Week 7 - Homework

STAT 420, Summer 2018, Unger

- Directions
    - Exercise 1 (EPA Emissions Data)
    - Exercise 2 (Hospital SUPPORT Data, White Blood Cells)
    - Exercise 3 (Hospital SUPPORT Data, Stay Duration)
    - Exercise 4 (t-test Is a Linear Model)

Directions

- Be sure to remove this section if you use this `.Rmd` file as a template.
- You may leave the questions in your final document.

Exercise 1 (EPA Emissions Data)

For this exercise, we will use the data stored in `epa2015.csv` (epa2015.csv). It contains detailed descriptions of 4,411 vehicles manufactured in 2015 that were used for fuel economy testing as performed by the Environment Protection Agency (https://www3.epa.gov/otaq/tcldata.htm). 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.

Load the epa2015 dataset

```
epa2015 = read.csv("epa2015.csv")
str(epa2015)
```

```
## 'data.frame':    4411 obs. of  16 variables:
##  $ Make     : Factor w/ 30 levels "aston martin",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Model    : Factor w/ 635 levels "1500 2WD","1500 4X4",..: 189 189 479 479 57
9 579 582 582 583 583 ...
##  $ ID       : Factor w/ 872 levels "08-UF2H","0C00007",..: 82 82 180 180 149 14
9 136 136 148 148 ...
##  $ disp     : num  5.9 5.9 6 6 6 6 4.7 4.7 4.7 4.7 ...
##  $ type     : Factor w/ 3 levels "Both","Car","Truck": 2 2 2 2 2 2 2 2 2 2 ...
##  $ horse    : int  510 510 552 552 565 565 420 420 430 430 ...
##  $ cyl      : int  12 12 12 12 12 12 8 8 8 8 ...
##  $ lockup   : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 1 1 2 2 ...
##  $ drive    : Factor w/ 5 levels "4","A","F","P",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ weight   : int  4500 4500 4750 4750 4250 4250 4000 4000 4000 4000 ...
##  $ axleratio: num  3.46 3.46 2.73 2.73 3.73 3.73 3.91 3.91 4.18 4.18 ...
##  $ nvratio  : num  31 31 22.4 22.4 33.6 33.6 38.6 38.6 36.2 36.2 ...
##  $ THC      : num  0.0251 0.0022 0.0269 0.0008 0.0248 ...
##  $ CO       : num  0.12 0.0118 0.5 0.06 0.61 ...
##  $ CO2      : num  550 344 512 297 603 ...
##  $ mpg      : num  16.1 25.8 17.3 29.9 14.8 25.3 16 26.5 16.7 28.8 ...
```

Verify “type” ia a factor variable

```
is.factor(epa2015$type)
```

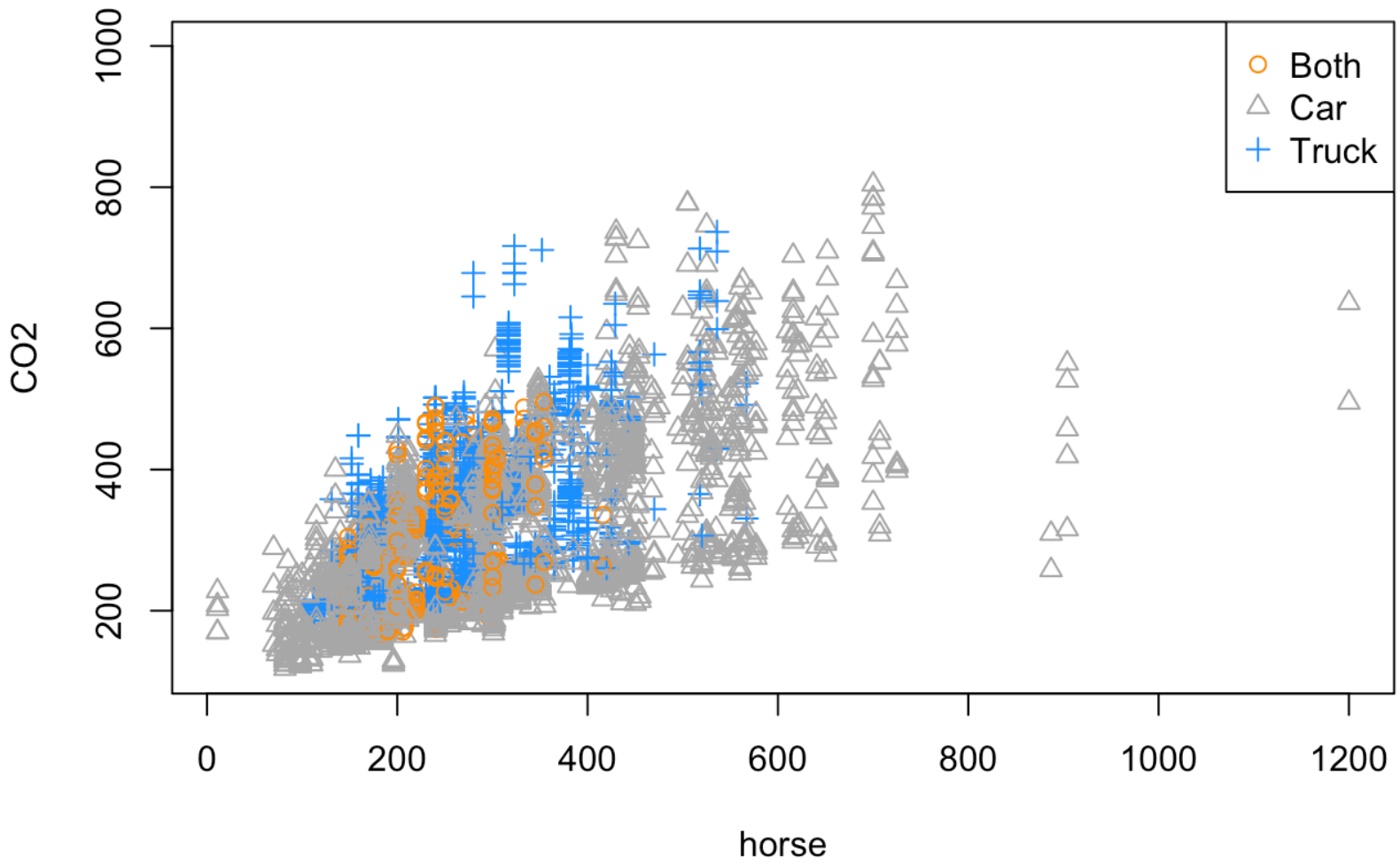
```
## [1] TRUE
```

The Type of the vehicle is a **factor variable**

(a) Do the following:

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

  ```
  plot_colors = c("Darkorange","Darkgrey","Dodgerblue")
  plot(CO2~horse,data=epa2015,col=plot_colors[type],pch=as.numeric(type))
  legend("topright",legend=c("Both","Car","Truck"),col = plot_colors,pch=c(1,2,
  3))
  ```



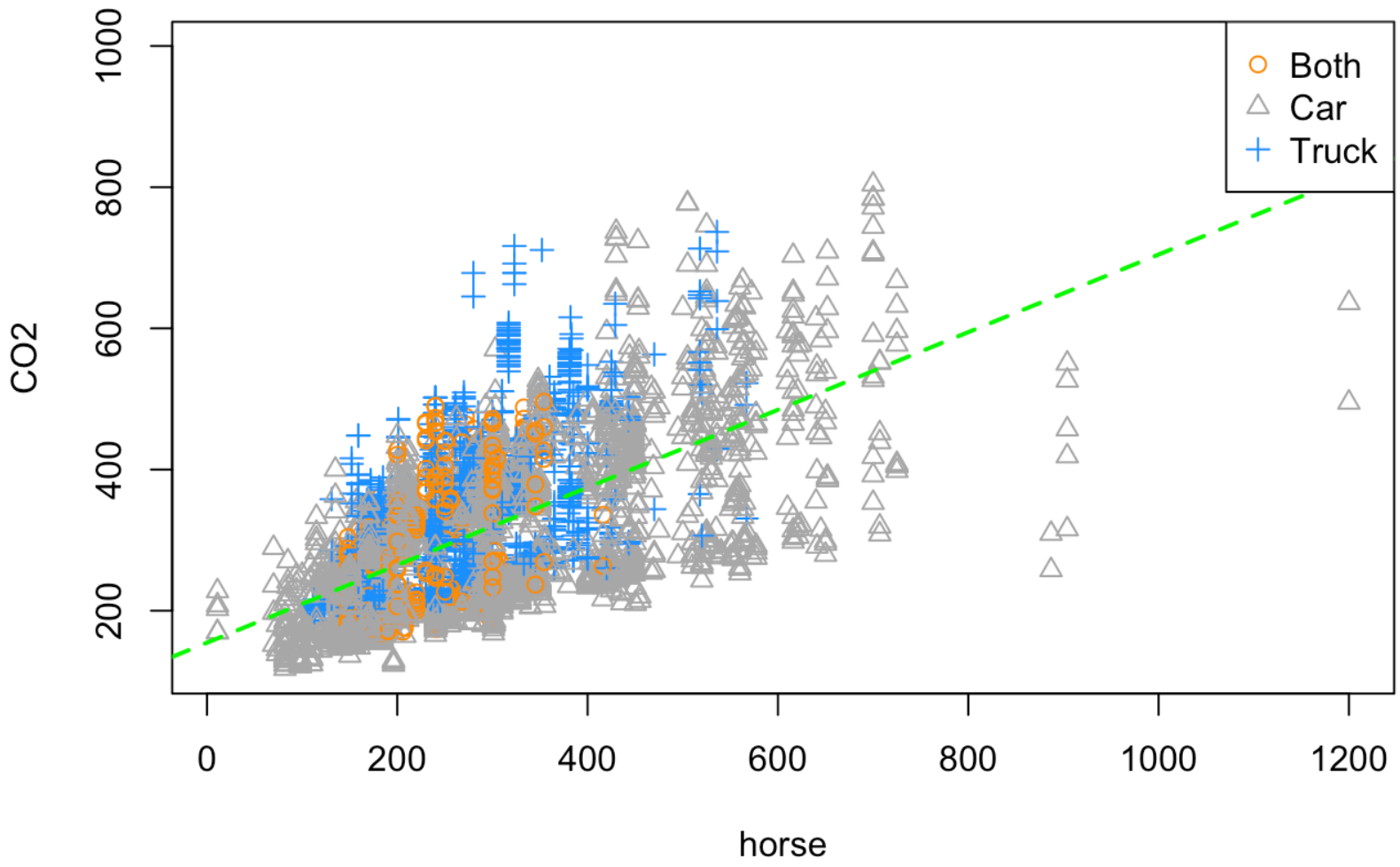
- Fit a simple linear regression model with `CO2` as the response and only `horse` as the predictor.

```
CO2_slr = lm(CO2~horse,data = epa2015)
CO2_slr
```

```
##
## Call:
## lm(formula = CO2 ~ horse, data = epa2015)
##
## Coefficients:
## (Intercept)        horse
##      154.72         0.55
```

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

```
plot(CO2~horse,data=epa2015,col=plot_colors[type],pch=as.numeric(type))
abline(CO2_slr,lwd=2,col="green",lty=2)
legend("topright",legend=c("Both","Car","Truck"),col = plot_colors,pch=c(1,2,
3))
```



Comment on how well this line models the data

The regression line in `green` color shows, which seems closesly fitting to the `Both` and `Car` dataset, however not doing well on the `Truck` dataset, where it seems `underfitting` for the `Truck` dataset

- 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` .

```
summary(CO2_slr)$coefficient[2,1]
```

```
## [1] 0.5499
```

The estimate for the average change in `CO2` for a one foot-pound per second increase in `horse` for a vehicle of type `car` is : **0.5499**

- 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.)

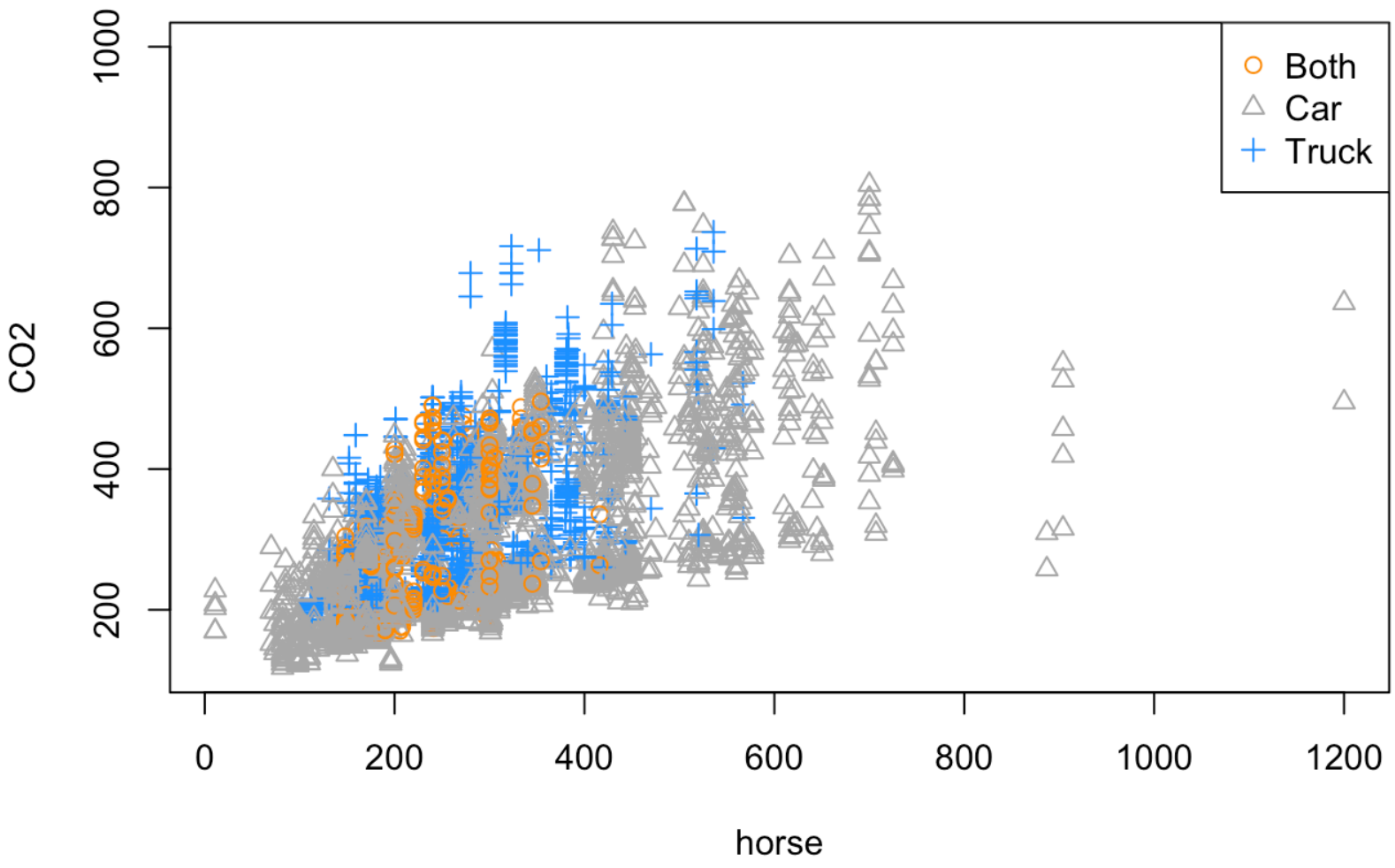
```
predict(CO2_slr,interval="prediction",level=0.90,newdata=data.frame(horse=148
))
```

```
##     fit   lwr   upr
## 1 236.1 89.46 382.7
```

(b) Do the following:

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

```
plot_colors = c("Darkorange","Darkgrey","Dodgerblue")
plot(CO2~horse,data=epa2015,col=plot_colors[type],pch=as.numeric(type))
legend("topright",legend=c("Both","Car","Truck"),col = plot_colors,pch=c(1,2,
3))
```



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

```
CO2_add = lm(CO2~horse+type,data = epa2015)
CO2_add
```

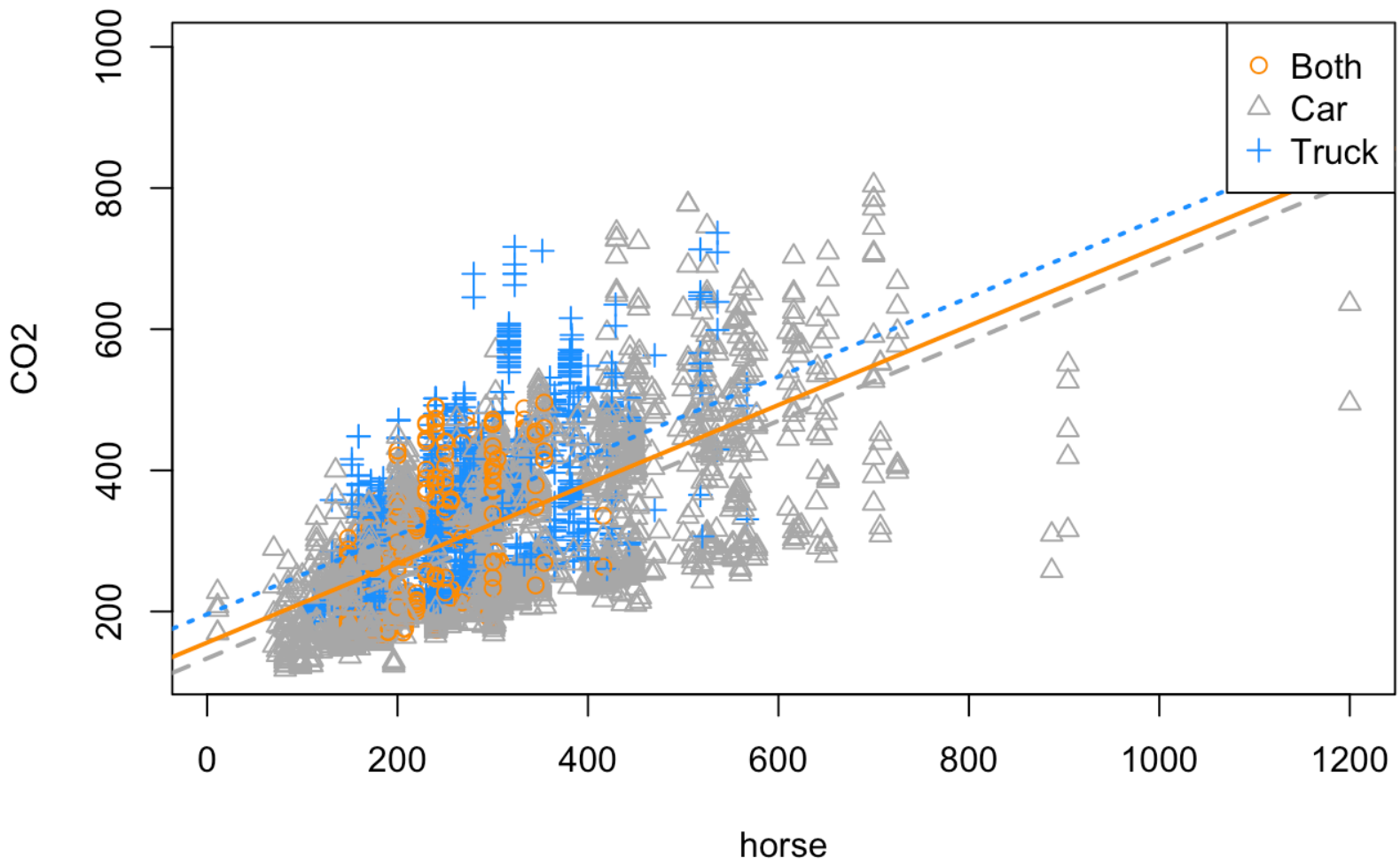
```
##
## Call:
## lm(formula = CO2 ~ horse + type, data = epa2015)
##
## Coefficients:
## (Intercept)        horse      typeCar    typeTruck
##     155.982        0.561      -22.425       40.074
```

- 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.

```
intercept_both = summary(CO2_add)$coefficient[1,1]
intercept_car = summary(CO2_add)$coefficient[1,1] + summary(CO2_add)$coefficient[3,1]
intercept_truck = summary(CO2_add)$coefficient[1,1] + summary(CO2_add)$coefficient[4,1]

slope_all_type = summary(CO2_add)$coefficient[2,1]

plot(CO2~horse,data = epa2015,col = plot_colors[type],pch = as.numeric(type))
abline(intercept_both,slope_all_type,lwd = 2,col = plot_colors[1],lty = 1)
abline(intercept_car,slope_all_type,lwd = 2,col = plot_colors[2],lty = 2)
abline(intercept_truck,slope_all_type,lwd = 2,col = plot_colors[3],lty = 3)
legend("topright",legend = c("Both","Car","Truck"),col = plot_colors,pch = c(1,2,3))
```



Comment on how well this line models the data

There are 3 regression line, with 3 different color for 3 vehicle type. The regression line in `grey` and `orange` color shows close enough to the `Car` and `Both` dataset respectively. However the `blue` line seems a bit underfitting for the `Truck` dataset.

- 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`.

```
summary(CO2_add)$coefficient[2,1]
```

```
## [1] 0.5611
```

The estimate for the average change in `CO2` for a one foot-pound per seconds increase in `horse` for a vehicle of type `car` is : **0.5611**

- 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`.

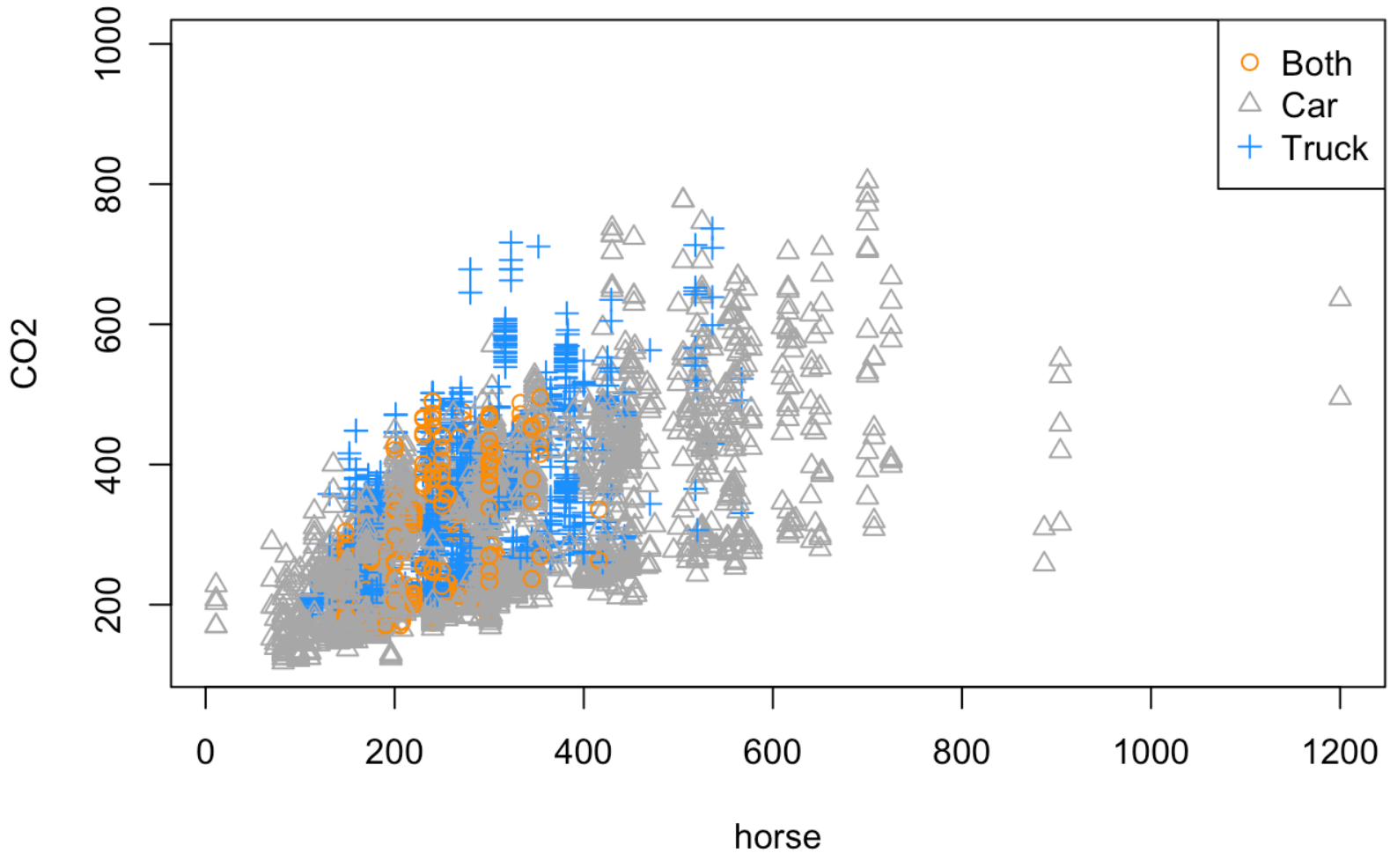
```
predict(CO2_add,interval="prediction",level=0.90,newdata=data.frame(horse=148
,type='Both'))
```

```
##   fit   lwr   upr
## 1 239 98.59 379.5
```

(c) Do the following:

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

```
plot(CO2~horse,data=epa2015,col=plot_colors[type],pch=as.numeric(type))
legend("topright",legend=c("Both","Car","Truck"),col = plot_colors,pch=c(1,2,
3))
```



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

```
CO2_int = lm(CO2~horse*type,data = epa2015)
CO2_int
```

```
##
## Call:
## lm(formula = CO2 ~ horse * type, data = epa2015)
##
## Coefficients:
##     (Intercept)            horse          typeCar        typeTruck
##        149.8971           0.5861         -11.1796           7.6640
##   horse:typeCar  horse:typeTruck
##         -0.0429           0.1153
```

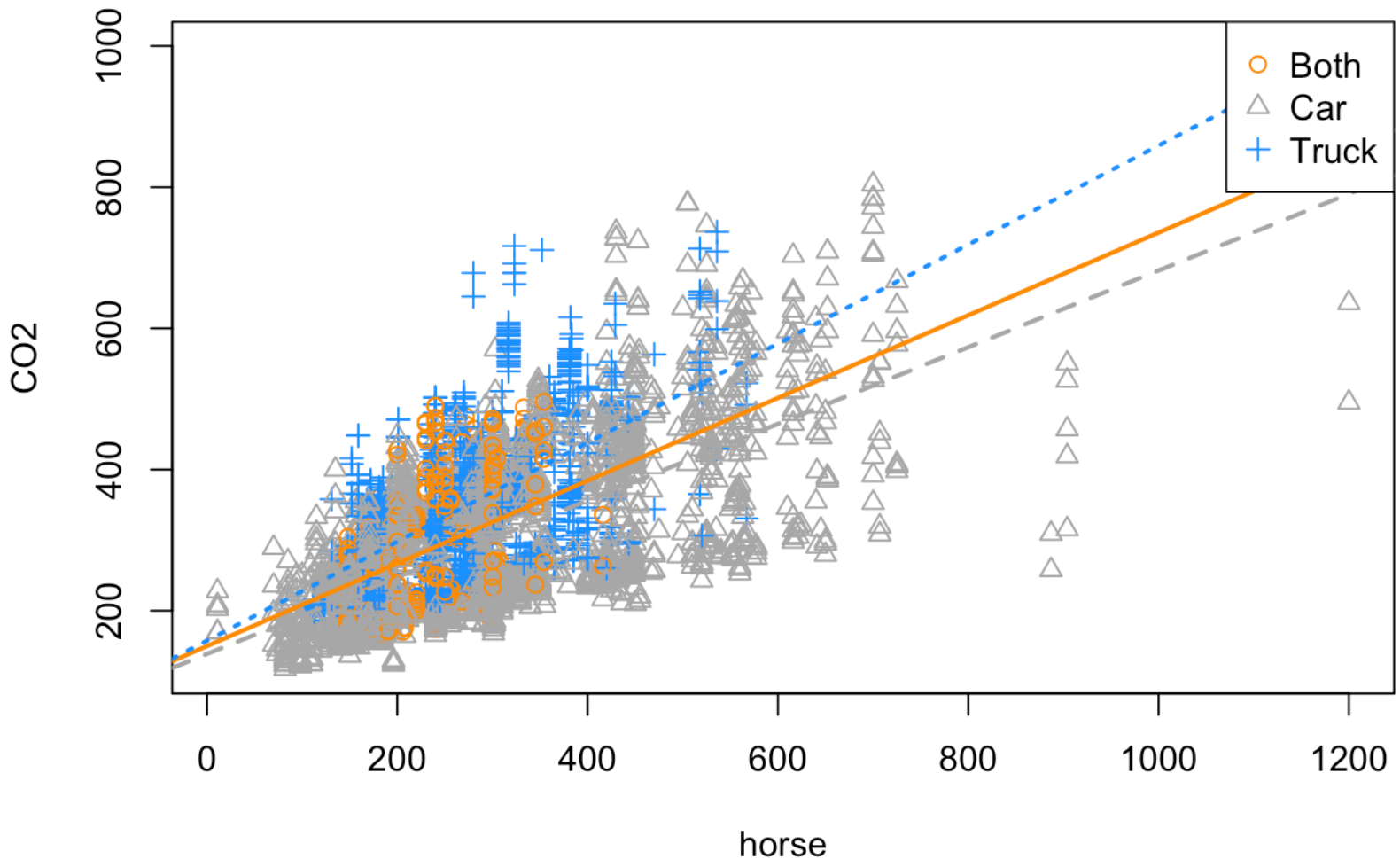
- Add the fitted regression “lines” to the scatterplot with the same colors as their respective points (one line for each vehicle type). Comment on how well this line models the data.

```
intercept_both  = summary(CO2_int)$coefficient[1,1]
intercept_car   = summary(CO2_int)$coefficient[1,1] + summary(CO2_int)$coeffi
cient[3,1]
intercept_truck = summary(CO2_int)$coefficient[1,1] + summary(CO2_int)$coeffi
cient[4,1]

slope_both  = summary(CO2_int)$coefficient[2,1]
slope_car   = summary(CO2_int)$coefficient[2,1] + summary(CO2_int)$coefficien
t[5,1]
slope_truck = summary(CO2_int)$coefficient[2,1] + summary(CO2_int)$coefficien
t[6,1]

plot(CO2~horse,data = epa2015,col = plot_colors[type],pch = as.numeric(type))
abline(intercept_both,slope_both,lwd = 2,col = plot_colors[1],lty = 1)
abline(intercept_car,slope_car,lwd = 2,col = plot_colors[2],lty = 2)
abline(intercept_truck,slope_truck,lwd = 2,col = plot_colors[3],lty = 3)

legend("topright",legend = c("Both","Car","Truck"),col = plot_colors,pch = c(
1,2,3))
```



Comment on how well this line models the data

There are 3 regression line, with 3 different color for 3 vehicle type. The regression line in `grey` and `orange` color shows close enough to the `Car` and `Both` dataset respectively. Also the `blue` line fitted well to the `Truck` dataset, which seems `underfitting` in the `additive` model (**previously**)

- 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` .

```
beta_1 = summary(CO2_int)$coefficient[2,1]
gamma_1 = summary(CO2_int)$coefficient[5,1]
beta_1 + gamma_1
```

```
## [1] 0.5432
```

The estimate for the average change in `CO2` for a one foot-pound per second increase in `horse` for a vehicle of type `car` is **0.5432**

- 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` .

```
predict(CO2_int,newdata = data.frame(horse=148,type="Both"),interval = "predi
ction", level = 0.90)
```

```
##     fit   lwr   upr
## 1 236.6 96.21 377.1
```

(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?

```
anova(CO2_add,CO2_int)
```

```
## 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   4407 32054899
## 2   4405 31894278  2    160621 11.1 0.000016 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(CO2_add,CO2_int)$'Pr(>F)'[2]
```

```
## [1] 0.00001567
```

The Anova F test gives the P value as **1.566510^{-5}** < 0.10 (α), where the P value (**1.566510^{-5}**) is very smaller than of 0.10 (α), hence **we reject the Additive Model**. Based on the test and significance level we will preferred the **Interaction Model**.

Exercise 2 (Hospital SUPPORT Data, White Blood Cells)

For this exercise, we will use the data stored in `hospital.csv` (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`?

Load the dataset

```
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        : Factor w/ 2 levels "female","male": 1 1 2 2 2 2 1 1 2 2 ...
##  $ Comorbidity: Factor w/ 2 levels "no","yes": 1 1 2 2 2 2 2 1 2 2 ...
##  $ EdYears    : int  11 22 18 16 8 12 12 13 16 30 ...
##  $ Education  : Factor w/ 2 levels "high","low": 2 1 1 1 2 2 2 1 1 1 ...
##  $ Income     : Factor w/ 2 levels "high","low": 1 1 1 1 1 1 2 1 1 1 ...
##  $ Charges    : num  9914 283303 320843 4173 13414 ...
##  $ Care       : Factor w/ 2 levels "high","low": 2 1 1 2 2 2 2 1 2 2 ...
##  $ Race       : Factor w/ 2 levels "non-white","white": 1 2 2 2 2 2 2 2 2 2 ...
##  $ 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 ...
```

Verify the Education is a Factor variable

```
is.factor(hospital$Education)
```

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

Verify the Income is a Factor variable

```
is.factor(hospital$Income)
```

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

Verify the Sex is a Factor variable

```
is.factor(hospital$Sex)
```

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

Levels of Education variable

```
levels(hospital$Education)
```

```
## [1] "high" "low"
```

Levels of Income variable

```
levels(hospital$Income)
```

```
## [1] "high" "low"
```

Levels of Sex variable

```
levels(hospital$Sex)
```

```
## [1] "female" "male"
```

(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`?

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

```
##
## Call:
## lm(formula = Blood ~ Age + Education + Income + Sex, data = hospital)
##
## Coefficients:
##  (Intercept)           Age  Educationlow     Incomelow       Sexmale
##      10.8662        0.0283        0.5967        0.1867       -1.8714
```

Below is what R choose as the Reference level

- Reference level for `Education` : **high**
- Reference level for `Income` : **high**
- Reference level for `Sex` : **female**

(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?

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

```
##
## Call:
## lm(formula = Blood ~ Age + Education + Income + Sex + Sex:Age +
##     Sex:Income, data = hospital)
##
## Coefficients:
##       (Intercept)                Age       Educationlow           Incomelow
##           10.6821             0.0435             0.5641            -1.2384
##           Sexmale        Age:Sexmale  Incomelow:Sexmale
##           -1.4898            -0.0263             2.6718
```

```
anova(blood_add,blood_int)
```

```
## 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 from the Additive and Interaction Model

```
anova(blood_add,blood_int)$'Pr(>F)'[2]
```

```
## [1] 0.1128
```

The P values from anova between the Additive model and Interaction model is **0.1128** > 0.10 (α), hence we **failed to reject the additive hypothesis**, hence we prefer the **Additive 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?

```
blood_int_3 = lm(Blood~Age+Education+Income+Sex+Sex:Age+Sex:Income+Age:Income+Sex:
Age:Income,data=hospital)
blood_int_3
```

```
##
## Call:
## lm(formula = Blood ~ Age + Education + Income + Sex + Sex:Age +
##     Sex:Income + Age:Income + Sex:Age:Income, data = hospital)
##
## Coefficients:
##           (Intercept)                    Age           Educationlow
##               14.3481                -0.0175                 0.5646
##             Incomelow                Sexmale            Age:Sexmale
##                -8.3236                -5.6283                 0.0424
##     Incomelow:Sexmale          Age:Incomelow  Age:Incomelow:Sexmale
##                10.9485                 0.1149                -0.1345
```

```
anova(blood_int,blood_int_3)
```

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

```
anova(blood_int,blood_int_3)$'Pr(>F)'[2]
```

```
## [1] 0.1254
```

The F Test Statistics shows, the P value is 0.1254 > 0.10 (α), hence we **failed to reject the model in (c)**, so we prefer the **model in (c)**

(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.

```
beta_1 = summary(blood_int_3)$coefficient[2,1]
gamma_2 = summary(blood_int_3)$coefficient[6,1]
gamma_4 = summary(blood_int_3)$coefficient[8,1]
gamma_5 = summary(blood_int_3)$coefficient[9,1]
(beta_1 + gamma_2 + gamma_4 + gamma_5)
```

```
## [1] 0.0053
```

Estimate of the change in average `Blood` for a one-unit increase in `Age` for a highly educated, low income, male patient is: **0.0053**

Exercise 3 (Hospital SUPPORT Data, Stay Duration)

For this exercise, we will again use the data stored in `hospital.csv` (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

$$H_{null} : \beta_5 = \beta_6 = \beta_7 = 0$$
$$H_{alternate} : \text{Atleast one of the parameter is not zero}, \beta_5 \neq 0, \beta_6 \neq 0, \beta_7 \neq 0$$

- The value of the test statistic

```
anova(days_add,days_int)$'F'[2]
```

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

The value of the test statistic (F Test) is : **2.0426**

- The p-value of the test

```
anova(days_add,days_int)$'Pr(>F)'[2]
```

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

The p-value of the test : **0.087**

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

  Since the **P value** from the F Test comes out as **0.087** < 0.1 (α)**, hence we** reject the null hypothesis**.

- Which model you prefer

  Since we **reject the null hypothesis**, we prefer the **interaction model**

(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.

The expression Is: $\beta_3 + 10\beta_5 + 139\beta_6 + 1390\beta_7$

(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.

The expression Is: β_3

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: Factor w/ 2 levels "A","B": 1 1 1 1 1 1 1 1 1 1 ...
##  $ 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(420)
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
## -9.604 -1.182 -0.332  2.010  6.536
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   41.283      0.824   50.09   <2e-16 ***
## groupsB        0.831      1.166    0.71     0.48
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.19 on 28 degrees of freedom
## Multiple R-squared:  0.0178, Adjusted R-squared:  -0.0172
## F-statistic: 0.508 on 1 and 28 DF,  p-value: 0.482
```

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

```
##
##  Two Sample t-test
##
## data:  values by groups
## t = -0.71, df = 28, p-value = 0.5
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.218  1.557
## sample estimates:
## mean in group A mean in group B
##           41.28           42.11
```

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.

```
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. 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.

```
set.seed(19770411)
for(i in 1:num_sims){
  sim_data$values = rnorm(n, mean = 42, sd = 3.5) # simulate response data
  lm_t[i] = summary(lm(values ~ groups, data = sim_data))$coefficient[2,3]
  lm_p[i] = summary(lm(values ~ groups, data = sim_data))$coefficient[2,4]
  tt_t[i] = t.test(values ~ groups, data = sim_data, var.equal = TRUE)$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!

The value coming from **`mean(lm_t == tt_t)`** 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!

The value coming from **`mean(lm_p == tt_p)`** is : **0.0267**

Yes, this seems suprising, even though the value looks matching, however not matching to the precission level

(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?

Value of **`all.equal(lm_p, tt_p)`** is: **TRUE**

The result from `all.equal` between lm_p and tt_p `TRUE` . This means that, all the p values comes from 2 sample t test is matching with the single two level factor variable in a linear model, however with a very small 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?

```
head(lm_t,8)
```

```
## [1]  0.89667 -2.02530  0.93800  0.89394 -0.08407 -1.49559 -0.78186 -0.70440
```

```
head(tt_t,8)
```

```
## [1] -0.89667  2.02530 -0.93800 -0.89394  0.08407  1.49559  0.78186  0.70440
```

What do we notice:

By looking at the test statistics values for both, it seems that the absolute values are matching, however sign is reverse

Is there a relationship between the two test statistics:

The relationship between the two test statistics is just the reverse of the sign, where as the data is equal

Why this is happening:

The difference in the test statistics is because of the sign change, if we compare with the absoulute value, all the t value will be matched (with a very small error tolerance), and while matching with `for.all` its coming as TRUE. The t value from the two-sample t-test have a differnt sign as compared to the coefficient of a single two-level factor variable, could be because of the mean of the one group (dummy variable) would be more/less than the mean of the other group (dummy variable). so, in order to check the value without the sign, we need to compare the t test statistics for the two with `abs` function.