Comparing Groups on Continuous Outcomes

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(emmeans)

## Warning: package 'emmeans' was built under R version 4.4.2

## Welcome to emmeans.  
## Caution: You lose important information if you filter this package's results.  
## See '? untidy'

# Add the code to import hsb2 here  
  
  
### Backup Import  
hsb2 <- within(read.csv("https://stats.idre.ucla.edu/wp-content/uploads/2016/02/hsb2-2.csv"), {  
 race <- as.factor(race)  
 schtyp <- as.factor(schtyp)  
 prog <- as.factor(prog)  
})  
  
# These variables are categorical factors  
# This is an important step  
attach(hsb2)  
hsb2$race <- as.factor(race)  
hsb2$schtyp <- as.factor(schtyp)  
hsb2$prog <- as.factor(prog)

## Your Turn 1

* Assume 22 degrees of freedom.
* Use the pt() function to calculate the value for . What is your conclusion?
* Use the pt() function to calculate the value for . What is your conclusion?
* Use the qt() function to determine the value of that corresponds to and compare it to the values of produced by the data.

pt(-7.5, 22)

## [1] 8.471024e-08

pt(0.5, 22)

## [1] 0.6889802

qt(0.975, df = 22)

## [1] 2.073873

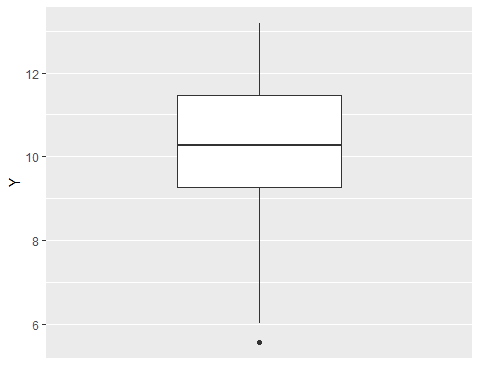
## Your Turn 2

* Uncomment the code below to generate a random sample of 50 observations and calculate their mean.
* Complete the code to create a boxplot of the data using ggplot().

set.seed(1)  
Y <- rnorm(50, 10, 2)  
df <- data.frame(Y)  
  
df %>%   
 summarise(mean=mean(Y))

## mean  
## 1 10.2009

df %>% ggplot() +  
 geom\_boxplot(aes(y=Y)) +   
 labs(x="") +  
 scale\_x\_discrete(labels=c(""))



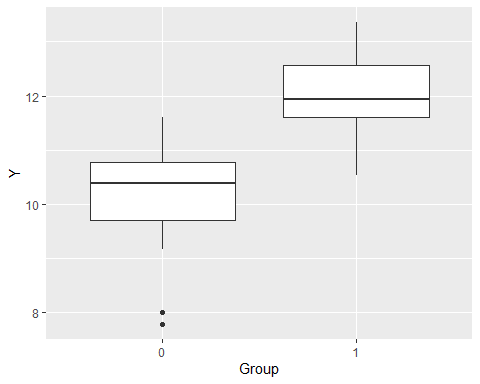
## Your Turn 3

* Uncomment the code below to generate a random sample of 16 observations per group and calculate their means by group.
* Complete the code to create a boxplot of the data using ggplot().

set.seed(1)  
X <- c(rep(0,25),rep(1,25))  
Y <- 2\*X + rnorm(50, 10, 1)  
df <- data.frame(cbind(X,Y))  
  
df %>%   
 group\_by(X) %>%   
 summarise(mean=mean(Y))

## # A tibble: 2 × 2  
## X mean  
## <dbl> <dbl>  
## 1 0 10.2  
## 2 1 12.0

df %>% ggplot() +  
 geom\_boxplot(aes(x=factor(X),y=Y)) +  
 labs(x="Group")



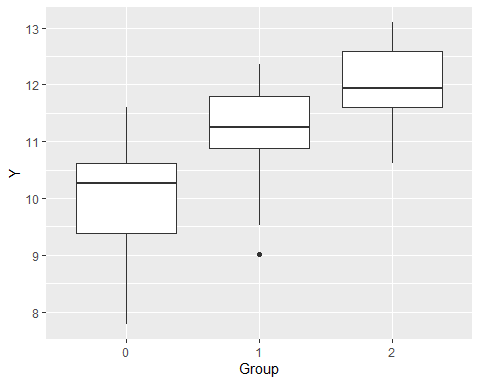
## Your Turn 4

* Set your seed to 1.
* Generate 48 random numbers: 16 from , 16 from , and 16 from .
* Calculate the group means using summarise().
* Create boxplots of the data by group using ggplot().

set.seed(1)  
X <- c(rep(0,16),rep(1,16),rep(2,16))  
Y <- X + rnorm(48, 10, 1)  
df <- data.frame(cbind(X,Y))  
  
df %>%   
 group\_by(X) %>%   
 summarise(mean=mean(Y))

## # A tibble: 3 × 2  
## X mean  
## <dbl> <dbl>  
## 1 0 10.1  
## 2 1 11.1  
## 3 2 12.0

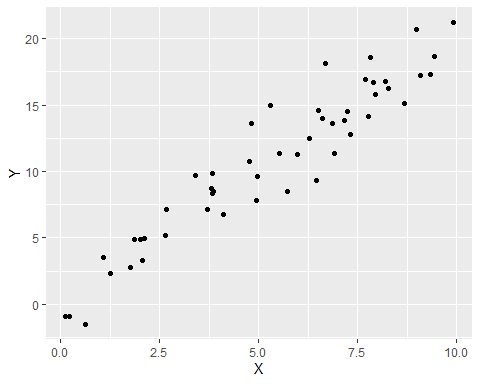
df %>% ggplot() +  
 geom\_boxplot(aes(x=factor(X),y=Y)) +  
 labs(x="Group")



## Your Turn 5

* Set your seed to 1.
* Generate 50 random numbers from a uniform distribution with a minimum of 0 and a maximum of 10 and save it as X.
* Generate 50 random numbers from a normal distribution with standard deviation 2 and a mean equal to and save it as Y.
* Combine X and Y into a dataframe named df.
* Create a scatterplot of the data using ggplot().

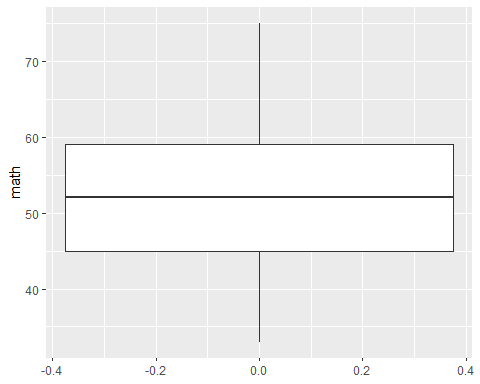
set.seed(1)  
X <- runif(50, 0, 10)  
Y <- 2\*X + rnorm(50, 0, 2)  
  
df <- data.frame(cbind(X,Y))  
  
df %>% ggplot(aes(x=X, y=Y)) +  
 geom\_point()



## Your Turn 6

* Using math, test versus .
* Create the boxplot.
* Calculate the 95% confidence interval for .
* What is your conclusion?

hsb2 %>% ggplot(aes(y=math)) +  
 geom\_boxplot()



t <- t.test(math, mu = 50)$statistic  
t > qt (0.975, 199, lower.tail = TRUE)

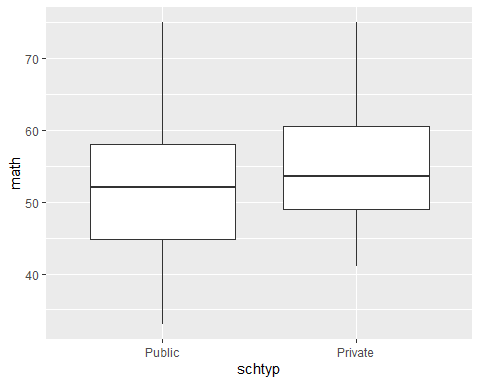
## t   
## TRUE

#Reject H0: mu = 50  
#Conclude H1: mu != 50

## Your Turn 7

* Compare math across schtyp by testing versus .
* Create the boxplots.
* Calculate the 95% confidence interval for .
* What is your conclusion?

hsb2 %>% ggplot(aes(x=schtyp, y=math)) +  
 geom\_boxplot() +  
 scale\_x\_discrete(labels=c("Public", "Private"))



t <- t.test(math~schtyp)$statistic  
t < qt(0.025, 55, 0.5, lower.tail = TRUE)

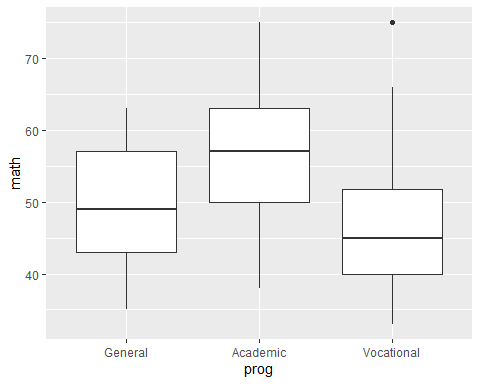
## t   
## FALSE

#Reject H0: mu1 = mu2  
#Conclude H1: mu1 != mu2

## Your Turn 8

* Compare math across prog by testing versus for some .
* Create the boxplots.
* Calculate the 95% confidence intervals for each .
* What is your conclusion?

hsb2 %>% ggplot(aes(x=prog,y=math)) +  
 geom\_boxplot() +  
 scale\_x\_discrete(labels=c("General", "Academic", "Vocational"))



m <- lm(math ~ prog, data = hsb2)  
anova(m)

## Analysis of Variance Table  
##   
## Response: math  
## Df Sum Sq Mean Sq F value Pr(>F)   
## prog 2 4002.1 2001.05 29.279 7.364e-12 \*\*\*  
## Residuals 197 13463.7 68.34   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

anova(m)$"F value"[1] > qf(0.95,2,197)

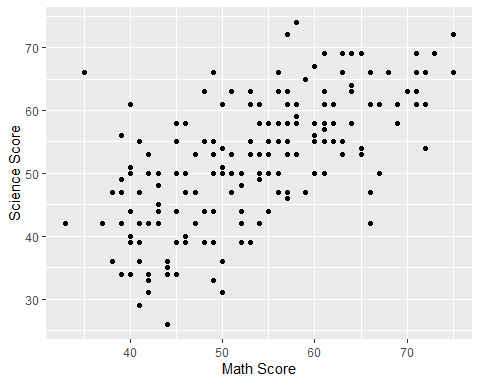
## [1] TRUE

#Reject H0: mu1 = mu2 = m3  
#Conclude H1: at least one muj != muj'

## Your Turn 9

* Test whether math is linearly related to science by testing versus .
* Create the scatterplot and overlay the fitted line.
* Calculate the 95% confidence interval for .
* What is your conclusion?

hsb2 %>% ggplot(aes(x=math,y=science)) +  
 geom\_point() +  
 labs(x = "Math Score",  
 y = "Science Score")



m <- lm(science~math, data = hsb2)  
summary(m)$coefficient[2,3] > qt(0.975, 198)

## [1] TRUE

#Reject H0: B1 = 0  
#Conclude H1: B1 != 0

# Take Aways

* Boxplots are good for comparing a continuous measure between groups.
* Scatterplots are good for investigating the association between two continuous measures.
* Student’s tests are appropriate for tests of a continuous measure in 1 or 2 groups.
* ANOVA is appropriate for tests of a continuous measure in 2+ groups.
* Simple linear regression is appropriate for testing whether two continuous measures are linearly associated.
* Use t.test(), lm(), and anova() to compare group means across 2 or more levels of .
* All these methods have assumptions that must be verified. Otherwise, non-parametric methods should be substituted.