Week 7 - Homework

STAT 420, Summer 2018, Connor Segneri - segneri3

Contents

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.

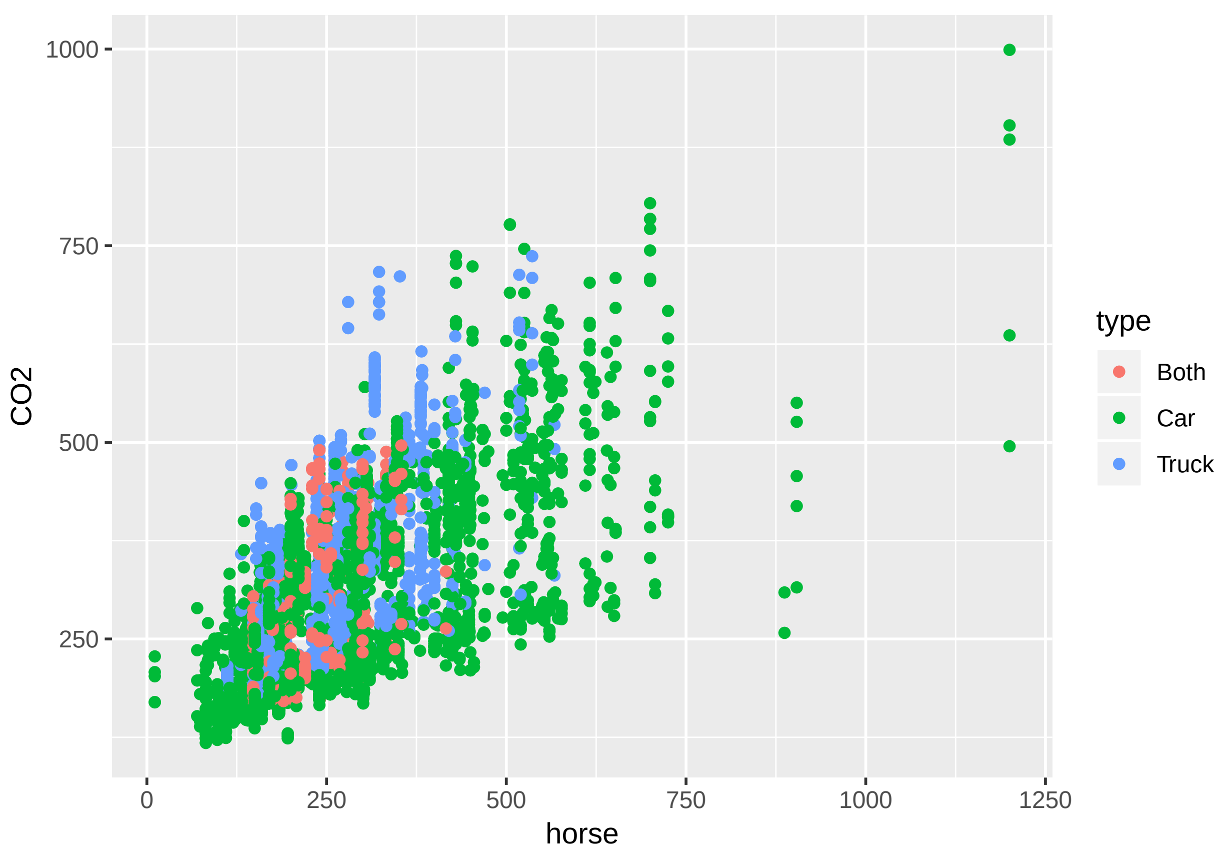
(a) Do the following:

- 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.)

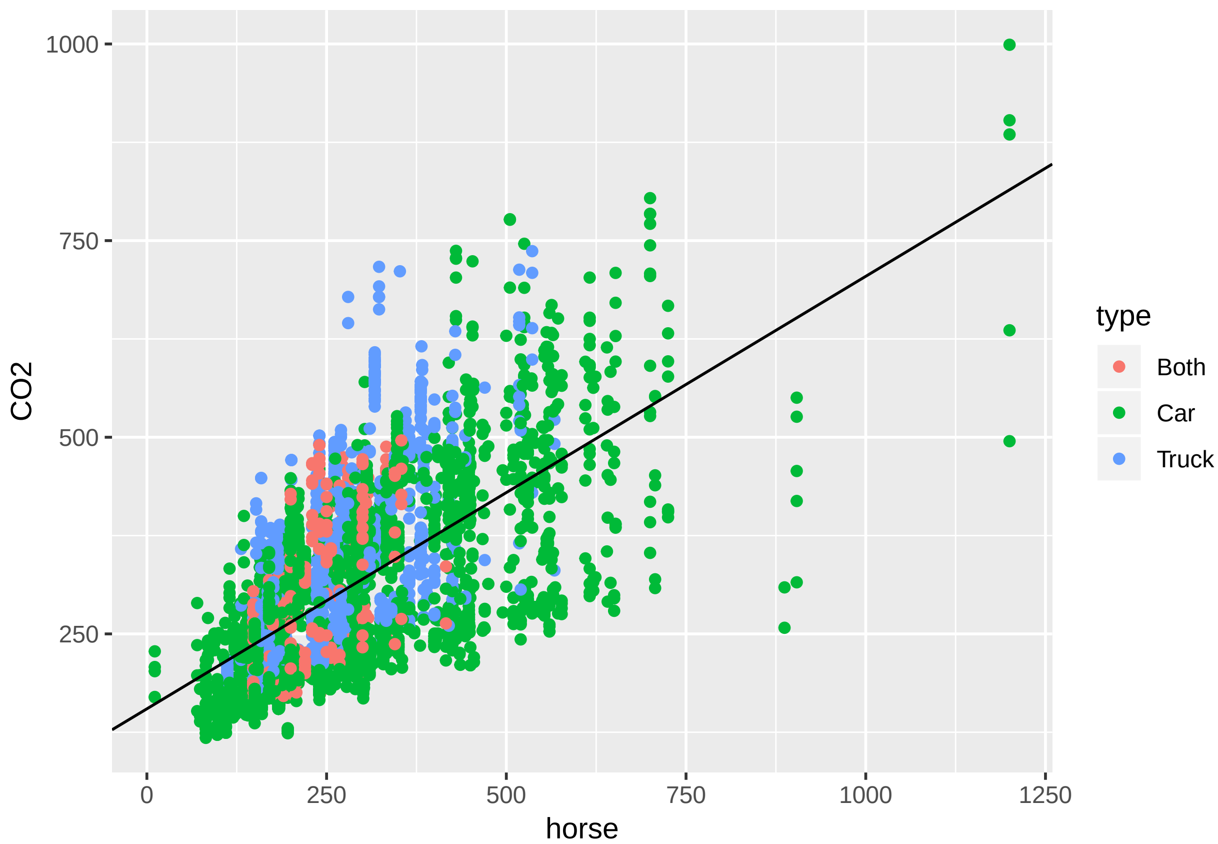
```
library(readr)
library(ggplot2)

epa2015 = read_csv('epa2015.csv')
epa2015$type = as.factor(epa2015$type)

ggplot(epa2015, aes(x=horse, y=CO2, color=type)) +
  geom_point()
```




```
model1 = lm(CO2 ~ horse, data = epa2015)
ggplot(epa2015, aes(x=horse, y=CO2, color=type)) +
  geom_point() +
  geom_abline(slope = coef(model1)[2], intercept = coef(model1)[1])
```



Looking at the plot, it seems like the linear regression line fits the data well.

Using this model, an estimate for the average change in `CO2` for a one foot-pound per second increase in `horse` for a vehicle type `car` is `0.5499`. This is the same for every type `car`, because this model only uses `horse` as a predictor.

```
predict(model1, interval = 'prediction', level = .90, newdata = data.frame(
  horse = 148
))
```

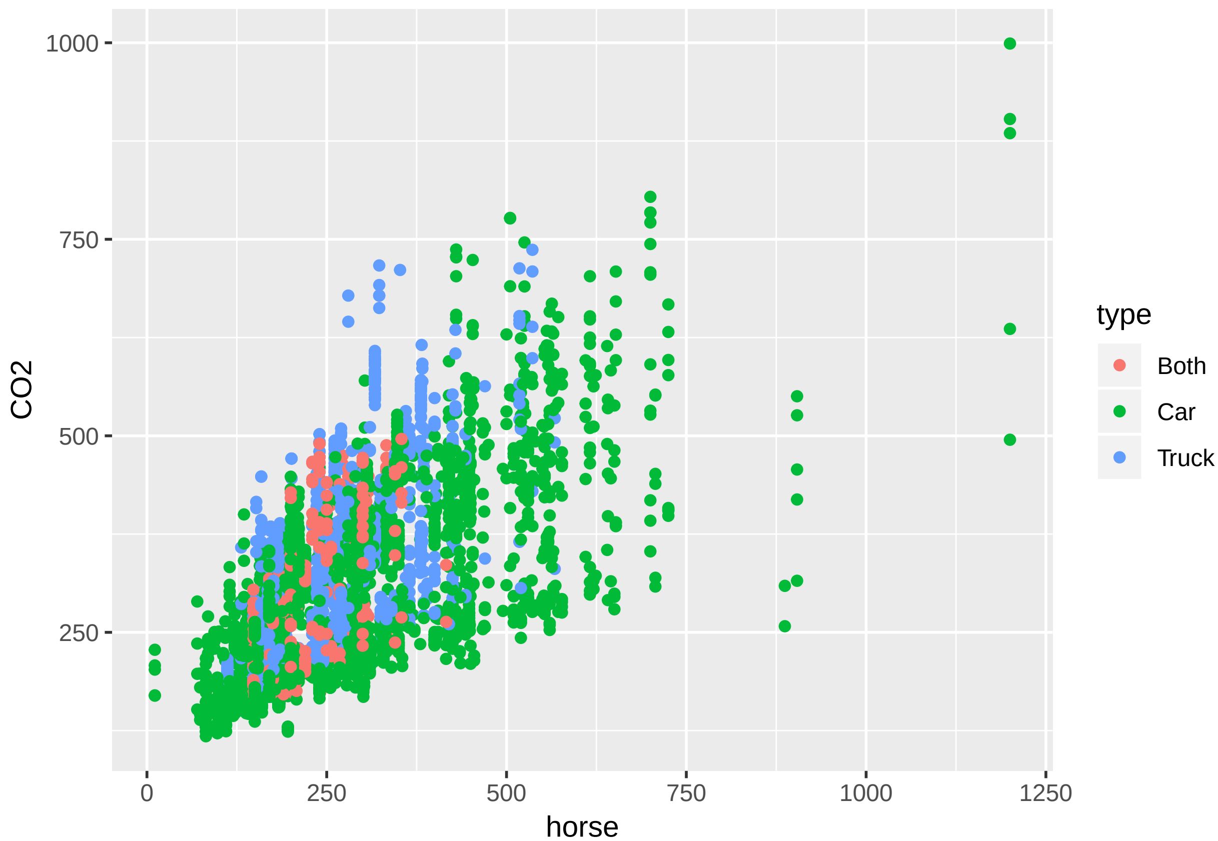
```
##     fit   lwr   upr
## 1 236.1 89.46 382.7
```

The prediction interval is seen above. Again, the only predictor used was horsepower.

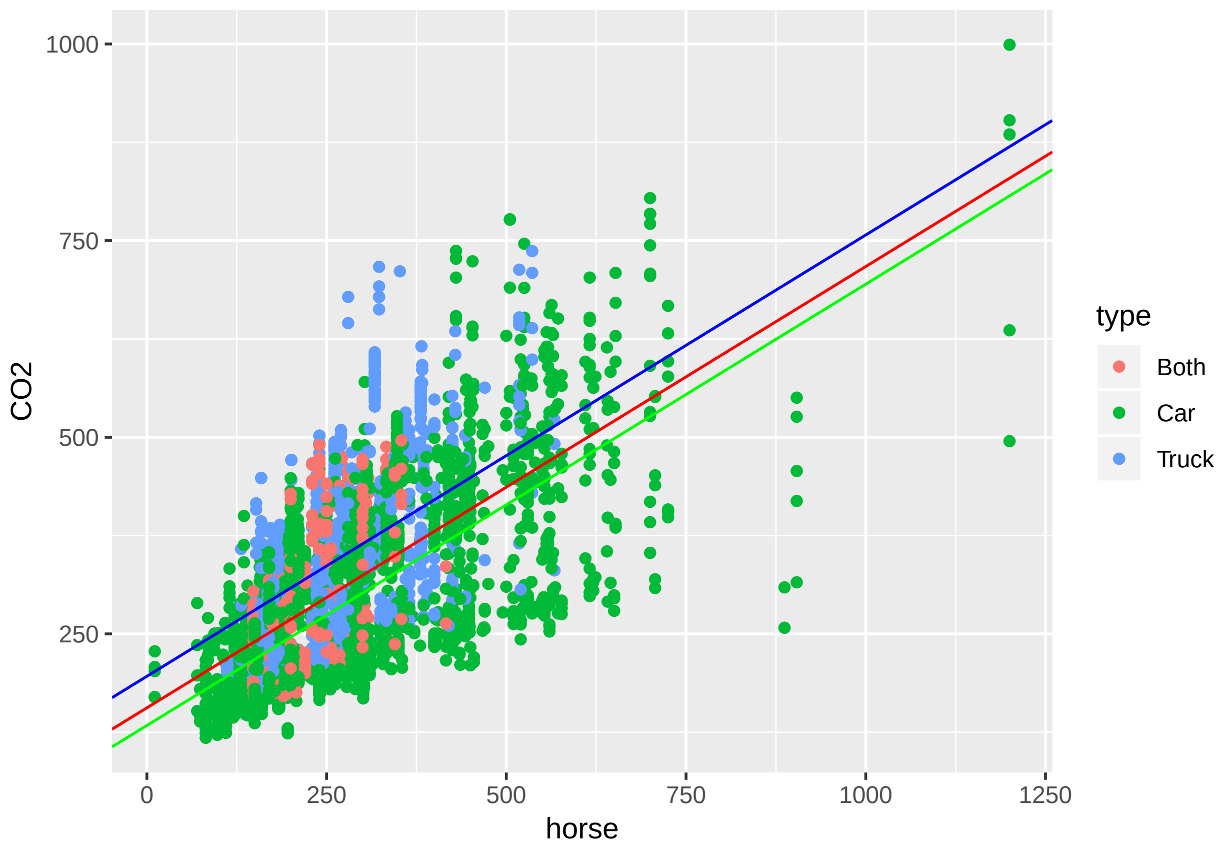
(b) Do the following:

- 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
ggplot(epa2015, aes(x=horse, y=CO2, color=type)) +
  geom_point()
```




```r
model2 = lm(CO2 ~ horse + type, data = epa2015)
ggplot(epa2015, aes(x=horse, y=CO2, color=type)) +
  geom_point() +
  geom_abline(slope = coef(model2)[2], intercept = coef(model2)[1], color = 'red') +
  geom_abline(slope = coef(model2)[2], intercept = coef(model2)[1] + coef(model2)[3], color = 'green') +
  geom_abline(slope = coef(model2)[2], intercept = coef(model2)[1] + coef(model2)[4], color = 'blue')
```



Looking at the plot above, it seems that all three lines still fit the data fairly well. Visually, it seems that type `car` may have the best fit.

Based on the 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` would be `0.5611`. This is the same change in `CO2` across all types, because the slope stays consistent across categorical variables in an additive model.

```
predict(model2, interval = 'prediction', level = .90, newdata = data.frame(
  horse = 148,
  type = 'Both'
))
```

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

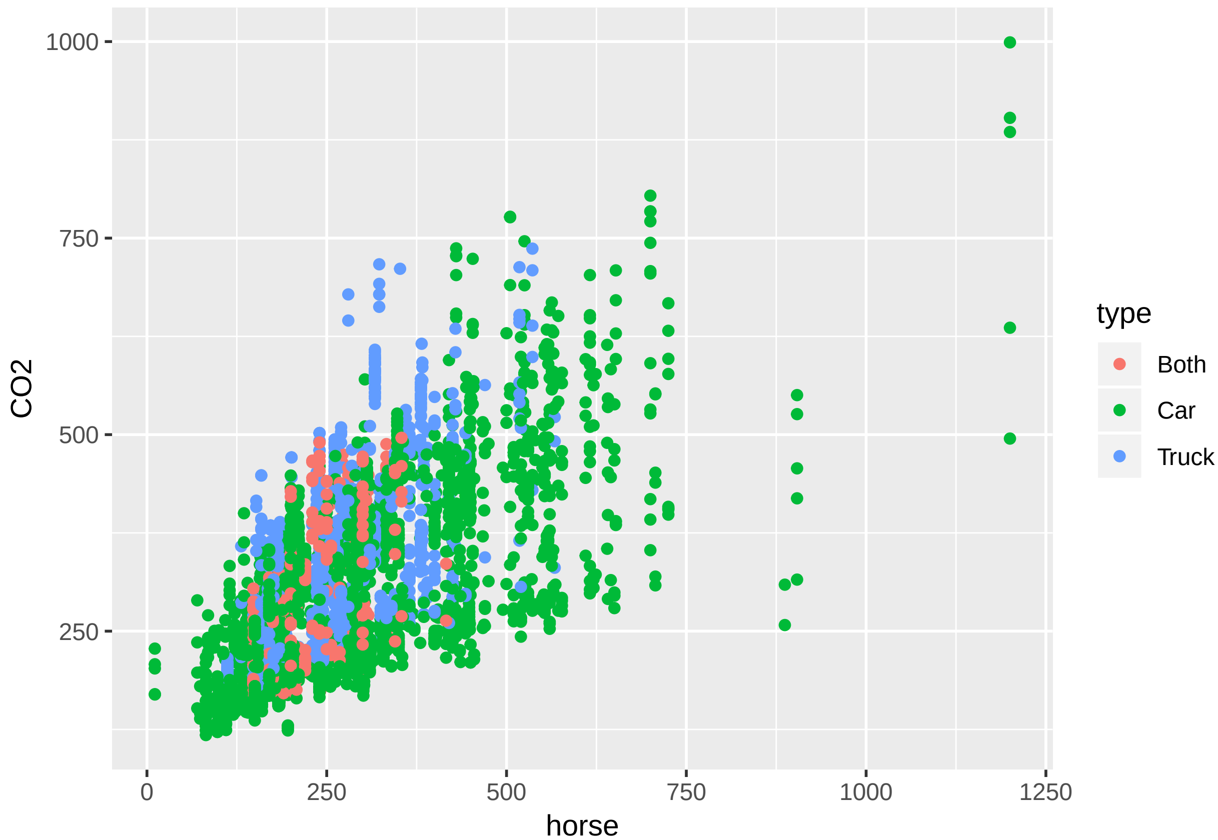
The prediction interval is above.

(c) Do the following:

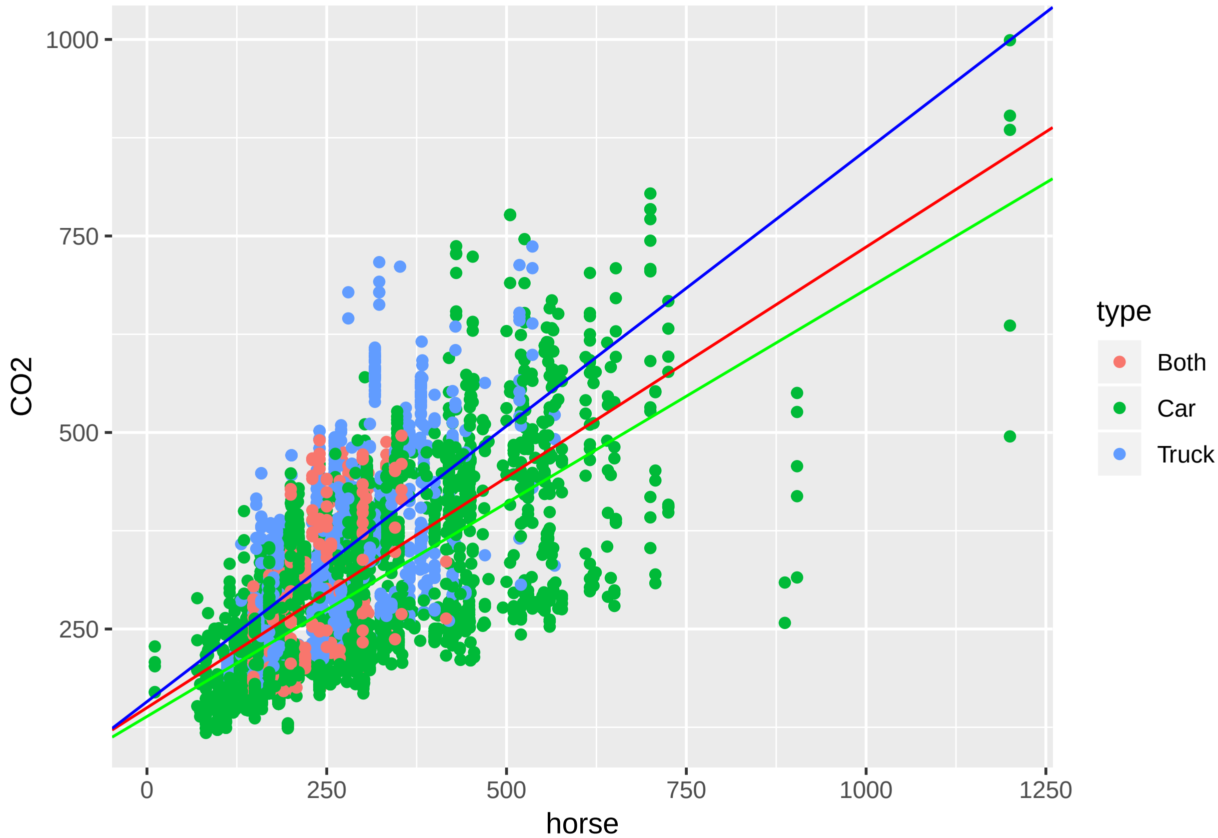
- Make a scatterplot of `CO2` versus `horse`. Use a different color point for each vehicle `type`.
- Fit an interaction multiple regression model with `CO2` as the response and `horse` and `type` as the predictors.
- Add the fitted regression "lines" to the scatterplot with the same colors as their respective 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`.

```
ggplot(epa2015, aes(x=horse, y=CO2, color=type)) +
  geom_point()
```




```
model3 = lm(CO2 ~ horse * type, data = epa2015)
ggplot(epa2015, aes(x=horse, y=CO2, color=type)) +
  geom_point() +
  geom_abline(slope = coef(model3)[2], intercept = coef(model3)[1], color = 'red') +
  geom_abline(slope = coef(model3)[2] + coef(model3)[5], intercept = coef(model3)[1] + coef(model3)[3],
  geom_abline(slope = coef(model3)[2] + coef(model3)[6], intercept = coef(model3)[1] + coef(model3)[4],
```



Because this is an interaction model, the slopes of each line changed for each `type`. Visually, this seems like each line fits their corresponding `type` data points much better.

Using this model, an estimate for the average change in `CO2` for a one foot-pound per second increase in `horse` for a vehicle type `car` is `0.5432`. This change is unique to type `car`, because this is an interaction model.

```
predict(model3, interval = 'prediction', level = .90, newdata = data.frame(
  horse = 148,
  type = 'Both'
))
```

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

The prediction interval is above.

(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(model2, model3)
```

```
## 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
```

The p-value above is well below $\alpha = 0.10$, so I would prefer the 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`?

```r
hospital = read_csv('hospital.csv')

hospital$Education = as.factor(hospital$Education)
hospital$Income = as.factor(hospital$Income)
hospital$Sex = as.factor(hospital$Sex)

levels(hospital$Education)
```

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

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

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

```r
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`?

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

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

`R` chooses `female` as the reference level for `Sex` and `high` as the reference level for both `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?

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

anova(model_add, model_intr_part)
```

```
## 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
```

Based on the p-value above and a significance level of $\alpha = 0.10$, I would prefer the model created in part **(b)**, the purely 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?

```r
model_intr_more = lm(Blood ~ Age + Education + Income + Sex + Sex:Age + Sex:Income + Income:Age + Age:I

anova(model_add, model_intr_more)
```

```
## Analysis of Variance Table
##
## Model 1: Blood ~ Age + Education + Income + Sex
```

```
## 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    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 above and a significance level of $\alpha = 0.10$, I would prefer the model created in this section, **(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.

Using the model in **(d)**, an 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`. 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

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

coef(days_add)
```

```
## (Intercept)    Pressure       Blood        Rate
##    -5.01784     0.08008     0.20956     0.13374
```

```
coef(days_int)
```

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

```
anova(days_add, days_int)
```

```
## Analysis of Variance Table
##
## Model 1: Days ~ Pressure + Blood + Rate
## Model 2: Days ~ Blood * Pressure * 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
```

The null hypothesis is that the interaction between the predictors is not useful in predicting `Days`. The alternative hypothesis is that the interaction between the predictors is useful.

The value of the test statistic is `2.0426`.

The p-value of the test is `0.087`

The statistical decision at $\alpha = 0.10$ is to reject the null hypothesis. The interaction between the predictors is useful in predicting `Days`. Because of this, I'd prefer the second model that contains the interaction between predictors.

(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.

```r
coef(days_int)
```

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

```r
coef(days_int)[4] + coef(days_int)[6]*10 + coef(days_int)[7]*139 + coef(days_int)[8]*139*10
```

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

Expression: $\beta_3 + \beta_5 x_1 + \beta_6 x_2 + \beta_7 x_1 x_2$

(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.

```r
coef(days_add)
```

```
## (Intercept)    Pressure       Blood        Rate
##    -5.01784     0.08008     0.20956     0.13374
```

```r
coef(days_add)[4]
```

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

Expression: 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.

```r
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.

```r
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
```

```r
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

```
set.seed(19951015)

for (i in 1:num_sims) {
  sim_data$values = rnorm(n, mean = 42, sd = 3.5) # simulate response data
  model = summary(lm(values ~ groups, data = sim_data))
  ttest = t.test(values ~ groups, data = sim_data, var.equal = TRUE)

  lm_t[i] = model$coefficients[2,3]
  lm_p[i] = model$coefficients[2,4]
  tt_t[i] = ttest$statistic
  tt_p[i] = ttest$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
```

(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] 0.03
```

(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?

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

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

The above assertion of `TRUE` means that the p-values obtained from the regression and the p-values obtained from the t.test are analytically equal.

(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
head(lm_t)
```

```
## [1] -0.1350  0.4832 -0.1327 -1.3821 -0.5582 -0.2843
```

```r
head(tt_t)
```

```
## [1]  0.1350 -0.4832  0.1327  1.3821  0.5582  0.2843
```

The two vectors are numerically equivalent, except for the fact that they have opposite signs. Where `lm_t` would have a positive, `tt_t` has the same value, but negative. This is why the initial comparison led me to believe that no values were equal, but when using `all.equal`, it showed that the two vectors were in fact, analytically equal.