Technical conditions for linear regression

INFERENCE FOR LINEAR REGRESSION IN R

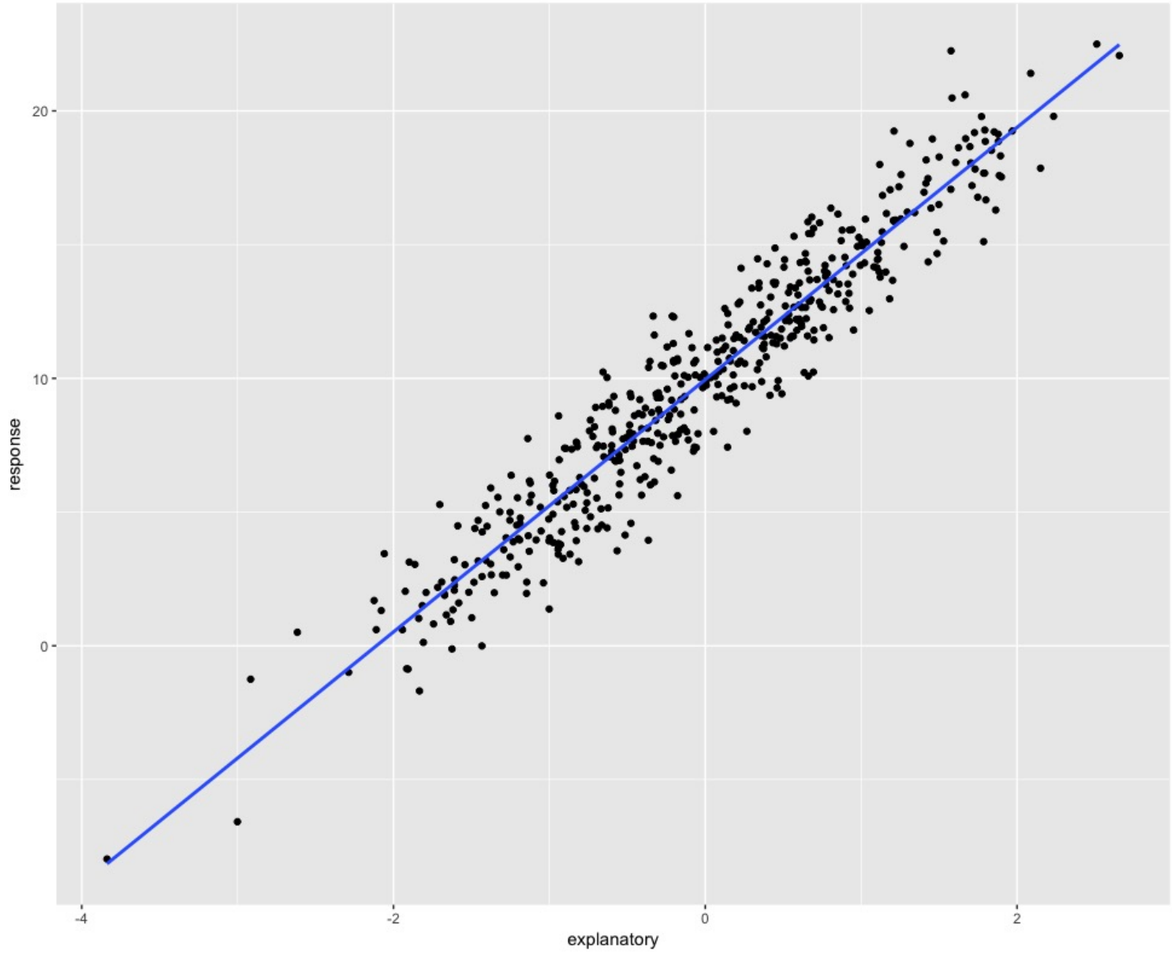
Jo Hardin
Professor, Pomona College

What are the technical conditions?

$$Y = \beta_0 + \beta_1 \cdot X + \epsilon$$

$$\epsilon \sim N(0, \sigma_\epsilon)$$

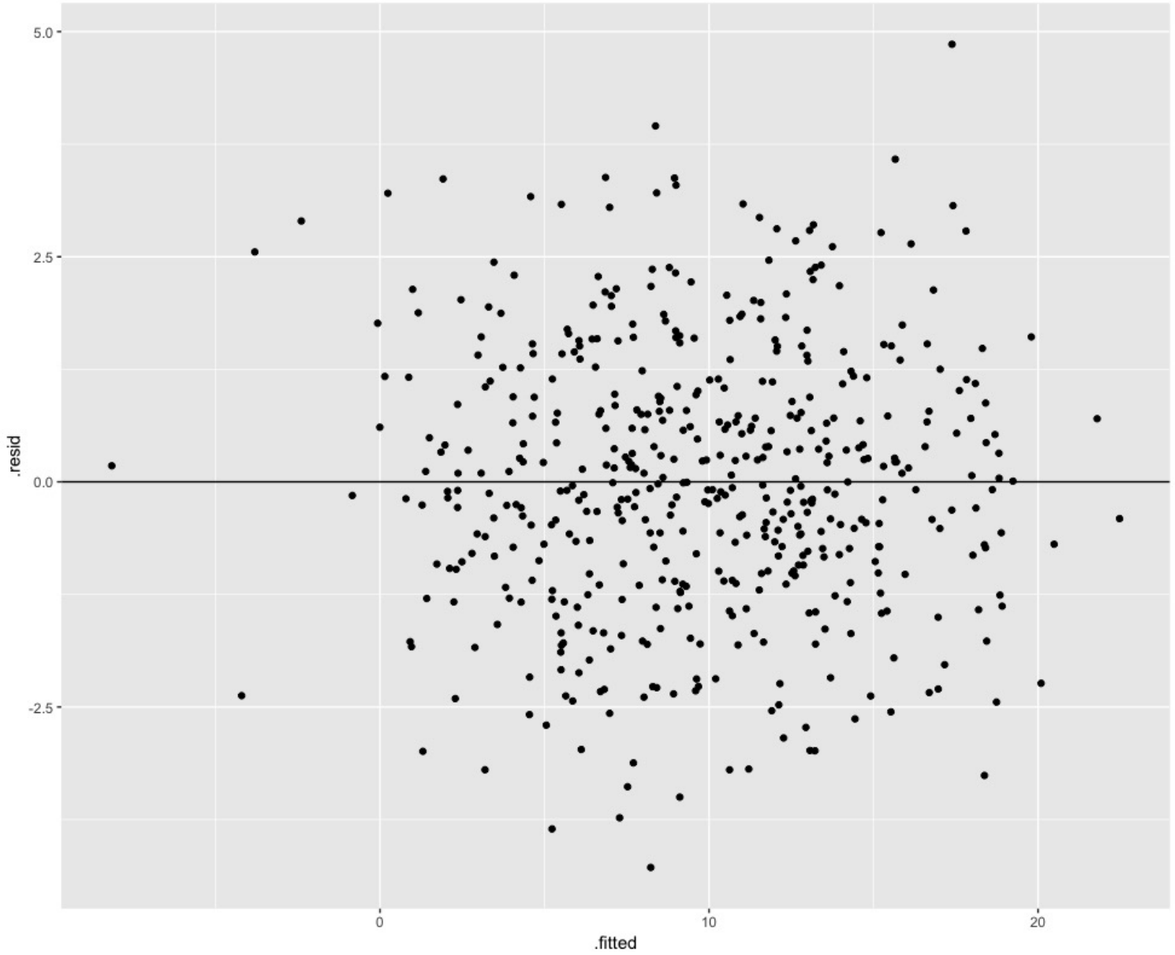
- L: linear model
- I: independent observations
- N: points are normally distributed around the line
- E: equal variability around the line for all values of the explanatory variable



Linear model: residuals

```
linear_lm <- augment(
  lm(response ~ explanatory,
     data = lineardata)
)

ggplot(linear_lm,
       aes(x =. fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept=0)
```



Fitted value: $\hat{Y}_i = b_0 + b_1 X_i$

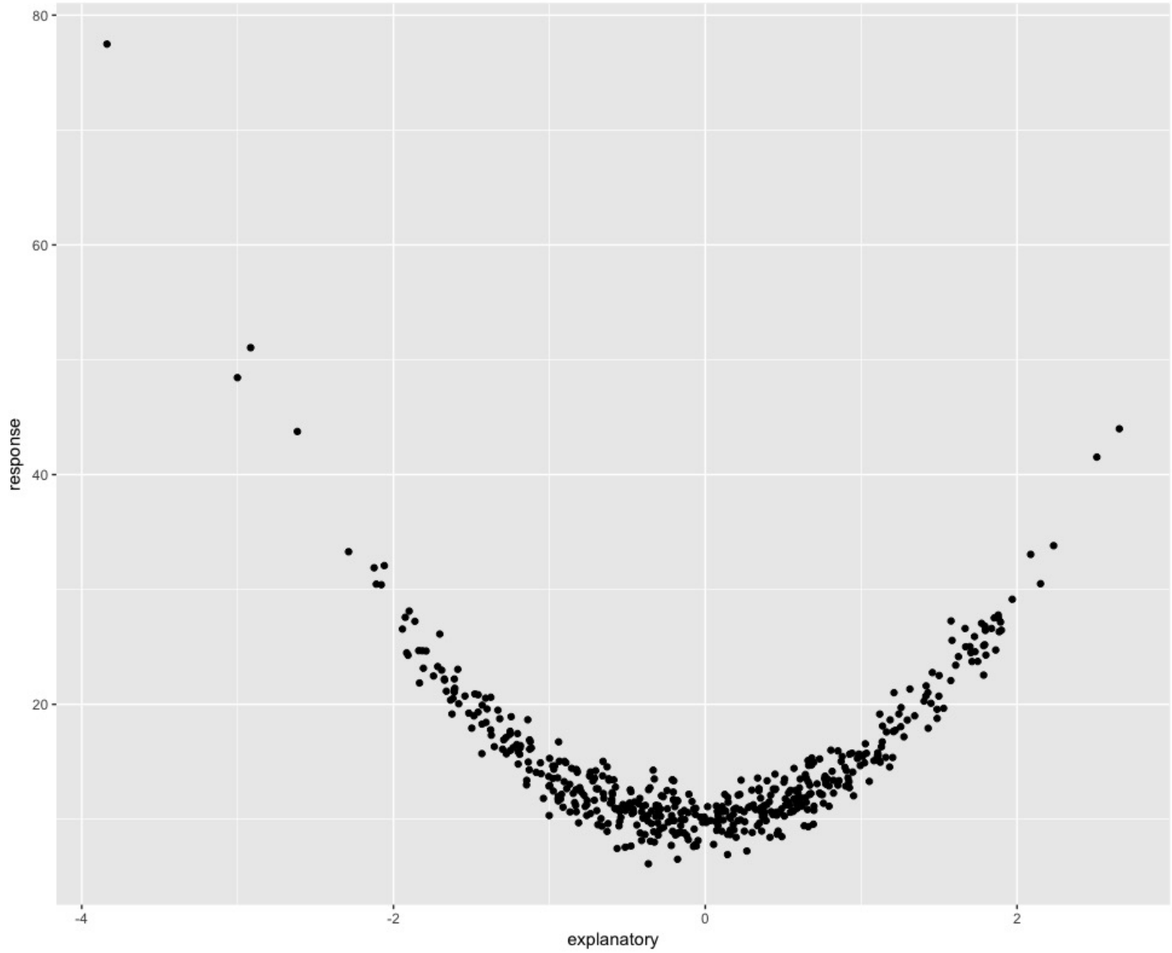
Residual: $e_i = Y_i - \hat{Y}_i$

Not linear

$$Y = \beta_0 + \beta_1 \cdot X + \epsilon$$

$$\epsilon \sim N(0, \sigma_\epsilon)$$

- L: linear model
- I: independent observations
- N: points are normally distributed around the line
- E: equal variability around the line for all values of the explanatory variable

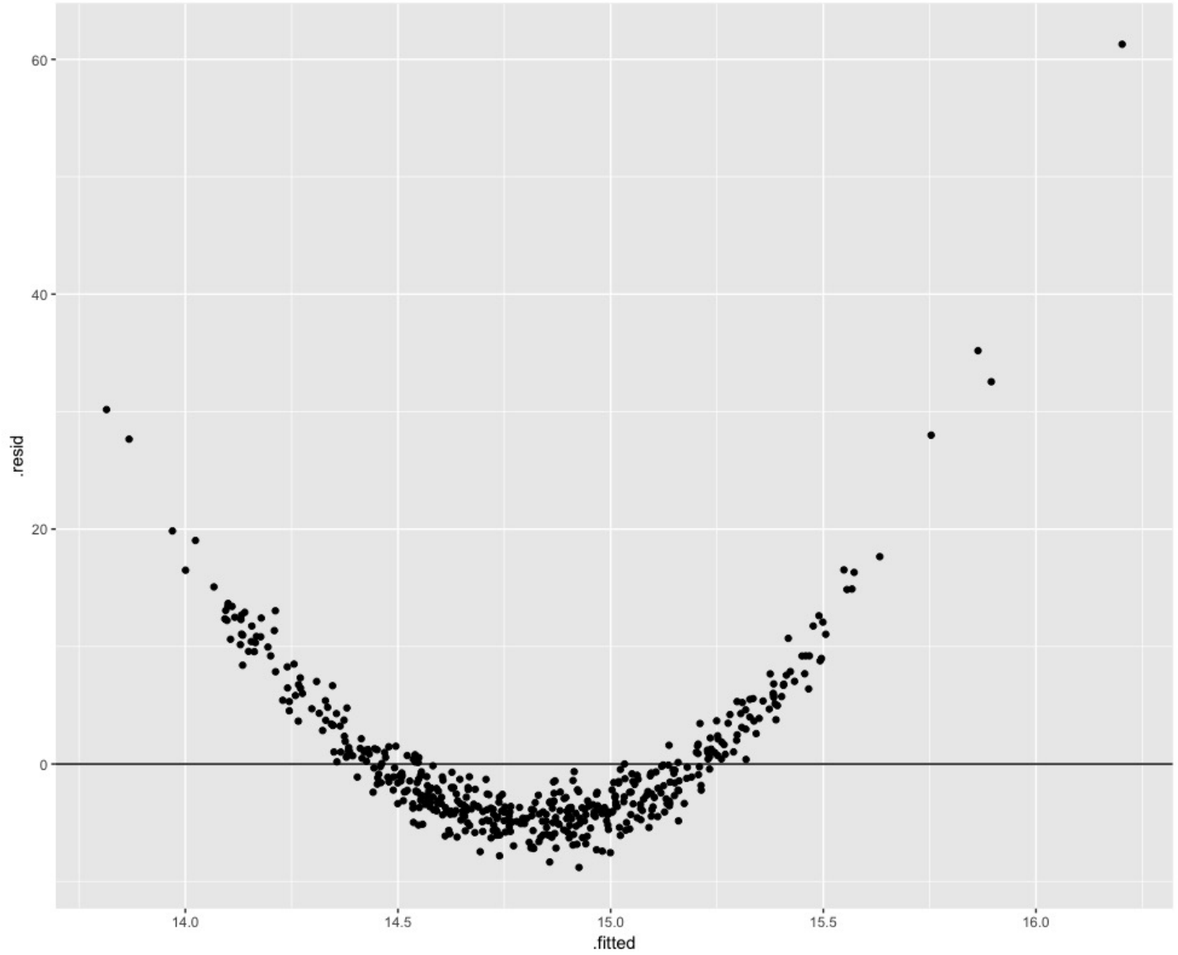


Not linear: residuals

```
nonlinear_lm <- augment(
  lm(response ~ explanatory,
     data = nonlineardata))
ggplot(nonlinear_lm,
       aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept=0)
```

fitted value: $\hat{Y}_i = b_0 + b_1 X_i$

residual: $e_i = Y_i - \hat{Y}_i$

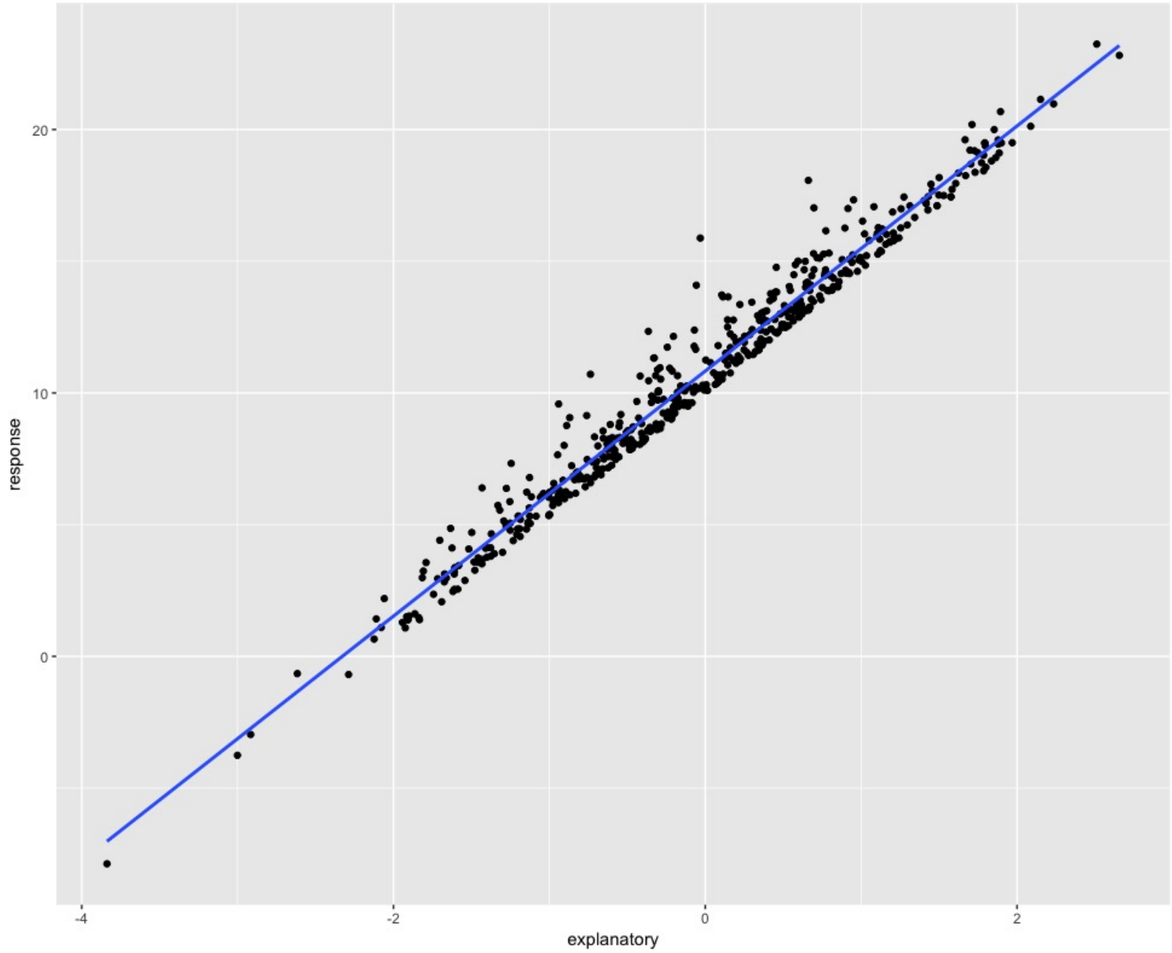


Not normal

$$Y = \beta_0 + \beta_1 \cdot X + \epsilon$$

$$\epsilon \sim N(0, \sigma_\epsilon)$$

- L: linear model
- I: independent observations
- N: points are normally distributed around the line
- E: equal variability around the line for all values of the explanatory variable

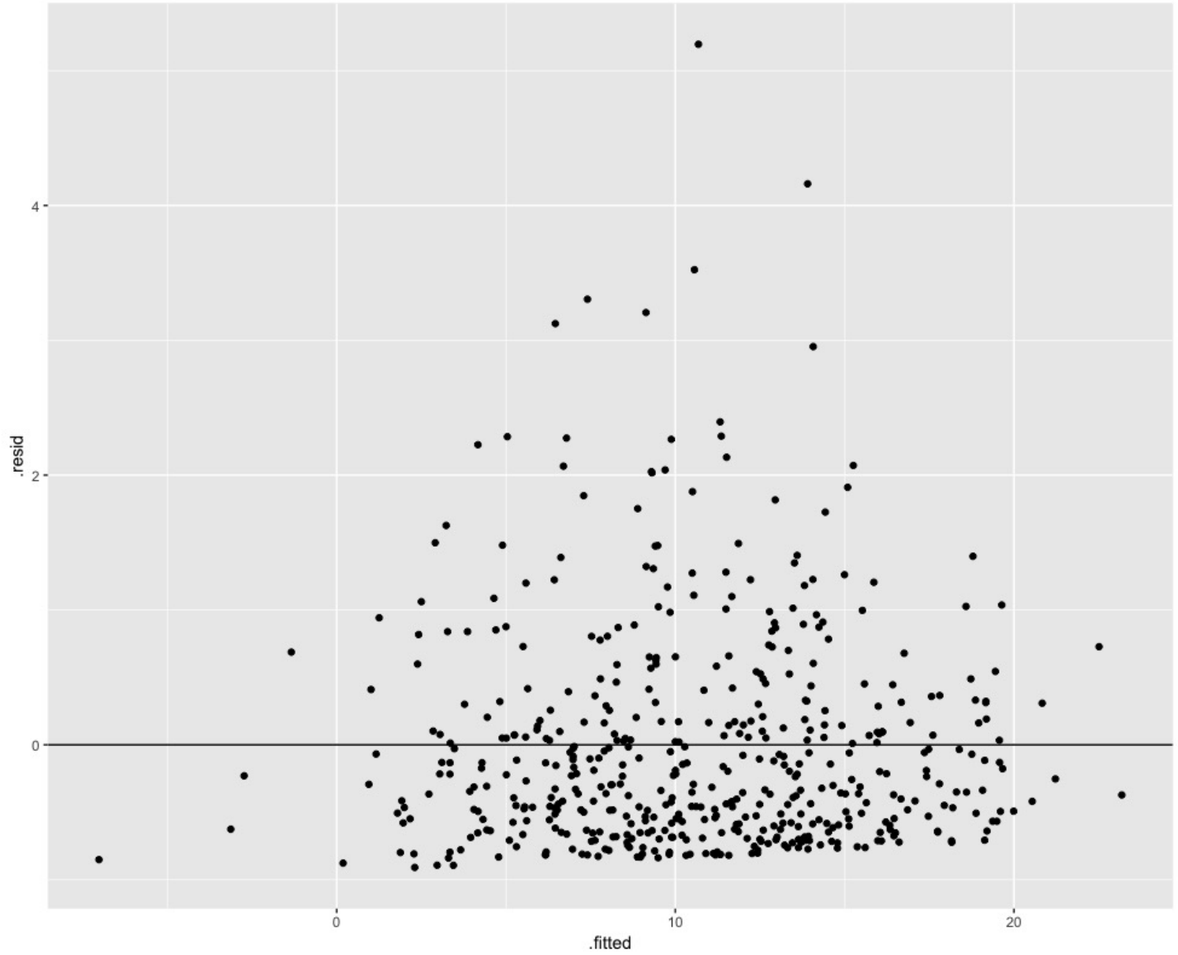


Not normal: residuals

```
nonnormal_lm <- augment(
  lm(response ~ explanatory,
     data = nonnormaldata))
ggplot(nonnormal_lm,
       aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0)
```

fitted value: $\hat{Y}_i = b_0 + b_1 X_i$

residual: $e_i = Y_i - \hat{Y}_i$

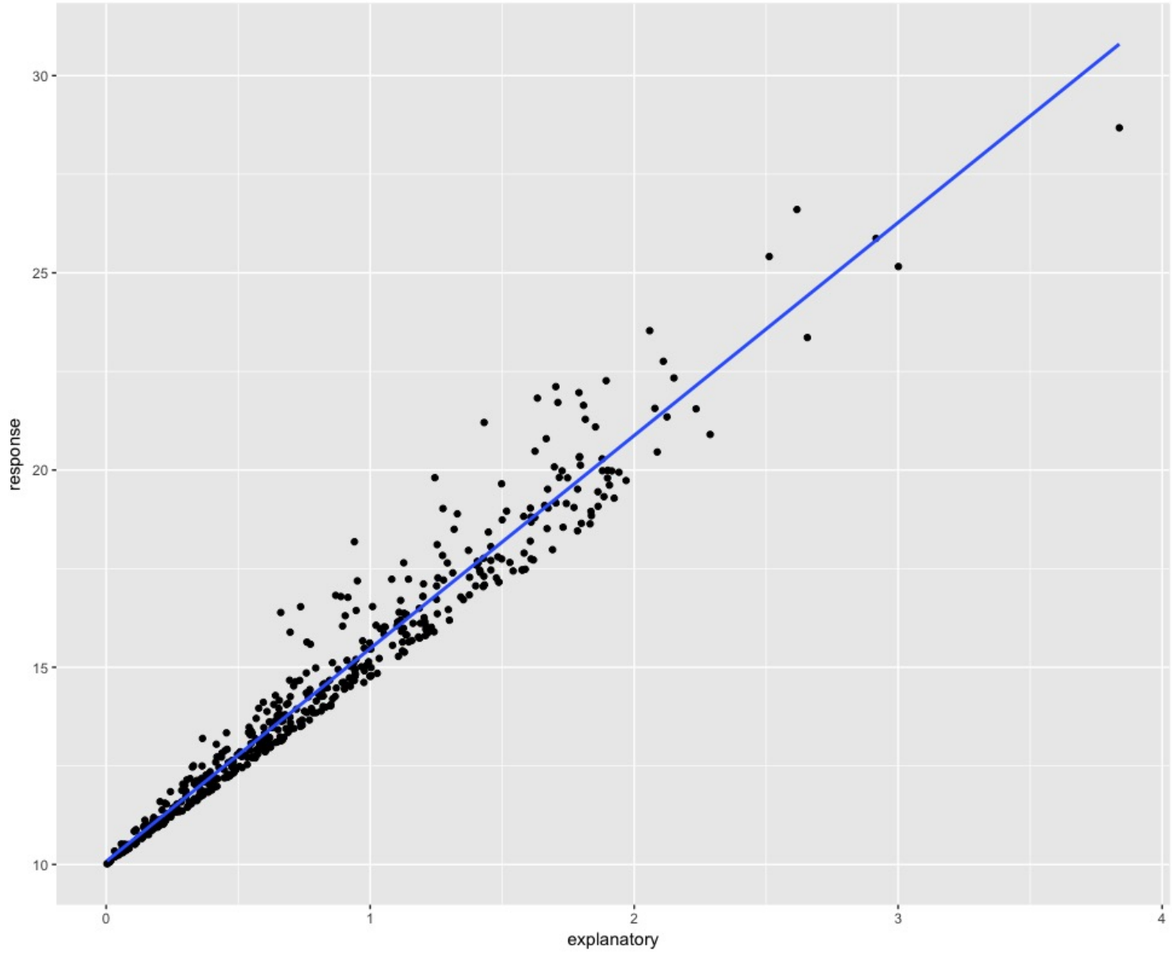


Not equal variance

$$Y = \beta_0 + \beta_1 \cdot X + \epsilon$$

$$\epsilon \sim N(0, \sigma_\epsilon)$$

- L: linear model
- I: independent observations
- N: points are normally distributed around the line
- E: equal variability around the line for all values of the explanatory variable

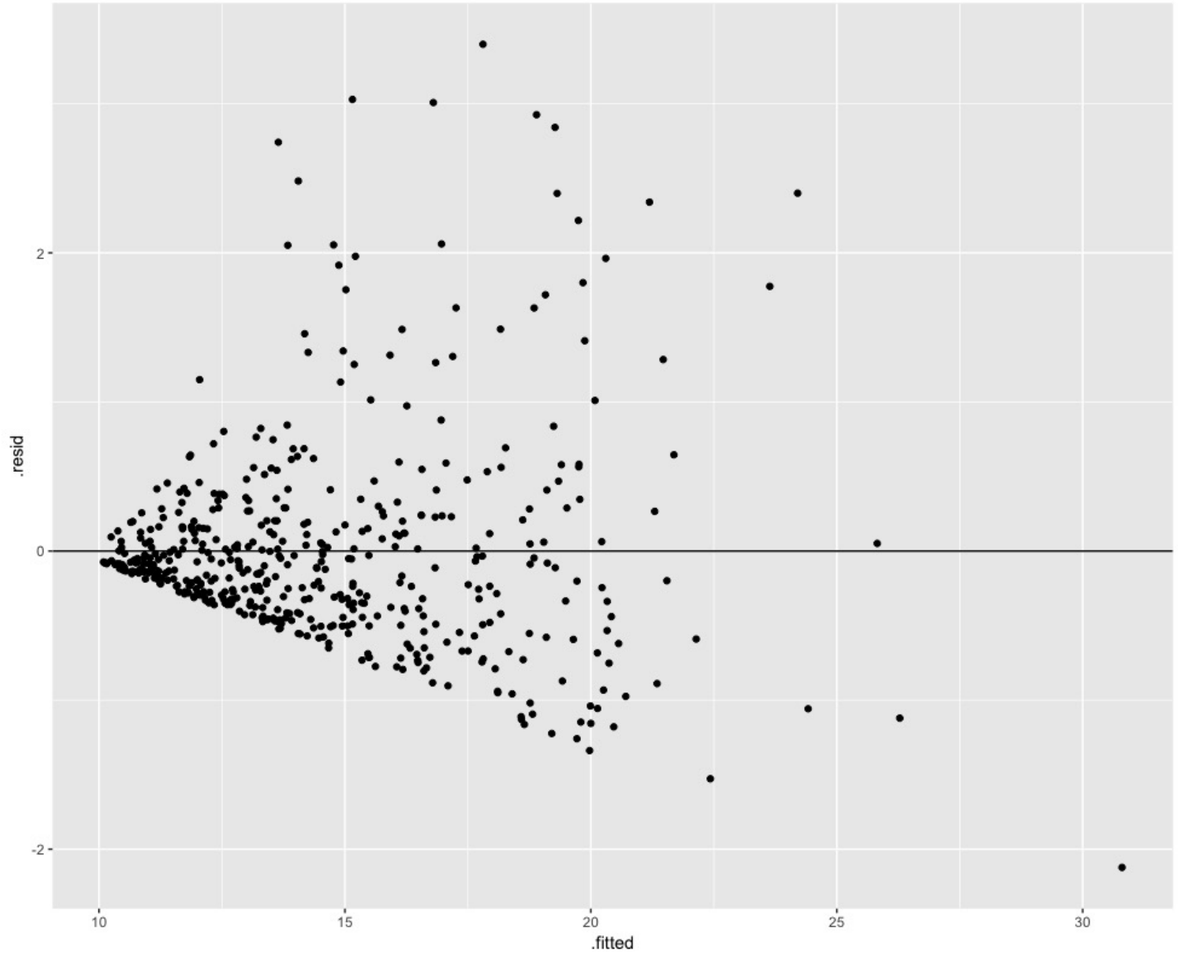


Not equal variance: residuals

```
nonequal_lm <- augment(
  lm(response ~ explanatory,
     data = nonequaldata))
ggplot(nonequal_lm,
       aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0)
```

fitted value: $\hat{Y}_i = b_0 + b_1 X_i$

residual: $e_i = Y_i - \hat{Y}_i$



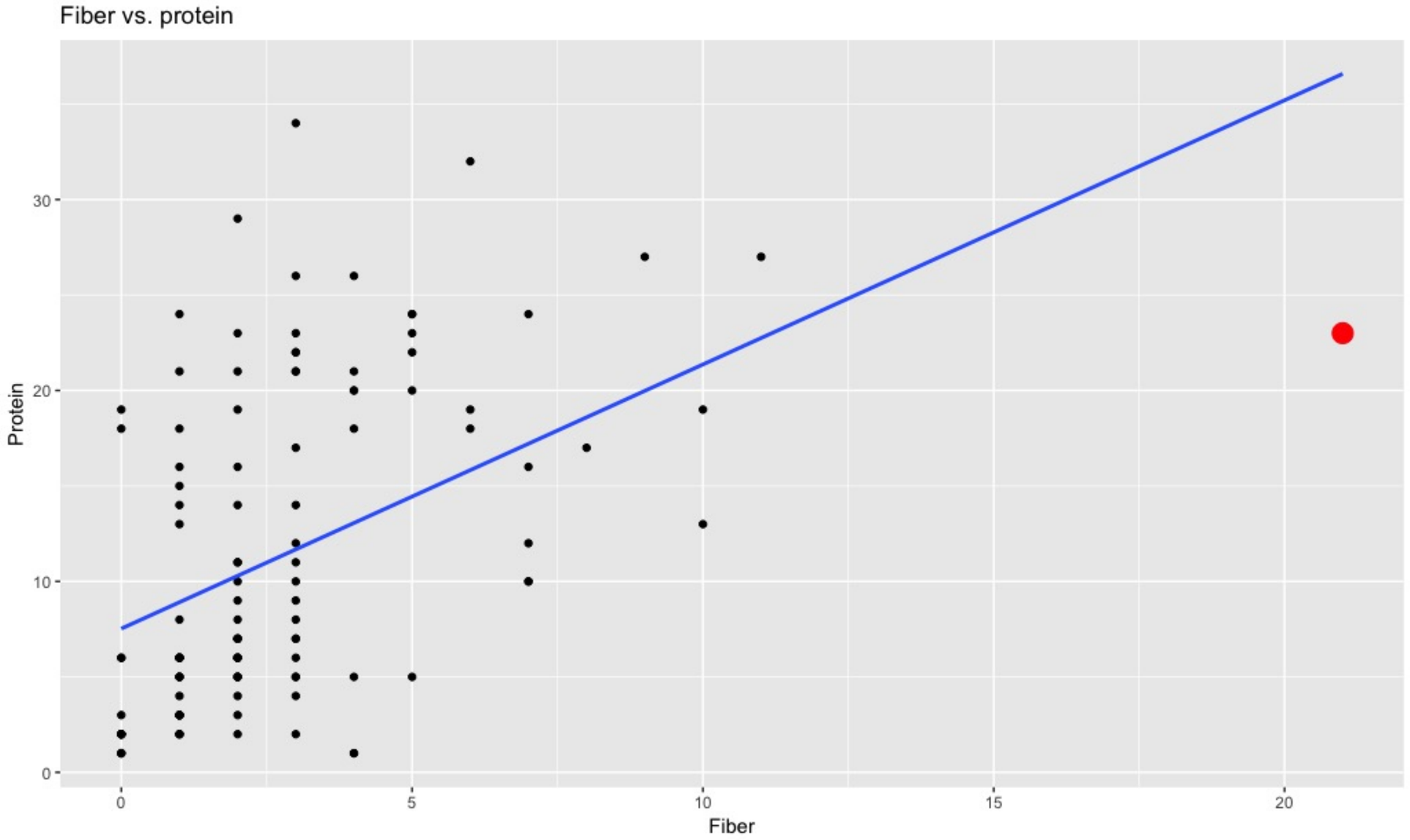
Let's practice!

INFERENCE FOR LINEAR REGRESSION IN R

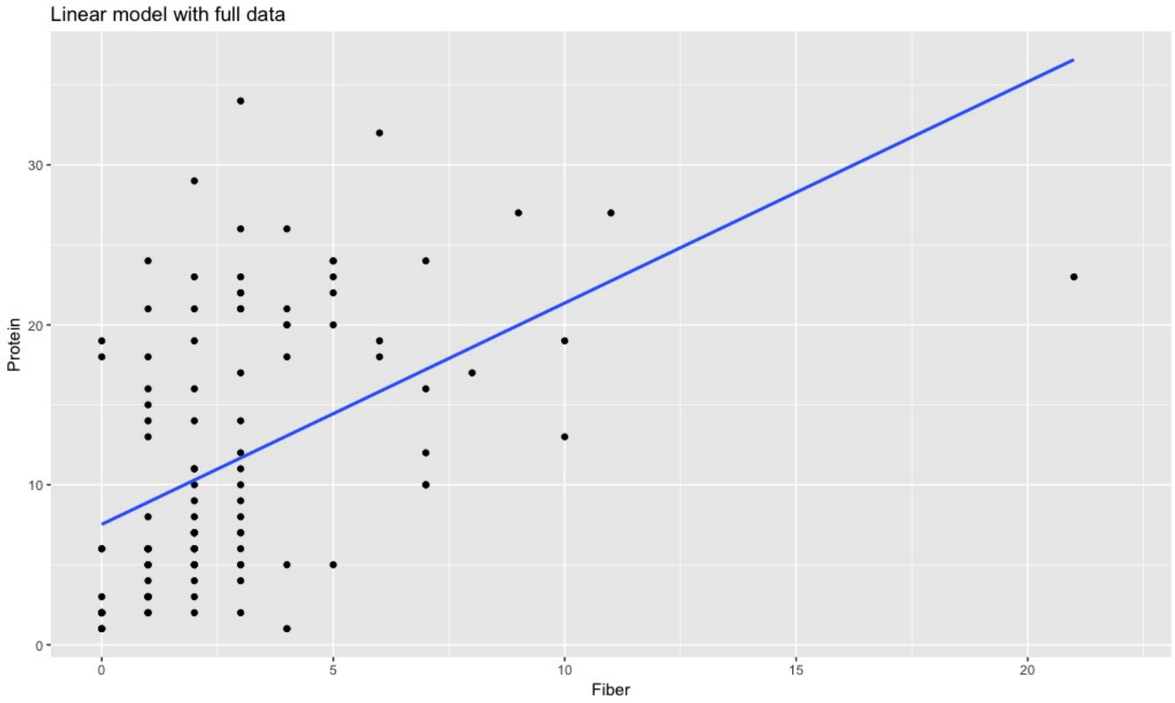
Effect of an outlier

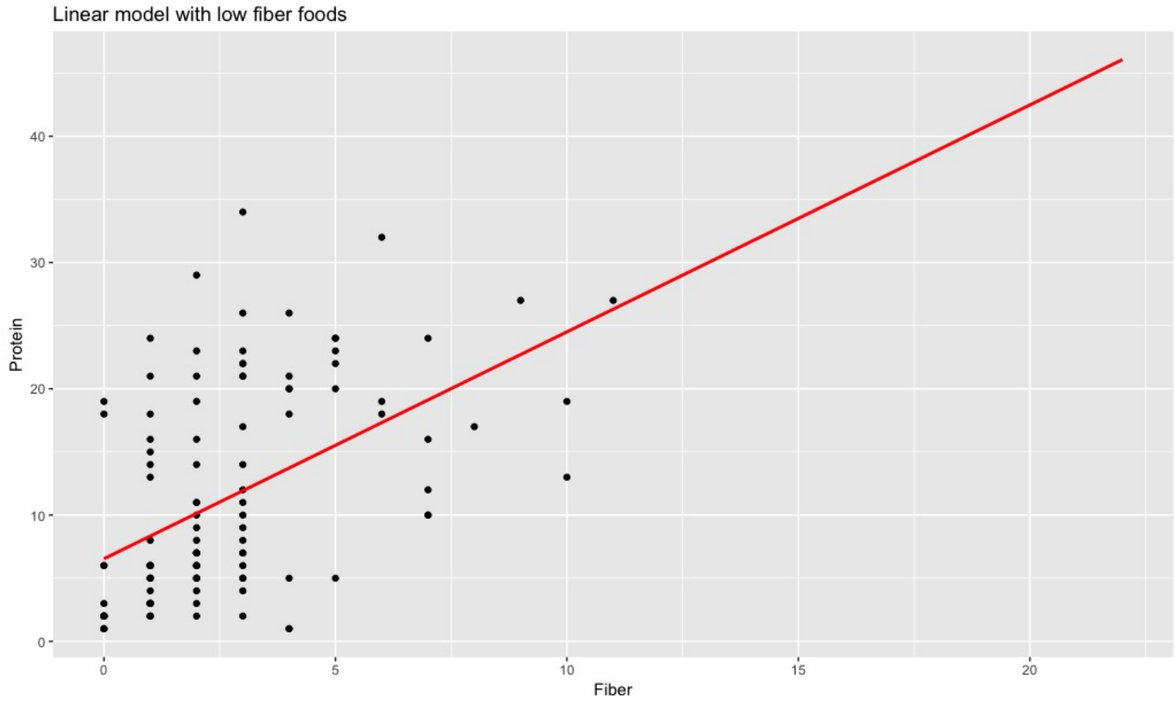
INFERENCE FOR LINEAR REGRESSION IN R

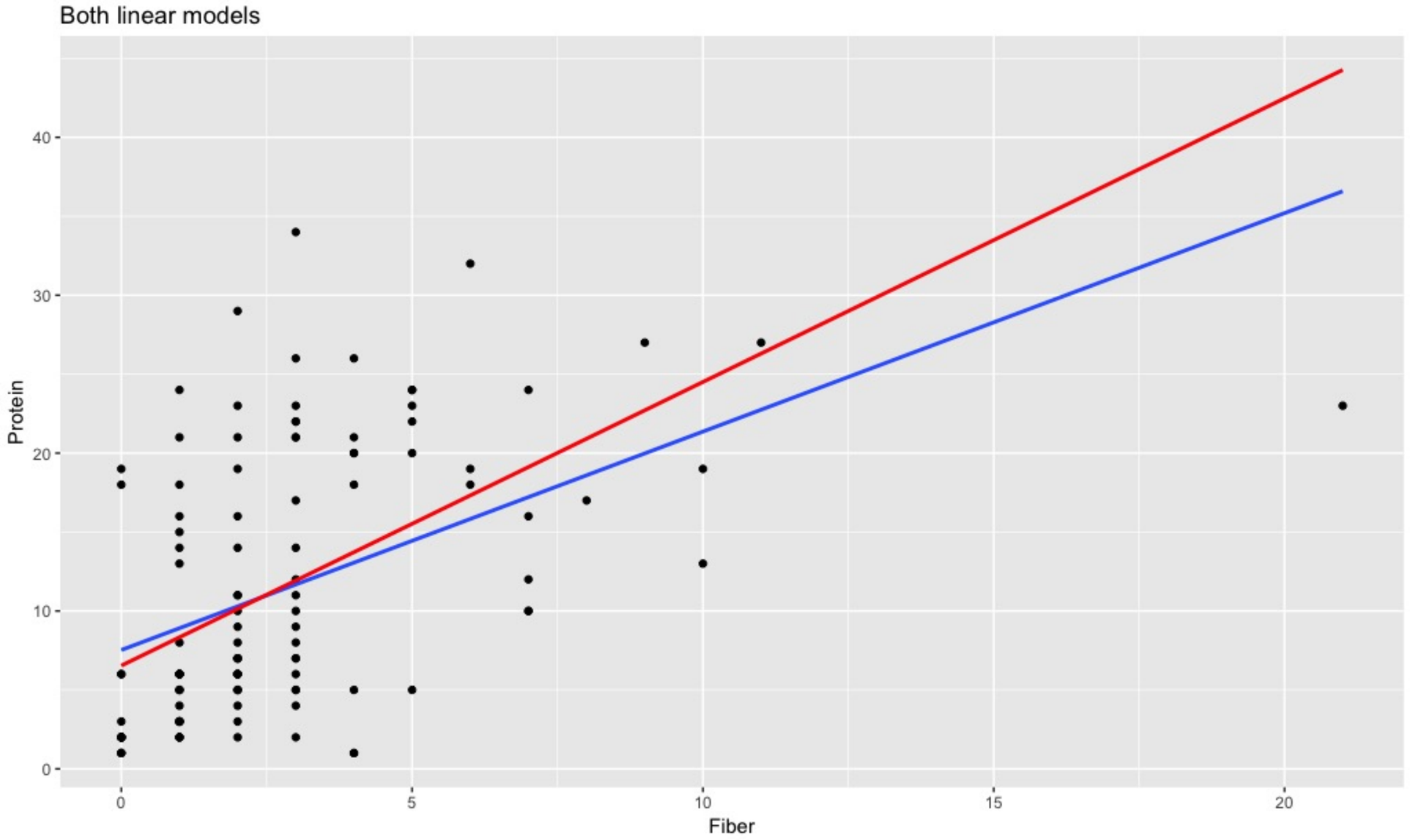
Jo Hardin
Professor, Pomona College


Fiber vs. protein
Protein
Fiber
0
10
20
30
0
5
10
15
20

Different regression lines






Both linear models
Protein
Fiber
0
10
20
30
40
0
5
10
15
20

Different regression models

```
starbucks_lowFib <- starbucks %>% filter(Fiber < 15)
lm(Protein ~ Fiber, data = starbucks) %>% tidy()
```

```
         term estimate std.error statistic      p.value
1 (Intercept) 7.526138 0.9924180  7.583637 1.101756e-11
2       Fiber 1.383684 0.2451395  5.644476 1.286752e-07
```

```
lm(Protein ~ Fiber, data = starbucks_lowFib) %>% tidy()
```

```
         term estimate std.error statistic      p.value
1 (Intercept) 6.537053 1.0633640  6.147521 1.292803e-08
2       Fiber 1.796844 0.2995901  5.997675 2.600224e-08
```

Different regression randomization tests

Full dataset

```
perm_slope %>% mutate(
    abs_perm_slope = abs(stat)) %>%
  summarize(
    p_value = mean(
      abs_perm_slope > abs(obs_slope)
    )
  )
```

```
 A tibble: 1 x 1
  p_value
    <dbl>
1       0
```

Low fiber dataset

```
perm_slope_lowFib %>% mutate(
  abs_perm_slope = abs(stat)) %>%
  summarize(
    p_value = mean(
      abs_perm_slope > abs(obs_slope_lowFib)
    )
  )
```

```
 A tibble: 1 x 1
  p_value
    <dbl>
1       0
```

Let's practice!

INFERENCE FOR LINEAR REGRESSION IN R

Moving forward when model assumptions are violated

INFERENCE FOR LINEAR REGRESSION IN R

Jo Hardin
Professor, Pomona College

datacamp

Linear model

$$Y = \beta_0 + \beta_1 \cdot X + \epsilon$$

$$\text{where } \epsilon \sim N(0, \sigma_\epsilon)$$

Transforming the explanatory variable

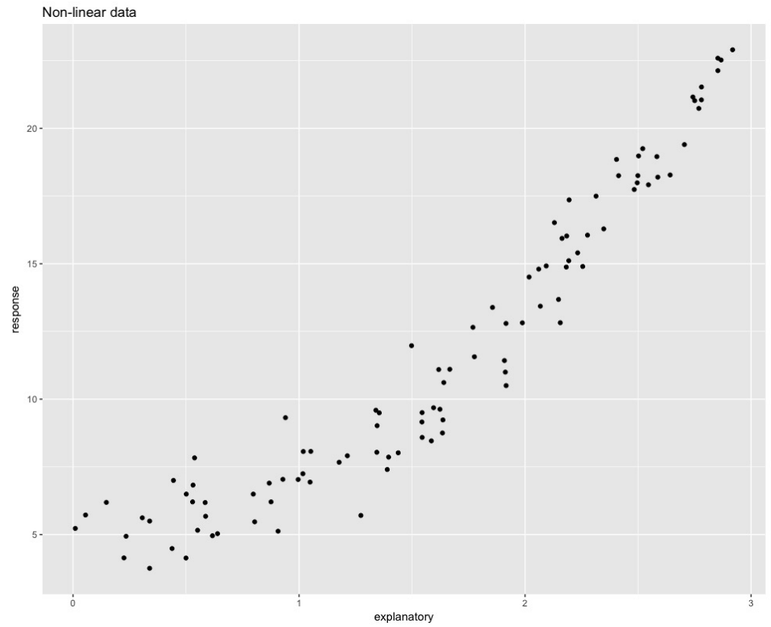
$Y = \beta_0 + \beta_1 \cdot X + \beta_2 \cdot X^2 + \epsilon$, where $\epsilon \sim N(0, \sigma_\epsilon)$

$Y = \beta_0 + \beta_1 \cdot \ln(X) + \epsilon$, where $\epsilon \sim N(0, \sigma_\epsilon)$

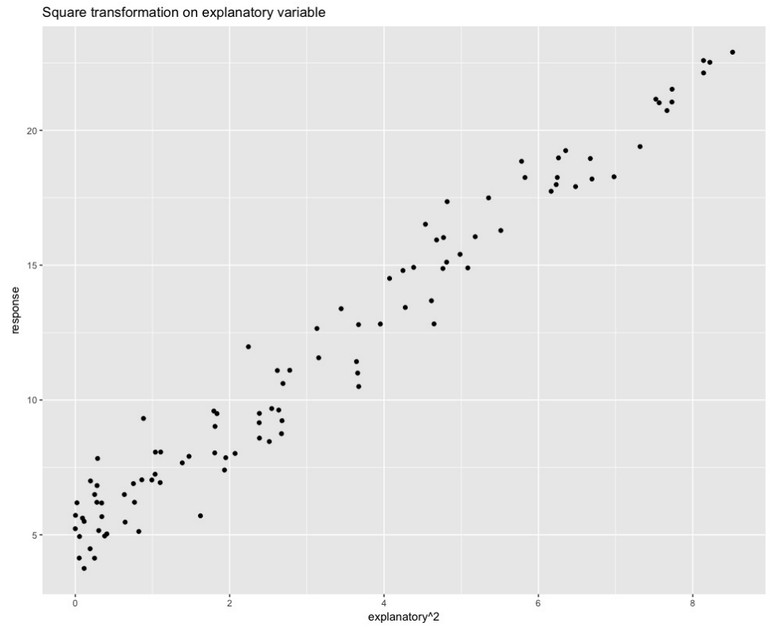
$Y = \beta_0 + \beta_1 \cdot \sqrt{X} + \epsilon$, where $\epsilon \sim N(0, \sigma_\epsilon)$

Squaring the explanatory variable

```
ggplot(data=data_nonlinear,
  aes(x=explanatory, y=response)) +
    geom_point()
```




```
ggplot(data=data_nonlinear,
  aes(x=explanatory^2, y=response))+
    geom_point()
```



Transforming the response variable

$Y^2 = \beta_0 + \beta_1 \cdot X + \epsilon$, where $\epsilon \sim N(0, \sigma_\epsilon)$

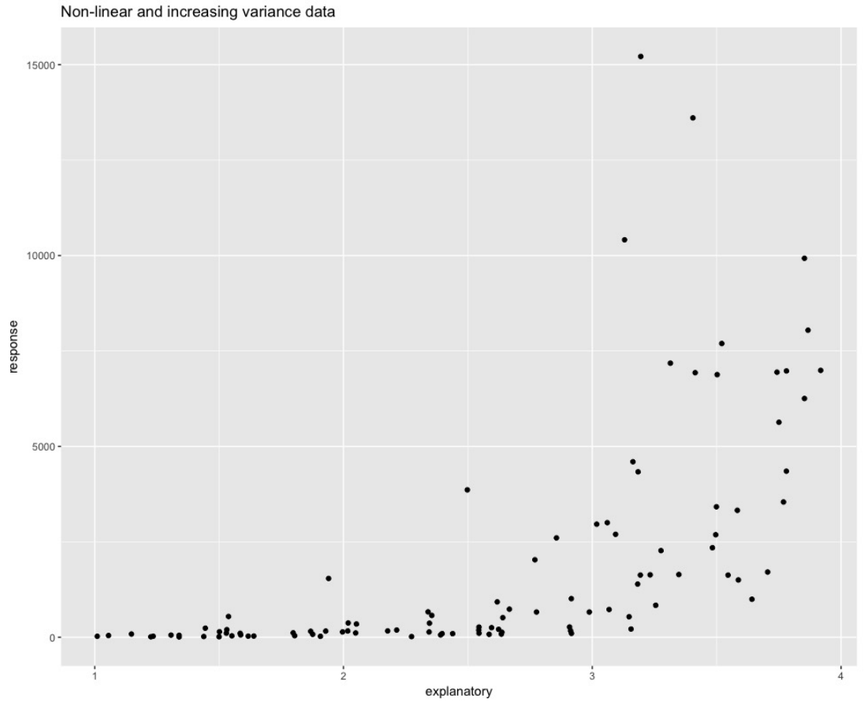
$\ln(Y) = \beta_0 + \beta_1 \cdot X + \epsilon$, where $\epsilon \sim N(0, \sigma_\epsilon)$

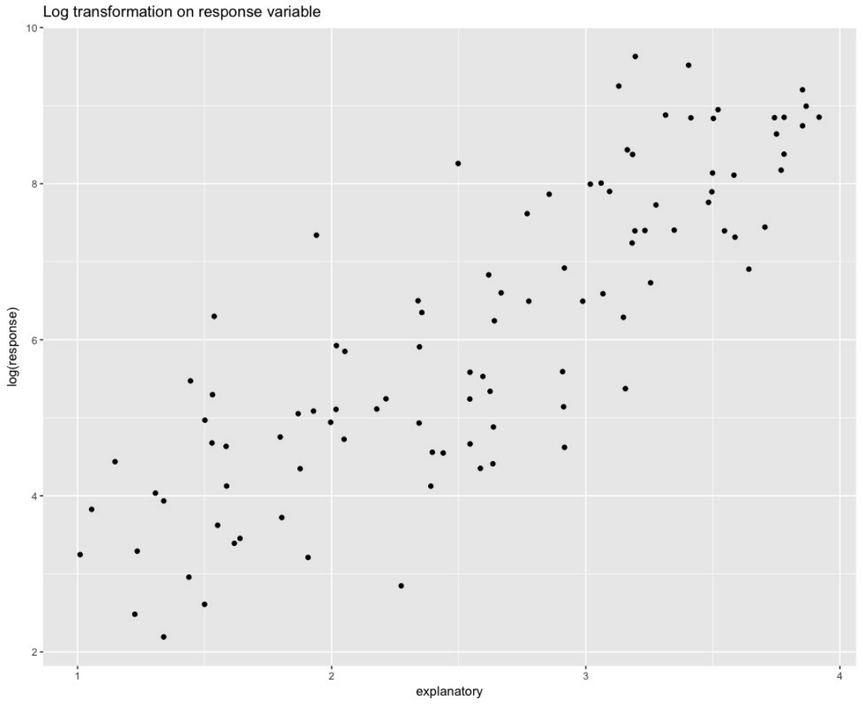
$\sqrt{(Y)} = \beta_0 + \beta_1 \cdot X + \epsilon$, where $\epsilon \sim N(0, \sigma_\epsilon)$

A natural log transformation

```
ggplot(data = data_nonnorm,
  aes(x = explanatory, y = response)) +
    geom_point()
```

```
ggplot(data = data_nonnorm,
  aes(x = explanatory, y = log(response))) +
    geom_point()
```





Let's practice!

INFERENCE FOR LINEAR REGRESSION IN R