Summary: Tractor Example

ECO 6416

2022-12-02

Contents

1 Tractor Data Description **2**

2 Bad Practice **2**
2.1 Assumption Testing 3

3 The Proper Practice **5**
3.1 Potential Ideas 5
3.2 Splitting the Data 5
3.3 Summary Statistics 5
3.4 Plots 6
3.5 Data Transformation 7
3.6 Models 9
3.7 Model 2: Quadratic Relationship 11
3.8 Performance 13

Here are all the packages needed to get started.

```
library(readxl) # reading in excel file
library(car) # for vif function
```

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

```
library(plotly) # for interactive visualizations
```

```
## Loading required package: ggplot2

##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(gt) # for better looking tables
library(gtsummary) # for better summary statistics
```

1 Tractor Data Description

The following data is of tractor sales and the characteristics of each tractor sold. It consists of 276 observations and 12 variables (4 quantitative and 8 categorical).

- saleprice: The selling price of the tractor (in dollars)
- horsepower: Horsepower of the engine
- age: Age of the tractor sold
- enginehours: Total running hours on the engine
- diesel: Dummy varaible indicating whether or not the fuel used is diesel
- fwd: Dummy variable indicating whether or not the tractor is forward or rear wheel drive
- manual: Dummy variable indicating whether or not it is manual transmission or automatic
- johndeere: Dummy variable indicating if the manufacter is John Deere
- cab: Dummy variable indicating if there is a saftey cab
- seasons: Indicator for spring, summer, winter with the default being fall

We can pull in the data and look at the data:

```
tractor <- read_xlsx("../Data/TractorRaw.xlsx")

gt(head(tractor)) # the gt function only makes it look nicer
```

saleprice	horsepower	age	enghours	diesel	fwd	manual	johndeere	cab	spring	summer	winter
16100	105	23	1800	1	0	1	0	1	0	1	0
10000	75	12	3730	1	0	1	0	1	0	0	0
25100	90	6	1757	1	1	1	0	1	1	0	0
15100	47	8	2500	1	1	1	0	1	0	1	0
25100	95	5	2360	1	1	1	0	1	1	0	0
10250	46	17	1021	1	0	1	0	1	0	1	0

2 Bad Practice

If we ignore all our training, we may just run a model without considering the center, shape, and spread of all the variables.

By simply running the model, we are also skipping the first step in regression analysis *reviewing literature and develop a theoretical model.* This ignores the possibility that there may be non-linear relationships between the independent and dependent variables.

```
bad_model <- lm(saleprice ~., data = tractor)

summary(bad_model)
```

```
##
## Call:
## lm(formula = saleprice ~ ., data = tractor)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -48532  -6089   -645   6263  92806
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13015.7894  4468.2593   2.913  0.00389 **
## horsepower    226.5840    15.1670  14.939  < 2e-16 ***
```

```
## age          -699.7279   146.8462  -4.765 3.12e-06 ***
## enghours       -1.9344     0.3934  -4.917 1.55e-06 ***
## diesel        444.3901  4000.6502   0.111  0.91164
## fwd          1491.0701  2413.9374   0.618  0.53731
## manual      -4214.1008  2550.8076  -1.652  0.09971 .
## johndeere   13709.8757  2972.6862   4.612 6.22e-06 ***
## cab          8072.0643  2597.6376   3.107  0.00209 **
## spring      -1815.2076  2672.9042  -0.679  0.49766
## summer      -4923.8739  2620.8553  -1.879  0.06138 .
## winter      -1579.6222  2933.8039  -0.538  0.59074
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16380 on 264 degrees of freedom
## Multiple R-squared:  0.6599, Adjusted R-squared:  0.6457
## F-statistic: 46.57 on 11 and 264 DF,  p-value: < 2.2e-16
```

```
# or (fancy output)

tbl_regression(bad_model,
               estimate_fun =  ~style_sigfig(.x, digits = 4)) %>% as_gt() %>%
  gt::tab_source_note(gt::md(paste0("Adjusted R-Squared: ",round(summary(bad_model)$adj.r.squared* 100,
```

Characteristic	Beta	95% CI[1]	p-value
horsepower	226.6	196.7, 256.4	<0.001
age	-699.7	-988.9, -410.6	<0.001
enghours	-1.934	-2.709, -1.160	<0.001
diesel	444.4	-7,433, 8,322	>0.9
fwd	1,491	-3,262, 6,244	0.5
manual	-4,214	-9,237, 808.4	0.10
johndeere	13,710	7,857, 19,563	<0.001
cab	8,072	2,957, 13,187	0.002
spring	-1,815	-7,078, 3,448	0.5
summer	-4,924	-10,084, 236.6	0.061
winter	-1,580	-7,356, 4,197	0.6

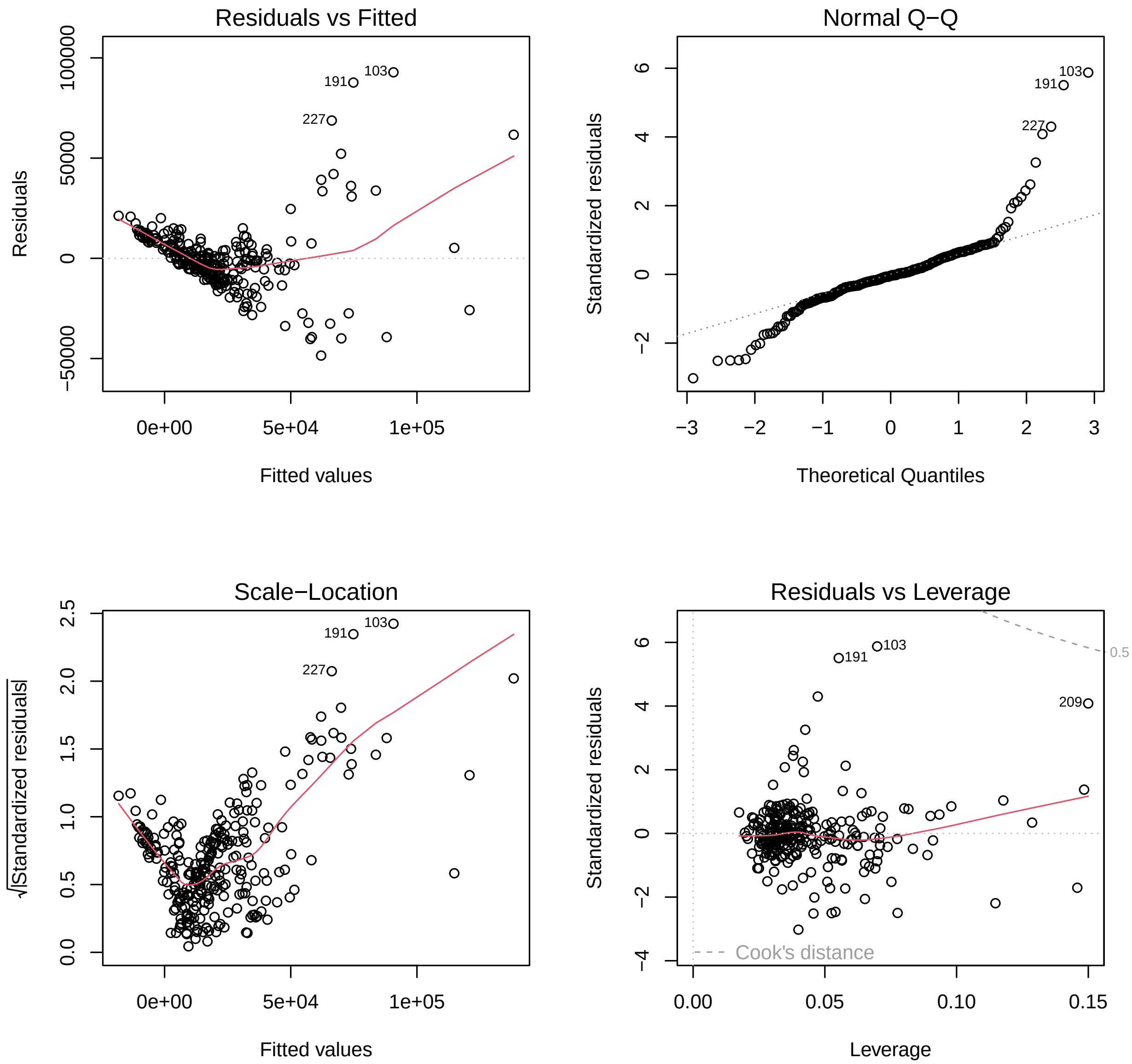
[1] CI = Confidence Interval
Adjusted R-Squared: 64.57%

One thing to note here. Our model states that John Deere tractors cost $13,710 more than the same tractor with a different name. To be thorough, I decided to check online for some tractors with similar characteristics. The true gap between brands was much smaller.

2.1 Assumption Testing

When we ignore the proper steps, we saw how our model is over-valuing John Deere tractors. We know this is the case because we mis-specified the model. The plots below also show that some of the Gauss-Markov assumptions have been violated.

```
par(mfrow=c(2,2))
plot(bad_model)
```



In this example, the first plot titled "Residuals vs. Fitted," you should not see a true pattern. In this case, since there is a non-linear relationship, you've already violated a classical assumption.

The second plot titled "Normal Q-Q" shows the assumption of a normally distributed dependent variable for a fixed set of predictors. If this were a 45-degree line upwards, we could verify this. Unfortunately we do not have it in this case.

The third plot titled "Scale-Location" checks for homoskedasticity. If this assumption were not violated, you'd see random points around a horizontal line. In this case, it is upwards sloping, so you can see there is a "fanning out" effect.

The last plot "Residuals vs. Leverage" keeps an eye out for regression outliers, influential observations, and high leverage points. (Do not worry about this last plot).

3 The Proper Practice

If you were doing this on your own and didn't have a dataset, you would need to think about what variables could explain the variation in tractor prices. Since the data was already collected for you, you need to think about the relationships between the dependent varaible and the independent variables a.k.a *reviewing literature and develop a theoretical model.*

3.1 Potential Ideas

Here are some thoughts that you may consider when looking at the relationships between independent and dependent variables.

- Quadratic relationship between horsepower and sales price
  - Horsepower improves performance up to a limit, then extra power does not add value, only consumes more fuel.
- Logarithmic relationship between horsepower and sales price
  - Horsepower improves performance more in the lower horsepower range than in the higher horsepower range. There are still some benefits, but not nearly as much.

You are not bound to only create variables, you can drop ones as well such as seasonality.

You could continue this with all the variables to test out different relationships. For this example, now that we've created two different models, we can start building.

3.2 Splitting the Data

First we need to split the data into testing and training data. Let's pull 10 observations

```
set.seed(123457)
index <- sample(seq_len(nrow(tractor)), size = 10)

train <- tractor[-index,]
test <- tractor[index,]
```

3.3 Summary Statistics

```
summary(train)
```

```
##    saleprice        horsepower          age           enghours      
##  Min.   :  1500   Min.   : 16.00   Min.   : 2.00   Min.   :    1.0  
##  1st Qu.:  7562   1st Qu.: 47.25   1st Qu.: 7.00   1st Qu.:  763.8  
##  Median : 11550   Median : 80.00   Median :14.50   Median : 2398.0  
##  Mean   : 20521   Mean   :100.03   Mean   :15.89   Mean   : 3538.6  
##  3rd Qu.: 20550   3rd Qu.:108.00   3rd Qu.:24.00   3rd Qu.: 5429.2  
##  Max.   :200000   Max.   :535.00   Max.   :33.00   Max.   :18744.0  
##      diesel           fwd             manual        johndeere     
##  Min.   :0.000   Min.   :0.0000   Min.   :0.000   Min.   :0.0000  
##  1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.0000  
##  Median :1.000   Median :1.0000   Median :1.000   Median :0.0000  
##  Mean   :0.906   Mean   :0.5677   Mean   :0.703   Mean   :0.1391  
##  3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:0.0000  
##  Max.   :1.000   Max.   :1.0000   Max.   :1.000   Max.   :1.0000  
##       cab             spring           summer           winter      
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000  
##  Median :1.0000   Median :0.0000   Median :0.0000   Median :0.0000  
```

```
##   Mean   :0.5338   Mean   :0.2218   Mean   :0.2331   Mean   :0.1692
##   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000
##   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
```

```
# or

train %>%
  tbl_summary(statistic = list(all_continuous() ~ c("{mean} ({sd})",
                                                     "{median} ({p25}, {p75})",
                                                     "{min}, {max}"),
                               all_categorical() ~ "{n} / {N} ({p}%)"),
              type = all_continuous() ~ "continuous2"
  )
```

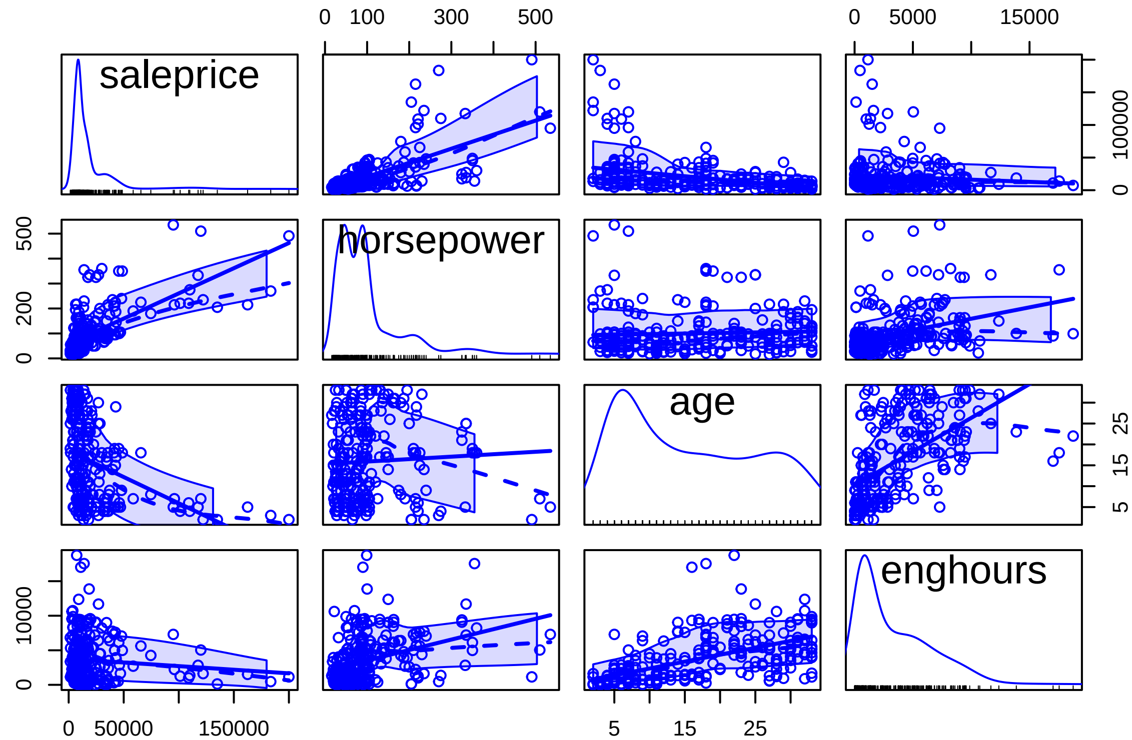
Characteristic	**N = 266**
saleprice	
Mean (SD)	20,521 (27,480)
Median (IQR)	11,550 (7,562, 20,550)
Range	1,500, 200,000
horsepower	
Mean (SD)	100 (84)
Median (IQR)	80 (47, 108)
Range	16, 535
age	
Mean (SD)	16 (10)
Median (IQR)	14 (7, 24)
Range	2, 33
enghours	
Mean (SD)	3,539 (3,415)
Median (IQR)	2,398 (764, 5,429)
Range	1, 18,744
diesel	241 / 266 (91%)
fwd	151 / 266 (57%)
manual	187 / 266 (70%)
johndeere	37 / 266 (14%)
cab	142 / 266 (53%)
spring	59 / 266 (22%)
summer	62 / 266 (23%)
winter	45 / 266 (17%)

One thing that is obvious here is that our dependent variable is skewed to the right. The mean is about 9 thousand dollars higher than the median and the standard deviation is high, and the range is from 1.5k to 200k. We may have outliers in our data.

3.4 Plots

Since we can only look at the quantitative variables in a scatter-plot and histogram, we are going to exclude the others.

```r
scatterplotMatrix(train[,1:4])
```



From here you can see some non-linear relationships and non-normally distributed variables.

3.5 Data Transformation

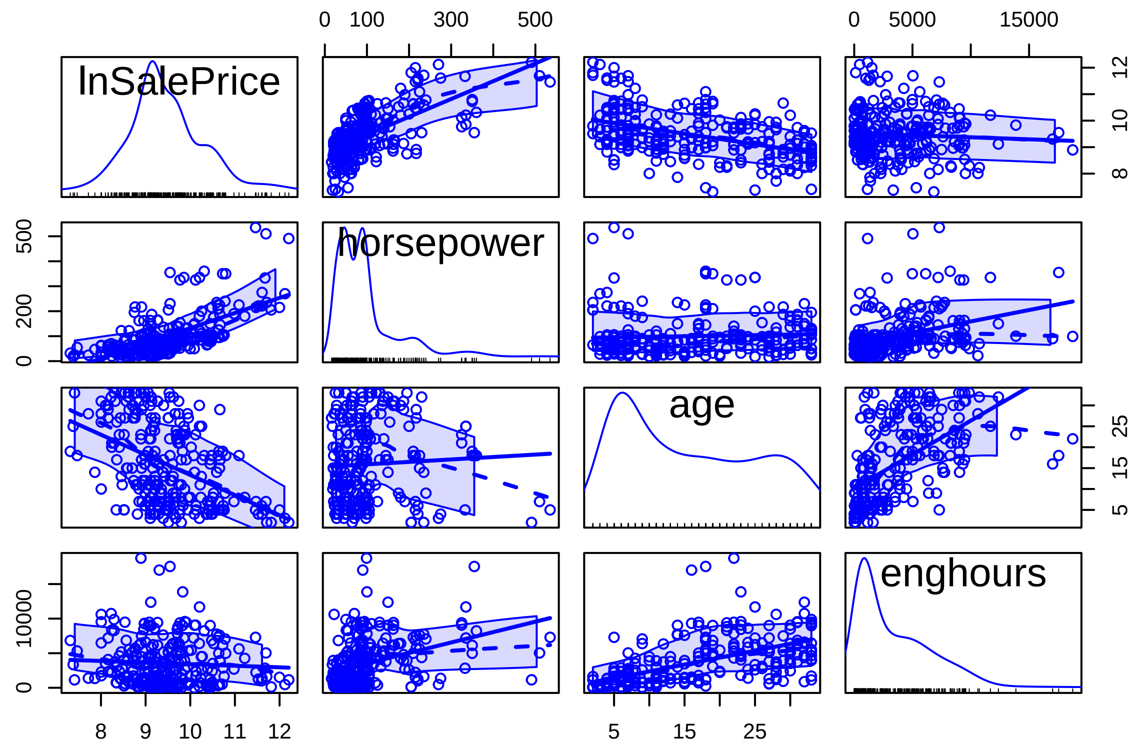
Let's take the natural logarithm of sales. Taking logs will bring outliers closer to the other tractor prices.

```r
par(mfrow=c(1,2))
hist(train$saleprice) #before

train$lnSalePrice <- log(train$saleprice)

hist(train$lnSalePrice) #after
```

That is much better. We now have something closer to a normal distribution.

3.5.1 Plotting the relationships After Transformation

```
scatterplotMatrix(train[,c(13,2,3,4)]) # grabbing lnSalesPrice
```



We can still see some nonlinearity between horsepower and sales price. It is hard to determine if it is logarithmic or quadratic.

```
train$lnHorsepower <- log(train$horsepower)
train$horsepowerSquared <- train$horsepower^2
```

We could look at engine hours as well and continue forward, for the sake of this document, I am going to skip that part.

3.6 Models

Let's build some models and look at the regression coefficients.

3.6.1 Model 1: Horsepower with a logaritmic shape

```
model_1 <- lm(lnSalePrice ~., data = train[,c(13,14,3:12)] ) #pulling only columns I want

summary(model_1)
```

```
##
## Call:
## lm(formula = lnSalePrice ~ ., data = train[, c(13, 14, 3:12)])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.75705 -0.22648  0.02128  0.25572  0.76159
##
```

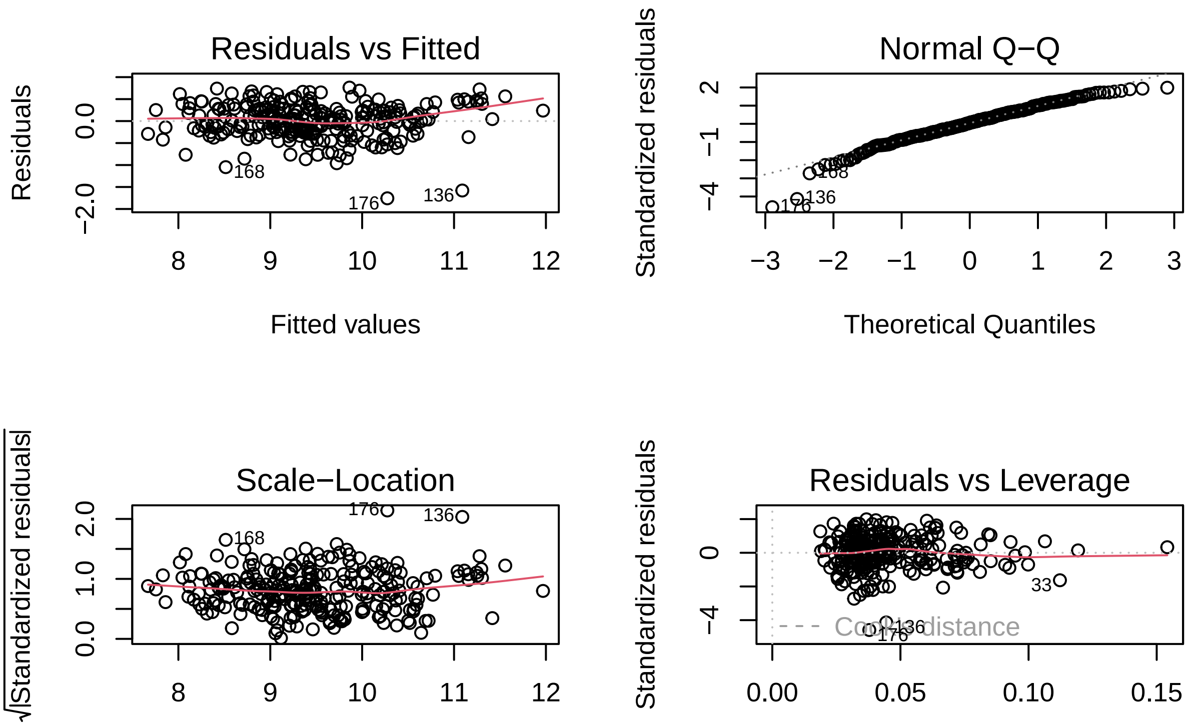
```
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.385e+00  2.034e-01  31.396  < 2e-16 ***
## lnHorsepower  7.654e-01  5.085e-02  15.053  < 2e-16 ***
## age          -2.928e-02  3.583e-03  -8.173 1.44e-14 ***
## enghours     -4.461e-05  9.625e-06  -4.635 5.72e-06 ***
## diesel        1.099e-01  9.765e-02   1.126  0.26123
## fwd           3.399e-01  5.885e-02   5.777 2.22e-08 ***
## manual       -2.068e-01  6.277e-02  -3.294  0.00113 **
## johndeere     3.438e-01  7.267e-02   4.731 3.71e-06 ***
## cab           4.094e-01  7.050e-02   5.808 1.89e-08 ***
## spring       -4.581e-02  6.475e-02  -0.707  0.47997
## summer       -7.509e-02  6.345e-02  -1.184  0.23771
## winter        4.276e-02  7.137e-02   0.599  0.54965
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3906 on 254 degrees of freedom
## Multiple R-squared:  0.8136, Adjusted R-squared:  0.8055
## F-statistic: 100.8 on 11 and 254 DF,  p-value: < 2.2e-16
```

```r
# or

tbl_regression(model_1,
               estimate_fun =  ~style_sigfig(.x, digits = 4)) %>% as_gt() %>%
  gt::tab_source_note(gt::md(paste0("Adjusted R-Squared: ",round(summary(model_1)$adj.r.squared* 100,di
```

Characteristic	**Beta**	**95% CI**[1]	**p-value**
lnHorsepower	0.7654	0.6653, 0.8655	<0.001
age	-0.0293	-0.0363, -0.0222	<0.001
enghours	0.0000	-0.0001, 0.0000	<0.001
diesel	0.1099	-0.0824, 0.3022	0.3
fwd	0.3399	0.2240, 0.4558	<0.001
manual	-0.2068	-0.3304, -0.0832	0.001
johndeere	0.3438	0.2007, 0.4869	<0.001
cab	0.4094	0.2706, 0.5483	<0.001
spring	-0.0458	-0.1733, 0.0817	0.5
summer	-0.0751	-0.2000, 0.0499	0.2
winter	0.0428	-0.0978, 0.1833	0.5

[1]CI = Confidence Interval
Adjusted R-Squared: 80.55%

```r
par(mfrow=c(2,2))
plot(model_1)
```



These are improvements to these assumptions.

3.7 Model 2: Quadratic Relationship

```
model_2 <- lm(lnSalePrice ~., data = train[,c(13,2:12,15)] ) #pulling only columns I want

summary(model_2)
```

```
##
## Call:
## lm(formula = lnSalePrice ~ ., data = train[, c(13, 2:12, 15)])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.69027 -0.23045  0.05296  0.29453  0.74126
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       8.737e+00  1.139e-01  76.738  < 2e-16 ***
## horsepower        1.111e-02  1.097e-03  10.125  < 2e-16 ***
## age              -3.248e-02  3.704e-03  -8.769 2.70e-16 ***
## enghours         -4.103e-05  9.836e-06  -4.171 4.17e-05 ***
## diesel            2.203e-01  1.001e-01   2.200   0.0287 *
## fwd               2.611e-01  6.074e-02   4.298 2.46e-05 ***
## manual           -1.474e-01  6.412e-02  -2.299   0.0223 *
## johndeere         3.377e-01  7.459e-02   4.528 9.18e-06 ***
```

```
## cab                4.848e-01  7.212e-02   6.723 1.18e-10 ***
## spring            -7.248e-02  6.655e-02  -1.089   0.2771
## summer            -6.108e-02  6.511e-02  -0.938   0.3491
## winter             2.329e-02  7.345e-02   0.317   0.7514
## horsepowerSquared -1.393e-05  2.302e-06  -6.051 5.15e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4006 on 253 degrees of freedom
## Multiple R-squared:  0.8047, Adjusted R-squared:  0.7954
## F-statistic: 86.85 on 12 and 253 DF,  p-value: < 2.2e-16
```

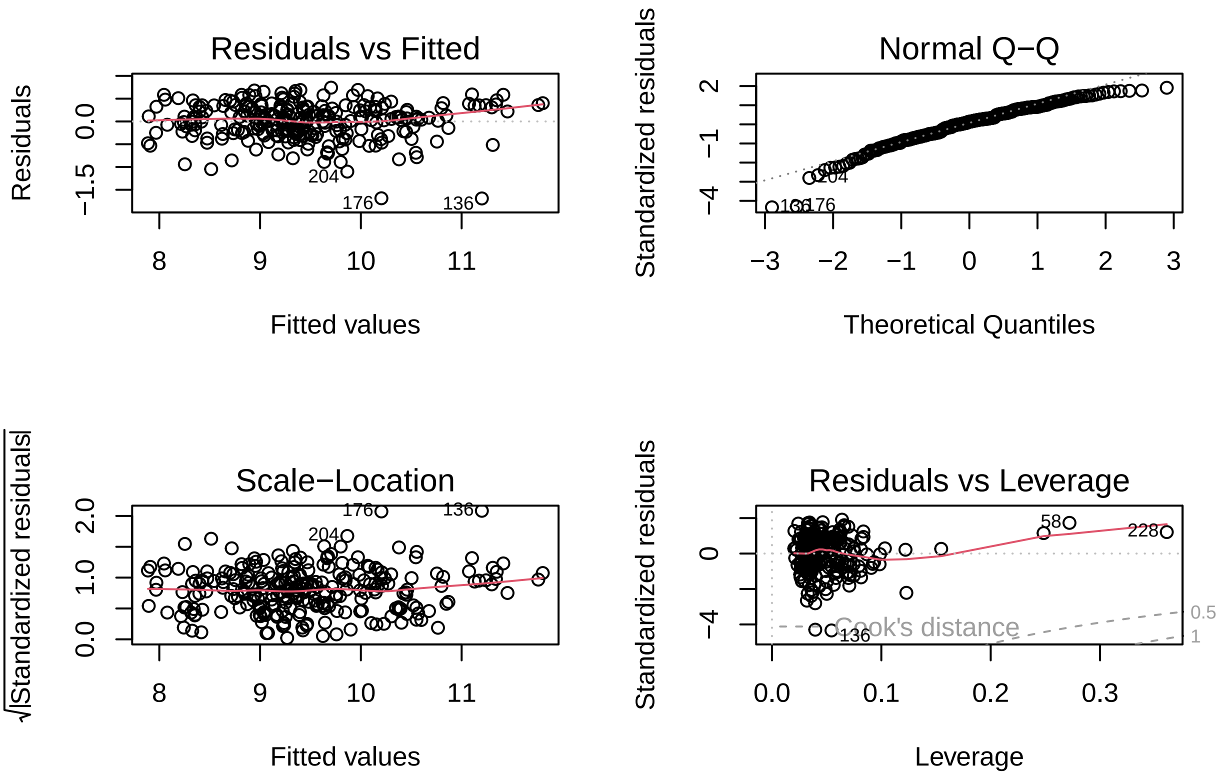
```
# or

tbl_regression(model_2,
               estimate_fun =  ~style_sigfig(.x, digits = 4)) %>% as_gt() %>%
  gt::tab_source_note(gt::md(paste0("Adjusted R-Squared: ",round(summary(model_2)$adj.r.squared* 100,di
```

Characteristic	**Beta**	**95% CI**[1]	**p-value**
horsepower	0.0111	0.0089, 0.0133	<0.001
age	-0.0325	-0.0398, -0.0252	<0.001
enghours	0.0000	-0.0001, 0.0000	<0.001
diesel	0.2203	0.0231, 0.4175	0.029
fwd	0.2611	0.1415, 0.3807	<0.001
manual	-0.1474	-0.2737, -0.0211	0.022
johndeere	0.3377	0.1908, 0.4846	<0.001
cab	0.4848	0.3428, 0.6268	<0.001
spring	-0.0725	-0.2035, 0.0586	0.3
summer	-0.0611	-0.1893, 0.0672	0.3
winter	0.0233	-0.1214, 0.1679	0.8
horsepowerSquared	0.0000	0.0000, 0.0000	<0.001

[1]CI = Confidence Interval
Adjusted R-Squared: 79.54%

Since the coefficient of horsepower is so small, it is hard to tell that it is showing a quadratic relationship.

```
par(mfrow=c(2,2))
plot(model_2)
```



Comparing to the base model, these are improvements to these assumptions.

3.8 Performance

First things first, we need to include the transformations to our dataset so that we can use them in our predictions.

```
test$lnSalePrice <- log(test$saleprice)
test$lnHorsepower <- log(test$horsepower)
test$horsepowerSquared <- test$horsepower^2
```

```
test$bad_model_pred <- predict(bad_model, newdata = test)

test$model_1_pred <- predict(model_1,newdata = test) %>% exp()

test$model_2_pred <- predict(model_2,newdata = test) %>% exp()

# Finding the error

test$error_bm <- test$bad_model_pred - test$saleprice

test$error_1 <- test$model_1_pred - test$saleprice

test$error_2 <- test$model_2_pred - test$saleprice
```

3.8.1 Bias

```r
# Bad Model
mean(test$error_bm)
```

```
## [1] 3222.167
```

```r
# Model 1
mean(test$error_1)
```

```
## [1] -2784.369
```

```r
# Model 2
mean(test$error_2)
```

```
## [1] -487.4141
```

3.8.2 MAE

```r
# I decided to create a function to calculate this

mae <- function(error_vector){
  error_vector %>%
  abs() %>%
  mean()
}

# Bad Model
mae(test$error_bm)
```

```
## [1] 10135.71
```

```r
# Model 1
mae(test$error_1)
```

```
## [1] 7863.525
```

```r
# Model 2
mae(test$error_2)
```

```
## [1] 7943.492
```

3.8.3 RMSE

```r
rmse <- function(error_vector){
   error_vector^2 %>%
  mean() %>%
  sqrt()

}

# Bad Model
rmse(test$error_bm)
```

```
## [1] 16057.09
```

```r
# Model 1
rmse(test$error_1)
```

```
## [1] 13281.24
```

```r
# Model 2
rmse(test$error_2)
```

```
## [1] 11689.06
```

3.8.4 MAPE

```r
mape <- function(error_vector, actual_vector){
  (error_vector/actual_vector) %>%
    abs() %>%
    mean()
}

# Bad Model
mape(test$error_bm, test$saleprice)
```

```
## [1] 0.4774086
```

```r
# Model 1
mape(test$error_1, test$saleprice)
```

```
## [1] 0.2723213
```

```r
# Model 2
mape(test$error_2, test$saleprice)
```

```
## [1] 0.3399796
```

3.8.5 Summary of Performance Metrics

Looking at these three models, the initial model was the worst performing (not surprising). Looking at the other two, the logarithmic relationship has lower bias, MAE, and MAPE. Model 2 has a lower RMSE meaning that there were not large prediction errors. Picking which model would depend on your time preference. If you are looking at the short-run, then Model 2. Model 1 if you are looking at the long-run.