STAT-S 432: Homework 1

Due January 24, 2019

1. Functions.

There are two functions below which are missing some or all of the body. The first one should generate data from a linear model. The second should estimate a linear model using an input dataframe and then make some plots to examine the fit.

Complete both functions.

```
generate.data <- function(n, p, sig.epsilon=1){
  ## you need some more inputs
  ## sig.epsilon - (optional), what is this?
  X = matrix(rnorm(p*n), ncol=p)
  epsilon = rnorm(n, sd = sig.epsilon)
  beta = p:1
  beta.0 = 3
  y = beta.0 + X%*%beta + epsilon
  df = data.frame(y, X)
  return(df)
}
estimate.and.plot <- function(form, dataframe, plotme = TRUE){
  ## Estimates and (optionally plots some diagnostics for) a linear model
  ## Takes in a formula, as formula('y~x') or somesuch
  ## and data frame
  ## plotme determines ...
  mdl = lm(form, data=dataframe)
  if(plotme){
    preds = labels(terms(form, data=dataframe))
    df = dataframe[preds]
    df$resids = residuals(mdl)  # how do you get residuals?
    df$fit =  fitted(mdl)# how do you get the fitted values?
    preds.vs.resids = df %>%
      gather(-c(resids,fit), key='predictor', value='value')
    # create a new dataframe for ggplot
    # what does this do? Takes the data and converts it to a 'long' form data.frame
    # Long form is for compatibility with ggplot
    # The predictor variable values are put into a single vector
    # the value vector is a factor vector which labels the values of the predictors with
    # the respective predictors name.
    # residuals and predicted values are then matched up with the predictor values
    # this means residual and predicted vectors are replicated for each predictor variable
    p1 <- ggplot(preds.vs.resids, aes(x=value, y=resids)) + geom_point() +
      geom_smooth() + facet_wrap(~predictor, scales = 'free')
      # creates a plot of the predictors versus residuals.
      # facet_wrap makes sure a graph for each predictor is created
      # geom_smooth adds a smooth curve to assess the mean value of the
      # residuals across the range of the predictors
    p2 <- ggplot(df, aes(sample=resids)) + geom_qq() + geom_qq_line()
    # creaetes a qq_plot of the residuals for assessing normality
    print(p1) # print out the first plot (wouldn't do this inside a function generally)
```

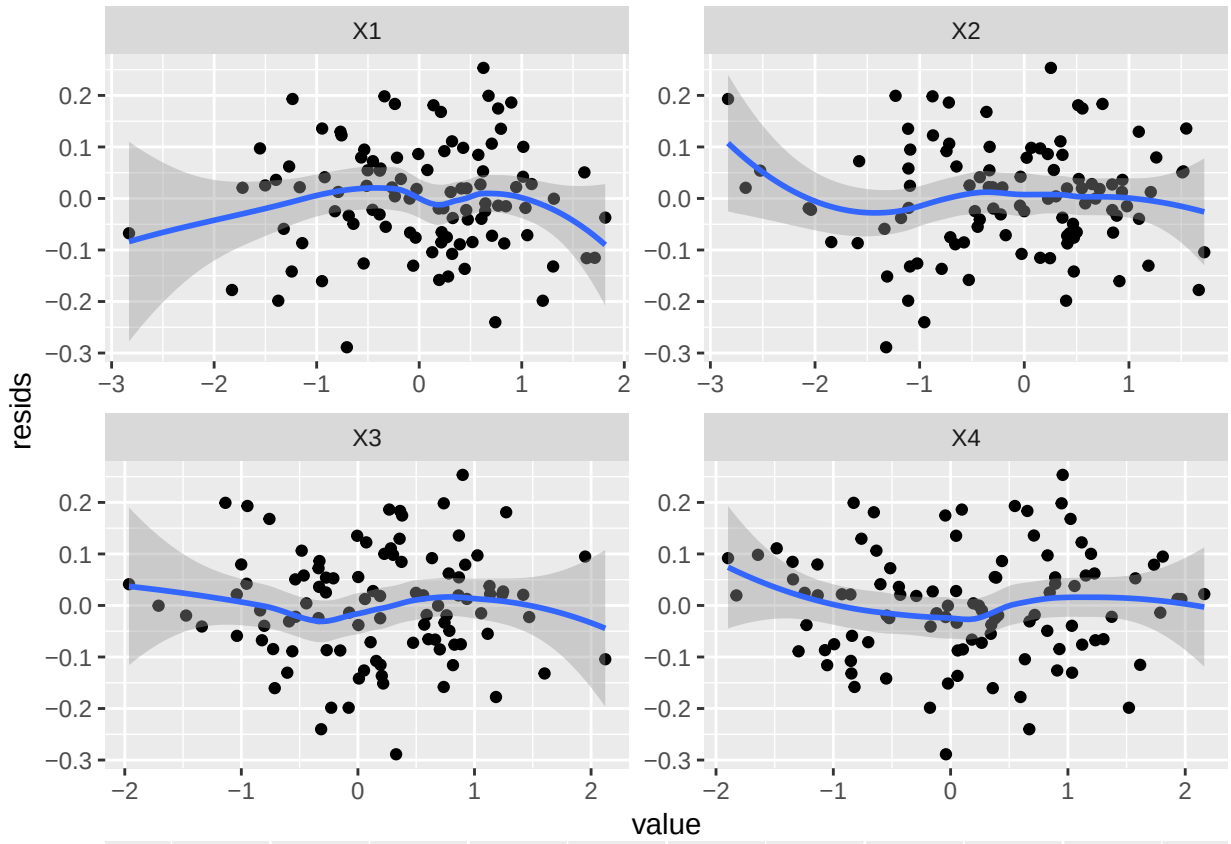
```
    print(p2) # print out the second plot
  }
  return(mdl) # output our fitted model
}
```

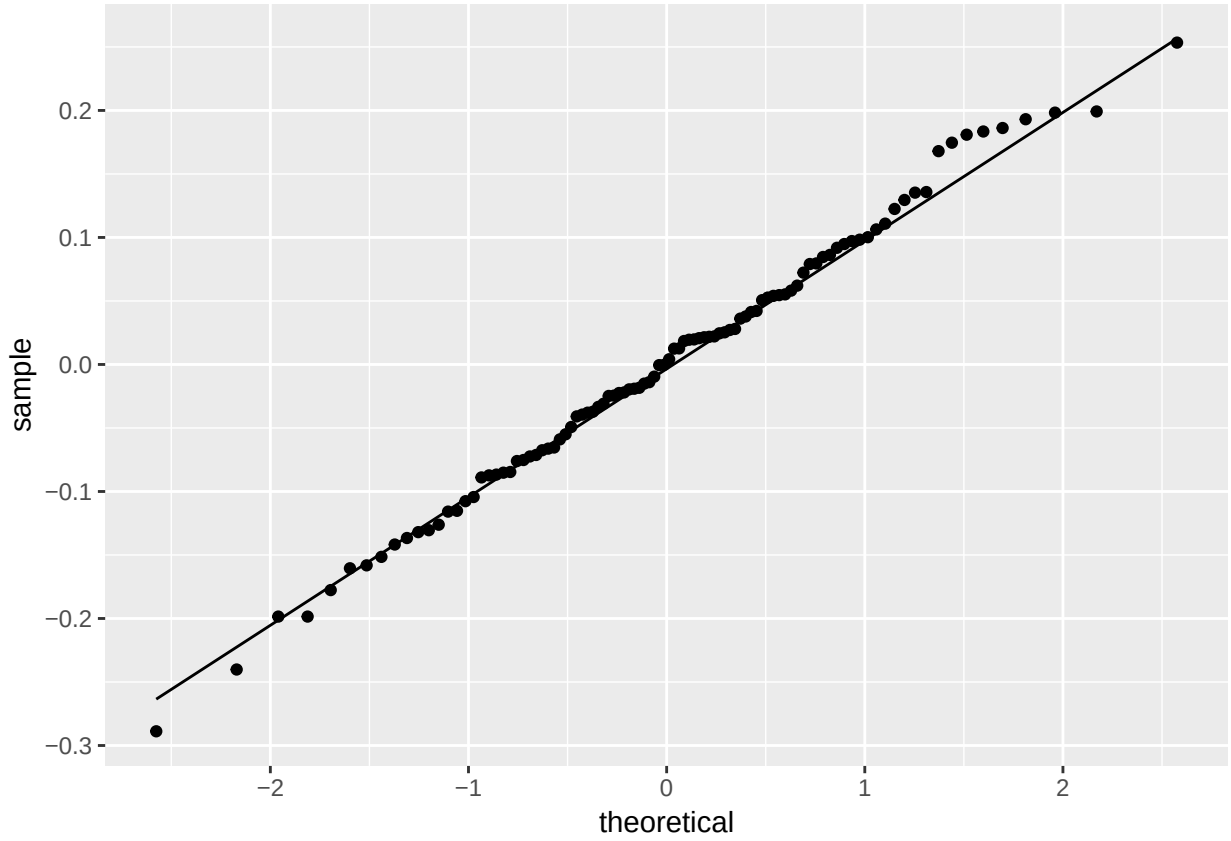
2. Function execution.

- Generate some data with the first function. Use 4 predictors (you can choose n and the noise SD yourself).
- Estimate the model with the second function. And produce the plots.
- Create a table which shows the coefficients, their standard errors, and p-values. You must use the `knitr::kable` function to do this. Print only 2 significant digits. Hint: there is a way to extract all of this information easily from the `lm` output.

```
# I like to establish my constants first.
n = 100
p = 4
sig = 0.1

# Create data. Four predictors.
dat <- generate.data(n, p, sig)

# This is not necessary, I just like having a generalized mechanism for sytems where the
# inputs are changed, e.g., now I only need to change n, p, and sig at top.
# Though I wish I could think of something more elegant than nested pastes...
# Anyway, this just creates the formula based on number of predictors.
formula <- as.formula(paste('y ~', paste(names(dat)[2:(p+1)], collapse = " + ")))

mdl <- estimate.and.plot(formula, dat)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```


X1
X2
X3
X4
resids
value


sample
theoretical

```
knitr::kable(summary(mdl)$coef, digits = 2)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.99	0.01	264.48	0
X1	4.00	0.01	315.91	0
X2	3.01	0.01	266.42	0
X3	2.00	0.01	145.62	0
X4	1.01	0.01	87.52	0

3. Linear models basics review.

Let's see if you can use your regression experience from previous courses. A dataset has been provided, *cars04.csv*. It is in the data folder of homeworks. The dataset describes various vehicles from 2004 (it is old I know...). The following is a brief description of the variables.

- **MSRP**: the Manufacturer Suggested Retail Price of the vehicle.
- **Engine**: the size of the vehicle engine in liters.
- **HP**: the measured horsepower of the vehicle.
- **HMPG**: the EPA rating of the Highway Miles Per Gallon of the vehicle.
- **Weight**: the weight of the vehicle in thousands of pounds.

1. Use the `lm` function to estimate the linear model of MSRP on the four predictor variables. Produce a table summarizing the output.

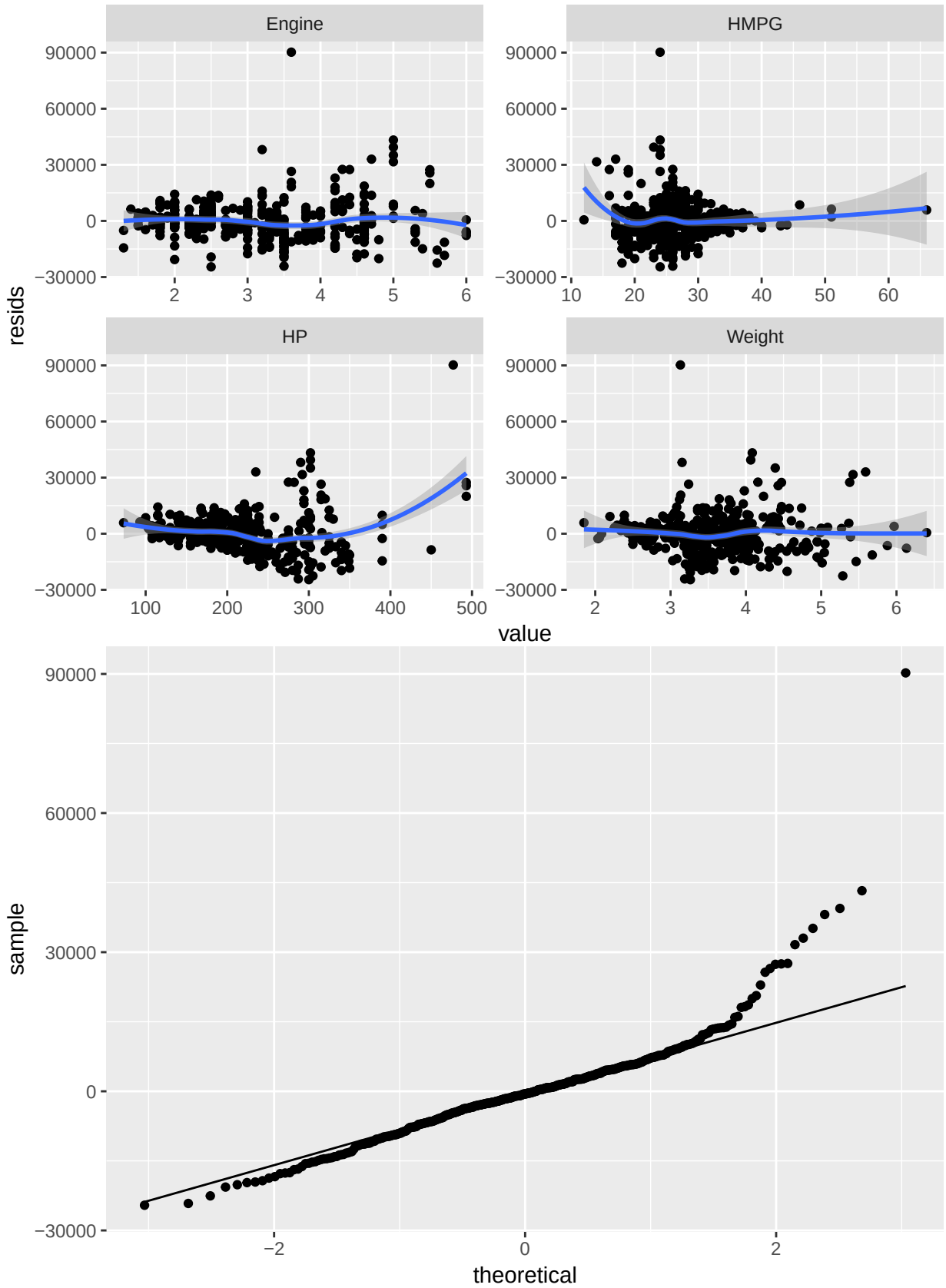
```
# Read data, then clean it of NAs

cars <- read.csv('https://raw.githubusercontent.com/STAT-S432SP2019/homeworks/master/data/cars04.csv')
cars <- cars[complete.cases(cars),]

# Formula for model.
form1 <- as.formula('MSRP ~ Engine + HP + HMPG + Weight')
# First cars model.
car.mdl1 <- lm(form1, cars)

# What a nice table.
coef1 <- summary(car.mdl1)$coef
knitr::kable(coef1, digits = 2)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-34805	8182	-4.25	0.00
Engine	-2313	1069	-2.16	0.03
HP	276	12	22.69	0.00
HMPG	470	157	2.99	0.00
Weight	841	1493	0.56	0.57

2. Make plots of the residuals against each predictor. Make a qq-plot of the residuals. Discuss what you see. Does the assumption of "normally distributed residuals" appear to be satisfied?

```
car.mdl1 <- estimate.and.plot(form1, cars)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



In the residual plots, the engine variable matches the asssumption on the residuals: there is a mostly even

spread across the horizontal axis and the mean value of residuals does not deviate from 0 in any noticeable way. The other plots present more clear violations of the residuals. HMPG, HP and Weight all indicate violations of the homoskedacity assumption. For HMPG, the residuals start with a large variability and then the variability decreases substantially. HP and Weight show residuals that start with low variability and then the variability increases. This is a larger issue in HP versus the residuals.

It should noted that across all plots there is a clear outlier that is the largest residual at about $90,000.

The Normality assumptions seems to hold pretty well for the lower tail of the QQ-Plot, but is clearly violated in the upper tail of the plot as evidenced by the deviation from the line.

3. Interpret the estimated coefficient on HMPG. Find and interpret a 90% confidence interval for β_{HMPG}. Test, with $\alpha = 0.05$, whether or not $\beta_{HMPG} = 0$. State your conclusion in the context of the problem.

The coefficient for HMPG is 469.65. This indicates, under the assumption we have a good model, that we would expect the average increase in price to be about $469.65 for each mile more a car can go on one gallon of gas during highway driving if all other predictors are held constant.

The confidence interval was calculated under the assumption that are residuals are 'well behaved', which they are not. A 90% confidence interval indicates that the coefficient for HMPG should be between 211.09 and 728.22. We would be 90% confident that the true increase in average price is between 211.09 and 728.22 for every 1 unit increase in HMPG if all other factors are held constant.

For the hypothesis test of the HMPG coefficient, we conclude that β_{HMPG} differs from 0 in a significant manner because the p-value ≈ 0 is less than any reasonable α value. Therefore, HMPG is a statistically significant predictor of MSRP, given that all other factors are already in the assumed $MSRP$ model.

4. Someone suggests that there is an interaction between the engine size and horsepower. Add this interaction to the model reinterpret the effect of HMPG on MSRP.

An interaction term for engine size and horsepower was added.

```
# Formula for model.
form2 <- as.formula('MSRP ~ Engine + HP + HMPG + Weight + Engine*HP')
# 2nd cars model.
car.mdl2 <- lm(form2, cars)

# Table for 2nd model.
coef2<- summary(car.mdl2)$coef
knitr::kable(coef2, digits = 2)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-7039	10384.7	-0.68	0.50
Engine	-9328	1968.8	-4.74	0.00
HP	173	27.2	6.35	0.00
HMPG	268	161.0	1.67	0.10
Weight	1118	1464.6	0.76	0.45
Engine:HP	28	6.8	4.21	0.00

Assuming that all other factors are held constant, this model indicates that now the average price would increase by $268.3 for every unit increase in HMPG. This seems fairly different from our previous model which tells us that there is some correlation between our predictor variables. Heavier/larger engine vehicles probably have lower HMPG.

5. Someone suggests that it would be better to use the log of MSRP. Repeat steps 1 to 3 with this change.

   Use the `lm` function to estimate the linear model of MSRP on the four predictor variables. Produce a table summarizing the output.

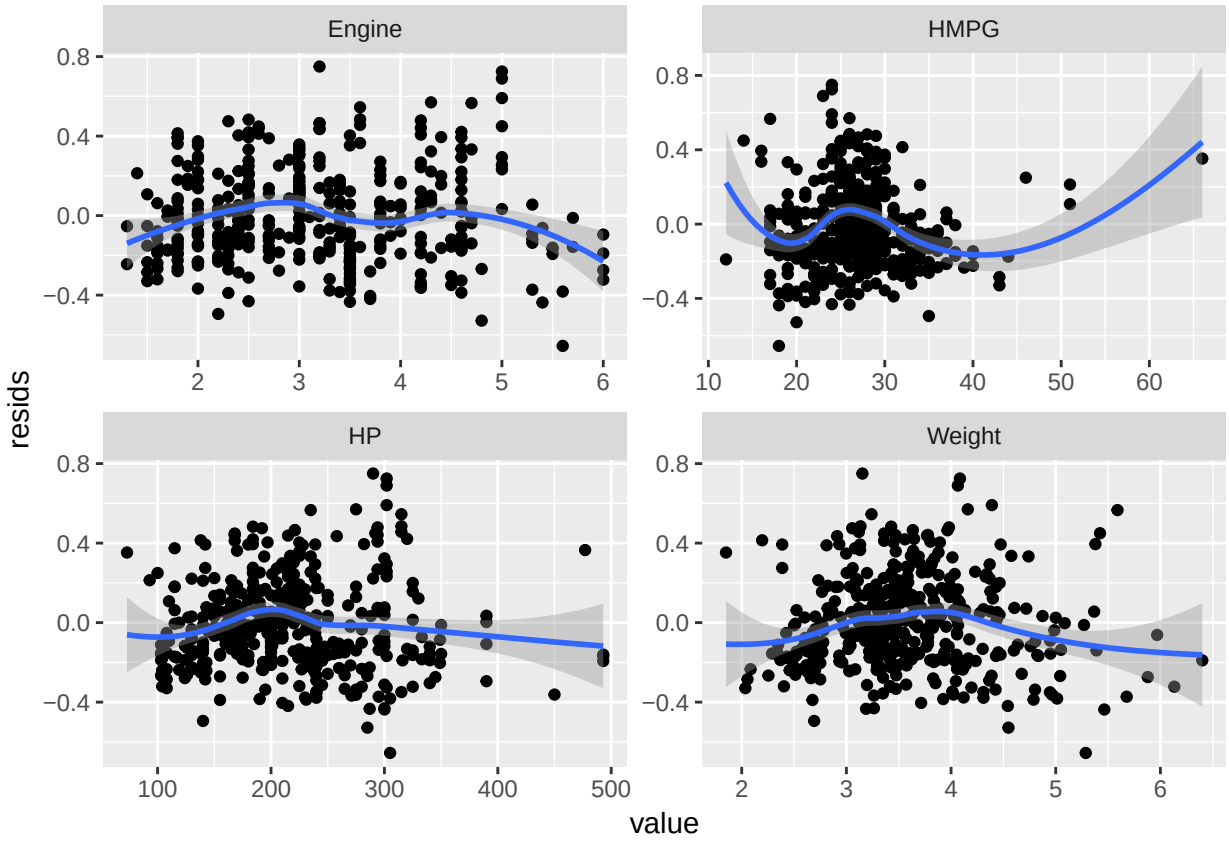
```r
options(digits = 3) # Coefficient values are small. More digit

# Formula for model.
form3 <- as.formula('log(MSRP) ~ Engine + HP + HMPG + Weight')
# First cars model.
car.mdl3 <- lm(form3, cars)

# What a nice table.
coef3 <- summary(car.mdl3)$coef
knitr::kable(coef3, digits = 3)
```

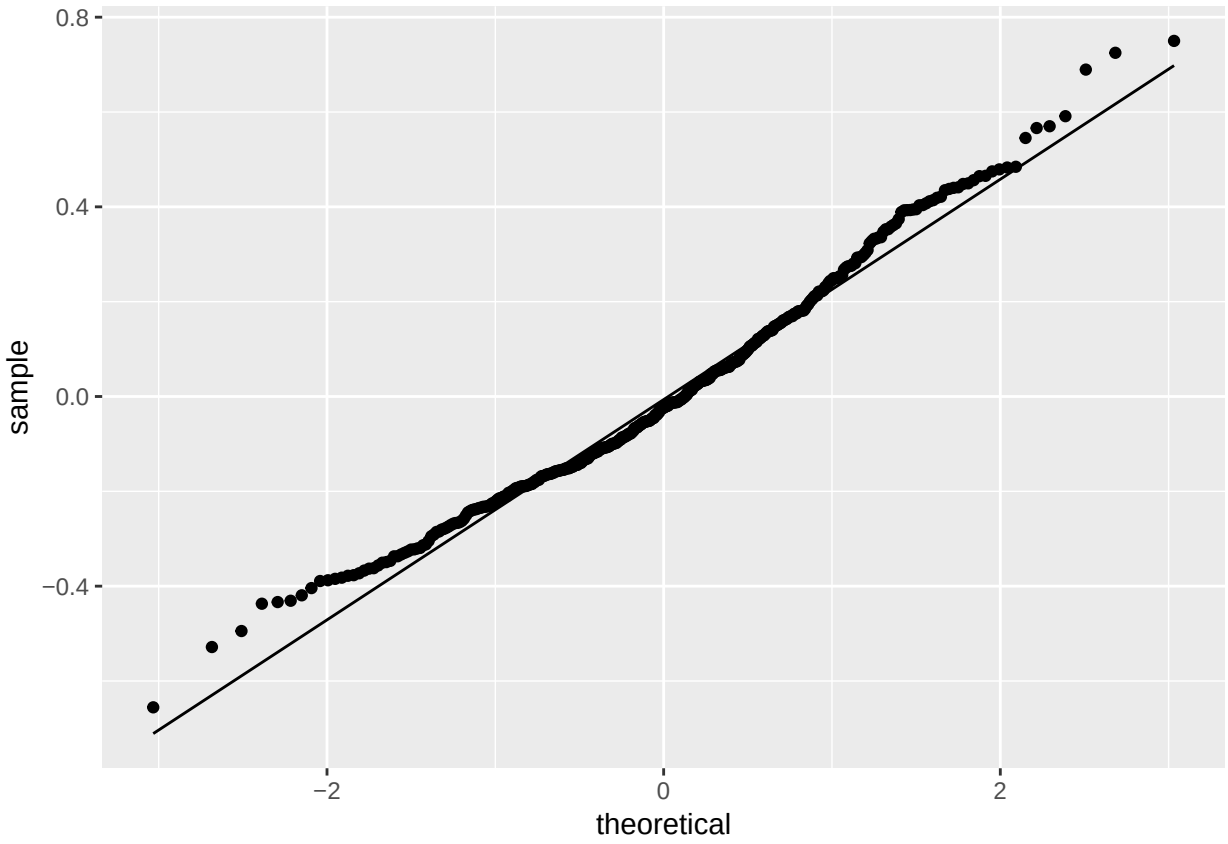
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	8.394	0.180	46.54	0.000
Engine	-0.078	0.024	-3.31	0.001
HP	0.006	0.000	23.62	0.000
HMPG	0.008	0.003	2.28	0.023
Weight	0.155	0.033	4.71	0.000

Make plots of the residuals against each predictor. Make a qq-plot of the residuals. Discuss what you see. Does the assumption of "normally distributed residuals" appear to be satisfied?

```r
car.mdl3 <- estimate.and.plot(form3, cars)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```





Interpret the estimated coefficient on HMPG. Find and interpret a 90% confidence interval for β_{HMPG}. Test, with $\alpha = 0.05$, whether or not $\beta_{HMPG} = 0$. State your conclusion in the context of the problem.

The coefficient for HMPG is 0.01. This indicates, under the assumption we have the correct model, that we would expect the average % increase in price to be about $100 \cdot \%(e^{\beta_{HMPG}} - 1) = 0.79\%$ for each mile more a car can go on one gallon of gas during highway driving if all other predictors are held constant.

The confidence interval was calculated under the assumption that are residuals are 'well behaved', which they are not. A 90% confidence interval indicates that the coefficient for HMPG should be between 0 and 0.01. We would be 90% confident that the true % increase in average price is between 0.22% and 1.37 for every 1 unit increase in HMPG if all other factors are held constant.

For the hypothesis test of the HMPG coefficient, we conclude that β_{HMPG} differs from 0 in a significant manner because the p-value = 0 is less than any reasonable α value. Therefore, HMPG is a statistically significant predictor of MSRP, given that all other factors are already in the assumed $\log(MSRP)$ model.