Lecture 15: Addressing Violations of Assumptions

Ailin Zhang

2023-06-16

Recap: Assumptions

- Assumptions about the model form
  - The relationship between the response (Y) and the predictors $(X_1, \ldots, X_p)$ is linear
- Assumptions about the errors
  - The errors are independent, with mean zero and constant variance. The errors can be normally distributed (optional)
- Assumptions about the predictors
  - The predictors are non-random, measured without error, and linearly independent
- Assumptions about the observations
  - All observations are equally reliable and have an approximately equal role in determining the regression results and in influencing conclusions

Violation of assumptions is very serious–it means that your linear model probably does a bad job at predicting your actual (non-linear) data.

Addressing Violations of Assumptions

We transform variables (including predictors and responses):

- to solve the non-linearity problem
- to reduce skewness (non-normal errors)
- to solve the non-constant variability problem
  - Variance-Stabilizing Transformation
  - Box-Cox Method

We have touched this topic when introducing polynomial models and ordinal categorical predictors

Linearizable Models

Linear models are linear in the parameters, not predictors:

All of the following are linear models:

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

Some Nonlinear Models Cannot Be Linearized

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

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

Linearizable Models

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

Transformations to Achieve Linearity

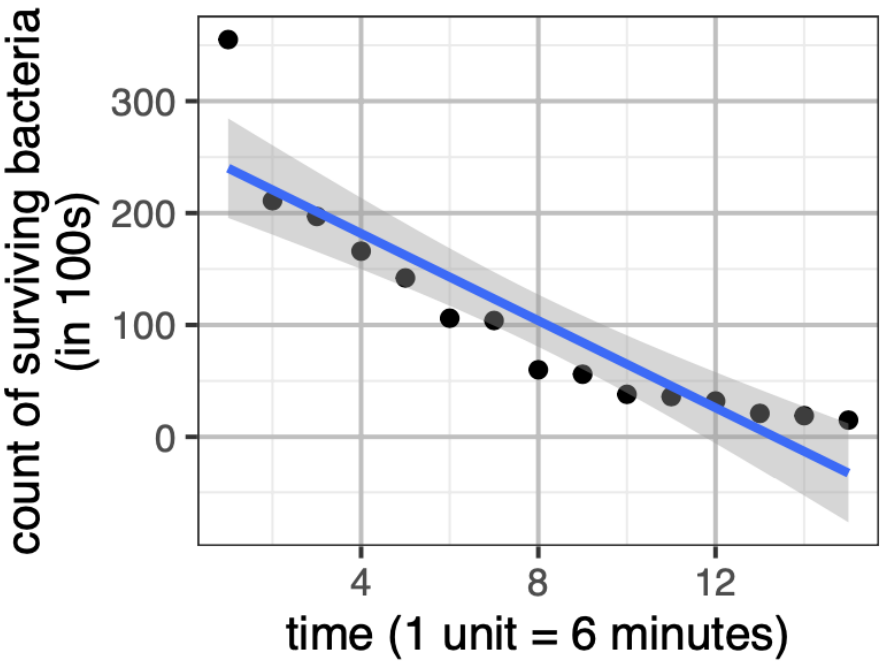
Data: Bacteria Deaths Due to X-Ray Radiation

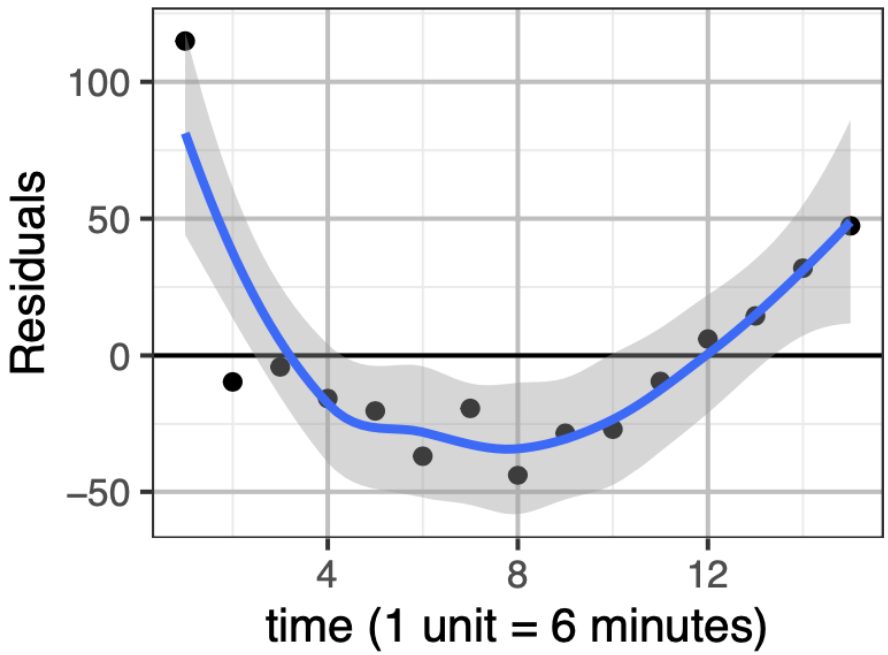
- t=time(1unit=6minutes)
- $N\_t = n_t =$ the number of surviving bacteria (in 100s) following exposure to 200-kilo-volt X-rays in after t units of time

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

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()
```





What is your finding?

- When not knowing what transformation to make, we would begin by looking at the scatterplot (left). 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 (right). 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:

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

Taking log of both sides, we get

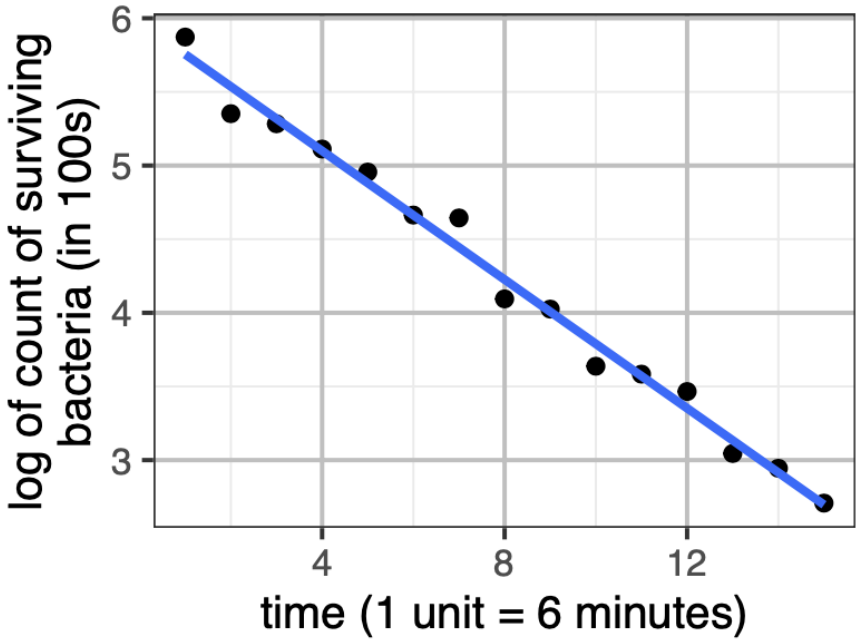
$$\log n_t = \log n_0 + \beta_1 t = \beta_0 + \beta_1 t,$$

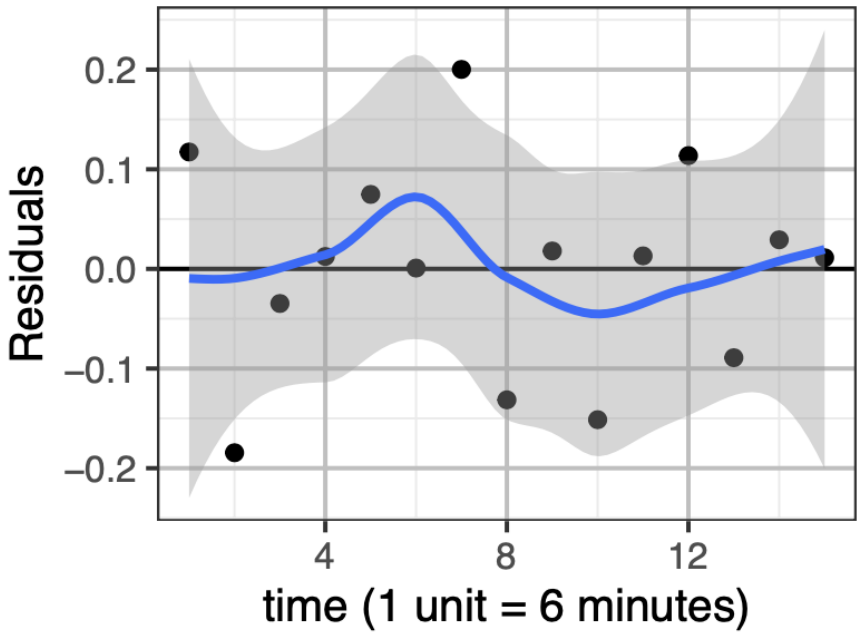
which suggests that we regress $\log n_t$ against t.

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

Example: Bacteria Deaths Due to X-Ray Radiation

```
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 cou
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

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

```
##                Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept)  5.9731603 0.059778097  99.92222 3.786354e-20
## t           -0.2184253 0.006574714 -33.22202 5.859568e-14
```

```
confint(lm2, "t")
```

```
##         2.5 %     97.5 %
## t -0.2326291 -0.2042214
```

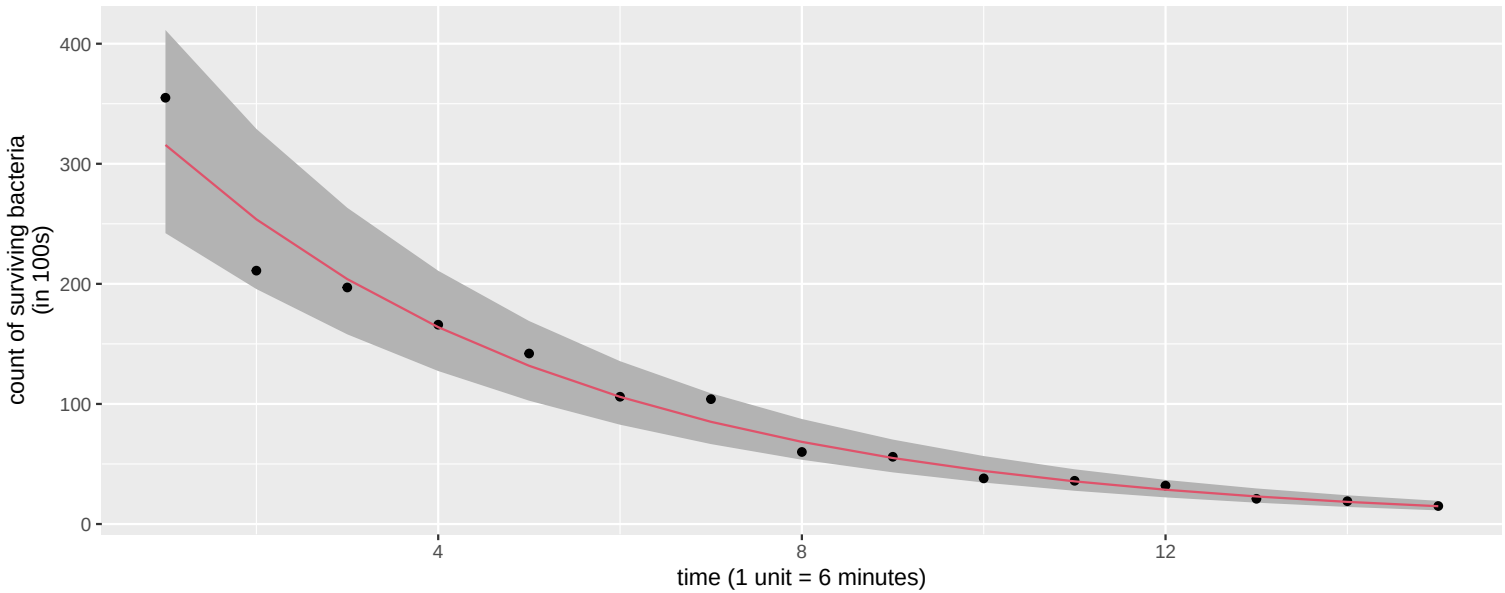
```
1-exp(confint(lm2, "t"))
```

```
##       2.5 %    97.5 %
## t 0.2075525 0.1847182
```

- Every 6 minutes, the number of surviving bacteria is estimated to decrease by $1 - e^{-0.2184} \approx 1 - 0.804 = 19.6\%$
- How to compute 95% CI?

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)")
```



Logarithm is the Most Commonly Used Transformation

- Rule of thumb #1: if a variable is about amount of money, take log
  - Income per capita
  - Price of diamond
  - Education cost
- Rule of thumb #2: if a variable represents the concentration of something, take log
  - Concentration of chemical in the blood, etc
- Rule of thumb #3: 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

- $\log(Y) = \beta_0 + \beta_1 X$

  When X is increased by 1, Y becomes e^{β_1} times as large

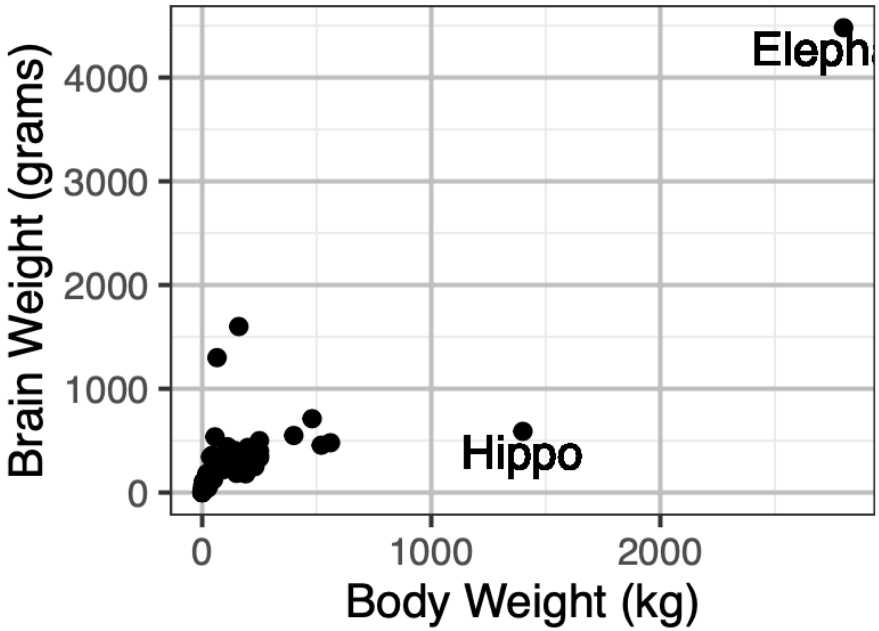
- $\log(Y) = \beta_0 + \beta_1 \log(X)$
  - When X is doubled ($X \to 2X$), Y becomes 2^{β_1} times as large
  - In Economics, β_1 in the log-log model $\log(Y) = \beta_0 + \beta_1 \log(X)$ is called Elasticity

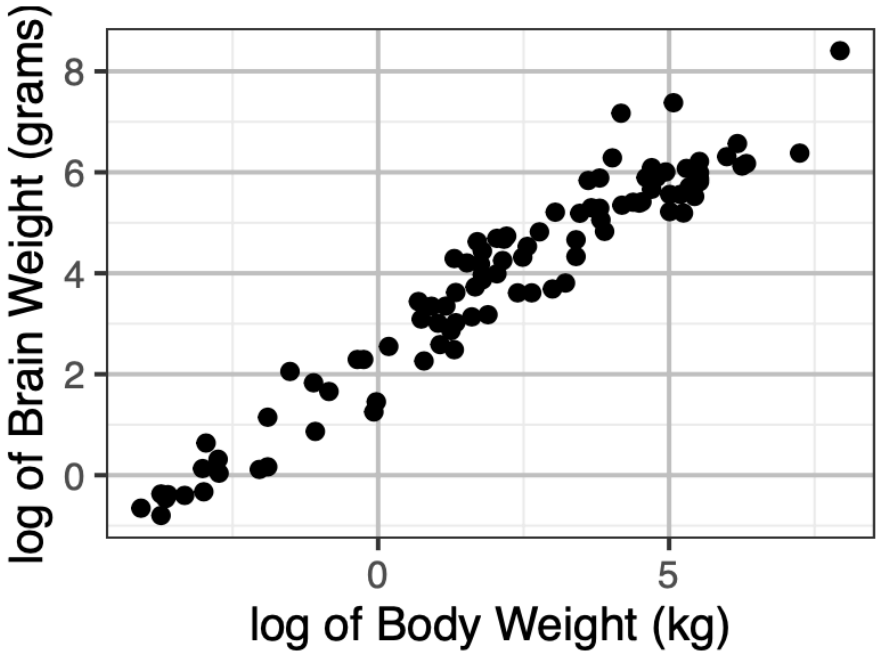
    This means a 1% increase in X ($dX/X = 1\% = 0.01$) would lead to a β_1% increase in Y ($dY/Y = \beta_1 \times 0.01$)

Transformations to Reduce 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.
- 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.
  - When there exist outliers, try transforming variables

Example: Brain and Body Weight of Mammals





Before transformation, what is the skewness of `Brain Weight` and `Body Weight`?

Transformations to Reduce Skewness

Skewness can often be ameliorated by a power transformation.

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

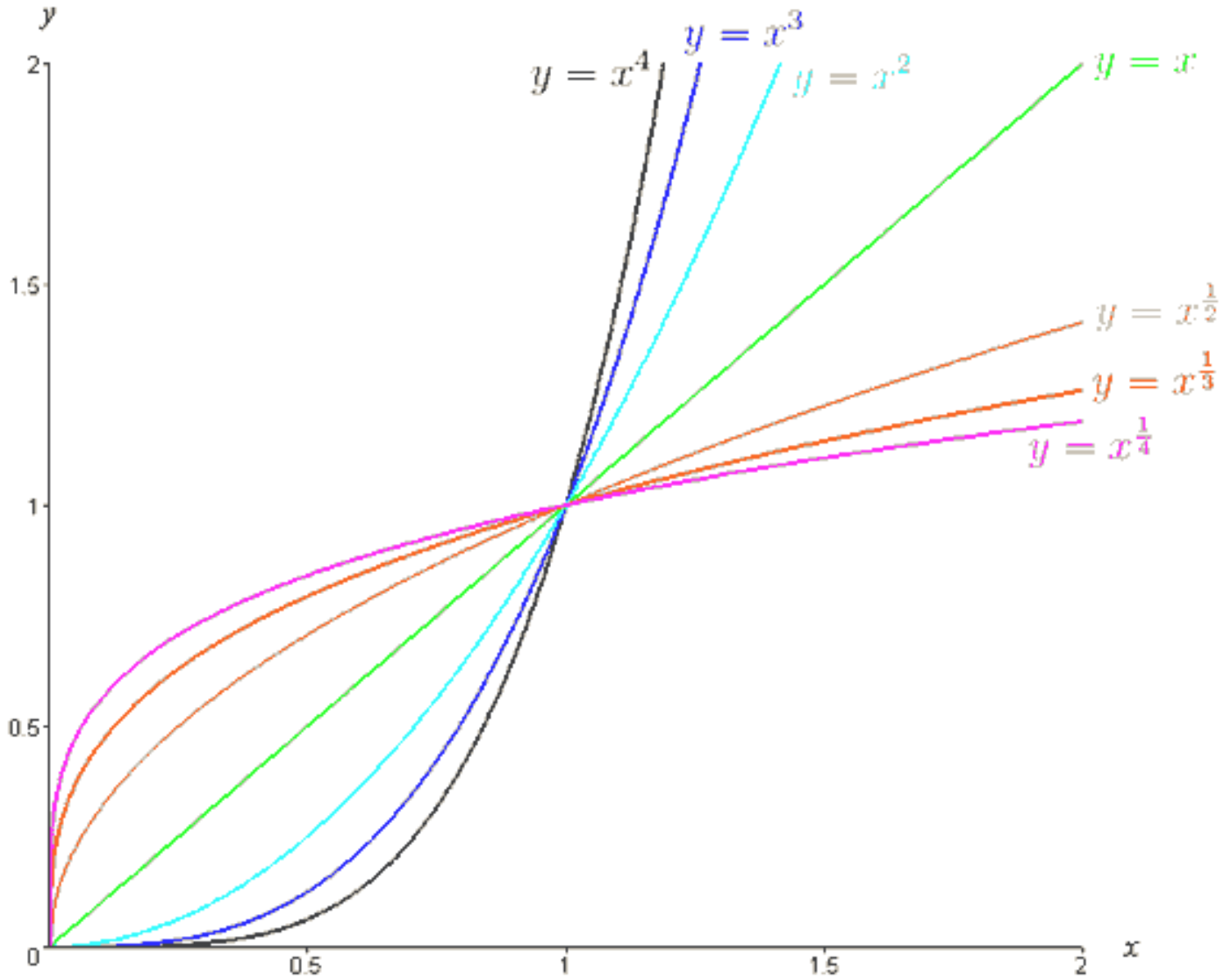
- If **right-skewed**, take $\lambda < 1$

$$y \longrightarrow \frac{1}{y}, \log(y), \sqrt{y}, \ldots (y^\lambda, \lambda < 1)$$

- If **left-skewed**, take $\lambda > 1$

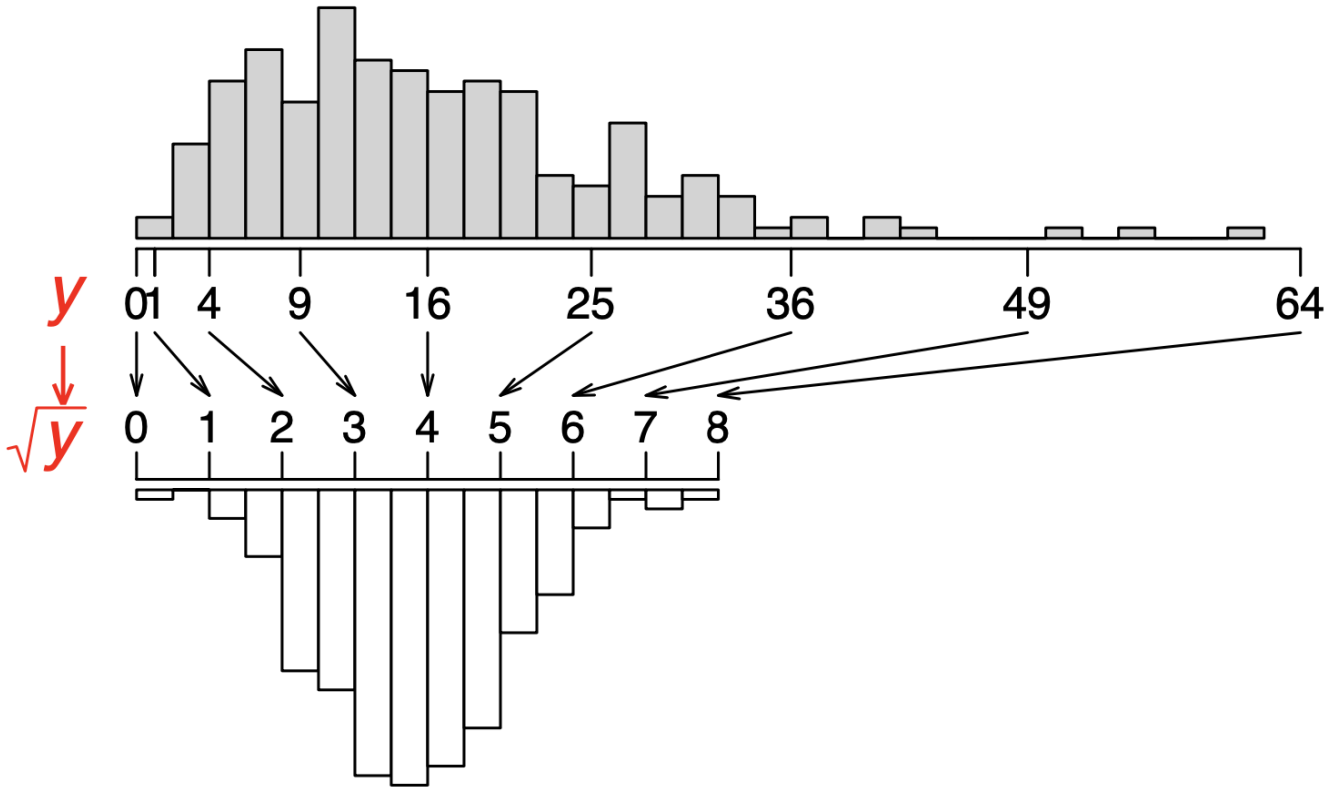
$$y \longrightarrow y^2, y^3, \ldots (y^\lambda, \lambda > 1)$$

Pattern of Power Functions



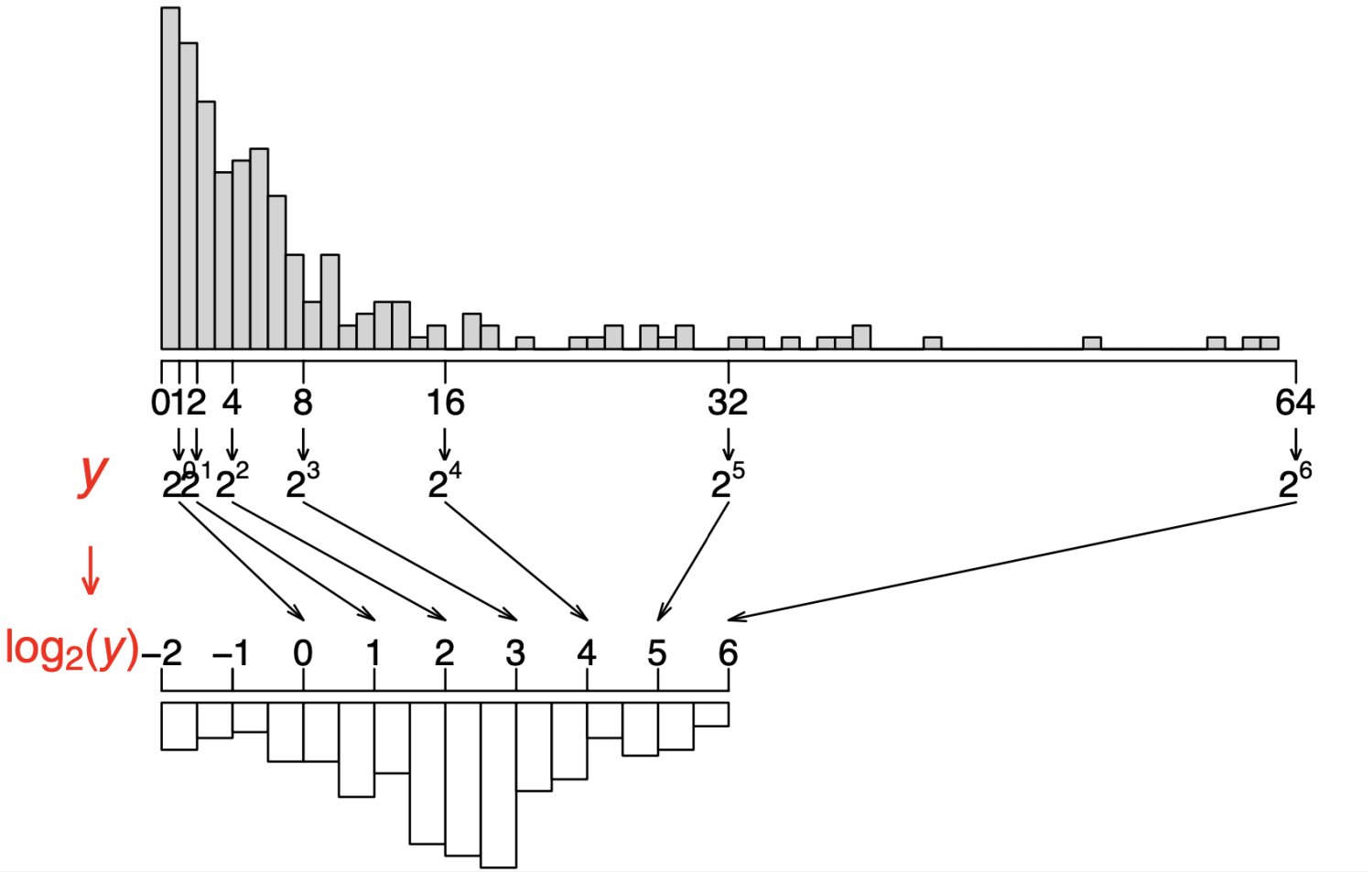
$\lambda < 1$ Transformation Can Reduce Right-Skewness

For example, let's take $\lambda = \frac{1}{2}$. The square-root transformation can shorten the upper tail and extend the lower tail, of a distribution and hence can reduce right-skewness.



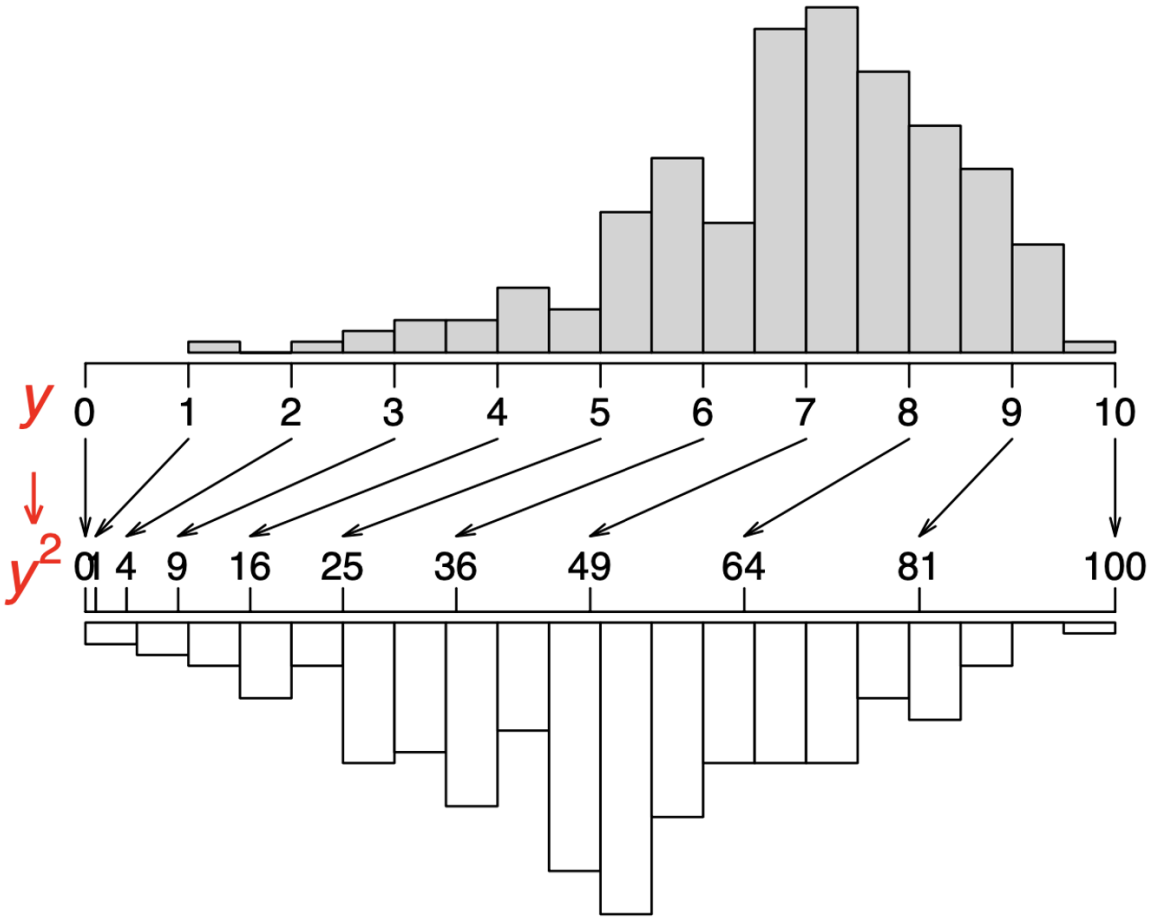
$\lambda < 1$ Transformation Can Reduce Right-Skewness

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



$\lambda > 1$ Transformation Can Reduce Left-Skewness

The square transformation can extend the upper tail and shorten the lower tail, and hence can reduces left-skewness (and increase right-skewness).



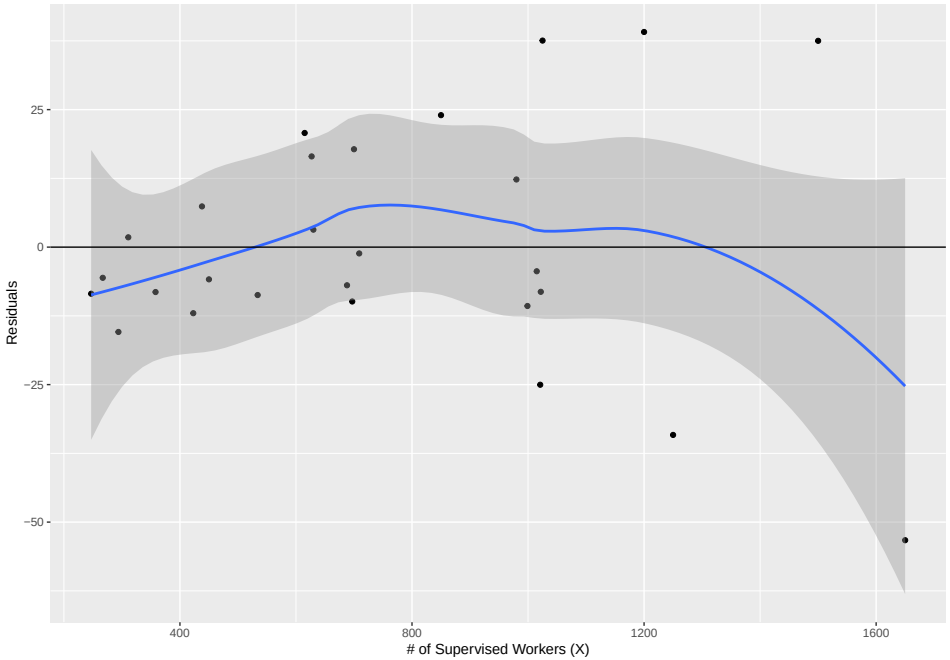
Transformations to

Transformations to Stablize Variance

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)")+geom_smooth()+
  ylab("Residuals")+ geom_hline(yintercept=0)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



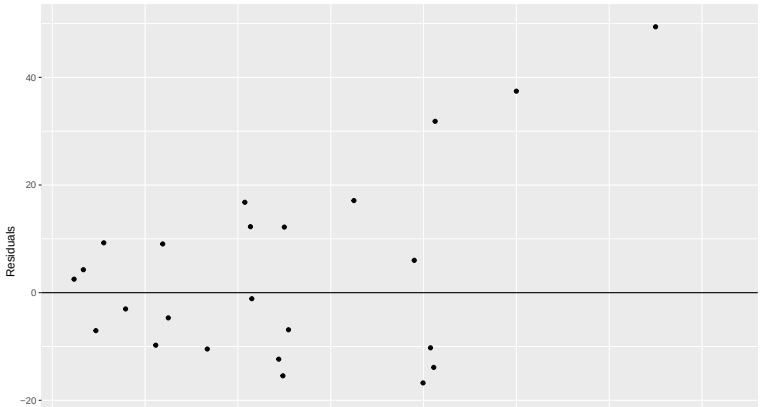
Transformations to Stablize Variance

From the previous figure, 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)
```



Transformations to Stablize Variance

- Now we need to address non-constant variance.
- Why heteroscedasticity is a problem?

  If heteroscedastic, 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 the $\sqrt{\text{fitted value}}$, i.e., the variance is proportional to the fitted value, then: $y \to \sqrt{y}$
- In general, if the SD is proportional to (the fitted values)$^{\alpha}$, $\alpha \neq 1$, then the variance-stabilizing transformation is $y \to y^{1-\alpha}$

Variance-Stabilizing Transformation

Probability Distribution of Y	$\text{Var}(Y)$ in Terms of Its Mean μ	Transformation	Resulting Variance
Poisson[a]	μ	$\sqrt{Y}$ or $(\sqrt{Y} + \sqrt{Y+1})$	0.25
Binomial[b]	$\mu(1-\mu)/n$	$\sin^{-1}\sqrt{Y}$ (degrees)	$821/n$
		$\sin^{-1}\sqrt{Y}$ (radians)	$0.25/n$
Negative Binomial[c]	$\mu + \lambda^2\mu^2$	$\lambda^{-1}\sinh^{-1}(\lambda\sqrt{Y})$ or	
		$\lambda^{-1}\sinh^{-1}(\lambda\sqrt{Y} + 0.5)$	0.25

Box-Cox Method

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

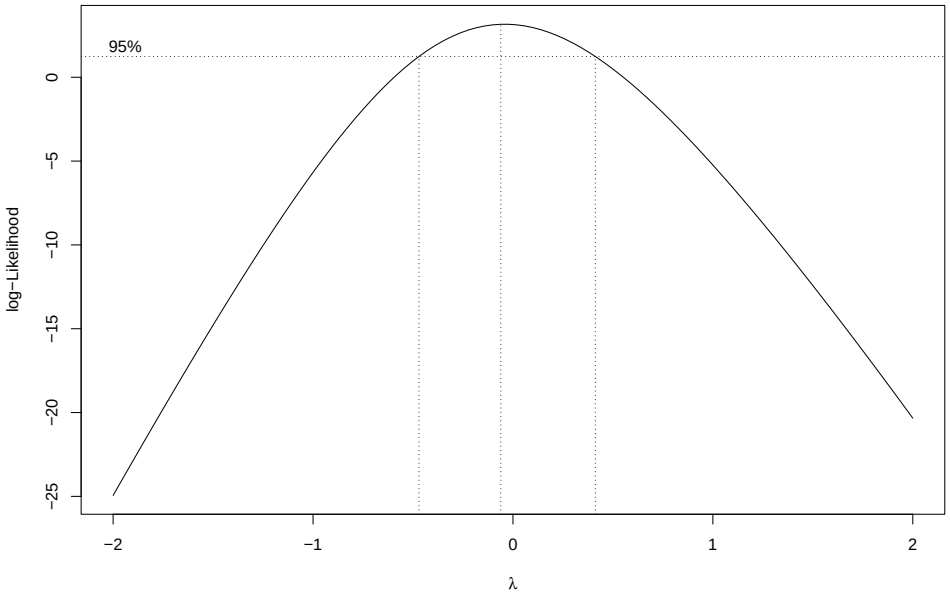
$$Y^{\lambda} = \beta_0 + \beta_1 X1 + \cdots + \beta_p X_p + \epsilon$$

closest to **normal** and **constant** variability.

- We usually round the optimalλ to a convenient power like $-1, -1/2, -1/3, 0, 1/3, 1/2, 1, 2, \ldots$ 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))
```

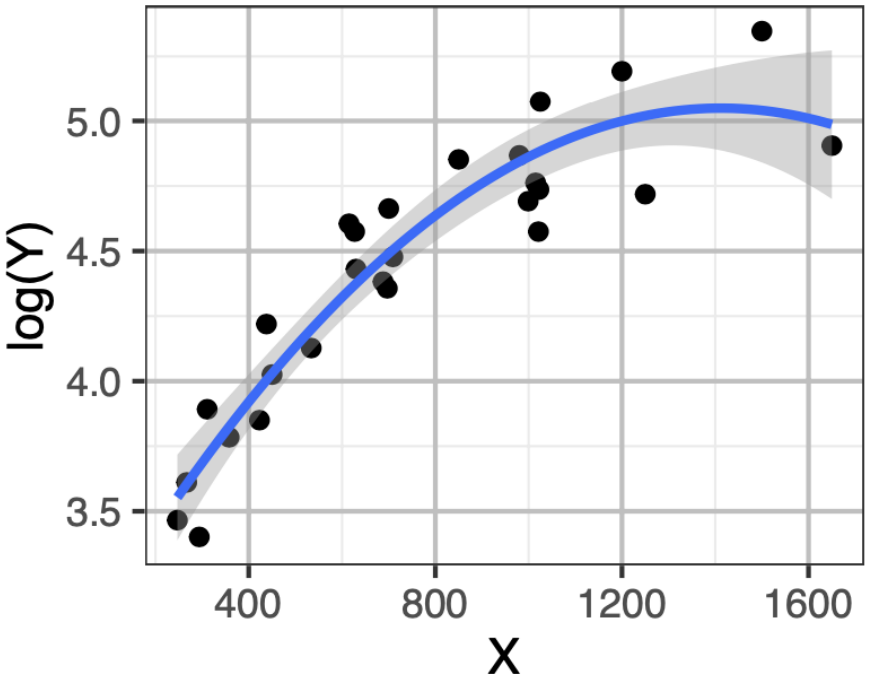


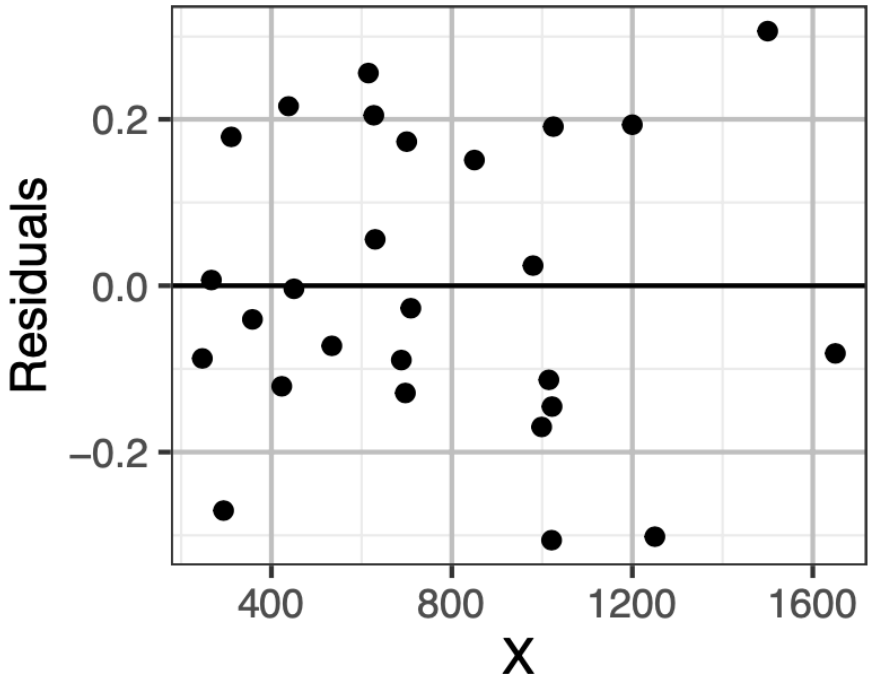
- The middle dash line marks the optimal λ, the right and left dash line mark the 95% C.I. for the optimal λ.
- For the plot, we see the optimal λ is around 0.1, and the 95% C.I. contains 0. For simplicity, we pick $\lambda = 0$ and use $\log Y$.

Box-Cox Method for Supervisor/Employee Data

```
lm3 = lm(log(Y) ~ X + I(X^2), data=supvis)
summary(lm3)$coef
```

```
##                   Estimate   Std. Error   t value     Pr(>|t|)
## (Intercept)  2.851600e+00 1.566401e-01 18.204788 1.496262e-15
## X            3.112674e-03 3.989301e-04  7.802554 4.897636e-08
## I(X^2)      -1.102256e-06 2.238069e-07 -4.925028 5.027216e-05
```





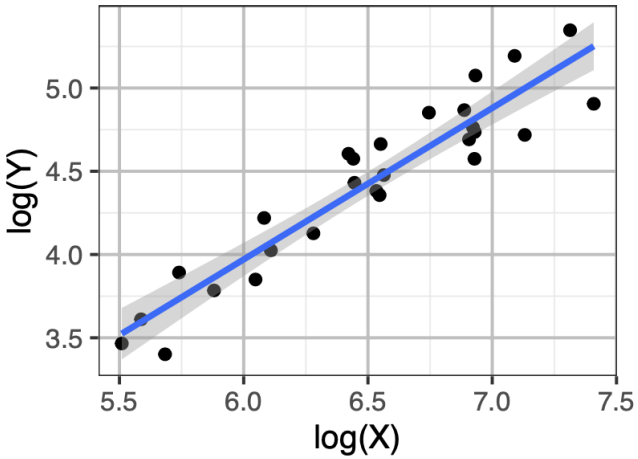
Conclusion after the transformation

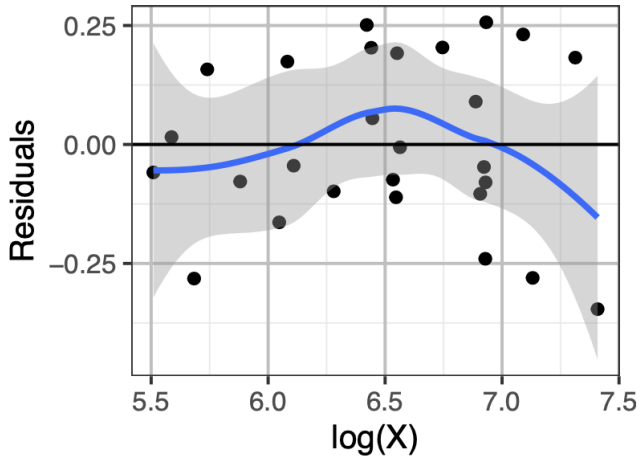
No clear nonlinearity or heteroscadasticity.

- The quadratic term X^2 is significant
- Relation between X and $log(Y)$ is NOT monotone based on the quadratic model (In reality, it might not be the case)

An alternative model

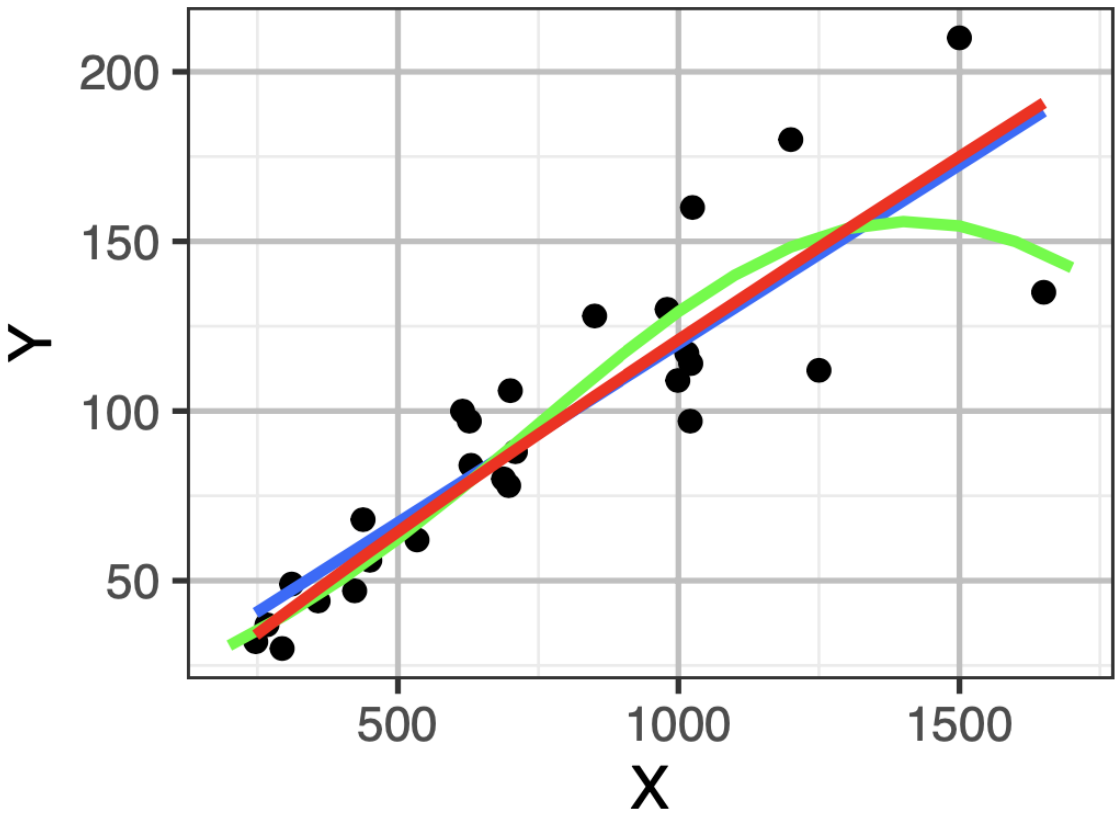
```
lm4 = lm(log(Y) ~ log(X), data=supvis)
```





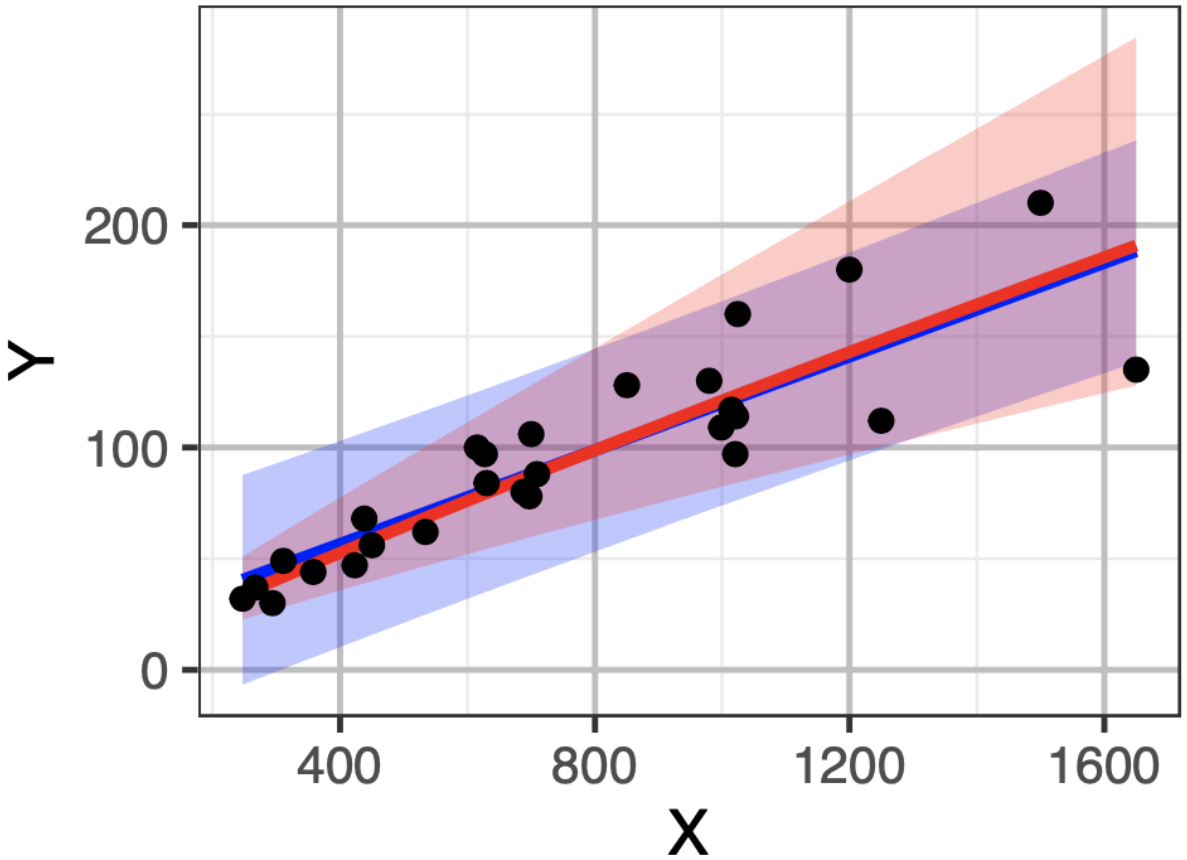
The line/curves for the 3 models

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



Implication of the transformation

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.



Be careful!

- 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 to solve the nonconstant variability problem if no appropriate transformation can be found.
  - 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.