STAT 471/571/701 Modern Data Mining, HW 1

Group Member 1
Group Member 2
Group Member 3

Due: 11:59PM, September 15, 2019

Contents

1 Overview

This is a fast-paced course that covers a lot of material. There will be a large amount of references. You may need to do your own research to fill in the gaps in between lectures and homework/projects. It is impossible to learn data science without getting your hands dirty. Please budget your time evenly. Last-minute work ethic will not work for this course.

1.1 Objectives

- Get familiar with `R-studio` and `RMarkdown`
- Learn data science essentials
  - gather data
  - clean data
  - summarize data
  - display data
  - conclusion
- Packages
  - `lm()`
  - `dplyr`
  - `ggplot`
- Methods
  - normality
  - sampling distribution
  - confidence intervals
  - p-values
  - linear models

1.2 Instructions

- **Homework assignments can be done in a group consisting of up to three members**. Please find your group members as soon as possible and register your group on our Canvas site.

- **All work submitted should be completed in the R markdown format.** You can find a cheat sheet for R Markdown here. For those who have never used it before, we urge you to start this homework as soon as possible.

- **Submit the following files, one submission for each group:** (1) Rmd file, (2) a compiled PDF or HTML version, and (3) all necessary data files. You can directly edit this file to add your answers. If you intend to work on the problems separately within your group, compile your answers into one Rmd file before submitting. We encourage that you at least attempt each problem by yourself before working with your teammates. Additionally, ensure that you can 'knit' or compile your Rmd file. It is also likely that you need to configure Rstudio to properly convert files to PDF. **These instructions** should be helpful.

- In general, be as concise as possible while giving a fully complete answer. All necessary datasets are available in the "Data" folder or this homework folder on Canvas. Make sure to document your code with comments so the teaching fellows can follow along. R Markdown is particularly useful because it follows a 'stream of consciousness' approach: as you write code in a code chunk, make sure to explain what you are doing outside of the chunk.

- A few good submissions will be used as sample solutions. When those are released, make sure to compare your answers and understand the solutions.

2 Review materials

- Study both R-tutorials

- Study lecture 1: EDA/Simple regression
- Study lecture 2: Multiple regression

3 Case study: Women in Science

Are women underrepresented in science in general? How does gender relate to the type of educational degree pursued? Does number of higher degrees increase over the years? In an attempt to answer these questions, we assembled a data set (`WomenData_06_16.xlsx`) from NSF about various degrees granted in the U.S. from 2006 to 2016. It contains the following variables: Field (Non-science-engineering (`Non-S&E`) and sciences (`Computer sciences`, `Mathematics and statistics`, etc.)), Degree (`BS`, `MS`, `PhD`), Sex (`M`, `F`), Number of degrees granted, and Year.

Our goal is to answer the above questions only through EDA (Exploratory Data Analyses) without formal testing.

3.1 Load the data

Notice the data came in as an excel file. We need to use a package `readxl` and the function `read_excel()` to read the data `WomenData_06_16.xlsx` into R.

1. Read the data into R.
2. Clean the names of each variables.
3. Set the variable natures properly.
4. Provide a quick summary of the data set.
5. Write a summary describing the data set provided here.

To help out, we have included some codes here as references. You should make this your own chunks filled with texts going through each items listed above. Make sure to hide the unnecessary outputs/code etc.

```r
# For the demonstration purpose, we show this R-chunk by taking echo=TRUE
# In your final report you should hide all the R-chunks to keep your report flowing well.
wsci <- read_excel("WomenData_06_16.xlsx")
# str(wsci)

# base R
names(wsci)[1] <- "Field"
names(wsci)[5] <- "Number"
# names(wsci)
wsci$Field  <-   as.factor(wsci$Field)
wsci$Degree <- as.factor(wsci$Degree)
wsci$Sex <- as.factor(wsci$Sex)
# summary(wsci)

# dplyr way
# wsci %<>%
#   rename(Field = "Field and sex",
#          Number = "Degrees Awarded") %>%
#   mutate(Field = as.factor(Field),
#          Degree = as.factor(Degree),
#          Sex = as.factor(Sex))
```

3.2 EDA

3.2.1 Focus on BS degree and in 2015

Is there evidence that more males are in science related fields vs `Non-S&E`? Provide summary statistics and a plot which shows the number of people by gender and by field. Write a brief summary to describe your findings.

3.2.2 In 2015

Describe the number of people by type of degree, field, and gender. Do you see any evidence of gender effects over types of degree? Again, provide graphs to summarize your findings.

3.2.3 Time effects

In this last portion of the EDA, we ask you to provide evidence graphically: Do the number of degrees change by gender, field, and time?

3.2.4 Women in Data Science

Finally, is there evidence showing that women are underrepresented in data science? Data science is an interdisciplinary field of computer science, math, and statistics. You may include year and/or degree.

3.3 Final brief report

Summarize your findings focusing on answering the questions regarding if we see consistent patterns that more males pursue science-related fields. Any concerns with the data set? How could we improve on the study?

3.4 Appendix

Here are several sample codes for your reference.

```r
wsci %>%  # to get the average number of ppl by gender
  group_by(Field, Sex) %>%
  summarise(deg = mean(Number))

wsci %>%
  mutate(SE = ifelse(Field!="Non-S&E" , "S&E", "Non-S&E")) %>%
  group_by(SE, Sex) %>%
  summarise(SE_number = sum(Number)) %>%
  ggplot(aes(x = SE, y = SE_number, fill = Sex)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(axis.text.y = element_text(angle = 60)) +
  ggtitle("Degrees granted by S&E vs non-S&E by gender")

wsci %>%
  filter(Year == 2007) %>%
  ggplot(aes(x = Field, y = Number, fill = Sex)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_grid(Degree~., scales = "free_y") +
  theme(axis.text.x = element_text(angle = 30, hjust = 1)) +
  ggtitle("Degrees granted across fields by degree and gender")

wsci %>%
  mutate(SE = ifelse(Field!="Non-S&E" , "S&E", "Non-S&E")) %>%
  group_by(SE, Sex, Year) %>%
  summarise(SE_number = sum(Number)) %>%
  group_by(SE, Year) %>%
  mutate(ratio = SE_number / sum(SE_number)) %>%
  filter(Sex == "Female") %>%
  ggplot(aes(x = Year, y = ratio, color = SE)) +
  geom_point() + geom_line() +
  ggtitle("Female proportion in SE/non-SE across year")

wsci %>%
  mutate(SE = ifelse(Field!="Non-S&E" , "S&E", "Non-S&E")) %>%
  group_by(SE, Sex, Year, Degree) %>%
  summarise(SE_number = sum(Number)) %>%
  group_by(SE, Year, Degree) %>%
  mutate(ratio = SE_number / sum(SE_number)) %>%
  filter(Sex == "Female") %>%
  ggplot(aes(x = Year, y = ratio, color = SE)) +
  geom_point() + geom_line() +
  facet_grid(~Degree)+
  ggtitle("Female proportion in SE/non-SE across year by degree")


wsci %>%
  mutate(SE = ifelse(Field!="Non-S&E" , "S&E", "Non-S&E")) %>%
  group_by(SE, Sex, Year, Degree) %>%
  summarise(SE_number = sum(Number)) %>%
  ggplot(aes(x = Year, y = SE_number, fill = Sex)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_grid(SE~Degree, scales = "free_y") +
  ggtitle("Degrees granted by sex, degree and SE")


wsci %>%
  mutate(SE = ifelse(Field!="Non-S&E" , "S&E", "Non-S&E")) %>%
  group_by(SE, Sex, Year, Degree) %>%
  summarise(SE_number = sum(Number)) %>%
  ggplot(aes(x = Year, y = SE_number, fill = Sex)) +
  geom_bar(stat = "identity", position = "fill") +
  facet_grid(SE~Degree, scales = "free_y") +
  ggtitle("Degrees granted proption by sex across degree and SE")


wsci %>%
  filter(Field %in% c("Computer sciences", "Mathematics and statistics")) %>%
  ggplot(aes(x = Year, y = Number, fill = Sex)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_grid(Field~Degree, scales = "free_y") +
  ggtitle("Degrees granted proption by sex across degree and SE")


wsci %>%
  ggplot(aes(x = Year, y = Number, fill = Sex)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_grid(Field~Degree, scales = "free_y") +
  ggtitle("Degrees granted proption by sex across degree and SE")
```

4 Simple Regression

4.1 Linear model through simulations

This exercise is designed to help you understand the linear model using simulations. In this exercise, we will generate (x_i, y_i) pairs so that all linear model assumptions are met.

Presume that $\mathbf{x}$ and $\mathbf{y}$ are linearly related with a normal error $\boldsymbol{\varepsilon}$, such that $\mathbf{y} = 1 + 1.2\mathbf{x} + \boldsymbol{\varepsilon}$. The standard deviation of the error ε_i is $\sigma = 2$.

We can create a sample input vector $(n = 40)$ for $\mathbf{x}$ with the following code:

```
# Generates a vector of size 40 with equally spaced values between 0 and 1, inclusive
x <- seq(0, 1, length = 40)
```

4.1.1 Generate data

Create a corresponding output vector for $\mathbf{y}$ according to the equation given above. Use `set.seed(1)`. Then, create a scatterplot with (x_i, y_i) pairs. Base R plotting is acceptable, but if you can, please attempt to use `ggplot2` to create the plot. Make sure to have clear labels and sensible titles on your plots.

4.1.2 Understand the model

i. Find the LS estimates of $\boldsymbol{\beta}_0$ and $\boldsymbol{\beta}_1$, using the `lm()` function. What are the true values of $\boldsymbol{\beta}_0$ and $\boldsymbol{\beta}_1$? Do the estimates look to be good?

ii. What is your RSE for this linear model fit? Is it close to $\sigma = 2$?

iii. What is the 95% confidence interval for $\boldsymbol{\beta}_1$? Does this confidence interval capture the true $\boldsymbol{\beta}_1$?

iv. Overlay the LS estimates and the true lines of the mean function onto a copy of the scatterplot you made above.

4.1.3 diagnoses

i. Provide residual plot where fitted $\mathbf{y}$-values are on the x-axis and residuals are on the y-axis.

ii. Provide a normal QQ plot of the residuals.

iii. Comment on how well the model assumptions are met for the sample you used.

4.1.4 Understand sampling distribution and confidence intervals

This part aims to help you understand the notion of sampling statistics and confidence intervals. Let's concentrate on estimating the slope only.

Generate 100 samples of size $n = 40$, and estimate the slope coefficient from each sample. We include some sample code below, which should guide you in setting up the simulation. Note: this code is easier to follow but suboptimal; see the appendix for a more optimal R-like way to run this simulation.

```
# Inializing variables. Note b_1, upper_ci, lower_ci are vectors
x <- seq(0, 1, length = 40)
n_sim <- 100              # number of simulations
b1 <- 0                   # n_sim many LS estimates of beta_1 (=1.2). Initialize to 0 for now
upper_ci <- 0             # upper bound for beta_1. Initialize to 0 for now.
lower_ci <- 0             # lower bound for beta_1. Initialize to 0 for now.
```

```
t_star <- qt(0.975, 38)   # Food for thought: why 38 instead of 40? What is t_star?

# Perform the simulation
for (i in 1:n_sim){I l
  y <- 1 + 1.2 * x + rnorm(40, sd = 2)
  lse <- lm(y ~ x)
  lse_output <- summary(lse)$coefficients
  se <- lse_output[2, 2]
  b1[i] <- lse_output[2, 1]
  upper_ci[i] <- b1[i] + t_star * se
  lower_ci[i] <- b1[i] - t_star * se
}
results <- as.data.frame(cbind(se, b1, upper_ci, lower_ci))

# remove unecessary variables from our workspace
rm(se, b1, upper_ci, lower_ci, x, n_sim, b1, t_star, lse, lse_out)
```

i. Summarize the LS estimates of β_1 (stored in `results$b1`). Does the sampling distribution agree with theory?

ii. How many of your 95% confidence intervals capture the true β_1? Display your confidence intervals graphically.

4.2 Major League Baseball

This question is about Major League Baseball (MLB) and team payrolls. Guiding questions: how do salaries paid to players affect team wins? How could we model win propensity?

We have put together a dataset consisting of the winning records and the payroll data of all 30 MLB teams from 1998 to 2014. There are 54 variables in the dataset, including:

- `payroll`: total team payroll (in $billions) over the 17-year period
- `avgwin`: the aggregated win percentage over the 17-year period
- winning percentage and payroll (in $millions) for each team are also broken down for each year.

The data is stored as `MLPayData_Total.csv` on Canvas.

4.2.1 Exploratory questions

For each of the following questions, there is a `dplyr` solution that you should try to answer with.

i. Which five teams spent the most money in total between years 2000 and 2004, inclusive?

ii. Between 1999 and 2000, inclusive, which team(s) "improved" the most? That is, had the biggest percentage gain in wins?

iii. Using `ggplot`, pick a single year, and plot the number of games won vs. `payroll` for that year (`payroll` on x-axis). You may use any 'geom' that makes sense, such as a scatterpoint or a label with the point's corresponding team name.

4.2.2 Effect of payroll

For a given year, is `payroll` a significant variable in predicting the winning percentage of that year? Choose a single year and run a regression to examine this. You may try this for a few different years. You can do this programmatically (i.e., for every year) if you are interested, but it is not required.

4.2.3 Reverse regression

With the aggregated information, use regression to analyze total payroll and overall winning percentage. Run appropriate model(s) to answer the following questions:

i. In this analysis, do the Boston Red Sox perform reasonably well given their total amount spent on payroll? [Use a 95% interval.]

ii. Given their winning percentage, how much would you have expected the Oakland A's to have spent on total payroll? (Use a 95% interval.)

5 Multiple Regression

5.1 Auto data set

This question utilizes the `Auto` dataset from ISLR. The original dataset contains 408 observations about cars. It is similar to the CARS dataset that we use in our lectures. To get the data, first install the package ISLR. The `Auto` dataset should be loaded automatically. We'll use this dataset to practice the methods learnt so far.

You can access the necessary data with the following code:

```
# Read in the Auto dataset
auto_data <- ISLR::Auto
```

Get familiar with this dataset first. Tip: you can use the command `?ISLR::Auto` to view a description of the dataset.

5.1.1 EDA

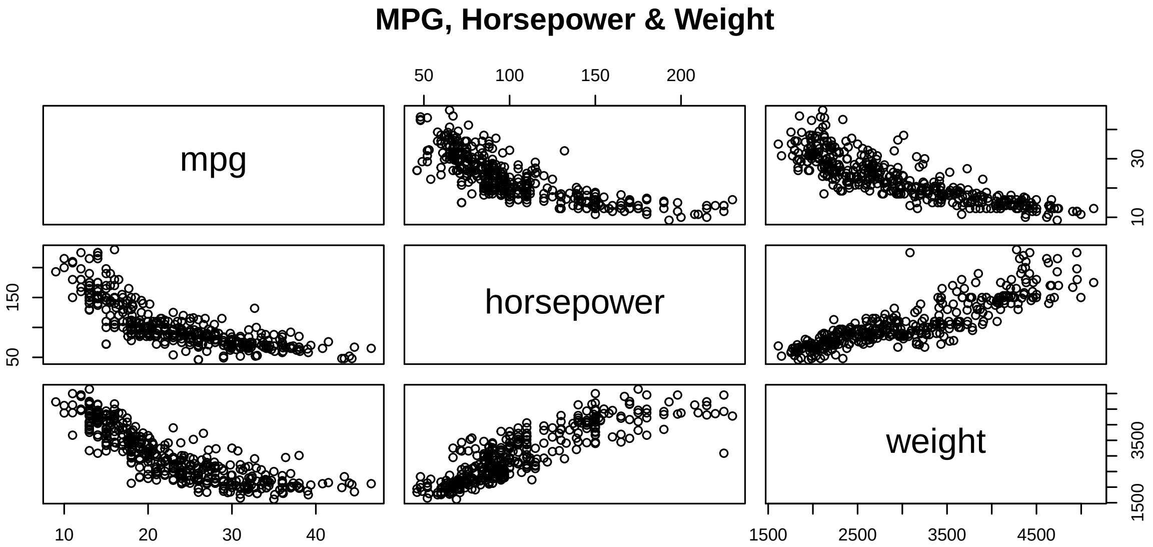
Explore the data, with particular focus on pairwise plots and summary statistics. Briefly summarize your findings and any peculiarities in the data.

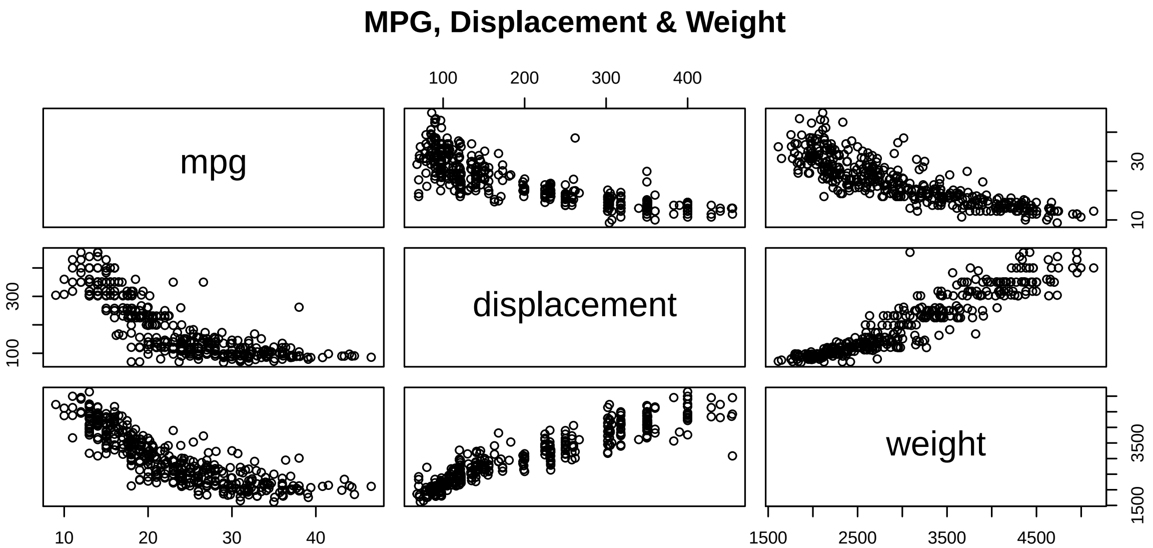
We will first observe a summary of our data to familiarize ourselves with types and magnitudes.

```
##       mpg         cylinders     displacement   horsepower
##  Min.   : 9.0   Min.   :3.00   Min.   : 68   Min.   : 46.0
##  1st Qu.:17.0   1st Qu.:4.00   1st Qu.:105   1st Qu.: 75.0
##  Median :22.8   Median :4.00   Median :151   Median : 93.5
##  Mean   :23.4   Mean   :5.47   Mean   :194   Mean   :104.5
##  3rd Qu.:29.0   3rd Qu.:8.00   3rd Qu.:276   3rd Qu.:126.0
##  Max.   :46.6   Max.   :8.00   Max.   :455   Max.   :230.0
##
##      weight      acceleration       year         origin
##  Min.   :1613   Min.   : 8.0   Min.   :70   Min.   :1.00
##  1st Qu.:2225   1st Qu.:13.8   1st Qu.:73   1st Qu.:1.00
##  Median :2804   Median :15.5   Median :76   Median :1.00
##  Mean   :2978   Mean   :15.5   Mean   :76   Mean   :1.58
##  3rd Qu.:3615   3rd Qu.:17.0   3rd Qu.:79   3rd Qu.:2.00
##  Max.   :5140   Max.   :24.8   Max.   :82   Max.   :3.00
##
##                  name
##  amc matador       :  5
##  ford pinto        :  5
##  toyota corolla    :  5
##  amc gremlin       :  4
##  amc hornet        :  4
##  chevrolet chevette:  4
##  (Other)           :365
```

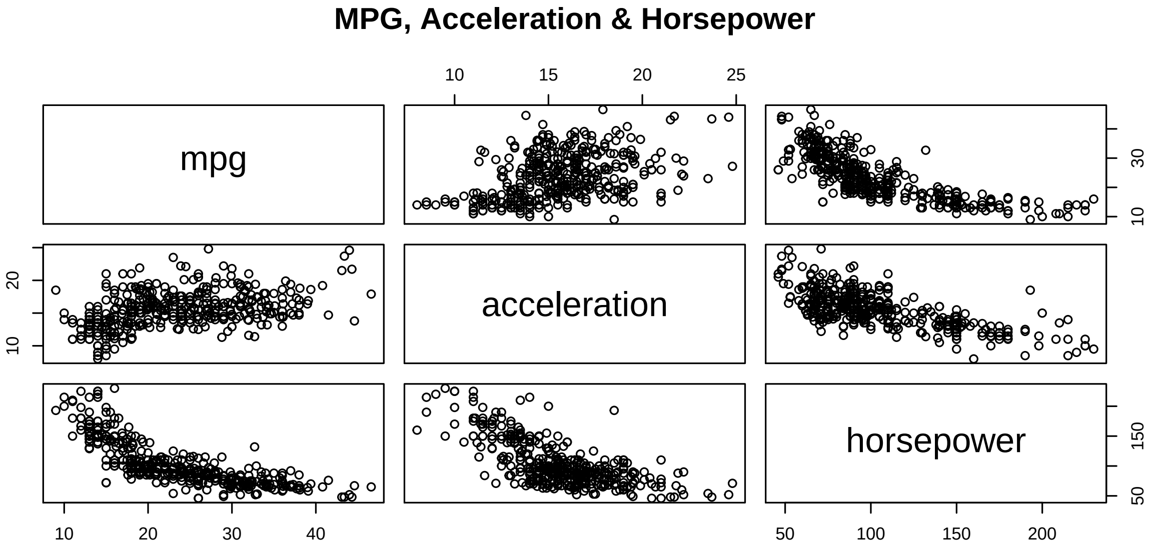
From the `summary`, we can see that most of our variables can be treated as continuous. The exceptions are `origin`, `name` (which is too specific to be significant in our modelling) and arguably `cylinders`, as it is a numerical but discrete variable that can take 6 consecutive integer values.

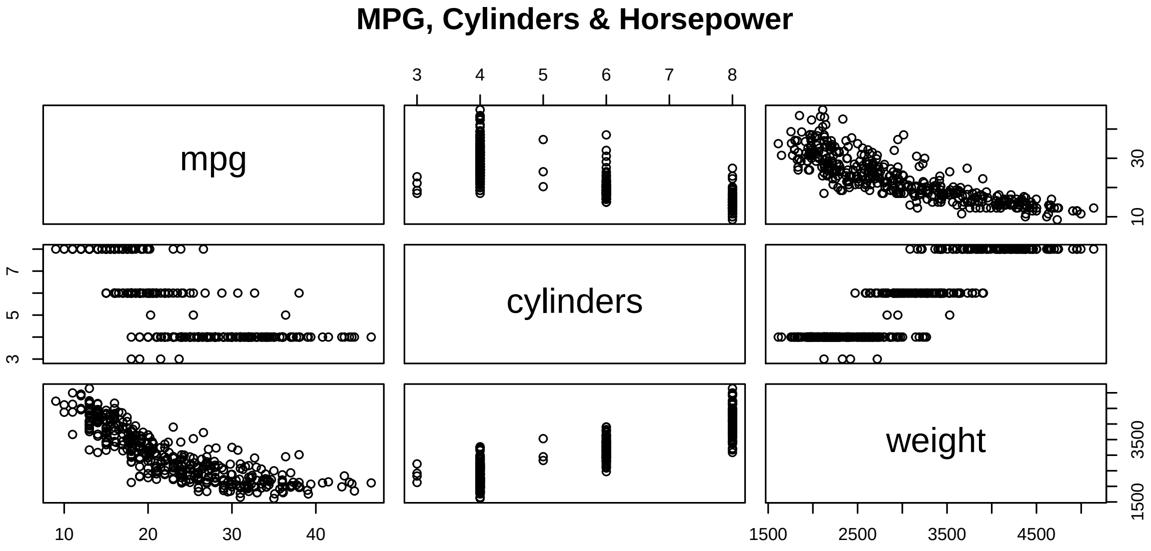
To get any insight about the relationships present among our variables, we must produce scatter plots. Note that we are not only interested in the relationship between any independent variable and `MPG`, but in potential relationships among the independent variables, as including highly correlated variables in our model may tamper with significance and produce unreliable coefficients. Spotting linear relationships between variables will allow us to adjust the future model accordingly to avoid multicorrelation and obtain a sound significance level for all of our predictors.

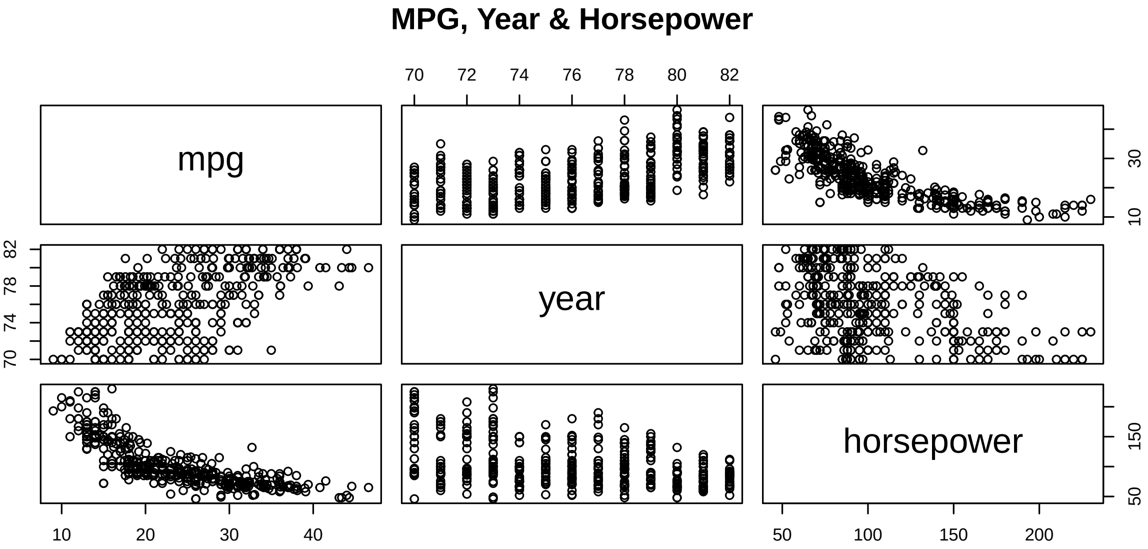

MPG, Horsepower & Weight


MPG, Displacement & Weight

From a first glance we see that the variables `displacement`, `horsepower` and `weight` are quite correlated with each other, and theyr scatterplots with mpg have a similar shape. Note that this shape is not precisely linear, but looks like a non-linearly decreasing variable. We may want to avoid including all three of these variables at once. Moreover, we may find productive to apply a transformation to one of these variables in order to obtain a truly linear relation and normal residuals after fitting our model.

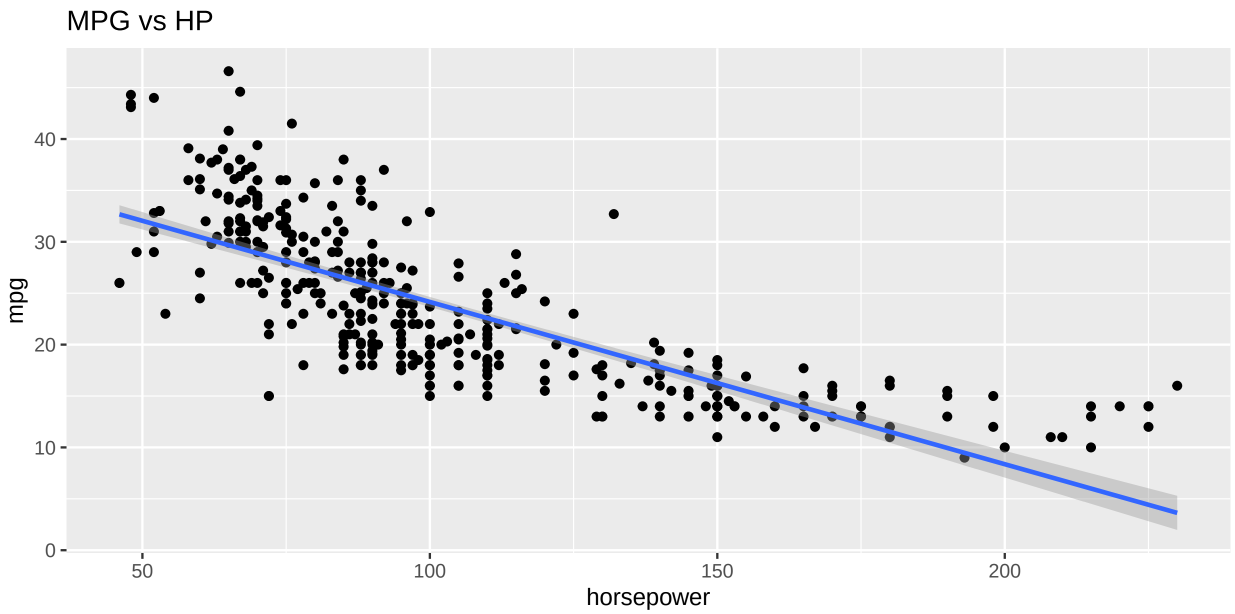




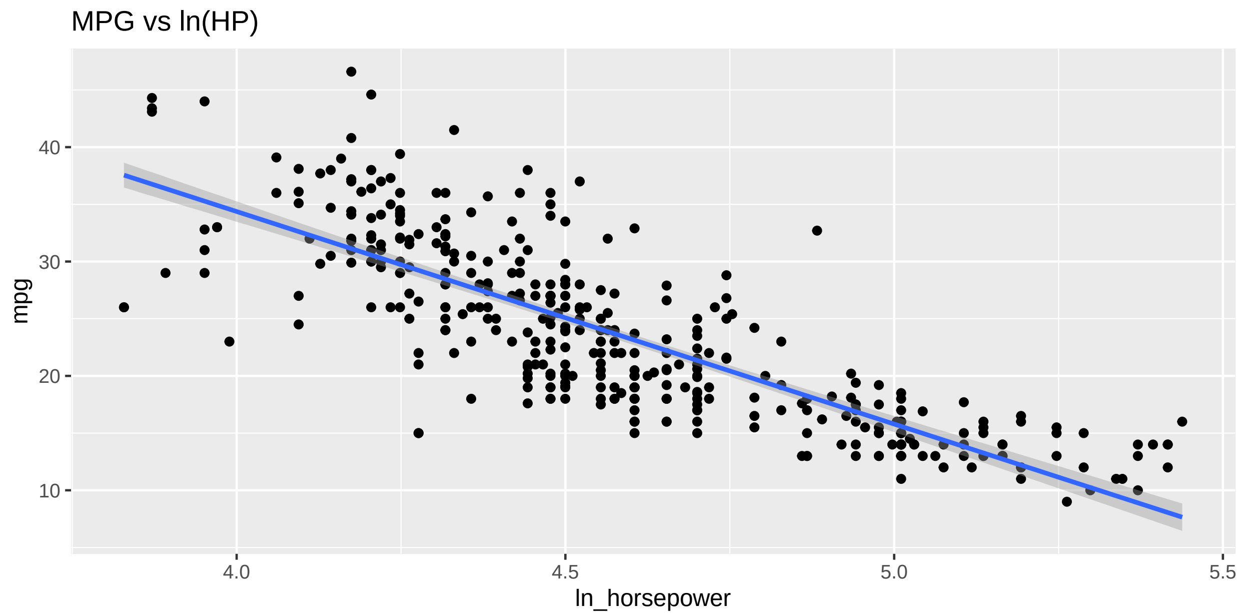


Note that acceleration's relationship with `mpg` is weak, and it is correlated with `horsepower`quite significantly, so we might deide to exclude it from the model. The variable `cylinders`show some peculiarities. Most observations have a `cylinders` value of 4, 6, or 8, so it may be better interpreted as a categorical variable. Finally, note that `year`is uncorrelated with `horsepower`, so including them both in the same model should not cause multicolinearity problems.

Now, we have noticed that some of our variables seem to have a negative but somewhat non-linear relationship with `mpg`. One of such vaiables is `horsepower`. Thus, we will now explore applying a transformation to `horsepower` and check if the transformed variables displays a more linear relationship with `mpg`. Let us examine the scatterplot in more detail:

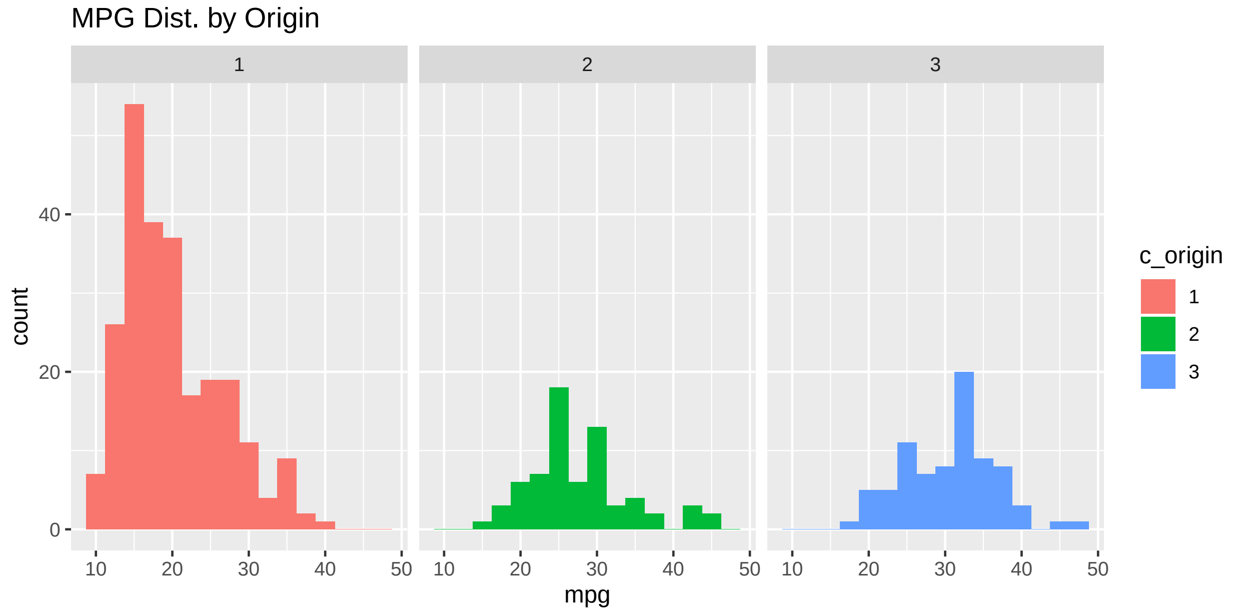


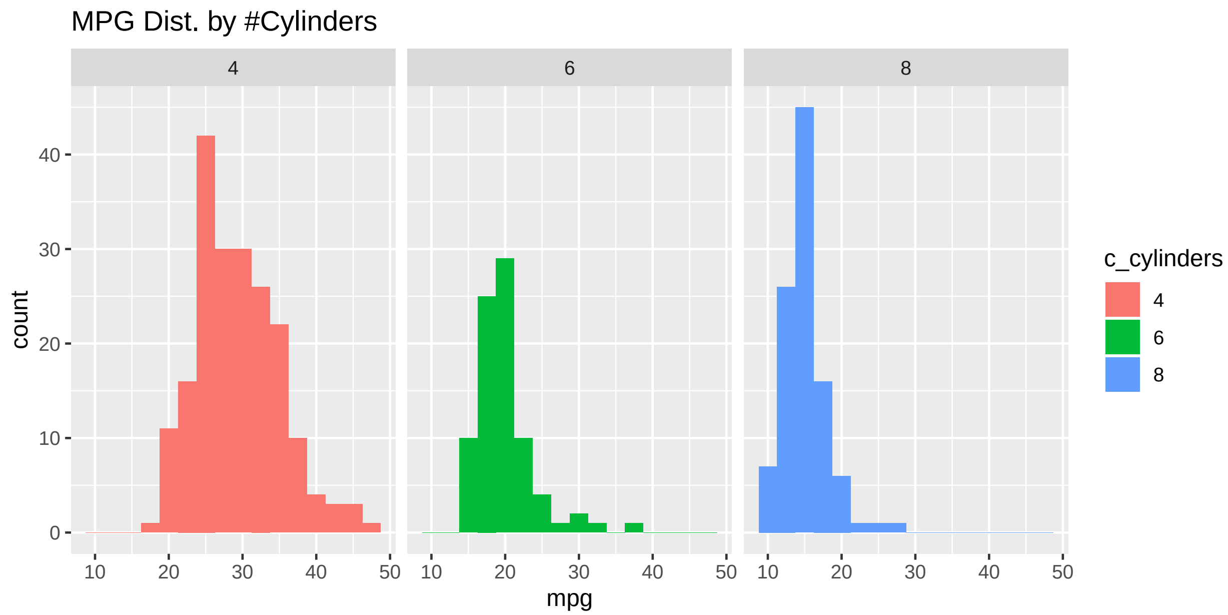
At a first glance, the data resembles an exp(-x) function. We will try a logarithmic transformation on `horsepower` an check the scatterplot for improvements:



Now the relationship is closer to linear. We may benefit from applying such a transformation to variables like `horsepower` in our final model.

We would also like to explore the relevance of our two categorical variables: `origin` and `cylinders` (interpreted as categorical). To assess different levels of `mpg` across categories, we will plot histograms of `mpg` for each. Note that as very few observations have `cylinders` values of 3 or 5, we will exclude this due to lack of data.





While variation across origins is moderate, level variation based on number of cylinders is clearer. Moreover, variation by origin might explained by other variables.

5.1.2 What effect does time have on MPG?

i. Start with a simple regression of `mpg` vs. `year` and report R's `summary` output. Is `year` a significant variable at the .05 level? State what effect `year` has on `mpg`, if any, according to this model.

```
##
## Call:
## lm(formula = mpg ~ year, data = auto_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.021  -5.441  -0.441   4.974  18.209
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -70.0117     6.6452   -10.5   <2e-16 ***
## year          1.2300     0.0874    14.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.36 on 390 degrees of freedom
## Multiple R-squared:  0.337,  Adjusted R-squared:  0.335
## F-statistic:  198 on 1 and 390 DF,  p-value: <2e-16
```

The p-value for the coefficient of `year` is very small, capped at 2e-16 $<$ 0.05; thus, `year` is a significant predictor at the 0.05 level. Our interretation of the `year` coefficient is that as it increases by 1, the average `mpg` of cars produced in the year is 1.23 higher.

ii. Add `horsepower` on top of the variable `year` to your linear model. Is `year` still a significant variable at the .05 level? Give a precise interpretation of the `year`'s effect found here.

```
##
## Call:
```

```
## lm(formula = mpg ~ horsepower + year, data = auto_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.077  -3.078  -0.431   2.588  15.315
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -12.73917    5.34903   -2.38    0.018 *
## horsepower   -0.13165    0.00634  -20.76   <2e-16 ***
## year          0.65727    0.06626    9.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.39 on 389 degrees of freedom
## Multiple R-squared:  0.685,  Adjusted R-squared:  0.684
## F-statistic:  424 on 2 and 389 DF,  p-value: <2e-16
```

After adding `horseower`, we confirm that year is still a significant predictor, with a very small p-value that remains far below 0.05. The coefficient of `year`has a different interpretation, as the "partial derivative" of `mpg` with respect to `year`. This means that an increase in `year` by 1, controlling for `horsepower`, results in an increase of 0.65 in the average car `mpg`.

iii. The two 95% CI's for the coefficient of year differ among (i) and (ii). How would you explain the difference to a non-statistician?

The CI's for the `year` coefficients are different because fundamentally, these coeffient represent two different concepts. In the simple model, we are in a way assuming we know nothing else about `mpg` apart from its relation to year, and we are using the latter to explain all teh variation in the former. In the multivariate model, we take more information into account, specifically from a second variable `horsepower`, and so we explain variation in `year` using the two, such that the coefficent of `year` only represent the expected change in average `mpg` for a 1-unit increase in `year` but controlling for `horsepower`, that is, assuming no chnage in `horsepower`. This new extra asusmption fundamentally changes the meaning of the coeffient, and thus the model produces a different estimate and CI accordingly.

iv. Create a model with interaction by fitting `lm(mpg ~ year * horsepower)`. Is the interaction effect significant at .05 level? Explain the year effect (if any).

```
##
## Call:
## lm(formula = mpg ~ horsepower * year, data = auto_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.349  -2.451  -0.456   2.406  14.444
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -1.27e+02   1.21e+01  -10.45   <2e-16 ***
## horsepower       1.05e+00   1.15e-01    9.06   <2e-16 ***
## year             2.19e+00   1.61e-01   13.59   <2e-16 ***
## horsepower:year -1.60e-02   1.56e-03  -10.22   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.9 on 388 degrees of freedom
## Multiple R-squared:  0.752,  Adjusted R-squared:  0.75
## F-statistic:  393 on 3 and 388 DF,  p-value: <2e-16
```

All our predictors, including the interaction effect, are statistically significant at the 0.05 level. The year effect, although not as straightforward as before, can still be derived mathematically by taking the partial derivative of our fitted `mpg` with respect to `year`. We can say that as `year` increases by 1, the average `mpg` is expected to increase by 2.19 + -0.016 * `horsepower`, controlling for `horsepower`.

5.1.3 Categorical predictors

Remember that the same variable can play different roles! Take a quick look at the variable `cylinders`, and try to use this variable in the following analyses wisely. We all agree that a larger number of cylinders will lower mpg. However, we can interpret `cylinders` as either a continuous (numeric) variable or a categorical variable.

i. Fit a model that treats `cylinders` as a continuous/numeric variable: `lm(mpg ~ horsepower + cylinders, ISLR::Auto)`. Is `cylinders` significant at the 0.01 level? What effect does `cylinders` play in this model?

The variable `cylinders` is significant at the 0.01 level. In this model, we take `cylinders` as a continuous variable, and so the its effect can be interpreted from the model in line with previous examples: as `cylinders` increases by one unit (as we consider a car with one more cylinder), average `mpg` is expeted to decrease by 1.9198.

ii. Fit a model that treats `cylinders` as a categorical/factor variable: `lm(mpg ~ horsepower + as.factor(cylinders), ISLR::Auto)`. Is `cylinders` significant at the .01 level? What is the effect of `cylinders` in this model? Use `anova(fit1, fit2)` and `Anova(fit2)` to help gauge the effect. Explain the difference between `anova()` and `Anova`.

```
## 
## Call:
## lm(formula = mpg ~ horsepower + as.factor(cylinders), data = auto_data)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
##  -9.59  -2.71  -0.61   1.90  16.33 
## 
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)    
## (Intercept)            30.7761     2.4128   12.76   <2e-16 ***
## horsepower             -0.1030     0.0113   -9.09   <2e-16 ***
## as.factor(cylinders)4   6.5734     2.1692    3.03   0.0026 ** 
## as.factor(cylinders)5   5.0737     3.2666    1.55   0.1212    
## as.factor(cylinders)6  -0.3441     2.1858   -0.16   0.8750    
## as.factor(cylinders)8   0.4974     2.2764    0.22   0.8272    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 4.27 on 386 degrees of freedom
## Multiple R-squared:  0.705,  Adjusted R-squared:  0.701
## F-statistic:  184 on 5 and 386 DF,  p-value: <2e-16
```

We immediately notice that some category coefficients are not significant. We conduct an F-test to make sure check if `cylinders` as a whole categorical variable is significant. We will do this by producing a reduced model with no `cylinder` categories and comparing it with the full model using the F-statistic.

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ horsepower
## Model 2: mpg ~ horsepower + as.factor(cylinders)
##   Res.Df  RSS Df Sum of Sq    F Pr(>F)
## 1    390 9386
## 2    386 7037  4      2349 32.2 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Anova Table (Type II tests)
##
## Response: mpg
##                      Sum Sq  Df F value Pr(>F)
## horsepower             1508   1    82.7 <2e-16 ***
## as.factor(cylinders)   2349   4    32.2 <2e-16 ***
## Residuals              7037 386
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As the F-test rejects the null, we confirm that `cylinders` is as a whole a statistically significant variables, but the t-test for the coefficient of some categories fails, possibly due to a small number of observations in some categories, as well as multicolinearity-like problems, where some categories of `cylinders` can't offer significant information once other variables are taken into account.

The effect of `cylinders` has a different interpretation, as it is now discretely divided into 5 categories - note that the category `cylinders = 3` is absorbed in the intercept. For car belonging to a certain category except 3, their average `mpg` is expected to change by the category coefficient, with respect to a car in category 3a and controlling for all other variables in the model.

Extra: Let's see if we can imporve the categorical `cylinders` model. We will get rid of the 3 and 5 cylinder categories and reproduce the model:

As we can see, all of our model coefficients are now significant to very precise levels. We will keep this result in mind when producing our final model.

iii. What are the fundamental differences between treating `cylinders` as a continuous and categorical variable in your models?

When `cylinders` is treated as a continuous variable, it is explicitly taken as a numeric vector and OLS will output a coefficient for it that will represent the maginal change of the dependent given a one-unit change in `cylinders`. When `cylinders` is taken as categorical, we mean to say that any observation belongs to one category within `cylinders`, and would like to find out whan we can say about observations belonging to each category. The OLS process will first create n-1 dummy variables for n categories, the first one assuems as the base line case and incorporated in our intercept. The vectors of 0's and 1's are then taken as literal numeric vectors and a coefficient is obtained for all n-1 categories. Such a coefficient represents the change (note, with respect to the baseline category) expected on average in the independent given an observation belongs to a specific category.

5.1.4 Results

Final modelling question: we want to explore the effects of each feature as best as possible. You may explore interactions, feature transformations, higher order terms, or other strategies within reason. The model(s) should be as parsimonious (simple) as possible unless the gain in accuracy is significant from your point of view.

i. Describe the final model. Include diagnostic plots with particular focus on the model residuals and diagnoses.

Our final model regresses `mpg` with respect to `ln(horsepower)`, `year`, `weight`, `year*weight` and a categorical variable defined based on cylinders: Less or equal to 4, or greater than 4. Before running the regression, we eliminated observations that had 3 or 5 cylinders, as they were too few to provide reliable information about these levels.

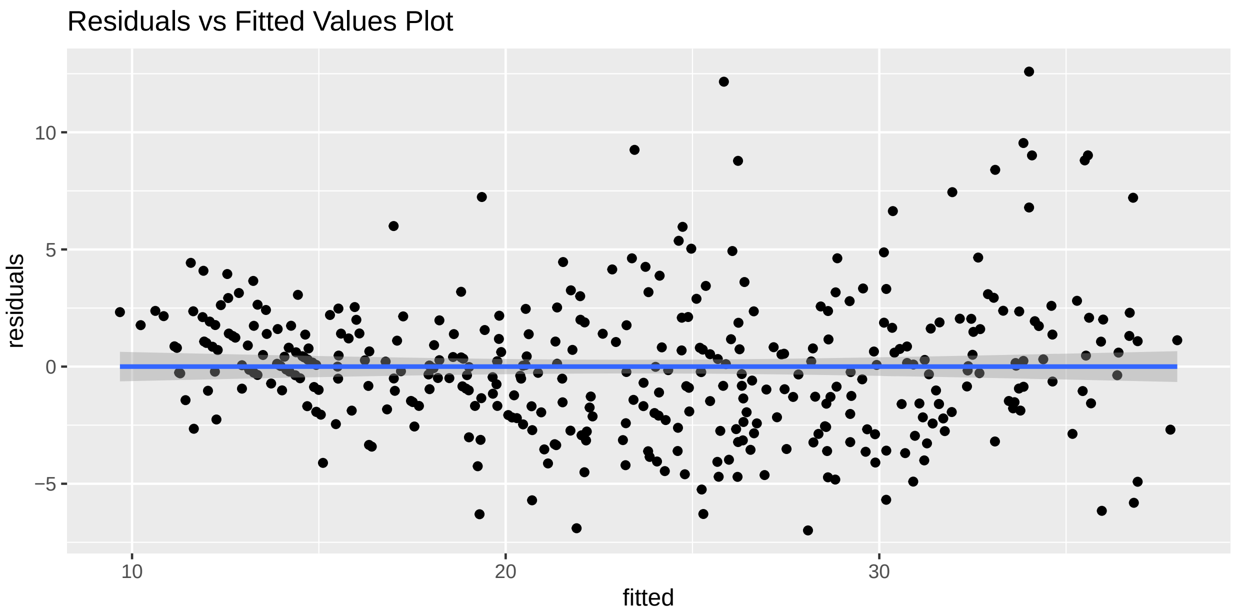
Below are the model output results:

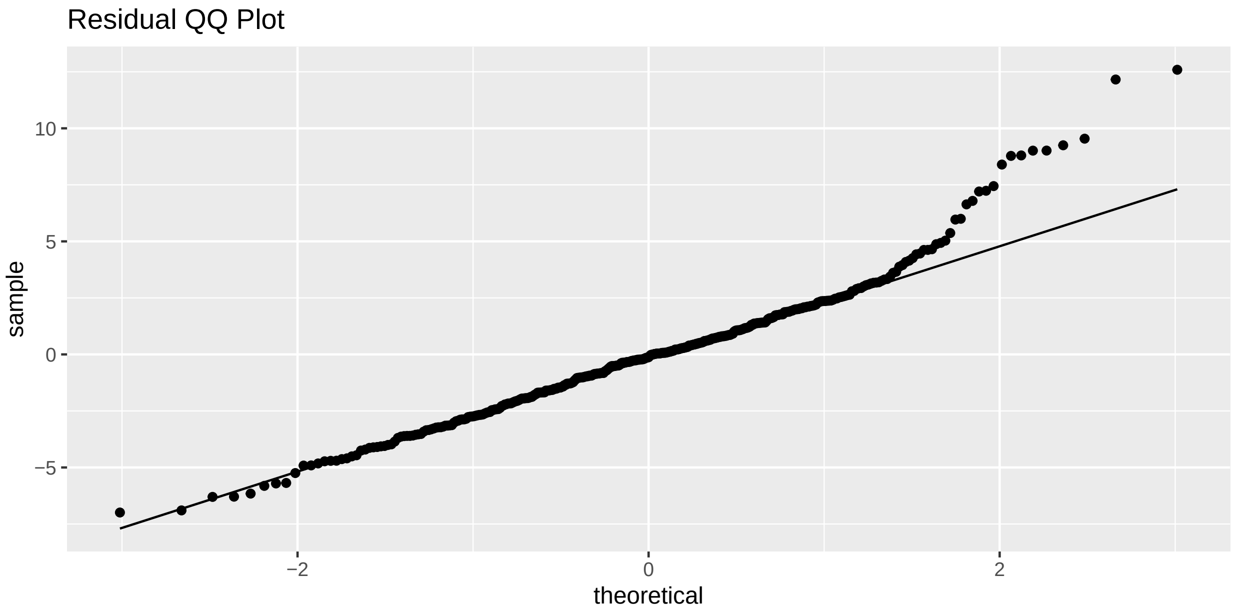
```
cars <- ISLR::Auto %>% filter(cylinders %in% c(4, 6, 8)) %>% mutate(ln_horsepower = log(horsepower), cy

fit <- lm(mpg ~ ln_horsepower + year * weight + as.factor(cyl_cat), cars)
summary(fit)
```

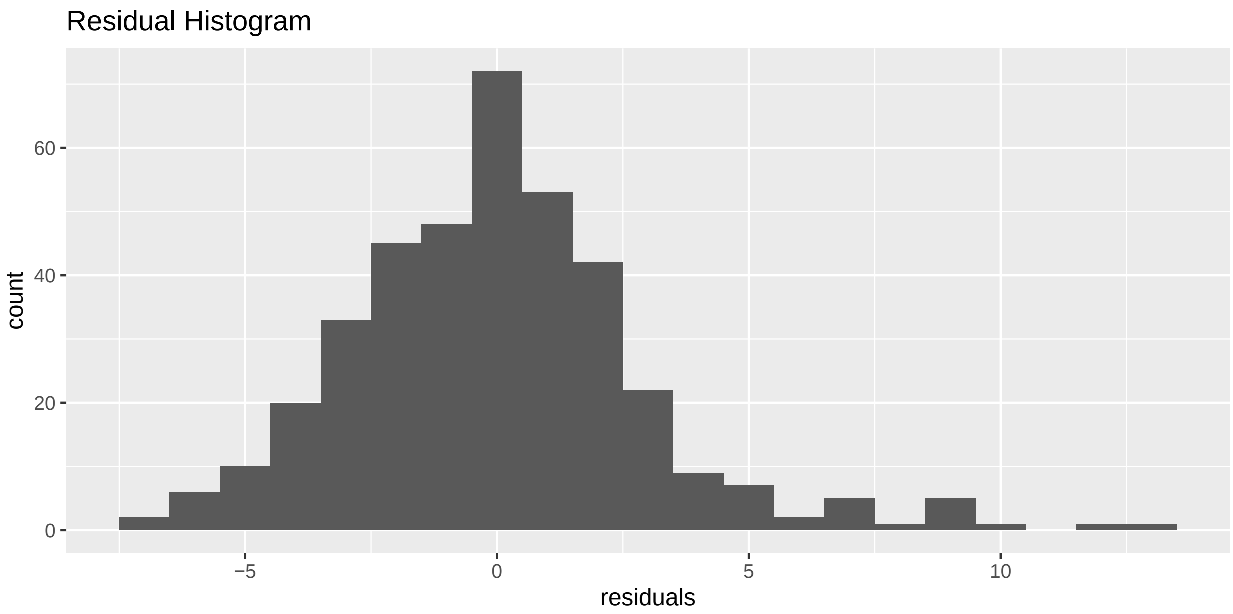
```
##
## Call:
## lm(formula = mpg ~ ln_horsepower + year * weight + as.factor(cyl_cat),
##     data = cars)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.992 -1.880 -0.125  1.481 12.590
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -8.70e+01   1.26e+01   -6.93  1.9e-11 ***
## ln_horsepower           -4.37e+00   9.66e-01   -4.52  8.1e-06 ***
## year                     1.91e+00   1.63e-01   11.72  < 2e-16 ***
## weight                   2.89e-02   4.21e-03    6.86  2.8e-11 ***
## as.factor(cyl_cat)TRUE -2.21e+00   5.40e-01   -4.09  5.3e-05 ***
## year:weight             -4.43e-04   5.63e-05   -7.87  3.7e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.95 on 379 degrees of freedom
## Multiple R-squared:  0.86,   Adjusted R-squared:  0.859
## F-statistic:  467 on 5 and 379 DF,  p-value: <2e-16
```

After several trials, this selection of variables and transformations produced a model with high predictive power whose estimated parameters are all statistically significant. We will now include some diagnostics information.





We confirm that the residuals are centered around 0 and have roughly constant variance across the fitted values, which is a good indicator of independence and homoscedasticity. Moreover, The residual QQ-plot reveals that residuals follow a normal distribution but are somewhat fat-tailed to the right. We plot a histogram of the residuals to confirm this.



Although we understand this is may be a relevant issue, thsi fenomenon was present in almost every other model we tested. Thus, we settle for this model as one of the best performers on the visual normality tests.

ii. Summarize the effects found.

The model coefficients estimate the marginal effects of our independent variables on the dependent `mpg`:

- Horsepower: Our model establishes that the relationship between `horsepower` and `mpg` is not linear, but logarithmic. Using a first approximation for the logarithm, we can say that an increase of 1% in `horsepower` results in a decrease in average `mpg` of -4.37.
- Year: We take into account the simple effet and the interaction effect. According to our model, an increase of 1 unit in `year` results in a change in average `mpg` of 1.91 - 0.000443 * `weight`.
- Weight: An increase of 1 unit in `weight` results in a change in average `mpg` of 0.0289 - 0.000443 * year.
- Cylnders: According to our model, cars with more than 4 cylinders have an lower average `mpg` by -2.21 compared to cars with less or eqal to 4 cylinders. Note: All of these effects are marginal, and must be interpreted as controlling for all other variables in the model. TO avoid redundance, we decided not to specify this fact in the explanation of every effect.

iii. Predict the `mpg` of the following car: A red car built in the US in 1983 that is 180 inches long, has eight cylinders, displaces 350 cu. inches, weighs 4000 pounds, and has a horsepower of 260. Also give a 95% CI for your prediction.

We create a new car observation following our dataframe structure:

Now we will use `predict` to retrieve a prediction for the `mpg` of our new car, and a confidence interval for this prediction:

```
##    fit lwr  upr
## 1 13.8 7.6 19.9
```

```
## [1] 1.05
```

```
## [1] 2.95
```

5.1.5 Appendix

This is code that is roughly equivalent to what we have used earlier but is more streamlined (simulations).

```r
simulate_lm <- function(n) {
  # note: `n` is an input but not used (don't worry about this hack)
  x <- seq(0, 1, length = 40)
  y <- 1 + 1.2 * x + rnorm(40, sd = 2)
  t_star <- qt(0.975, 38)
  lse <- lm(y ~ x)
  lse_out <- summary(lse)$coefficients
  se <- lse_out[2, 2]
  b1 <- lse_out[2, 1]
  upper_CI = b1 + t_star * se
  lower_CI = b1 - t_star * se
  return(data.frame(se, b1, upper_CI, lower_CI))
}

# this step runs the simulation 100 times,
# then matrix transposes the result so rows are observations
sim_results <- data.frame(t(sapply(X = 1:100, FUN = simulate_lm)))
```