Chapter 2

DJM, Revised: NAK

29 January 2018

Chapter Overview

Problems with regression, and in particular, linear regression

1. Linearity is almost always an approximation.
   - What do we mean by linearity though?
   - What does this mean for the residuals
2. Collinearity of predictor variables **can** cause difficulties for numerics and interpretation.
   - "Collinearirty" versus "Correlated"
3. The the "fit" of our model depends strongly on the marginal distribution of $\underline{X}$.
   - Manipulating R^2
   - Transformations on $\underline{X}$
4. Hidden variables can affect our estimated model more than you may think.
5. Probabilitic assumptions and what conclusions we make from them.

Regression in general: The Linearity Assumption

- If I want to predict Y from X, it is almost always the case that

$$\mu(\underline{x}) = \mathbb{E}\left[Y \mid \underline{X} = \underline{x}\right] \neq \underline{x}^\top \underline{\beta}$$

- Therefore, there is some sort of **bias** involved.
  - Global bias?
  - Local bias?
- We can include as many predictors as we like, but this doesn't change the fact that the world is **non-linear**.

What is bias? Statistically...

- If θ is a parameter we want to estimate, and $\widehat{\theta}$ is a suggested estimate:

$$\text{Bias}\left[\widehat{\theta}\right] = \mathbb{E}\left[\widehat{\theta}\right] - \theta$$

- Bias is certainly not good, but is not necessarily all that bad either.
- A very simple example: let $Z_1, \ldots, Z_n \sim N(\mu, 1)$. μ is unknown $\rightarrow$ use the data to estimate.
- 3 potential estimators:
  1. $\widehat{\mu}_1 = 12$,
  2. $\widehat{\mu}_2 = Z_6$,
  3. $\widehat{\mu}_3 = \overline{Z}$.
- Calculate the bias and variance of each estimator.

Asymptotic efficiency

This and MLE are covered in 420.

There are many properties one can ask of estimators $\widehat{\theta}$ of parameters θ

1. Unbiased: $\mathbb{E}\left[\widehat{\theta}\right] - \theta = 0$
2. Consistent: $\widehat{\theta} \xrightarrow{n\to\infty} \theta$
3. Efficient: $\mathbb{V}\left[\widehat{\theta}\right]$ is the smallest of all unbiased estimators
4. Asymptotically efficient: Maybe not efficient for every n, but in the limit, the variance is the smallest of all unbiased estimators.
5. Minimax: over all possible estimators in some class, this one has the smallest MSE for the worst problem.
6. ...

Approximating $\mu(\underline{x})$

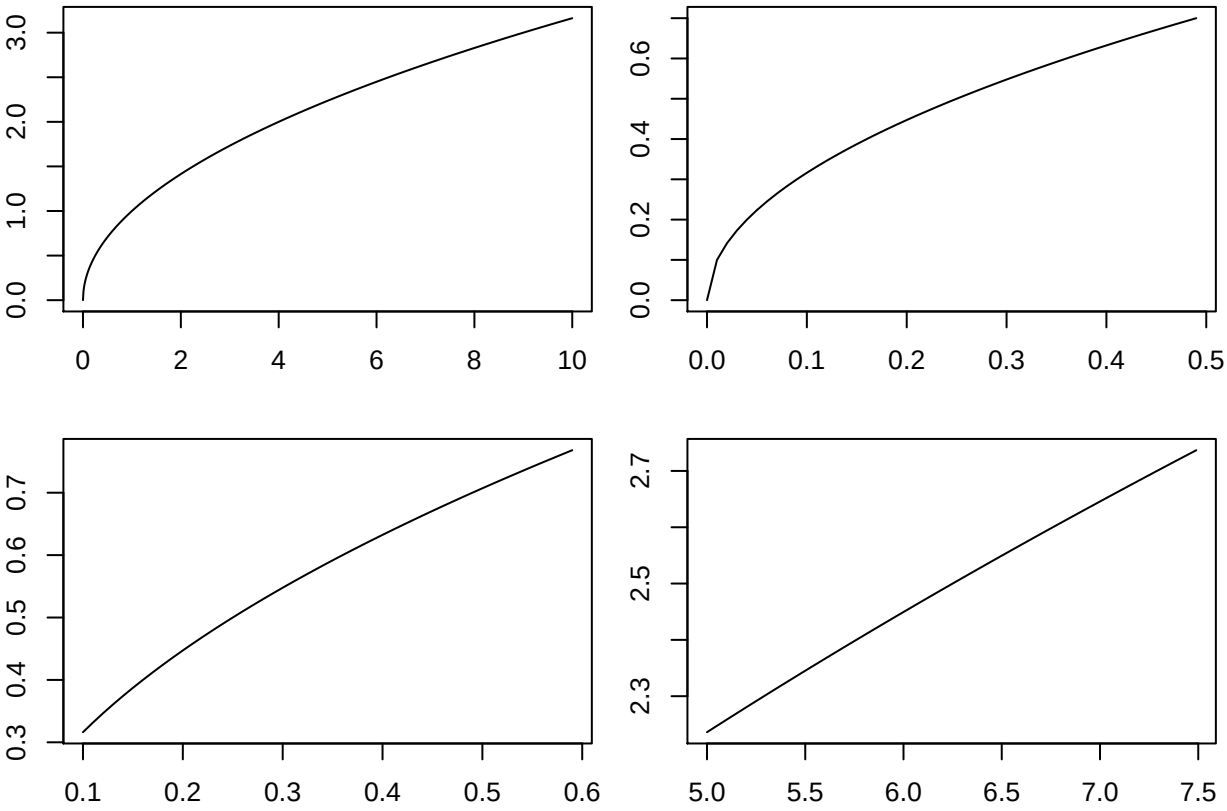
- The Taylor series expansion of the mean function $\mu(\underline{x})$ at some point u

$$\mu(\underline{x}) = \mu(\underline{u}) + (\underline{x} - \underline{u}) \cdot \nabla\mu(\underline{u}) + O(\|\underline{x} - \underline{u}\|^2)$$

- The notation $f(x) = O(g(x))$ means that for any x there exists a constant C such that $f(x)/g(x) < C$.
- More intuitively, this notation means that the remainder (all the higher order terms) are about the size of the distance between x and u or smaller.
- So as long as we are looking at points $\underline{u}$ near by $\underline{x}$, a linear approximation to $\mu(\underline{x}) = \mathbb{E}[Y \mid \underline{X} = \underline{x}]$ is reasonably accurate.

Example: Approximating $\sqrt{x}$

Each of these plots is of the $\sqrt{x}$ which is not linear, but the domain displayed is shifted.



Global Prediction Error vs. Local

- In theory, we have (if we know things about the state of nature)

$$\underline{\beta}^* = \arg\min_{\underline{\beta}} \mathbb{E}\left[\|Y - \underline{X}\underline{\beta}\|^2\right] = \text{Cov}\left[\underline{X},\ \underline{X}\right]^{-1} \text{Cov}\left[\underline{X},\ Y\right]$$

- Define $\underline{V}^{-1} = \text{Cov}\left[\underline{X},\ \underline{X}\right]^{-1}$.
- Using this optimal value β^*, what is $\text{Cov}\left[Y - \underline{X}\underline{\beta}^*,\ X\right]$?

$$\begin{aligned}
\text{Cov}\left[Y - \underline{X}\underline{\beta}^*,\ \underline{X}\right] &= \text{Cov}\left[Y,\ \underline{X}\right] - \text{Cov}\left[\underline{X}\underline{\beta}^*,\ \underline{X}\right] && \text{(Cov is linear)} \\
&= \text{Cov}\left[Y,\ \underline{X}\right] - \text{Cov}\left[\underline{X}(\underline{V}^{-1}\text{Cov}\left[\underline{X},\ Y\right]),\ \underline{X}\right] && \text{(substitute the def. of } \beta^*) \\
&= \text{Cov}\left[Y,\ \underline{X}\right] - \text{Cov}\left[\underline{X},\ \underline{X}\right]\underline{V}^{-1}\text{Cov}\left[Y,\ \underline{X}\right] \\
&= \text{Cov}\left[Y,\ \underline{X}\right] - \text{Cov}\left[Y,\ \underline{X}\right] = 0.
\end{aligned}$$

- This means the average of $Y - \underline{X}\underline{\beta} = 0$ across the entire line.
- What about locally, i.e., $E\left[Y - \underline{X}\underline{\beta}|\underline{X} = \underline{x}\right]$

Collinearity

We very often take it for granted that all predictor variables are linearly independent. What is linear independence of vectors?

Say you have model that has included weight in pounds (X_1) and weight and kilograms ($X_2 = X_1/2.2$).

$$\begin{aligned}
\widehat{\mu}(\underline{X}) &= \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p \\
&= 0X_1 + (2.2\beta_1 + \beta_2)X_2 + \cdots + \beta_p X_p \\
&= (\beta_1 + \beta_2/2.2)X_1 + 0X_2 + \cdots + \beta_p X_p \\
&= -2200X_1 + (1000 + \beta_1 + \beta_2)X_2 + \cdots + \beta_p X_p
\end{aligned}$$

When two variables are collinear, a few things happen.

1. We cannot **numerically** calculate $(\mathbf{X}^\top\mathbf{X})^{-1}$. It is rank deficient.
2. We cannot **intellectually** separate the contributions of the two variables.
3. We can (and should) drop one of them. This will not change the bias of our estimator, but it will alter our interpretations.
4. Collinearity appears most frequently with many categorical variables.
5. In these cases, software **automatically** drops one of the levels resulting in the baseline case being in the intercept. Alternately, we could drop the intercept!
6. High-dimensional problems (where we have more predictors than observations) also lead to rank deficiencies.
7. There are methods (regularizing) which attempt to handle this issue (both the numerics and the interpretability). We may have time to cover them slightly.

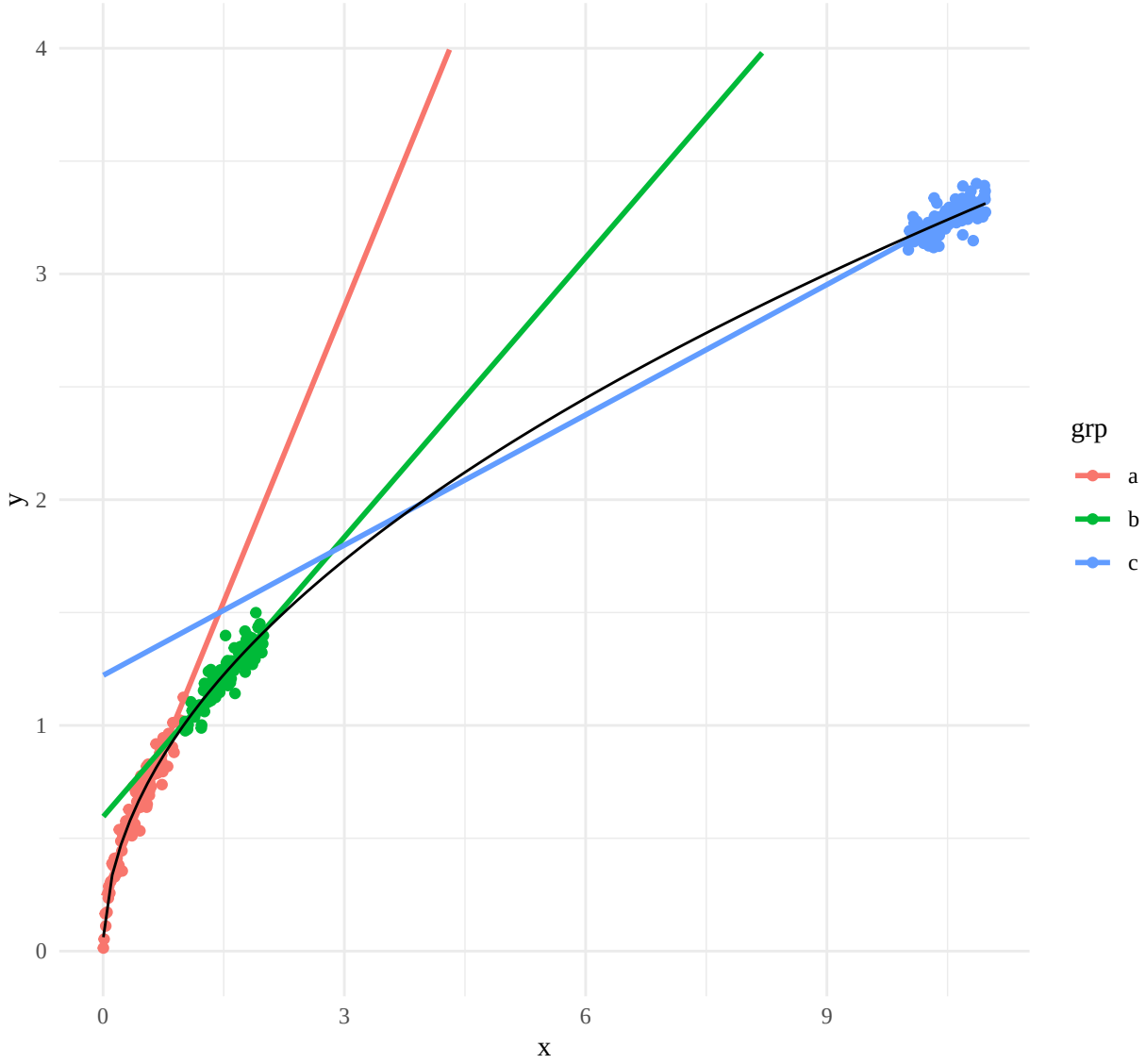
Problems with R-squared

$$R^2 = 1 - \frac{SSE}{\sum_{i=1}^n (Y_i - \overline{Y})^2} = 1 - \frac{MSE}{\frac{1}{n}\sum_{i=1}^n (Y_i - \overline{Y})^2} = 1 - \frac{SSE}{SST}$$

- This gets spit out by software
- X and Y are both normal with (empirical) correlation r, then $R^2 = r^2$
- In this nice case, it measures how tightly grouped the data are about the regression line
- Data that are tightly grouped about the regression line can be predicted accurately by the regression line.
- Unfortunately, the implication does not go both ways.
- High R^2 can be achieved in many ways, same with low R^2
- You should just ignore it completely (and the adjusted version), and encourage your friends to do the same

High R-squared with non-linear relationship

```
genY <- function(X, sig) Y = sqrt(X)+sig*rnorm(length(X))
sig=0.05; n=100
X1 = runif(n,0,1)
X2 = runif(n,1,2)
X3 = runif(n,10,11)
df = data.frame(x=c(X1,X2,X3), grp = rep(letters[1:3],each=n))
df$y = genY(df$x,sig)
ggplot(df, aes(x,y,color=grp)) + geom_point() +
  geom_smooth(method = 'lm', fullrange=TRUE,se = FALSE) +
  ylim(0,4) + stat_function(fun=sqrt,color='black')
```



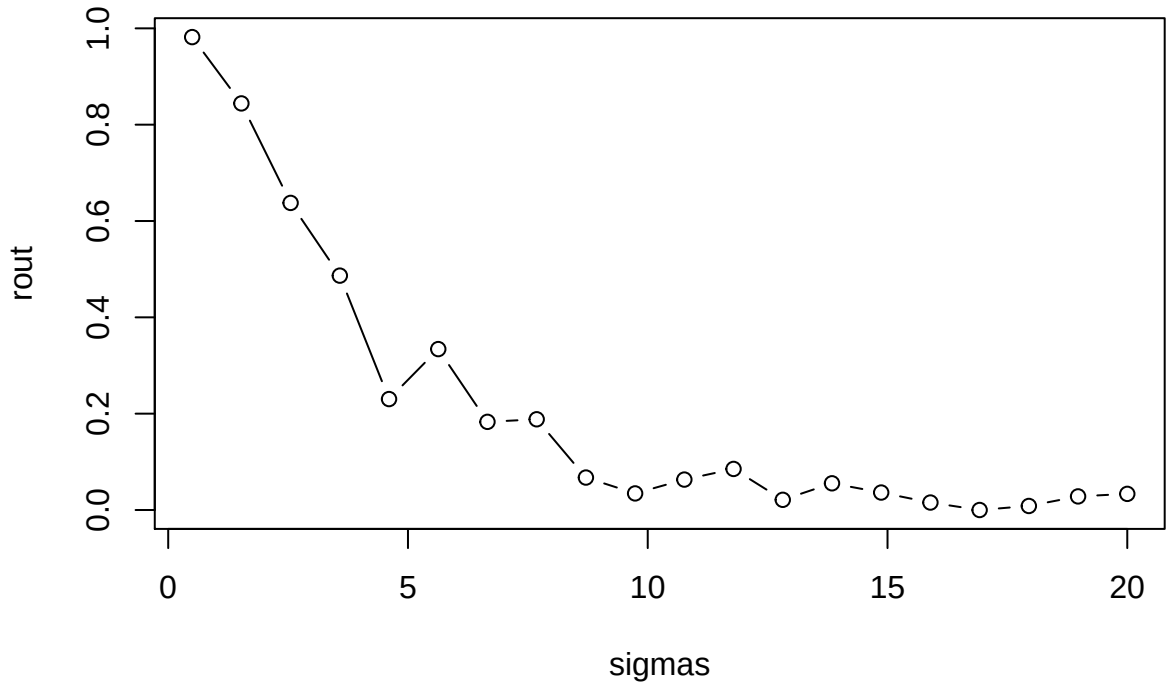

```
df %>% group_by(grp) %>% summarise(rsq = summary(lm(y~x))$r.sq)
```

```
## # A tibble: 3 x 2
##   grp     rsq
##   <fct> <dbl>
## 1 a     0.901
## 2 b     0.832
## 3 c     0.551
```

Details with R^2

R-Squared can be arbitrarily low when the model is completely correct.

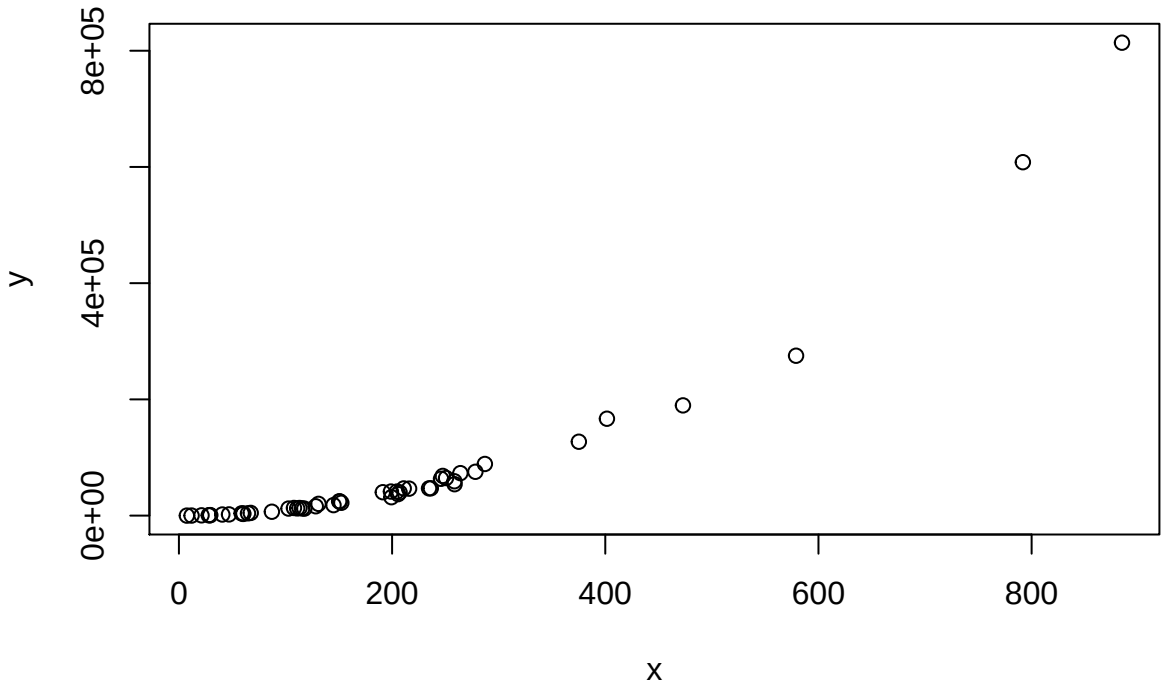
```
  r2.0 <- function(sig){
  x <- seq(1,10,length.out = 100)         # our predictor
  y <- 2 + 1.2*x + rnorm(100,0,sd = sig) # our response; a function of x plus some random noise
  summary(lm(y ~ x))$r.squared            # print the R-squared value
}

sigmas <- seq(0.5,20,length.out = 20)
rout <- sapply(sigmas, r2.0)              # apply our function to a series of sigma values
plot(rout ~ sigmas, type="b")
```



And Then...

R-squared can be arbitrarily close to 1 when the model is totally wrong.

```
set.seed(1)
x <- rexp(50,rate=0.005)                    # our predictor is data from an exponential distribution
y <- (x-1)^2 * runif(50, min=0.8, max=1.2)   # non-linear data generation
plot(x,y)                        # clearly non-linear
```



And So On...

R-squared cannot be compared across datasets, even when the same model is used.

```
x <- seq(1,10,length.out = 100)
set.seed(1)
y <- 2 + 1.2*x + rnorm(100,0,sd = 0.9)
mod1 <- lm(y ~ x)
summary(mod1)$r.squared
```

```
## [1] 0.9383379
```

```
sum((fitted(mod1) - y)^2)/100 # Mean squared error
```

```
## [1] 0.6468052
```

```
x <- seq(1,2,length.out = 100)       # new range of x
set.seed(1)
y <- 2 + 1.2*x + rnorm(100,0,sd = 0.9)
mod1 <- lm(y ~ x)
summary(mod1)$r.squared
```

```
## [1] 0.1502448
```

```
sum((fitted(mod1) - y)^2)/100        # Mean squared error
```

```
## [1] 0.6468052
```

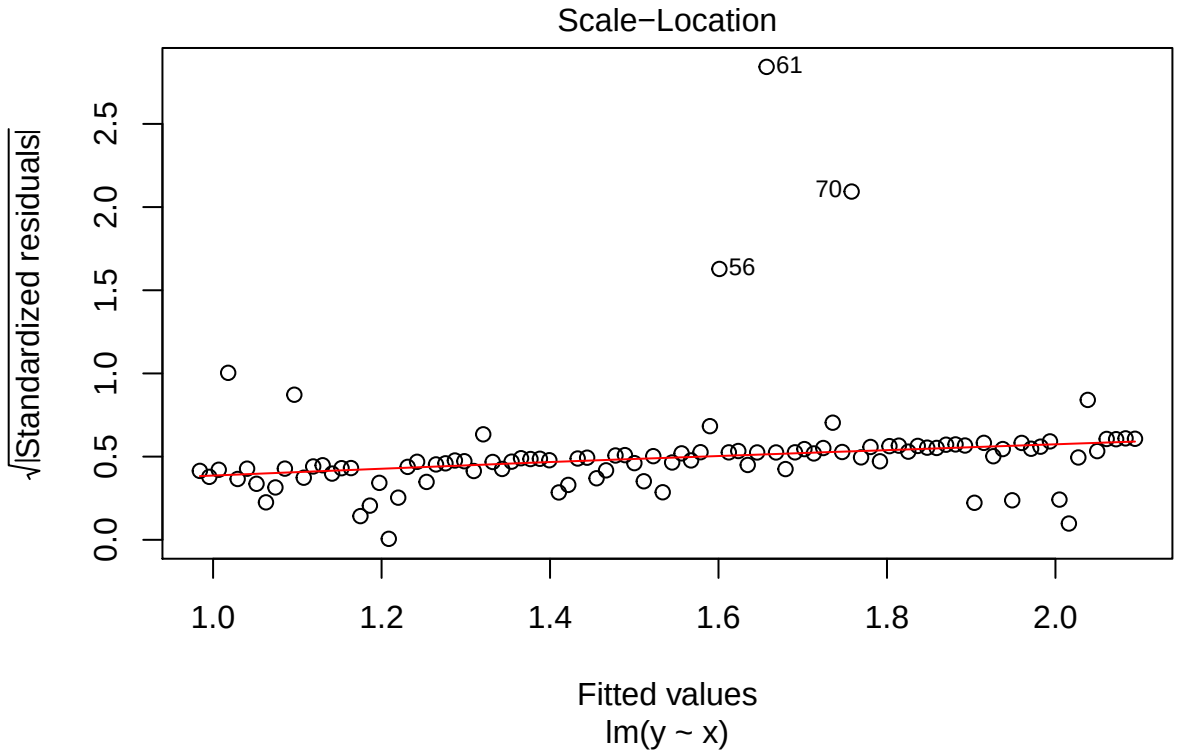
Last one, I promise

R-squared cannot be compared between a model with untransformed Y and one with transformed Y, or between different transformations of Y. R-squared can easily go down when the model assumptions are better fulfilled.

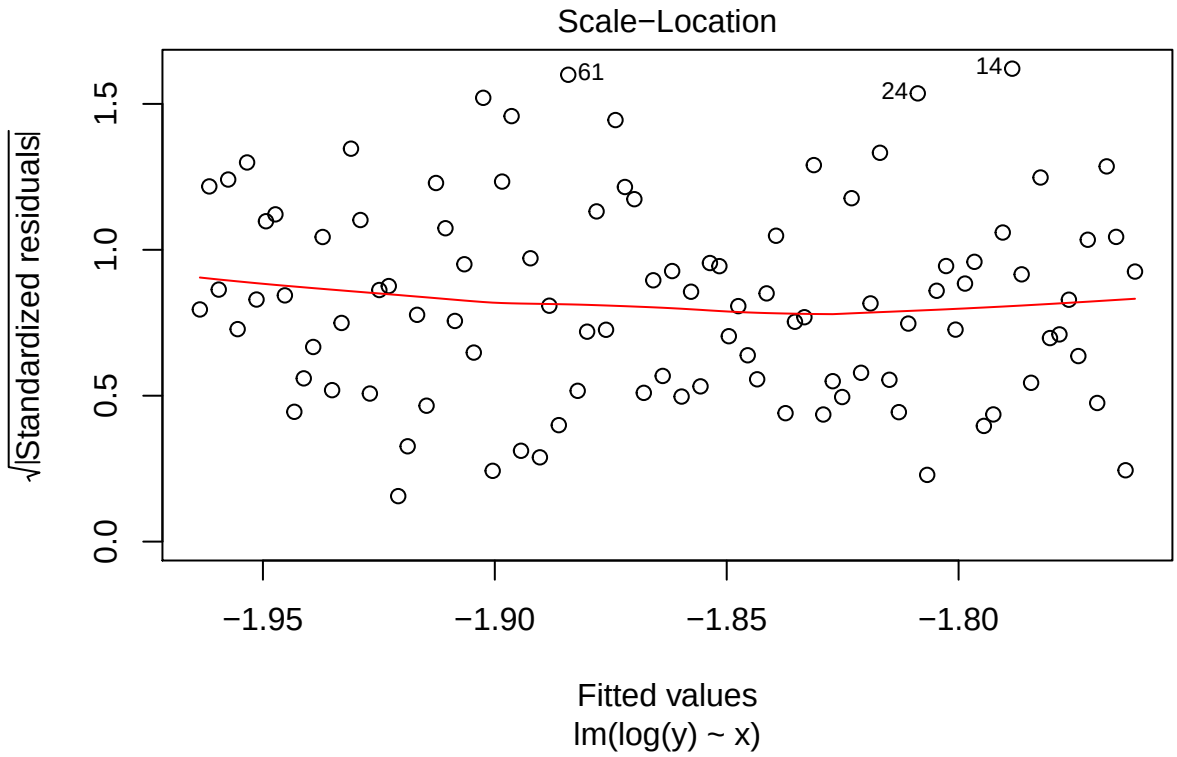
```r
x <- seq(1,2,length.out = 100)
set.seed(1)
y <- exp(-2 - 0.09*x + rnorm(100,0,sd = 2.5))
summary(lm(y ~ x))$r.squared
```

```
## [1] 0.003281718
```

```r
plot(lm(y ~ x), which=3)
```




```r
plot(lm(log(y)~x),which = 3)
```




```
summary(lm(log(y)~x))$r.squared
```

```
## [1] 0.0006921086
```