Appendix

Jadon Fowler, STA 570 Section 1

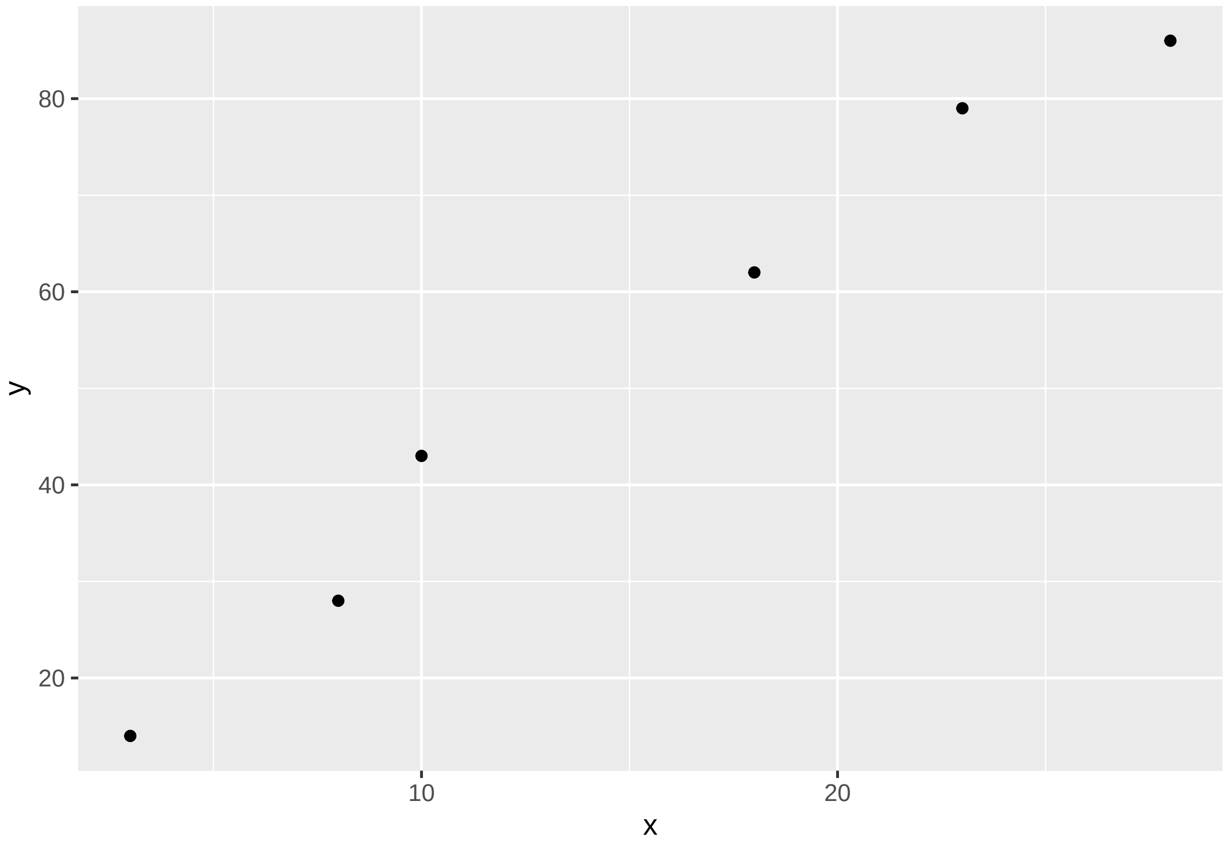
Homework 10, 2023/05/02

```
library(car)
library(ggplot2)
library(dplyr)
library(ggfortify)
library(emmeans)
library(multcompView)
library(multcomp)
```

1.a.

```
# read in the data
p1.data <- data.frame(
 x = c( 3, 8, 10, 18, 23, 28),
 y = c(14, 28, 43, 62, 79, 86) )

# make a nice graph
ggplot(p1.data, aes(x=x, y=y)) +
 geom_point()
```



1.c.

```
p1.model <- lm(y ~ x, p1.data)
anova(p1.model)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x          1 4003.1  4003.1  193.27 0.0001552 ***
## Residuals  4   82.9    20.7
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
predict(p1.model)
```

```
##     1     2     3     4     5     6
## 16.60 31.35 37.25 60.85 75.60 90.35
```

```
resid(p1.model)
```

```
##     1     2     3     4     5     6
## -2.60 -3.35  5.75  1.15  3.40 -4.35
```

```
summary(p1.model)
```

```
##
## Call:
## lm(formula = y ~ x, data = p1.data)
##
## Residuals:
##     1     2     3     4     5     6
## -2.60 -3.35  5.75  1.15  3.40 -4.35
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.7500     3.6855   2.103 0.103325
## x             2.9500     0.2122  13.902 0.000155 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4.551 on 4 degrees of freedom
## Multiple R-squared:  0.9797, Adjusted R-squared:  0.9747
## F-statistic: 193.3 on 1 and 4 DF,  p-value: 0.0001552
```

1.d.

```
# 'x' here is the slope CI
confint(p1.model)
```

```
##                 2.5 %    97.5 %
## (Intercept) -2.482697 17.982697
## x            2.360849  3.539151
```

1.e.

```
summary(p1.model)
```

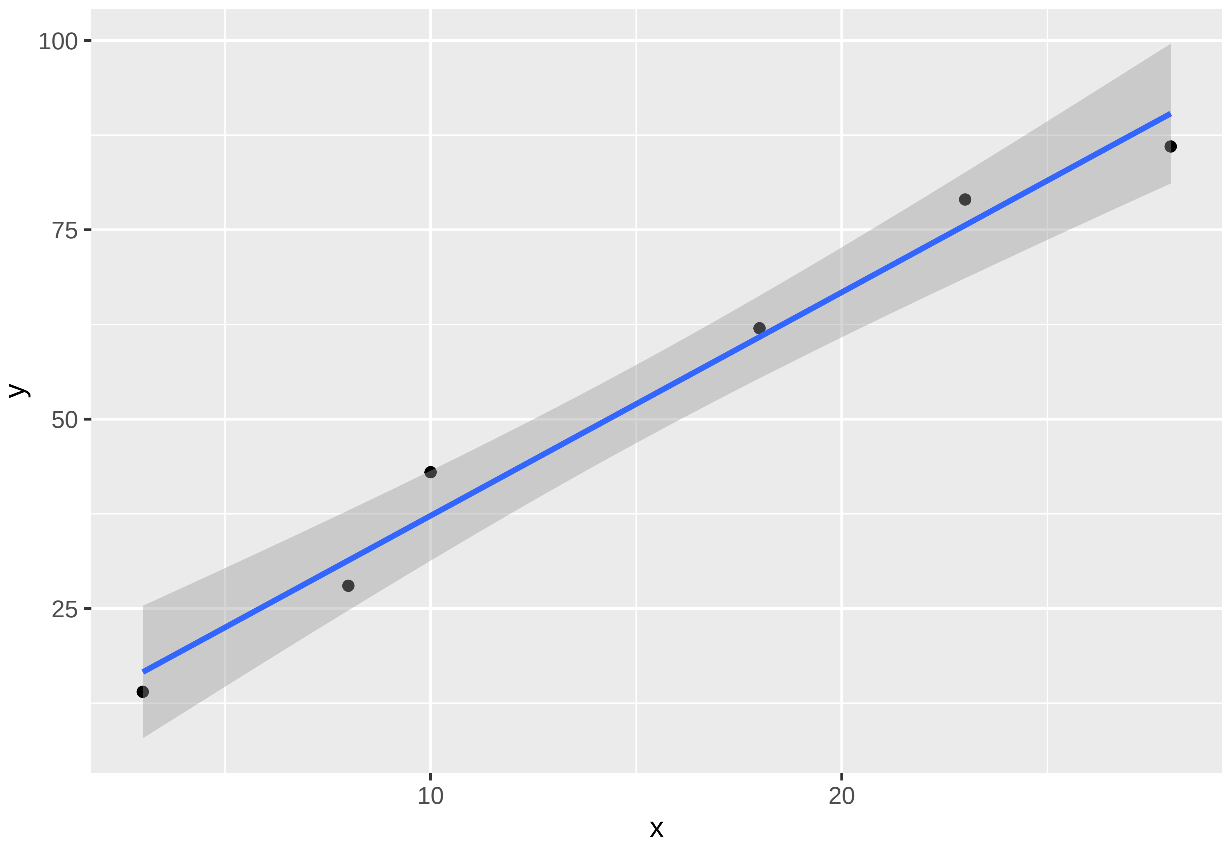
```
##
## Call:
## lm(formula = y ~ x, data = p1.data)
##
## Residuals:
##     1     2     3     4     5     6
## -2.60 -3.35  5.75  1.15  3.40 -4.35
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.7500     3.6855   2.103 0.103325
## x             2.9500     0.2122  13.902 0.000155 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4.551 on 4 degrees of freedom
## Multiple R-squared:  0.9797, Adjusted R-squared:  0.9747
## F-statistic: 193.3 on 1 and 4 DF,  p-value: 0.0001552
```

```
# e. p = 0.000155, we can reject the null hypothesis
# f. R^2 = 0.9797
# g. typical distance = stddev = 4.551
```

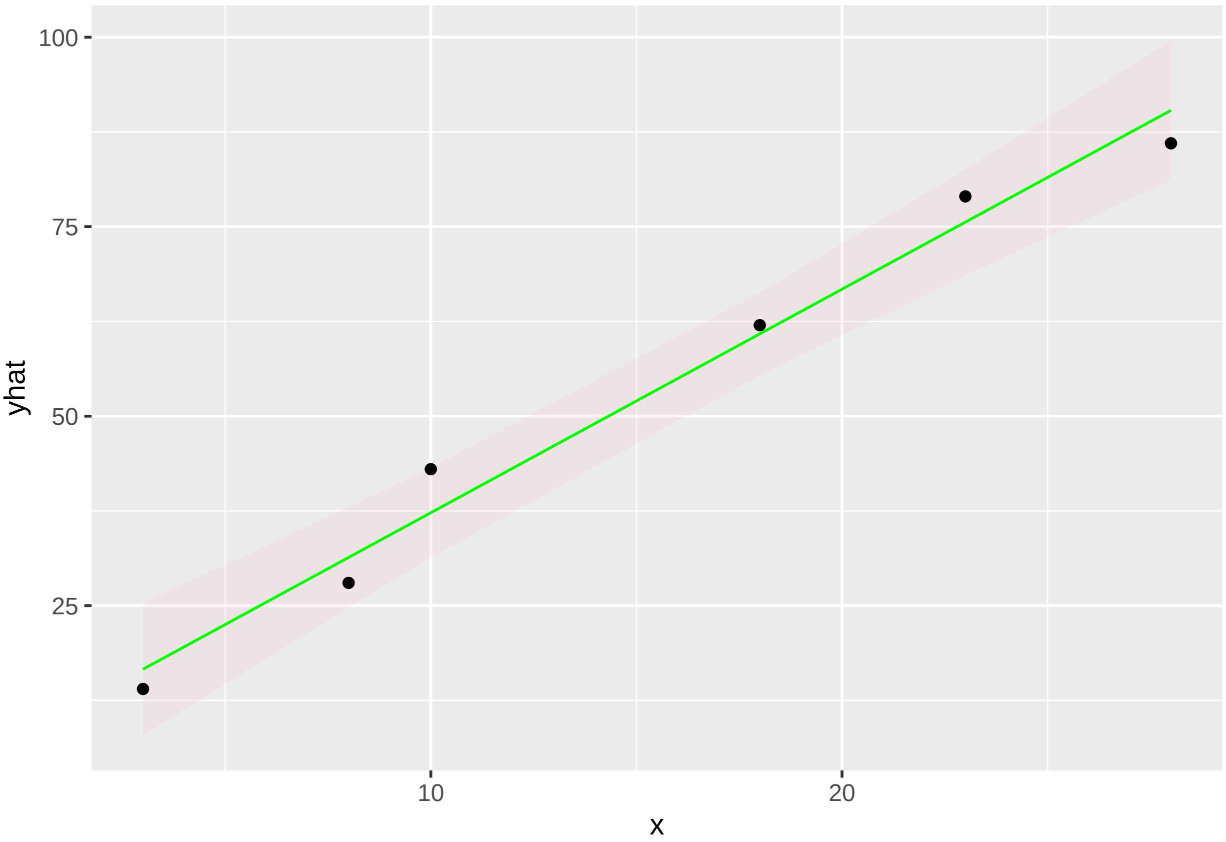
1.h.

```
# make a nice graph
ggplot(p1.data, aes(x=x, y=y)) +
 geom_point() +
 geom_smooth(method='lm')
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

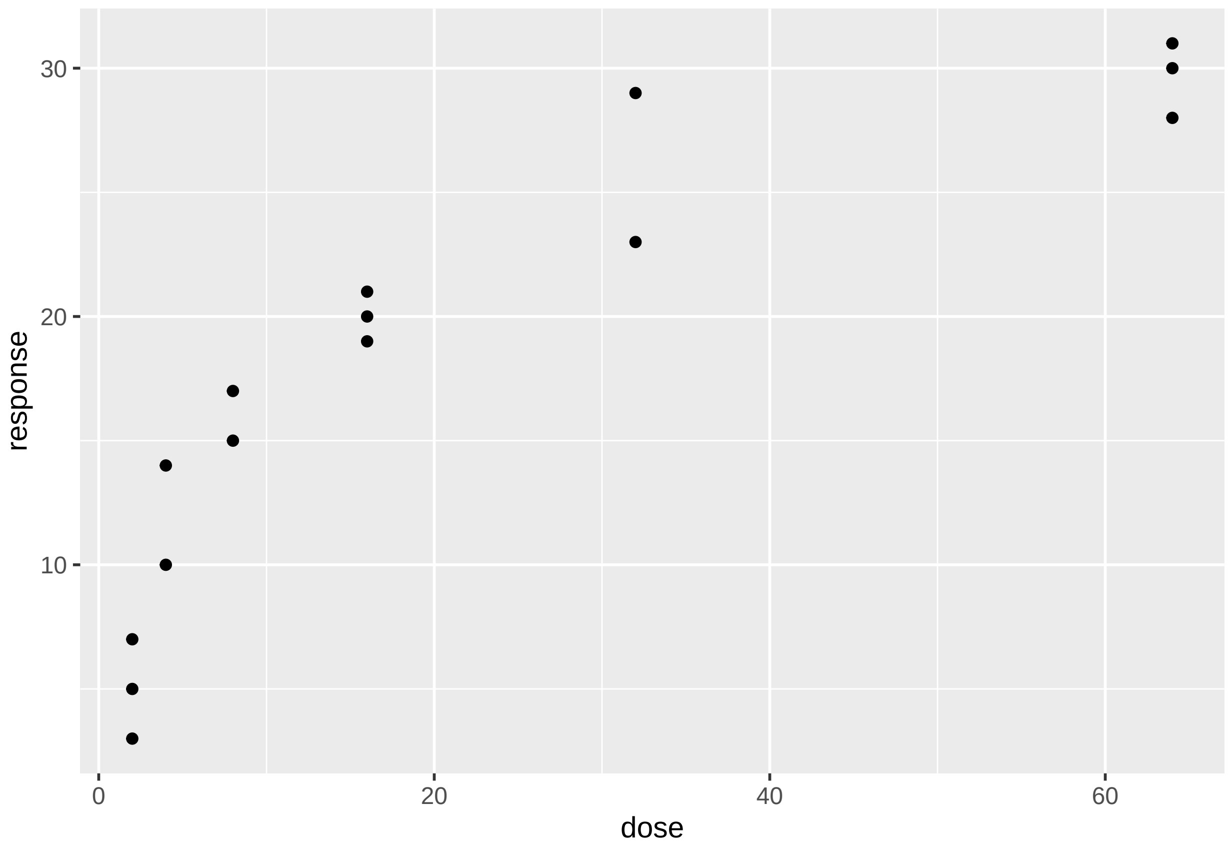


```
# Often I want to create the confidence region myself (perhaps to use a prediction
# interval instead of a confidence interval), and we could use the following code:
p1.data2 <- p1.data %>%
 mutate( yhat = predict(p1.model),
 lwr = predict(p1.model, interval='confidence')[,2],
 upr = predict(p1.model, interval='confidence')[,3] )
# make a nice graph
ggplot(p1.data2, aes(x=x)) +
 geom_ribbon( aes(ymin=lwr, ymax=upr), fill='pink', alpha=.2 ) +
 geom_line( aes( y=yhat), color='green' ) +
 geom_point( aes( y=y ), color='black' )
```

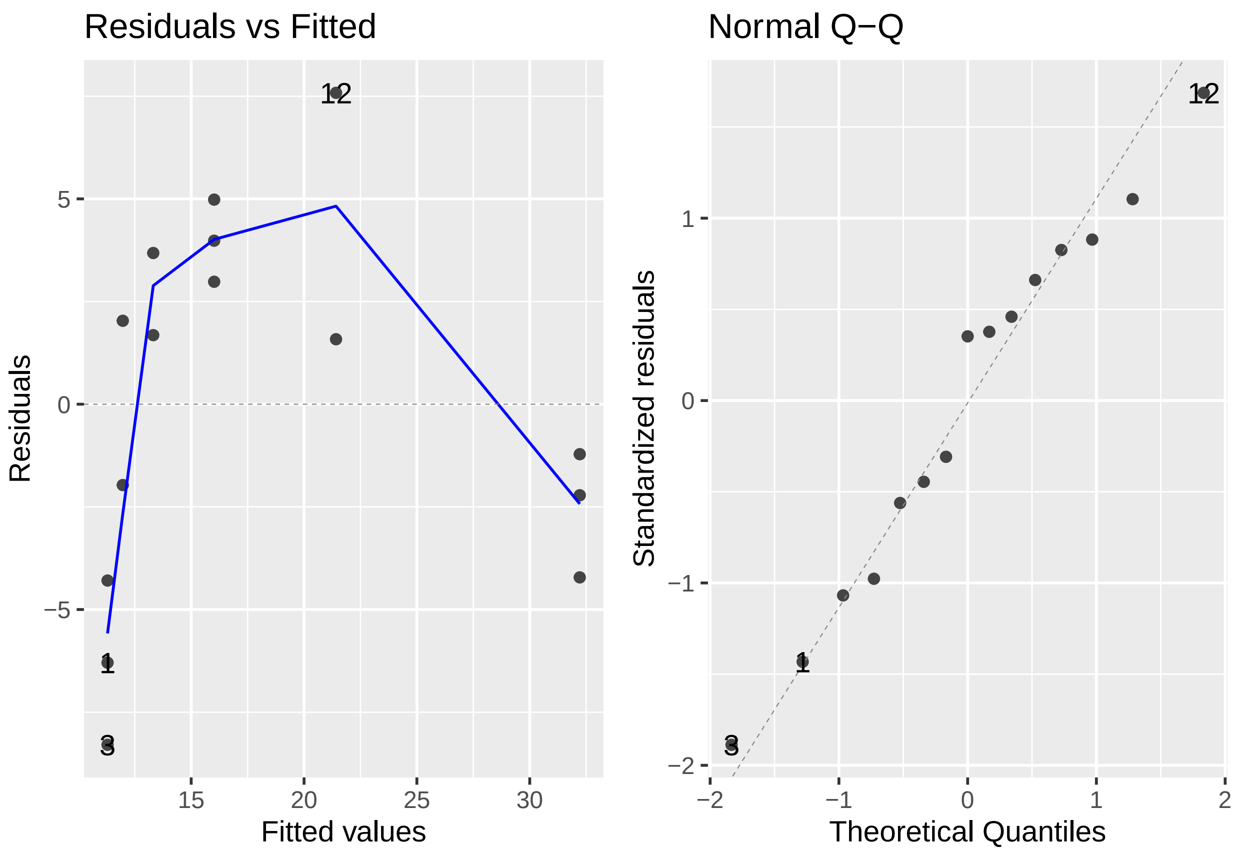


3.a.

```
drug.data <- data.frame(
 dose = c(2,2,2,4,4,8,8,16,16,16,32,32,64,64,64),
 response = c(5,7,3,10,14,15,17,20,21,19,23,29,28,31,30) )
ggplot(drug.data) +
  geom_point(aes(x=dose, y=response))
```

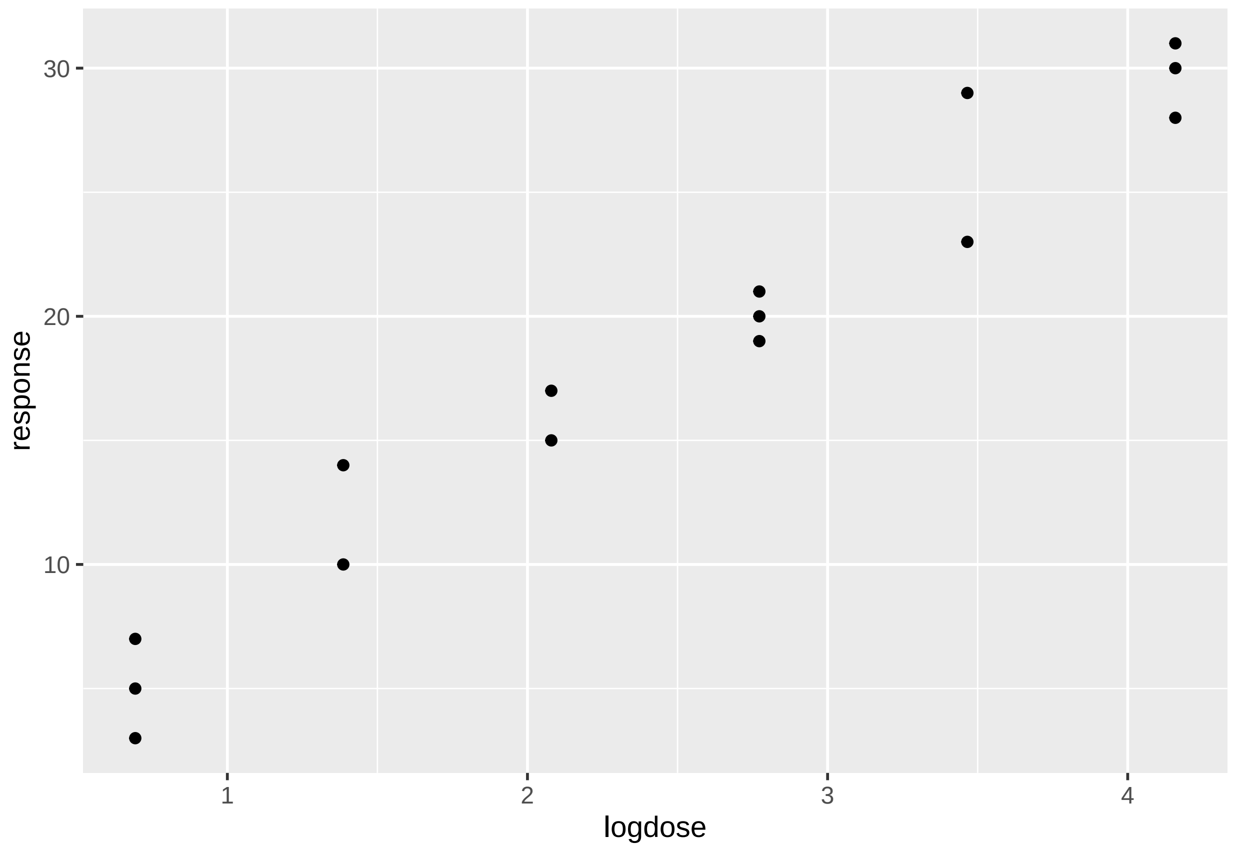


```
drug.model <- lm(response ~ dose, drug.data)
autoplot(drug.model, which=c(1,2))
```



3.b.

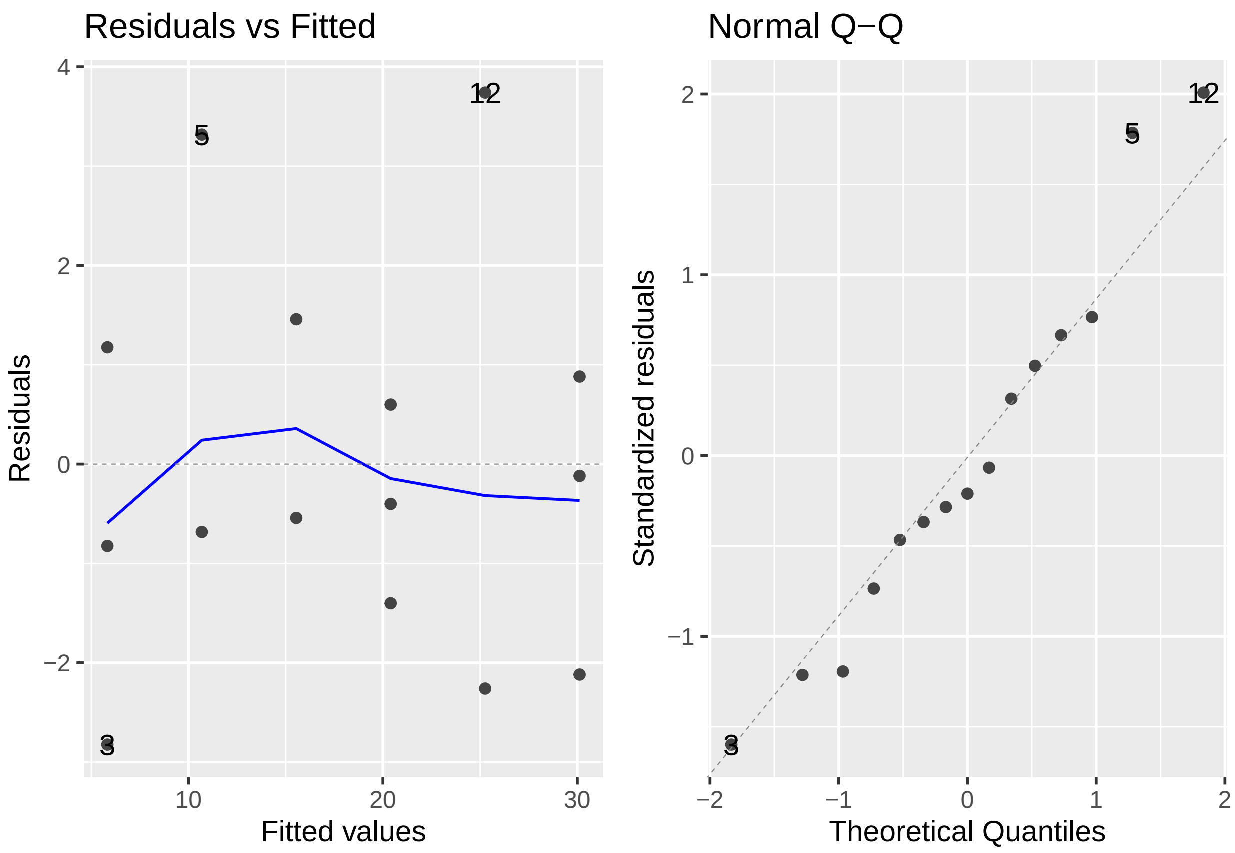
```
# i
drug.data <- drug.data %>% mutate(logdose = log(dose))
ggplot(drug.data) +
  geom_point(aes(x=logdose, y=response))
```



```
# A linear model seems appropriate

# ii
drug.model <- lm(response ~ logdose, drug.data)

# iii
autoplot(drug.model, which=c(1,2))
```



```
# The residuals have far less curvature, so the linear model seems appropriate.

# iv
shapiro.test(resid(drug.model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(drug.model)
## W = 0.95469, p-value = 0.6011
```

```
# The model appears to be Normally Distributed

# v
# slope = 7

# vi
confint(drug.model)
```

```
##                 2.5 %   97.5 %
## (Intercept) -1.481133 3.411301
## logdose      6.118344 7.901590
```