Lab 3

Daniel Tshiani

2025-05-16

```r
library(readr)
library(ggplot2)
library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

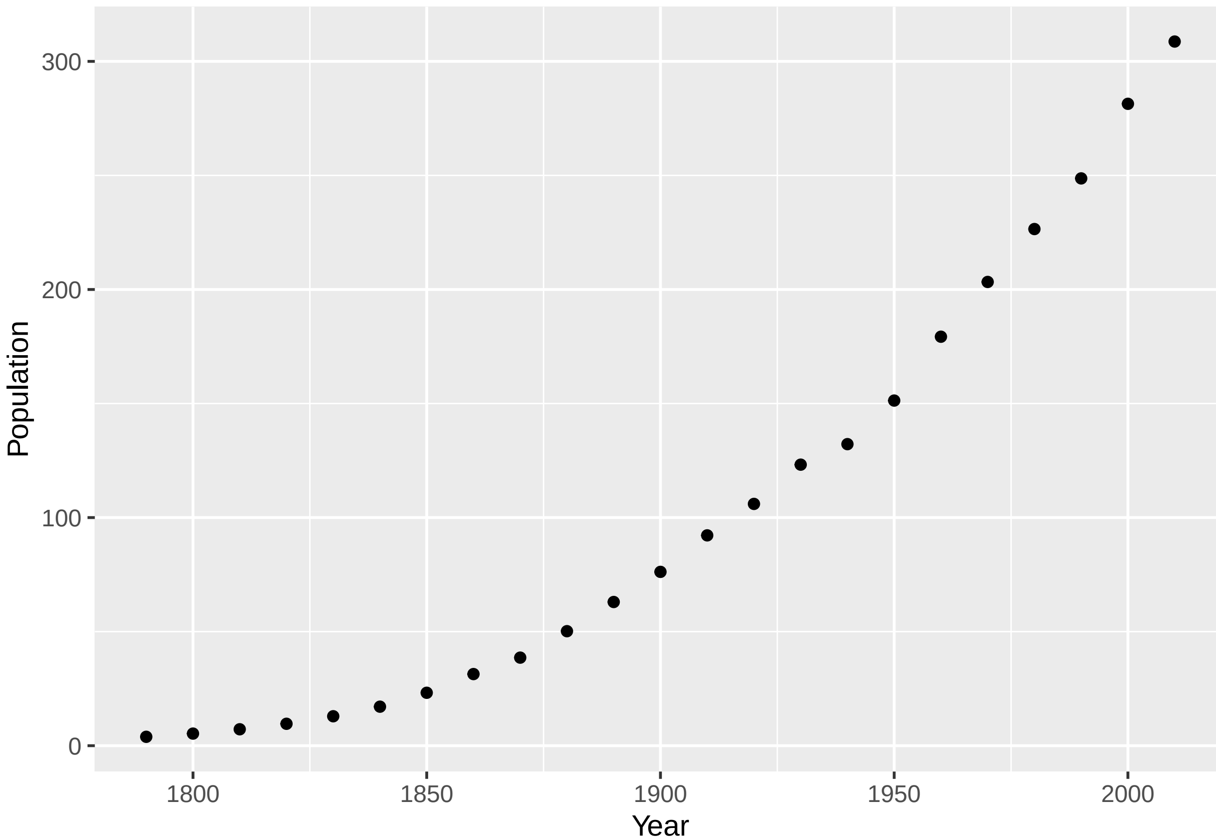
Excersice 1

a

```r
USpop <- read_csv("../data/USpop.csv")
```

```
## Rows: 23 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (2): Year, Population
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
ggplot(data = USpop, mapping = aes(y = Population, x = Year))+
  geom_point()
```



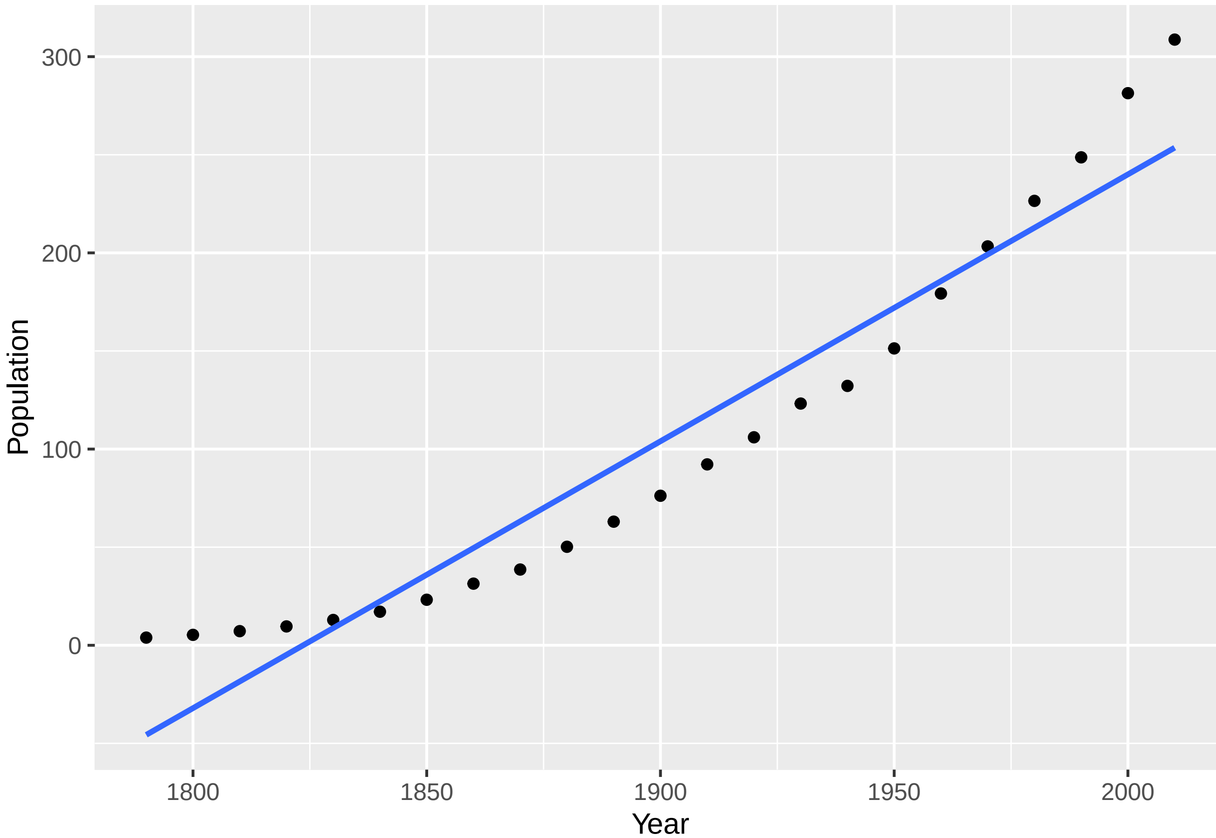
b

```r
model <- lm(Population ~ Year, data = USpop)
summary(model)
```

```
##
## Call:
## lm(formula = Population ~ Year, data = USpop)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.774 -24.872  -6.295  18.374  55.087
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.481e+03  1.672e+02  -14.84 1.33e-12 ***
## Year         1.360e+00  8.794e-02   15.47 5.93e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.97 on 21 degrees of freedom
## Multiple R-squared:  0.9193, Adjusted R-squared:  0.9155
## F-statistic: 239.3 on 1 and 21 DF,  p-value: 5.927e-13
```

```r
ggplot(data = USpop, mapping = aes(y = Population, x = Year))+
  geom_point() +
  geom_smooth(method = "lm", se=F)
```

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



when looking at the plot, it looks like the linear model does not provide the best fit.

c

```r
summary(model)
```

```
##
## Call:
## lm(formula = Population ~ Year, data = USpop)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.774 -24.872  -6.295  18.374  55.087
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.481e+03  1.672e+02  -14.84 1.33e-12 ***
## Year         1.360e+00  8.794e-02   15.47 5.93e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.97 on 21 degrees of freedom
## Multiple R-squared:  0.9193, Adjusted R-squared:  0.9155
## F-statistic: 239.3 on 1 and 21 DF,  p-value: 5.927e-13
```

multiple R-squared is 0.9193 and adjusted R-squared is 0.9155. They both suggest the linear model is a good choice however, when I look at the plot i wouldn't agree with that.

d

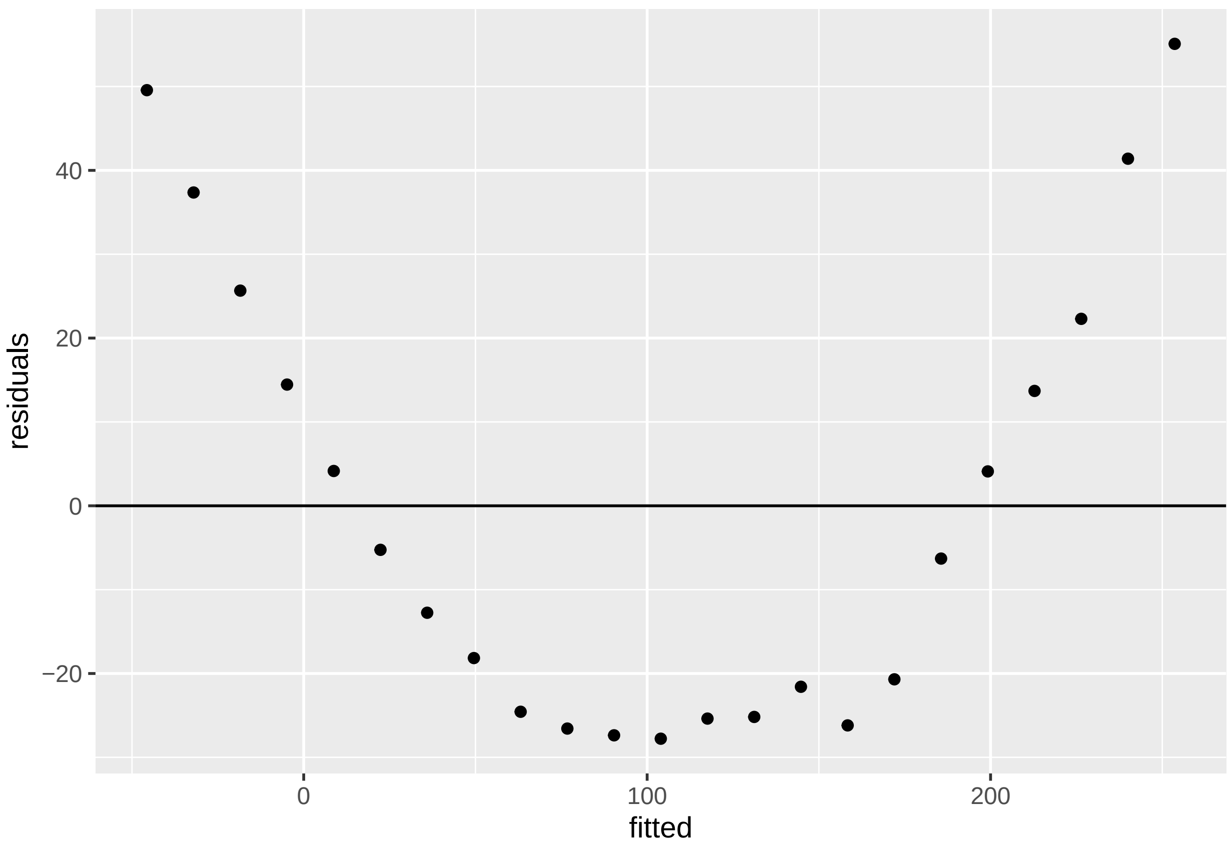
```
predict(model, newdata = data.frame(Year = 2030))
```

```
##        1
## 280.8202
```

I dont think its the best prediction because when i look at the graph, it looks like the population increases exponentially rather than linearly.

e

```
USpop$residuals <- resid(model)
USpop$fitted <- fitted(model)

ggplot(data = USpop, aes(x = fitted, y = residuals)) +
  geom_point()+
  geom_hline(yintercept = 0)
```



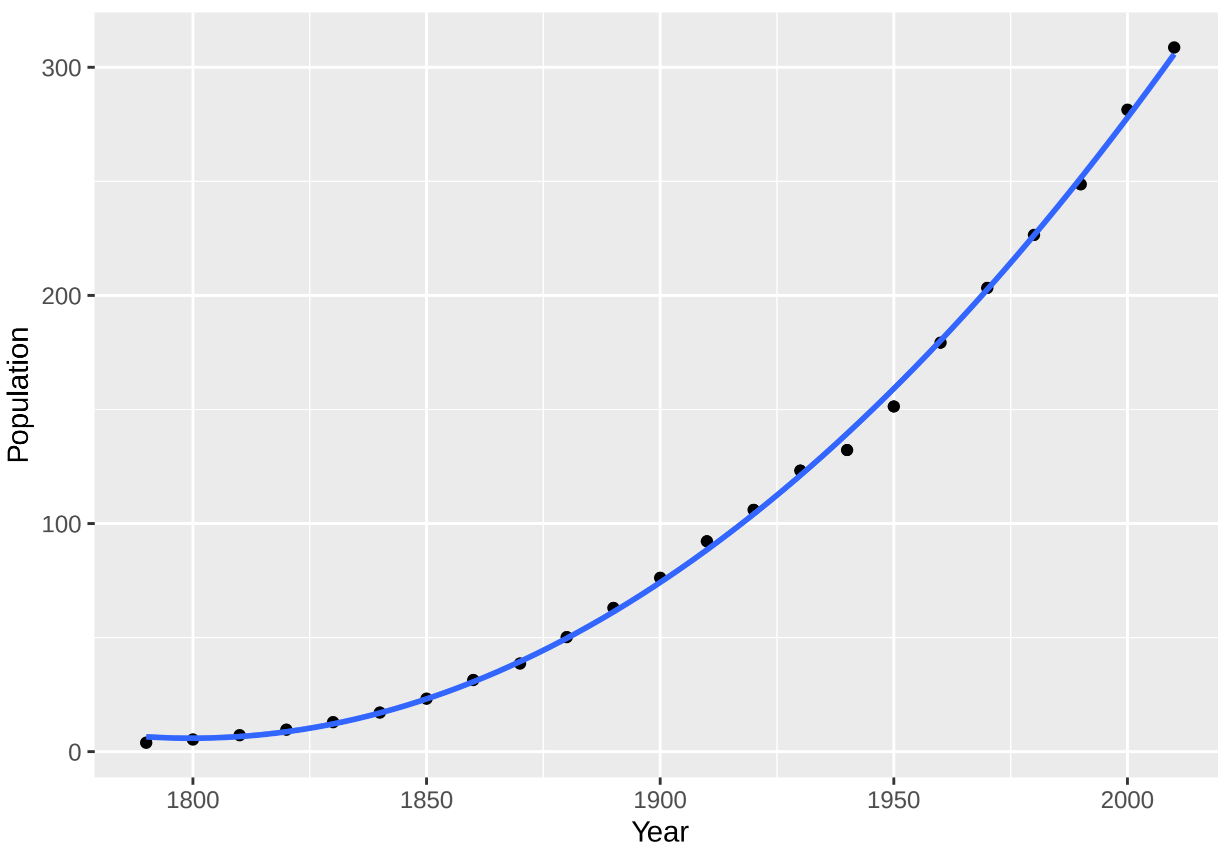
the residual plot looks like a quadratic function so key variables could be ommitted or we need to use a quadric variable in our model.

f

```r
model2 <- lm(Population ~ Year +I(Year^2), data = USpop)
summary(model2)
```

```
##
## Call:
## lm(formula = Population ~ Year + I(Year^2), data = USpop)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.8220 -0.7130  0.5961  1.8344  3.7487
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.194e+04  5.795e+02   37.87   <2e-16 ***
## Year        -2.438e+01  6.105e-01  -39.94   <2e-16 ***
## I(Year^2)    6.774e-03  1.606e-04   42.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.023 on 20 degrees of freedom
## Multiple R-squared:  0.9991, Adjusted R-squared:  0.999
## F-statistic: 1.113e+04 on 2 and 20 DF,  p-value: < 2.2e-16
```

```r
ggplot(data = USpop, aes(x = Year, y = Population)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x + I(x^2), se = F)
```



this looks like a good fit.

g

```r
predict(model2, newdata = data.frame(Year = 2030))
```

```
##        1
## 365.4891
```

I think this is a reasonable prediction.

exercise 2

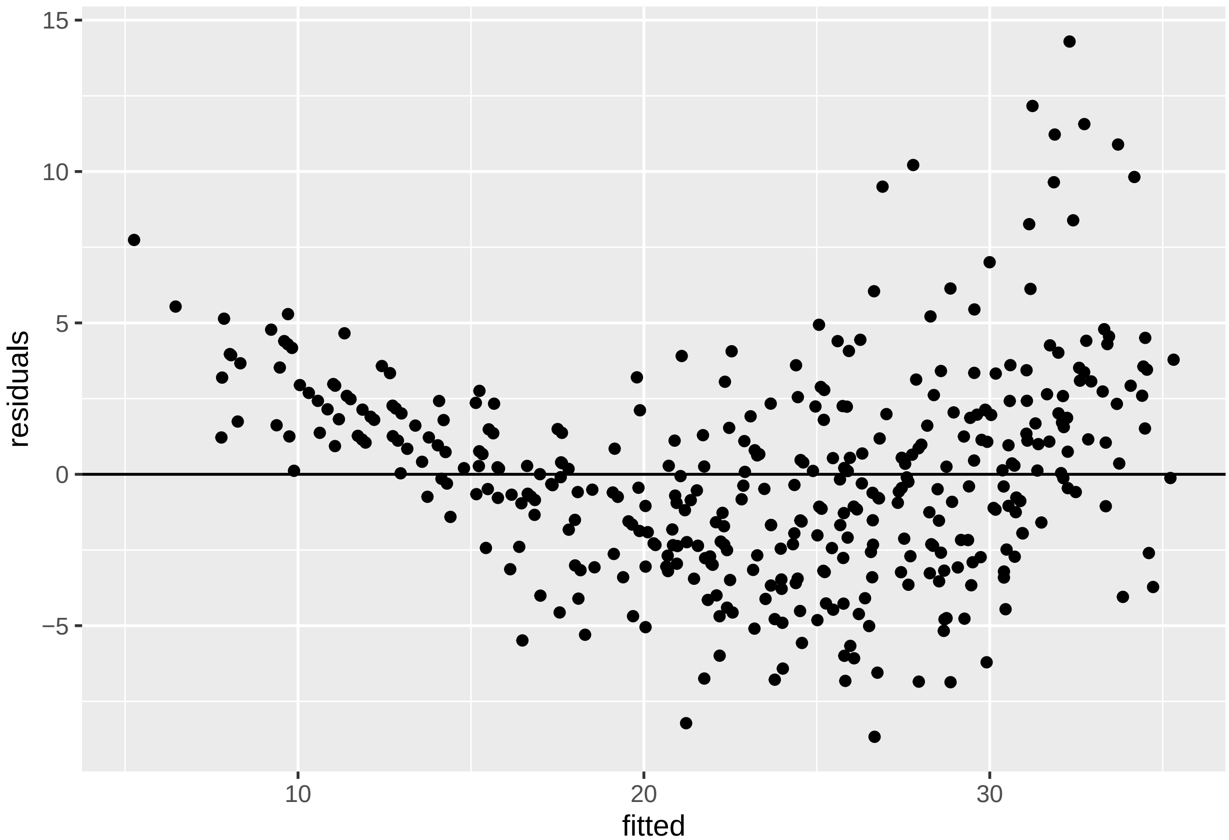
a

```r
rm(model, model2, USpop)
load("../data/Auto-3.rda")
```

```r
model <- lm(mpg ~ year + acceleration + horsepower + weight, data = Auto)
summary(model)
```

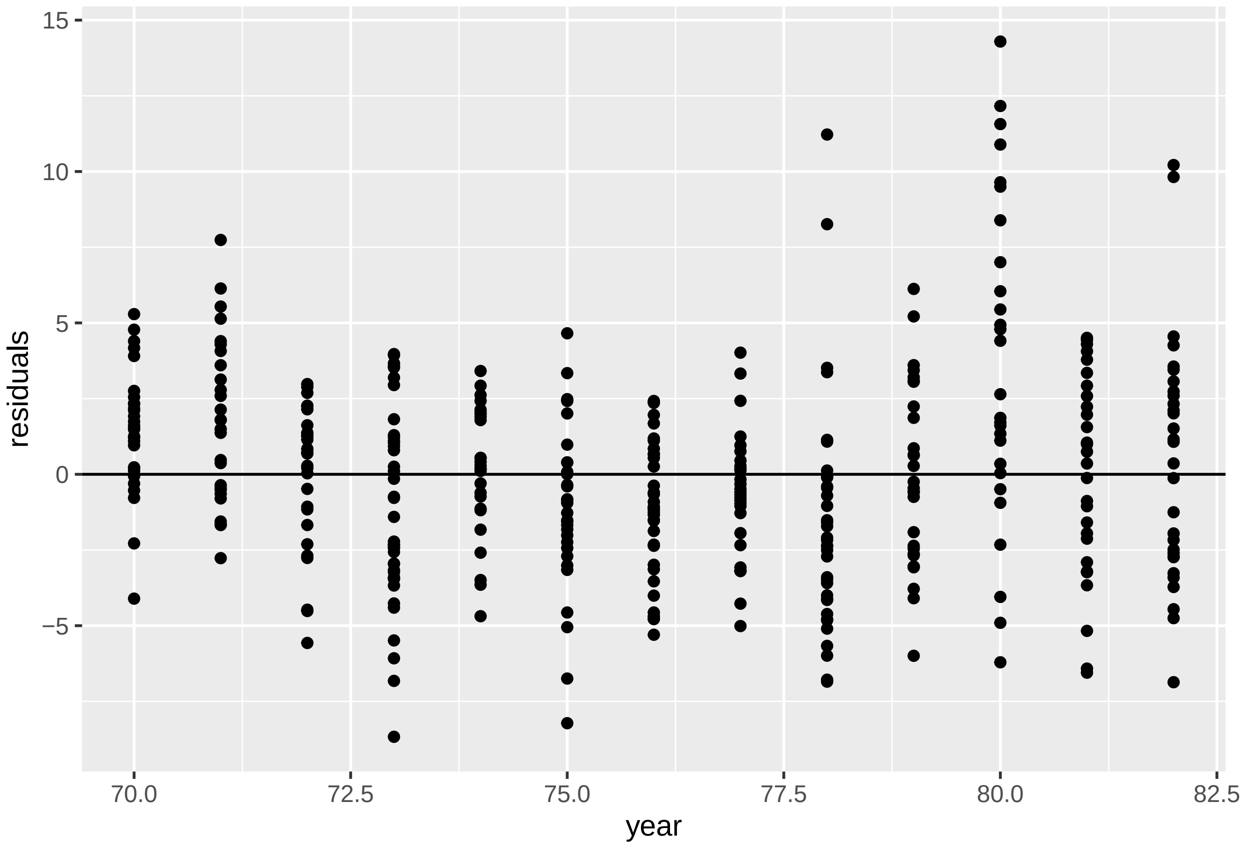
```
##
## Call:
## lm(formula = mpg ~ year + acceleration + horsepower + weight,
##     data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.6693 -2.3618 -0.0982  2.0105 14.2926
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.539e+01  4.671e+00  -3.294  0.00108 **
## year          7.511e-01  5.223e-02  14.381  < 2e-16 ***
## acceleration  8.022e-02  9.986e-02   0.803  0.42228
## horsepower    2.622e-03  1.339e-02   0.196  0.84483
## weight       -6.634e-03  4.706e-04 -14.099  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.432 on 387 degrees of freedom
## Multiple R-squared:  0.8086, Adjusted R-squared:  0.8067
## F-statistic: 408.8 on 4 and 387 DF,  p-value: < 2.2e-16
```

```r
Auto$residuals <- resid(model)
Auto$fitted <- fitted(model)

ggplot(data = Auto, aes(x = fitted, y = residuals)) +
  geom_point() +
  geom_hline(yintercept = 0)
```

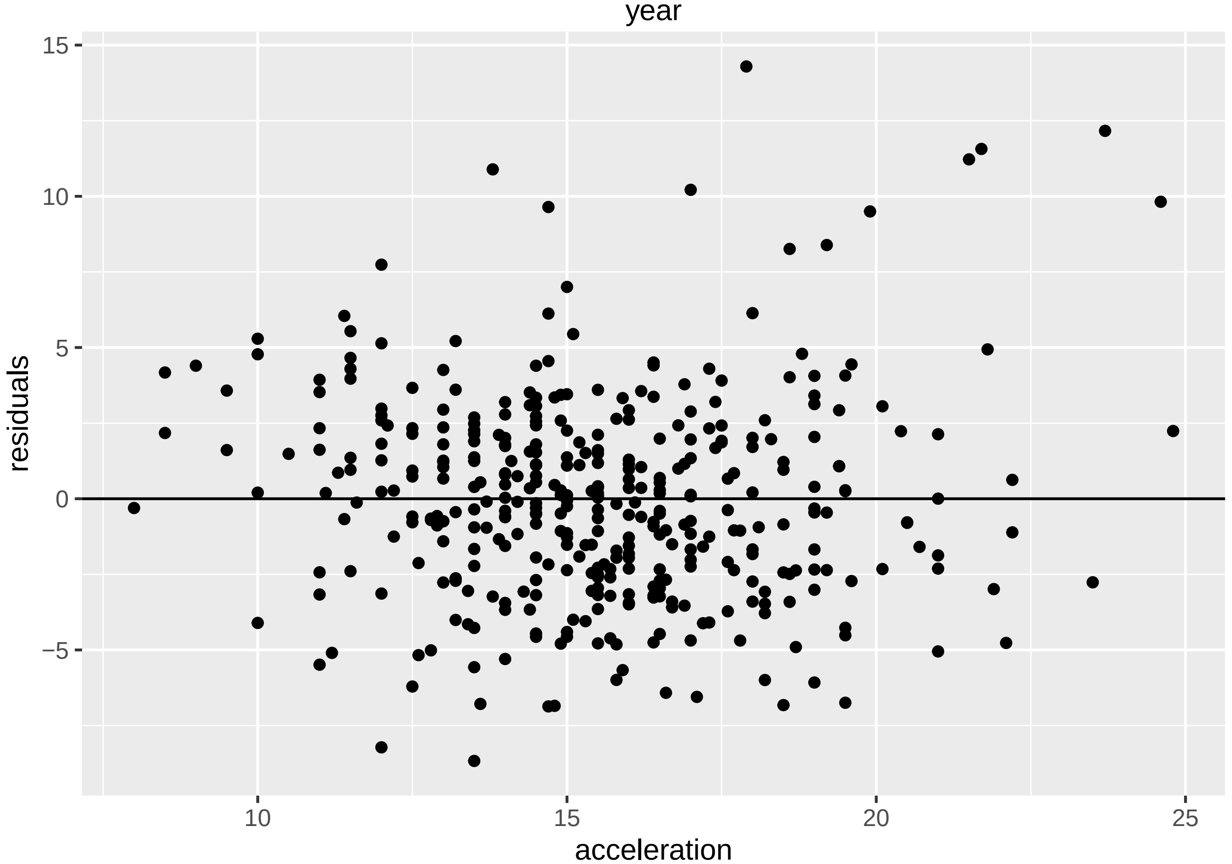



```
predictors <- c("year", "acceleration", "horsepower", "weight")

for (var in predictors) {
  p <- ggplot(Auto, aes_string(x = var, y = "residuals")) +
    geom_point() +
    geom_hline(yintercept = 0)
  print(p)
}
```

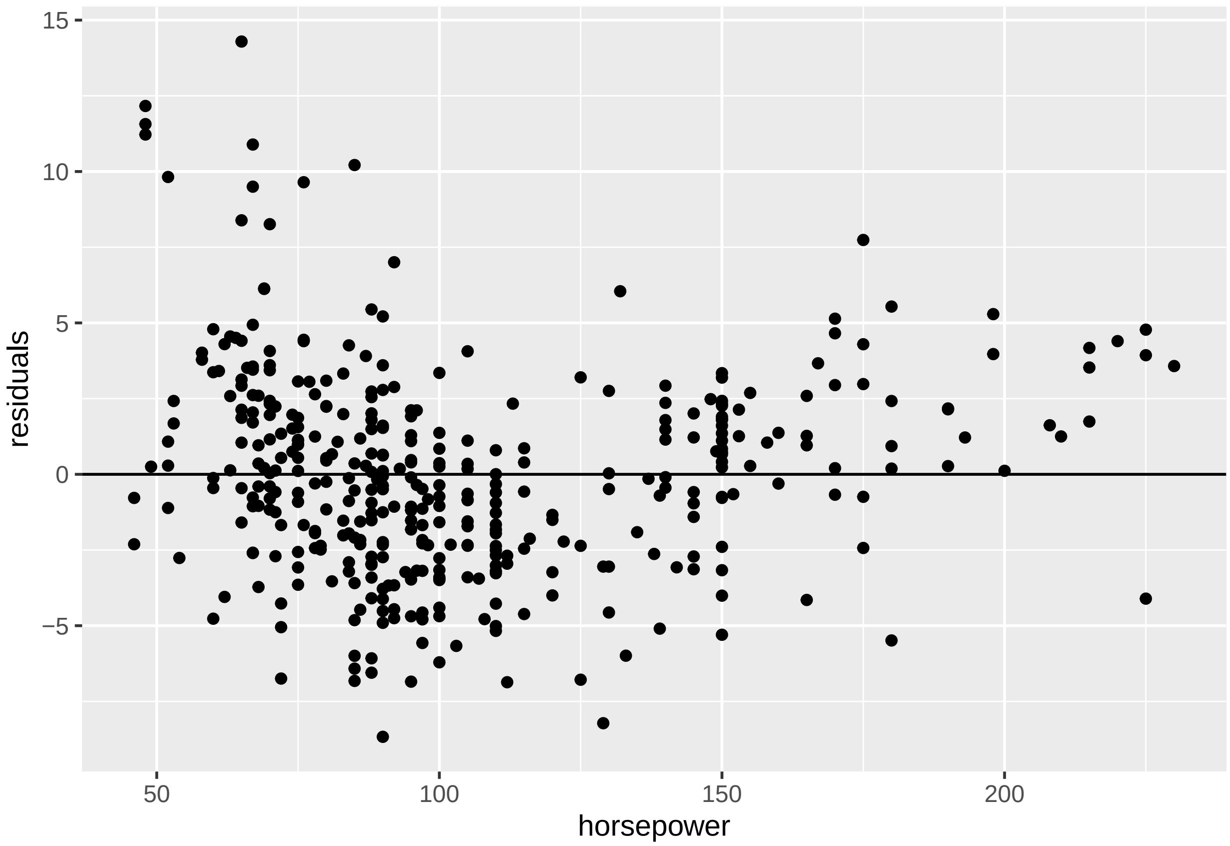
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

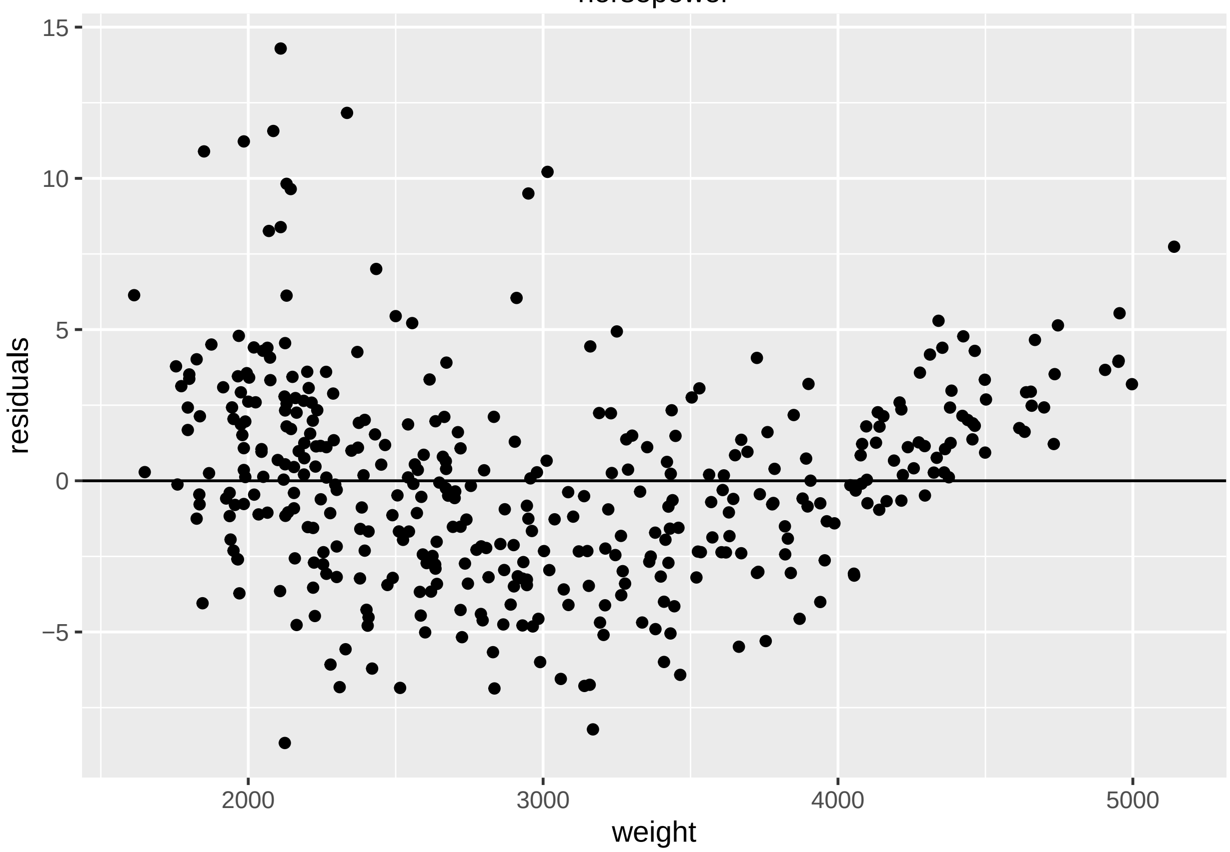


15
10
5
0
−5
residuals
70.0
72.5
75.0
77.5
80.0
82.5
year



15
10
5
0
−5
residuals
10
15
20
25
acceleration


residuals
15
10
5
0
−5
50
100
150
200
horsepower


residuals
15
10
5
0
−5
2000
3000
4000
5000
weight

b

```
student_res <- rstudent(model)
n <- nrow(Auto)
alpha <- 0.05
df <- model$df.residual
t_crit <- qt(1 - alpha / (2 * n), df)
```

```
outliers <- which(abs(student_res) > t_crit)
Auto[outliers, ]
```

```
##      mpg cylinders displacement horsepower weight acceleration year origin
## 323 46.6         4           86         65   2110         17.9   80      3
##          name residuals   fitted
## 323 mazda glc  14.29259 32.30741
```

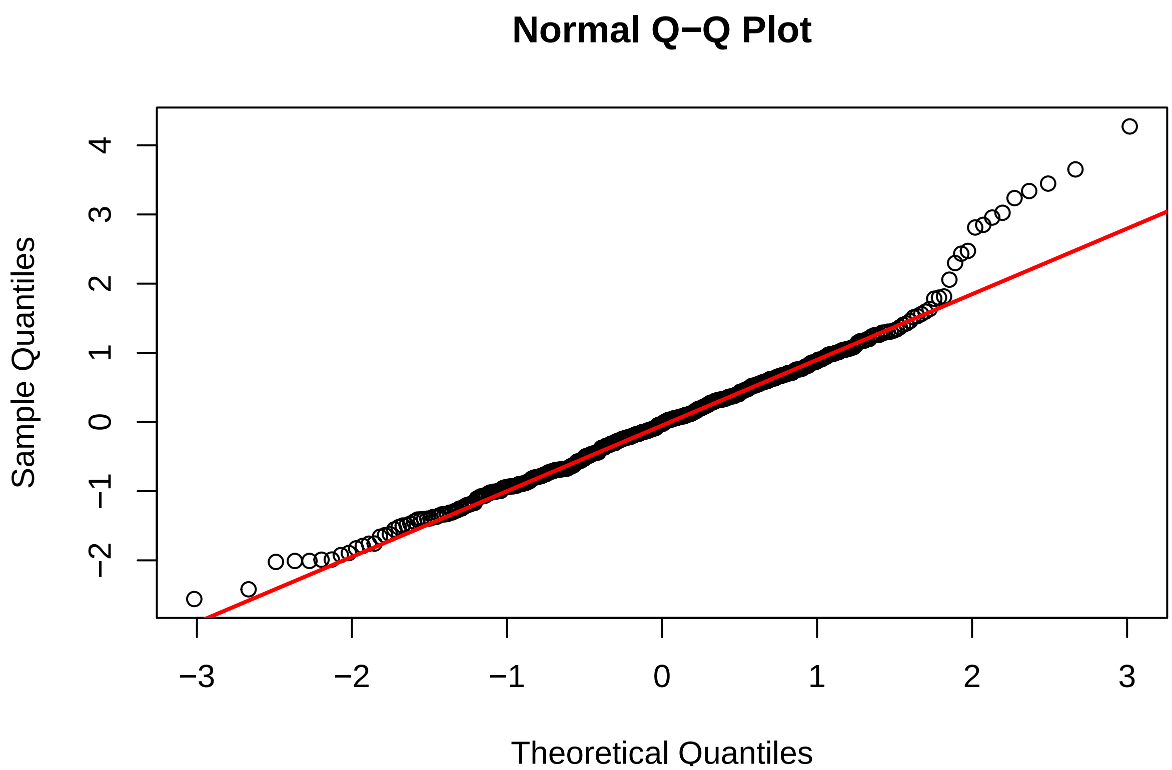
c

```
shapiro.test(student_res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  student_res
## W = 0.97109, p-value = 5.101e-07
```

the p value is less than 0.05 which means we reject the null. the residuals are likely not normally distributed.

```
qqnorm(student_res)
qqline(student_res, col = "red", lwd = 2)
```



the qq plot confirms non normal distribution because the point deviated from the reference line towards the ends.

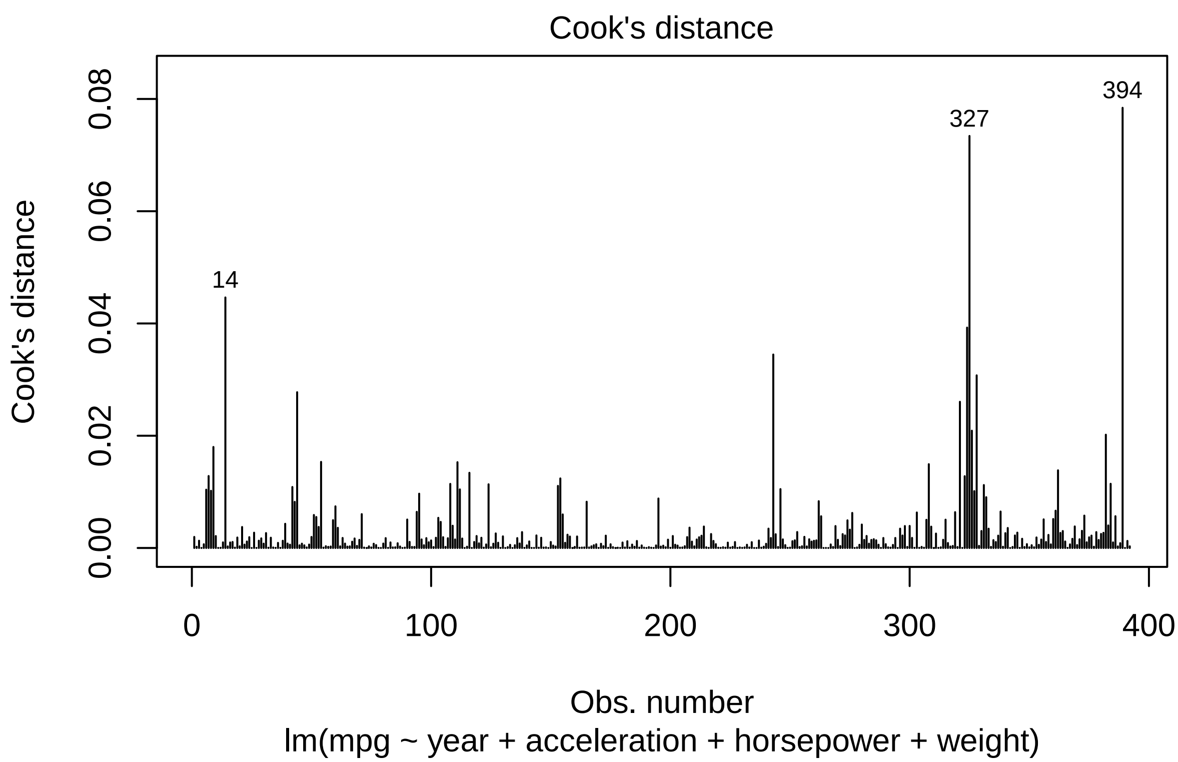
d

```r
bptest(model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 25.352, df = 4, p-value = 4.274e-05
```

the p value is less than 0.05 which mean we reject the null. this suggest that the residuals do not have constant variance. ## e

```r
plot(model, which = 4)
```



I would say there are a handful of influential data.