Module 1 - Peer reviewed

Outline:

In this homework assignment, there are four objectives.

1. To assess your knowledge of ANOVA/ANCOVA models
2. To apply your understanding of these models to a real-world datasets

General tips:

1. Read the questions carefully to understand what is being asked.
2. This work will be reviewed by another human, so make sure that you are clear and concise in what you are attempting to explain or answer.

```
In [1]: # Load Required Packages
        library(tidyverse)
        library(ggplot2)
        library(dplyr)
```

```
── Attaching packages ─────────────────────────────────────── tidyverse 1
.3.0 ──

  ggplot2 3.3.0      purrr   0.3.4
  tibble  3.0.1      dplyr   0.8.5
  tidyr   1.0.2      stringr 1.4.0
  readr   1.3.1      forcats 0.5.0

── Conflicts ────────────────────────────────────────── tidyverse_conflicts
() ──
  dplyr::filter() masks stats::filter()
  dplyr::lag()    masks stats::lag()
```

Problem #1: Simulate ANCOVA Interactions

In this problem, we will work up to analyzing the following model to show how interaction terms work in an ANCOVA model.
$Y_i = \beta_0 + \beta_1 X + \beta_2 Z + \beta_3 XZ + \varepsilon_i$

This question is designed to enrich understanding of interactions in ANCOVA models. There is no additional coding required for this question, however we recommend messing around with the coefficents and plot as you see fit. Ultimately, this problem is graded based on written responses to questions asked in part **(a)** and **(b)**.

To demonstrate how interaction terms work in an ANCOVA model, let's generate some data. First, we consider the model

$Y_i = \beta_0 + \beta_1 X + \beta_2 Z + \varepsilon_i$

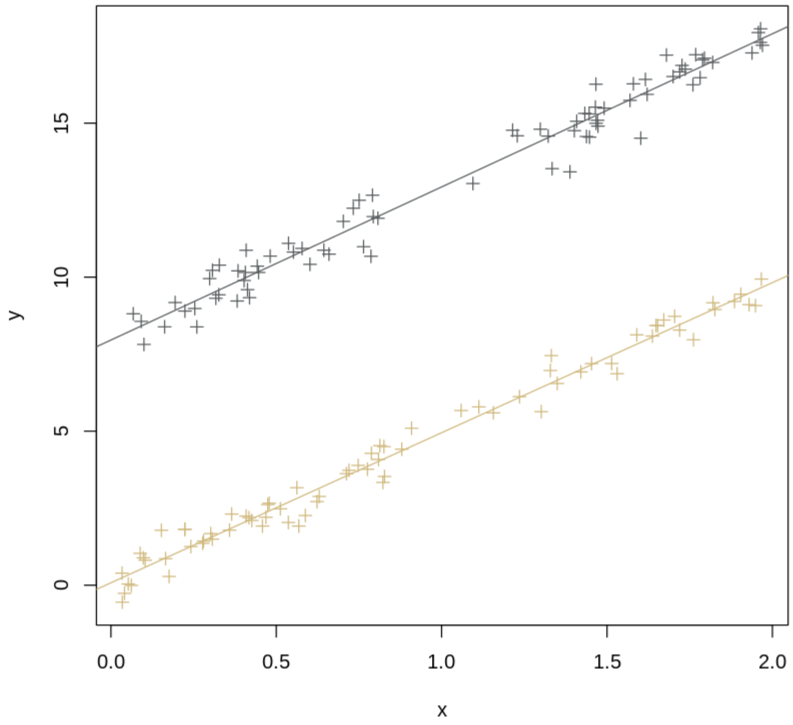
where X is a continuous covariate, Z is a dummy variable coding the levels of a two level factor, and $\varepsilon_i$iid ~ $N(0, \sigma^2)$. We choose values for the parameters below (b0,...,b2).

In [30]:

```
rm(list = ls())
set.seed(99)

#simulate data
n = 150
# choose these betas
b0 = 0; b1 = 5; b2 = 8; eps = rnorm(n, 0, 0.5);
x = runif(n,0,2); z = runif(n,-2,2);
z = ifelse(z > 0,1,0);
# create the model:
y = b0 + b1*x + b2*z + eps
df = data.frame(x = x,z = as.factor(z),y = y)
head(df)

#plot separate regression lines
with(df, plot(x,y, pch = 3, col = c("#CFB87C","#565A5C")[z]))
abline(coef(lm(y[z == 0] ~ x[z == 0], data = df)), col = "#CFB87C")
abline(coef(lm(y[z == 1] ~ x[z == 1], data = df)), col = "#565A5C")
```

A data.frame: 6 × 3

	x	z	y
	<dbl>	<fct>	<dbl>
1	0.09159879	1	8.5649752
2	1.96439135	1	18.0617858
3	0.57805656	1	10.9341972
4	0.03370108	0	0.3904347
5	1.82614045	0	8.9492833
6	0.71220319	0	3.6223530



1. (a) What happens with the slope and intercept of each of these lines?

In this case, we can think about having two separate regression lines--one for Y against X when the unit is in group Z = 0 and another for Y against X when the unit is in group Z = 1. What do we notice about the slope of each of these lines?

Solution The slopes are approximately parallel. Changing b0 affects the intercept (when x is 0). b1 determines the slope of the two lines. Also, it appears that increasing b1 causes the data points to be packed closer to the regression line. b2 determines the distance between the two lines. For example, setting b2 = 4 will result in a difference of 4 vertical units between the two regression lines.

1. (b) Now, let's add the interaction term (let β_3 = 3). What happens to the slopes of each line now?

The model now is of the form:
$Y_i = \beta_0 + \beta_1 X + \beta_2 Z + \beta_3 XZ + \varepsilon_i$

where X is a continuous covariate, Z is a dummy variable coding the levels of a two level factor, and $\varepsilon_i$iid ~ $N(0, \sigma^2)$. We choose values for the parameters below (b0,...,b3).

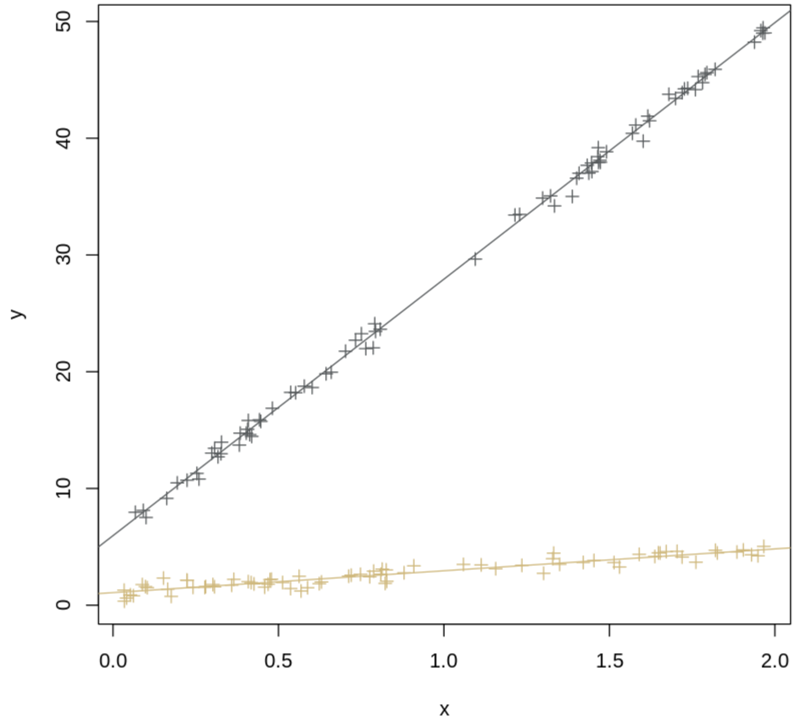
In [39]:

```
#simulate data
set.seed(99)
n = 150
# pick the betas
b0 = 1; b1 = 2; b2 = 5; b3 = 20; eps = rnorm(n, 0, 0.5);
```

```r
#create the model
y = b0 + b1*x + b2*z + b3*(x*z) + eps
df = data.frame(x = x,z = as.factor(z),y = y)
head(df)

lmod = lm(y ~ x + z, data = df)
lmodz0 = lm(y[z == 0] ~ x[z == 0], data = df)
lmodz1 = lm(y[z == 1] ~ x[z == 1], data = df)
# summary(lmod)
# summary(lmodz0)
# summary(lmodz1)

# lmodInt = lm(y ~ x + z + x*z, data = df)
# summary(lmodInt)

#plot separate regression lines
with(df, plot(x,y, pch = 3, col = c("#CFB87C","#565A5C")[z]))
abline(coef(lm(y[z == 0] ~ x[z == 0], data = df)), col = "#CFB87C")
abline(coef(lm(y[z == 1] ~ x[z == 1], data = df)), col = "#565A5C")
```

A data.frame: 6 × 3

	x	z	y
	<dbl>	<fct>	<dbl>
1	0.09159879	1	8.122155
2	1.96439135	1	49.456439
3	0.57805656	1	18.761159
4	0.03370108	0	1.289331
5	1.82614045	0	4.470862
6	0.71220319	0	2.485743



In this case, we can think about having two separate regression lines--one for Y against X when the unit is in group Z = 0 and another for Y against X when the unit is in group Z = 1. **What do you notice about the slope of each of these lines?**

Solution b0, b1, and b2 affect the regression lines in similar ways to problem 1(a). However, change in b3 leads to a much greater change in the black regression line, while leaving the gold line seemingly unaffected. The greater the value of b3, the greater the slope of the black line.

Problem #2

In this question, we ask you to analyze the `mtcars` dataset. The goal if this question will be to try to explain the variability in miles per gallon (mpg) using transmission type (am), while adjusting for horsepower (hp).

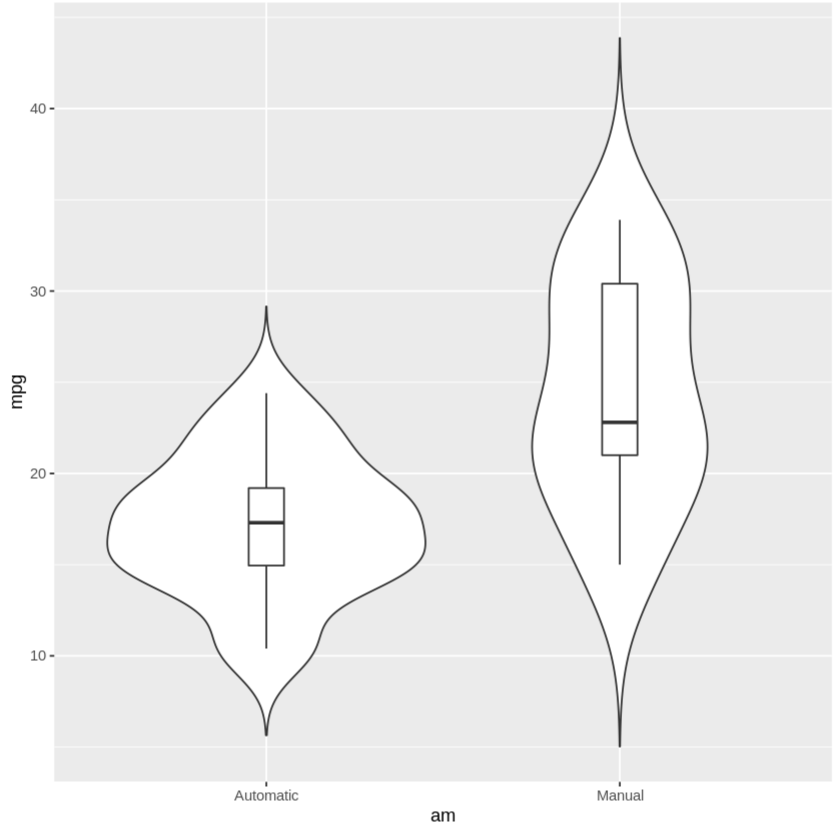
To load the data, use `data(mtcars)`

2. (a) Rename the levels of am from 0 and 1 to "Automatic" and "Manual" (one option for this is to use the revalue() function in the plyr package). Then, create a boxplot (or violin plot) of mpg against am. What do you notice? Comment on the plot

In [18]:

```
data(mtcars)

# your code here
```

```
library(plyr)
mtcars <- mtcars %>%
                    mutate(am = mapvalues(am, from = c(0, 1), to = c("Automati
c", "Manual")))
p = ggplot(mtcars, aes(x=am, y=mpg)) + geom_violin(trim = FALSE) + geom_bo
xplot(width = 0.1)
p
```



Solution Based on the plots, automatic transmission has a slightly lower fuel efficiency compared to manual trasnsmission. There is a greater variability in manual transmission mpg, since drivers determine the fuel efficiency based on driving style and frequency of errors while shifting gears.

2. (b) Calculate the mean difference in mpg for the Automatic group compared to the Manual group.

In [19]:

```
# your code here
lm(formula = mpg ~ am, data = mtcars)
Automatic_mean = 17.147
Manual_mean = 17.147 + 7.245
Manual_mean - Automatic_mean
```

```
Call:
lm(formula = mpg ~ am, data = mtcars)

Coefficients:
(Intercept)     amManual  
     17.147        7.245  
```

7.245

Solution The mean difference in mpg between the two transmission methods is 7.245 mpg.

2. (c) Construct three models:

1. An ANOVA model that checks for differences in mean mpg across different transmission types.
2. An ANCOVA model that checks for differences in mean mpg across different transmission types, adjusting for horsepower.
3. An ANCOVA model that checks for differences in mean mpg across different transmission types, adjusting for horsepower and for interaction effects between horsepower and transmission type.

Using these three models, determine whether or not the interaction term between transmission type and horsepower is significant.

In [20]:

```
# your code here
mod1 = lm(mpg ~ am, data = mtcars)
mod2 = lm(mpg ~ hp + am, data = mtcars)
mod3 = lm(mpg ~ hp + am + am:hp, data = mtcars)
summary(mod1)
summary(mod2)
summary(mod3)
```

```
Call:
lm(formula = mpg ~ am, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.3923 -3.0923 -0.2974  3.2439  9.5077 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   17.147      1.125  15.247 1.13e-15 ***
amManual       7.245      1.764   4.106 0.000285 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.902 on 30 degrees of freedom
Multiple R-squared:  0.3598,	Adjusted R-squared:  0.3385 
F-statistic: 16.86 on 1 and 30 DF,  p-value: 0.000285

Call:
lm(formula = mpg ~ hp + am, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.3843 -2.2642  0.1366  1.6968  5.8657 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 26.584914   1.425094  18.655  < 2e-16 ***
hp          -0.058888   0.007857  -7.495 2.92e-08 ***
amManual     5.277085   1.079541   4.888 3.46e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 2.909 on 29 degrees of freedom
Multiple R-squared:  0.782,	Adjusted R-squared:  0.767 
F-statistic: 52.02 on 2 and 29 DF,  p-value: 2.55e-10

Call:
lm(formula = mpg ~ hp + am + am:hp, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.3818 -2.2696  0.1344  1.7058  5.8752 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 26.6248479  2.1829432  12.197 1.01e-12 ***
hp          -0.0591370  0.0129449  -4.568 9.02e-05 ***
amManual     5.2176534  2.6650931   1.958   0.0603 .  
hp:amManual  0.0004029  0.0164602   0.024   0.9806    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.961 on 28 degrees of freedom
Multiple R-squared:  0.782,	Adjusted R-squared:  0.7587 
F-statistic: 33.49 on 3 and 28 DF,  p-value: 2.112e-09
```

Solution

Comparing each of the summary, we can see that the interaction term is not significant, having a p-value much larger than alpha = 0.05 level. However, it can also be noted that the transmission method becomes less significant.

2. (d) Construct a plot of mpg against horsepower, and color points based in transmission type. Then, overlay the regression lines with the interaction term, and the lines without. How are these lines consistent with your answer in (b) and (c)?

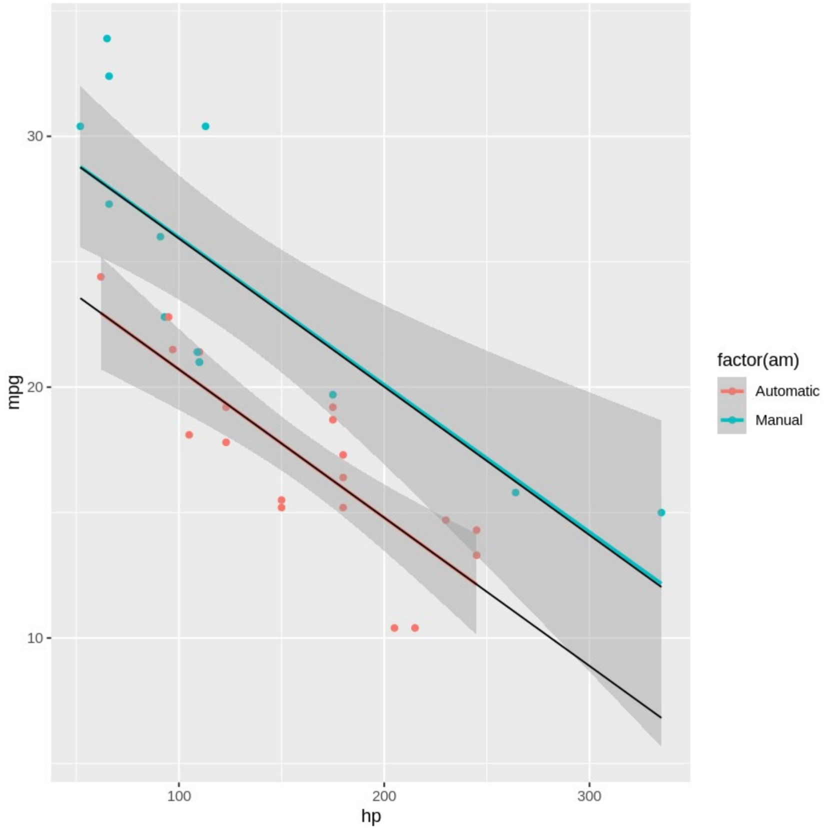
In [40]:

```
# your code here
library(tidyverse)
library(ggplot2)


equation0 = function(x){coef(mod3)[2]*x + coef(mod3)[1]}
equation1 = function(x){coef(mod3)[2]*x + coef(mod3)[1] + coef(mod3)[3] + coef(mod3)[4]}
equation0
equation1
plot1 = ggplot(mtcars, aes(x = hp, y = mpg, color=factor(am))) + geom_point() + geom_smooth(method = 'lm') + stat_function(fun=equation0, geom="path", color = "black") + stat_function(fun=equation1, geom="path", color="black")
plot1
```

```
function (x) 
{
    coef(mod3)[2] * x + coef(mod3)[1]
}
```

```
function (x)
{
    coef(mod3)[2] * x + coef(mod3)[1] + coef(mod3)[3] + coef(mod3)[4]
}
```

```
`geom_smooth()` using formula 'y ~ x'
```



Solution The two black lines represent the regression lines with interaction terms included. For automatic transmission, we can see that the black line completely overlaps the red line (regression line without the interaction term). However, there is a very slight offset between the black and green regression lines for manual transmission, as observed in parts (a) and (b). The offset is small enough to be negligible, confirming that the interaction term is not significant.

In [ ]: