Modeling nonlinear relationships

David Puelz

Polynomial models

Log transformations

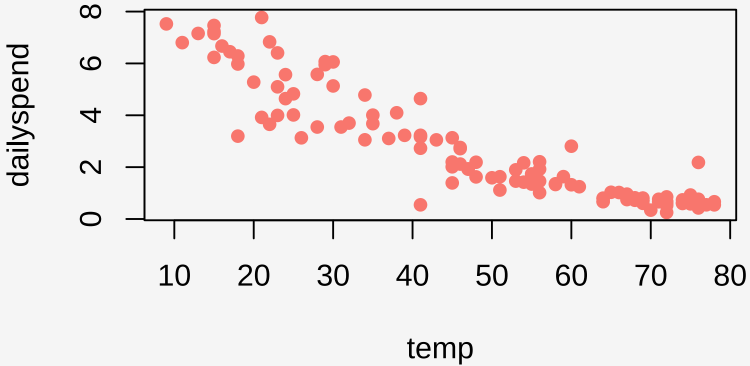
Log-log models and price elasticity

Utilities data set

The data set `utilities` contains information on the utility bills for a house in Minnesota. We'll focus on two variables:

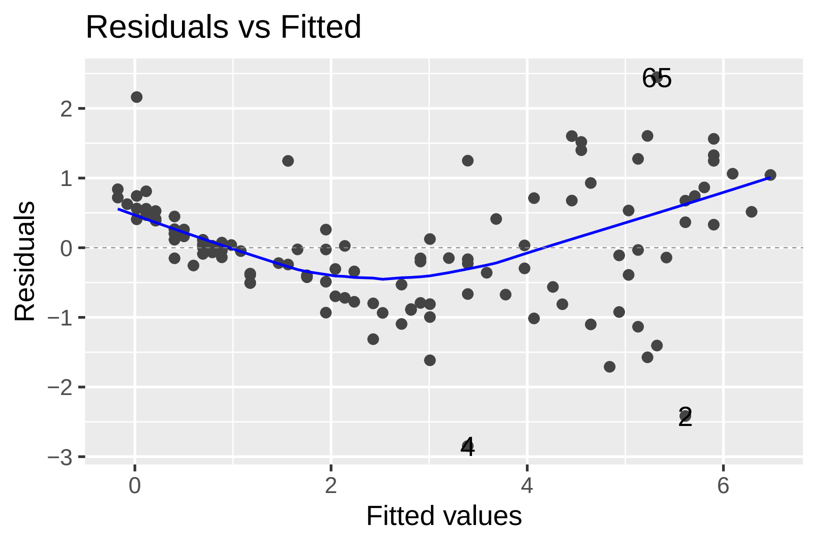
- **dailyspend** is the average amount of money spent on utilities (e.g. heating) for each day during the month
- **temp** is the average temperature outside for that month

What problems do you see here?



```r
model <- lm(dailyspend ~ temp, data=utilities)
summary(model)


Call:
lm(formula = dailyspend ~ temp, data = utilities)

Residuals:
   Min     1Q Median     3Q    Max
-2.847 -0.504 -0.024  0.515  2.448

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.34762    0.20645    35.6   <2e-16 ***
temp        -0.09643    0.00391   -24.7   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.866 on 115 degrees of freedom
Multiple R-squared:  0.841,Adjusted R-squared:  0.84
F-statistic:  608 on 1 and 115 DF,  p-value: <2e-16
```

Linearity and homoscedasticity are violated

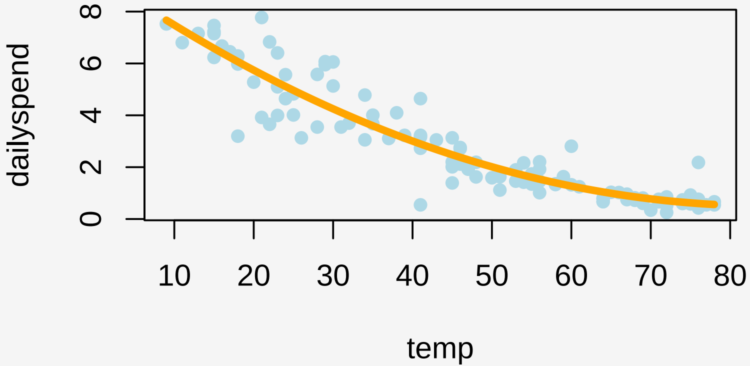


Using polynomial regression to fix problems

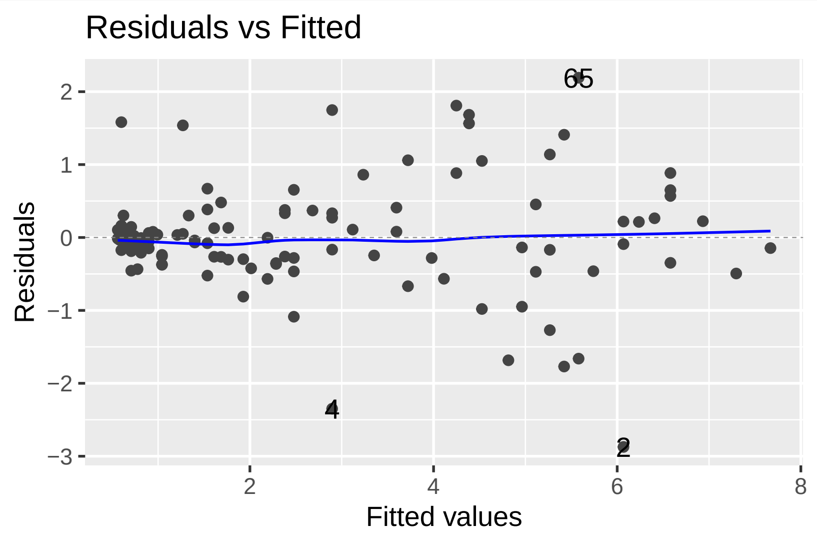
- If a polynomial curve (e.g., quadratic, cubic, etc) would be a better fit for the data than a line, we can fit a curve to the data
- The way we do this is by adding X^2, X^3, etc. terms to the model as additional predictors

```
poly.model <- lm(dailyspend ~ temp + I(temp^2), data=utilities)
summary(poly.model)


Call:
lm(formula = dailyspend ~ temp + I(temp^2), data = utilities)

Residuals:
    Min      1Q  Median      3Q     Max
-2.8725 -0.2805 -0.0393  0.2639  2.1912

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.472289   0.390789   24.24  < 2e-16 ***
temp        -0.211555   0.019105  -11.07  < 2e-16 ***
I(temp^2)    0.001248   0.000204    6.12  1.3e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.755 on 114 degrees of freedom
Multiple R-squared:  0.88,Adjusted R-squared:  0.878
F-statistic:  419 on 2 and 114 DF,  p-value: <2e-16
```

Adding an X^2 term fits a parabola to the data



It solves the linearity problem!

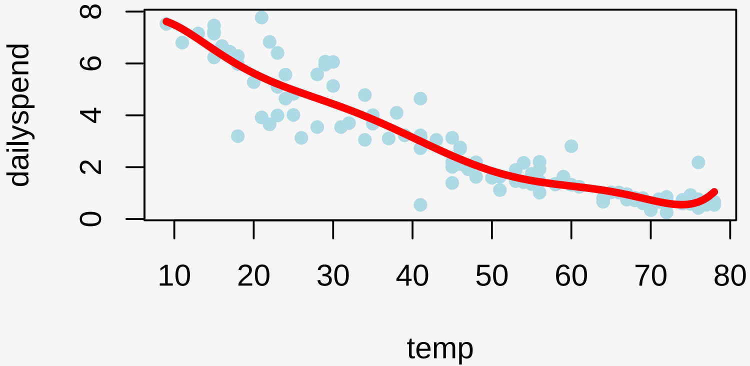


What about a higher-order polynomial?

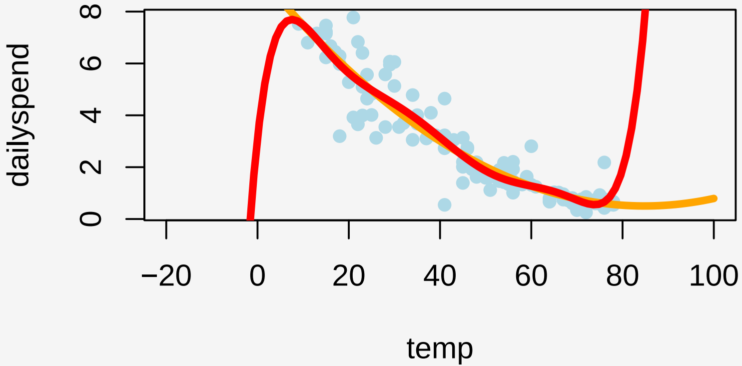
- We could fit a cubic curve by adding an X^3 term (in addition to the X and X^2 terms)
- Making the polynomial higher order will increase R^2—why not go nuts and fit a 7th degree polynomial?

Degree		R^2
1	linear	0.841
2	quadratic	0.8803
3	cubic	0.8812
4	quartic	0.8813
5		0.8815
6		0.8822
7		0.8837

Too high a degree "overfits" the data



Too high a degree creates dangers with extrapolation



Building polynomial models

- Start simple: only add higher-degree terms to the extent it gives you a substantial increase in R^2, or makes an assumption hold that wasn't holding before

Building polynomial models

- Start simple: only add higher-degree terms to the extent it gives you a substantial increase in R^2, or makes an assumption hold that wasn't holding before
- Be careful about overfitting

Building polynomial models

- Start simple: only add higher-degree terms to the extent it gives you a substantial increase in R^2, or makes an assumption hold that wasn't holding before
- Be careful about overfitting
- Be particularly careful about extrapolating beyond the range of the data

Building polynomial models

- Start simple: only add higher-degree terms to the extent it gives you a substantial increase in R^2, or makes an assumption hold that wasn't holding before
- Be careful about overfitting
- Be particularly careful about extrapolating beyond the range of the data
- Mind-bender: We can think about an X^2 term as an interaction of X with itself: in a parabola, the slope depends on the value of X!

Polynomial models

Log transformations

Log-log models and price elasticity

Using transformations to enhance interpretability and fix problems

- Sometimes, a violation of regression assumptions can be fixed by transforming one or the other of the variables (or both).
- When we transform a variable, we have to also transform our interpretation of the model
- Often the new interpretation is more meaningful!


LOG!

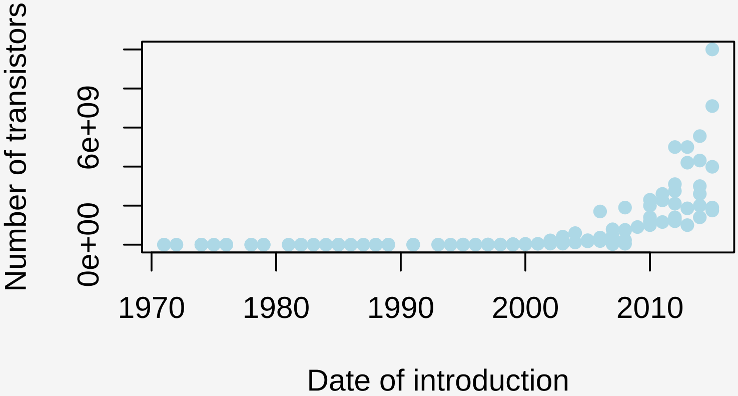
The log transformation

The log transformation is frequently useful in regression, because many nonlinear relationships are naturally exponential.

- $\log_b x = y$ when $b^y = x$
- For example, $\log_{10} 1000 = 3$, $\log_{10} 100 = 2$, and $\log_{10} 10 = 1$
- The natural log is $\log_e$, where $e \approx 2.72$ — when we say "log" we will usually mean "natural log" (although for our purposes the base doesn't matter)

Moore's Law

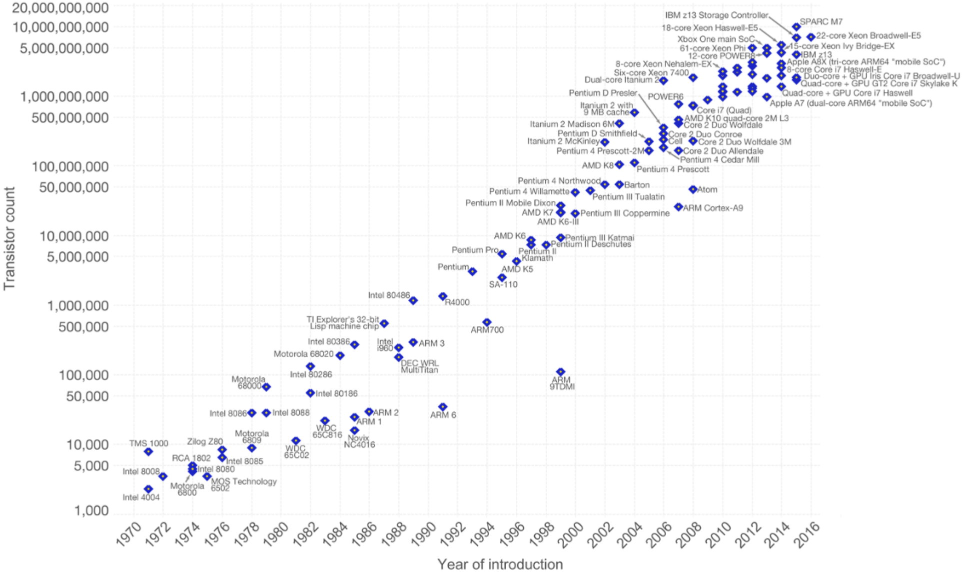
- Moore's Law was a prediction made by Gordon Moore in 1965 (!) that the number of transistors on computer chips would double every 2 years
- This implies exponential growth, so a linear model won't fit well (and neither will any polynomial)


Number of transistors
0e+00
6e+09
1970
1980
1990
2000
2010
Date of introduction

Moore's Law – The number of transistors on integrated circuit chips (1971-2016)

Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.

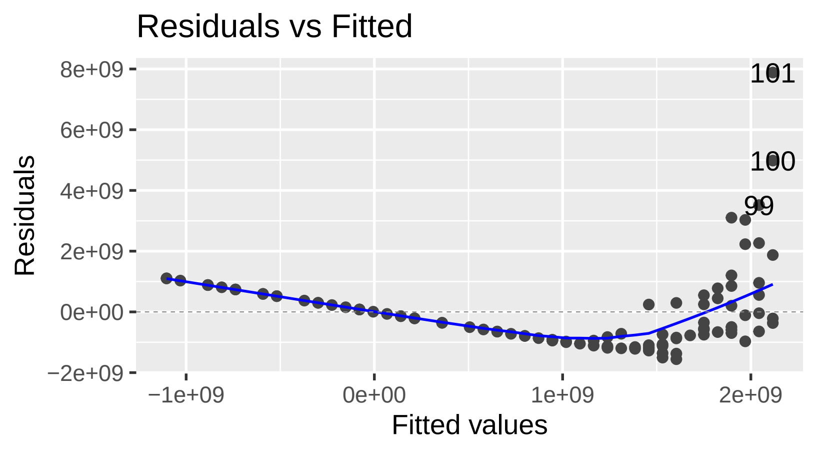


Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)

The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

A linear model is a spectacular fail

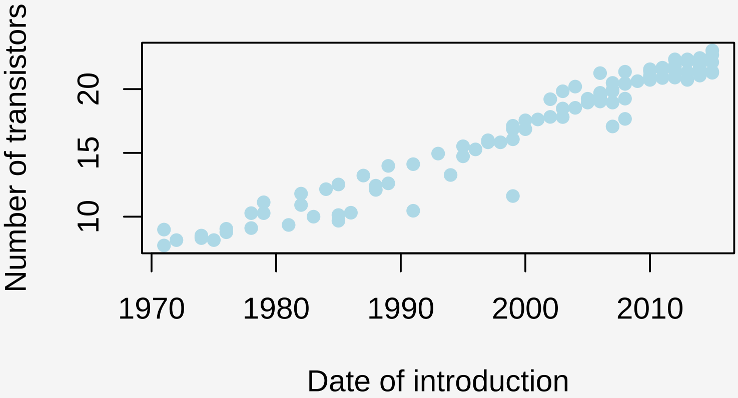
```
linear.model <- lm(Transistor.count ~ Date.of.introduction,
                   data=moores.law)
autoplot(linear.model, which=1, ncol=1)
```



Modeling exponential growth

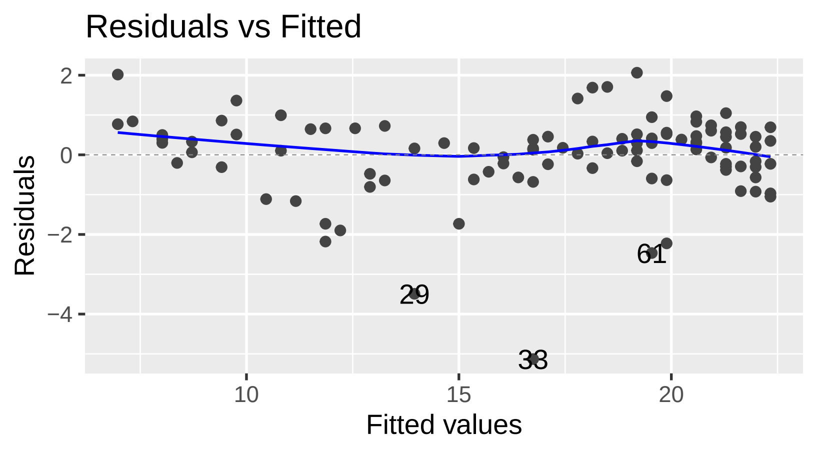
- If $Y = ae^{bX}$, then $\log(Y) = \underbrace{\log(a)}_{\beta_0} + \underbrace{b}_{\beta_1} X$
- In other words, $\log(Y)$ is a linear function of X
- So to model Y as an exponential function of X, predict $\log(Y)$ as a linear function of X

log(Transistors) does have a linear relationship with year

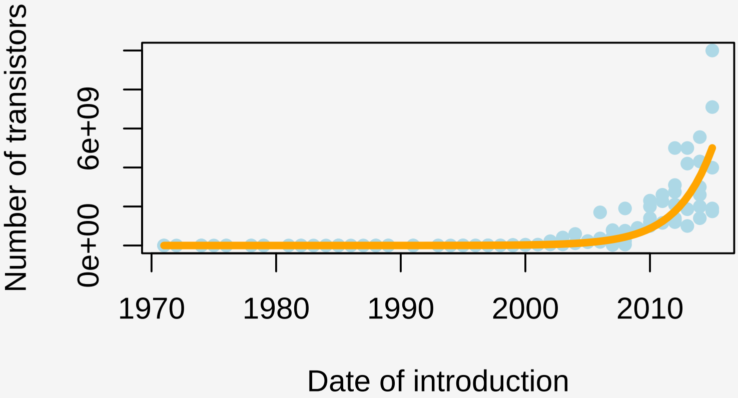


A log-linear model satisfies the linearity assumption

```
log.model <- lm(log(Transistor.count) ~ Date.of.introduction,
                data=moores.law)
autoplot(log.model, which=1, ncol=1)
```



Interpreting the log-linear model



Interpreting the log-linear model

Our model is

$$\widehat{\log(\text{Transistors})} = -681.21 + 0.35 \cdot \text{Year}$$

Interpreting the log-linear model

Our model is

$$\log(\widehat{\text{Transistors}}) = -681.21 + 0.35 \cdot \text{Year}$$

Two interpretations of the slope coefficient:

- Every year, the predicted log of transistors goes up by 0.35

Interpreting the log-linear model

Our model is

$$\log(\widehat{\text{Transistors}}) = -681.21 + 0.35 \cdot \text{Year}$$

Two interpretations of the slope coefficient:

- Every year, the predicted log of transistors goes up by 0.35
- More useful: Every year, the predicted number of transistors goes up by 35%

Slope interpretations in the presence of logs

Model	Equation	Interpretation
Linear	$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$	1 unit increase in $X \longrightarrow$ $\hat{\beta}_1$ unit increase in $\hat{Y}$
Log-linear	$\log(\hat{Y}) = \hat{\beta}_0 + \hat{\beta}_1 X$	1 unit increase in $X \longrightarrow$ $\approx (100\hat{\beta}_1)\%$ increase in $\hat{Y}$
Linear-log	$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 \log(X)$	1% increase in $X \longrightarrow$ $\approx 0.01\hat{\beta}_1$ unit increase in $\hat{Y}$
Log-log	$\log(\hat{Y}) = \hat{\beta}_0 + \hat{\beta}_1 \log(X)$	1% increase in $X \longrightarrow$ $\approx \hat{\beta}_1\%$ increase in $\hat{Y}$

Slope interpretations in the presence of logs

Model	Example	Interpretation
Linear	$\hat{Y} = 2 + 0.3X$	1 unit increase in $X \longrightarrow$ 0.3 unit increase in $\hat{Y}$
Log-linear	$\log(\hat{Y}) = 2 + 0.3X$	1 unit increase in $X \longrightarrow$ $\approx$ 30% increase in $\hat{Y}$
Linear-log	$\hat{Y} = 2 + 0.3\log(X)$	1% increase in $X \longrightarrow$ $\approx$ 0.003 unit increase in $\hat{Y}$
Log-log	$\log(\hat{Y}) = 2 + 0.3\log(X)$	1% increase in $X \longrightarrow$ $\approx$ 0.3% increase in $\hat{Y}$

When is the log transformation useful?

- You can transform X, Y, or both
- Anytime you need to "squash" one of the variables (logs make huge numbers not so big!), try transforming it with a log
- In this case, Transistors is skewed right so it is a good candidate for log
- You may need to do a little bit of trial and error to see what works best
- Other transformations are possible!

Polynomial models

Log transformations

Log-log models and price elasticity

An elasticity interpretation for log-log models

- The beer data set contains sales data for cases of beer sold at a supermarket chain
- PRICE12PK is the price of 12-packs, and CASES12PK is the number of 12-packs sold
- There are also variables for 18-packs and 30-packs

What is price elasticity?

- Price elasticity is the ratio of how much demand (quantity sold) will increase when the price increases:

$$\begin{aligned}\text{elasticity} &= \frac{\%\ \text{change in demand}}{\%\ \text{change in price}} \\ &= \%\ \text{increase in demand when price increases by 1\%}\end{aligned}$$

What is price elasticity?

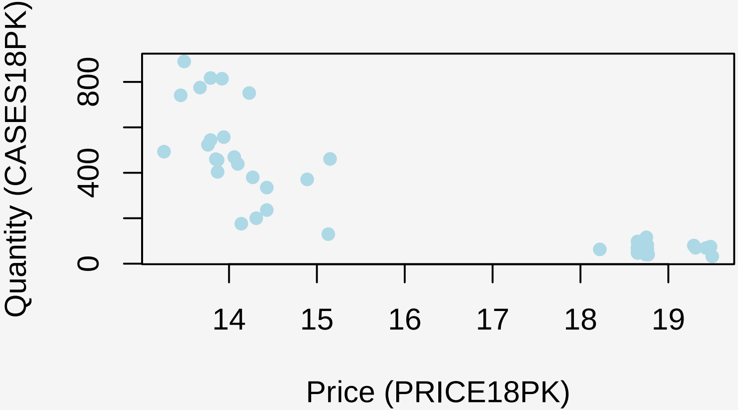
- Price elasticity is the ratio of how much demand (quantity sold) will increase when the price increases:

$$\begin{aligned}\text{elasticity} &= \frac{\%\ \text{change in demand}}{\%\ \text{change in price}} \\ &= \%\ \text{increase in demand when price increases by 1\%}\end{aligned}$$

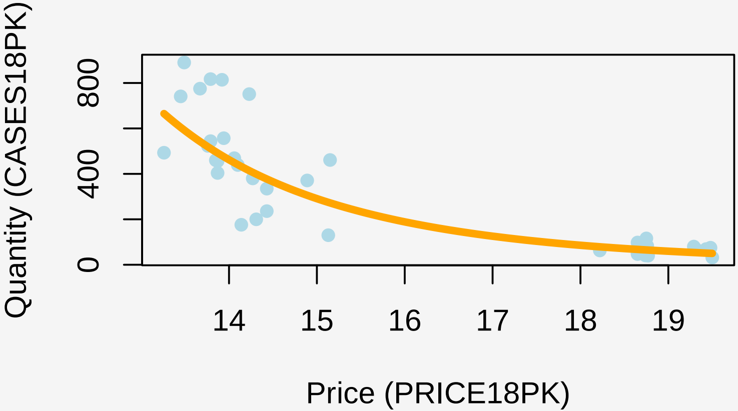
- This is exactly our interpretation of slope coefficients in a log-log model!

Will a linear model work to predict quantity from price?

This is essentially a demand curve:



Fitting a log-log model



Log transformations can also fix heteroscedasticity!

Compare the residual plots for the linear model (left) vs the log-log model (right):