STAT 224 Lecture 13
Chapter 6 Transformation of Variables

Yibi Huang

Why Transform Variables?

Why Transform Variables?

We transform variables (including predictors and responses) primarily for two reasons:

- to solve the non-linearity problem
- to solve the non-constant variability problem
    - Variance-Stabilizing Transformation
    - Box-Cox Method

Linear and Nonlinear Models

Recall linear models are **linear in the parameters**, not predictors.

All of the following are linear models:

- $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \varepsilon$
- $Y = \beta_0 + \beta_1 \log(X) + \varepsilon$
- $Y = \beta_0 + \beta_1 \sqrt{X} + \varepsilon$

Whereas the following is not a linear model since it's not linear in β_1.

$$Y = \beta_0 + \exp(\beta_1 X) + \varepsilon$$

Linearizable Models

Some nonlinear models can be turn into linear model after transforming variables

- Ex1: exponential growth or decay models

$$Y = \alpha e^{\beta X}.$$

  Taking the log of both sides yields

$$\log(Y) = \log(\alpha) + \beta X.$$

- Ex2: Learning theory in psychology states that the time to perform a task (T_i) on the i occasion follows

$$T_i = \alpha\beta^i, \quad \alpha > 0, \quad 0 < \beta < 1$$

  Taking the log of both sides yields

$$\log(T_i) = \log(\alpha) + \log(\beta)i.$$

Linearizable Models (Table 6.1 on p.165)

Function	Transformation	Linear Form
$Y = \alpha X^{\beta}$	$Y' = \log Y, X' = \log X$	$Y' = \log \alpha + \beta X'$
$Y = \alpha e^{\beta X}$	$Y' = \log Y$	$Y' = \log \alpha + \beta X$
$Y = \alpha + \beta \log X$	$X' = \log X$	$Y = \alpha + \beta X'$
$Y = \frac{X}{\alpha X - \beta}$	$Y' = \frac{1}{Y}, X' = \frac{1}{X}$	$Y' = \alpha - \beta X'$
$Y = \frac{e^{\alpha+\beta X}}{1+e^{\alpha+\beta X}}$	$Y' = \log \frac{Y}{1-Y}$	$Y' = \alpha + \beta X$

These nonlinear models can be turned linear after transformation and the tools in MLR can still be applied.

Some Nonlinear Models Cannot Be Linearized

Ex.

- $Y = \delta + \alpha\beta^X$
- $Y = \alpha_1 e^{\beta_1 X} + \alpha_2 e^{\beta_2 X}$

The strictly nonlinear models (i.e., those not linearizable by variable transformation) require very different methods. (not covered in STAT 224)

Transformations to Achieve Linearity

Example: Bacteria Deaths Due to X-Ray Radiation (p.168)

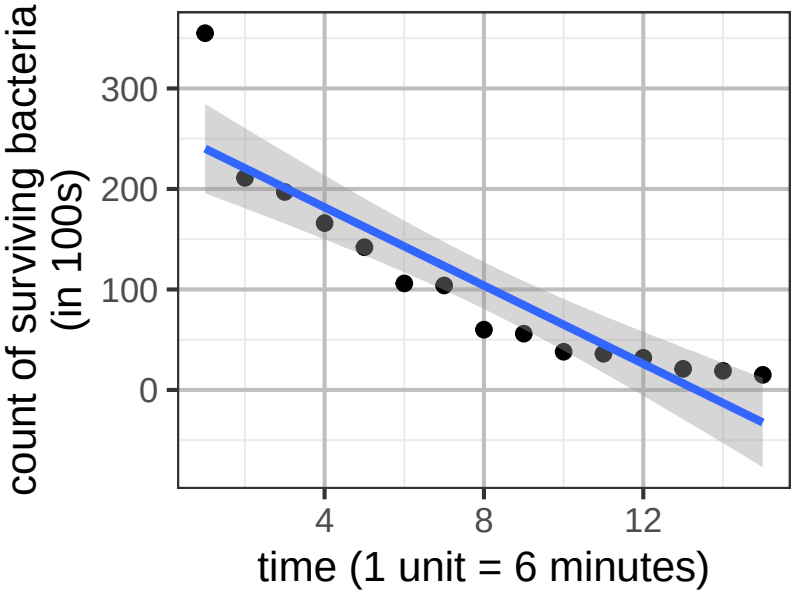
Data: http://www.stat.uchicago.edu/~yibi/s224/data/P168.txt

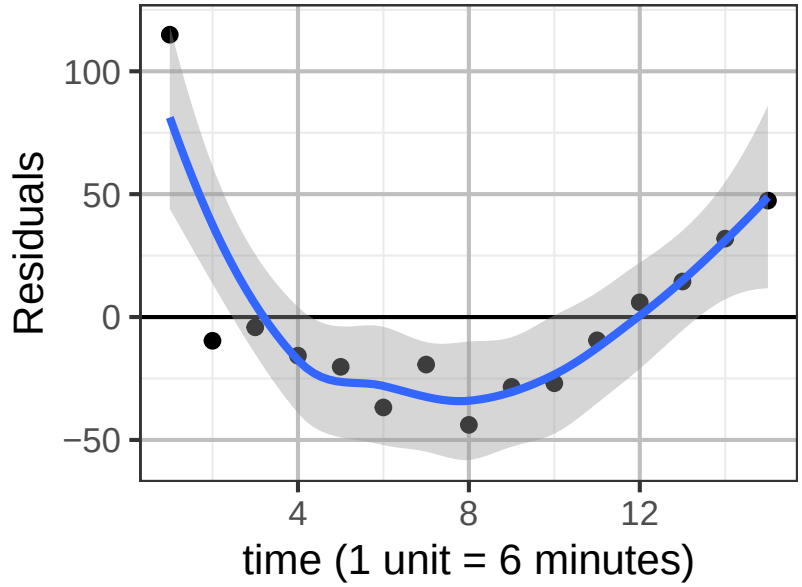
```
bact = read.table("P168.txt", header=T)
```

- t = time (1 unit = 6 minutes)
- N_t = n_t = the number of surviving bacteria (in 100s) following exposure to 200-kilo-volt X-rays in after t units of time

If we blindly fit an SLR model `lm1 = lm(N_t ~ t, data=bact)`

```
library(ggplot2)
ggplot(bact, aes(x=t, y=N_t))+geom_point()+
  geom_smooth(method='lm')+ xlab("time (1 unit = 6 minutes)")+
  ylab("count of surviving bacteria\n(in 100s)")
lm1 = lm(N_t ~ t, data=bact)
ggplot(bact, aes(x=t, y=lm1$res))+geom_point() +
  labs(x="time (1 unit = 6 minutes)", y="Residuals")+
  geom_hline(yintercept=0) + geom_smooth()
```





When not knowing what transformation to make, we would begin by looking at the scatterplot.

- In this case, the scatterplot is obviously non-linear.

In some cases, non-linearity may not be obvious in the scatterplot. Should always check the residual plot as well for this reason.

- For this example, we see that non-linearity is more obvious in the residual plot than in the scatterplot.

Example: Bacteria Deaths Due to X-Ray Radiation

According to theory, we expect a exponential decay in the count of bacteria in time:

$$n_t = n_0 e^{\beta_1 t}, \quad \text{where} \begin{cases} n_0 = \text{initial population size} \\ \beta_1 = \text{decay rate} \end{cases}$$

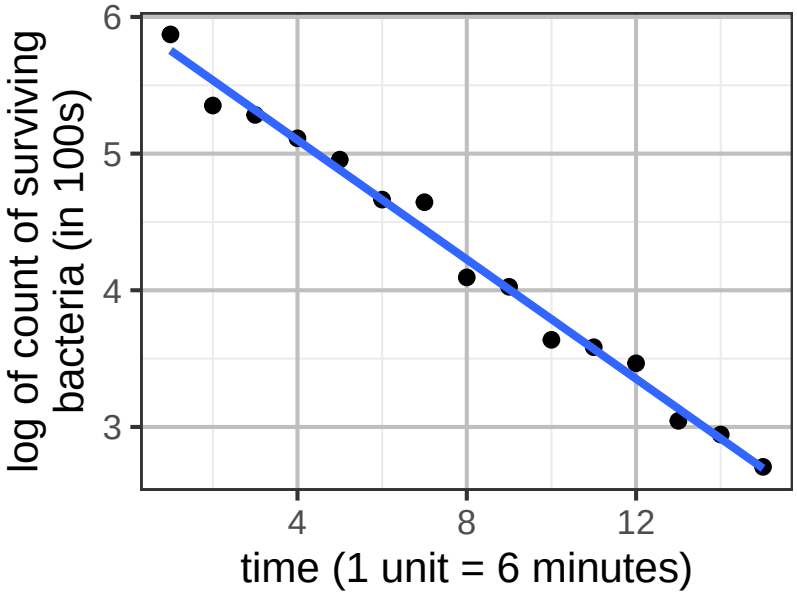
Taking log of both sides, we get

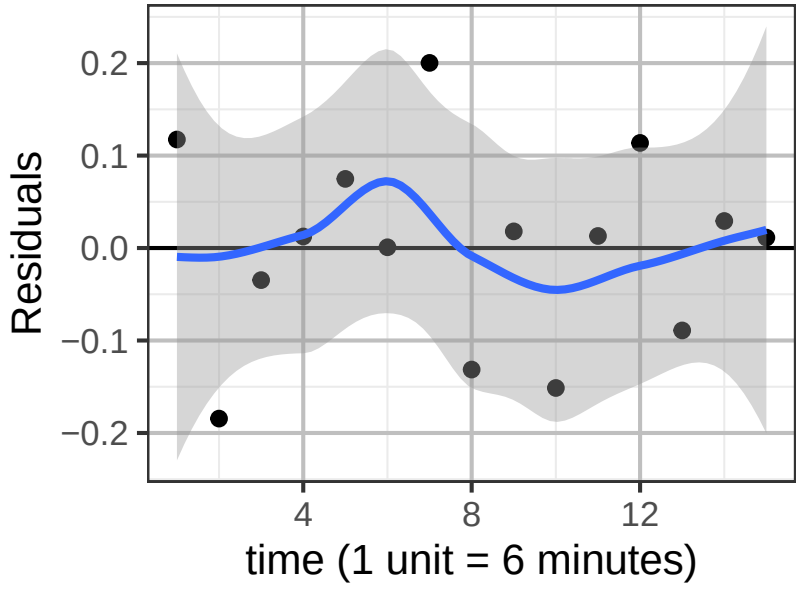
$$\log n_t = \log n_0 + \beta_1 t = \beta_0 + \beta_1 t,$$

which suggests that we regress $\log n_t$ against t.

```
lm2 = lm(log(N_t) ~ t, data=bact)
```

```
ggplot(bact, aes(x=t, y=log(N_t)))+geom_point()+
  geom_smooth(method='lm', se=F)+xlab("time (1 unit = 6 minutes)")+
  ylab("log of count of surviving\nbacteria (in 100s)")
ggplot(bact, aes(x=t, y=lm2$res))+geom_point() +
  xlab("time (1 unit = 6 minutes)")+ ylab("Residuals")+
  geom_hline(yintercept=0) + geom_smooth()
```





- Scatterplot shows transformation achieves linearity
- Residual plot shows no clear violation of model assumptions

Interpretation of the Exponential Decay Model

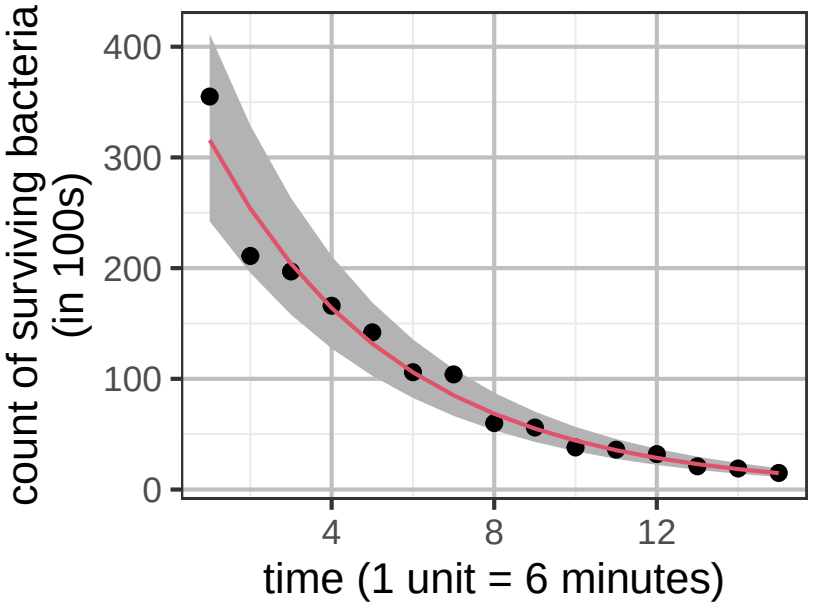
For the exponential decay model $n_t = n_0 e^{\beta_1 t}$, for every extra unit of time, the number of surviving bacteria becomes e^{β_1} times as large.

```
lm2 = lm(log(N_t) ~ t, data=bact)
summary(lm2)$coef
             Estimate Std. Error t value  Pr(>|t|)
(Intercept)   5.9732   0.059778   99.92 3.786e-20
t            -0.2184   0.006575  -33.22 5.860e-14
confint(lm2, "t")
    2.5 %  97.5 %
t -0.2326 -0.2042
1-exp(confint(lm2, "t"))
   2.5 % 97.5 %
t 0.2076 0.1847
```

Every 6 minutes, the number of surviving bacteria is estimated to decrease by $1 - e^{-0.2184} \approx 1 - 0.804 = 19.6\%$
(95% CI is $1 - e^{-0.2326} \approx 18.5\%$ to $1 - e^{-0.2042} \approx 20.8\%$).

Back to the Original Scale

```
pred.log = predict(lm2, data.frame(t = 1:15), interval="prediction")
pred.orig = exp(pred.log)
ggplot(bact, aes(x=t, y=N_t))+
geom_ribbon(aes(ymin=pred.orig[,2],ymax=pred.orig[,3]),fill="grey70")+
  geom_point()+geom_line(aes(y=pred.orig[,1]), col=2)+
  xlab("time (1 unit = 6 minutes)")+
  ylab("count of surviving bacteria\n(in 100s)")
```



About the Log-Transformation

Logarithm is the Most Commonly Used Transformation

- When the size of error is proportional to the mean, take log

$$Y = f(X_1, \ldots, X_p)(1+\varepsilon) \quad \Rightarrow \quad \log(Y) = \log f(X_1, \ldots, X_p) + \log(1+\varepsilon)$$

- Rule of thumb #1: if a variable is about *amount of money*, take log
  - Ex1: Education Expenditure data in HW4
    Both `y` = Per capita expenditure on public education,
    and `x1` = Per capita personal income,
    are log-transformed in HW4
  - Ex2: `Income2005` in the NLSY data
  - Ex3: `price` in the diamonds data in `L09.pdf`
- Rule of thumb #2: if a variable represents the **concentration** of something, take log
  - e.g., concentration of chemical in the blood, etc
- When the values of a variable varies by several order of magnitude, (e.g. some are 10 or 100 times larger than others), take log

Interpretation of Log-Transformed Variables

- $\log(Y) = \beta_0 + \beta_1 X \quad \Rightarrow \quad Y = e^{\beta_0} e^{\beta_1 X}$
  When X is increased by 1, Y becomes e^{β_1} times as large
- $\log(Y) = \beta_0 + \beta_1 \log(X) \quad \Rightarrow \quad Y = e^{\beta_0} X^{\beta_1}$
  - When X is doubled ($X \to 2X$), Y becomes 2^{β_1} times as large
  - In Economics, β_1 in the log-log model $\log(Y) = \beta_0 + \beta_1 \log(X)$ is called **Elasticity** since

$$Y = e^{\beta_0} X^{\beta_1} \Rightarrow \frac{dY}{dX} = \beta_1 e^{\beta_0} X^{\beta_1 - 1} = \beta_1 \frac{Y}{X} \Rightarrow \boxed{\frac{dY}{Y} = \beta_1 \frac{dX}{X}}$$

  This means a 1% increase in X ($dX/X = 1\% = 0.01$) would lead to a $\beta_1\%$ increase in Y ($dY/Y = \beta_1 \times 0.01$)

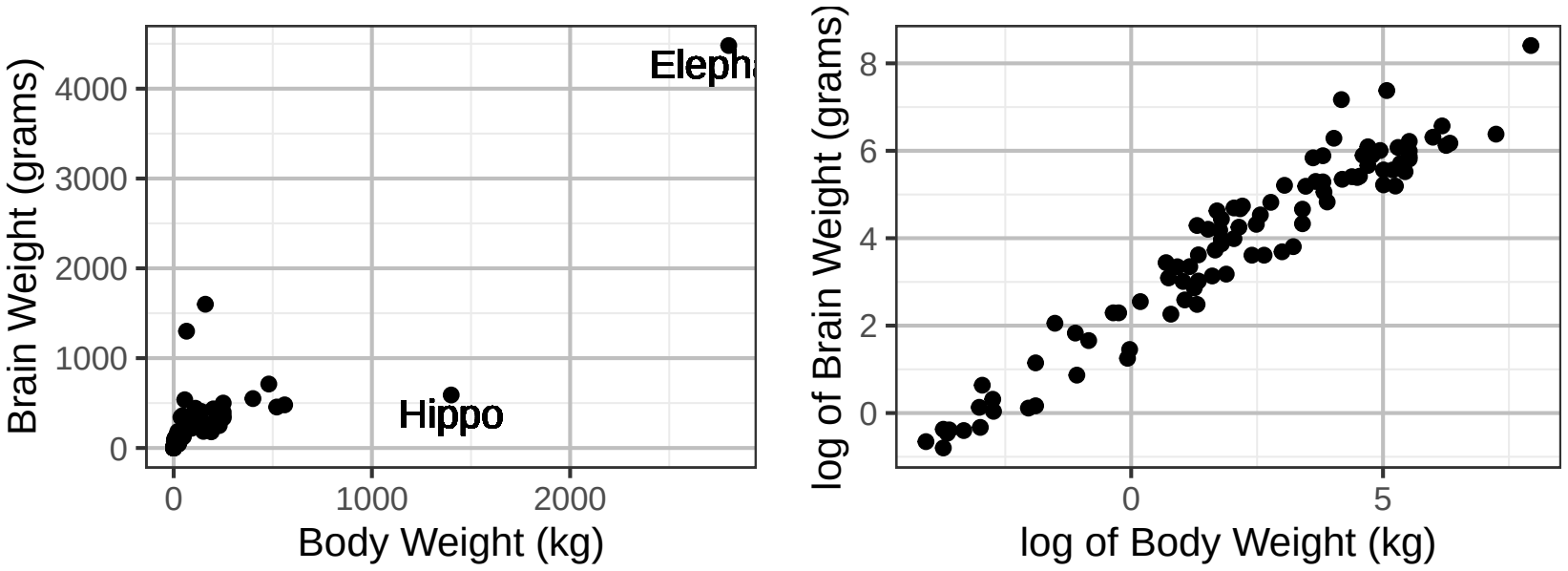
Transformations to Reduce Skewness

Why Worry About Skewness?

- If the response is skewed, the normality assumption of the noise ε is probably violated
  - non-normality is not a big problem if it's the only issue (no non-linearity or non-constant variability issues), may leave it alone.
  - e.g., in the NLSY data, `log(Income2005)` is left-skewed
- If a predictor is highly-skewed, there might be extreme outliers or influential points. Transforming the predictor might make the extreme outliers less extreme and reduce the impact of influential points.
  - i.e., when there exist outliers, try transforming variables

Example: Brain and Body Weight of Mammals

Data: http://www.stat.uchicago.edu/~yibi/s224/data/mammals.txt



Before transformation, both `Brain` weight and `Body` weight are highly right-skewed.

Transformations to Reduce Skewness

Skewness can often be ameliorated by a power transformation.

$$f_\lambda(y) = \begin{cases} y^\lambda, & \text{if } \lambda \neq 0 \\ \log(y), & \text{if } \lambda = 0. \end{cases}$$

- If **right-skewned**, try taking square root, logarithm, or other powers $\lambda < 1$

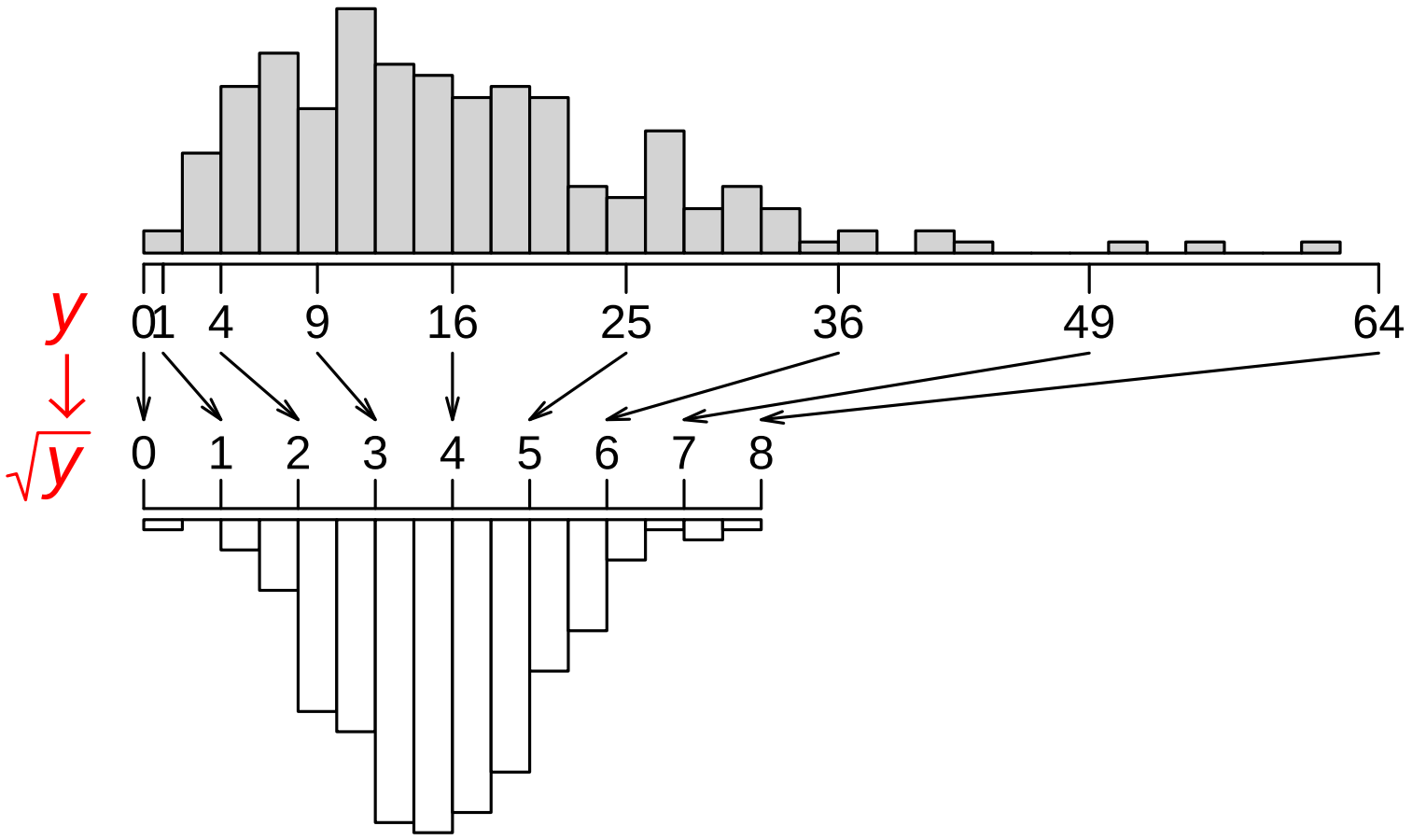
$$y \longrightarrow 1/y, \log(y), \sqrt{y}, \text{or } y^\lambda \text{ with } \lambda < 1$$

- If **left-skewned**, try squaring, cubing, or other powers $\lambda > 1$

$$y \longrightarrow y^2, y^3, \text{or } y^\lambda \text{ with } \lambda > 1$$

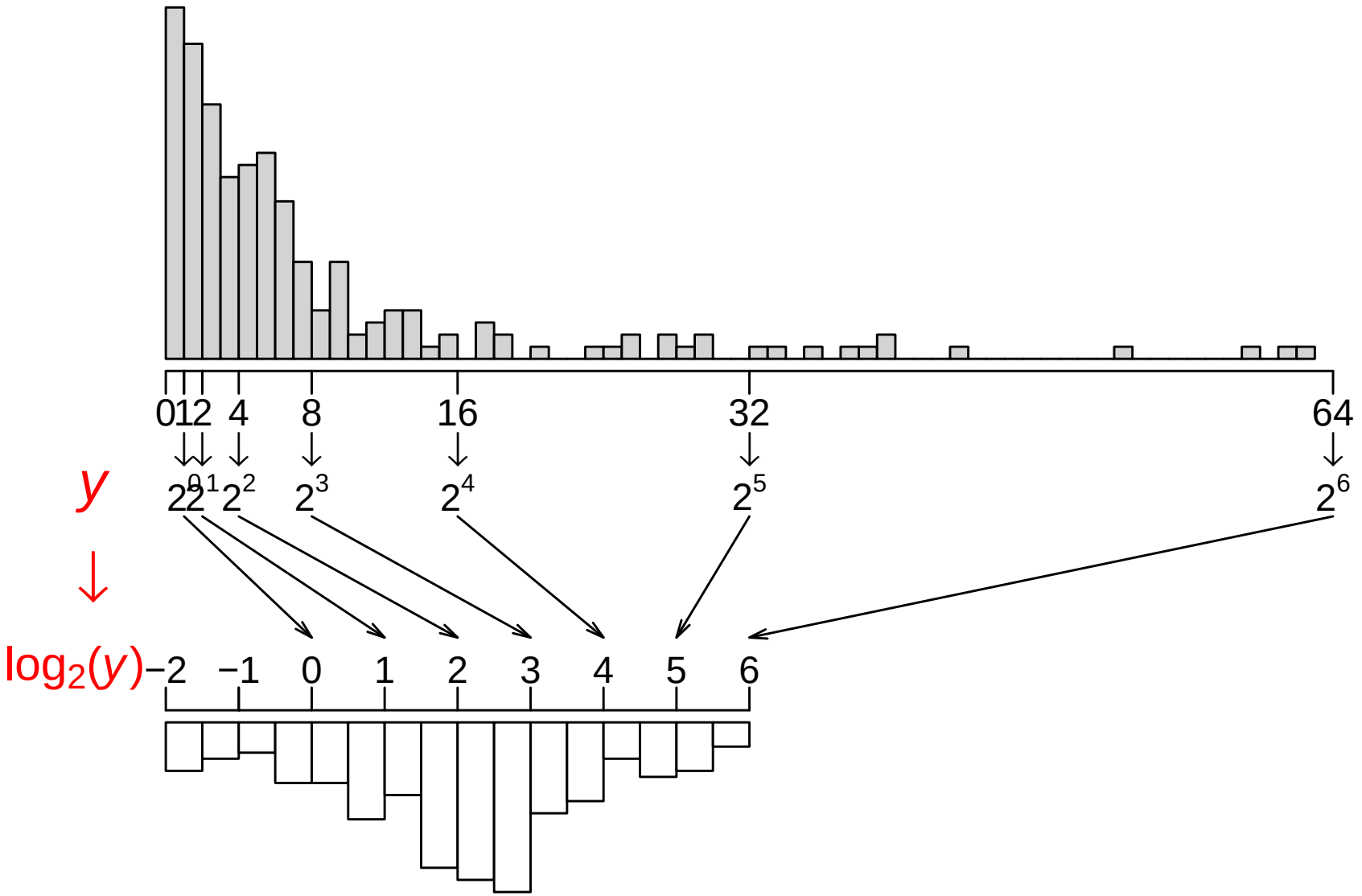
Square-Root Transformation Can Reduce Right-Skewness

The square-root transformation can shorten the upper tail and extend the lower tail, of a distribution and hence can reduce right-skewness.



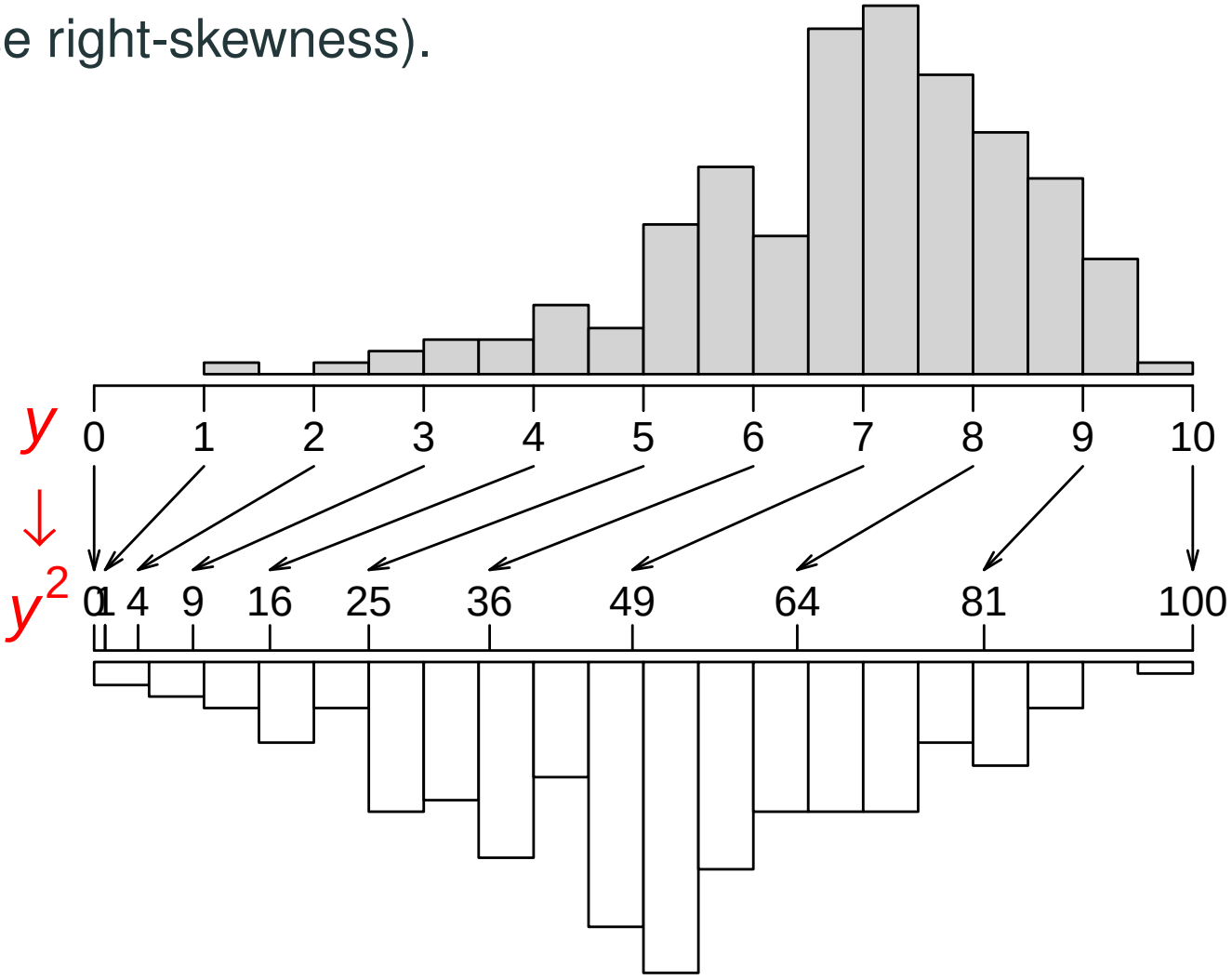
Log Transformation Reduces Right-Skewness Even More!

Logarithm can shorten the upper tail and extend the lower tail even more



Square Transformation Can Reduce Left-Skewness

The square transformation ($y \rightarrow y^2$) can extend the upper tail and shorten the lower tail, and hence can reduces left-skewness (and increase right-skewness).



Transformations to Stablize Variance

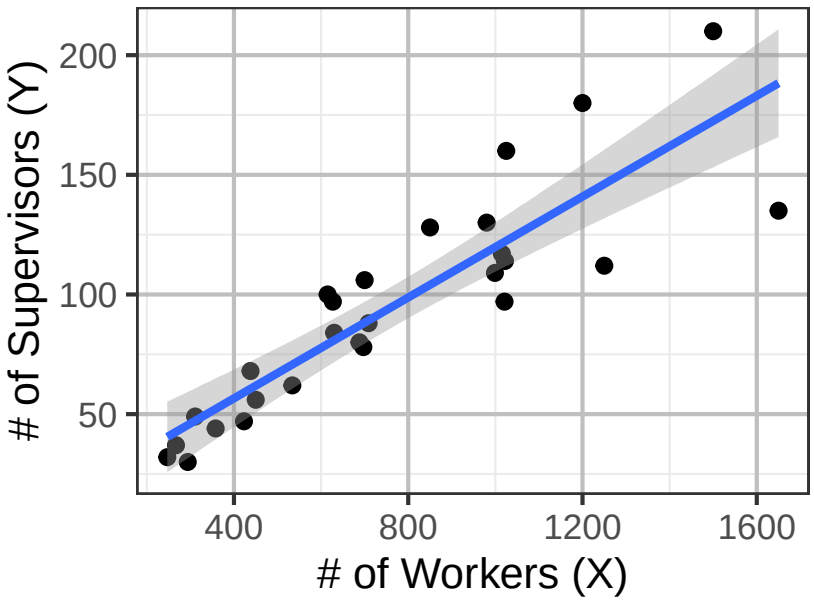
Supervisor/Employee Data (p.176)

Data: http://www.stat.uchicago.edu/~yibi/s224/data/P176.txt

X = # of Supervised Workers

Y = # of Supervisors in 27 Industrial Establishments

```
supvis = read.table("P176.txt", h=T)
ggplot(supvis, aes(x=X, y=Y))+geom_point()+geom_smooth(method='lm')+
  labs(x="# of Workers (X)", y="# of Supervisors (Y)")
```



If we blindly fit an SLR model `lm1 = lm(Y ~ X, data=supvis)`, here is the residual plot.

```
lm1 = lm(Y ~ X, data=supvis)
ggplot(supvis, aes(x=X, y=lm1$res))+geom_point() +
  xlab("# of Supervised Workers (X)")+
  ylab("Residuals")+ geom_hline(yintercept=0)
```

We see

- non-linearity
- heteroscedasticity (non-constant variance) Specifically, variance increases with fitted values

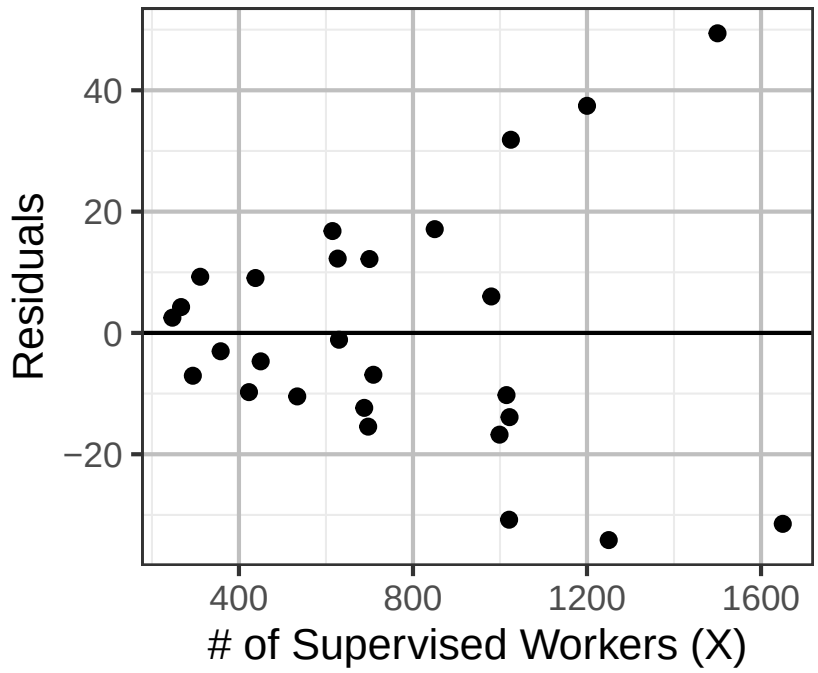
If we just deal with the non-linearity by adding a quadratic term X^2 in the model, here is the residual plot

```
lm2 = lm(Y ~ X + I(X^2), data=supvis)
```

```
ggplot(supvis, aes(x=X, y=lm2$res))+geom_point() +
  xlab("# of Supervised Workers (X)")+
  ylab("Residuals")+ geom_hline(yintercept=0)
```

Why heteroscedasticity is a problem?

Ans: Confidence intervals and prediction intervals would be
too wide at small X
too narrow at large X



Variance-Stabilizing Transformation

If the SD σ of noise (residuals) changes the mean μ of the response (the fitted values), you can try a **variance-stabilizing transformation** of the response to make the variance (closer to) constant.

- if the SD is proportional to the fitted value, then

$$y \to \log(y)$$

- if the SD is proportional to $\sqrt{\text{the fitted value}}$, i.e., the variance is proportional to the fitted value, then

$$y \to \sqrt{y}$$

- In general, if the SD σ is proportional to (the fitted values)$^\alpha$, then the variance-stabilizing transformation is

$$y \to \begin{cases} y^{1-\alpha} & \text{for } \alpha \neq 1 \\ \log(y) & \text{for } \alpha = 1 \end{cases}$$

Box-Cox Method

Box-Cox method is an automatic procedure to select the "best" power λ that make the residuals of the model

$$Y^{\lambda} = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \varepsilon$$

closest to *normal* and *constant variability*.

- We usually round the optimal λ to a *convenient power* like

$$-1,\ -\frac{1}{2},\ -\frac{1}{3},\ 0,\ \frac{1}{3},\ \frac{1}{2},\ 1,\ 2, \ldots$$

  since the practical difference of $y^{0.5827}$ and $y^{0.5}$ is usually small, but the square-root transformation is much easier to interpret.

Box-Cox Method

Box-Cox method is an automatic procedure to select the "best" power λ that make the residuals of the model

$$Y^{\lambda} = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \varepsilon$$

closest to *normal* and *constant variability*.

- We usually round the optimal λ to a *convenient power* like

$$-1,\ -\frac{1}{2},\ -\frac{1}{3},\ 0,\ \frac{1}{3},\ \frac{1}{2},\ 1,\ 2, \ldots$$

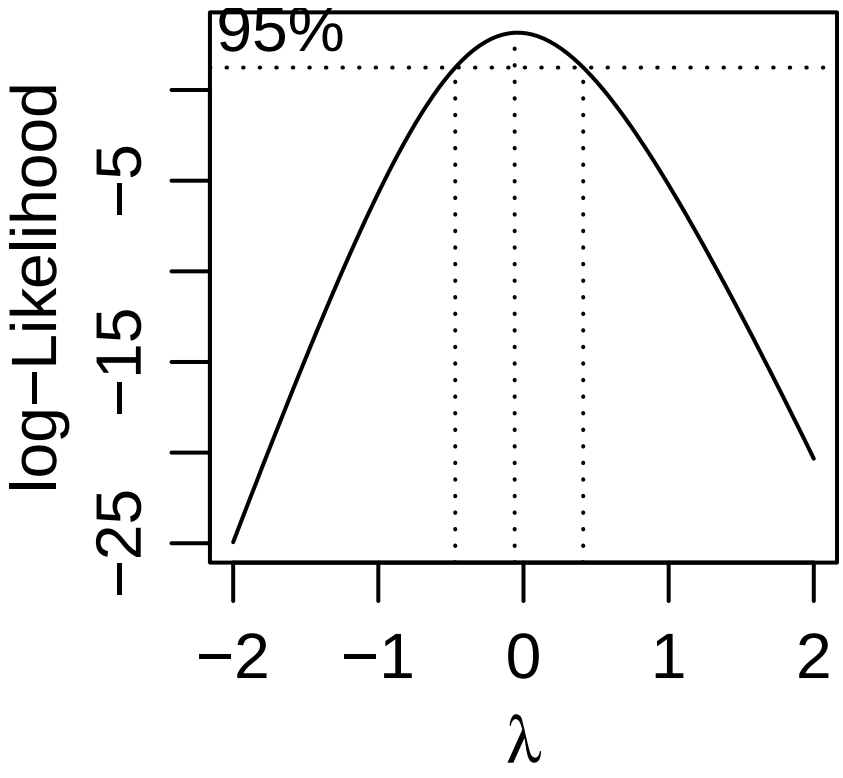
  since the practical difference of $y^{0.5827}$ and $y^{0.5}$ is usually small, but the square-root transformation is much easier to interpret.
- A confidence interval for the optimal λ can also be obtained (formula and details omitted).
  We usually select a convenient power λ^* in this C.I.

Box-Cox Method for Supervisor/Employee Data

```
library(MASS)
boxcox(lm(Y ~ X + I(X^2), data=supvis))
```

The middle dash line marks the optimal λ, the right and left dash line mark the 95% C.I. for the optimal λ.

For the plot, we see the optimal λ is around 0.1, and the 95% C.I. contains 0. For simplicity, we pick $\lambda = 0$ and use log of `fev` as our response.

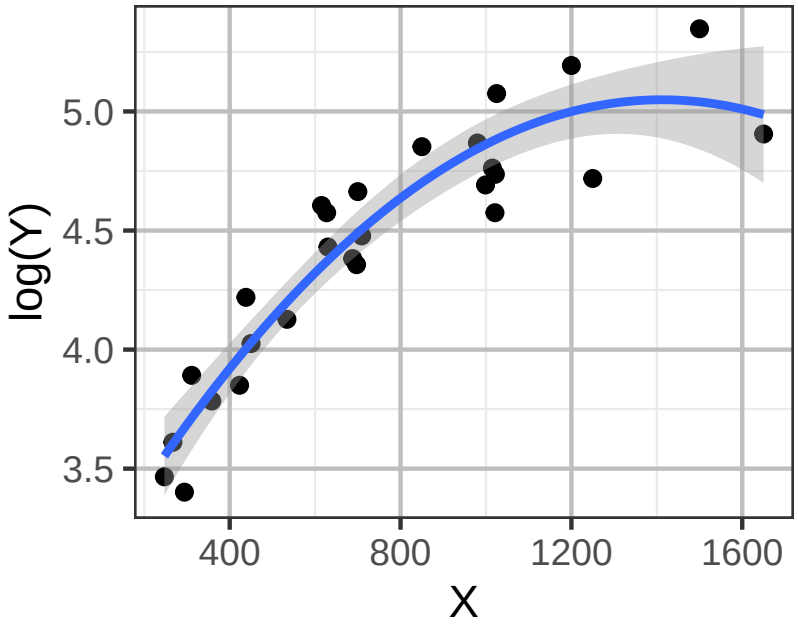


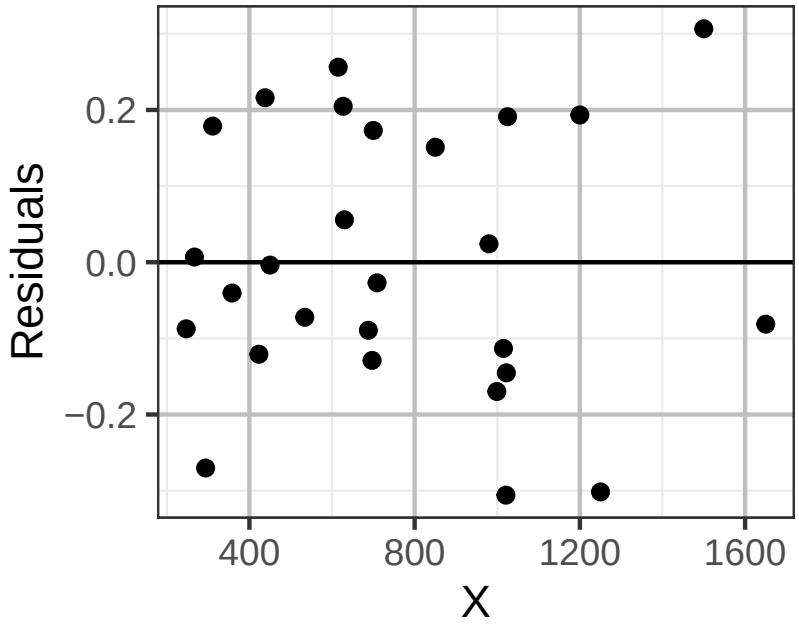
Box-Cox says take log of Y.

```
lm3 = lm(log(Y) ~ X + I(X^2), data=supvis)
summary(lm3)$coef
                Estimate   Std. Error t value  Pr(>|t|)
(Intercept)  2.851600363 0.1566401294  18.205 1.496e-15
X            0.003112674 0.0003989301   7.803 4.898e-08
I(X^2)      -0.000001102 0.0000002238  -4.925 5.027e-05
```

Left figure: scatterplot of X v.s. $\log(Y)$, overlay the fitted curve of `log(Y) ~ X + I(X^2)`

Right figure: residual plot for the model `log(Y) ~ X + I(X^2)`.



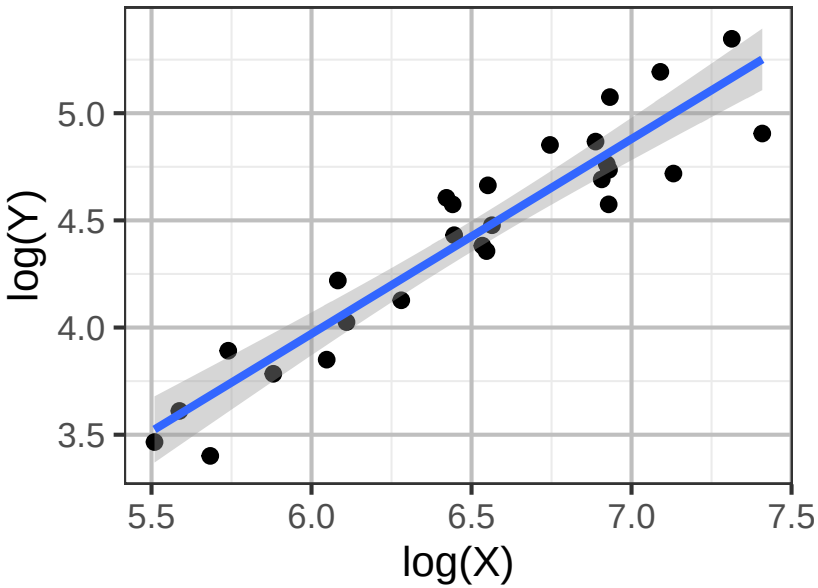


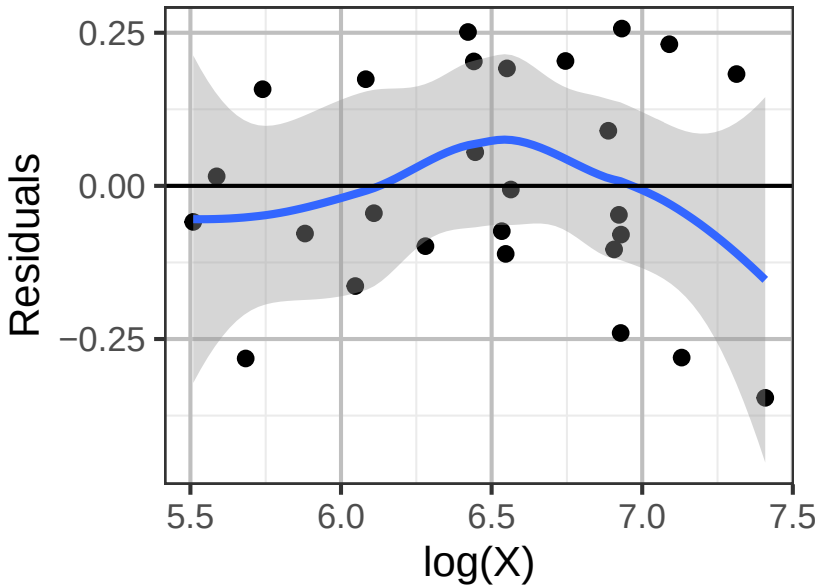
- No clear nonlinearity or heteroscadasticity.
- The quadratic term X^2 is significant
  (R output on previous page)
- Relation between X and $\log(Y)$ is **NOT monotone** based on the quadratic model `log(Y) ~ X + I(X^2)`

When X & Y are Both Log-Transformed

We can try taking log of both X and Y.

```
lm4 = lm(log(Y) ~ log(X), data=supvis)
```



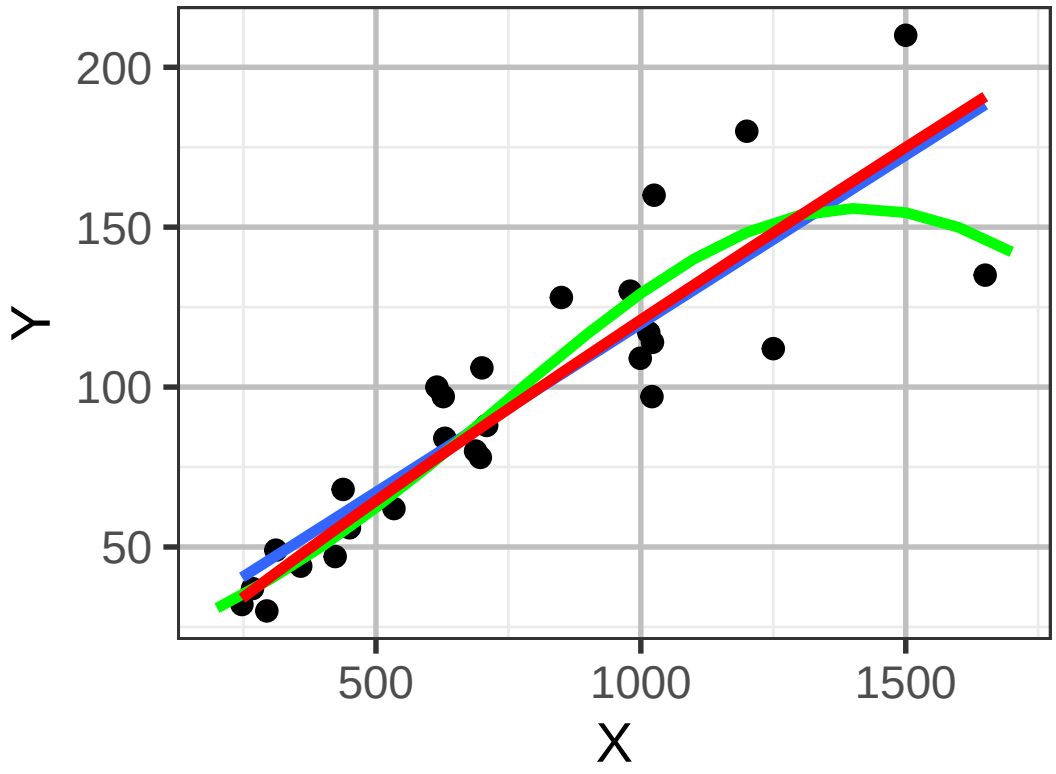


- Also an acceptable model based on the scatterplot and the residual plot
- Better interpretation than the quadratic model `log(Y) ~ X + I(X^2)` since `log(Y)~log(X)` assumes a monotone relation

```
summary(lm4)$coef
             Estimate Std. Error t value  Pr(>|t|)
(Intercept)  -1.4846    0.43544  -3.409 2.215e-03
log(X)        0.9092    0.06673  13.625 4.508e-13
```

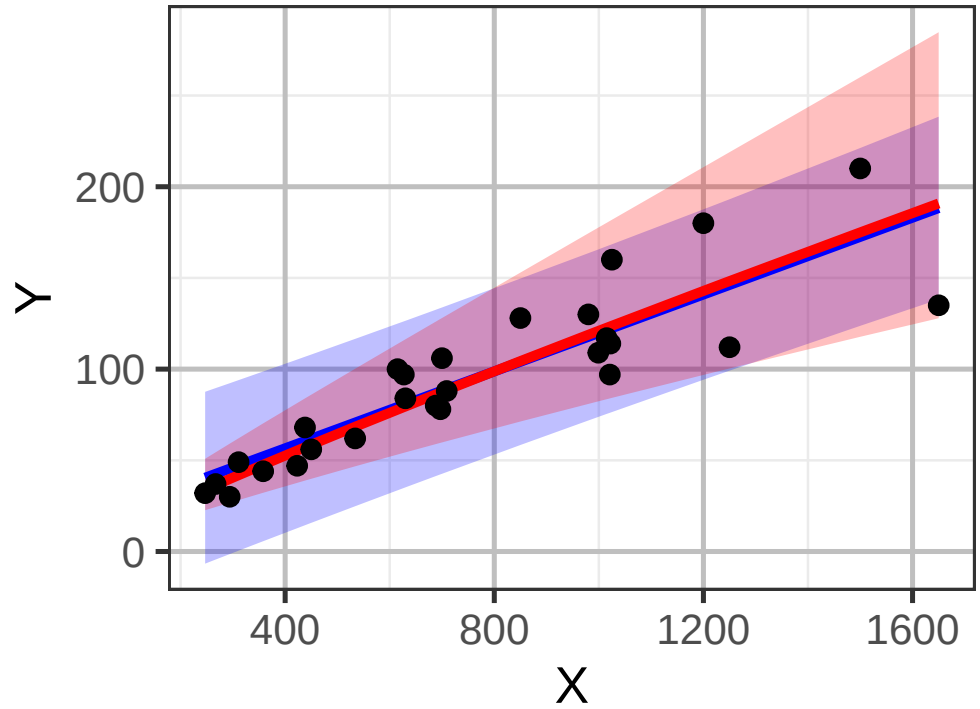
The line/curves for the 3 models below

- Blue: `lm(Y ~ X, data=supvis)`
- Green: `lm(log(Y) ~ X + I(X^2), data=supvis)`
- Red: `lm(log(Y) ~ log(X), data=supvis)`



Blue region: 95% prediction intervals based on the model `Y ~ X`

Red region: 95% prediction intervals based on the model `log(Y) ~ log(X)`



Though the model `Y~X` and `log(Y)~log(X)` have nearly identical fitted line/curve, their prediction intervals are very different. The former one is nearly constant in width, while the width of the latter one increases with X.

Caution on Transforming Variables

- Transformations are useful tools – we transform (rescale, generally) the variables in the model so that the linear regression model becomes (more) appropriate.
- Transformations, however, **cannot fix all problems**
  - a non-linear model may be needed,
  - one may try using **weighted least square** in Chapter 7 to solve the nonconstant variability problem if no appropriate transformation can be found.

Caution on Transforming Variables

- Transformed variables might be difficult to interpret
- There are often many ways of transforming the variables in a model, and there is seldom “the right one”. You might try more than one, and choose that which provides the right balance of model fit and ease of interpretation.
- Remember – whenever you transform your variables, all your estimates and confidence intervals are expressed in that scale. To report your results, you need to convert BACK to the original scale.