InClassAssignment4

Christopher Hainzl, Christopher Hakkenberg, Peter Bitanga, Ryan Podzielny, Christopher Barbieri

2022-11-04

Read the wire data using read_excel.

```
wires <- read_excel("Wire_bond.xlsx")
wires2 <- wires[,c(2:8)]
wires3 <- wires2 %>% filter(y != "pull strength")
wires4 <- lapply(wires3, as.numeric)
wires5 <- data.frame(wires4)
wires5
```

```
##       y  X.1   x2   x3   x4  x5 X.6
## 1   8.0  5.2 17.0 28.6 83.0 1.9 1.6
## 2   8.0  5.2 19.6 29.6 94.9 2.1 2.3
## 3   8.3  5.8 19.8 32.4 89.7 2.1 1.8
## 4   8.5  6.4 19.6 31.0 96.2 2.0 2.0
## 5   8.8  5.8 19.4 32.4 95.6 2.2 2.1
## 6   9.0  5.2 18.6 28.6 86.5 2.0 1.8
## 7   9.3  5.6 18.8 30.6 84.5 2.1 2.1
## 8   9.3  6.0 20.4 32.4 88.8 2.2 1.9
## 9   9.5  5.2 19.0 32.6 85.7 2.1 1.9
## 10  9.8  5.8 20.8 32.2 93.6 2.3 2.1
## 11 10.0  6.4 19.9 31.8 86.0 2.1 1.8
## 12 10.3  6.0 18.0 32.6 87.1 2.0 1.6
## 13 10.5  6.2 20.6 33.4 93.1 2.1 2.1
## 14 10.8  6.2 20.2 31.8 83.4 2.2 2.1
## 15 11.0  6.2 20.2 32.4 94.5 2.1 1.9
## 16 11.3  5.6 19.2 31.4 83.4 1.9 1.8
## 17 11.5  6.0 17.0 33.2 85.2 2.1 2.1
## 18 11.8  5.8 19.8 35.4 84.1 2.0 1.8
## 19 12.0  6.5 19.4 32.2 88.5 2.1 1.9
## 20 12.3  5.6 18.8 34.0 86.9 2.1 1.8
## 21 12.5 18.6 18.6 34.2 83.0 1.9 2.0
## 22 12.5 18.6 20.8 35.4 96.2 2.3 2.3
```

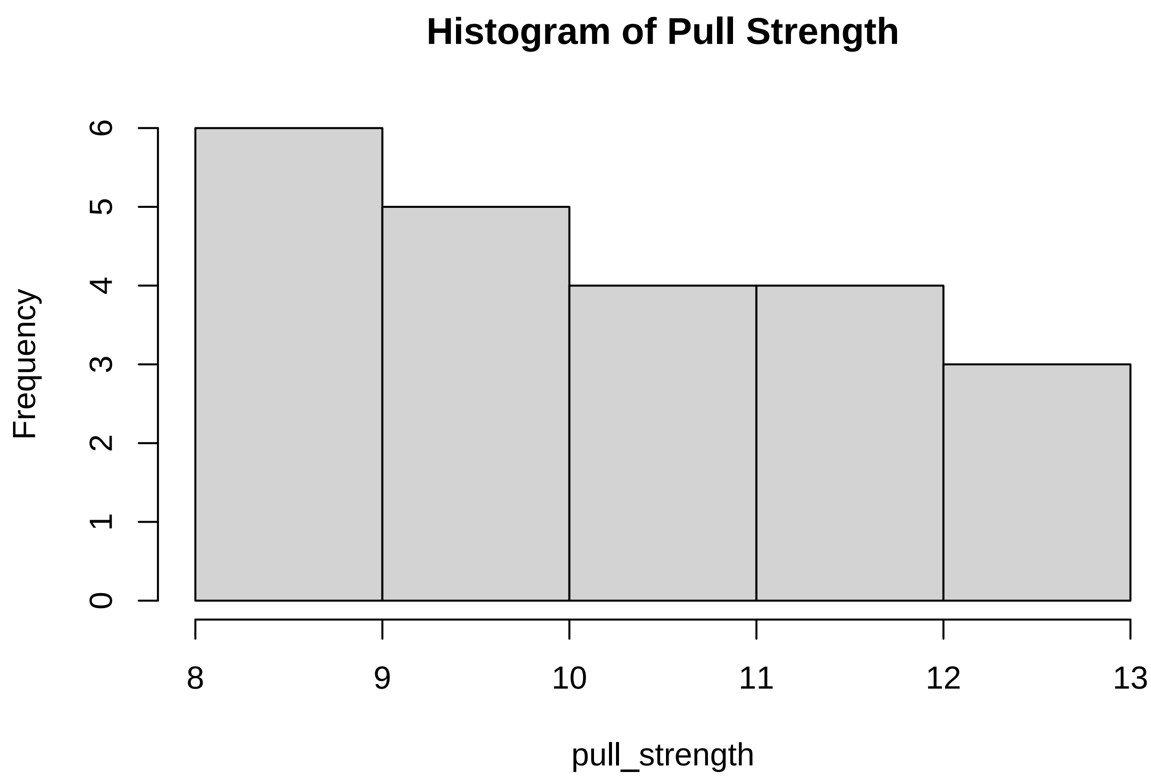
```
attach(wires5)
```

a. We decided that a multiple linear regression would be best since there is one response variable (pull strength) and 6 predictor variables (die height, post height, loop height, wire length, bond width on the die, and bond width on the post). Since those 6 predictors could have an impact on our response, we want to see if there exists any sort of trend.

First, we looked at the histogram of our response to see if there is a normal distribution and if it is good for a multiple linear regression.

Although the data did not appear to have a normal distribution, we wanted to continue with the multiple linear regression. We also decided to do a correlation matrix of the data to see if we should explore further.

We did see some variables correlate with the response (y), so we continued with the multiple regression.




```
cor(wires5)
```

```
##                y        X.1         x2         x3          x4         x5
## y    1.00000000 0.52759084 0.08252147 0.74417560 -0.22471996 0.02787245
## X.1  0.52759084 1.00000000 0.14814194 0.50563155  0.08904081 0.06258709
## x2   0.08252147 0.14814194 1.00000000 0.32651787  0.55814522 0.61980242
## x3   0.74417560 0.50563155 0.32651787 1.00000000  0.08216081 0.29037410
## x4  -0.22471996 0.08904081 0.55814522 0.08216081  1.00000000 0.53880391
## x5   0.02787245 0.06258709 0.61980242 0.29037410  0.53880391 1.00000000
## X.6  0.07530502 0.35084032 0.43912929 0.18276383  0.52715255 0.60295342
##            X.6
## y   0.07530502
## X.1 0.35084032
## x2  0.43912929
## x3  0.18276383
## x4  0.52715255
## x5  0.60295342
## X.6 1.00000000
```

b.

```
Multiple.model <- lm(y ~ X.1 + x2 + x3 + x4 + x5 + X.6, data = wires5)
summary(Multiple.model)
```

```
##
## Call:
```

```
## lm(formula = y ~ X.1 + x2 + x3 + x4 + x5 + X.6, data = wires5)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.74699 -0.51092 -0.08305  0.38206  1.83533
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.67368    5.89126   0.114  0.91048
## X.1          0.07229    0.07560   0.956  0.35409
## x2           0.02256    0.28620   0.079  0.93820
## x3           0.56022    0.15559   3.601  0.00262 **
## x4          -0.09780    0.06143  -1.592  0.13221
## x5          -1.02170    3.02096  -0.338  0.73990
## X.6          0.70783    1.66409   0.425  0.67662
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.002 on 15 degrees of freedom
## Multiple R-squared:  0.6772, Adjusted R-squared:  0.548
## F-statistic: 5.244 on 6 and 15 DF,  p-value: 0.004298
```

Aside from X.1 (die height) and x3 (loop height), the standard deviation of the pull strength is negligible.

Variation: 0.6772

```r
Multiple.model1 <- lm(y ~ X.1 + x3, data = wires5)
summary(Multiple.model1)
```

```
##
## Call:
## lm(formula = y ~ X.1 + x3, data = wires5)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.94223 -0.62204 -0.04863  0.79457  1.80766
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.40274    4.34649  -1.703  0.10484
## X.1          0.08023    0.06764   1.186  0.25022
## x3           0.53024    0.14169   3.742  0.00138 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.01 on 19 degrees of freedom
## Multiple R-squared:  0.5846, Adjusted R-squared:  0.5408
## F-statistic: 13.37 on 2 and 19 DF,  p-value: 0.0002376
```

c.

```r
confint(Multiple.model, level = 0.95)
```

```
##                    2.5 %      97.5 %
## (Intercept) -11.88324388 13.23059462
## X.1          -0.08884212  0.23343123
## x2           -0.58746247  0.63259206
```

```
## x3           0.22859932  0.89184584
## x4          -0.22872546  0.03312879
## x5          -7.46072560  5.41731748
## X.6         -2.83908442  4.25475025
```

x5 (bond width on the die) has the highest distance between confidence levels. Meanwhile, X.1 and x3 have the lowest distances between confidence levels. For each predictor value, we are 95% confident that they lie within the specified ranges above.

d.

```
summary(Multiple.model)
```

```
##
## Call:
## lm(formula = y ~ X.1 + x2 + x3 + x4 + x5 + X.6, data = wires5)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.74699 -0.51092 -0.08305  0.38206  1.83533
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.67368    5.89126   0.114  0.91048
## X.1          0.07229    0.07560   0.956  0.35409
## x2           0.02256    0.28620   0.079  0.93820
## x3           0.56022    0.15559   3.601  0.00262 **
## x4          -0.09780    0.06143  -1.592  0.13221
## x5          -1.02170    3.02096  -0.338  0.73990
## X.6          0.70783    1.66409   0.425  0.67662
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.002 on 15 degrees of freedom
## Multiple R-squared:  0.6772, Adjusted R-squared:  0.548
## F-statistic: 5.244 on 6 and 15 DF,  p-value: 0.004298
```

For every unit change in x4, there is a 0.0978 decrease in the average value of y.

e.

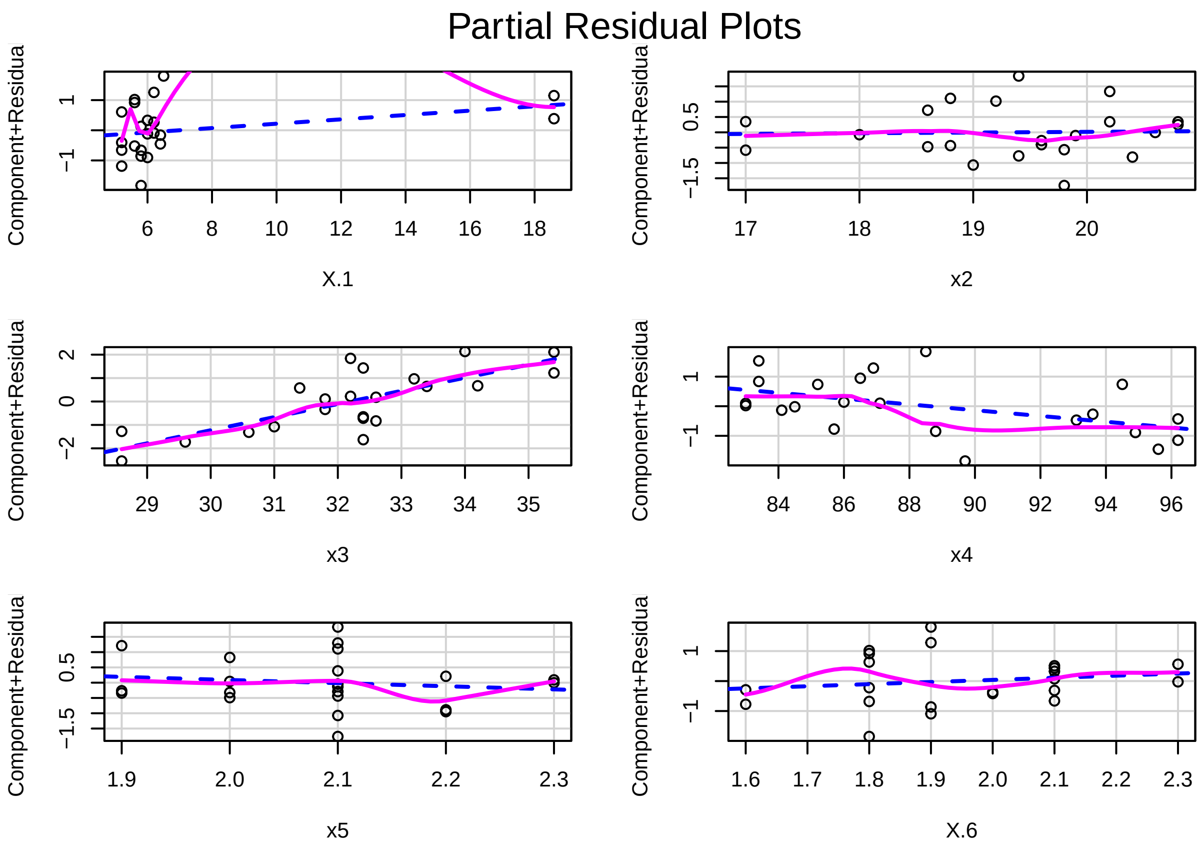
```
new <- data.frame(X.1 = 5.5, x2 = 19.3, x3 = 30.2, x4 = 90, x5 = 2, X.6 = 1.85)
pred <- predict(Multiple.model, newdata = new, se.fit = TRUE, interval = "prediction", level = 0.95)
pred
```

```
## $fit
##       fit      lwr      upr
## 1 8.88975 6.590916 11.18858
##
## $se.fit
## [1] 0.398635
##
## $df
## [1] 15
##
## $residual.scale
## [1] 1.002157
```

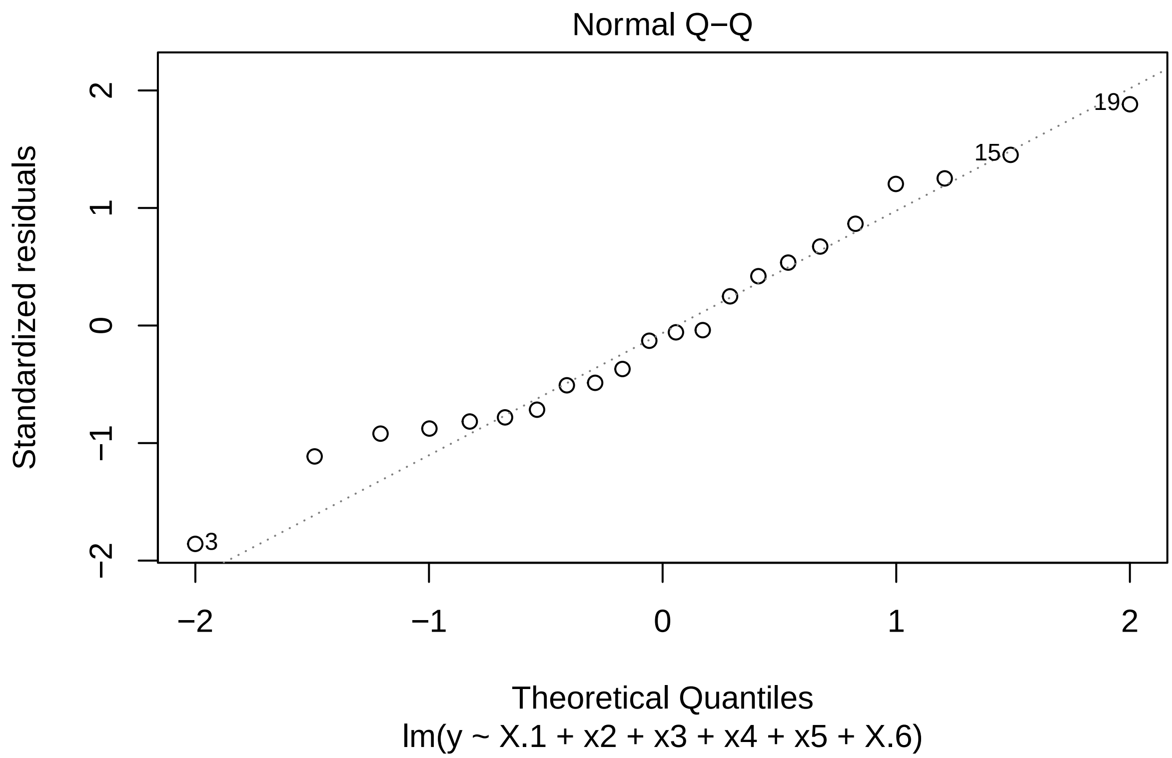
The predicted value for y is 8.88975.

f.

```
crPlots(Multiple.model, main = "Partial Residual Plots")
```




```
plot(Multiple.model, 2)
```



Only one of our assumptions was wrong: it turns out pull strength isn't impacted by die height when looking at partial residual plots. When looking at normality through a normal qqplot, we see that all the points approximately are along the reference line. This means we can assume normality.

Contributions:

Christopher Hainzl - I wrote the code.

Peter Bitanga - I helped with the processing of the data, as well as organization.

Ryan Podzielny - I helped with the code and interpretations.

Christopher Hakkenberg - I helped with the interpretations and analyzing the data.

Christopher Barbieri - I helped with the interpretations.