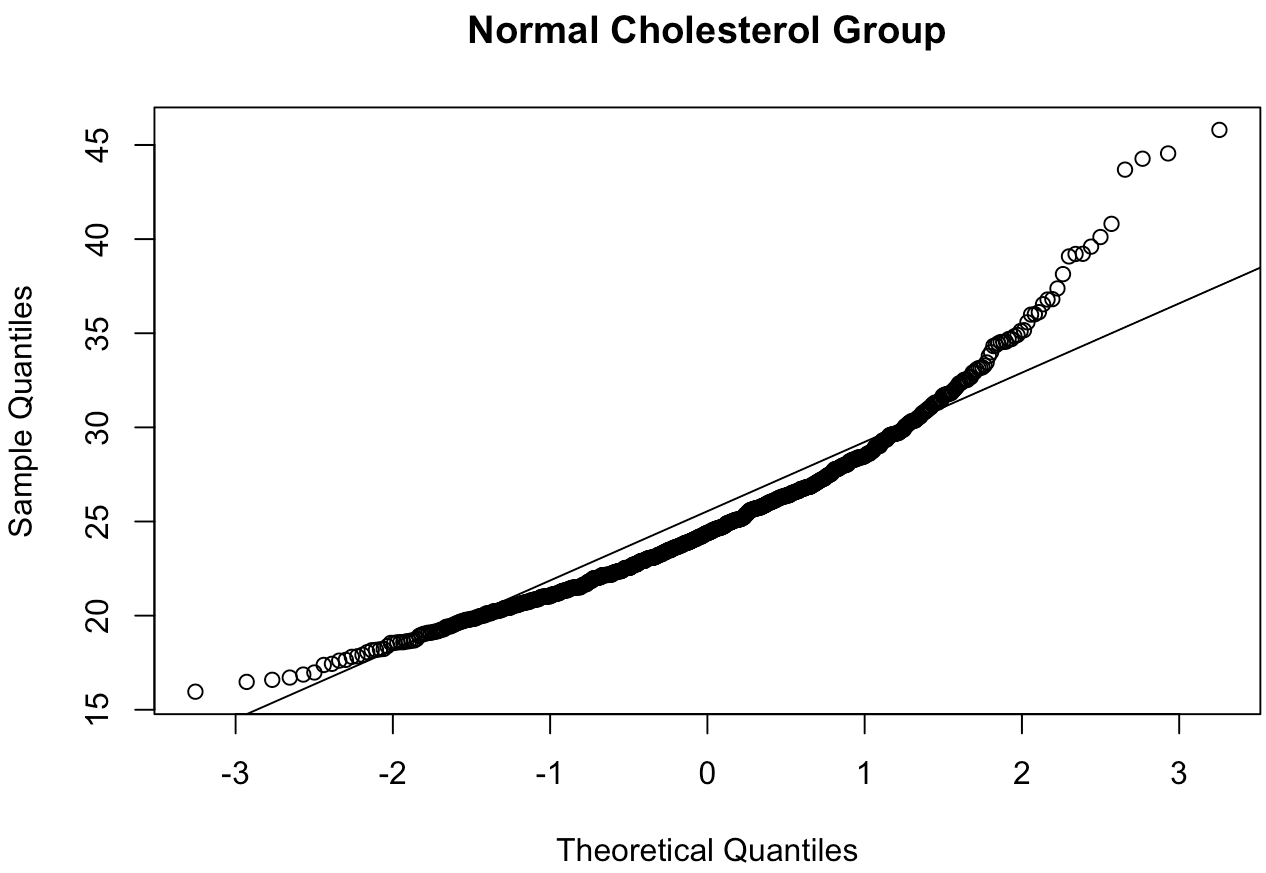
Objective 2:

Determine if there is a difference in BMI between the Chol_group (i.e, normal, borderline, and high cholesterol groups)?

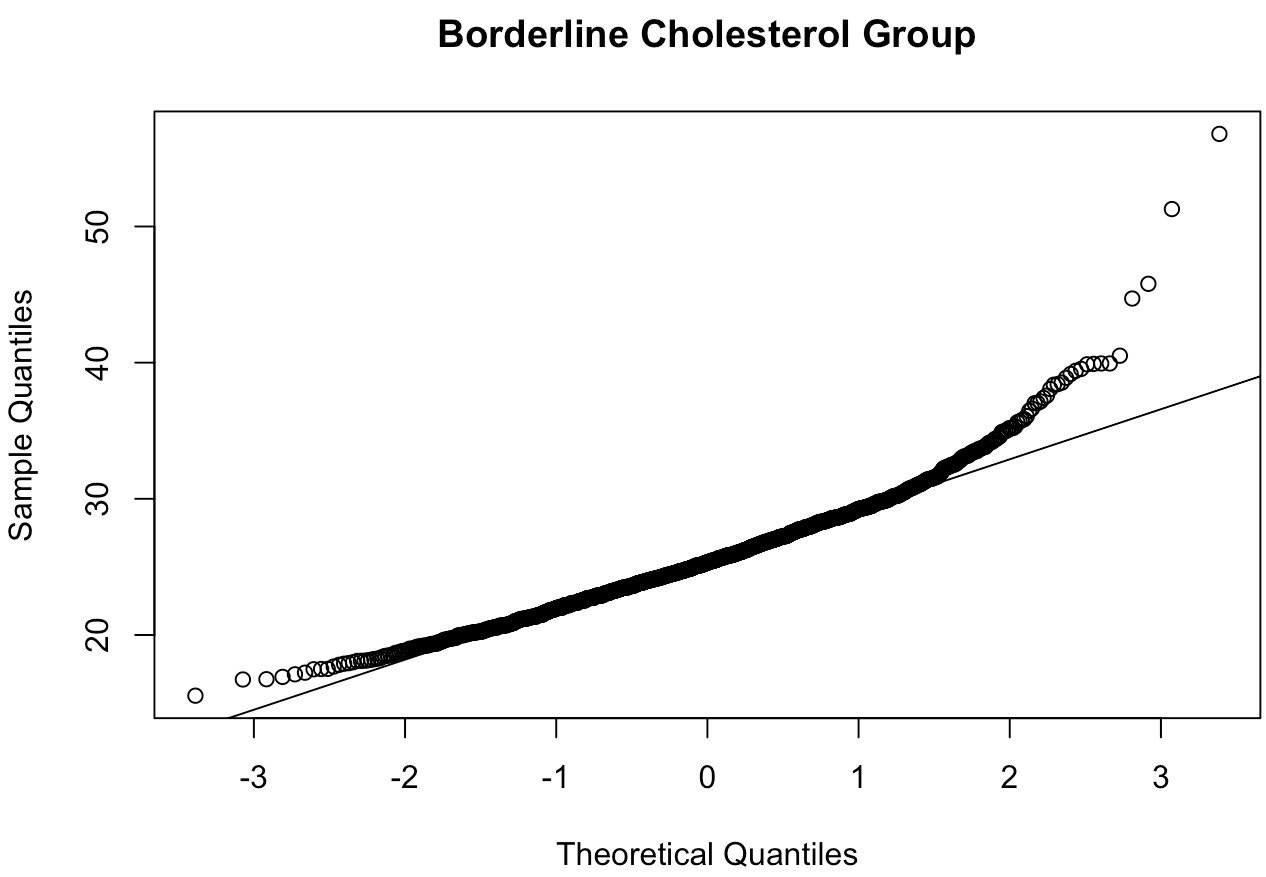
```
sum_stats = df %>% group_by(Chol_group) %>%
  summarise(
    n = n(),
    mean = mean(BMI),
    sd = sd(BMI),
    se = sd/sqrt(n),
    median = median(BMI))

describeBy(df$BMI, df$Chol_group)
```

```
##
##  Descriptive statistics by group
## group: normal
##    vars   n  mean   sd median trimmed  mad   min  max range skew kurtosis   se
## X1    1 884 24.95 4.18  24.38   24.58 3.58 15.96 45.8 29.84 1.13     2.45 0.14
## ------------------------------------------------------------
## group: borderline
##    vars    n  mean   sd median trimmed  mad   min  max range skew kurtosis   se
## X1    1 1414 25.69 4.07  25.31   25.45 3.62 15.54 56.8 41.26 1.14      4.4 0.11
## ------------------------------------------------------------
## group: high
##    vars    n  mean   sd median trimmed mad   min   max range skew kurtosis   se
## X1    1 1874 26.27 3.96  25.88      26 3.6 16.59 44.09  27.5  0.9     1.77 0.09
```

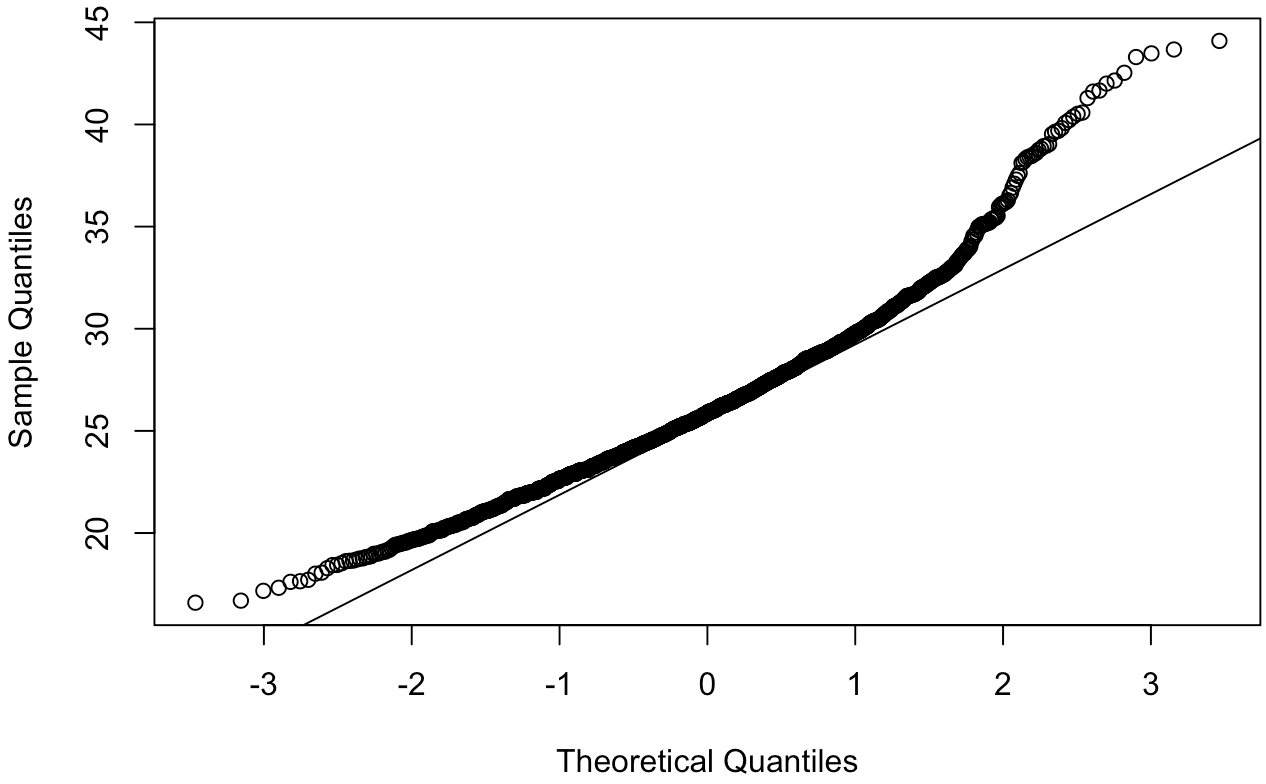
```
qqnorm(df$BMI[df$Chol_group== "normal"], main = "Normal Cholesterol Group")
qqline(df$BMI)
```




```
qqnorm(df$BMI[df$Chol_group== "borderline"], main = "Borderline Cholesterol Group")
qqline(df$BMI)
```




```
qqnorm(df$BMI[df$Chol_group== "high"], main = "High Cholesterol Group")
qqline(df$BMI)
```




```
# Equal variance: Brown-Forsythe Test
leveneTest(BMI ~ Chol_group, data = df)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    2  0.8247 0.4385
##       4169
```

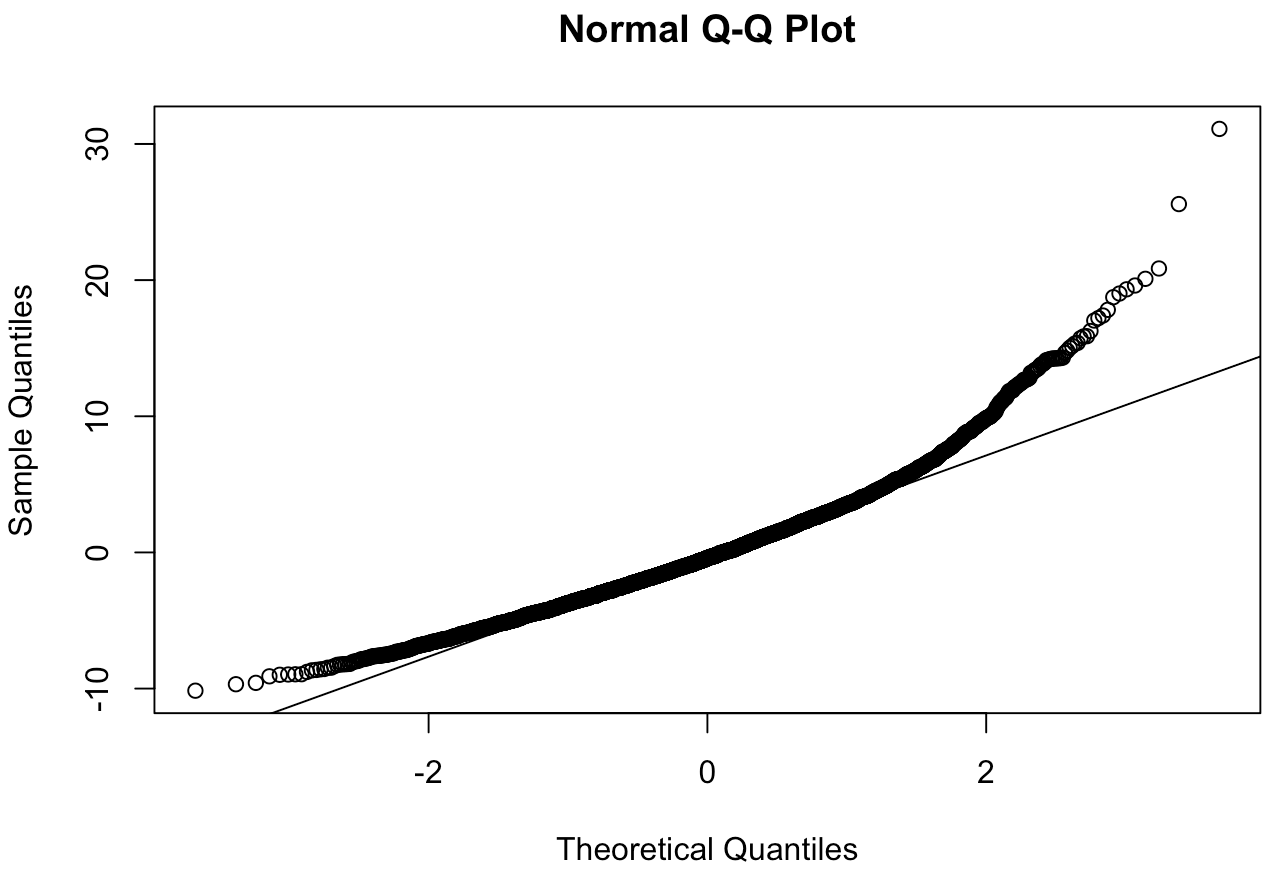
```
# pvalue =0.4385
```

Statistical inference: One-way ANOVA (Analysis of Variance)

```
# Perform one-way ANOVA
anova_model <- aov(BMI ~ Chol_group, data = df)

# View the ANOVA table
summary(anova_model)
```

```
##               Df Sum Sq Mean Sq F value   Pr(>F)
## Chol_group     2   1074   537.0   32.79 7.42e-15 ***
## Residuals   4169  68272    16.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Check normality of residuals
qqnorm(residuals(anova_model))
qqline(residuals(anova_model))
```



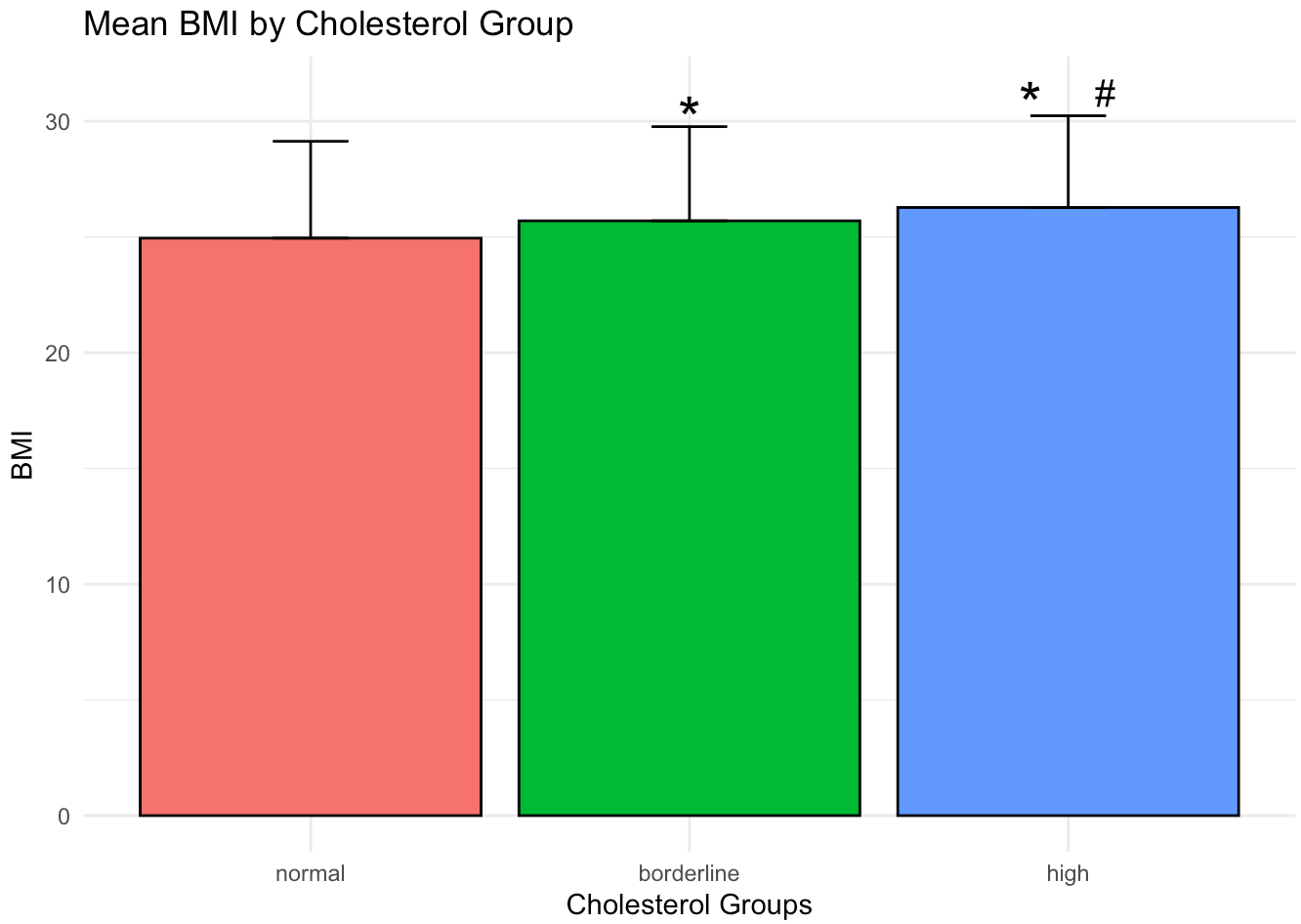

```
# Perform Tukey's Honest Significant Differences test
# post-hoc test used after performing an ANOVA to find out which specific group means
# are significantly different from each other.
# It compares all possible pairs of means and adjusts for multiple comparisons,
# to control the family-wise error rate
posthoc <- TukeyHSD(anova_model)
posthoc
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = BMI ~ Chol_group, data = df)
##
## $Chol_group
##                         diff       lwr       upr    p adj
## borderline-normal 0.7488122 0.3420053 1.1556191 4.84e-05
## high-normal       1.3238560 0.9367317 1.7109802 0.00e+00
## high-borderline   0.5750438 0.2408329 0.9092547 1.65e-04
```

There is a significant difference (p = 7.42e-15) in mean BMI between cholesterol groups. Tukey's HSD post-hoc analysis revealed significant differences between borderline (p = 4.84e-05) and high (p = 0.00e+00) cholesterol groups compared to normal, and a significant difference in mean BMI between high compared to borderline cholesterol groups (p = 1.65e-04). Overall mean BMI for normal, borderline, and high cholesterol groups were 24.9 +- 4.18, 25.7 +- 4.07, and 26.3 +- 3.96, respectively.

Data visualization
*significantly different from normal
significantly different from borderline

```
# Create the boxplot
p <- ggplot(sum_stats, aes(x = Chol_group, y = mean, fill = Chol_group)) +
  geom_bar(stat = "identity", color = "black") +
  geom_errorbar(aes(ymin = mean, ymax = mean + sd), width = 0.2) +
  theme_minimal() +
  labs(title = "Mean BMI by Cholesterol Group",
       x = "Cholesterol Groups",
       y = "BMI") +
  theme(legend.position = "none")  # Hide the legend

# Add annotations
# Assuming these positions are appropriate for your plot
p +
  annotate("text", x = 2, y = (sum_stats$mean[1] + sum_stats$sd[1]) + 1, label = "*", si
ze = 8, color = "black") +  # Normal vs. Borderline
  annotate("text", x = 2.9, y = (sum_stats$mean[2] + sum_stats$sd[2]) + 1, label = "*",
size = 8, color = "black") +  # Normal vs. High
  annotate("text", x = 3.1, y = (sum_stats$mean[3] + sum_stats$sd[3]) + 1, label = "#",
size = 5, color = "black")  # Borderline vs. High
```



If the normality assumption is violated: Kruskal-Wallis test

```
kruskal.test(BMI ~ Chol_group, data = df)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  BMI by Chol_group
## Kruskal-Wallis chi-squared = 85.724, df = 2, p-value < 2.2e-16
```

Non-parametric post-hoc comparisons after a Kruskal-Wallis test: Dunn's test or the pairwise Wilcoxon rank-sum test

Dunn's Test

```
dunn.test(df$BMI, df$Chol_group, method = "bonferroni") # or "hs" for Holm-Sidak adjustment
```

```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 85.7236, df = 2, p-value = 0
##
##
##                            Comparison of x by group
##                                  (Bonferroni)
## Col Mean-|
## Row Mean |   borderli       high
## ---------+----------------------
##     high |  -4.239563
##          |    0.0000*
##          |
##   normal |   5.287793   9.216729
##          |    0.0000*    0.0000*
##
## alpha = 0.05
## Reject Ho if p <= alpha/2
```

Pairwise Wilcoxon rank-sum test

```
pairwise.wilcox.test(df$BMI, df$Chol_group, p.adjust.method = "bonferroni")
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  df$BMI and df$Chol_group
##
##            normal  borderline
## borderline 3.3e-07 -
## high       < 2e-16 6.3e-05
##
## P value adjustment method: bonferroni
```

If equal variance assumption is violated: Welch's ANOVA

```
oneway.test(BMI ~ Chol_group, data = df, var.equal = FALSE)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  BMI and Chol_group
## F = 32.014, num df = 2.0, denom df = 2234.4, p-value = 1.959e-14
```

Performing Pairwise Comparisons with Games-Howell Test

```
games_howell_test(BMI ~ Chol_group, data = df)
```

```
## # A tibble: 3 × 8
##   .y.   group1     group2     estimate conf.low conf.high     p.adj p.adj.signif
## * <chr> <chr>      <chr>         <dbl>    <dbl>     <dbl>     <dbl> <chr>
## 1 BMI   normal     borderline    0.749    0.332     1.17  0.0000768 ****
## 2 BMI   normal     high          1.32     0.930     1.72  0         ****
## 3 BMI   borderline high          0.575    0.243     0.907 0.000151  ***
```