Week 7 - Homework

STAT 420, Summer 2023, Adriane Yi

2023 07 03

Exercise 1 (EPA Emissions Data)

For this exercise, we will use the data stored in `epa2017.csv`. It contains detailed descriptions of vehicles manufactured in 2017 that were used for fuel economy testing as performed by the Environment Protection Agency. The variables in the dataset are:

- `Make` - Manufacturer
- `Model` - Model of vehicle
- `ID` - Manufacturer defined vehicle identification number within EPA's computer system (not a VIN number)
- `disp` - Cubic inch displacement of test vehicle
- `type` - Car, truck, or both (for vehicles that meet specifications of both car and truck, like smaller SUVs or crossovers)
- `horse` - Rated horsepower, in foot-pounds per second
- `cyl` - Number of cylinders
- `lockup` - Vehicle has transmission lockup; N or Y
- `drive` - Drivetrain system code
  - A = All-wheel drive
  - F = Front-wheel drive
  - P = Part-time 4-wheel drive
  - R = Rear-wheel drive
  - 4 = 4-wheel drive
- `weight` - Test weight, in pounds
- `axleratio` - Axle ratio
- `nvratio` - n/v ratio (engine speed versus vehicle speed at 50 mph)
- `THC` - Total hydrocarbons, in grams per mile (g/mi)
- `CO` - Carbon monoxide (a regulated pollutant), in g/mi
- `CO2` - Carbon dioxide (the primary byproduct of all fossil fuel combustion), in g/mi
- `mpg` - Fuel economy, in miles per gallon

We will attempt to model `CO2` using both `horse` and `type`. In practice, we would use many more predictors, but limiting ourselves to these two, one numeric and one factor, will allow us to create a number of plots.

Load the data, and check its structure using `str()`. Verify that `type` is a factor; if not, coerce it to be a factor.

(a)

Do the following:

- Make a scatterplot of `CO2` versus `horse`. Use a different color point for each vehicle `type`.
- Fit a simple linear regression model with `CO2` as the response and only `horse` as the predictor.
- Add the fitted regression line to the scatterplot. Comment on how well this line models the data.
- Give an estimate for the average change in `CO2` for a one foot-pound per second increase in `horse` for a vehicle of type `car`.
- Give a 90% prediction interval using this model for the `CO2` of a Subaru Impreza Wagon, which is a vehicle with 148 horsepower and is considered type `Both`. (Interestingly, the dataset gives the wrong drivetrain for most Subarus in this dataset, as they are almost all listed as `F`, when they are in fact all-wheel drive.)

```r
# Read Data
library(readr)

epa = read.csv("epa2017.csv")
```

```r
is.factor(epa$type)
```

Importing Data

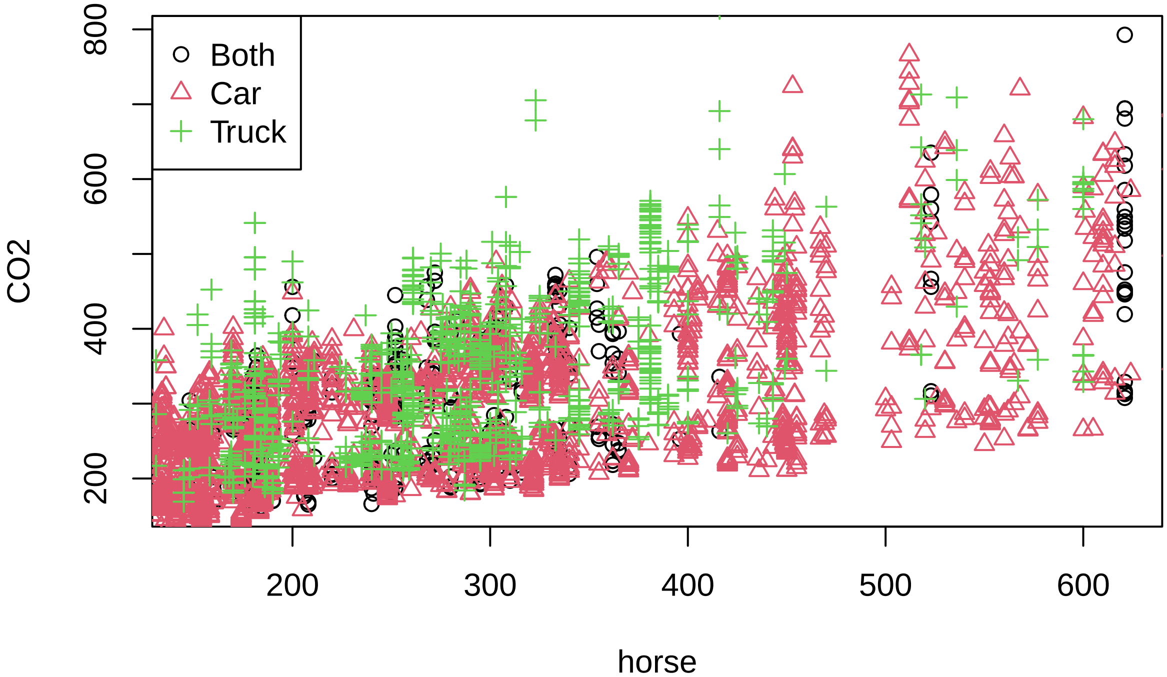
```
## [1] FALSE
```

```r
epa$type = as.factor(epa$type)
class(epa$type)
```

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

```r
plot(CO2 ~ horse, data = subset(epa,type=="Both"), col =1, pch = 1)
points(CO2 ~ horse, data = subset(epa,type=="Car"), col =2, pch = 2)
points(CO2 ~ horse, data = subset(epa,type=="Truck"), col =3, pch = 3)
legend("topleft", c("Both", "Car", "Truck"), col = c(1, 2, 3), pch = c(1, 2, 3))
```

Make a scatterplot of `CO2` versus `horse`. Use a different color point for each vehicle `type`.

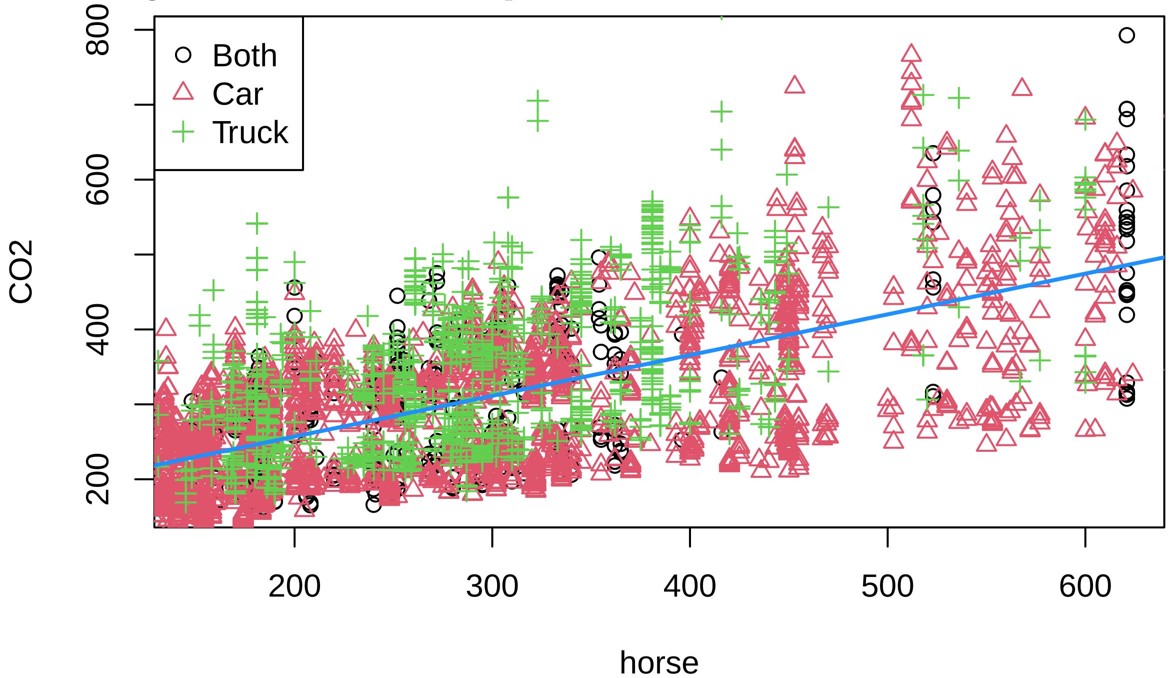



```
sim_model = lm(CO2 ~ horse, data = epa)
```

Fit a simple linear regression model with `CO2` as the response and only `horse` as the predictor.

```
plot(CO2 ~ horse, data = subset(epa,type=="Both"), col =1, pch = 1)
points(CO2 ~ horse, data = subset(epa,type=="Car"), col =2, pch = 2)
points(CO2 ~ horse, data = subset(epa,type=="Truck"), col =3, pch = 3)
legend("topleft", c("Both", "Car", "Truck"), col = c(1, 2, 3), pch = c(1, 2, 3))
abline(sim_model, lwd = 2, col = "dodgerblue")
```

Add the fitted regression line to the scatterplot. Comment on how well this line models the



data.

Solution:

The regression model, with CO2 as response and horse as the predictor, seems to be underfitting Truck data and overfitting Car data. It seems for Both data, the regression model fits fairly well.

```
coef(sim_model)[2]
```

Give an estimate for the average change in `CO2` for a one foot-pound per second increase in `horse` for a vehicle of type `car`.

```
##  horse
## 0.5436
```

Solution:

For a one foot-pound/sec increase in `horse`, for a vehicle of type `car`, 0.5436 is the estimate for average change in CO2

```
newdata = data.frame(horse = 148, type= "Both")
(prediction = predict(sim_model, newdata = newdata, interval = "prediction", level = 0.90))
```

Give a 90% prediction interval using this model for the `CO2` of a Subaru Impreza Wagon, which is a vehicle with 148 horsepower and is considered type `Both`. (Interestingly, the dataset gives the wrong drivetrain for most Subarus in this dataset, as they are almost all listed as `F`, when they are in fact all-wheel drive.)

```
##     fit  lwr upr
## 1 228.8 91.5 366
```

```r
prediction[2]
```

```
## [1] 91.5
```

Solution:.

90% prediction interval is [91.5033 , 366.0446]

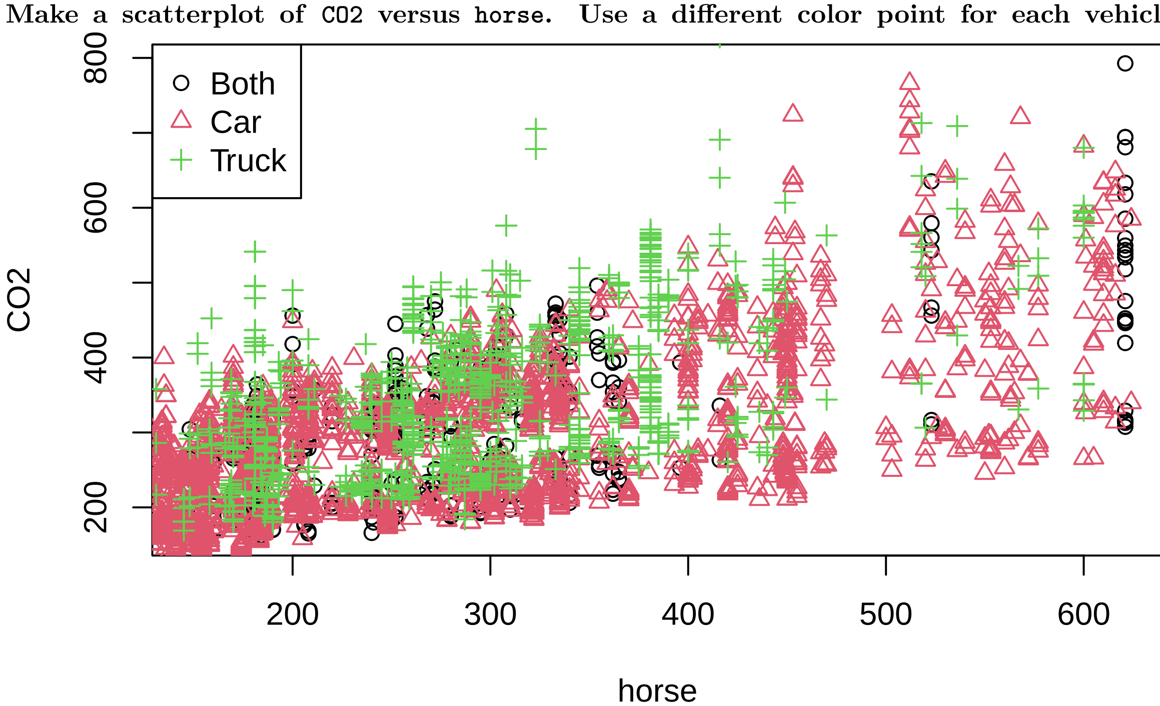
(b)

Do the following:

- Make a scatterplot of `CO2` versus `horse`. Use a different color point for each vehicle `type`.
- Fit an additive multiple regression model with `CO2` as the response and `horse` and `type` as the predictors.
- Add the fitted regression "lines" to the scatterplot with the same colors as their respective points (one line for each vehicle type). Comment on how well this line models the data.
- Give an estimate for the average change in `CO2` for a one foot-pound per second increase in `horse` for a vehicle of type `car`.
- Give a 90% prediction interval using this model for the `CO2` of a Subaru Impreza Wagon, which is a vehicle with 148 horsepower and is considered type `Both`.

```r
plot(CO2 ~ horse, data = subset(epa,type=="Both"), col =1, pch = 1)
points(CO2 ~ horse, data = subset(epa,type=="Car"), col =2, pch = 2)
points(CO2 ~ horse, data = subset(epa,type=="Truck"), col =3, pch = 3)
legend("topleft", c("Both", "Car", "Truck"), col = c(1, 2, 3), pch = c(1, 2, 3))
```

Make a scatterplot of `CO2` versus `horse`. Use a different color point for each vehicle `type`.



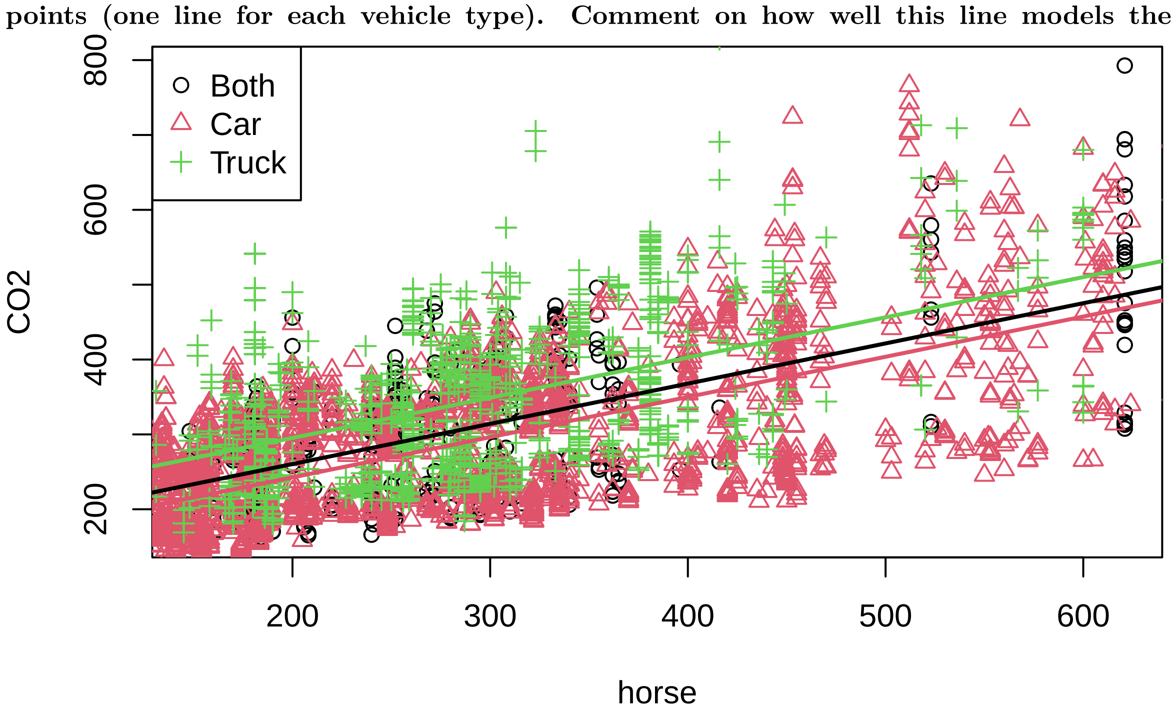
```r
add_model = lm(CO2 ~ horse + type, data = epa)

int_Both = coef(add_model)[1]
int_Car = coef(add_model)[1] + coef(add_model)[3]
int_Truck = coef(add_model)[1] + coef(add_model)[4]
slope = coef(add_model)[2]
```

Fit an additive multiple regression model with `CO2` as the response and `horse` and `type` as the predictors.

```r
plot(CO2 ~ horse, data = subset(epa,type=="Both"), col =1, pch = 1)
points(CO2 ~ horse, data = subset(epa,type=="Car"), col =2, pch = 2)
points(CO2 ~ horse, data = subset(epa,type=="Truck"), col =3, pch = 3)
legend("topleft", c("Both", "Car", "Truck"), col = c(1, 2, 3), pch = c(1, 2, 3))
abline(int_Both, slope, col = 1, lty = 1, lwd = 2)
abline(int_Car, slope, col = 2, lty = 1, lwd = 2)
abline(int_Truck, slope, col = 3, lty = 1, lwd = 2)
```

Add the fitted regression "lines" to the scatterplot with the same colors as their respective points (one line for each vehicle type). Comment on how well this line models the data.



Solution:.

These lines corresponding to each type are fitting the data better than the previous single line. However, notice the slope for Truck needs to be adjusted(increased) to fit data better.

```r
coef(add_model)[2]
```

Give an estimate for the average change in `CO2` for a one foot-pound per second increase in `horse` for a vehicle of type `car`.

```
##  horse
## 0.5372
```

Estimate for the average change in CO2 in a one foot-pound/sec increase in `horse` for a vehicle of type `car` is 0.5372.

```r
newdata = data.frame(horse = 148, type = "Both")
(prediction = predict(add_model, newdata = newdata, interval = "prediction", level = 0.90))
```

Give a 90% prediction interval using this model for the `CO2` of a Subaru Impreza Wagon, which is a vehicle with 148 horsepower and is considered type `Both`.

```
##     fit lwr   upr
## 1 232.4 100 364.9
```

Solution:.
90% prediction interval is [100.0012, 364.8952].

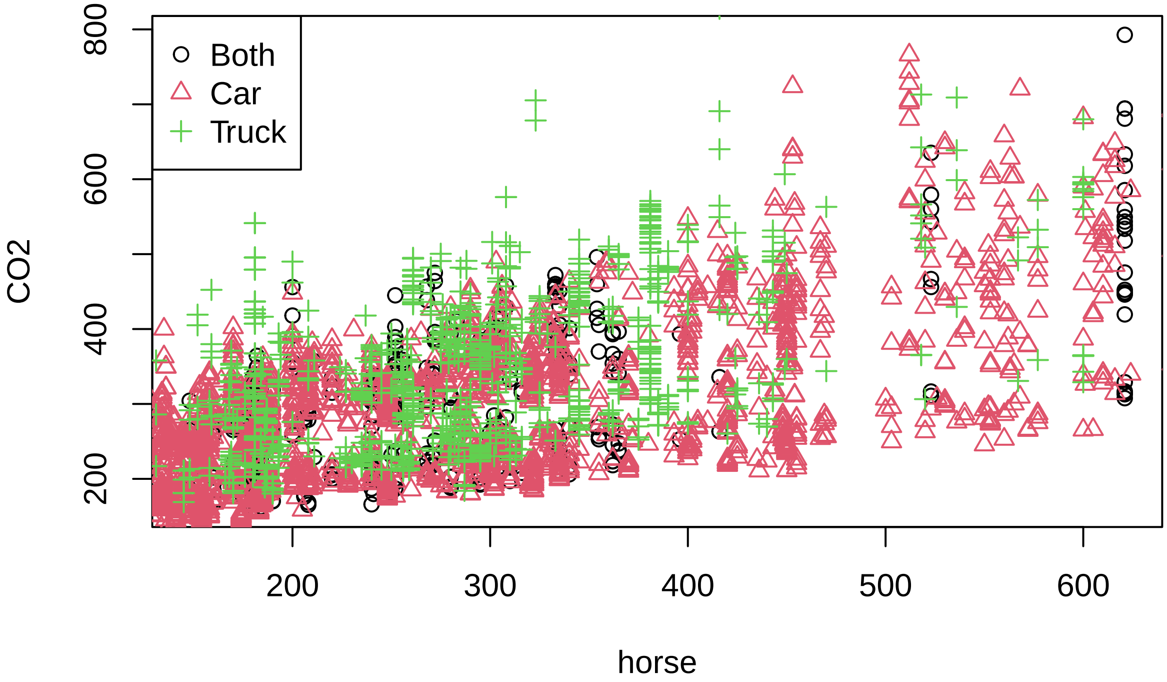
(c)

Do the following:

- Make a scatterplot of `CO2` versus `horse`. Use a different color point for each vehicle `type`.
- Fit an interaction multiple regression model with `CO2` as the response and `horse` and `type` as the predictors.
- Add the fitted regression "lines" to the scatterplot with the same colors as their respective points (one line for each vehicle type). Comment on how well this line models the data.
- Give an estimate for the average change in `CO2` for a one foot-pound per second increase in `horse` for a vehicle of type `car`.
- Give a 90% prediction interval using this model for the `CO2` of a Subaru Impreza Wagon, which is a vehicle with 148 horsepower and is considered type `Both`.

```r
plot(CO2 ~ horse, data = subset(epa,type=="Both"), col =1, pch = 1)
points(CO2 ~ horse, data = subset(epa,type=="Car"), col =2, pch = 2)
points(CO2 ~ horse, data = subset(epa,type=="Truck"), col =3, pch = 3)
legend("topleft", c("Both", "Car", "Truck"), col = c(1, 2, 3), pch = c(1, 2, 3))
```

Make a scatterplot of `CO2` versus `horse`. Use a different color point for each vehicle `type`.




```
int_model = lm(CO2 ~ horse * type, data = epa)
coef(int_model)
```

Fit an interaction multiple regression model with `CO2` as the response and `horse` and `type` as the predictors.

```
##     (Intercept)           horse         typeCar       typeTruck   horse:typeCar
##       141.96745         0.57781        -2.82108        26.04407        -0.05522
## horse:typeTruck
##         0.02764
```

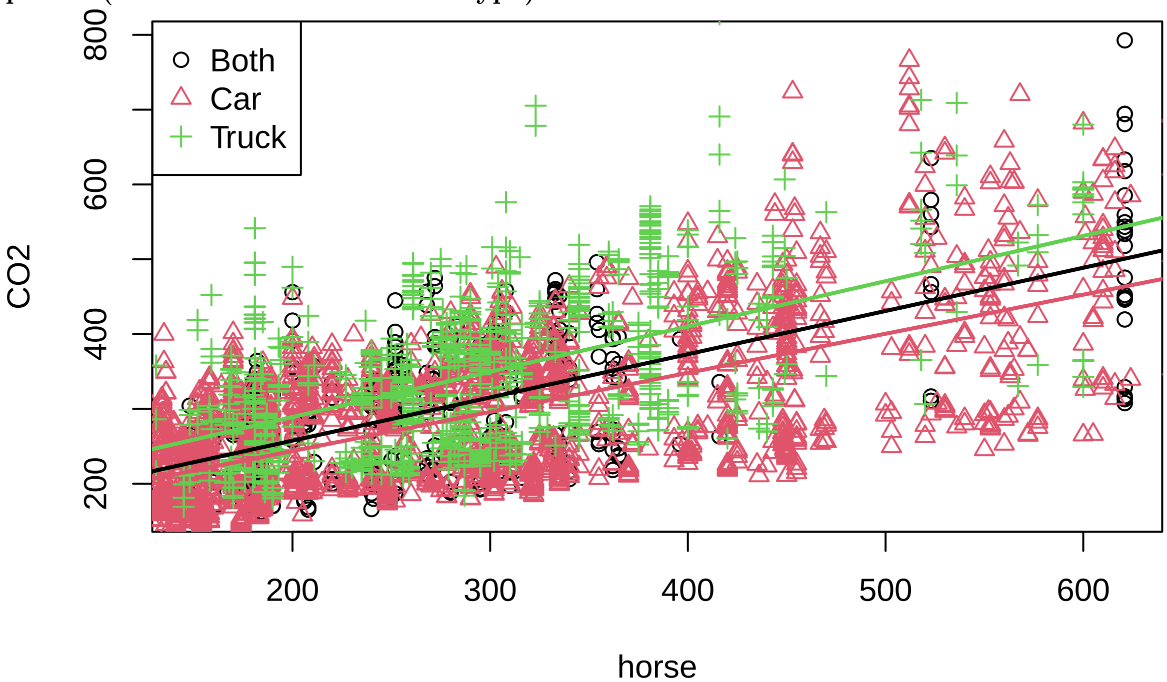
```
int_Both = coef(int_model)["(Intercept)"]
int_Car = coef(int_model)["(Intercept)"] + coef(int_model)["typeCar"]
int_Truck = coef(int_model)["(Intercept)"] + coef(int_model)["typeTruck"]

slope_Both = coef(int_model)["horse"]
slope_Car = coef(int_model)["horse"] + coef(int_model)["horse:typeCar"]
slope_Truck = coef(int_model)["horse"] + coef(int_model)["horse:typeTruck"]
```

```
plot(CO2 ~ horse, data = subset(epa,type=="Both"), col =1, pch = 1)
points(CO2 ~ horse, data = subset(epa,type=="Car"), col =2, pch = 2)
points(CO2 ~ horse, data = subset(epa,type=="Truck"), col =3, pch = 3)
legend("topleft", c("Both", "Car", "Truck"), col = c(1, 2, 3), pch = c(1, 2, 3))
abline(int_Both, slope_Both, col = 1, lty = 1, lwd = 2)
```

```r
abline(int_Car, slope_Car, col = 2, lty = 1, lwd = 2)
abline(int_Truck, slope_Truck, col = 3, lty = 1, lwd = 2)
```

Add the fitted regression "lines" to the scatterplot with the same colors as their respective points (one line for each vehicle type). Comment on how well this line models the data.




```r
slope_Car = coef(int_model)["horse"] + coef(int_model)["horse:typeCar"]
```

Give an estimate for the average change in `CO2` for a one foot-pound per second increase in `horse` for a vehicle of type `car`. **Solution:** 0.5226

```r
newdata = data.frame(horse = 148, type = "Both")
(prediction = predict(int_model, newdata = newdata, interval = "prediction", level = 0.90))
```

Give a 90% prediction interval using this model for the `CO2` of a Subaru Impreza Wagon, which is a vehicle with 148 horsepower and is considered type `Both`.

```
##     fit   lwr upr
## 1 227.5 95.01 360
```

Solution:. 90% prediction interval : [95.0055, 359.9619]

(d) Based on the previous plots, you probably already have an opinion on the best model. Now use an ANOVA F-test to compare the additive and interaction models. Based on this test and a significance level of $\alpha = 0.10$, which model is preferred?

```r
anova(add_model, int_model)
```

```
## Analysis of Variance Table
##
## Model 1: CO2 ~ horse + type
## Model 2: CO2 ~ horse * type
##   Res.Df      RSS Df Sum of Sq    F Pr(>F)
## 1   4033 26073761
## 2   4031 26007441  2     66320 5.14 0.0059 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
p_value = anova(add_model, int_model)[2, "Pr(>F)"]
alpha = 0.10
if(p_value < alpha){
  result = "Reject H0"
}else{
  result = "Fail to Reject H0"
}
result
```

```
## [1] "Reject H0"
```

** Solution:**. Based on the previous plots, it seems the interaction model is fitting data better. Based on the p-value from F-test, since it is low and smaller than alpha, the interaction model is also preferred.

Exercise 2 (Hospital SUPPORT Data, White Blood Cells)

For this exercise, we will use the data stored in `hospital.csv`. It contains a random sample of 580 seriously ill hospitalized patients from a famous study called "SUPPORT" (Study to Understand Prognoses Preferences Outcomes and Risks of Treatment). As the name suggests, the purpose of the study was to determine what factors affected or predicted outcomes, such as how long a patient remained in the hospital. The variables in the dataset are:

- `Days` - Days to death or hospital discharge
- `Age` - Age on day of hospital admission
- `Sex` - Female or male
- `Comorbidity` - Patient diagnosed with more than one chronic disease
- `EdYears` - Years of education
- `Education` - Education level; high or low
- `Income` - Income level; high or low
- `Charges` - Hospital charges, in dollars
- `Care` - Level of care required; high or low
- `Race` - Non-white or white
- `Pressure` - Blood pressure, in mmHg
- `Blood` - White blood cell count, in gm/dL
- `Rate` - Heart rate, in bpm

For this exercise, we will use `Age`, `Education`, `Income`, and `Sex` in an attempt to model `Blood`. Essentially, we are attempting to model white blood cell count using only demographic information.

(a) Load the data, and check its structure using `str()`. Verify that `Education`, `Income`, and `Sex` are factors; if not, coerce them to be factors. What are the levels of `Education`, `Income`, and `Sex`?

```
hospital = read.csv("hospital.csv")
str(hospital)
```

```
## 'data.frame':    580 obs. of  13 variables:
##  $ Days       : int  8 14 21 4 11 9 25 26 9 16 ...
##  $ Age        : num  42.3 63.7 41.5 42 52.1 ...
##  $ Sex        : chr  "female" "female" "male" "male" ...
##  $ Comorbidity: chr  "no" "no" "yes" "yes" ...
##  $ EdYears    : int  11 22 18 16 8 12 12 13 16 30 ...
##  $ Education  : chr  "low" "high" "high" "high" ...
##  $ Income     : chr  "high" "high" "high" "high" ...
##  $ Charges    : num  9914 283303 320843 4173 13414 ...
##  $ Care       : chr  "low" "high" "high" "low" ...
##  $ Race       : chr  "non-white" "white" "white" "white" ...
##  $ Pressure   : int  84 69 66 97 89 57 99 115 93 102 ...
##  $ Blood      : num  11.3 30.1 0.2 10.8 6.4 ...
##  $ Rate       : int  94 108 130 88 92 114 150 132 86 90 ...
```

(b) Fit an additive multiple regression model with `Blood` as the response using `Age`, `Education`, `Income`, and `Sex` as predictors. What does `R` choose as the reference level for `Education`, `Income`, and `Sex`?

```
hospital_add_model = lm(Blood ~ Age + Education + Income + Sex, data = hospital)
summary(hospital_add_model)
```

```
##
## Call:
## lm(formula = Blood ~ Age + Education + Income + Sex, data = hospital)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -12.79  -5.13  -1.58   3.07  47.37
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   10.8662     1.4284    7.61  1.1e-13 ***
## Age            0.0283     0.0207    1.37   0.1725
## Educationlow   0.5967     0.7557    0.79   0.4301
## Incomelow      0.1867     0.7139    0.26   0.7938
## Sexmale       -1.8714     0.6613   -2.83   0.0048 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.88 on 575 degrees of freedom
## Multiple R-squared:  0.0197, Adjusted R-squared:  0.0129
## F-statistic:  2.9 on 4 and 575 DF,  p-value: 0.0216
```

** Solution:**
Looking at summary(), R chose `high` as reference level for `Education` and `Income`.

(c) Fit a multiple regression model with `Blood` as the response. Use the main effects of `Age`, `Education`, `Income`, and `Sex`, as well as the interaction of `Sex` with `Age` and the interaction of `Sex` and `Income`. Use a statistical test to compare this model to the additive model using a significance level of $\alpha = 0.10$. Which do you prefer?

```
hospital_int_model1 = lm(Blood ~ Age + Education + Income + Sex + Sex:Age + Sex:Income, data = hospital
anova(hospital_add_model, hospital_int_model1)
```

```
## Analysis of Variance Table
##
## Model 1: Blood ~ Age + Education + Income + Sex
## Model 2: Blood ~ Age + Education + Income + Sex + Sex:Age + Sex:Income
##   Res.Df   RSS Df Sum of Sq    F Pr(>F)
## 1    575 35694
## 2    573 35423  2       271 2.19   0.11
```

```
p_value = anova(hospital_add_model, hospital_int_model1)[2, "Pr(>F)"]
alpha = 0.10
if(p_value < alpha){
  result = "Reject H0"
}else{
  result = "Fail to Reject H0"
}
result
```

```
## [1] "Fail to Reject H0"
```

Solution:.
Based on F-test, given high p-value, it is failed to reject H0. This more complex model doesn't provide model to fit data significantly better. We prefer the addition model over this interaction model.

(d) Fit a model similar to that in **(c)**, but additionally add the interaction between `Income` and `Age` as well as a three-way interaction between `Age`, `Income`, and `Sex`. Use a statistical test to compare this model to the preferred model from **(c)** using a significance level of $\alpha = 0.10$. Which do you prefer?

```
income_age_int_model = lm(Blood ~ Age + Education + Income + Sex + Sex:Age + Sex:Income + Income:Age + 
```

```
anova(hospital_int_model1, income_age_int_model)
```

```
## Analysis of Variance Table
##
## Model 1: Blood ~ Age + Education + Income + Sex + Sex:Age + Sex:Income
## Model 2: Blood ~ Age + Education + Income + Sex + Sex:Age + Sex:Income +
##     Income:Age + Age:Income:Sex
##   Res.Df   RSS Df Sum of Sq    F Pr(>F)
## 1    573 35423
## 2    571 35166  2       257 2.08   0.13
```

```
alpha = 0.10
p_value = anova(hospital_int_model1, income_age_int_model)[2, "Pr(>F)"]
if(p_value < alpha){
  result = "Reject H0"
```

```r
}else{
  result = "Fail to Reject H0"
}

result
```

```
## [1] "Fail to Reject H0"
```

Solution: .
Based on F-test, obtained P value is larger than alpha, it is failed to reject H0. Therefore, the preferred model is model from **(c)** which is the simpler additive model.

(e) Using the model in **(d)**, give an estimate of the change in average `Blood` for a one-unit increase in `Age` for a highly educated, low income, male patient.

```r
summary(income_age_int_model)
```

```
##
## Call:
## lm(formula = Blood ~ Age + Education + Income + Sex + Sex:Age +
##     Sex:Income + Income:Age + Age:Income:Sex, data = hospital)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -13.79  -4.95  -1.79   3.23  47.11
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              14.3481     2.6074    5.50  5.6e-08 ***
## Age                      -0.0175     0.0415   -0.42    0.674
## Educationlow              0.5646     0.7546    0.75    0.455
## Incomelow                -8.3236     3.6496   -2.28    0.023 *
## Sexmale                  -5.6283     3.5532   -1.58    0.114
## Age:Sexmale               0.0424     0.0566    0.75    0.455
## Incomelow:Sexmale        10.9485     5.2999    2.07    0.039 *
## Age:Incomelow             0.1149     0.0570    2.02    0.044 *
## Age:Incomelow:Sexmale    -0.1345     0.0838   -1.60    0.109
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.85 on 571 degrees of freedom
## Multiple R-squared:  0.0342, Adjusted R-squared:  0.0207
## F-statistic: 2.53 on 8 and 571 DF,  p-value: 0.0104
```

```r
(est = as.numeric(coef(income_age_int_model)["Age"] +
coef(income_age_int_model)["Incomelow"] +
coef(income_age_int_model)["Sexmale"] +
coef(income_age_int_model)["Age:Sexmale"] +
coef(income_age_int_model)["Age:Incomelow"] +
coef(income_age_int_model)["Age:Incomelow:Sexmale"]))
```

```
## [1] -13.95
```

Solution:.
-13.9467 is an estimate of the change in average `Blood` for a one-unit increase in `Age` for a highly educated, low income, male patient.

Exercise 3 (Hospital SUPPORT Data, Stay Duration)

For this exercise, we will again use the data stored in `hospital.csv`. It contains a random sample of 580 seriously ill hospitalized patients from a famous study called "SUPPORT" (Study to Understand Prognoses Preferences Outcomes and Risks of Treatment). As the name suggests, the purpose of the study was to determine what factors affected or predicted outcomes, such as how long a patient remained in the hospital. The variables in the dataset are:

- `Days` - Days to death or hospital discharge
- `Age` - Age on day of hospital admission
- `Sex` - Female or male
- `Comorbidity` - Patient diagnosed with more than one chronic disease
- `EdYears` - Years of education
- `Education` - Education level; high or low
- `Income` - Income level; high or low
- `Charges` - Hospital charges, in dollars
- `Care` - Level of care required; high or low
- `Race` - Non-white or white
- `Pressure` - Blood pressure, in mmHg
- `Blood` - White blood cell count, in gm/dL
- `Rate` - Heart rate, in bpm

For this exercise, we will use `Blood`, `Pressure`, and `Rate` in an attempt to model `Days`. Essentially, we are attempting to model the time spent in the hospital using only health metrics measured at the hospital.

Consider the model

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_1 x_2 + \beta_5 x_1 x_3 + \beta_6 x_2 x_3 + \beta_7 x_1 x_2 x_3 + \epsilon,$$

where

- Y is `Days`
- x_1 is `Blood`
- x_2 is `Pressure`
- x_3 is `Rate`.

(a) Fit the model above. Also fit a smaller model using the provided `R` code.

```
days_add = lm(Days ~ Pressure + Blood + Rate, data = hospital)
days_int = lm(Days ~ Pressure * Blood * Rate, data = hospital)
```

Use a statistical test to compare the two models. Report the following:

- The null and alternative hypotheses in terms of the model given in the exercise description
- $H_0 : \beta_3 = 0$

- $H_1 : \beta_3 \neq 0$
- The value of the test statistic

```
days_add = lm(Days ~ Pressure + Blood + Rate, data = hospital)
days_int = lm(Days ~ Pressure * Blood * Rate, data = hospital)
(F_val = anova(days_add, days_int)$F[2])
```

```
## [1] 2.043
```

```
(P_val = anova(days_add, days_int)[2,"Pr(>F)"])
```

```
## [1] 0.08705
```

Solution: 2.0426.

- The p-value of the test.

Solution: 0.087.

- A statistical decision using a significance level of $\alpha = 0.10$

```
alpha = 0.10
if(P_val < alpha){
  result = "Reject H0"
}else{
  result = "Fail to Reject H0"
}
result
```

```
## [1] "Reject H0"
```

- Which model you prefer **Solution:** . Based on the F test and P-value obtained, we reject null hypothesis and prefer interaction model. The interaction terms in the H_1 model is significant and improves the model fit to the given data.

(b) Give an expression based on the model in the exercise description for the true change in length of hospital stay in days for a 1 bpm increase in `Rate` for a patient with a `Pressure` of 139 mmHg and a `Blood` of 10 gm/dL. Your answer should be a linear function of the βs.

```
coef(days_int)
```

```
##         (Intercept)            Pressure               Blood                Rate
##         28.73384543         -0.33393758         -0.81114256         -0.16089311
##      Pressure:Blood       Pressure:Rate          Blood:Rate Pressure:Blood:Rate
##          0.01248076          0.00368262          0.00711664         -0.00009251
```

```
coef(days_int)["Rate"] + coef(days_int)["Pressure:Rate"] * 139 + coef(days_int)["Blood:Rate"] * 10 + coe
```

```
##   Rate
## 0.2936
```

(c) Give an expression based on the additive model in part **(a)** for the true change in length of hospital stay in days for a 1 bpm increase in `Rate` for a patient with a `Pressure` of 139 mmHg and a `Blood` of 10 gm/dL. Your answer should be a linear function of the βs.

```
coef(days_add)["Rate"]
```

```
##   Rate
## 0.1337
```

Exercise 4 (t-test Is a Linear Model)

In this exercise, we will try to convince ourselves that a two-sample t-test assuming equal variance is the same as a t-test for the coefficient in front of a single two-level factor variable (dummy variable) in a linear model.

First, we set up the data frame that we will use throughout.

```
n = 30

sim_data = data.frame(
  groups = c(rep("A", n / 2), rep("B", n / 2)),
  values = rep(0, n))
str(sim_data)
```

```
## 'data.frame':    30 obs. of  2 variables:
##  $ groups: chr  "A" "A" "A" "A" ...
##  $ values: num  0 0 0 0 0 0 0 0 0 0 ...
```

We will use a total sample size of `30`, `15` for each group. The `groups` variable splits the data into two groups, `A` and `B`, which will be the grouping variable for the t-test and a factor variable in a regression. The `values` variable will store simulated data.

We will repeat the following process a number of times.

```
set.seed(20)
sim_data$values = rnorm(n, mean = 42, sd = 3.5) # simulate response data
summary(lm(values ~ groups, data = sim_data))
```

```
##
## Call:
## lm(formula = values ~ groups, data = sim_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -9.04  -1.11  -0.14   2.23   7.33
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   40.922      0.950   43.07   <2e-16 ***
## groupsB        0.029      1.344    0.02     0.98
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.68 on 28 degrees of freedom
## Multiple R-squared:  1.66e-05,   Adjusted R-squared:  -0.0357
## F-statistic: 0.000465 on 1 and 28 DF,  p-value: 0.983
```

```r
t.test(values ~ groups, data = sim_data, var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  values by groups
## t = -0.022, df = 28, p-value = 1
## alternative hypothesis: true difference in means between group A and group B is not equal to 0
## 95 percent confidence interval:
##  -2.781  2.723
## sample estimates:
## mean in group A mean in group B
##           40.92           40.95
```

We use `lm()` to test

$$H_0 : \beta_1 = 0$$

for the model

$$Y = \beta_0 + \beta_1 x_1 + \epsilon$$

where Y is the values of interest, and x_1 is a dummy variable that splits the data in two. We will let `R` take care of the dummy variable.

We use `t.test()` to test

$$H_0 : \mu_A = \mu_B$$

where μ_A is the mean for the `A` group, and μ_B is the mean for the `B` group.

The following code sets up some variables for storage.

```r
num_sims = 300
lm_t = rep(0, num_sims)
lm_p = rep(0, num_sims)
tt_t = rep(0, num_sims)
tt_p = rep(0, num_sims)
```

- `lm_t` will store the test statistic for the test $H_0 : \beta_1 = 0$.
- `lm_p` will store the p-value for the test $H_0 : \beta_1 = 0$.
- `tt_t` will store the test statistic for the test $H_0 : \mu_A = \mu_B$.

- `tt_p` will store the p-value for the test $H_0 : \mu_A = \mu_B$.

The variable `num_sims` controls how many times we will repeat this process, which we have chosen to be `300`.

(a) Set a seed equal to your birthday. (Month and day are sufficient without year.) Then write code that repeats the above process `300` times. Each time, store the appropriate values in `lm_t`, `lm_p`, `tt_t`, and `tt_p`. Specifically, each time you should use `sim_data$values = rnorm(n, mean = 42, sd = 3.5)` to update the data. The grouping will always stay the same.

```
bd = 19870503
set.seed(bd)
for(i in 1:num_sims){
  sim_data$values = rnorm(n, mean = 42, sd = 3.5)
  lm_t[i] = summary(lm(values ~ groups, data = sim_data))$coef[2, "t value"]
  lm_p[i] = summary(lm(values ~ groups, data = sim_data))$coef[2, "Pr(>|t|)"]
  tt_t[i] = t.test(values ~ groups, data = sim_data, var.equal = TRUE)$statistic
  tt_p[i] = t.test(values ~ groups, data = sim_data, var.equal = TRUE)$p.value
}
```

(b) Report the value obtained by running `mean(lm_t == tt_t)`, which tells us what proportion of the test statistics is equal. The result may be extremely surprising!

```
mean(lm_t == tt_t)
```

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

Solution: Surprisingly the result is 0.

(c) Report the value obtained by running `mean(lm_p == tt_p)`, which tells us what proportion of the p-values is equal. The result may be extremely surprising!

```
mean(lm_p == tt_p)
```

```
## [1] 0.02
```

Solution: Surprisingly the result is 0.02

(d) If you have done everything correctly so far, your answers to the last two parts won't indicate the equivalence we want to show! What the heck is going on here? The first issue is one of using a computer to do calculations. When a computer checks for equality, it demands **equality**; nothing can be different. However, when a computer performs calculations, it can only do so with a certain level of precision. So, if we calculate two quantities we know to be analytically equal, they can differ numerically. Instead of `mean(lm_p == tt_p)` run `all.equal(lm_p, tt_p)`. This will perform a similar calculation, but with a very small error tolerance for each equality. What is the result of running this code? What does it mean?

```
all.equal(lm_p, tt_p)
```

```
## [1] TRUE
```

Solution: The result of all.equal() True, suggests that with the values are approximately equal within the error tolerance.

(e) Your answer in **(d)** should now make much more sense. Then what is going on with the test statistics? Look at the values stored in `lm_t` and `tt_t`. What do you notice? Is there a relationship between the two? Can you explain why this is happening?

```r
all.equal(tt_t, lm_t)
```

```
## [1] "Mean relative difference: 2"
```

```r
head(tt_t)
head(lm_t)
```

```r
all.equal(abs(tt_t), abs(lm_t))
```

```
## [1] TRUE
```

Solution:.
tt_t and lm_t values are same in magnitude and reversed signs. So absolute values are TRUE for all.equal()

We can see that from the summary(lm(values ~ groups, data = sim_data)), R has chosen Group A as the reference group per alphabetical order. And in terms of calculation, subtraction happens from Estimate of group A, mean(groupA)-mean(groupB) In t.test() the obtained t value is calculated from difference of the sample means of the two groups by subtracting mean(groupB)-mean(groupA) which in reverse order compared to in lm(). Thus the opposite sign between the two values obtained.