

Week 7 - Homework

STAT 420, Summer 2018, Unger

Exercise 1 (EPA Emissions Data)

For this exercise, we will use the data stored in [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](#). 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.

```
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",...: 189 189 479 479 579 579 582 582 583 583 ...
## $ ID       : Factor w/ 872 levels "08-UF2H","0C00007",...: 82 82 180 180 149 149 136 136 148 148 ...
## $ 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 ...
```

```
is.factor(eps2015$type)
```

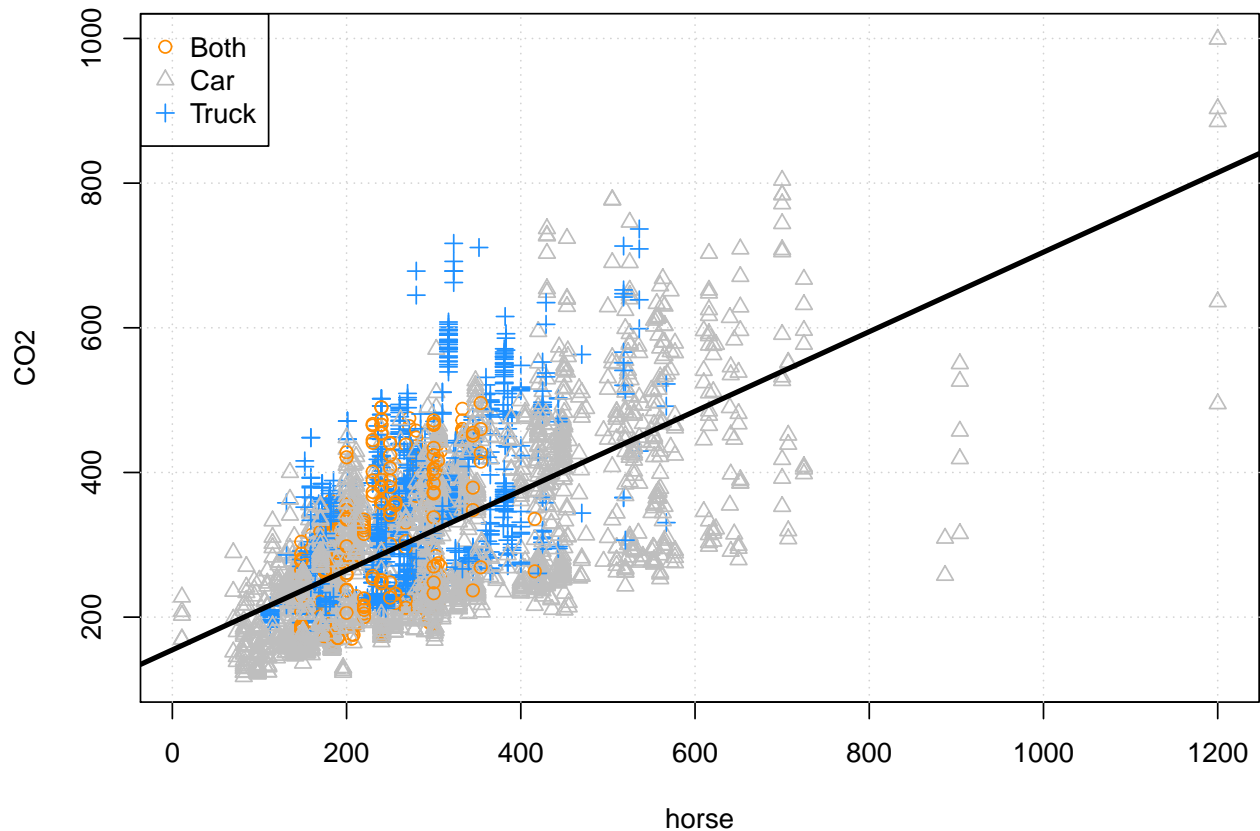
```
## [1] TRUE
```

(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.)

Solution:

```
co2_slr = lm(CO2 ~ horse, data = eps2015)
type_colors = ifelse(eps2015$type == "Both", "Darkorange",
                     ifelse(eps2015$type == "Car", "Grey", "Dodgerblue"))
plot(CO2 ~ horse, data = eps2015, col = type_colors, pch = as.numeric(type))
grid()
abline(co2_slr, lwd = 3)
legend("topleft", c("Both", "Car", "Truck"),
      col = c("Darkorange", "Grey", "Dodgerblue"),
      pch = c(1, 2, 3))
```



```
slr_coef = summary(co2_slr)$coef[, 1]

int_both    = slr_coef[1]
int_car     = slr_coef[1]
int_truck   = slr_coef[1]

slope_both  = slr_coef[2]
slope_car   = slr_coef[2]
slope_truck = slr_coef[2]

impreza_wagon = data.frame(horse = 148, type = "Both")
predict(co2_slr, impreza_wagon, interval = "prediction", level = 0.90)
```

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

- This model does not seem to fit the data well. Based on the plot, we can see the line is consistently underestimating CO2 for Truck type vehicles.
- Using this model, an estimate for the average change in CO2 for a one foot-pound per second increase in horse for a vehicle of type car is given by 0.5499.
- The 90% prediction interval for the Subaru Impreza Wagon is: (89.4593, 382.7466).

(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.

Solution:

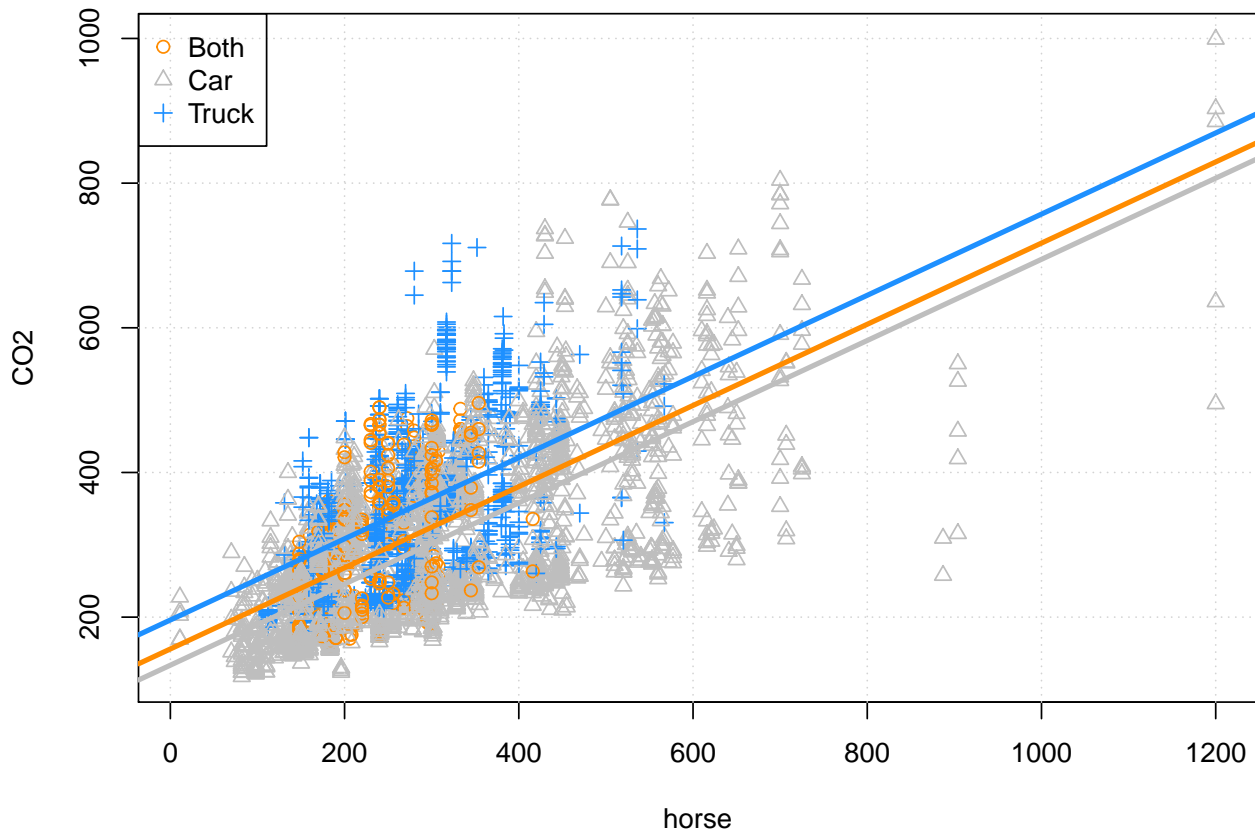
```
co2_add = lm(CO2 ~ horse + type, data = epa2015)
type_colors = ifelse(epa2015$type == "Both", "Darkorange",
                     ifelse(epa2015$type == "Car", "Grey", "Dodgerblue"))
plot(CO2 ~ horse, data = epa2015, col = type_colors, pch = as.numeric(type))
grid()
legend("topleft", c("Both", "Car", "Truck"),
      col = c("Darkorange", "Grey", "Dodgerblue"),
      pch = c(1, 2, 3))

add_coef = summary(co2_add)$coef[,1]

int_both    = add_coef[1]
int_car     = add_coef[1] + add_coef[3]
int_truck   = add_coef[1] + add_coef[4]

slope_both  = add_coef[2]
slope_car   = add_coef[2]
slope_truck = add_coef[2]

abline(int_both, slope_both, lwd = 3, col = "Darkorange")
abline(int_car, slope_car, lwd = 3, col = "Grey")
abline(int_truck, slope_truck, lwd = 3, col = "Dodgerblue")
```



```
predict(co2_add, impreza_wagon, interval = "prediction", level = 0.90)
```

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

- This model seems to fit slightly better than the previous, but it still has some issues. For example, the slopes for Truck and Both may need to be steeper.
- Using this model, an estimate for the average change in C02 for a one foot-pound per second increase in horse for a vehicle of type car is given by 0.5611.
- The 90% prediction interval for the Subaru Impreza Wagon is: (98.5935, 379.4566).

(c) Do the following:

- Make a scatterplot of C02 versus horse. Use a different color point for each vehicle type.
- Fit an interaction multiple regression model with C02 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 C02 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 C02 of a Subaru Impreza Wagon, which is a vehicle with 148 horsepower and is considered type Both.

Solution:

```
co2_int = lm(C02 ~ horse * type, data = epa2015)
type_colors = ifelse(epa2015$type == "Both", "Darkorange",
                     ifelse(epa2015$type == "Car", "Grey", "Dodgerblue"))
plot(C02 ~ horse, data = epa2015, col = type_colors, pch = as.numeric(type))
```

```

grid()
legend("topleft", c("Both", "Car", "Truck"),
      col = c("Darkorange", "Grey", "Dodgerblue"),
      pch = c(1, 2, 3))

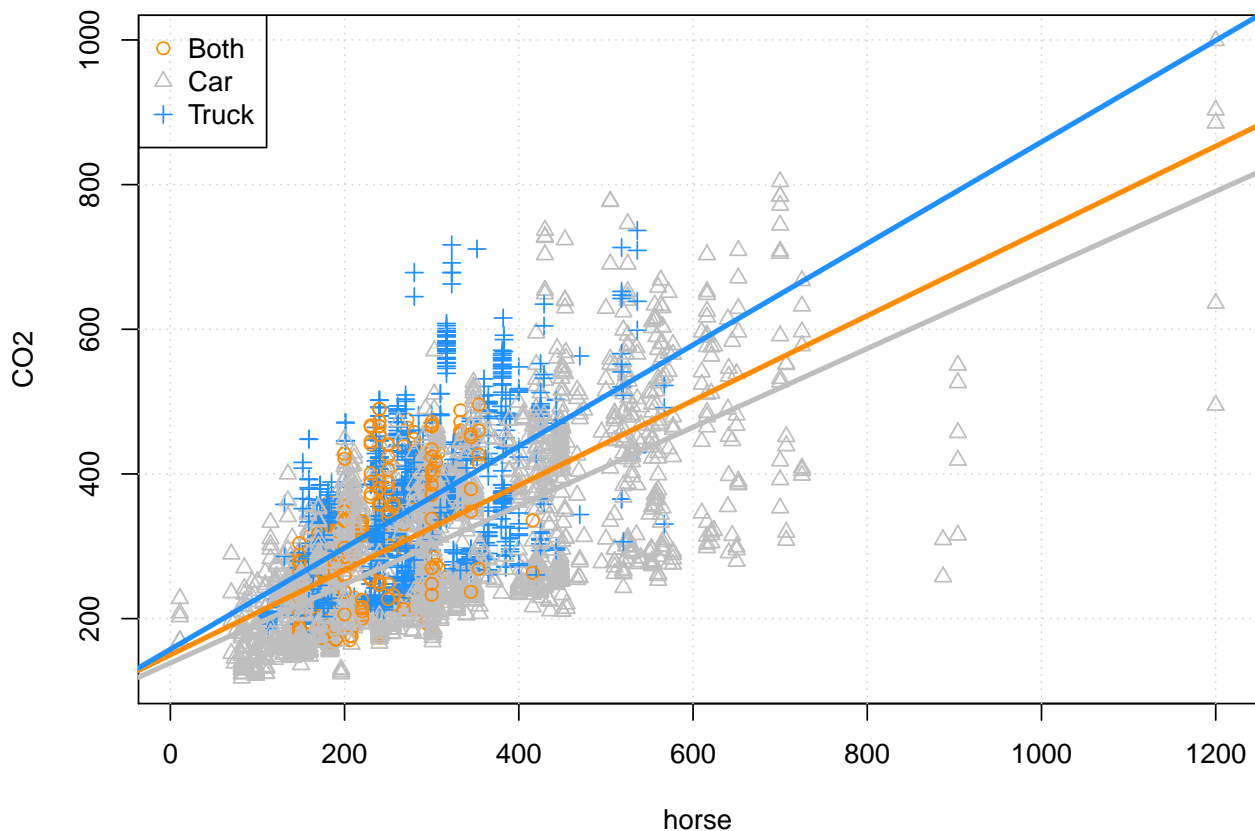
int_coef = summary(co2_int)$coef[,1]

int_both    = int_coef[1]
int_car     = int_coef[1] + int_coef[3]
int_truck   = int_coef[1] + int_coef[4]

slope_both  = int_coef[2]
slope_car   = int_coef[2] + int_coef[5]
slope_truck = int_coef[2] + int_coef[6]

abline(int_both, slope_both, lwd = 3, col = "Darkorange")
abline(int_car, slope_car, lwd = 3, col = "Grey")
abline(int_truck, slope_truck, lwd = 3, col = "Dodgerblue")

```



```

predict(co2_int, impreza_wagon, interval = "prediction", level = 0.90)

```

```

##      fit   lwr   upr
## 1 236.6 96.21 377.1

```

- Of the models we have seen, this one seems to fit the best. We see that the slope for Truck is now greater than the slope for Car, which seems to match the plot.
- Using this model, an estimate for the average change in CO2 for a one foot-pound per second increase in horse for a vehicle of type car is given by 0.5432.

- The 90% prediction interval for the Subaru Impreza Wagon is: (96.205, 377.0629).

(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?

Solution:

```
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
```

Here we see an extremely small p-value, so we reject the null hypothesis. Thus, we prefer the larger, interaction model.

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**?

Solution:

```
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 ...
```

```
all(is.factor(hospital$Education),
    is.factor(hospital$Income),
    is.factor(hospital$Sex))
```

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

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

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

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

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

```
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?

Solution:

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

```
## (Intercept)      Age Educationlow    Incomelow      Sexmale
##   10.86624    0.02828    0.59667    0.18667    -1.87144
```

- The reference level for Education is "high"
- The reference level for Income is "high"
- The reference level for Sex is "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?

Solution:

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

```
## Analysis of Variance Table
```

```
##
```

```
## Model 1: Blood ~ Age + Education + Income + Sex
```



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

Based on the p-value of 0.1128 we reject the null. We prefer the smaller model without interaction terms.

(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?

Solution:

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

```
## Analysis of Variance Table
##
## Model 1: Blood ~ Age + Education + Income + Sex
## Model 2: Blood ~ Income * Sex * Age + Education
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1     575 35694
## 2     571 35166   4      528 2.14  0.074 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on the p-value of 0.0744 we reject the null. We prefer the largest model fit in part (d).

(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.

Solution:

```
coef(blood_big)

##           (Intercept)           Incomelow           Sexmale
##           14.34808           -8.32364           -5.62832
##              Age      Educationlow      Incomelow:Sexmale
##          -0.01748           0.56459           10.94854
##      Incomelow:Age      Sexmale:Age      Incomelow:Sexmale:Age
##           0.11488           0.04237           -0.13447

sum(coef(blood_big)[c("Age", "Incomelow:Age", "Sexmale:Age", "Incomelow:Sexmale:Age")])

## [1] 0.0053
```

Based on this model, an estimate for the change in average Blood for a one-unit increase in **Age** for a well-educated, low income, male patient is given by 0.0053.

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)
```

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
- The value of the test statistic
- The p-value of the test
- A statistical decision using a significance level of $\alpha = 0.10$
- Which model you prefer

Solution:

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

## Analysis of Variance Table
##
## Model 1: Days ~ Pressure + Blood + Rate
## Model 2: Days ~ Pressure * Blood * Rate
##   Res.Df    RSS Df Sum of Sq   F Pr(>F)
## 1     576 279493
## 2     572 275557   4     3936 2.04 0.087 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Null: $\beta_4 = \beta_5 = \beta_6 = \beta_7 = 0$
- Alternative: At least one of β_4 , β_5 , β_6 , and β_7 is not 0

- Test Statistic: $F = 2.0426$
- P-Value: 0.087
- Decision: Reject H_0 .
- Preference: The larger model, which includes all possible interactions

(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 β s.

Solution:

First, rearrange the model:

$$\begin{aligned} Y &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_1 x_2 + \beta_5 x_1 x_3 + \beta_6 x_2 x_3 + \beta_7 x_1 x_2 x_3 + \epsilon \\ &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_4 x_1 x_2 + (\beta_3 + \beta_5 x_1 + \beta_6 x_2 + \beta_7 x_1 x_2) x_3 + \epsilon \end{aligned}$$

Then choose the correct coefficient.

$$(\beta_3 + \beta_5 x_1 + \beta_6 x_2 + \beta_7 x_1 x_2)$$

Lastly, plug in the requested values.

$$(\beta_3 + \beta_5 \times 139 + \beta_6 \times 10 + \beta_7 \times 139 \times 10) = (\beta_3 + \beta_5 \times 139 + \beta_6 \times 10 + \beta_7 \times 1390)$$

(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 β s.

Solution:

This model is,

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon.$$

So, our answer is simply,

$$\beta_3.$$

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.

Solution:

```
birthday = 18760613
set.seed(birthday)

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)$stat
  tt_p[i] = t.test(values ~ groups, data = sim_data, var.equal = TRUE)$p.val
}
```

(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!

Solution:

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

```
## [1] 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!

Solution:

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

```
## [1] 0.02667
```

(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?

Solution:

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

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

The p-value is the same for both tests each time!

(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?

Solution:

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

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

One is the negative of the other! This is R's fault due to its choice for the dummy variable in the regression. The group A and group B relationship was being used differently in the regression and *t*-test.