m1-peer-reviewed

June 27, 2023

1 Module 1 - Peer reviewed

1.0.1 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.

[1]:
```r
# 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()
```

1.0.2 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

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

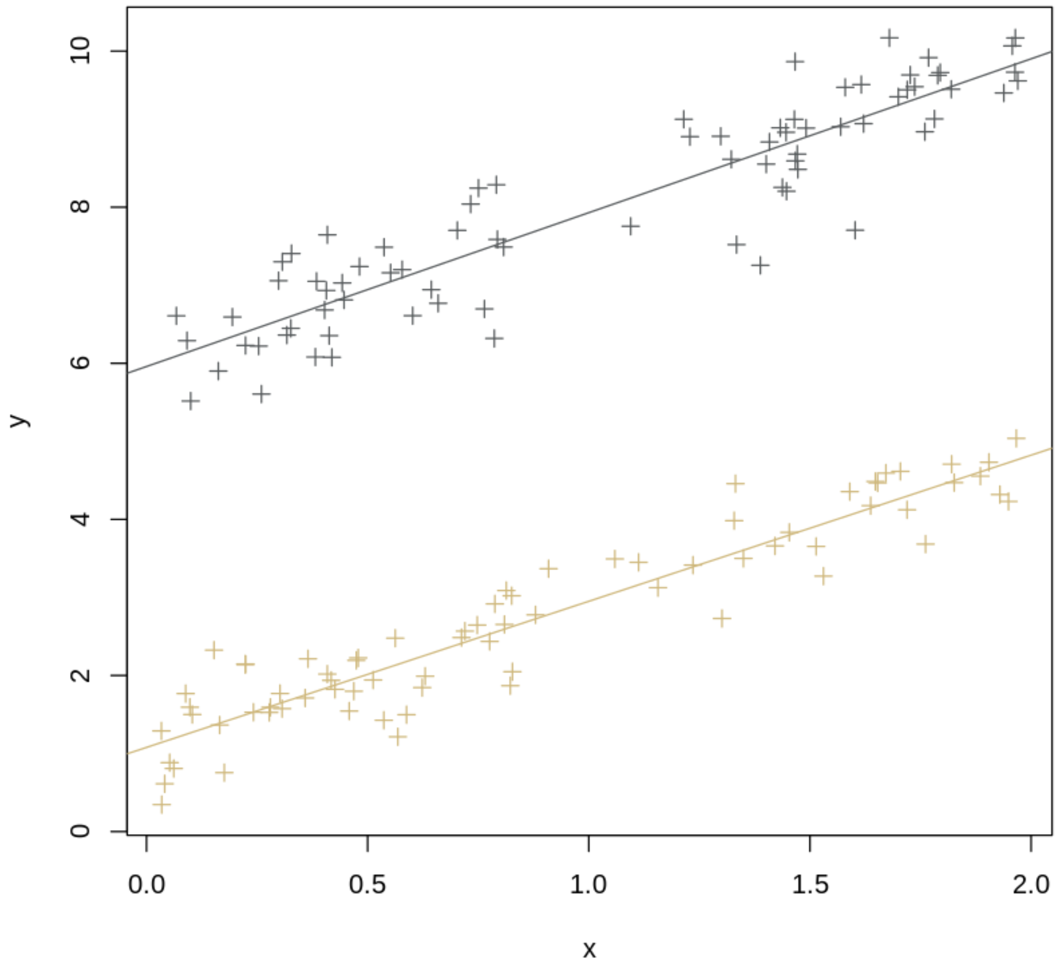
where X is a continuous covariate, Z is a dummy variable coding the levels of a two level factor, and $\varepsilon_i \overset{iid}{\sim} N(0, \sigma^2)$. We choose values for the parameters below (b0,...,b2).

[3]:

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

#simulate data
n = 150
# choose these betas
b0 = 1; b1 = 2; b2 = 5; 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 <dbl>	z <fct>	y <dbl>
1	0.09159879	1	6.290179
2	1.96439135	1	10.168612
3	0.57805656	1	7.200027
4	0.03370108	0	1.289331
5	1.82614045	0	4.470862
6	0.71220319	0	2.485743



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?

When we examine the two regression lines in our plot:

For the line where $Z = 0$, the slope is determined by β_1, and the intercept is β_0.

For the line where $Z = 1$, the slope combines the effects of β_1 and the interaction term $\beta_3 * X$, and the intercept is the sum of β_0 and β_2.

These relationships become more complex once an interaction term is introduced, as it allows the slope and intercept to change depending on both X and Z.

1. (b) Now, let's add the interaction term (let $\beta_3 = 3$). What happens to the slopes of each line now? The model now is of the form:

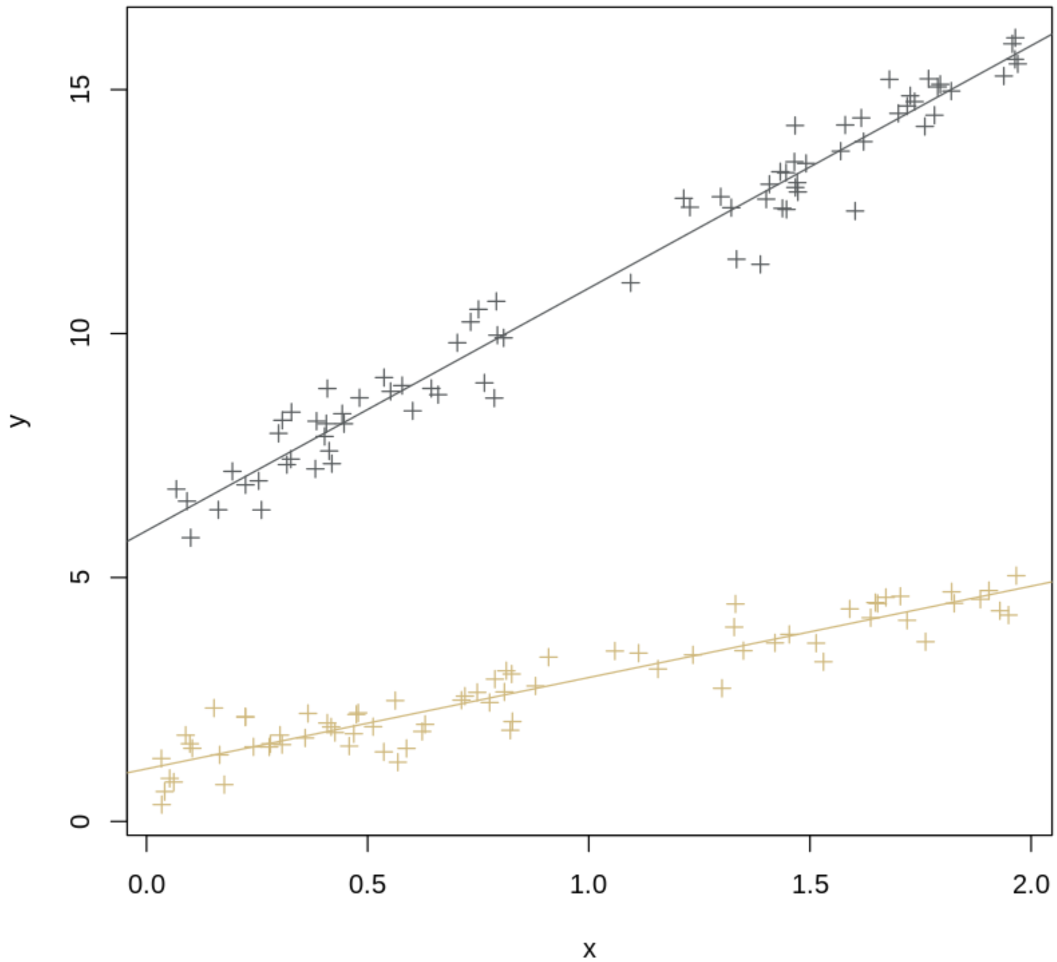
$$Y_i = \beta_0 + \beta_1 X + \beta_2 Z + \beta_3 XZ + \varepsilon_i$$

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

[2]:
```r
#simulate data
set.seed(99)
n = 150
# pick the betas
b0 = 1; b1 = 2; b2 = 5; b3 = 3; eps = rnorm(n, 0, 0.5);

#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	6.564975
2	1.96439135	1	16.061786
3	0.57805656	1	8.934197
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?**

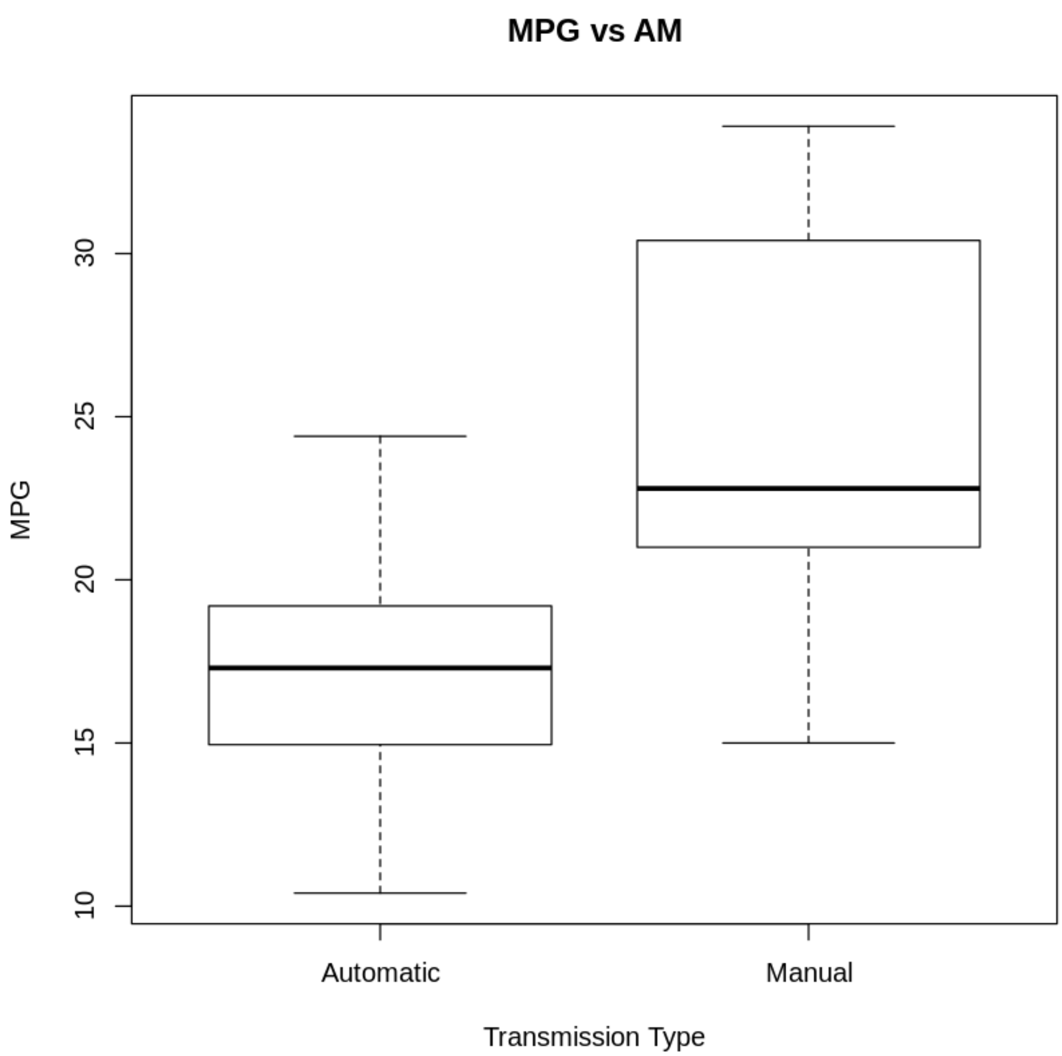
1.1 Problem #2

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

[5]:
```
data(mtcars)

# your code here
mtcars$am =factor(mtcars$am, labels = c("Automatic", "Manual"))
boxplot(mpg ~ am, data = mtcars, main = "MPG vs AM", xlab = "Transmission␣
↪Type", ylab = "MPG")
```



The boxplot offers a quick comparison of fuel economy ('mpg') between automatic and manual cars. Preliminary observations suggest a difference in 'mpg' based on transmission type, which needs further statistical validation.

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

[6]:
```
# your code here
mean_difference = mean(mtcars$mpg[mtcars$am == "Manual"]) -␣
↪mean(mtcars$mpg[mtcars$am == "Automatic"])
mean_difference
```

7.24493927125506

This code calculates the mean of 'mpg' for both transmission types separately and then computes the difference. The resulting value represents the mean difference in 'mpg' between the Manual and Automatic groups.

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.

[7]:
```
# your code here
model1 = lm(mpg ~ am, data = mtcars)
summary(model1)
model2 = lm(mpg ~ am + hp, data = mtcars)
summary(model2)
model3 = lm(mpg ~ am * hp, data = mtcars)
summary(model3)
```

```
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 ~ am + hp, 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 ***
amManual     5.277085   1.079541   4.888 3.46e-05 ***
hp          -0.058888   0.007857  -7.495 2.92e-08 ***
---
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 ~ 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 ***
amManual     5.2176534  2.6650931   1.958   0.0603 .
hp          -0.0591370  0.0129449  -4.568 9.02e-05 ***
amManual:hp  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
```

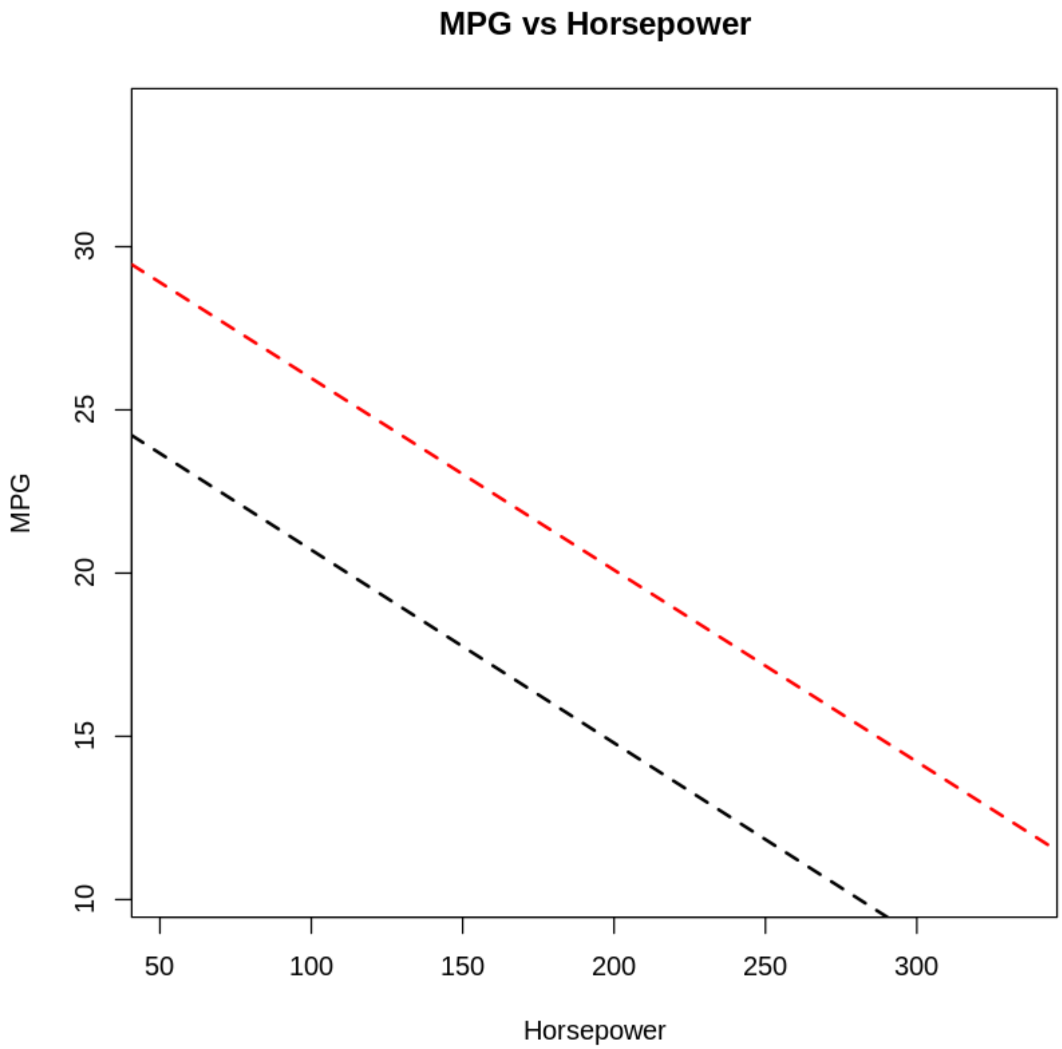
To interpret our results, we need to check the p-value associated with the am:hp term in the summary of model3. If the p-value is less than the typical significance level (0.05), this suggests the interaction term is significant. After evaluating the results, we can see that the p-values are significant.

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)?

[8]:
```
# your code here
plot(mtcars$hp, mtcars$mpg, col = mtcars$am + 1,
     main = "MPG vs Horsepower", xlab = "Horsepower", ylab = "MPG",
     pch = 16, cex = 1.5)
auto_lm = lm(mpg ~ hp, data = mtcars, subset = (am == "Automatic"))
manu_lm = lm(mpg ~ hp, data = mtcars, subset = (am == "Manual"))
abline(auto_lm, col = "black", lwd = 2, lty = 2) # Automatic line
abline(manu_lm, col = "red", lwd = 2, lty = 2) # Manual line
auto_lm_interact = lm(mpg ~ hp + hp:am, data = mtcars, subset = (am ==␣
↪"Automatic"))
manu_lm_interact = lm(mpg ~ hp + hp:am, data = mtcars, subset = (am ==␣
↪"Manual"))
abline(auto_lm_interact, col = "black", lwd = 2)
abline(manu_lm_interact, col = "red", lwd = 2)
legend("topright", legend = c("Automatic", "Manual"),
       col = c("black", "red"), pch = 16, cex = 1)
```

```
Warning message in Ops.factor(mtcars$am, 1):
"‘+’ not meaningful for factors"
```

```
        Error in `contrasts<-`(`*tmp*`, value = contr.funs[1 + isOF[nn]]):␣
↪contrasts can be applied only to factors with 2 or more levels
    Traceback:


        1. lm(mpg ~ hp + hp:am, data = mtcars, subset = (am == "Automatic"))

        2. model.matrix(mt, mf, contrasts)

        3. model.matrix.default(mt, mf, contrasts)

        4. `contrasts<-`(`*tmp*`, value = contr.funs[1 + isOF[nn]])

        5. stop("contrasts can be applied only to factors with 2 or more levels")
```



this plot, the dashed lines represent the fitted values from the models without interaction and the solid lines represent the fitted values from the models with interaction. Black color represents "Automatic" and red represents "Manual".

The slopes of the regression lines visually represent the effect of horsepower on mpg for each transmission type. A difference in slopes between transmission types implies an interaction effect. The plot seem to align with our answers from parts (b) and (c).