THE UNIVERSITY OF TEXAS AT AUSTIN
McCOMBS SCHOOL OF BUSINESS

Diagnostics & Transformations

Lecture 12

STA 371G

Predicting the fuel economy (miles per gallon) for different car models of the 70s.

Predicting the fuel economy (miles per gallon) for different car models of the 70s.

“LINE” assumptions:

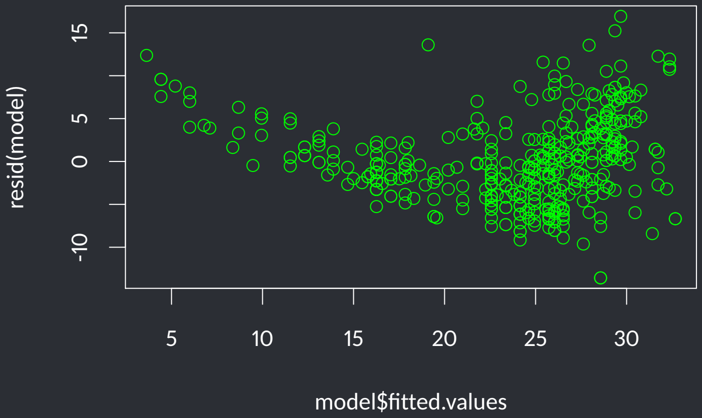
- Linearity
- Independent errors
- Normally distributed errors
- Equal Variance (Homoscedasticity)

Predicting MPG from Horsepower

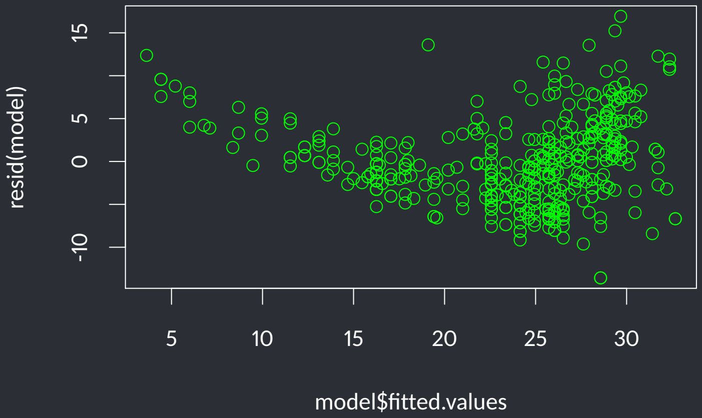
```
model<-lm(MPG ~ HP, data=auto_mpg)
summary(model)
```

```

Call:
lm(formula = MPG ~ HP, data = auto_mpg)

Residuals:
     Min       1Q   Median       3Q      Max 
-13.5710  -3.2592  -0.3435   2.7630  16.9240 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 39.935861   0.717499   55.66   <2e-16 ***
HP          -0.157845   0.006446  -24.49   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.906 on 390 degrees of freedom
Multiple R-squared:  0.6059,Adjusted R-squared:  0.6049 
F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

```r
plot(model$fitted.values, resid(model), col='green')
```



```r
plot(model$fitted.values, resid(model), col='green')
```

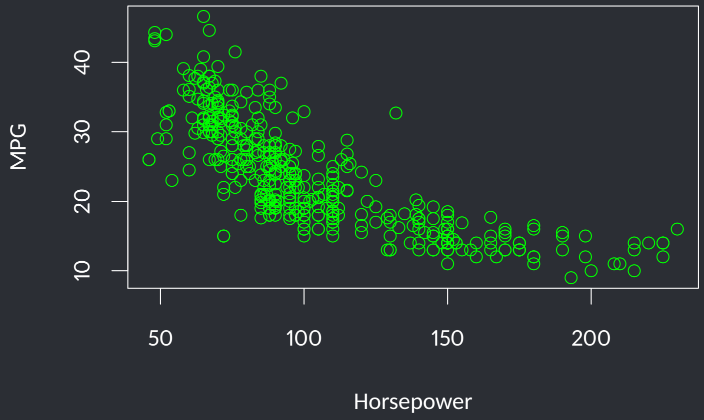


The trend in the residuals implies linearity issues.
The “funnel” implies equal variance issues.

Addressing the linearity issue

The relation between MPG and horsepower does not seem to be linear.

```
plot(auto_mpg$HP, auto_mpg$MPG, col='green',
     xlab='Horsepower', ylab='MPG')
```



Addressing the linearity issue

If we could horizontally shift the data on the far right towards left, the plot would look “more” linear.

Addressing the linearity issue

If we could horizontally shift the data on the far right towards left, the plot would look "more" linear.

Predict the MPG of a car not from the horsepower, but from a "transformation" of the horsepower.

Addressing the linearity issue

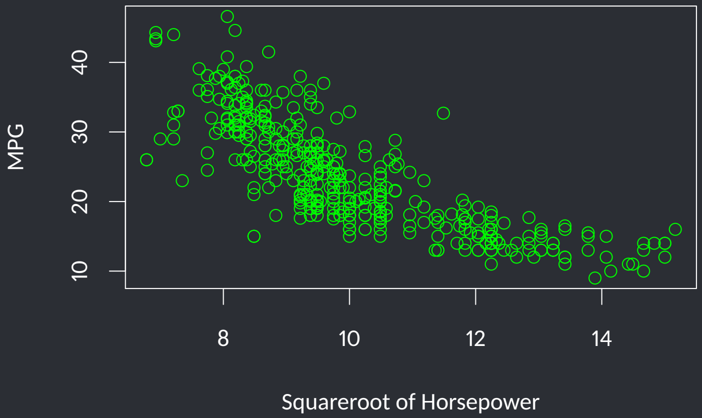
If we could horizontally shift the data on the far right towards left, the plot would look “more” linear.

Predict the MPG of a car not from the horsepower, but from a “transformation” of the horsepower.

For example, the relation between MPG and HP is not linear, the one between MPG and $\sqrt{\text{HP}}$ is!

Addressing the linearity issue

```
auto_mpg$HP_sqrt <- sqrt(auto_mpg$HP)
plot(auto_mpg$HP_sqrt, auto_mpg$MPG, col='green',
     xlab='Squareroot of Horsepower', ylab='MPG')
```



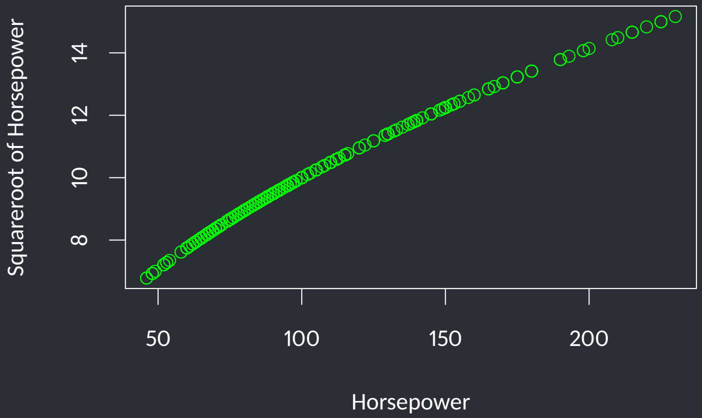
Addressing the linearity issue

It indeed seems a bit better. Notice the change in the range of the horizontal axis.

Addressing the linearity issue

It indeed seems a bit better. Notice the change in the range of the horizontal axis. It has changed from [49,225] to [7,15]. The shift is larger for the data on the far right.

```
plot(auto_mpg$HP, auto_mpg$HP_sqrt, col='green',
    xlab='Horsepower', ylab='Squareroot of Horsepower')
```

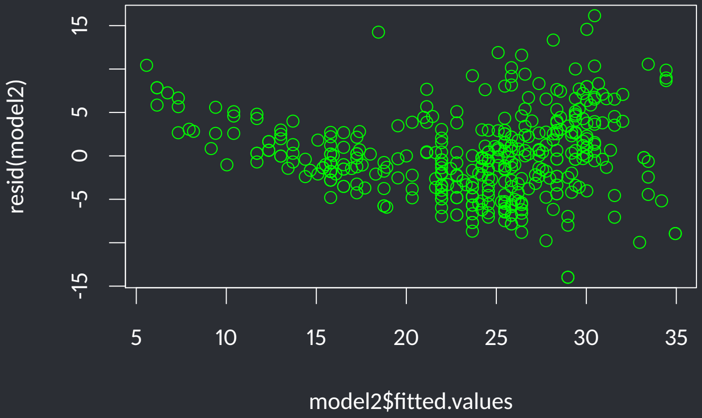


Addressing the linearity issue

```
model2<-lm(MPG ~ HP_sqrt, data=auto_mpg)
summary(model2)
```

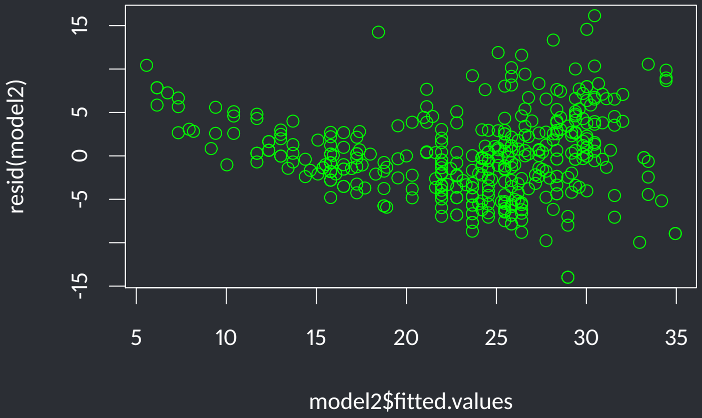
```
Call:
lm(formula = MPG ~ HP_sqrt, data = auto_mpg)

Residuals:
     Min       1Q   Median       3Q      Max
-13.9768  -3.2239  -0.2252   2.6881  16.1411

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   58.705      1.349   43.52   <2e-16 ***
HP_sqrt       -3.503      0.132  -26.54   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.665 on 390 degrees of freedom
Multiple R-squared:  0.6437,Adjusted R-squared:  0.6428
F-statistic: 704.6 on 1 and 390 DF,  p-value: < 2.2e-16
```

```r
plot(model2$fitted.values, resid(model2), col='green')
```



The trend flattened a bit.

```r
plot(model2$fitted.values, resid(model2), col='green')
```



The trend flattened a bit.

Can we do better? Let's try some other transformation.

Logarithmic transformation

One of the most common transformations is the logarithmic transformation with base e (natural logarithm).

Logarithmic transformation

One of the most common transformations is the logarithmic transformation with base e (natural logarithm).

$$e = 2.7182818284\ldots$$

Logarithmic transformation

One of the most common transformations is the logarithmic transformation with base e (natural logarithm).

$$e = 2.7182818284\ldots$$

$$e^2 = 7.389$$

Logarithmic transformation

One of the most common transformations is the logarithmic transformation with base e (natural logarithm).

$$e = 2.7182818284\ldots$$

$$e^2 = 7.389$$

$$\log 7.389 = 2$$

Logarithmic transformation

One of the most common transformations is the logarithmic transformation with base e (natural logarithm).

$$e = 2.7182818284\ldots$$

$$e^2 = 7.389$$

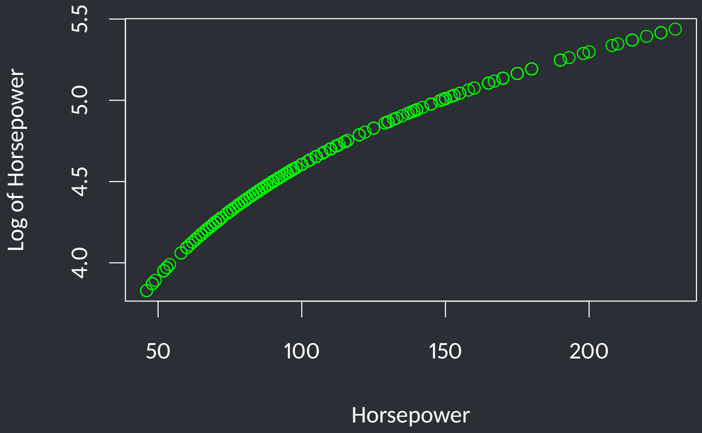
$$\log 7.389 = 2$$

In general:

$$y = e^x \rightarrow \log y = x.$$

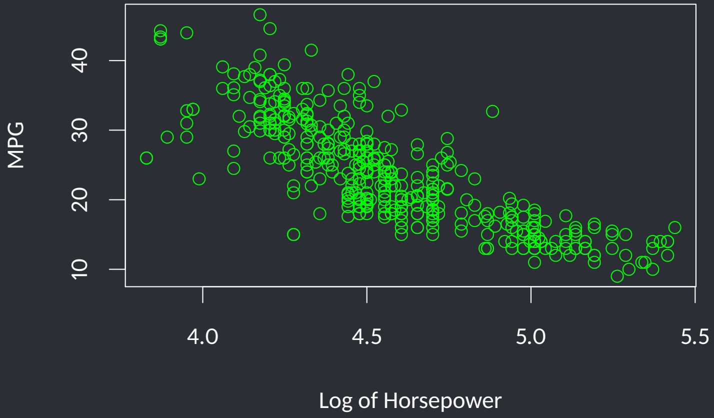
Logarithmic transformation

```
auto_mpg$HP_ln <- log(auto_mpg$HP)
plot(auto_mpg$HP, auto_mpg$HP_ln, col='green',
    xlab='Horsepower', ylab='Log of Horsepower')
```



Logarithmic transformation

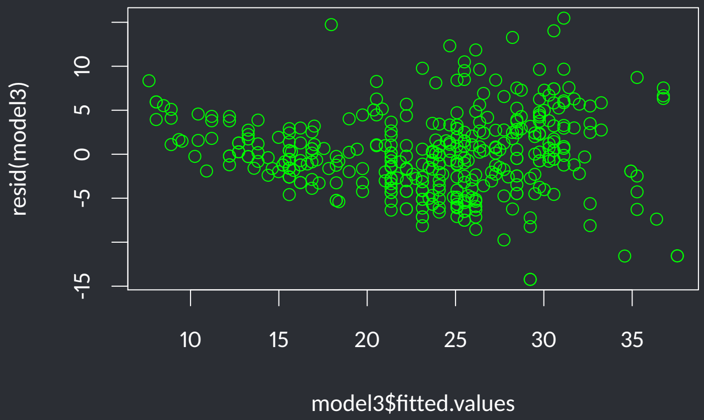
```
plot(auto_mpg$HP_ln, auto_mpg$MPG, col='green',
     xlab='Log of Horsepower', ylab='MPG')
```



```r
model3<-lm(MPG ~ HP_ln, data=auto_mpg)
summary(model3)
```

```
Call:
lm(formula = MPG ~ HP_ln, data = auto_mpg)

Residuals:
     Min       1Q   Median       3Q      Max 
-14.2299  -2.7818  -0.2322   2.6661  15.4695 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 108.6997     3.0496   35.64   <2e-16 ***
HP_ln       -18.5822     0.6629  -28.03   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.501 on 390 degrees of freedom
Multiple R-squared:  0.6683,Adjusted R-squared:  0.6675 
F-statistic: 785.9 on 1 and 390 DF,  p-value: < 2.2e-16
```

```r
plot(model3$fitted.values, resid(model3), col='green')
```



The trend flattened even more.

Transforming a Predictor

It is equivalent to “cutting the distribution of X into vertical slices and changing the spacing of the slices.”

Transforming a Predictor

It is equivalent to “cutting the distribution of X into vertical slices and changing the spacing of the slices.”

It does not affect the vertical locations of the data (MPG did not change!).

Transforming a Predictor

It is equivalent to “cutting the distribution of X into vertical slices and changing the spacing of the slices.”

It does not affect the vertical locations of the data (MPG did not change!).

When the nonlinearity is the biggest issue in the model, transforming the predictor is a good start.

Transforming a Predictor

It is equivalent to "cutting the distribution of X into vertical slices and changing the spacing of the slices."

It does not affect the vertical locations of the data (MPG did not change!).

When the nonlinearity is the biggest issue in the model, transforming the predictor is a good start.

Finding the right transformation is a bit of art, field knowledge and trial and error.