STAC33 A8

Zhan Li

Contents

```
library(tidyverse)
library(broom)
```

(a)

```
url <- "http://ritsokiguess.site/STAC32/heightfoot.csv"
data <- read_csv(url)
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   height = col_double(),
##   foot = col_double()
## )
```

```
data
```

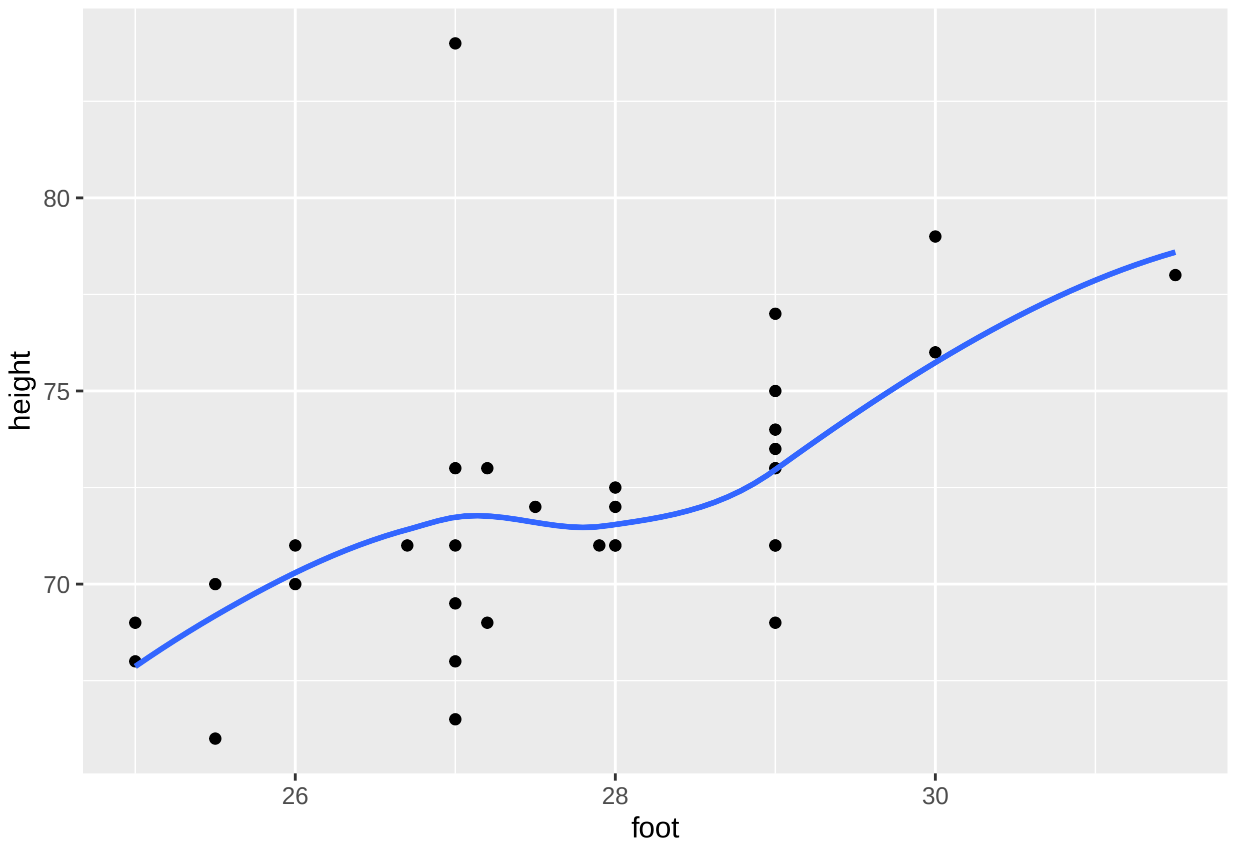
```
## # A tibble: 33 x 2
##    height  foot
##     <dbl> <dbl>
##  1   66.5  27
##  2   73.5  29
##  3   70    25.5
##  4   71    27.9
##  5   73    27
##  6   71    26
##  7   71    29
##  8   69.5  27
##  9   73    29
## 10   71    27
## # ... with 23 more rows
```

- Since the file format ends in "csv", therefore use "read_csv".

(b)

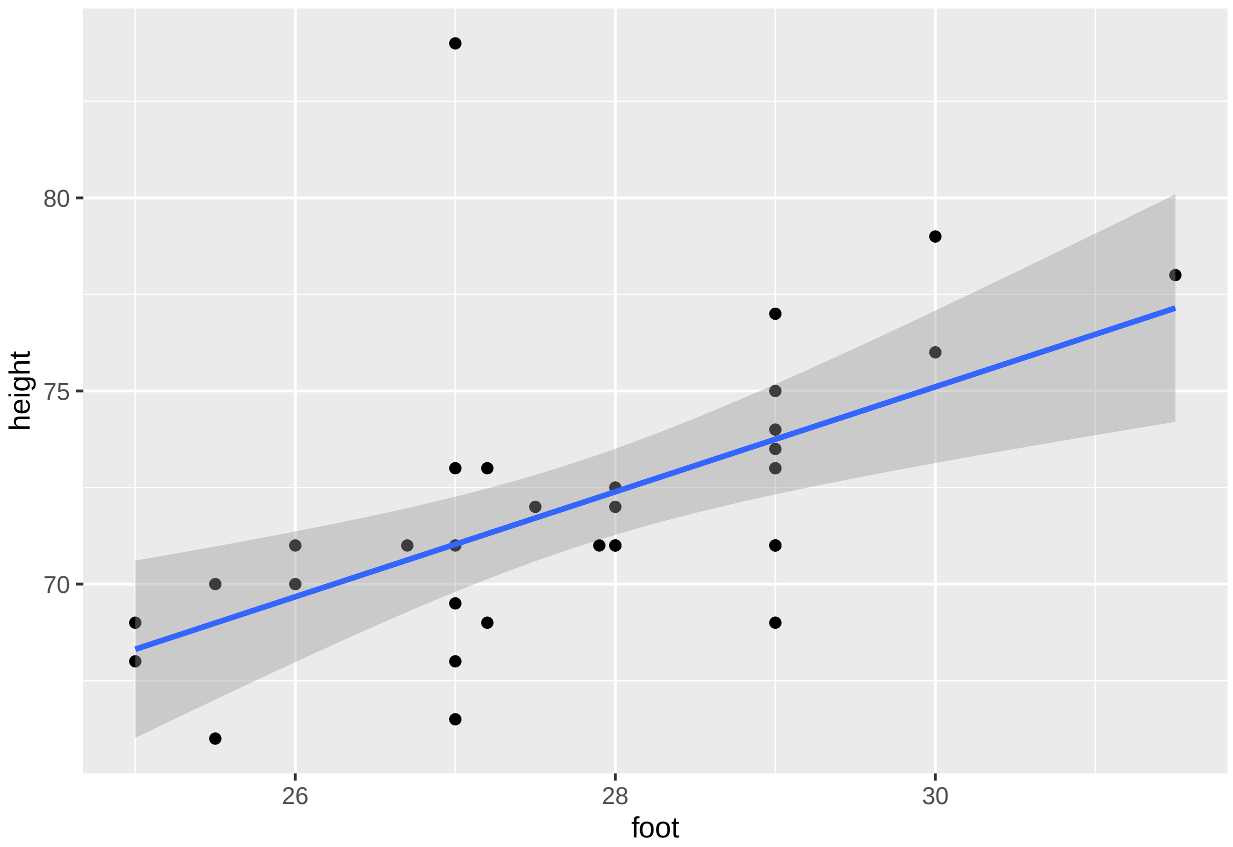
```
ggplot(data, aes(x = foot, y = height)) + geom_point() + geom_smooth(se = F)
```

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



```
ggplot(data, aes(x = foot, y = height)) + geom_point() + geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



- Since we have two quantitative variables, we should use scatterplot.
- Foot is the x variable and height is the y variable, because we are interested in how much does foot length affect height. So height is our response variable and it should be the y variable.
- We don't want to assume linear, so add "se = F" in the trend.
- But if we assume linear, the trend does not deviate too much from the first plot, So linear is ok.
- most of the points are relatively close to the trend.
- There is a relationship, as foot gets higher, height also generally gets higher.

(c)

- There seems to be a outlier when foot is 27 and height is around 84, because it's very far away from the trend, and it's a lot higher than the other heights—even higher than the height of the largest foot.
- To me, the rest of the points are relatively close to the trend.

(d)

```r
g <- lm(height ~ foot, data)
summary(g)
```

```
##
## Call:
## lm(formula = height ~ foot, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7491 -1.3901 -0.0310  0.8918 12.9690
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  34.3363     9.9541   3.449 0.001640 ** 
## foot          1.3591     0.3581   3.795 0.000643 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 3.102 on 31 degrees of freedom
## Multiple R-squared:  0.3173, Adjusted R-squared:  0.2952 
## F-statistic: 14.41 on 1 and 31 DF,  p-value: 0.0006428
```

```
glance(g)
```

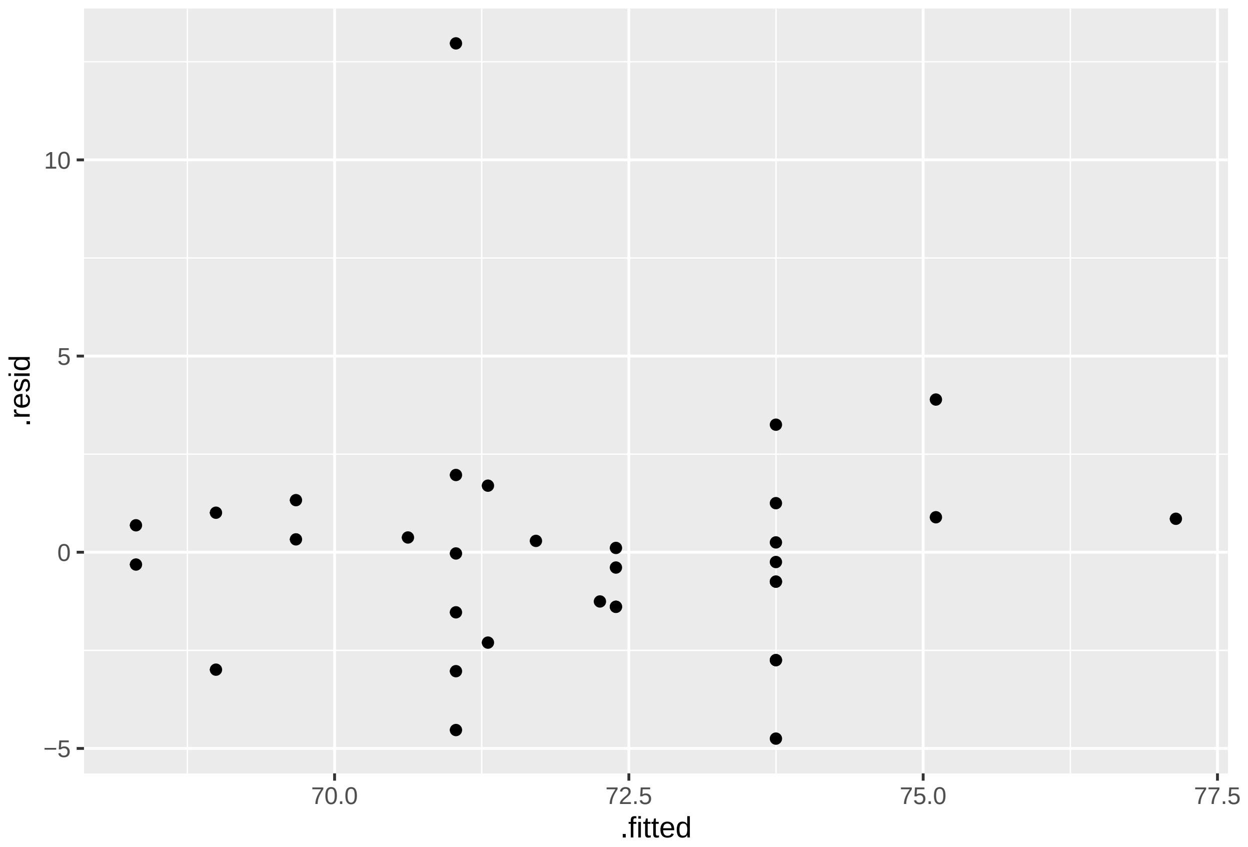
```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.317         0.295  3.10      14.4 6.43e-4     1  -83.1  172.  177.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
tidy(g)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    34.3      9.95       3.45 0.00164 
## 2 foot            1.36     0.358      3.80 0.000643
```

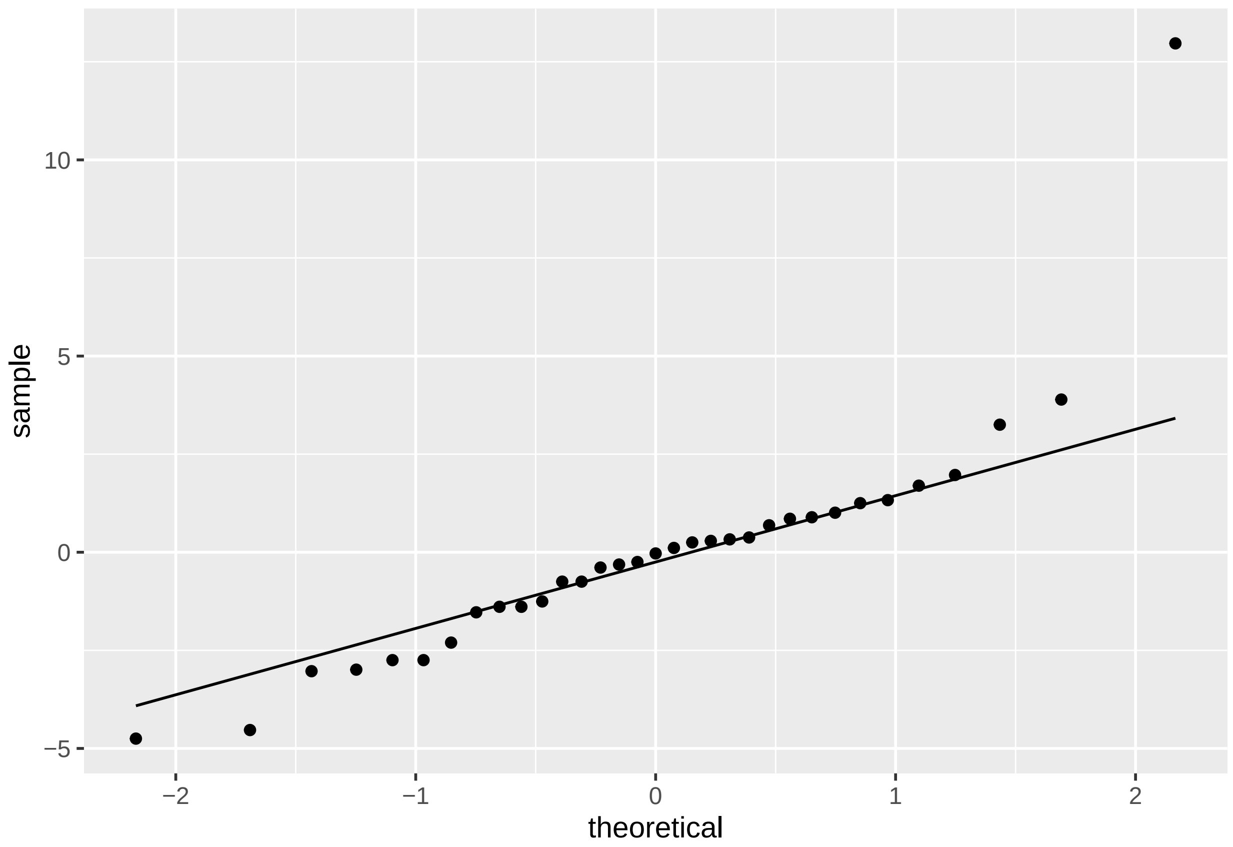
- R squared is only 0.317259, which is quite low.

```
ggplot(g, aes(y = .resid, x = .fitted)) + geom_point()
```



- Because of the unusual observation, the residual axis on the plot ranges from -5 to over 10. Otherwise, the residual plot looks random enough to me on either side of the line when residual is 0.

```
ggplot(g, aes(sample = .resid)) + stat_qq() + stat_qq_line()
```



- Once again, from the normal quantile plot, there is a an observation that deviates significantly from the line.

(e)

- On the residual plot, the observation shows up where the residual is near 12.5, which is very far from the rest of the residuals.
- On the normal quantile plot, the observation shows up where the sample quantile is near 12.5, which is again very far from the rest of the points and the line.

(f)

- Looking back at the first scatterplot, the observation occurs when the height is over 82.5, and there is only one such point.
- Therefore, use maximum to find such height, and filter it out.

```
max(data$height)
```

```
## [1] 84
```

```
new_data <- data %>% filter(!height == (max(data$height)))
new_data
```

```
## # A tibble: 32 x 2
##    height  foot
##     <dbl> <dbl>
##  1   66.5  27
##  2   73.5  29
##  3   70    25.5
##  4   71    27.9
##  5   73    27
##  6   71    26
##  7   71    29
##  8   69.5  27
##  9   73    29
## 10   71    27
## # ... with 22 more rows
```

- Now we can see there is one less row of data compared to before.

(g)

```
new_g <- lm(height ~ foot, new_data)
summary(new_g)
```

```
##
## Call:
## lm(formula = height ~ foot, data = new_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5097 -1.0158  0.4757  1.1141  3.9951
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  30.1502     6.5411   4.609 7.00e-05 ***
## foot          1.4952     0.2351   6.360 5.12e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.029 on 30 degrees of freedom
## Multiple R-squared:  0.5741, Adjusted R-squared:  0.5599
## F-statistic: 40.45 on 1 and 30 DF,  p-value: 5.124e-07
```

```
glance(new_g)
```

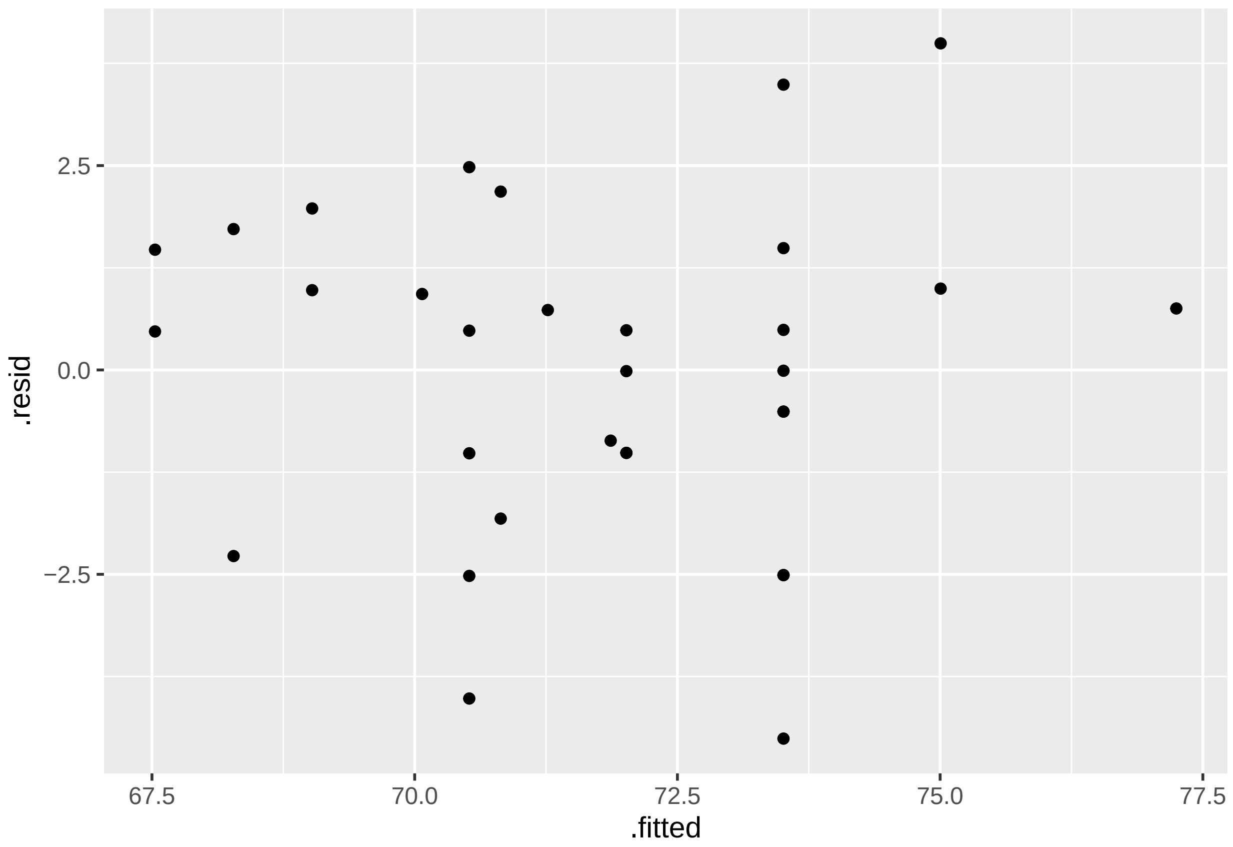
```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.574         0.560  2.03      40.4 5.12e-7     1  -67.0  140.  144.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
tidy(new_g)
```

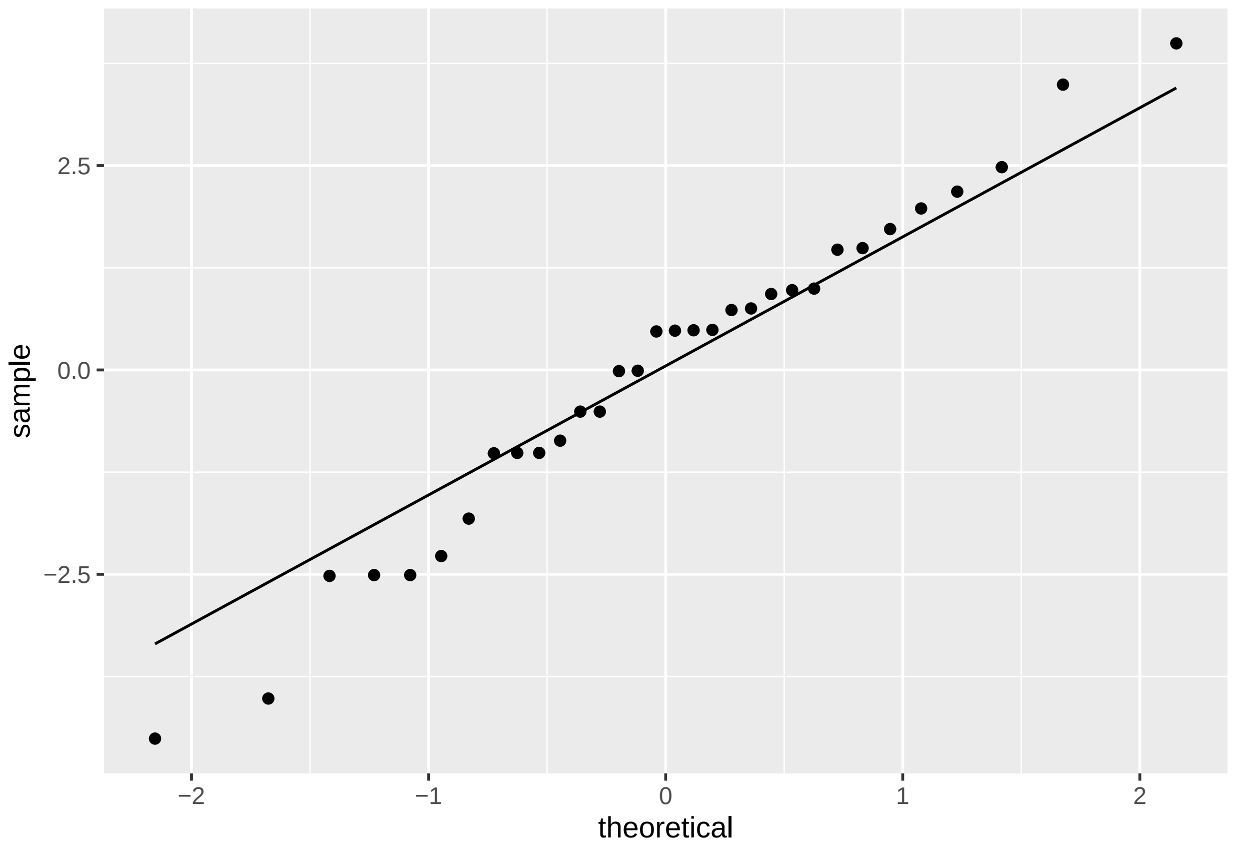
```
## # A tibble: 2 x 5
##   term        estimate std.error statistic     p.value
##   <chr>          <dbl>     <dbl>     <dbl>       <dbl>
## 1 (Intercept)    30.2      6.54       4.61 0.0000700
## 2 foot            1.50     0.235      6.36 0.000000512
```

- The corrected data now has R squared 0.5741425, which is much higher than the previous data.

```
ggplot(new_g, aes(y = .resid, x = .fitted)) + geom_point()
```



```
ggplot(new_g, aes(sample = .resid)) + stat_qq() + stat_qq_line()
```



- The residual axis on the residual plot has better scale with 0 right in the middle. - On the normal quantile plot, all the point adhere to the line pretty closely.

(h)

Personally, I don't see any problems with the two plots:

- On the residue plot, all the residuals are scattered with no apparent patterns which is good. Moreover, the residual=0 is right in the middle, which seems to suggest there is no skewness.
- On the normal quantile plot, almost every point sticks very closely to the line with only a very few exceptions. Even then, not all points have to stick closely to the line. The general pattern is good and we cannot jump to any conclusions that the points are not normal enough.

(i)

- Use "rbind" to combine rows with the same variables. (source:https://www.statmethods.net/management/merging.html)
- Use "mutate" to create a new column of variables. We assign a type to each data set for differentiation purposes.

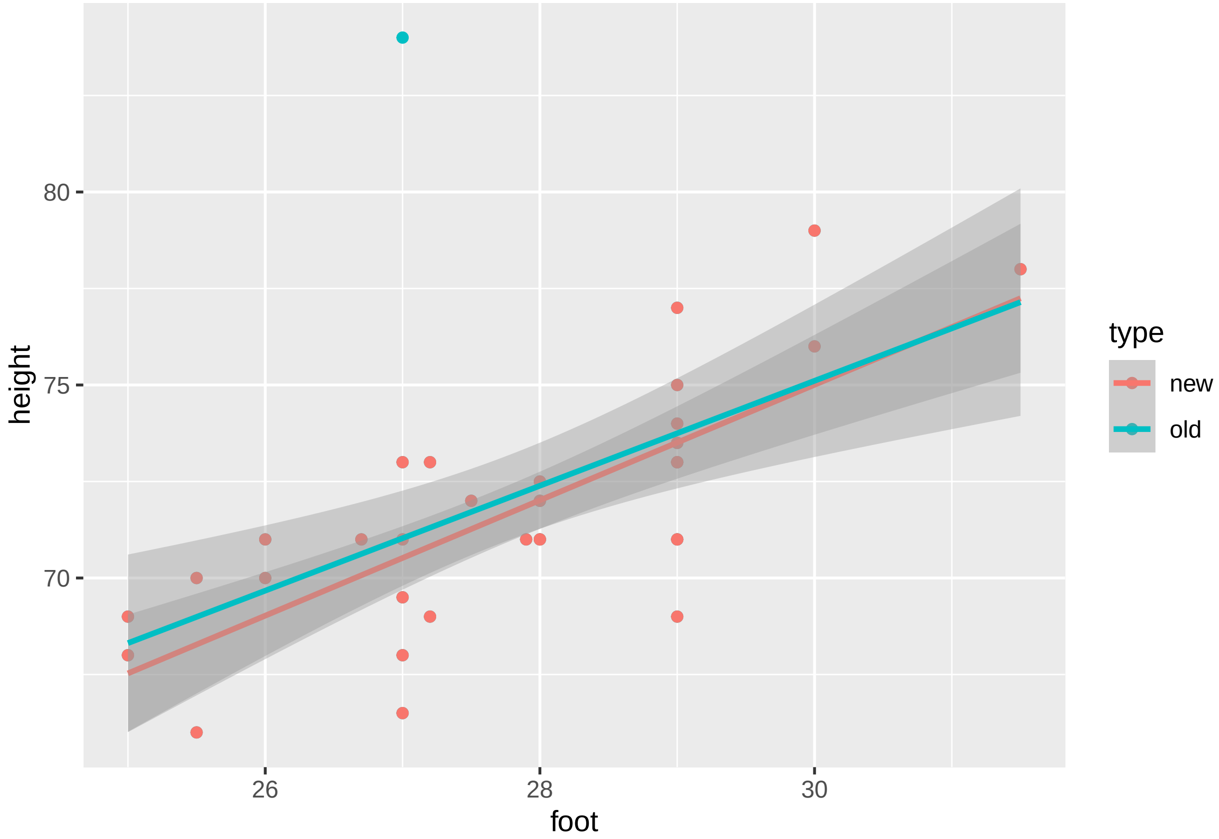
```
old <- data %>% mutate(type = "old")
new <- new_data %>% mutate(type = "new")
combined <- rbind(old, new)
combined
```

```
## # A tibble: 65 x 3
##    height  foot type
##     <dbl> <dbl> <chr>
##  1   66.5   27   old
##  2   73.5  29   old
##  3   70    25.5 old
##  4   71    27.9 old
##  5   73    27   old
##  6   71    26   old
##  7   71    29   old
##  8   69.5  27   old
##  9   73    29   old
## 10   71    27   old
## # ... with 55 more rows
```

- Now, we have a bigger data set that contains data from before and after we removed the unusual observation. They add up to 65 rows, which is what we expected.

```
ggplot(combined, aes(x = foot, y =  height, color = type)) + geom_point() + geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



- Use color to differentiate old and new data set. We can see they only differ by a single point, which is the one we removed earlier. Red represents new, blue represents old—points and lines altogether.

(j)

```
tidy(g)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    34.3      9.95       3.45 0.00164
## 2 foot             1.36     0.358      3.80 0.000643
```

```
tidy(new_g)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic     p.value
##   <chr>          <dbl>     <dbl>     <dbl>       <dbl>
## 1 (Intercept)    30.2      6.54       4.61 0.0000700
## 2 foot             1.50     0.235      6.36 0.000000512
```

- By removing the observation, the slope of the regression line increases. This is to be expected because we removed a point that has high height.
- Comparing the regression analysis of old with new data, it checks out as well because the new regression line has slope 1.495156 compared to 1.359062.