Multiple Linear Regression

David Gerard

2017-11-20

Learning Objectives

- Multiple linear regression (with testing/CI's).
- Stepwise procedures.
- Model checking.
- Sections 8.1 through 8.3 in DBC

Multiple Linear Regression

Mario Data

```
library(openintro)
library(tidyverse)
data(marioKart)
glimpse(marioKart)
```

```
Observations: 143
Variables: 12
$ ID         <dbl> 150377422259, 260483376854, 32043234...
$ duration   <int> 3, 7, 3, 3, 1, 3, 1, 1, 3, 7, 1, 1, ...
$ nBids      <int> 20, 13, 16, 18, 20, 19, 13, 15, 29, ...
$ cond       <fctr> new, used, new, new, new, new, used...
$ startPr    <dbl> 0.99, 0.99, 0.99, 0.99, 0.01, 0.99, ...
$ shipPr     <dbl> 4.00, 3.99, 3.50, 0.00, 0.00, 4.00, ...
$ totalPr    <dbl> 51.55, 37.04, 45.50, 44.00, 71.00, 4...
$ shipSp     <fctr> standard, firstClass, firstClass, s...
$ sellerRate <int> 1580, 365, 998, 7, 820, 270144, 7284...
```

Mario Data

- `totalPr`: Total price, which equals the auction price plus the shipping price.
- `cond`: Game condition, either new or used.
- `stockPhoto`: Whether the auction feature photo was a stock photo or not. If the picture was used in many auctions, then it was called a stock photo.
- `duration`: Auction length, in days.
- `wheels`: Number of Wii wheels included in the auction. These are steering wheel attachments to make it seem as though you are actually driving in the game. When used with the controller, turning the wheel actually causes the character on screen to turn.

Create Indicator Variables

```
marioKart$cond_new    <- (marioKart$cond == "new") * 1
marioKart$stock_photo <- (marioKart$stockPhoto == "yes") * 1
mario <- select(marioKart, totalPr, cond_new,
                stock_photo, duration, wheels)
head(mario)
  totalPr cond_new stock_photo duration wheels
1   51.55        1           1        3      1
2   37.04        0           1        7      1
3   45.50        1           0        3      1
4   44.00        1           1        3      1
5   71.00        1           1        1      2
6   45.00        1           1        3      0
```

set up

We have

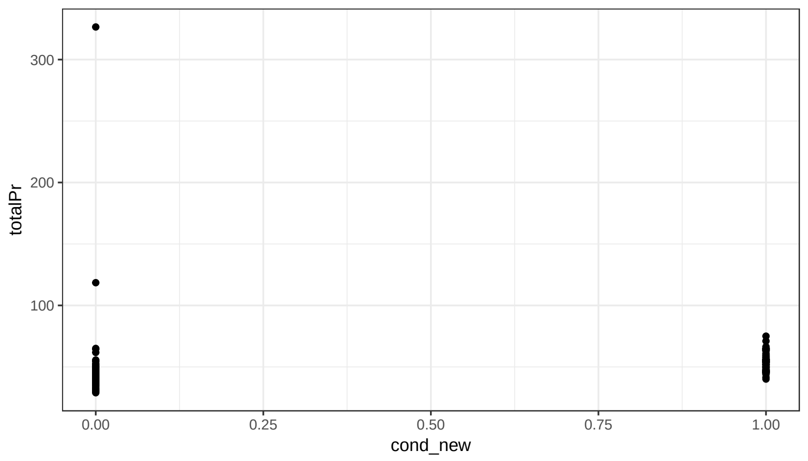
- a single response variable y
- several predictor/explanatory variables $x_1, \ldots, x_p$

Data for multiple linear regression consist of the values of y and $x_1, \ldots, x_p$ for n individuals. We write the data in the form:

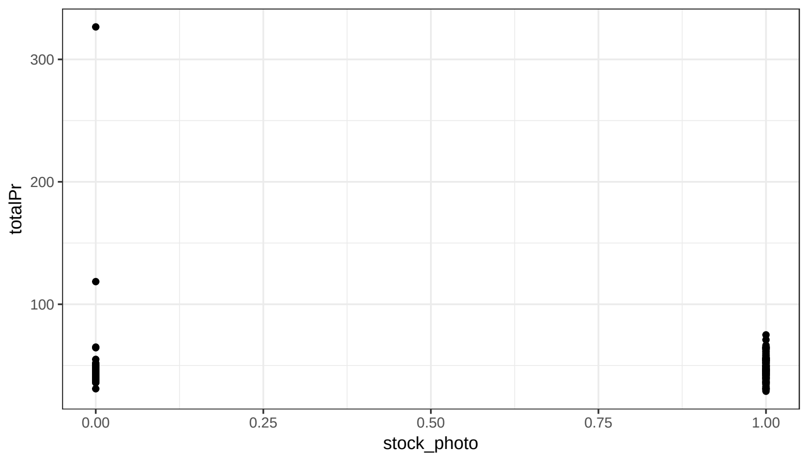
Individual	Predictors				Response
i	x_1	x_2	$\cdots$	x_p	y
1	x_{11}	x_{12}	$\cdots$	x_{1p}	y_1
2	x_{21}	x_{22}	$\cdots$	x_{2p}	y_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
n	x_{n1}	x_{n2}	$\cdots$	x_{np}	y_n

Following our principles of data analysis, we look first at each variable separately.

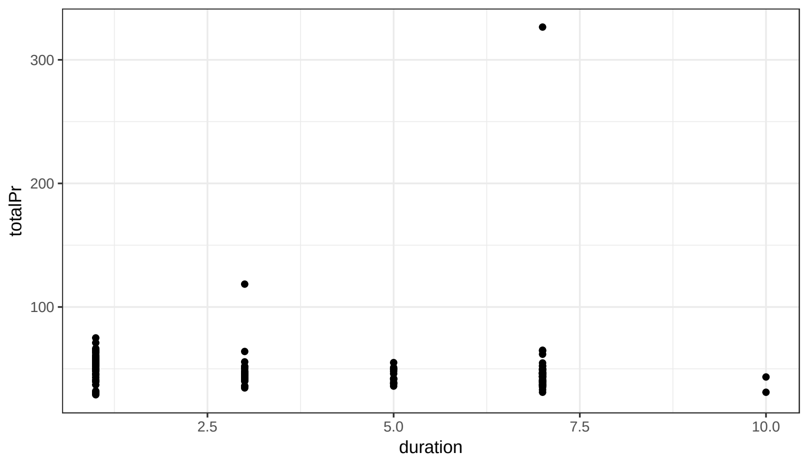
EDA



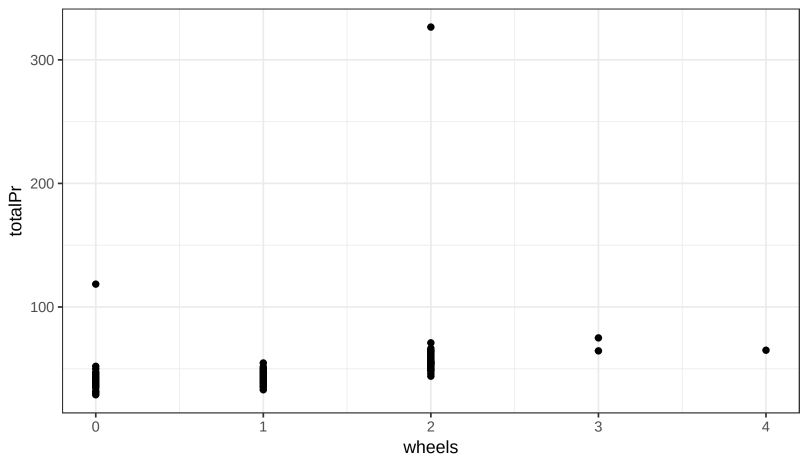
EDA



EDA



EDA



Correlations

```
round(cor(mario), digits = 2)

            totalPr cond_new stock_photo duration wheels
totalPr        1.00     0.13       -0.09    -0.04   0.33
cond_new       0.13     1.00        0.38    -0.48   0.43
stock_photo   -0.09     0.38        1.00    -0.37   0.07
duration      -0.04    -0.48       -0.37     1.00  -0.30
wheels         0.33     0.43        0.07    -0.30   1.00
```

EDA results

- It seems that marginally (i.e. just looking at one predictor at a time), `price` is positively associated with `cond_new` and `wheels` and perhaps negatively associated with `stock_photo` and `duration`, though these latter two relationships are possibly non-existant (a result of noise) or just weak.
- The predictors are also moderately correlated with each other.
- There is one huge outlier and a moderate outlier.

A first fit

```
lmout <- lm(totalPr ~ cond_new, data = mario)
summary(lmout)

Call:
lm(formula = totalPr ~ cond_new, data = mario)

Residuals:
   Min     1Q Median     3Q    Max
-18.17  -7.77  -3.15   1.86 279.36

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    47.15       2.79   16.90   <2e-16
cond_new        6.62       4.34    1.52     0.13

Residual standard error: 25.6 on 141 degrees of freedom
Multiple R-squared:  0.0162,Adjusted R-squared:  0.00924
F-statistic: 2.32 on 1 and 141 DF,  p-value: 0.13
```

A first fit: without outlier

```
lmout <- lm(totalPr ~ cond_new, data = mario[-c(20, 65), ])
summary(lmout)

Call:
lm(formula = totalPr ~ cond_new, data = mario[-c(20, 65), ])

Residuals:
    Min      1Q  Median      3Q     Max
-13.891  -5.831   0.129   4.129  22.149

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   42.871      0.814   52.67  < 2e-16
cond_new      10.900      1.258    8.66  1.1e-14

Residual standard error: 7.37 on 139 degrees of freedom
Multiple R-squared:  0.351,Adjusted R-squared:  0.346
F-statistic:   75 on 1 and 139 DF,  p-value: 1.06e-14
```

Uh Oh

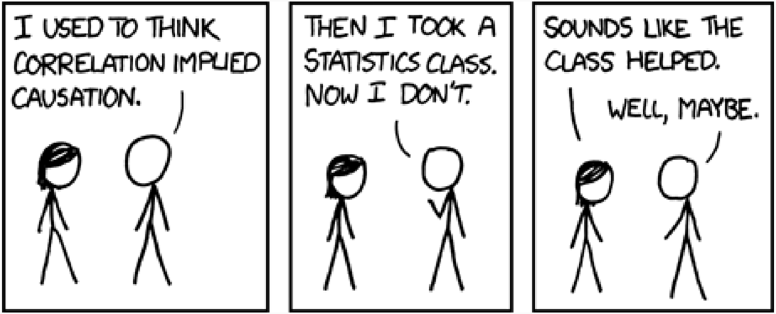
- If you have outliers, the first thing to do is try to explain those outliers.
- The second thing to do is fit the model both with and without the outliers. Hopefully you get the same results.
- If the results change consult a statistician: they will either (1) fit a “robust” procedure (e.g. minimize the sum of absolution deviations rather than the sum of squared deviations) or (2) try to incorporate the outliers in the model.
- We’ll just remove them for now.

```
mario <- mario[-c(20, 65), ]
```

Interpret

- New Mario Kart games tend to cost an average of \$10.90 more than used Mario Kart games on Ebay.
- The association is significant ($p \approx 1.1 \times 10^{-14}$).
- Don't confuse this with *causation*. E.g. new games come with more Wii wheels which could be what is actaully causing the increase in price.

Correlation vs Causation



Goals

- We will try to find associations between the response and each each predictor while controlling for the other predictors.
- This will still not allow us to make claims of causality.

Multiple Linear Regression

A multiple linear regression model is a linear model with many predictors. In general, we write the model as

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon,$$

where p is the number of predictors and ϵ is some noise term (often assumed to be distributed $N(0, \sigma^2)$).

Interpretation

- Intro stat interpretation: β_j is the change in y for each unit change in x_j when holding all other predictors constant.
- Some statisticians think this sounds too causal, so they use more verbose language: β_j is the difference in the average y's between two populations that are the same in every respect except that they differ by 1 in x_j.
- That is, we aren't *changing* x_j, we're just looking at two populations that have different x_j's.

Wii example

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \epsilon.$$

Estimating the Regression Coefficients

The true population parameters $\beta_0, \beta_1, \ldots, \beta_p$ and σ are estimated from the data by the least squares method. That is, we minimize the *residual sum of squares*

$$\begin{aligned} \mathrm{SSE} &= \sum_{i=1}^{n} (e_i)^2 \\ &= \sum_{i=1}^{n} (y_i - \hat{y})^2 \\ &= \sum_{i=1}^{n} (y_i - b_0 - b_1 x_{i1} - \cdots - b_p x_{ip})^2 \end{aligned}$$

Estimating the Variance

The estimator of σ^2 is

$$s^2 = \frac{\text{SSE}}{n-p-1} = \frac{\sum (e_i)^2}{n-p-1}$$

where $n-p-1$ is the number of degrees of freedom.
Number of samples n minus the number of parameters $p+1$.

Mario Example

```
lmout <- lm(totalPr ~ cond_new + stock_photo +
              duration + wheels,
            data = mario)
sumout <- summary(lmout)
round(sumout$coefficients, digits = 2)
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    36.21       1.51   23.92     0.00
cond_new        5.13       1.05    4.88     0.00
stock_photo     1.08       1.06    1.02     0.31
duration       -0.03       0.19   -0.14     0.89
wheels          7.29       0.55   13.13     0.00
```

Fitted model and interpretation

$$y_i = 36.21 + 5.13x_{1i} + 1.08x_{2i} + -0.03x_{3i} + 7.29x_{4i} + \epsilon_i.$$

- If game i and game j differ only in that game i only has one more wheel than game j, then we would expect individual i's total price to be about 7.29 dollars more.

Model Assumptions

- The sample is a SRS from the population
  This can't be checked; this needs to be taken care of when the sample is drawn.
- There is a linear relationship in the population
  Checking this isn't as straightforward as with simple linear regression, but we should draw a plot of *residuals vs. fitted values* and check for any patterns.
- The standard deviation of the residuals is constant.
  Using the same plot as above, check for non-uniformity in the spread of residuals around the center line.
- The response varies Normally about the population regression line.
  Check with a *Normal quantile plot* of the residuals.

Inference for Regression Coefficients

A 95% confidence interval for β_j is

$$\hat{\beta}_j \pm t^* \mathrm{SE}(\hat{\beta}_j)$$

where t^* is the number such that 95% of the area of the t_{n-p-1} distribution falls between $-t^*$ and t^*

To test the hypothesis

$$H_0 : \beta_j = 0 \qquad (\beta_i \text{ arbitrary for } i \neq j)$$

compute the t-statistic

$$T = \frac{\hat{\beta}_j}{\mathrm{SE}(\hat{\beta}_j)}$$

More on Testing

- the p-value for this test statistic is computed from the t_{n-p-1} distribution
  $\rightarrow$ for $H_a : \beta_j > 0$, p-value is $P(t_{n-p-1} > T)$
  $\rightarrow$ for $H_a : \beta_j < 0$, p-value is $P(t_{n-p-1} < T)$
  $\rightarrow$ for $H_a : \beta_j \neq 0$, p-value is $2P(t_{n-p-1} > |T|)$
- if the regression model assumptions are true, testing $H_0 : \beta_j = 0$ corresponds to testing whether or not x_j is a significant predictor of y, *assuming all the other predictors are already in the model*.

ANOVA table for Multiple Regression

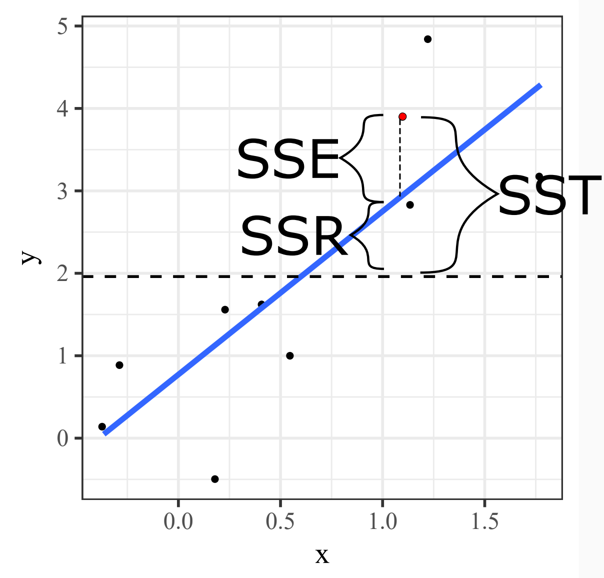
The basic ideas of the regression ANOVA table are the same in simple and multiple regression.

ANOVA expresses variation in the form of sums of squares. It breaks the total variation into two parts: SSR and SSE:

Source	SS	df
Regression (SSR)	$\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$	p
Residual (SSE)	$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	$n-p-1$
Total	$\sum_{i=1}^{n}(y_i - \bar{y})^2$	$n-1$

$$\mathrm{SST} = \mathrm{SSR} + \mathrm{SSE}$$

ANOVA Decomposition



R^2

The statistic

$$R^2 = 1 - \frac{\text{SSE}}{\text{SST}} = \frac{\text{SSR}}{\text{SST}} = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

is the proportion of the variation of the response variable y that is explained by the explanatory variables $x_1, x_2, \cdots, x_p$. R^2 is called the **multiple correlation coefficient**.

Adjusted R^2

The R^2 increases with every additional predictor. This is a mathematical fact. But some predictors may not be particularly useful in the regression.
Use Adjusted-R^2 :
$R^2_{adj} = 1 - \frac{\text{SSE}_{/(n-p-1)}}{\text{SST}_{/(n-1)}}$
Adjusted R^2 does not necessarily increase with more predictors.
The adjusted R^2 compares the estimated sigmas - the numerator in the fraction is s. The denominator is fixed. So if s is smaller a model is better.

Model Selection

Model with duration

```
lmout1 <- lm(totalPr ~ cond_new + stock_photo +
               duration + wheels,
             data = mario)
sumout1 <- summary(lmout1)
sumout1$adj.r.squared
[1] 0.7108
```

Model **without** `duration`

```
lmout2 <- lm(totalPr ~ cond_new + stock_photo +
             wheels,
            data = mario)
sumout2 <- summary(lmout2)
sumout2$adj.r.squared
[1] 0.7128
```

Bigger model isn't always the best!

- The estimated proportion of variance explained by the second model is larger than from the first.
- Intuition: `duration` has no affect on price so our model fruitlessly works too hard to estimate it's effect.
- So we should prefer this second (simpler) model without duration.

Backwards Elimination

- Fit the "full" model (that with every predictor included).
- Remove the predictor that results in the greatest increase in adjusted R^2.
- Keep removing predictors in this way until you cannot increase R^2.

Backwards Elimination

Iteration 1

Model	adjusted R^2
Full	0.711
No `cond_new`	0.663
No `stock_photo`	0.711
No `duration`	0.713
No `wheels`	0.349

Backwards Elimination

Iteration 1

Model	adjusted R^2
Full	0.711
No cond_new	0.663
No stock_photo	0.711
No duration	0.713
No wheels	0.349

Backwards Elimination

Iteration 2

Model	adjusted R^2
No `duration`	0.713
No `duration` and no `cond_new`	0.659
No `duration` and no `stock_photo`	0.712
No `duration` and no `wheels`	0.341

- No increase in adjusted R^2, so stop with this model.

Final Model

```
lmout <- lm(totalPr ~ cond_new + stock_photo + wheels,
            data = mario)
sumout <- summary(lmout)
round(sumout$coefficients, digits = 2)
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    36.05       0.97   36.99     0.00
cond_new        5.18       1.00    5.20     0.00
stock_photo     1.12       1.02    1.10     0.27
wheels          7.30       0.54   13.40     0.00
```

The final model is

$$\text{totalPr} = 36.1 + 5.2\text{cond\_new} + 1.1\text{stock\_photo} + 7.3\text{wheels} + \text{error}.$$

Other methods of backwards elmination

- There are other statistics you can use to do backwards elimination (e.g. based on *p*-values).
- R uses something called AIC.

```
full_model <- lm(totalPr ~ cond_new + stock_photo +
                    duration + wheels, data = mario)
backout <- step(full_model, direction = "backward",
                trace = FALSE)
```

Backwards Elimination Results

```
backout


Call:
lm(formula = totalPr ~ cond_new + wheels, data = mario)

Coefficients:
(Intercept)     cond_new       wheels
      36.78         5.58         7.23
```

Forward Selection

- Start with the model including just the intercept term and keep adding predictors until you can't increase the R^2 (or AIC or decrease the p-values, etc.)

```
base_model <- lm(totalPr ~ 1, data = mario)
full_model <- lm(totalPr ~ cond_new + stock_photo +
                   duration + wheels, data = mario)
forout <- step(object = base_model,
               scope = list(lower = base_model,
                            upper= full_model),
               direction = "forward",
               trace = FALSE)
```

Forward Selection Results

```
forout

Call:
lm(formula = totalPr ~ wheels + cond_new, data = mario)

Coefficients:
(Intercept)       wheels     cond_new
      36.78         7.23         5.58
```

Checking fit

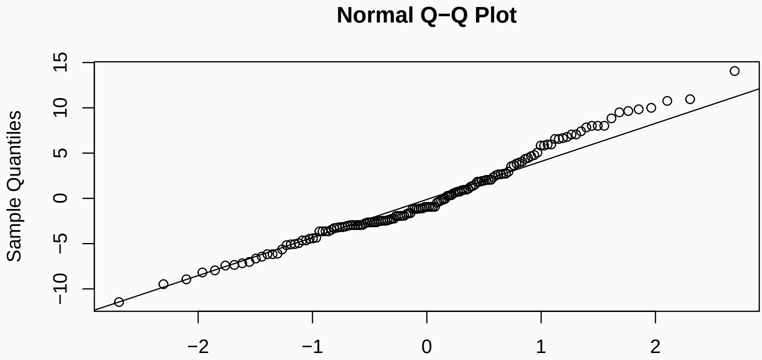
Normality Assumption

Error term is nearly normal.

- This is less important for large n if all you want is to estimate/infer the β_j's. This follows from the CLT.
- This assumption is super important for prediction intervals.
- You can check that the residuals are nearly normal.
- Use qq-plots.

Nearly normal

```
lmout <- lm(totalPr ~ cond_new + stock_photo +
                wheels, data = mario)
residuals <- resid(lmout)
qqnorm(residuals)
qqline(residuals)
```



Other Assumptions

- Variability of error term is nearly constant.
- Error terms are independent.
    - The book says that the "residuals are independent". This is very wrong (why?).
- Each variable is linearly related to the outcome.

Testing these assumptions

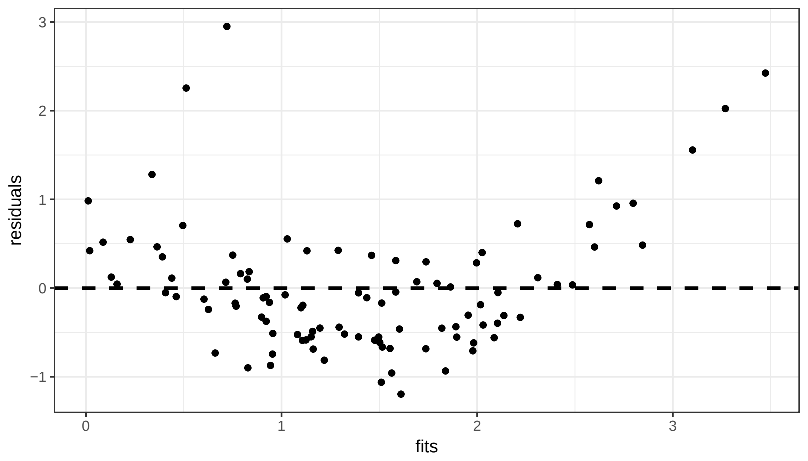
To test these assumptions, plot the residuals against:

- the predictors,
- the absolute value of the responses,
- the absolute value of the fitted responses, and
- the ordering of the observations.

If you don't see anything pattern, then the model assumptions are looking pretty good.

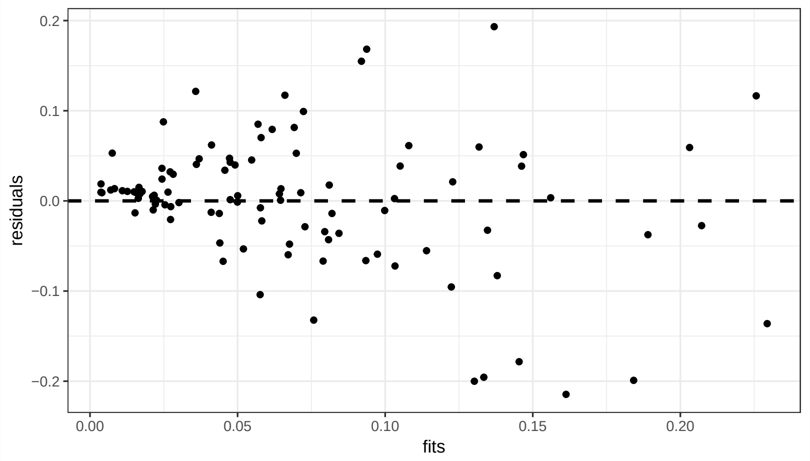
Simulated Data 1

Example of Non-linear relationship:



Simulated Data 2

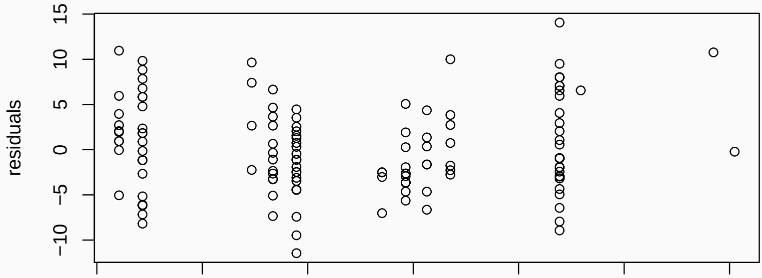
Example of Non-constant Variance:



Mario Resids

Resids vs Fits

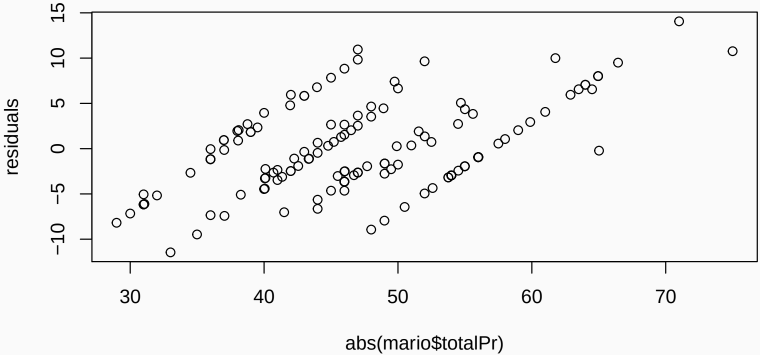
```
lmout     <- lm(totalPr ~ cond_new + stock_photo +
                 wheels, data = mario)
residuals <- resid(lmout)
fits      <- predict(lmout)
plot(abs(fits), residuals)
```



Mario Resids

Resids vs y.

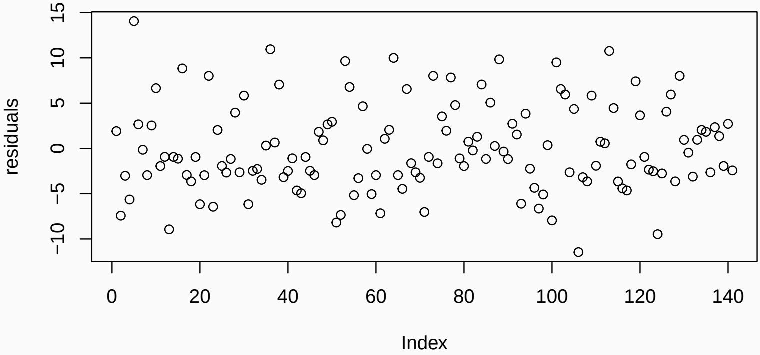
```
plot(abs(mario$totalPr), residuals)
```



Mario Resids

Resids vs order.

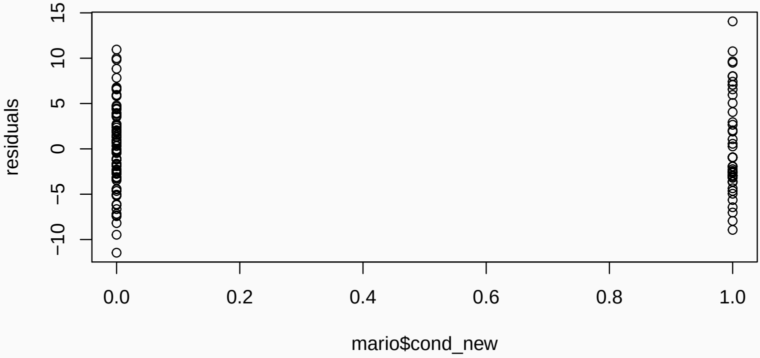
```
plot(residuals)
```



Mario Resids

Resids vs `cond_new`.

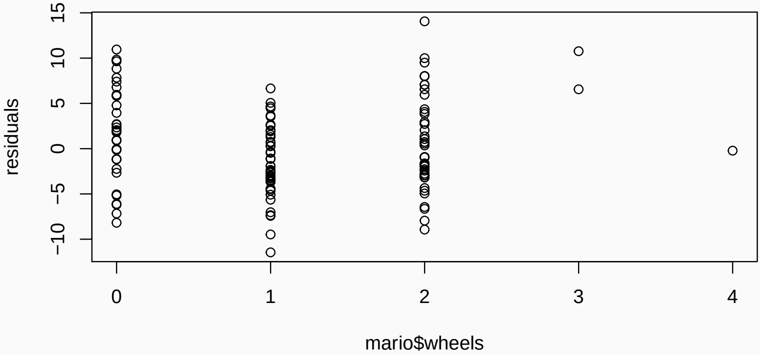
```
plot(mario$cond_new, residuals)
```



Mario Resids

Resids vs `wheels`.

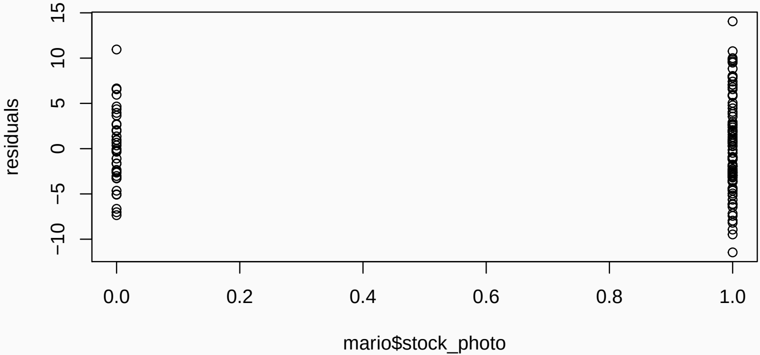
```
plot(mario$wheels, residuals)
```



Mario Resids

Resids vs `stock_photo`.

```
plot(mario$stock_photo, residuals)
```

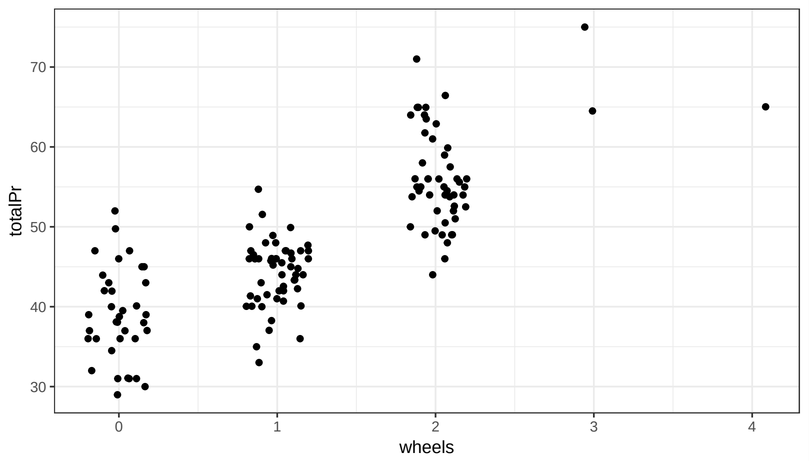


Conclusions

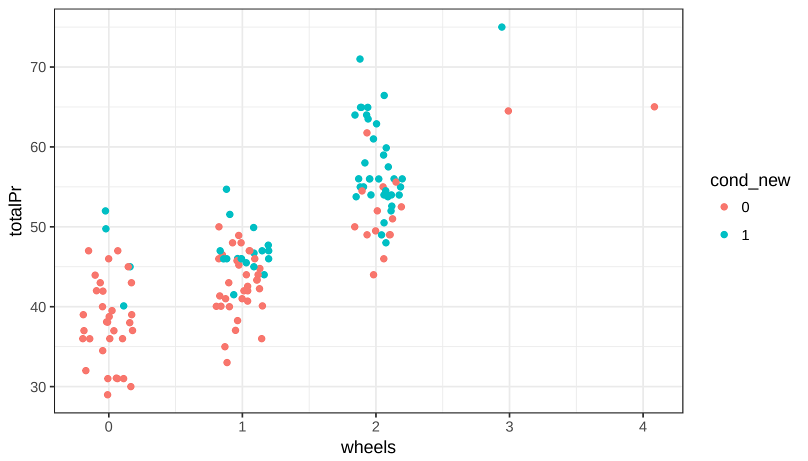
- Some possible problems.
- Doesn't look *too* bad, but could look better.

Intuition behind Indicator Variables

totalPr vs cond_new and wheels

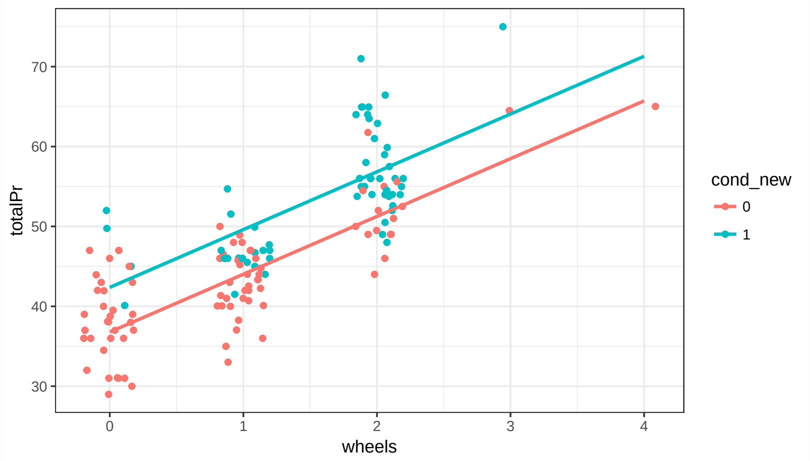


totalPr vs cond_new and wheels

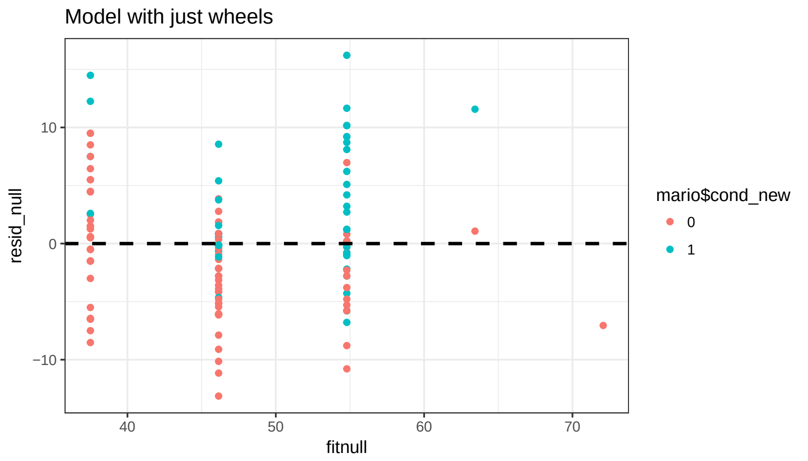


Different Intercepts

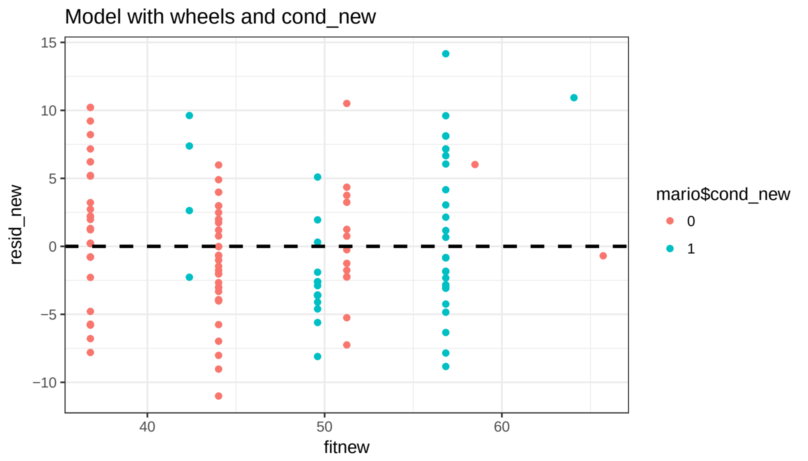
$$\text{price} = \beta_0 + \beta_1 \text{cond\_new} + \beta_2 \text{wheels} + \text{error}.$$



Color Code Residuals



Color Code Residuals



Different Slopes and Intercepts

$$\text{price} = \beta_0 + \beta_1 \text{cond\_new} + \beta_2 \text{wheels} + \beta_3 \text{cond\_new} \times \text{wheels} + \text{error}.$$