431 Class 09

github.com/THOMASELOVE/2019-431

2019-09-24

Today's Agenda (Notes, Chapters 11-13)

1. Building Linear Models
   - Fundamental Summaries of a Regression Model
   - Understanding Regression Residuals
2. Measuring Association with Correlations
   - Pearson and Spearman approaches
   - Thinking about the impact of transformations
3. Adding a categorical predictor (factor) to a model
   - Using `fct_recode` from `forcats` (tidyverse)
   - Interpreting an indicator variable regression

What will we hear about today?

- The central role of linear regression in understanding associations between quantitative variables.
- The interpretation of a regression model as a prediction model.
- Assessment of key regression summaries, including residuals.
- Using `tidy`, `glance` and `augment` from `broom` to summarize a model.
- Measuring association through correlation coefficients.
- How we might think about "adjusting" for the effect of a categorical predictor on a relationship between two quantitative ones.
- How a transformation might help us "linearize" the relationship shown in a scatterplot.

Installing the patchwork package

I'll be using the patchwork package today (and in the future) to build composite plots from ggplot. To install the patchwork package on your system, use the following code:

```
devtools::install_github("thomasp85/patchwork")
```

- Visit https://github.com/thomasp85/patchwork for more on patchwork.
- Other ways to compose plots include grid.arrange() from gridExtra and plot_grid() from cowplot.

Today's Packages and Loading the VHL Data

```
library(magrittr); library(janitor); library(patchwork)
library(broom); library(tidyverse)
```

```
VHL <- read_csv("vonHippel-Lindau.csv")
```

VHL Variables

- p.ne = plasma norepinephrine (pg/ml)
- tumorvol = tumor volume (ml)
- disease = 1 for patients with multiple endocrine neoplasia type 2
- disease = 0 for patients with von Hippel-Lindau disease

A Simple Linear Regression

model1: A Linear Model for p.ne based on tumorvol

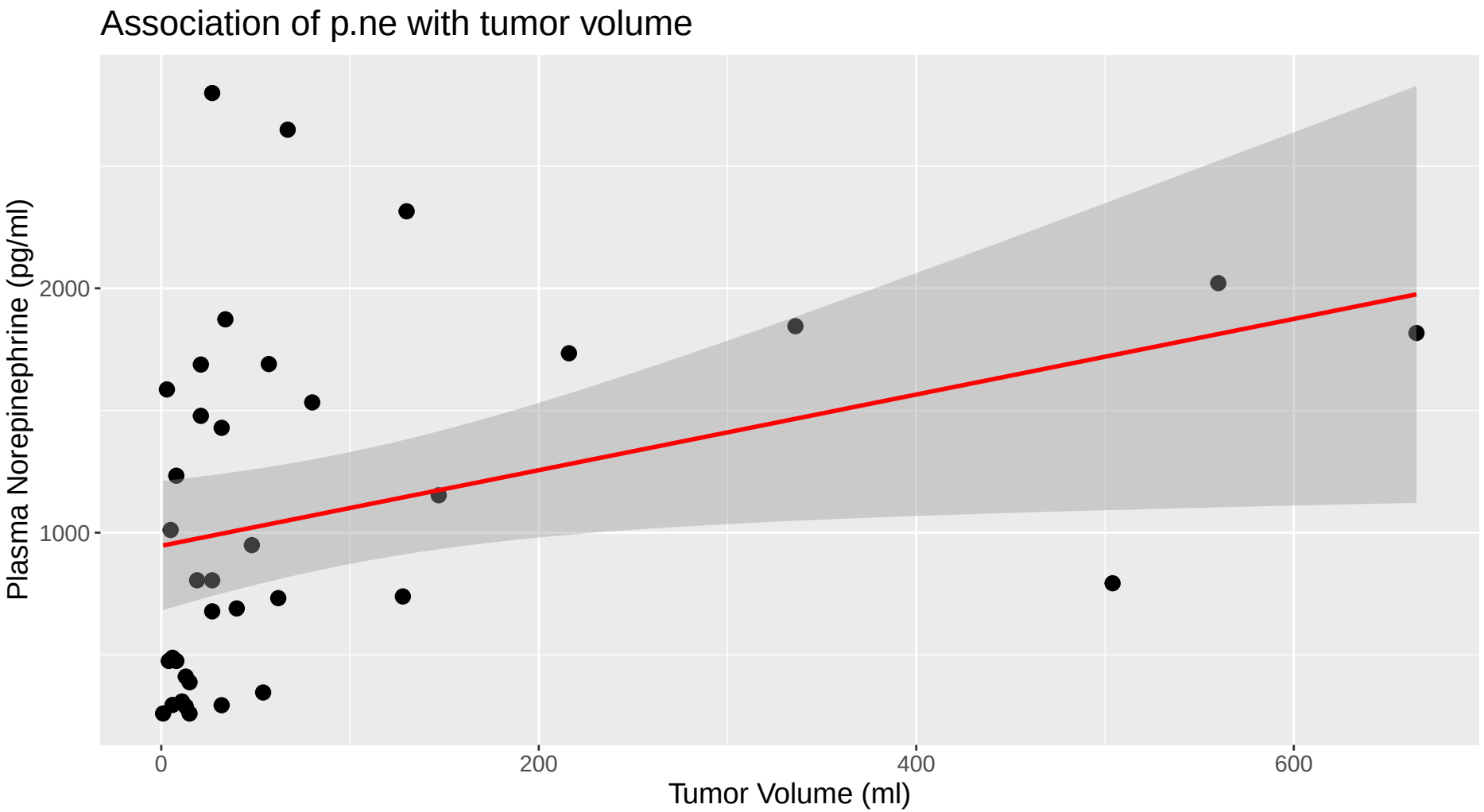
```
model1 <- lm(p.ne ~ tumorvol, data = VHL)
model1
```

```

Call:
lm(formula = p.ne ~ tumorvol, data = VHL)

Coefficients:
(Intercept)     tumorvol
    946.185        1.547
```

The (simple regression / prediction / ordinary least squares) model is

- p.ne = 946.2 + 1.55 * tumorvol.

Linear model using ordinary least squares (OLS).



Summary of our Linear (OLS) Model

```
> summary(model1)

Call:
lm(formula = p.ne ~ tumorvol, data = VHL)

Residuals:
   Min     1Q Median     3Q    Max
-933.1 -555.3 -170.6  453.6 1811.0

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 946.1846   130.4810   7.252 1.81e-08 ***
tumorvol      1.5474     0.7079   2.186   0.0356 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 685.2 on 35 degrees of freedom
Multiple R-squared:  0.1201,    Adjusted R-squared:  0.09497
F-statistic: 4.778 on 1 and 35 DF,  p-value: 0.03561
```

Key Elements of the Summary (1)

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 946.1846   130.4810   7.252 1.81e-08 ***
tumorvol      1.5474     0.7079   2.186   0.0356 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- The straight line model for these data fitted by ordinary least squares is p.ne = 946 + 1.55 `tumorvol`.
- The slope of `tumorvol` is positive, which indicates that as `tumorvol` increases, we expect that `p.ne` will also increase.
- Specifically, we expect that for every additional ml of `tumorvol`, the `p.ne` is increased by 1.55 pg/ml.

Tidying the Model Coefficients

```
model1 <- lm(p.ne ~ tumorvol, data = VHL)

tidy(model1, conf.int = TRUE, conf.level = 0.90) %>%
  knitr::kable(digits = 2)
```

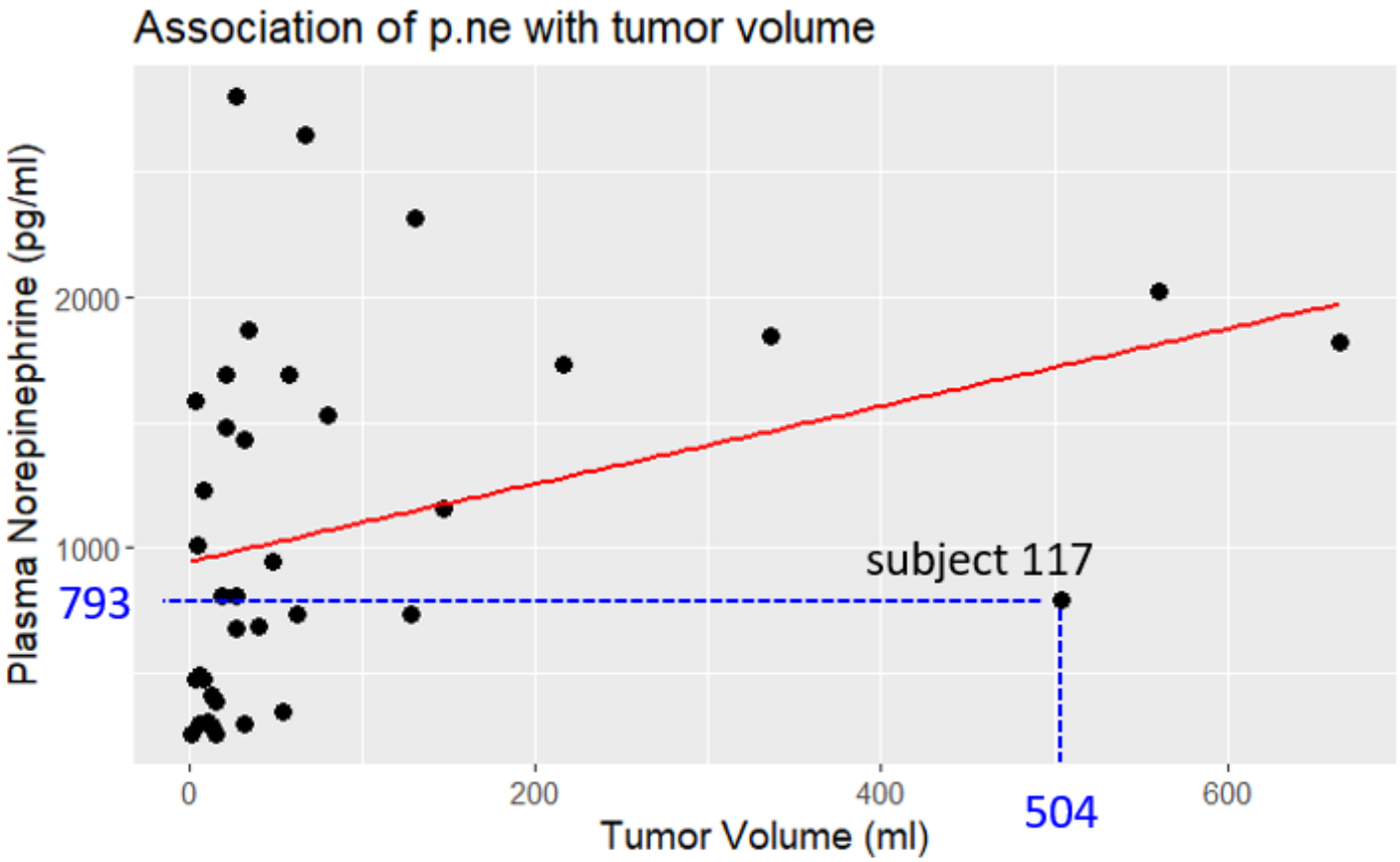
term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	946.18	130.48	7.25	0.00	725.73	1166.64
tumorvol	1.55	0.71	2.19	0.04	0.35	2.74

Key Elements of the Summary (2)

```
Call:
lm(formula = p.ne ~ tumorvol, data = VHL)

Residuals:
    Min      1Q  Median      3Q     Max
-933.1 -555.3 -170.6  453.6 1811.0
```

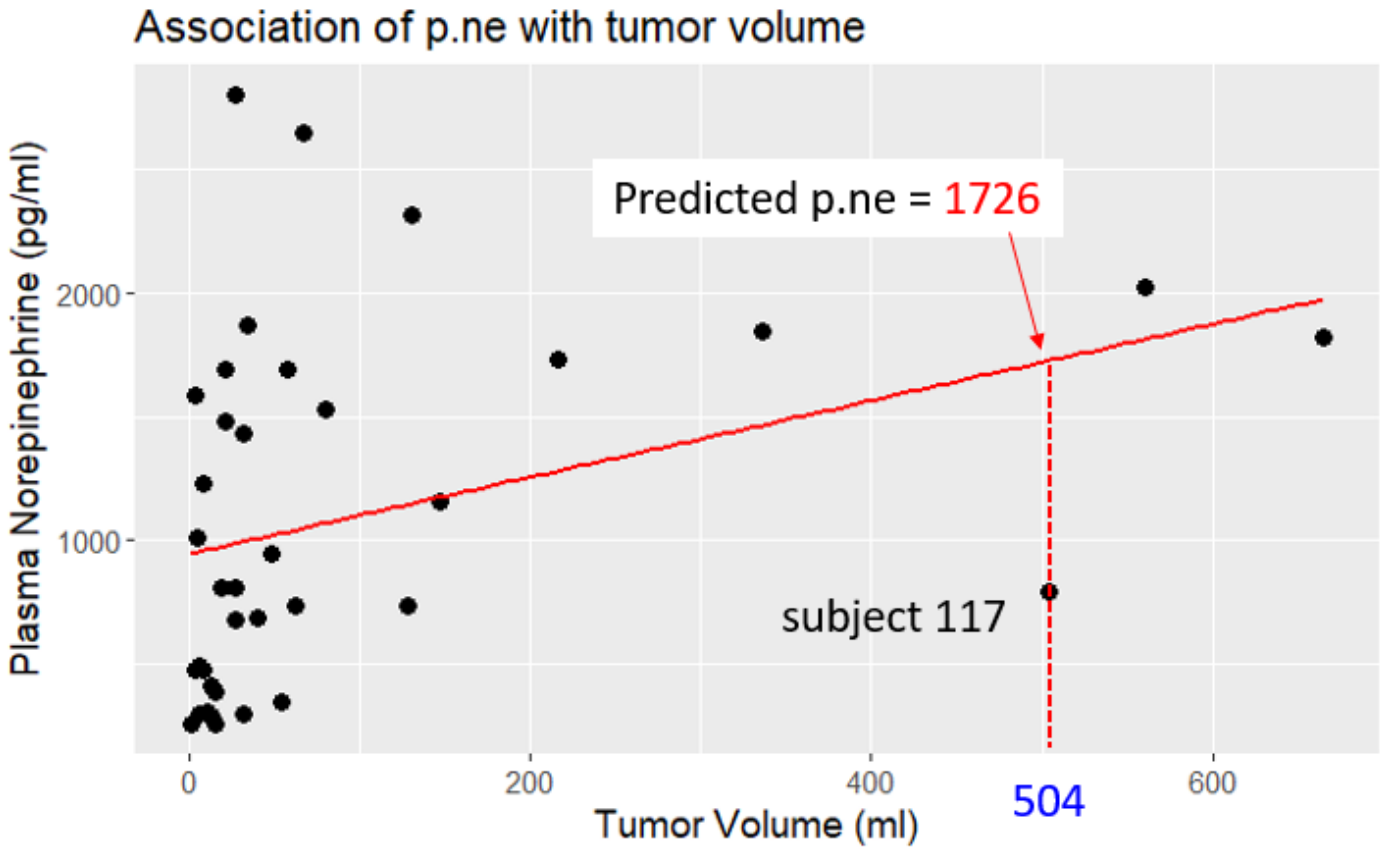
- Here, the **outcome** is `p.ne`, and the **predictor** is `tumorvol`.
- The **residuals** are the observed `p.ne` values minus the model's predicted `p.ne`. The sample residuals are the prediction errors.
  - The biggest miss is for a subject whose observed `p.ne` was 1,811 pg/nl higher than the model predicts based on the subject's tumor volume.
  - The mean residual will always be zero in an OLS model.

Understanding Regression Residuals (A)



Subject 117 has tumorvol = 504, and observed p.ne = 793 pg/nl.

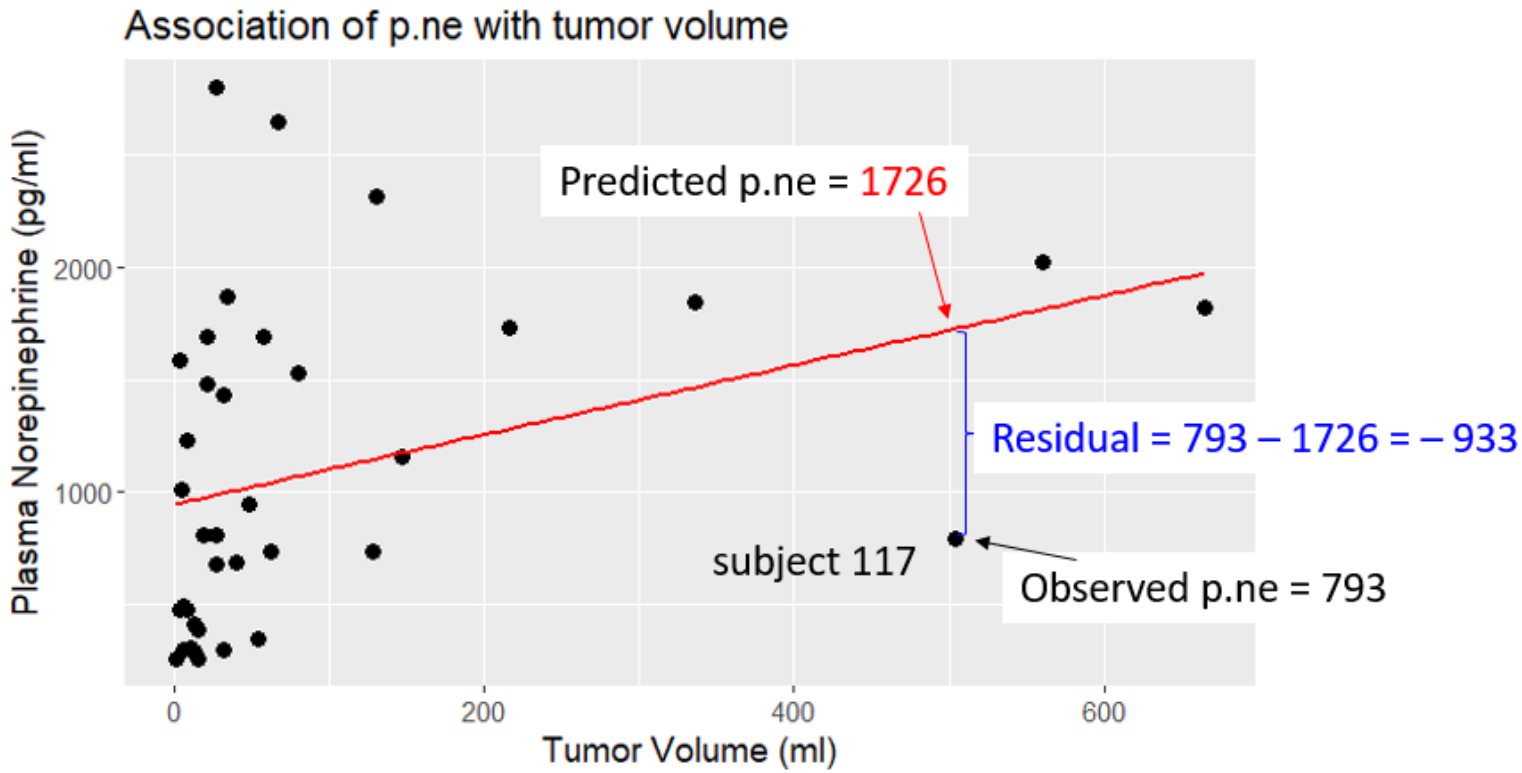
Understanding Regression Residuals (B)



Subject 117 has tumorvol = 504, and observed p.ne = 793 pg/nl.

Model predicts p.ne is 946.2 + 1.55(504) = 1726 pg/nl.

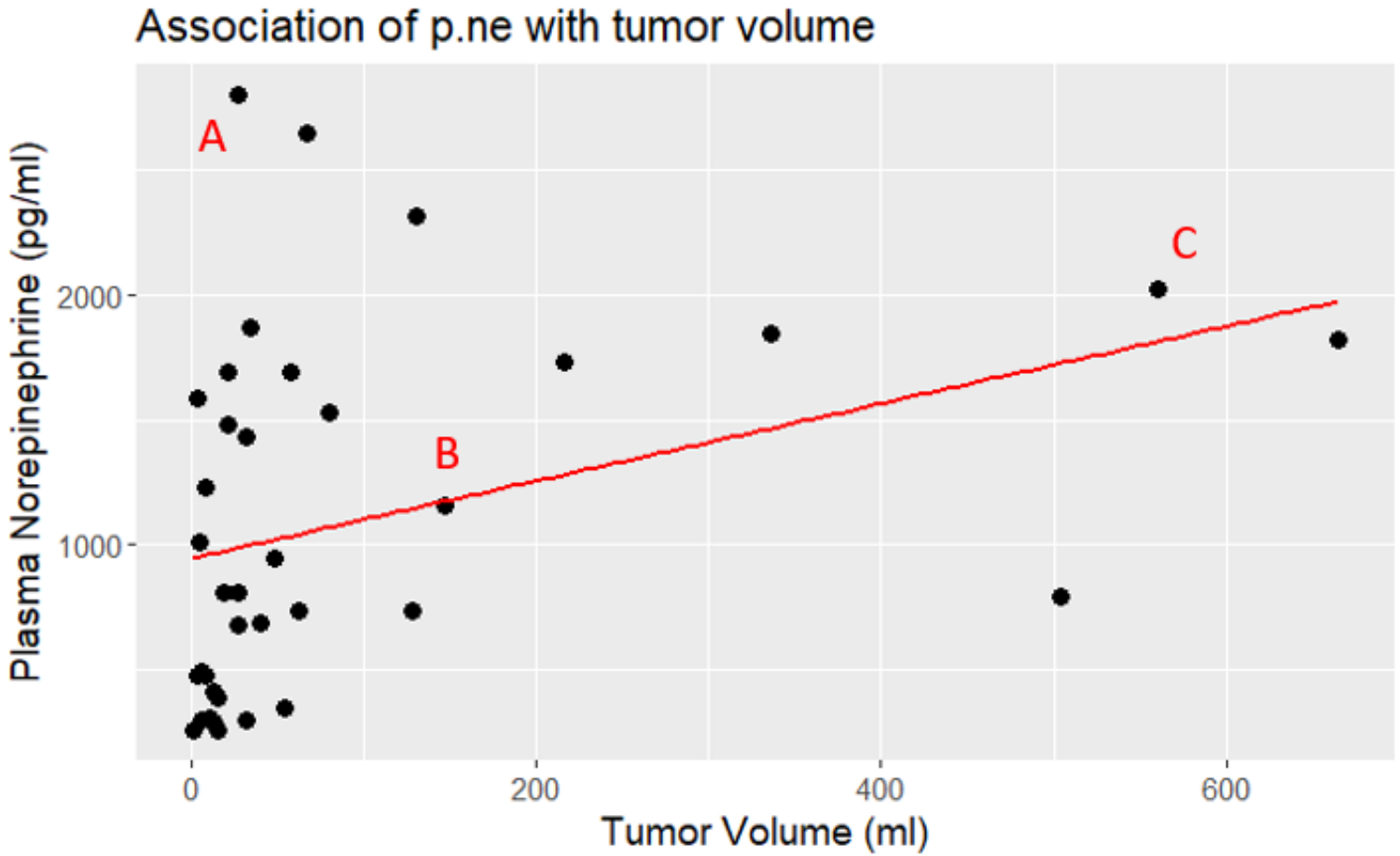
Understanding Regression Residuals (C)



Subject 117 has tumorvol = 504, and observed p.ne = 793 pg/nl.

Model predicts p.ne is 946.2 + 1.55(504) = 1726. So, residual = 793 – 1726 = **-933**

Understanding Regression Residuals (D)



Which point (A, B or C) has the largest positive residual?

Do the residuals follow a Normal model well?

```
model1$residuals %>% round(digits = 1)
```

```
     1      2      3      4      5      6      7 
-677.3 -701.7 1811.0 1599.1 -683.7  655.6 -170.6 
     8      9     10     11     12     13     14 
 -20.7 -310.0 -158.2 -581.4 -660.5  208.3  874.2 
    15     16     17     18     19     20     21 
-477.4 1167.7 -933.1 -654.2 -687.7  709.3 -555.3 
    22     23     24     25     26     27     28 
 378.9 -310.1 -467.5  -71.5  499.3 -183.0  274.4 
    29     30     31     32     33     34     35 
-483.6   57.1 -405.3 -709.4  433.3 -318.1  635.2 
    36     37 
 453.6  463.0 
```

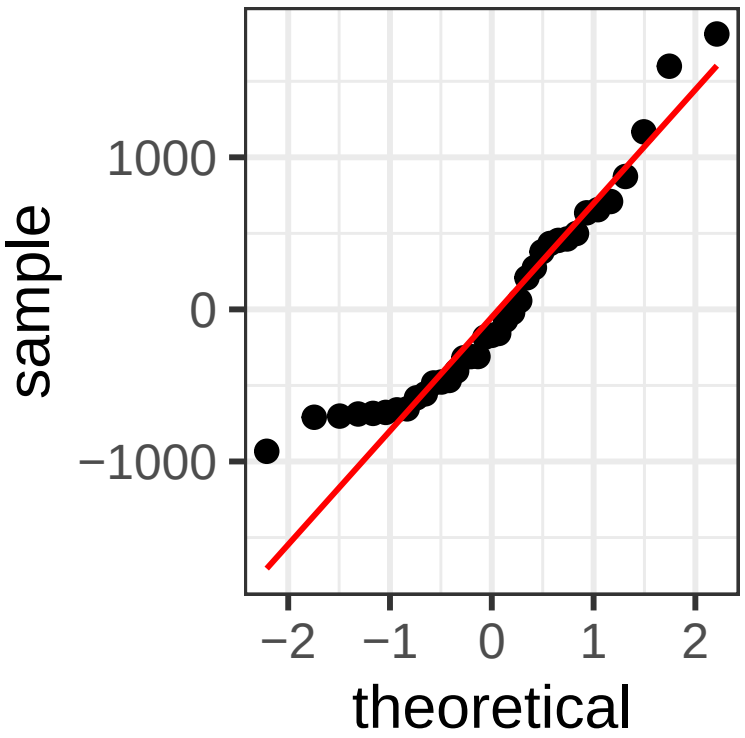
Residuals from `model1`

```
model1_aug <- broom::augment(model1)

head(model1_aug,3)
```

```
# A tibble: 3 x 9
   p.ne tumorvol .fitted .se.fit .resid   .hat .sigma
  <dbl>    <dbl>   <dbl>   <dbl>  <dbl>  <dbl>  <dbl>
1   289       13    966.    126.  -677. 0.0339   685.
2   294       32    996.    121.  -702. 0.0310   684.
3  2799       27    988.    122.  1811. 0.0317   619.
# ... with 2 more variables: .cooksd <dbl>,
#   .std.resid <dbl>
```

model1 residuals: Normally distributed?

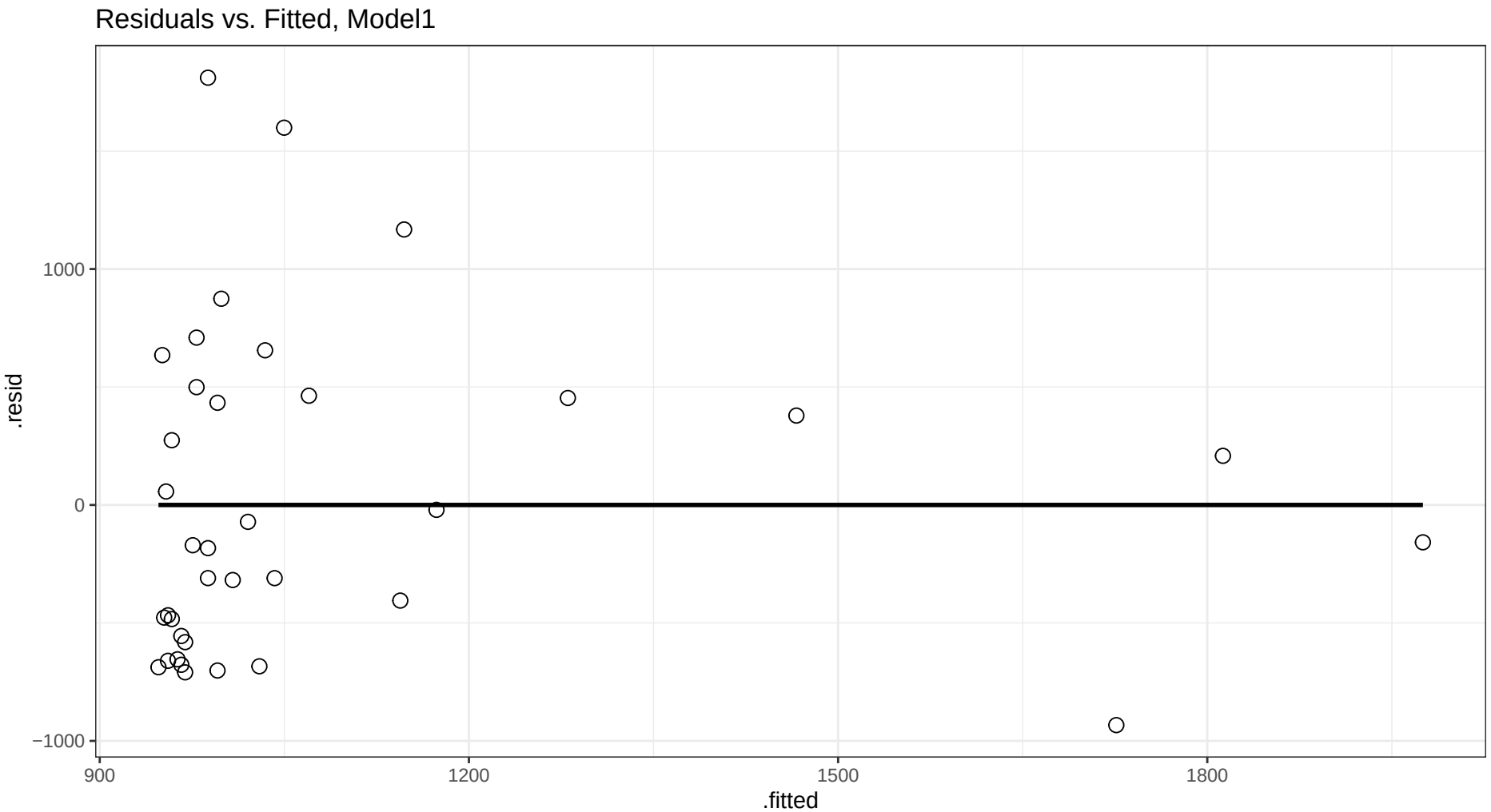
```
ggplot(model1_aug, aes(sample = .resid)) +
  geom_qq() + geom_qq_line(col = "red") + theme_bw()
```



Residuals vs. Fitted Values plot

```
ggplot(model1_aug, aes(x = .fitted, y = .resid)) +
  geom_point(shape = 1, size = 3) +
  geom_smooth(method = "lm", se = FALSE, col = "black") +
  theme_bw() +
  labs(title = "Residuals vs. Fitted, Model1")
```

Residuals vs. Fitted Values plot



Key Elements of the Summary (3)

```
Residual standard error: 685.2 on 35 degrees of freedom
Multiple R-squared:  0.1201,    Adjusted R-squared:  0.09497
F-statistic: 4.778 on 1 and 35 DF,  p-value: 0.03561
```

- The multiple R-squared (squared correlation coefficient) is 0.12, which implies that 12% of the variation in `p.ne` is explained using this linear model with `tumorvol`.
- It also implies that the Pearson correlation between `p.ne` and `tumorvol` is the square root of 0.12, or 0.347.

```
cor(VHL$p.ne, VHL$tumorvol)
```

```
[1] 0.3465646
```

Model 1, summarized at a glance, with broom

```
broom::glance(model1)
```

```
# A tibble: 1 x 11
  r.squared adj.r.squared sigma statistic p.value    df
      <dbl>         <dbl> <dbl>     <dbl>   <dbl> <int>
1     0.120        0.0950  685.      4.78  0.0356     2
# ... with 5 more variables: logLik <dbl>, AIC <dbl>,
#   BIC <dbl>, deviance <dbl>, df.residual <int>
```

Key Elements of glance for us now...

```
glance(model1) %>%
  select(r.squared, adj.r.squared, sigma) %>%
  knitr::kable(digits = 3)
```

r.squared	adj.r.squared	sigma
0.12	0.095	685.168

Measuring Correlation between Quantities

Correlation Coefficients

Two key types of correlation coefficient to describe an association between quantities.

- The one most often used is called the *Pearson* correlation coefficient, symbolized r or sometimes rho (ρ).
- Another is the Spearman rank correlation coefficient, also symbolized by ρ, or sometimes ρ_s.

```
cor(VHL$p.ne, VHL$tumorvol)
```

```
[1] 0.3465646
```

```
cor(VHL$p.ne, VHL$tumorvol, method = "spearman")
```

```
[1] 0.5414319
```

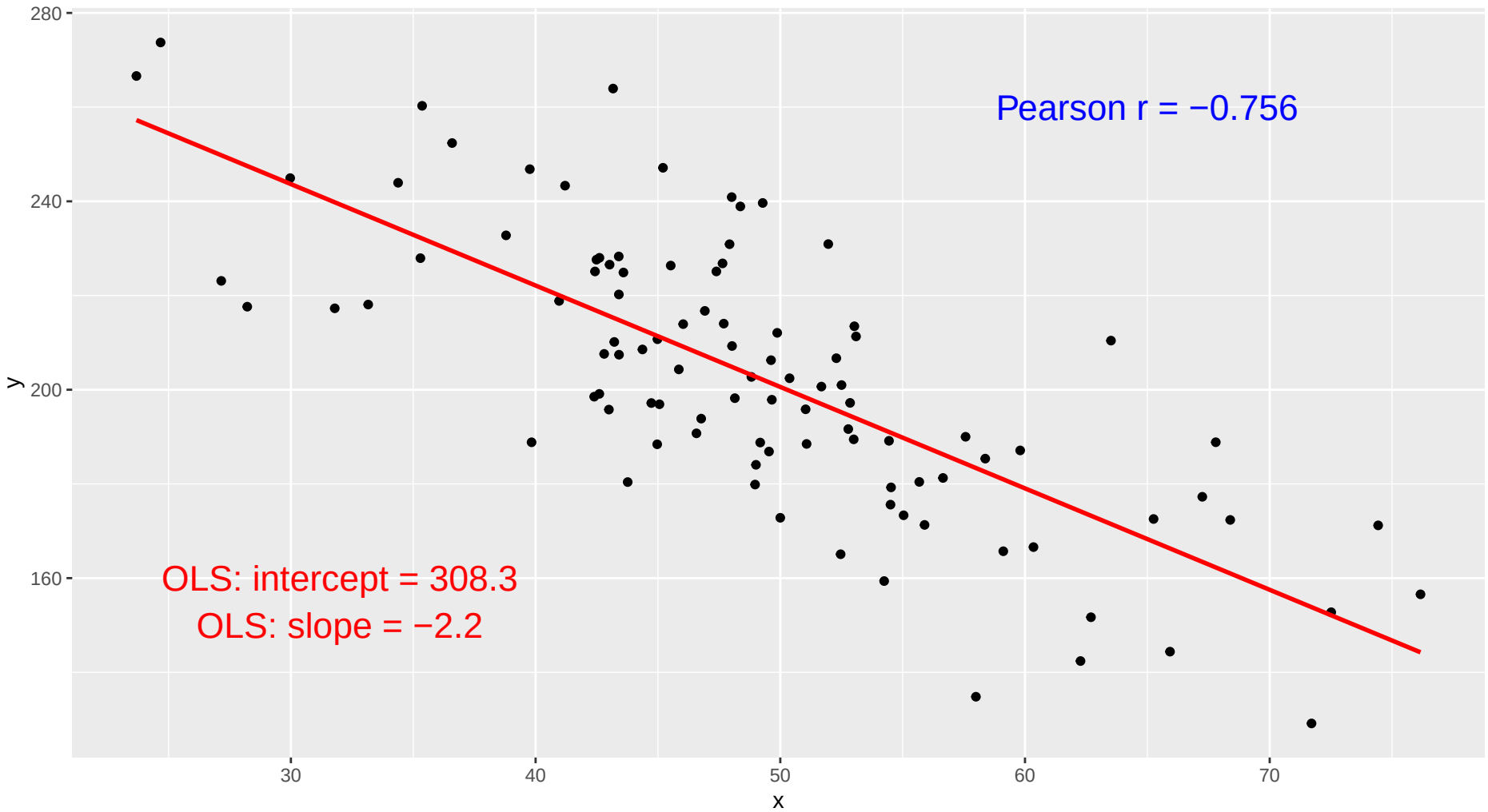
Meaning of Pearson Correlation

The Pearson correlation coefficient assesses how well the relationship between X and Y can be described using a linear function.

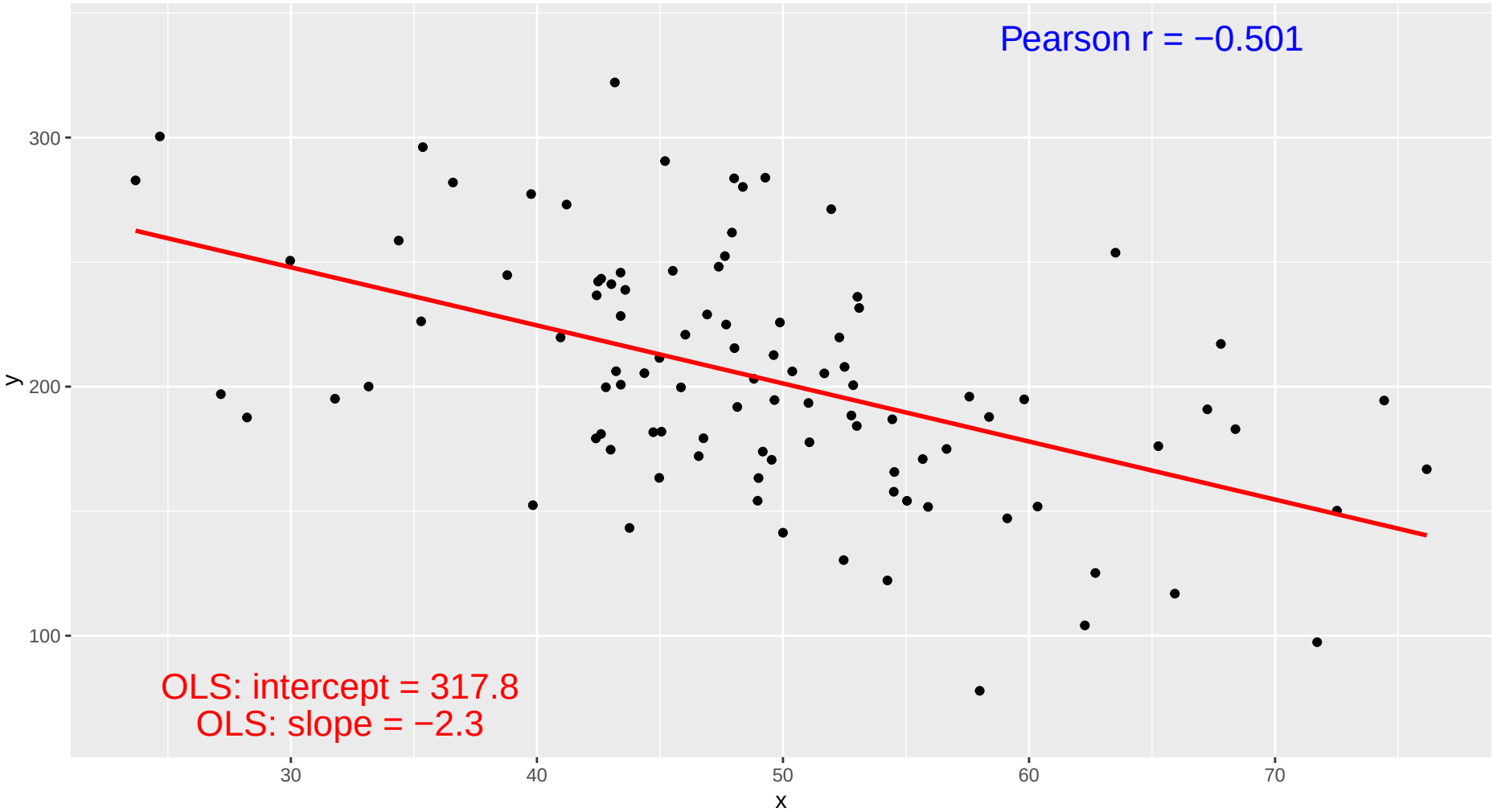
- The Pearson correlation is dimension-free.
- It falls between -1 and +1, with the extremes corresponding to situations where all the points in a scatterplot fall exactly on a straight line with negative and positive slopes, respectively.
- A Pearson correlation of zero corresponds to the situation where there is no linear association.
- Unlike the estimated slope in a regression line, the sample correlation coefficient is symmetric in x and y, so it does not depend on labeling one of them (y) the response variable, and one of them (x) the predictor.

$$r_{XY} = \frac{1}{n-1} \Sigma_{i=1}^{n} \left(\frac{x_i - \bar{x}}{s_x} \right) \left(\frac{y_i - \bar{y}}{s_y} \right)$$

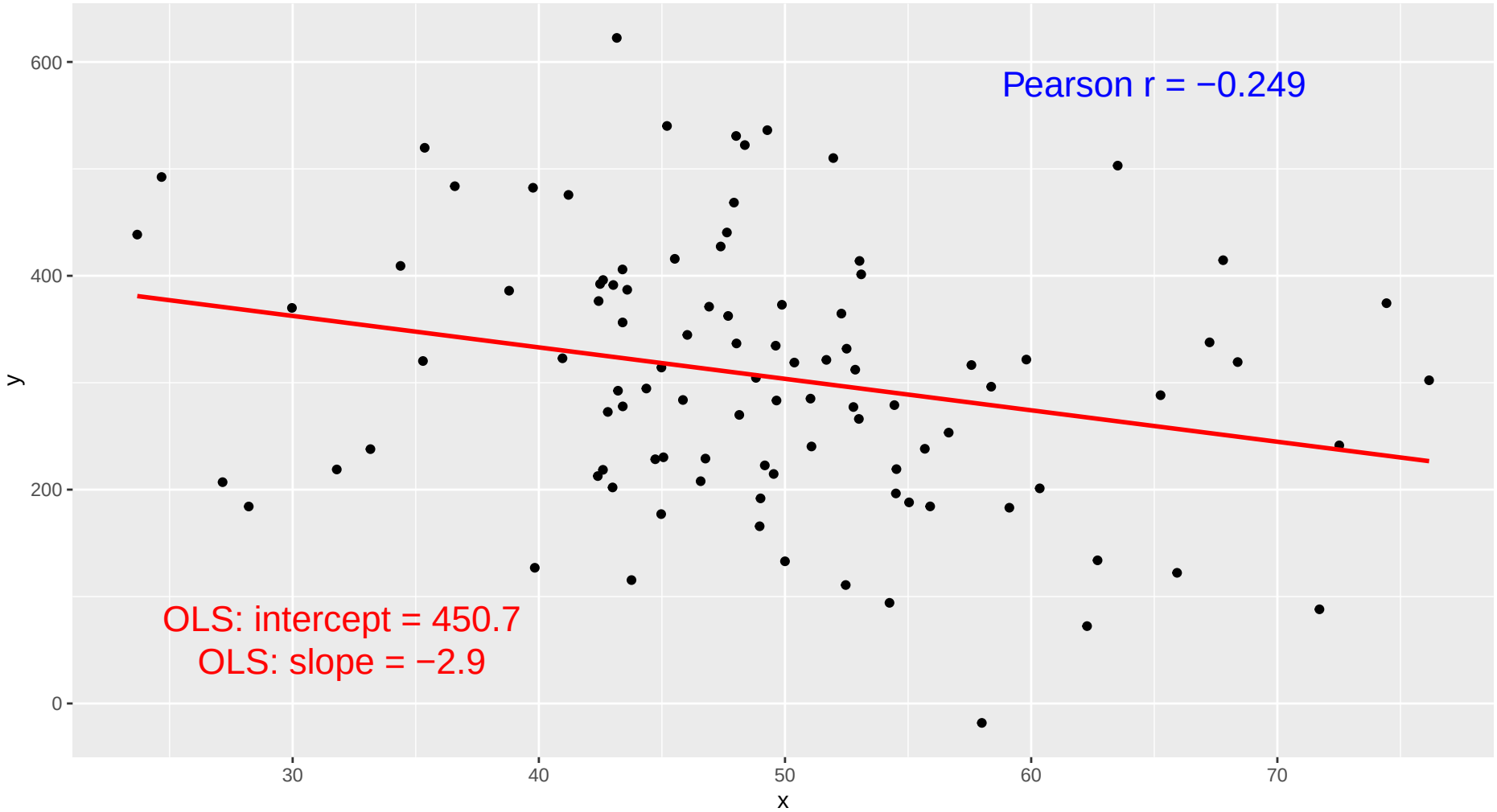
Simulated Example 1



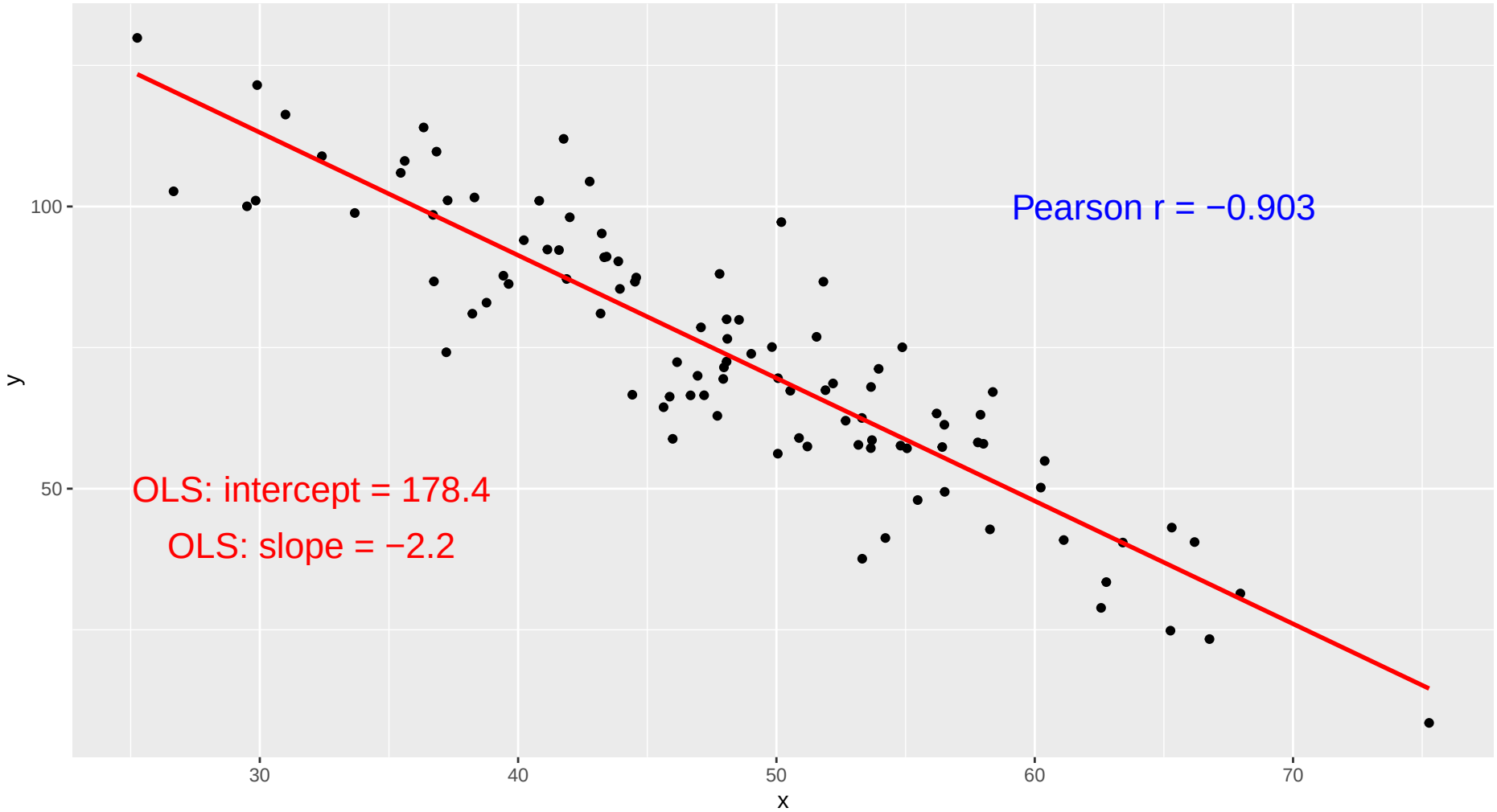
Simulated Example 2



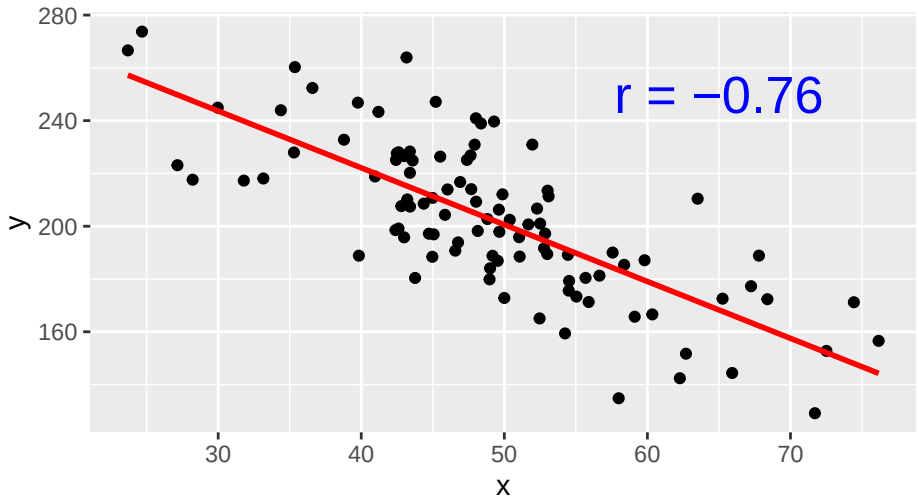
Simulated Example 3

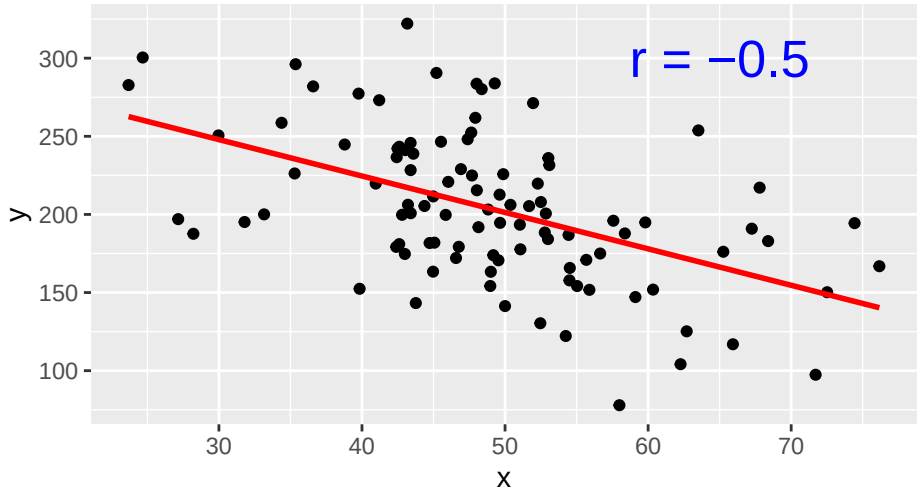


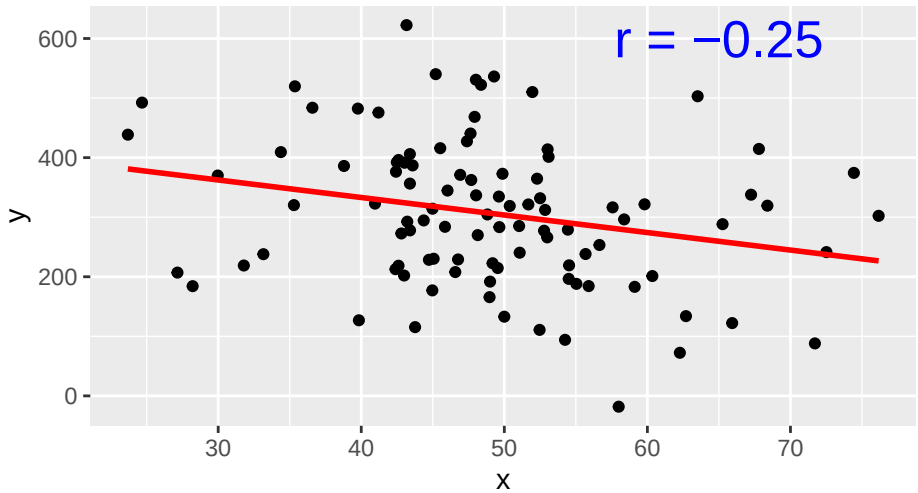
Simulated Example 4

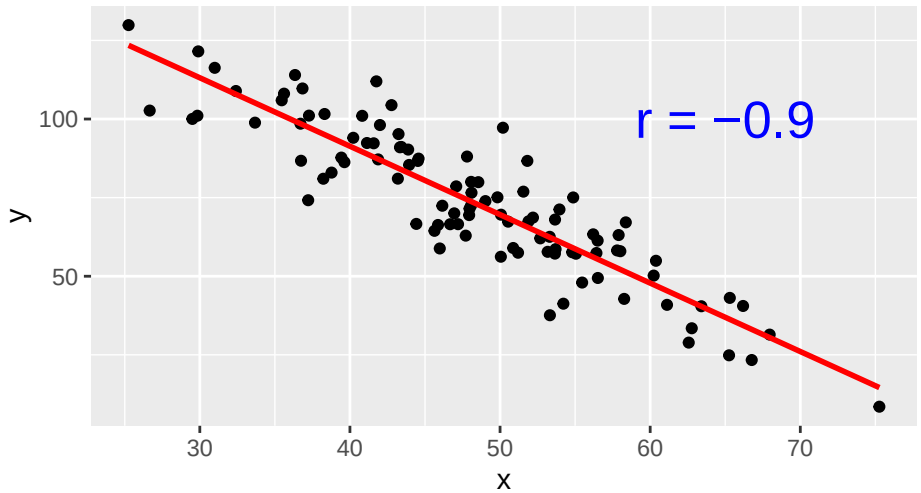


Calibrate Yourself on Correlation Coefficients

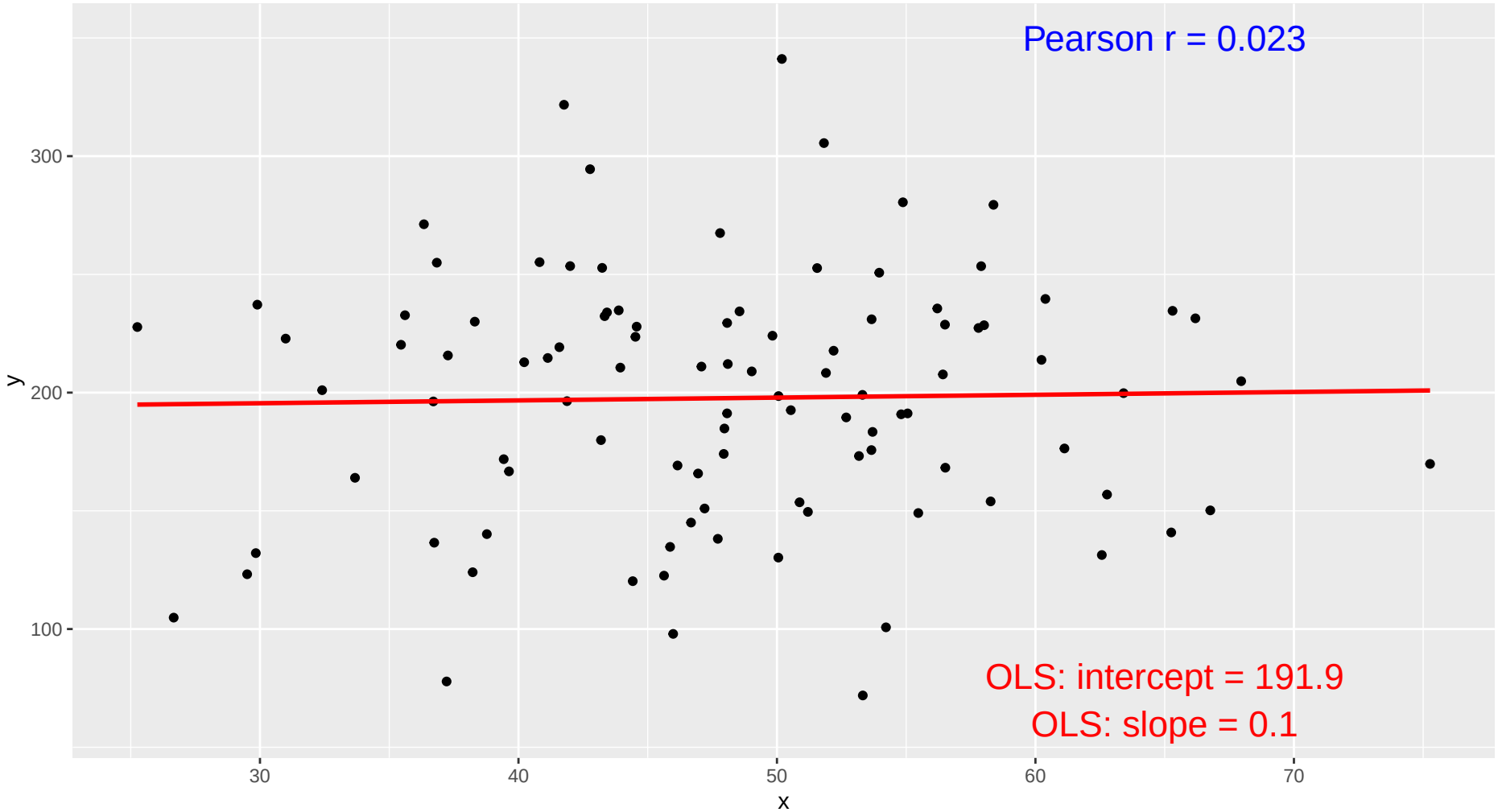




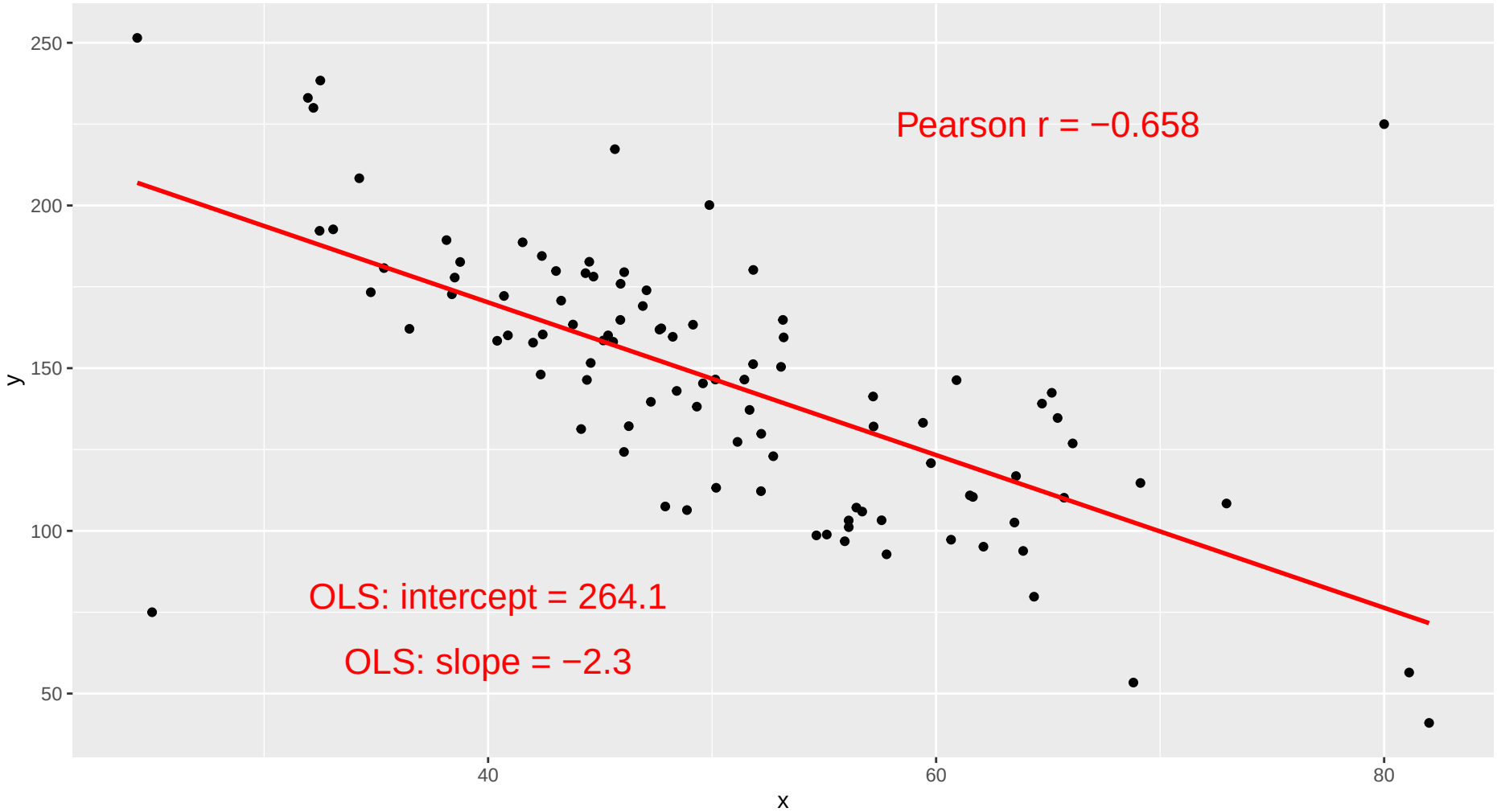




Simulated Example 5

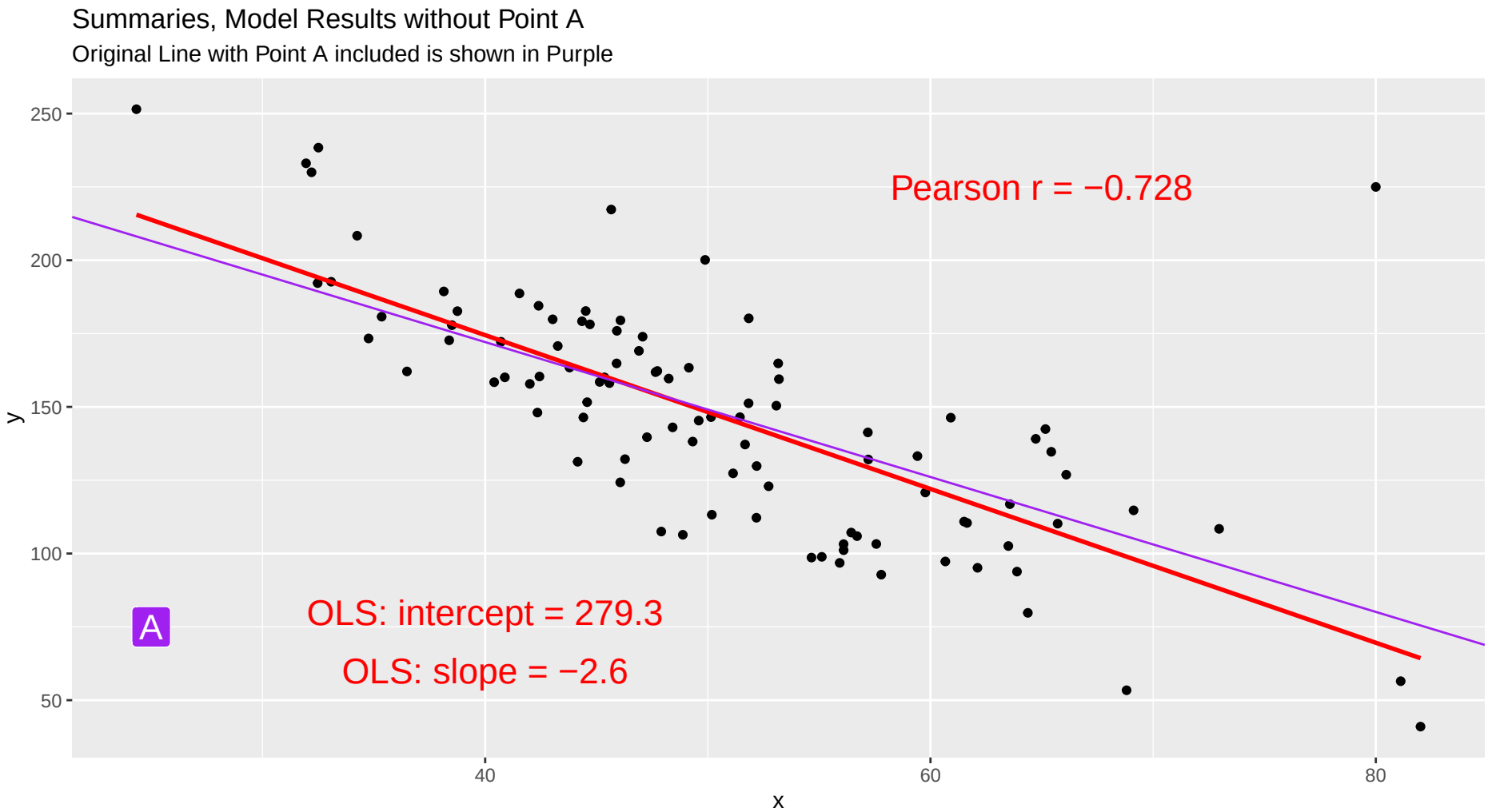


Simulated Example 6

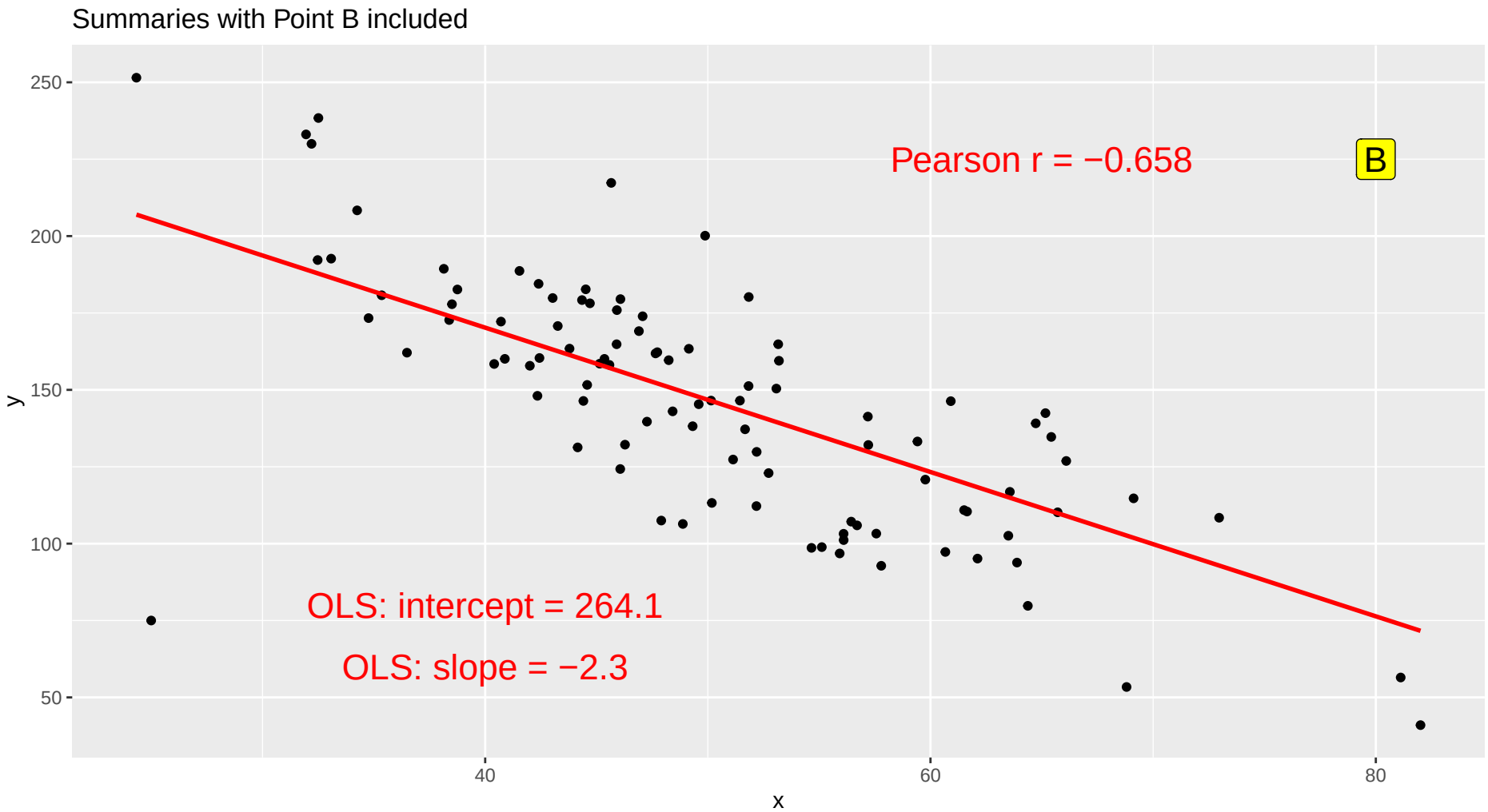


Example 6: What would happen if we omit Point A?

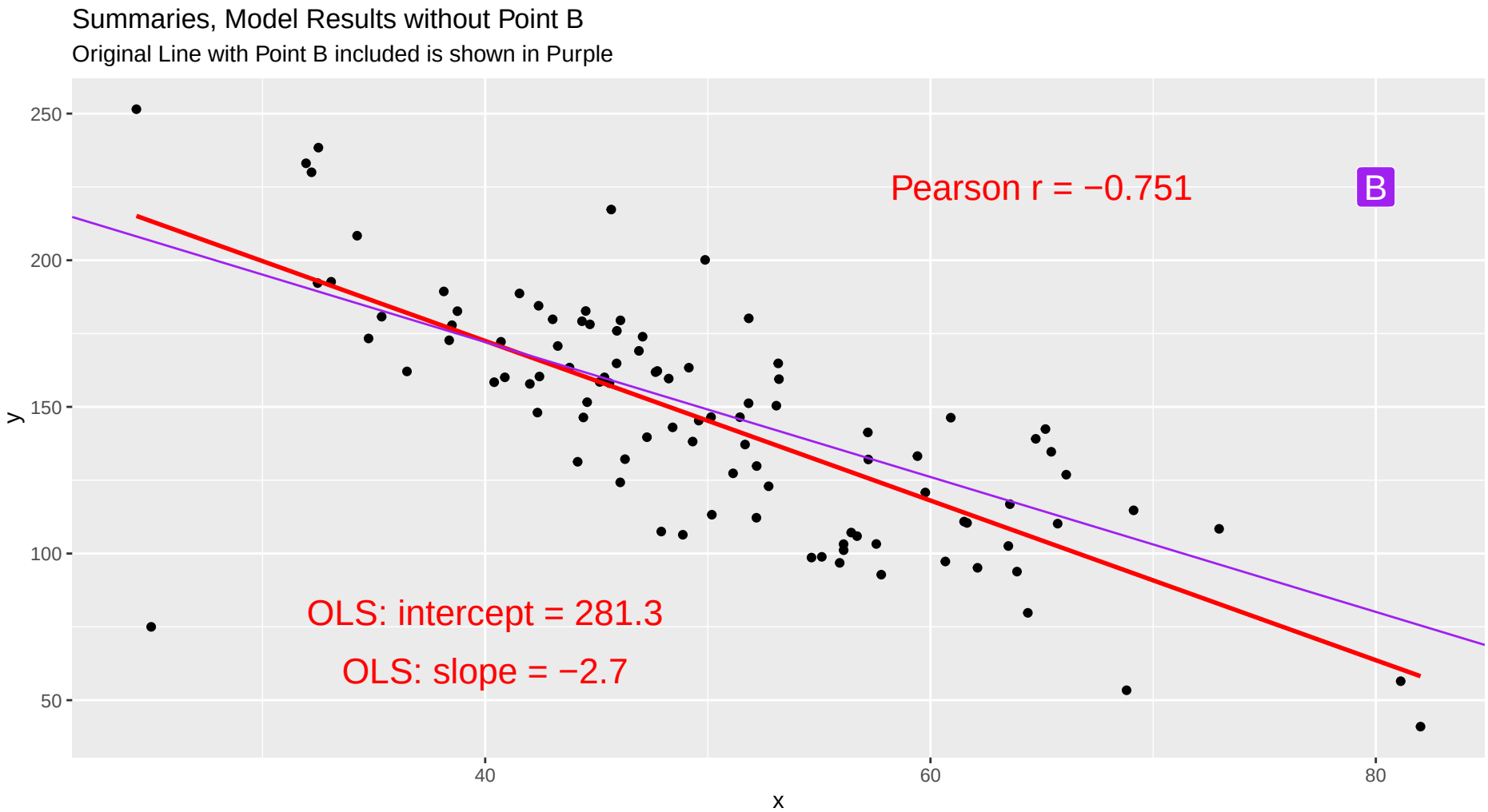
Example 6: Result if we omit Point A



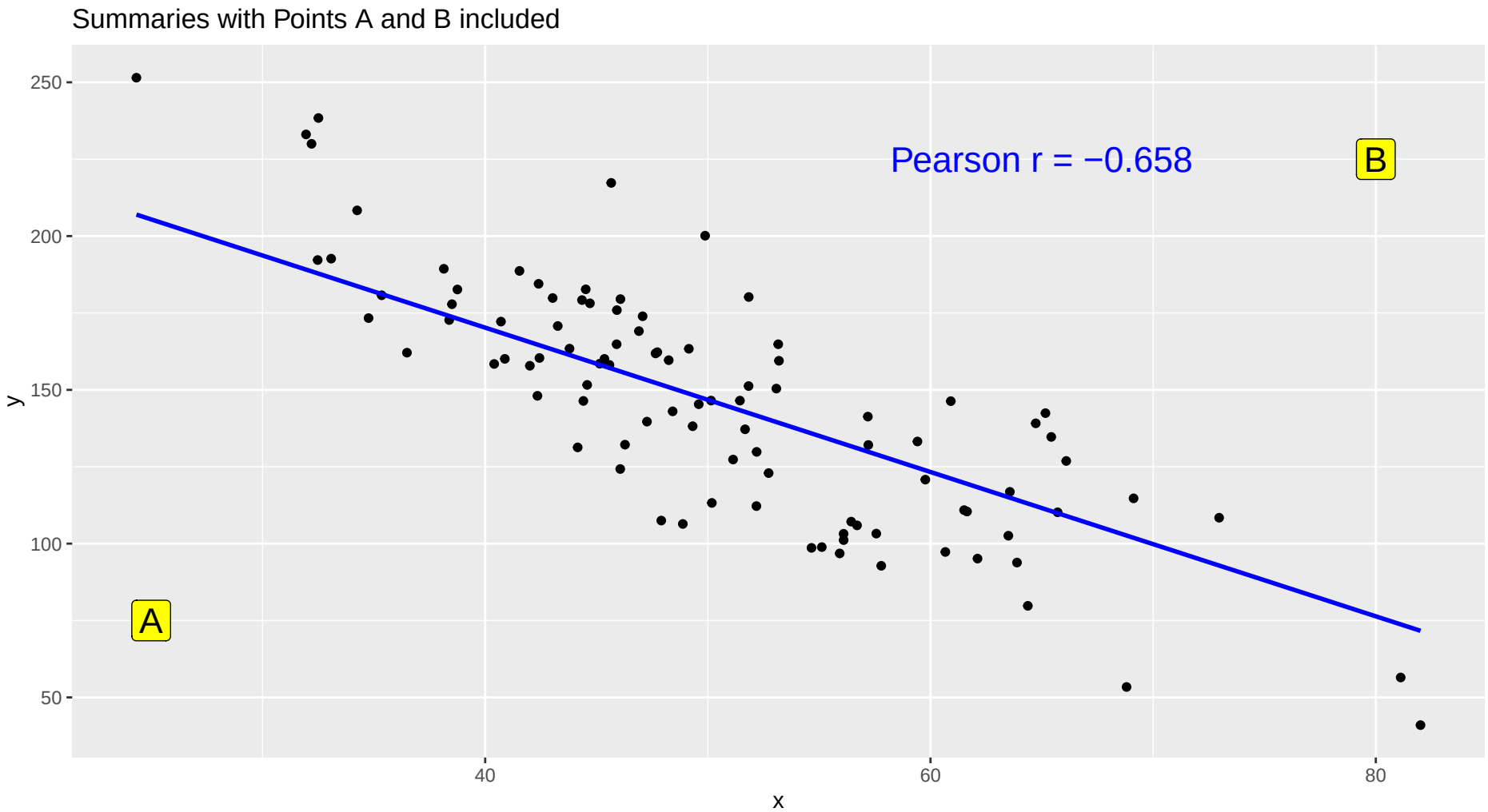
Example 6: What would happen if we omit Point B?



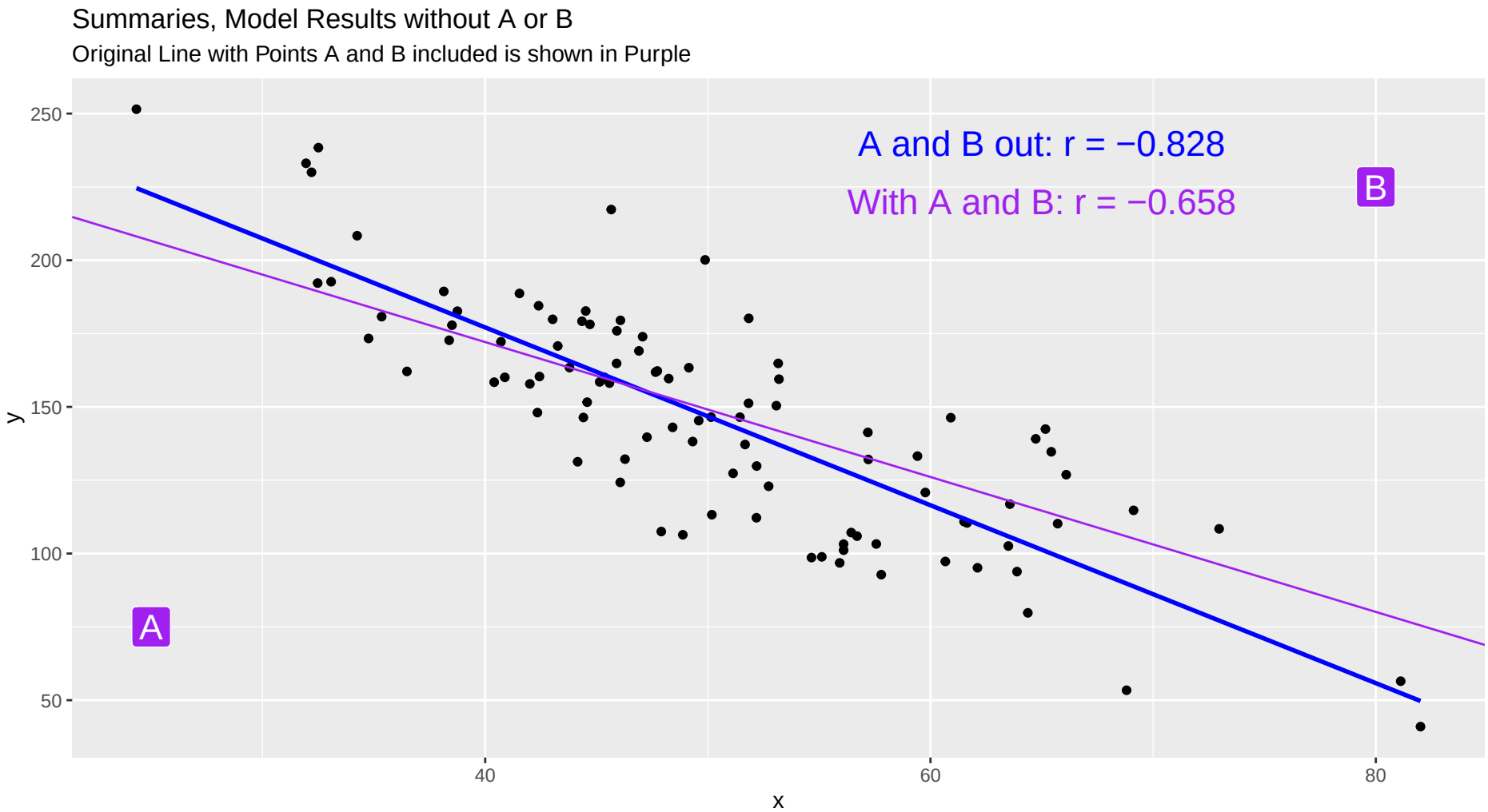
Example 6: Result if we omit Point B



Example 6: What if we omit Point A AND Point B?



Example 6: Result if we omit Points A and B

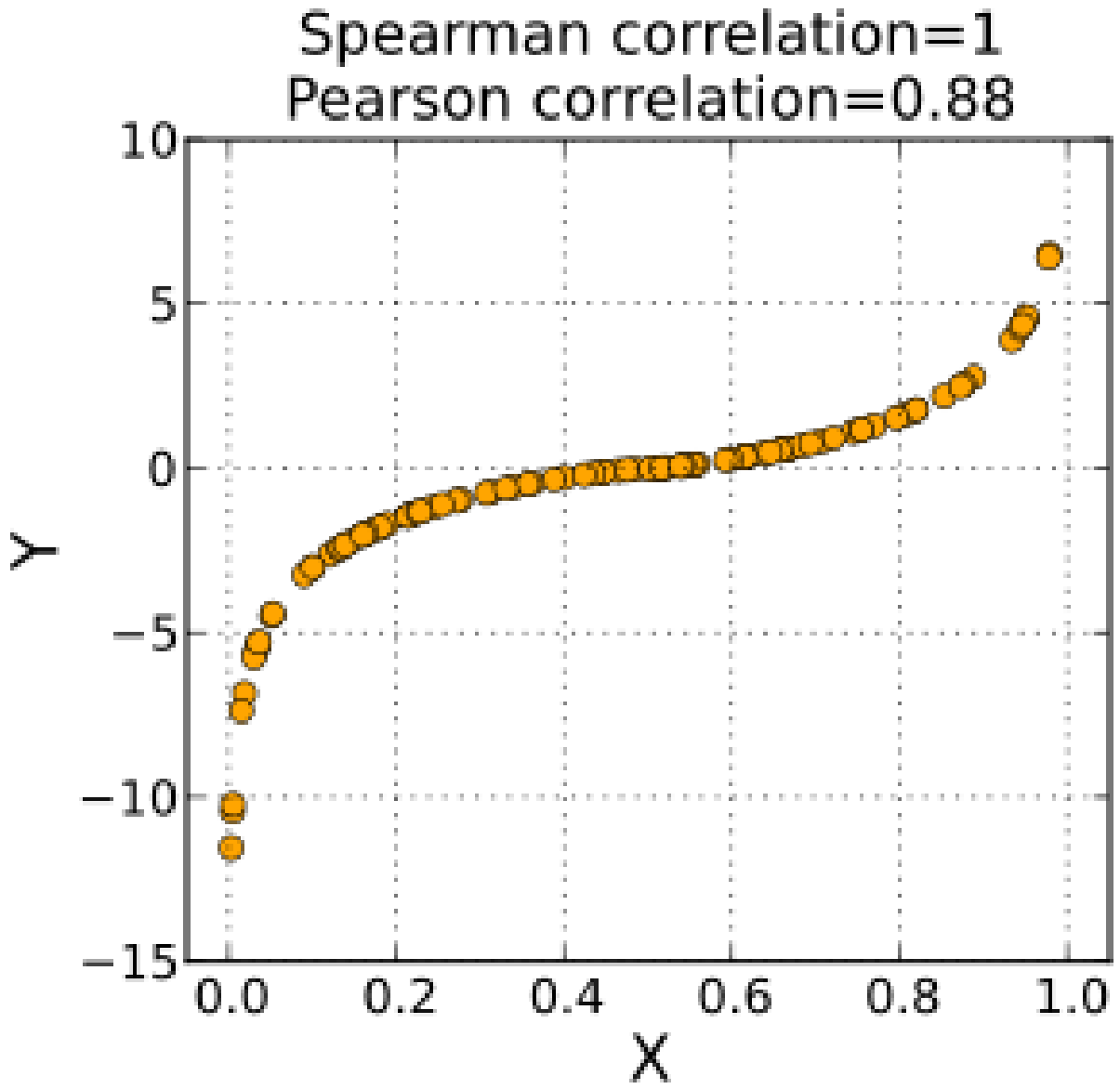


The Spearman Rank Correlation

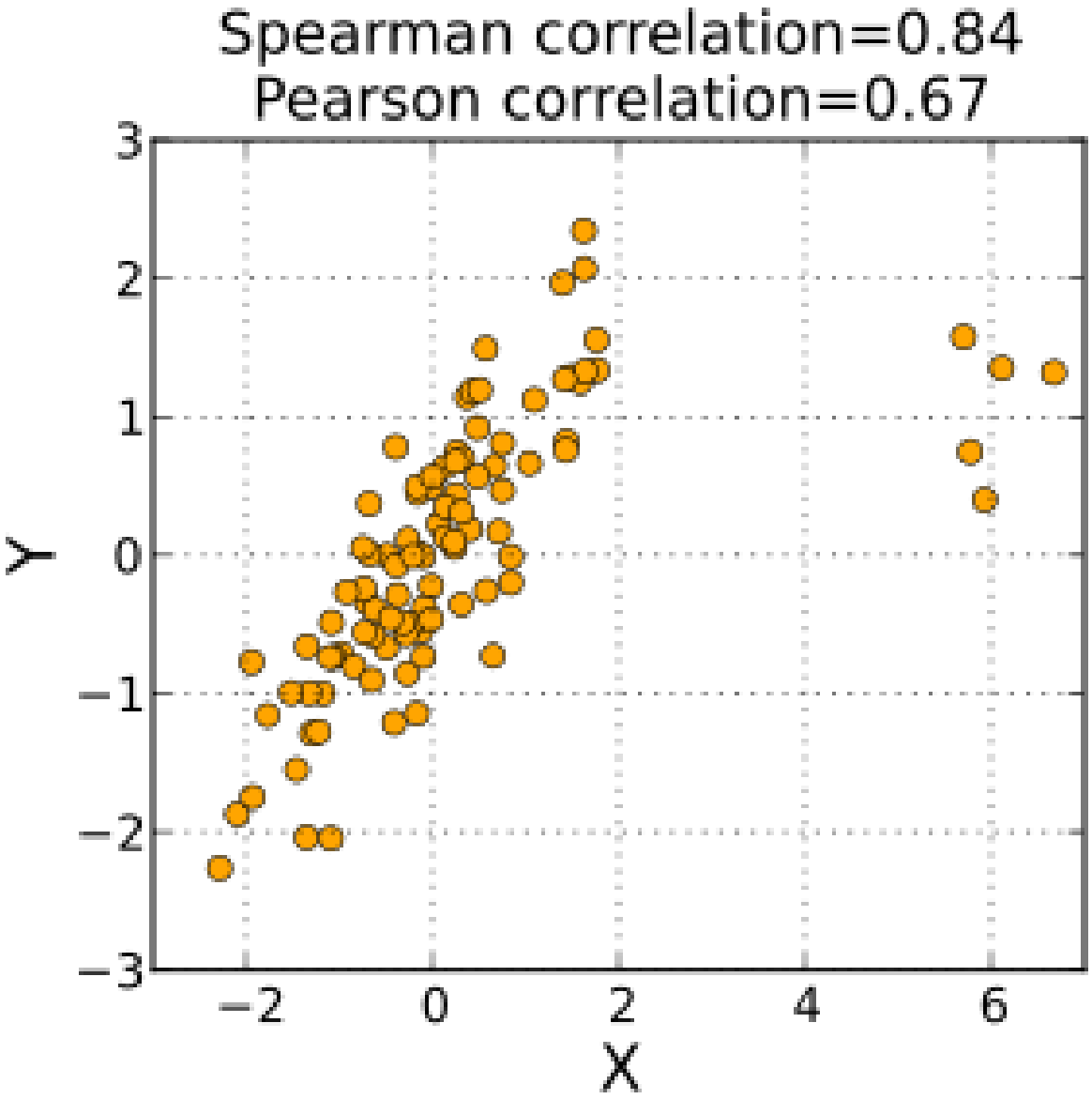
The Spearman rank correlation coefficient assesses how well the association between X and Y can be described using a **monotone function** even if that relationship is not linear.

- A monotone function preserves order - that is, Y must either be strictly increasing as X increases, or strictly decreasing as X increases.
- A Spearman correlation of 1.0 indicates simply that as X increases, Y always increases.
- Like the Pearson correlation, the Spearman correlation is dimension-free, and falls between -1 and $+1$.
- A positive Spearman correlation corresponds to an increasing (but not necessarily linear) association between X and Y, while a negative Spearman correlation corresponds to a decreasing (but again not necessarily linear) association.

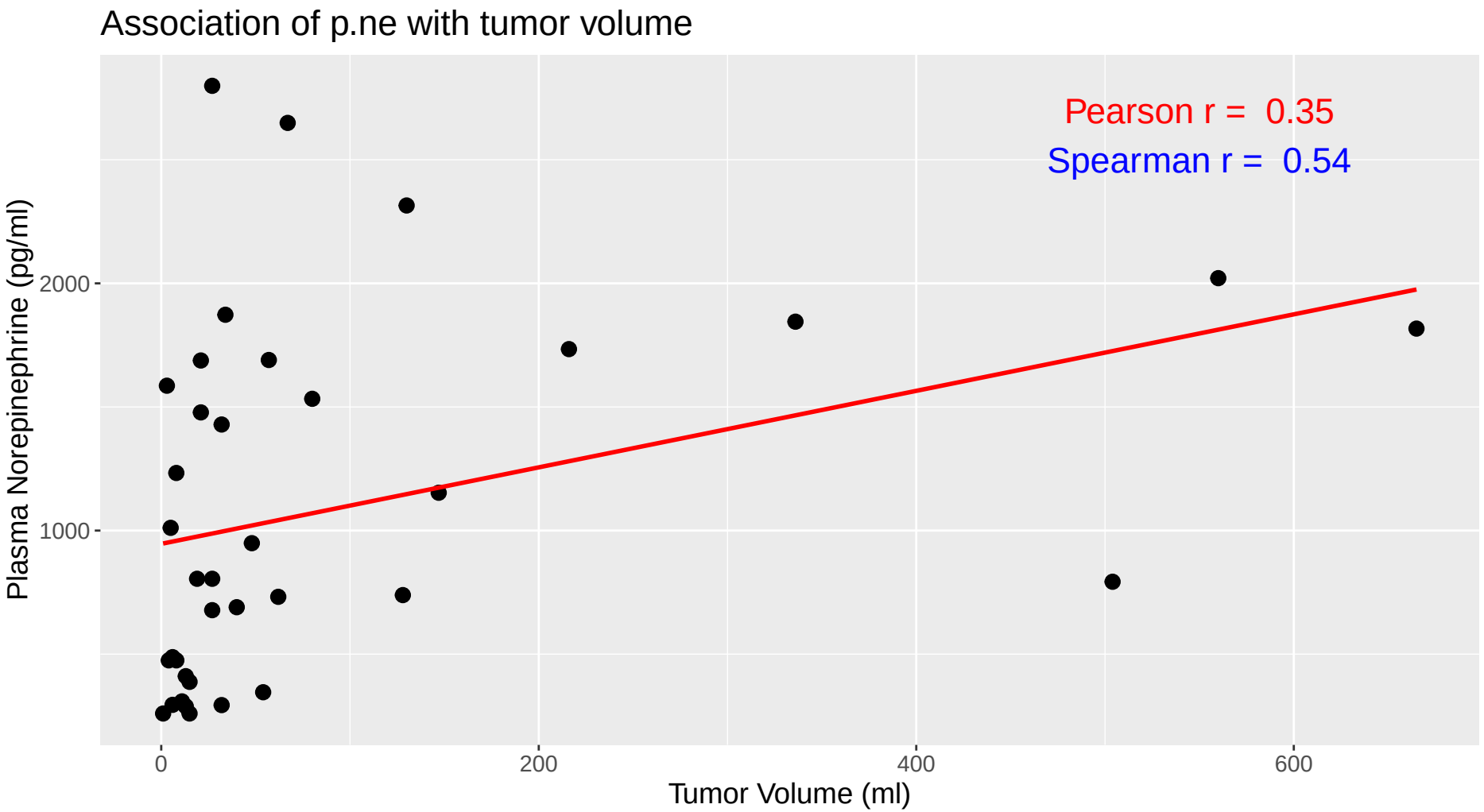
Monotone Association (Source: Wikipedia)



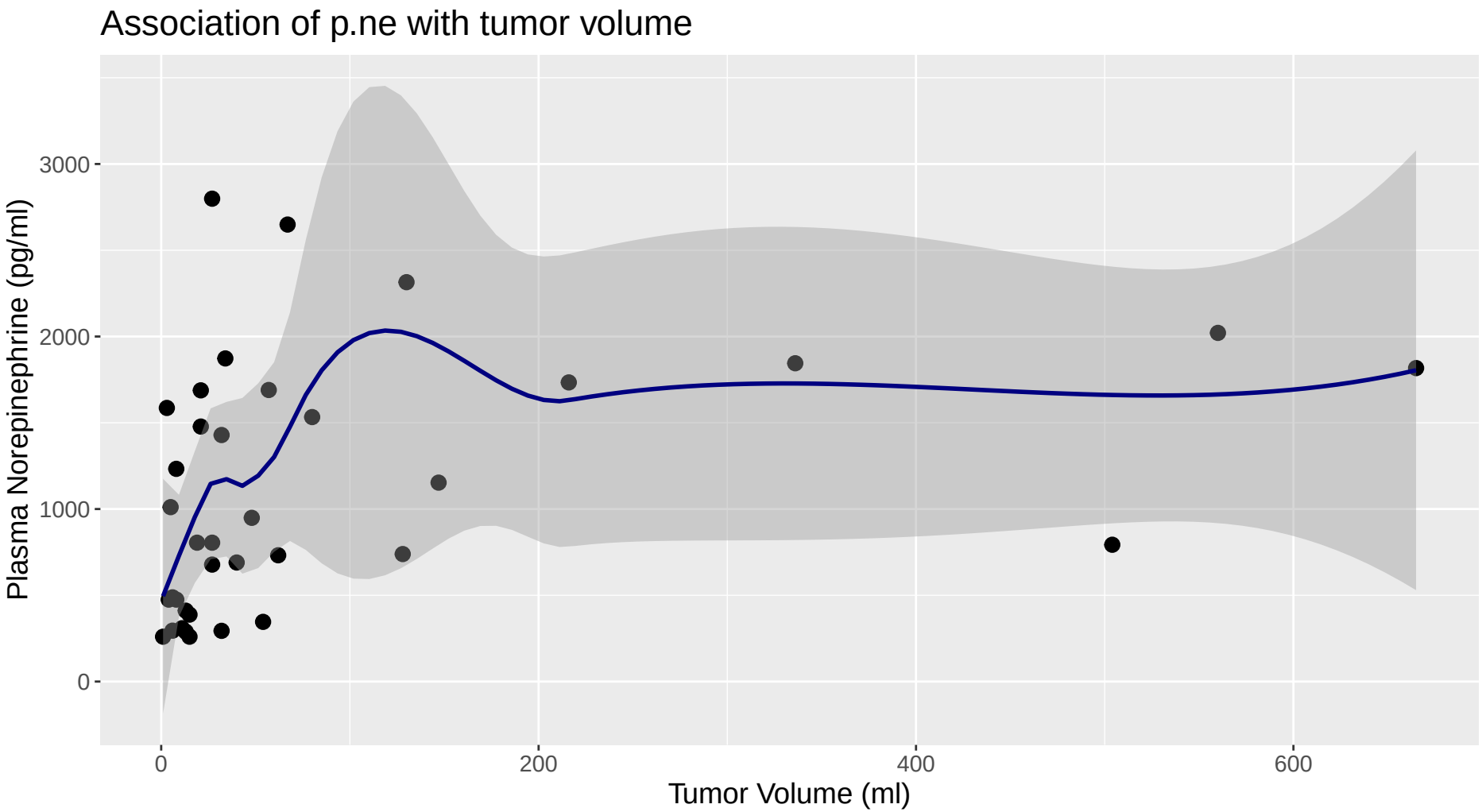
Spearman correlation reacts less to outliers



Our Key Scatterplot again

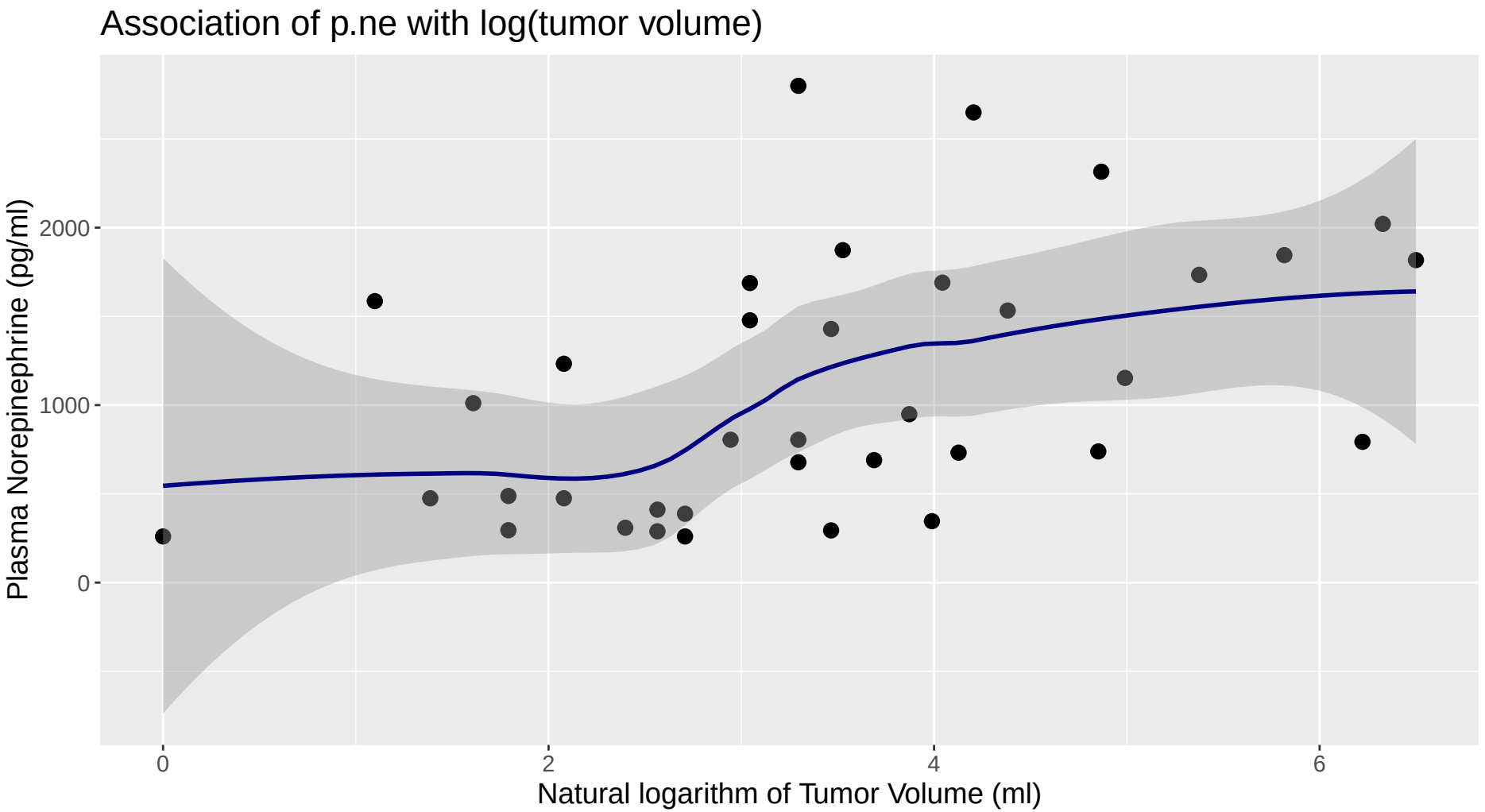


Smoothing using loess, instead

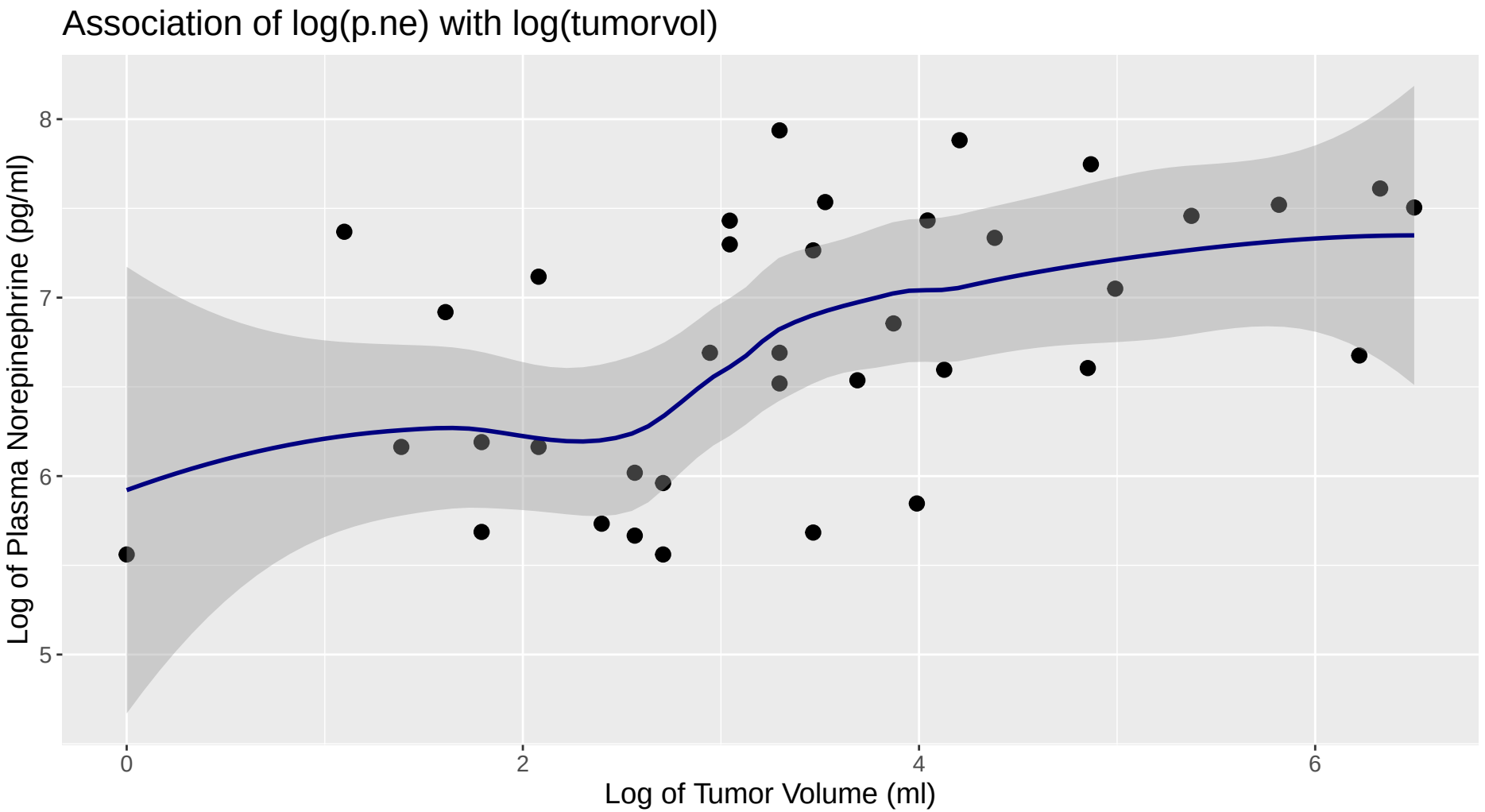


Can we transform X or Y to get to something more linear?

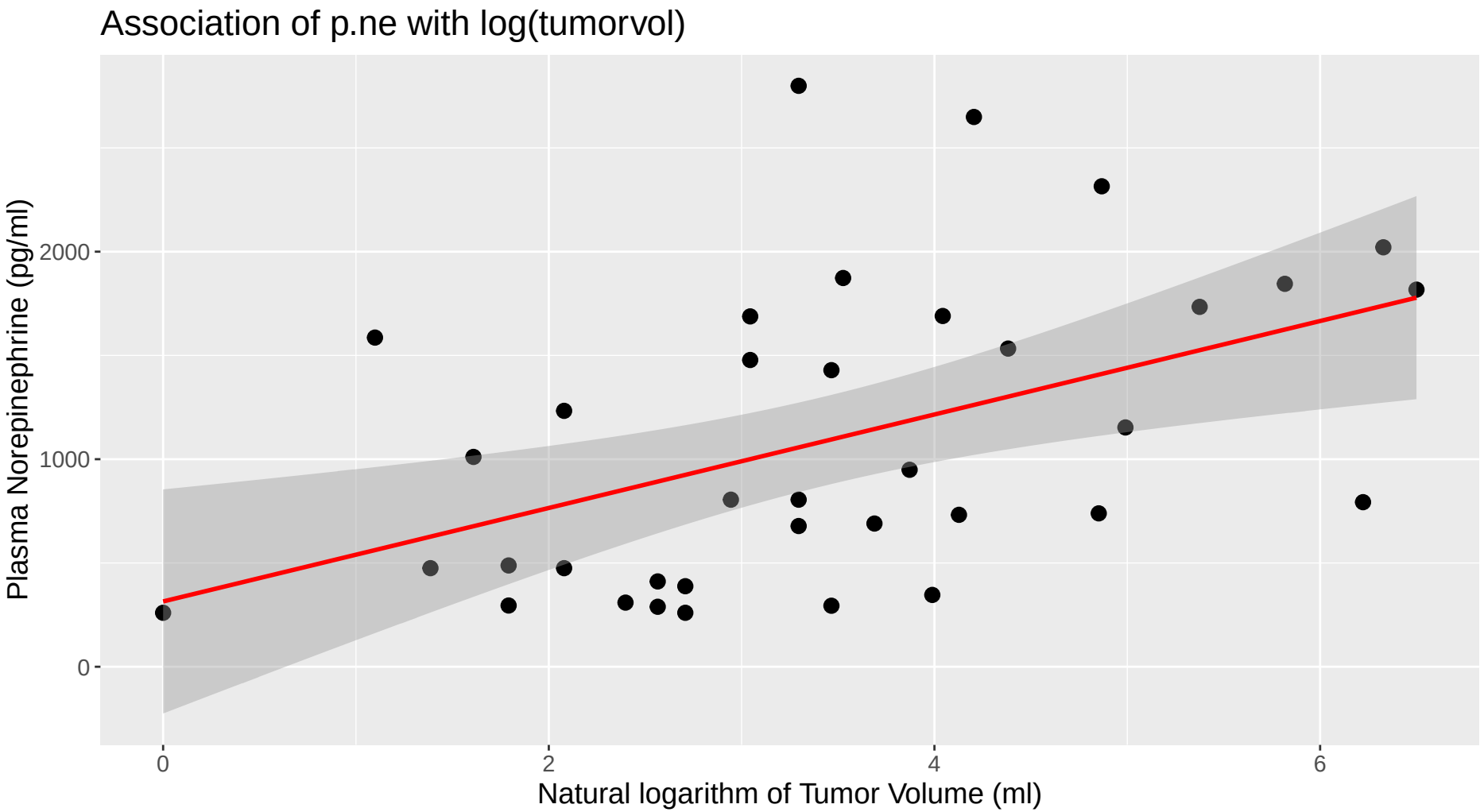
Using the Log transform to spread out the Volumes



Does a Log-Log model seem like a good choice?



Linear Model for p.ne using log(tumor volume)



Fitting that model (p.ne using log(tumorvol))

```
m1log <- lm(p.ne ~ log(tumorvol), data = VHL)

tidy(m1log, conf.int = TRUE, conf.level = 0.90) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  knitr::kable(digits = 2)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	314.6	265.95	-134.74	763.93
log(tumorvol)	225.2	70.85	105.49	344.92

Glancing at the model fit

```
m1log <- lm(p.ne ~ log(tumorvol), data = VHL)

glance(m1log) %>%
  select(r.squared, adj.r.squared, sigma) %>%
  knitr::kable(digits = 3)
```

r.squared	adj.r.squared	sigma
0.224	0.202	643.454

Summarizing the model's fit

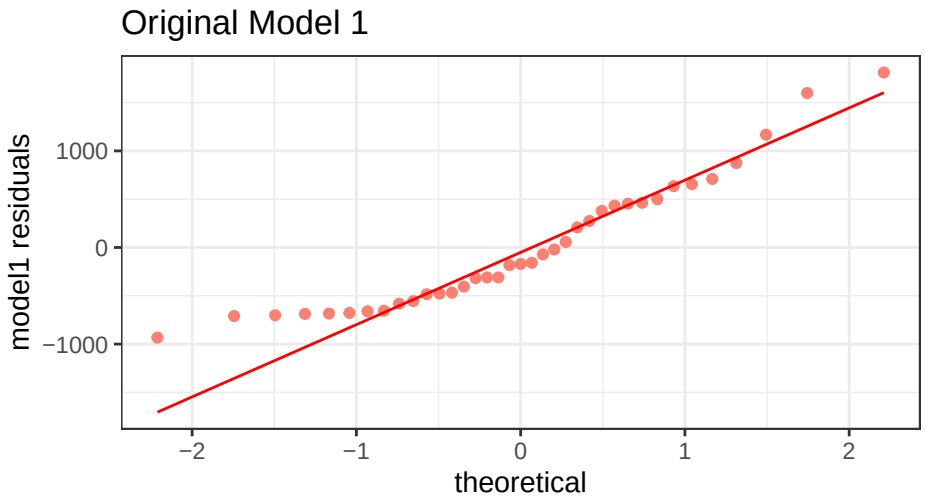
```
> summary(m1log)

Call:
lm(formula = p.ne ~ log(tumorvol), data = VHL)

Residuals:
   Min     1Q Median     3Q    Max
-922.9 -481.2 -172.7  333.9 1742.2

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     314.60     265.95   1.183  0.24481
log(tumorvol)   225.20      70.85   3.178  0.00309 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 643.5 on 35 degrees of freedom
Multiple R-squared:  0.224,     Adjusted R-squared:  0.2018
F-statistic:  10.1 on 1 and 35 DF,  p-value: 0.003092
```

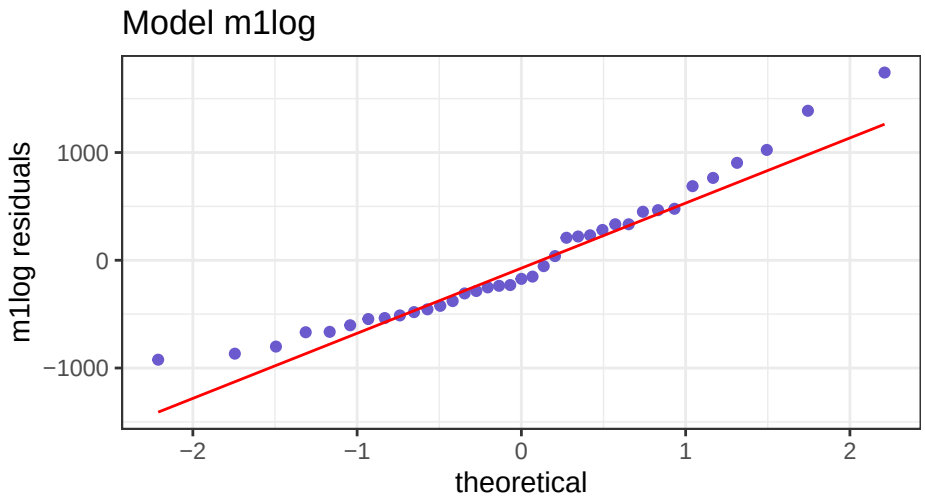
Residuals from m1log

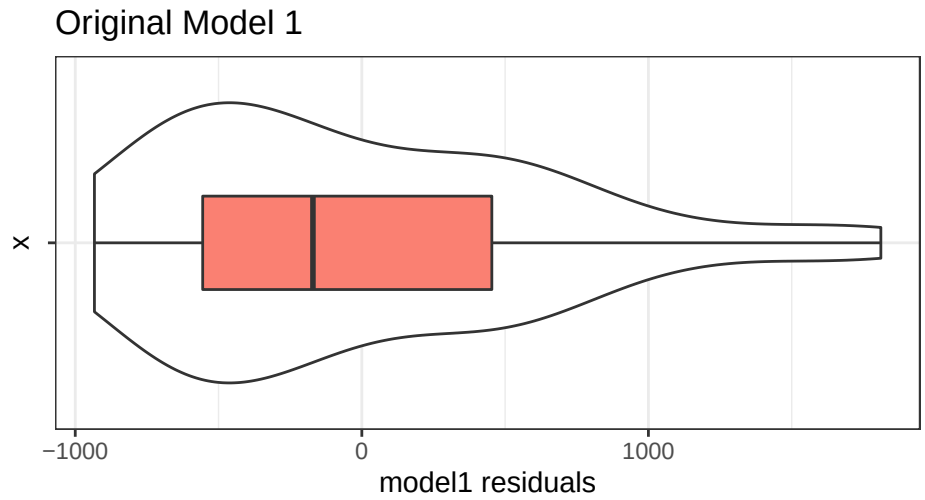
```
m1log_aug <- augment(m1log)

head(m1log_aug,3)
```

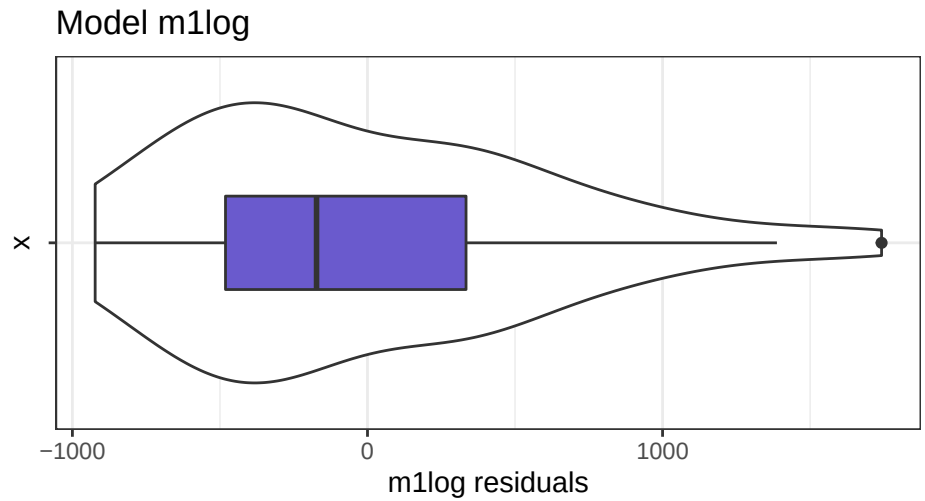
```
# A tibble: 3 x 9
   p.ne log.tumorvol. .fitted .se.fit .resid   .hat
  <dbl>         <dbl>   <dbl>   <dbl>  <dbl>  <dbl>
1   289          2.56    892.    123.  -603. 0.0364
2   294          3.47   1095.    106.  -801. 0.0270
3  2799          3.30   1057.    106.  1742. 0.0273
# ... with 3 more variables: .sigma <dbl>,
#   .cooksd <dbl>, .std.resid <dbl>
```

m1log residuals: Normally distributed?

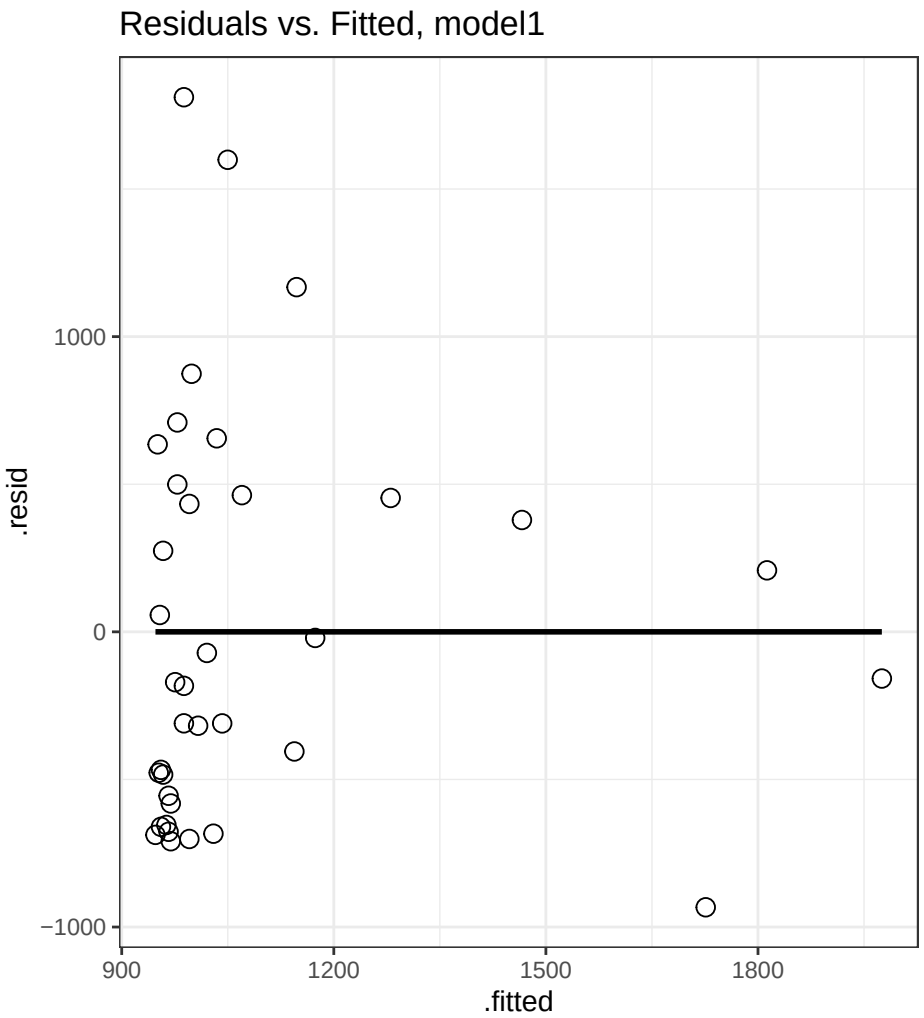


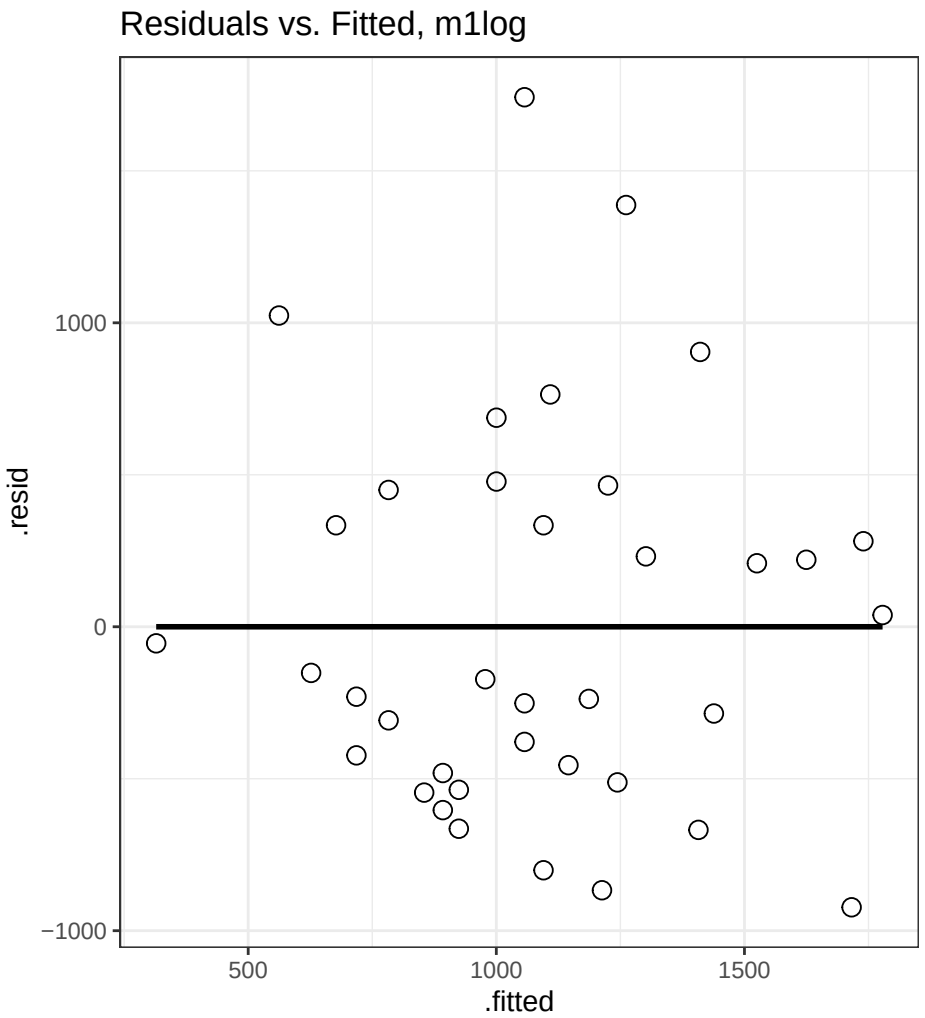






Residuals vs. Fitted plots (model1 and m1log)





Adding diagnosis to our model

Creating a Factor to represent disease category

We want to add a new variable, specifically a factor, called `diagnosis`, which will take the values `von H-L` or `neoplasia`.

- Recall `disease` is a numeric 1/0 variable (0 = von H-L, 1 = neoplasia)
- Use `fct_recode` from the `forcats` package. . .

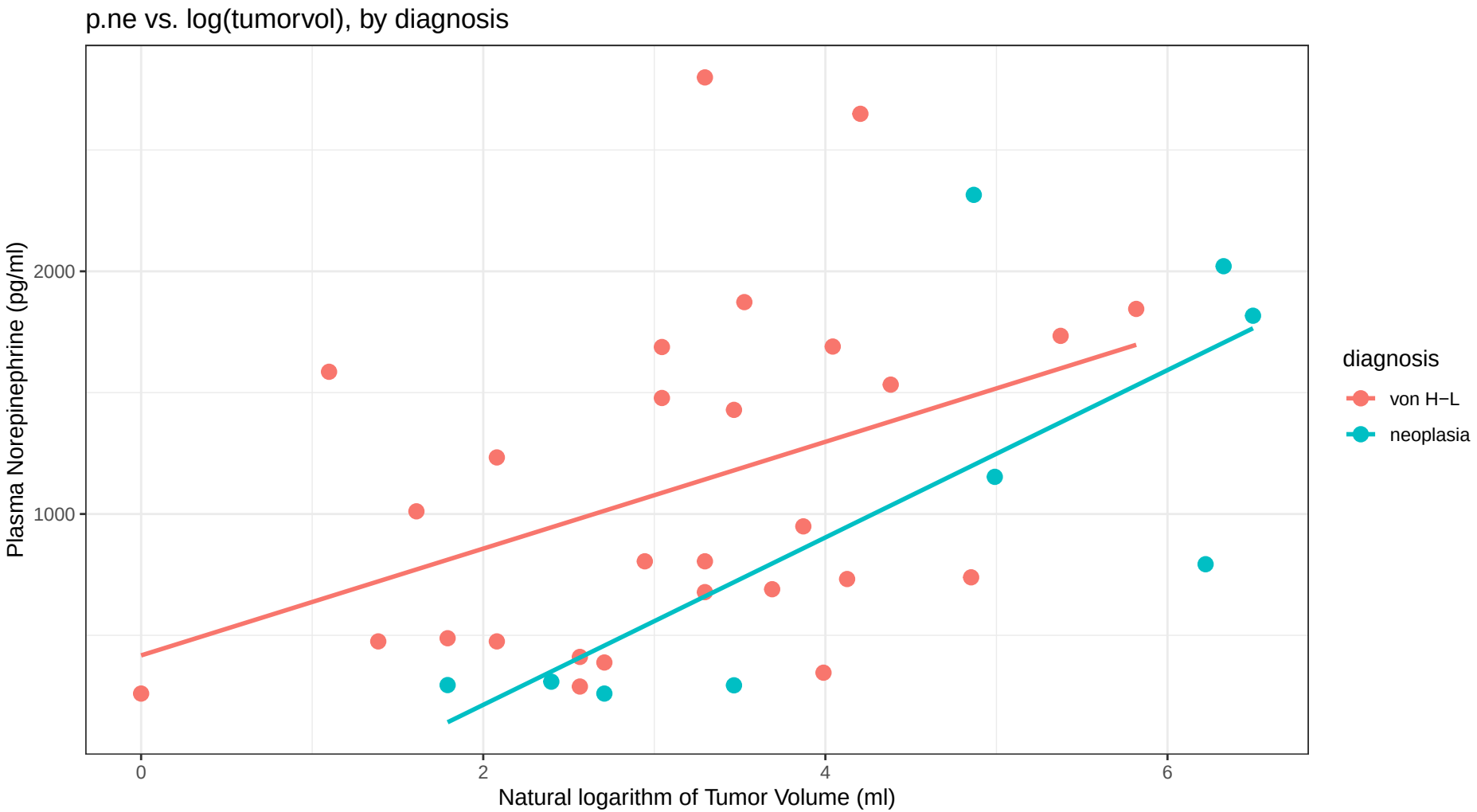
```
VHL <- VHL %>%
  mutate(diagnosis =
           fct_recode(factor(disease),
                      "neoplasia" = "1",
                      "von H-L" = "0")
  )
```

Now, what does VHL look like?

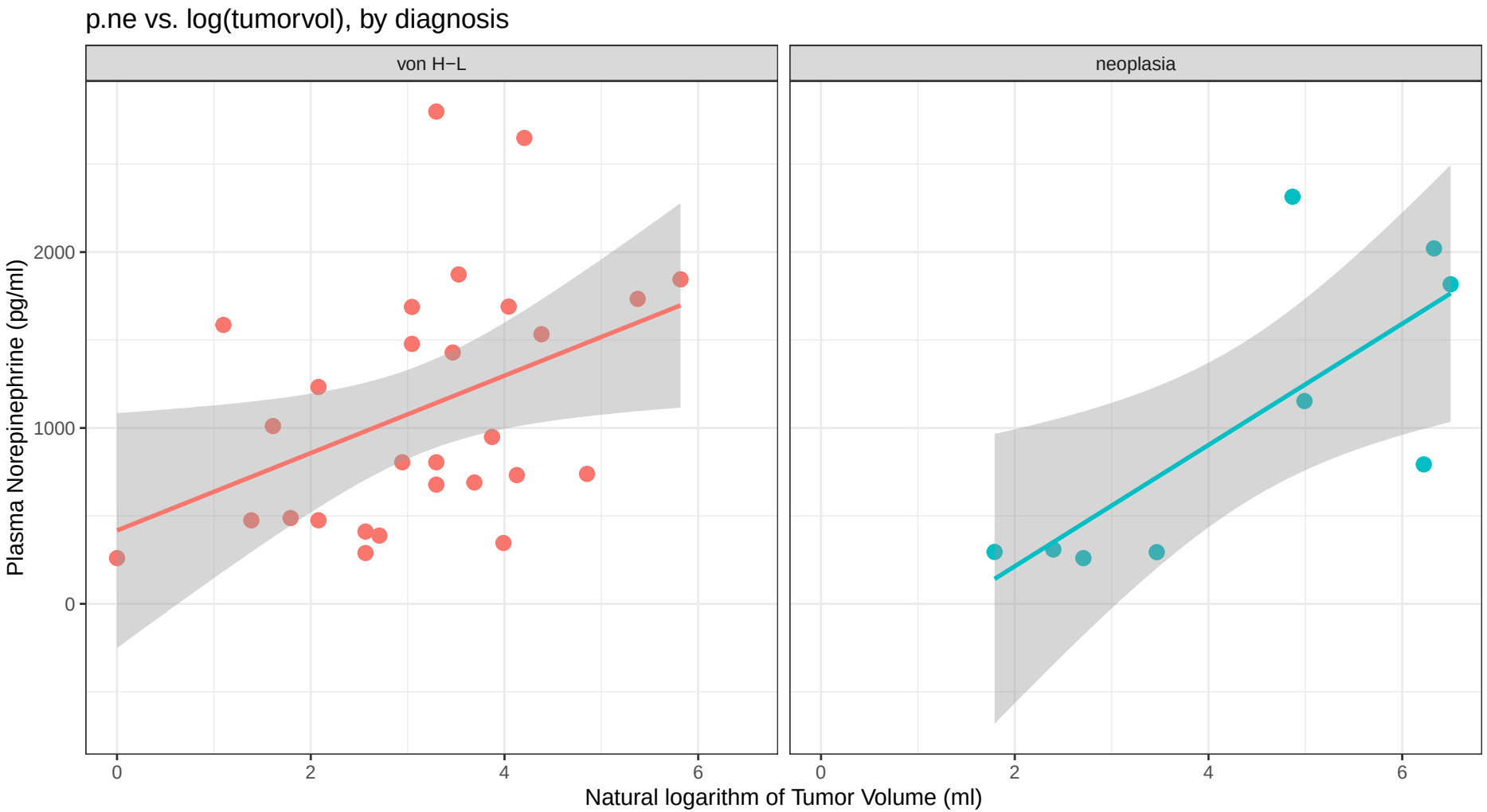
```
VHL
```

```
# A tibble: 37 x 5
      id disease  p.ne tumorvol diagnosis
   <dbl>   <dbl> <dbl>    <dbl> <fct>
 1   101       0   289       13 von H-L
 2   102       1   294       32 neoplasia
 3   103       0  2799       27 von H-L
 4   104       0  2649       67 von H-L
 5   105       0   346       54 von H-L
 6   106       0  1690       57 von H-L
 7   107       0   805       19 von H-L
 8   108       1  1153      147 neoplasia
 9   109       0   678       27 von H-L
10   110       1  1817      665 neoplasia
# ... with 27 more rows
```

Compare the patients by diagnosis



Faceted Scatterplots by diagnosis



Separate Models by Diagnosis?

```
model2_vhl <- lm(p.ne ~ log(tumorvol),
                 data = filter(VHL, diagnosis == "von H-L"))

coef(model2_vhl)
```

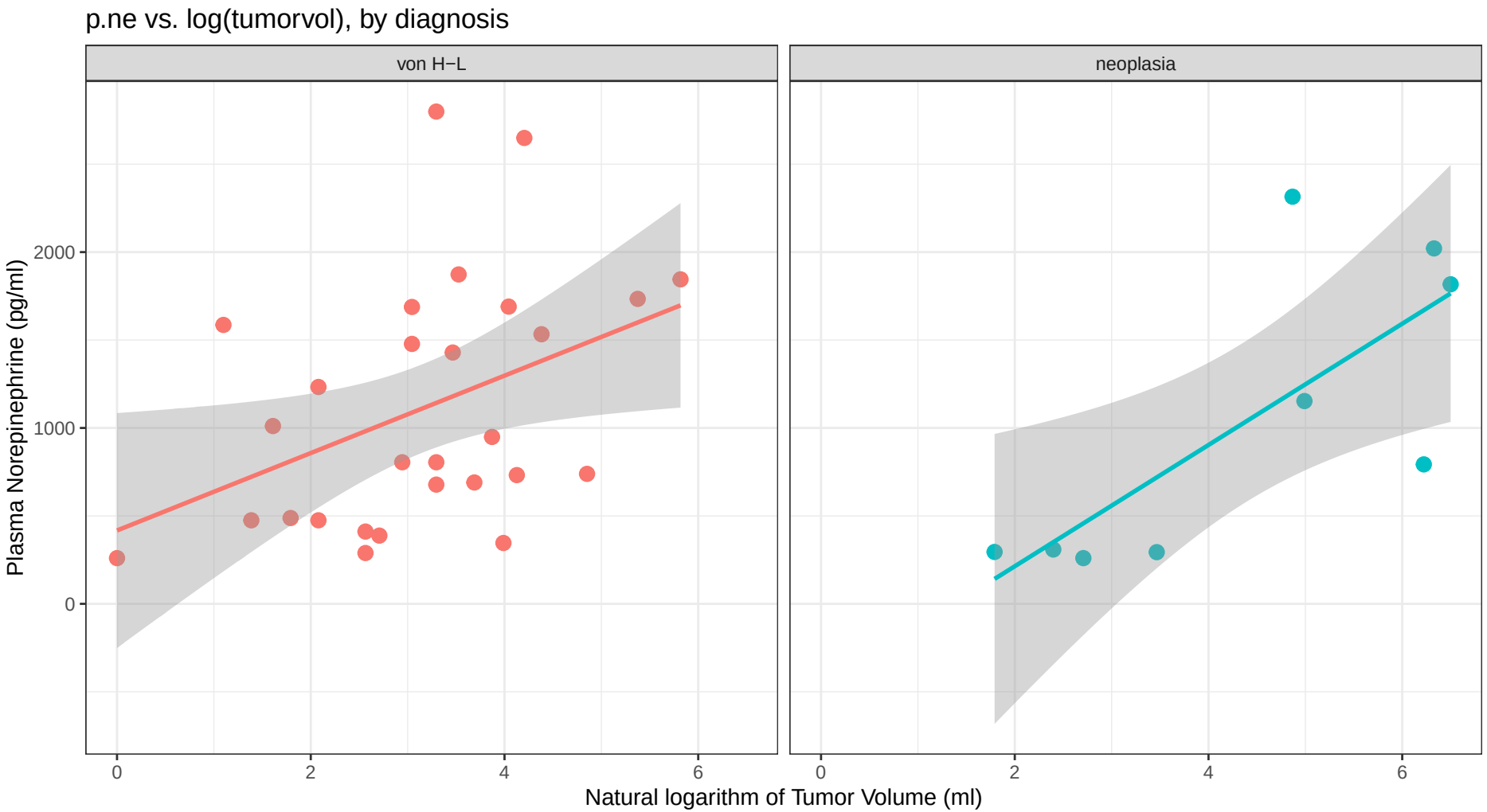
```
  (Intercept) log(tumorvol)
     417.2040      220.0463
```

```
model2_neo <- lm(p.ne ~ log(tumorvol),
                 data = filter(VHL, diagnosis == "neoplasia"))

coef(model2_neo)
```

```
  (Intercept) log(tumorvol)
    -476.0978      344.8253
```

Does this match our plot?

Faceted Scatterplots by diagnosis, again



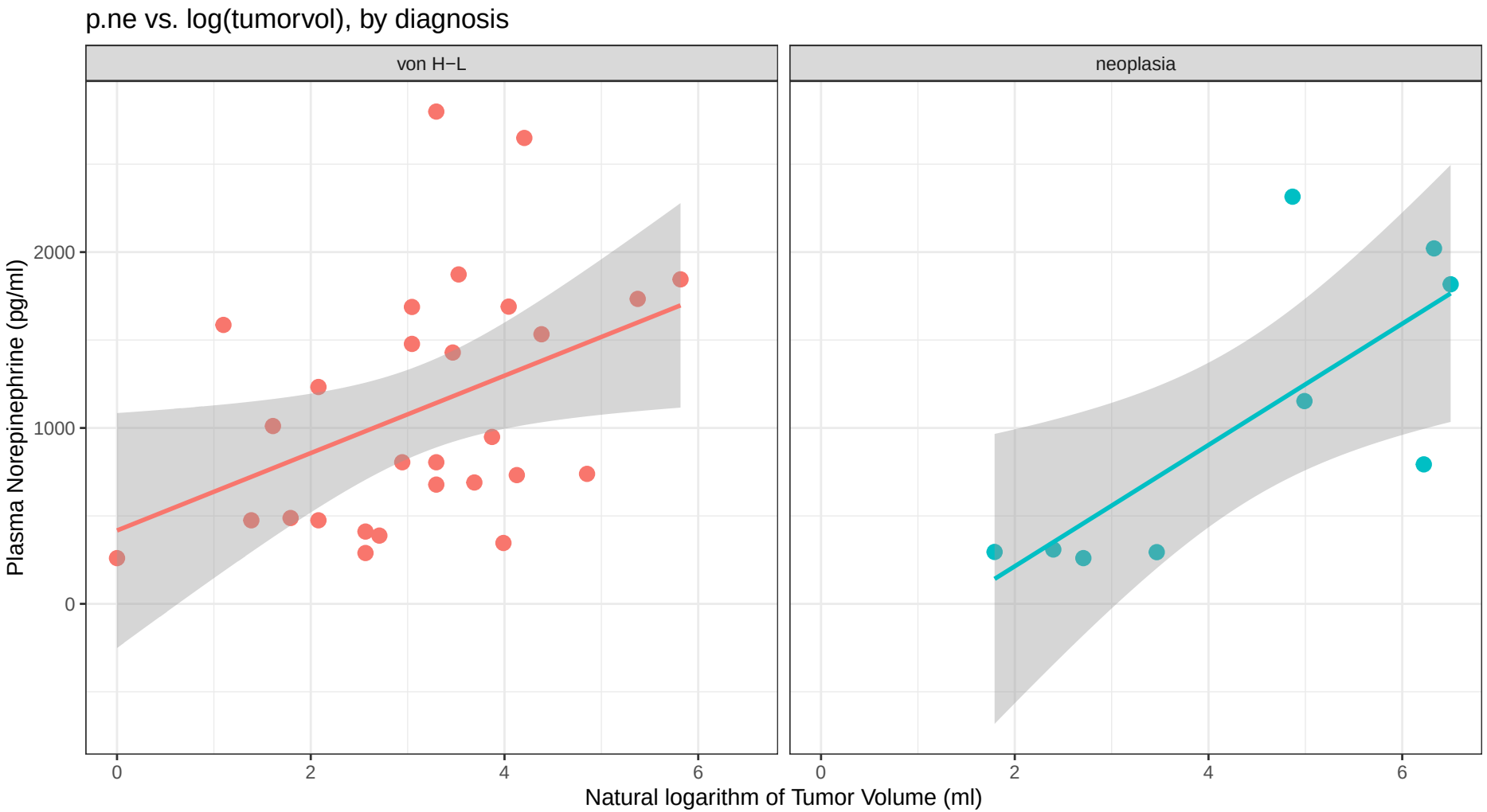
Correlation Coefficients

```
VHL %>%
  group_by(diagnosis) %>%
  summarize(Correlation = cor(log(tumorvol), p.ne),
            Rsquare = (cor(log(tumorvol), p.ne)^2) )
```

```
# A tibble: 2 x 3
  diagnosis Correlation Rsquare
  <fct>           <dbl>   <dbl>
1 von H-L         0.412   0.169
2 neoplasia       0.756   0.572
```

Does this match our plot?

Faceted Scatterplots by diagnosis, one more time



What do we predict if log(tumorvol) = 3?

`log(tumorvol)` = 3 implies `tumorvol` = `exp(3)` = 20.0855369 ml.

From our `model2_vhl`, we'd predict:

- 417 + 220 (3) = 1,077 pg/nl of `p.ne` for a VHL patient with `tumorvol` = 20.0855369 ml.

From our `model2_neo`, we'd predict:

- -476 + 345 (3) = 559 pg/nl of `p.ne` for a Neoplasia patient with `tumorvol` = 20.0855369 ml.

Model including two predictors

```
model3 <- lm(p.ne ~ log(tumorvol) + diagnosis, data = VHL)
model3
```

```
Call:
lm(formula = p.ne ~ log(tumorvol) + diagnosis, data = VHL)

Coefficients:
       (Intercept)       log(tumorvol)
             273.2               265.8
diagnosisneoplasia
            -404.4
```

But this model only changes the intercept?

```
coef(model3)
```

```
       (Intercept)      log(tumorvol)
          273.1745           265.7977
diagnosisneoplasia
          -404.4333
```

- Model for VHL is `p.ne` = 273 + 266 `log(tumorvol)`.
  - `p.ne` prediction if `log(tumorvol)` = 3 is 1,071 pg/nl.
- Model for neoplasia is `p.ne` = (273 - 404) + 266 `log(tumorvol)`, or -131 + 266 `log(tumorvol)`.
  - `p.ne` prediction if `log(tumorvol)` = 3 is 667 pg/nl.

Is that what we want?

Model accounting for different slopes *and* intercepts

```
model4 <- lm(p.ne ~ log(tumorvol) * diagnosis, data = VHL)
model4
```

```
Call:
lm(formula = p.ne ~ log(tumorvol) * diagnosis, data = VHL)

Coefficients:
                      (Intercept)
                            417.2
                    log(tumorvol)
                            220.0
               diagnosisneoplasia
                           -893.3
log(tumorvol):diagnosisneoplasia
                            124.8
```

model4 results

p.ne = 417 + 220 log(tumorvol) - 893 (diagnosis = neoplasia) + 125 (diagnosis = neoplasia)*log(tumorvol)

where the indicator variable (diagnosis = neoplasia) = 1 for neoplasia subjects, and 0 for other subjects. . .

- Model for p.ne in von H-L patients:
  - 417 + 220 log(tumorvol)
- Model for p.ne in neoplasia patients:
  - (417 - 893) + (220 + 125) log(tumorvol)
  - -476 + 345 log(tumorvol)

These are our initial (separated) models, in this case.

model4 Predictions

What is the predicted p.ne for a single new subject with tumorvol = 200 ml (so log(tumorvol) = 5.3) in each diagnosis category?

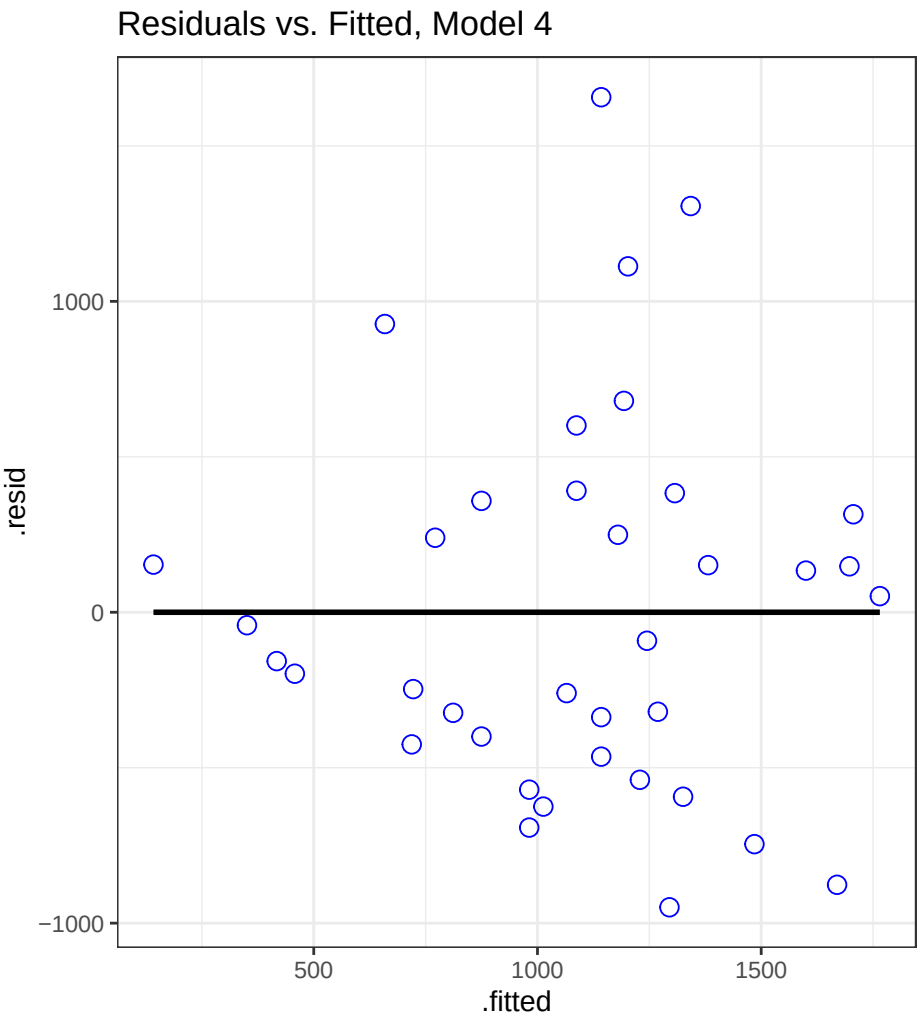
```
predict(model4, newdata = tibble(tumorvol = 200,
        diagnosis = "neoplasia"), interval = "prediction")
```

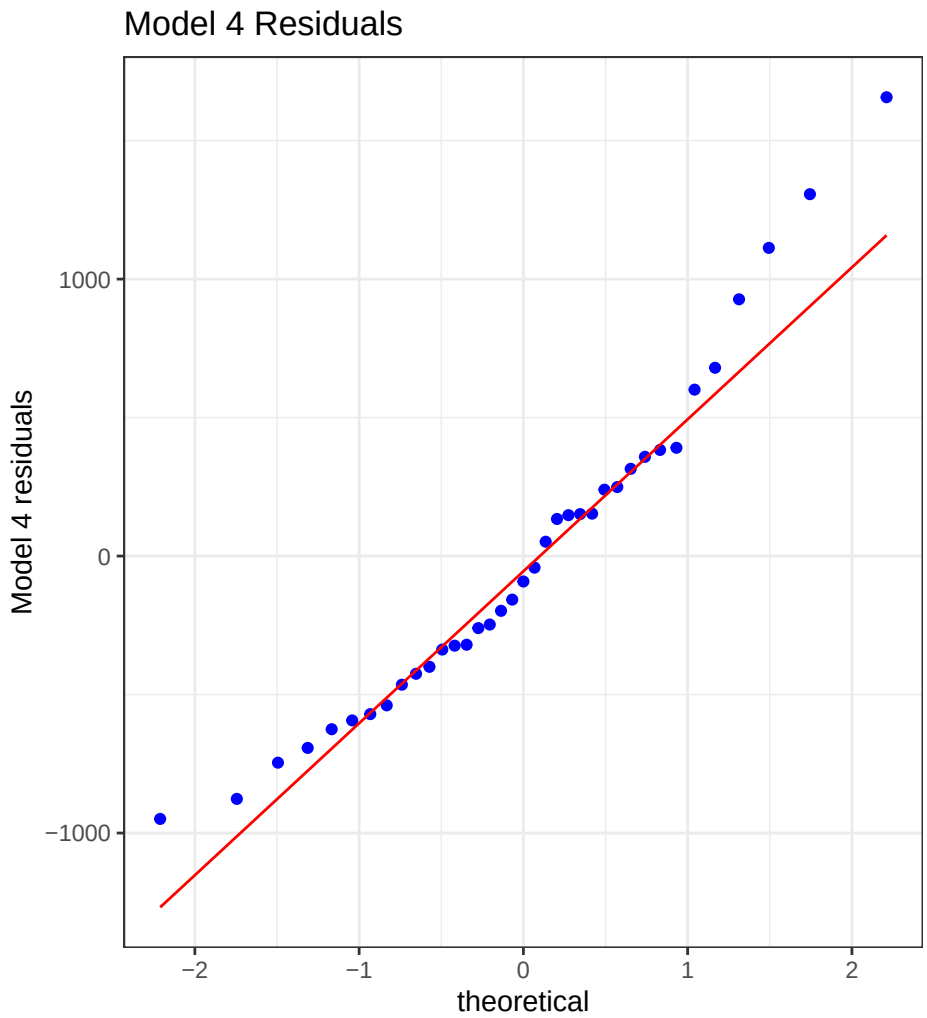
```
       fit      lwr     upr
1 1350.896 -28.0571 2729.85
```

```
predict(model4, newdata = tibble(tumorvol = 200,
        diagnosis = "von H-L"), interval = "prediction")
```

```
       fit      lwr      upr
1 1583.079 208.6489 2957.509
```

How about the Residuals of model4?





Tidying the model4 coefficients, with broom

```
tidy(model4, conf.int = TRUE, conf.level = 0.90) %>%
  select(term, estimate, conf.low, conf.high) %>%
  knitr::kable(digits = 1)
```

term	estimate	conf.low	conf.high
(Intercept)	417.2	-120.9	955.4
log(tumorvol)	220.0	61.7	378.4
diagnosisneoplasia	-893.3	-2007.8	221.2
log(tumorvol):diagnosisneoplasia	124.8	-136.7	386.3

model4, summarized at a glance, with broom

```
glance(model4) %>% select(r.squared, sigma, AIC)
```

```
# A tibble: 1 x 3
  r.squared sigma   AIC
      <dbl> <dbl> <dbl>
1     0.290  634.  588.
```

Compare this to m1log...

```
glance(m1log) %>% select(r.squared, sigma, AIC)
```

```
# A tibble: 1 x 3
  r.squared sigma   AIC
      <dbl> <dbl> <dbl>
1     0.224  643.  587.
```

Conclusions about VHL data

- Model 4, accounting for the interaction of diagnosis with the log of tumor volume, was able to account for about 29% of the variation in the plasma norepinephrine levels.
- m1log, which didn't include diagnosis but just the log of tumor volume, accounts for about 22% of the variation in plasma norepinephrine levels.
- Model 1, our original linear model, which didn't account for diagnosis and didn't fit assumptions well (using raw tumor volume) accounted for about 12% of the variation in plasma norepinephrine levels.

Can we draw a lot more from this yet?

So what did we hear about today?

- The central role of linear regression in understanding associations between quantitative variables.
- The interpretation of a regression model as a prediction model.
- Assessment of key regression summaries, including residuals.
- Using `tidy`, `glance` and `augment` from `broom` to summarize the model.
- Measuring association through correlation coefficients.
- How we might think about "adjusting" for the effect of a categorical predictor on a relationship between two quantitative ones.
- How a transformation might help us "linearize" the relationship shown in a scatterplot.