STOR 590 HW5 Solution

Taebin Kim

Page 66 Exercise 3

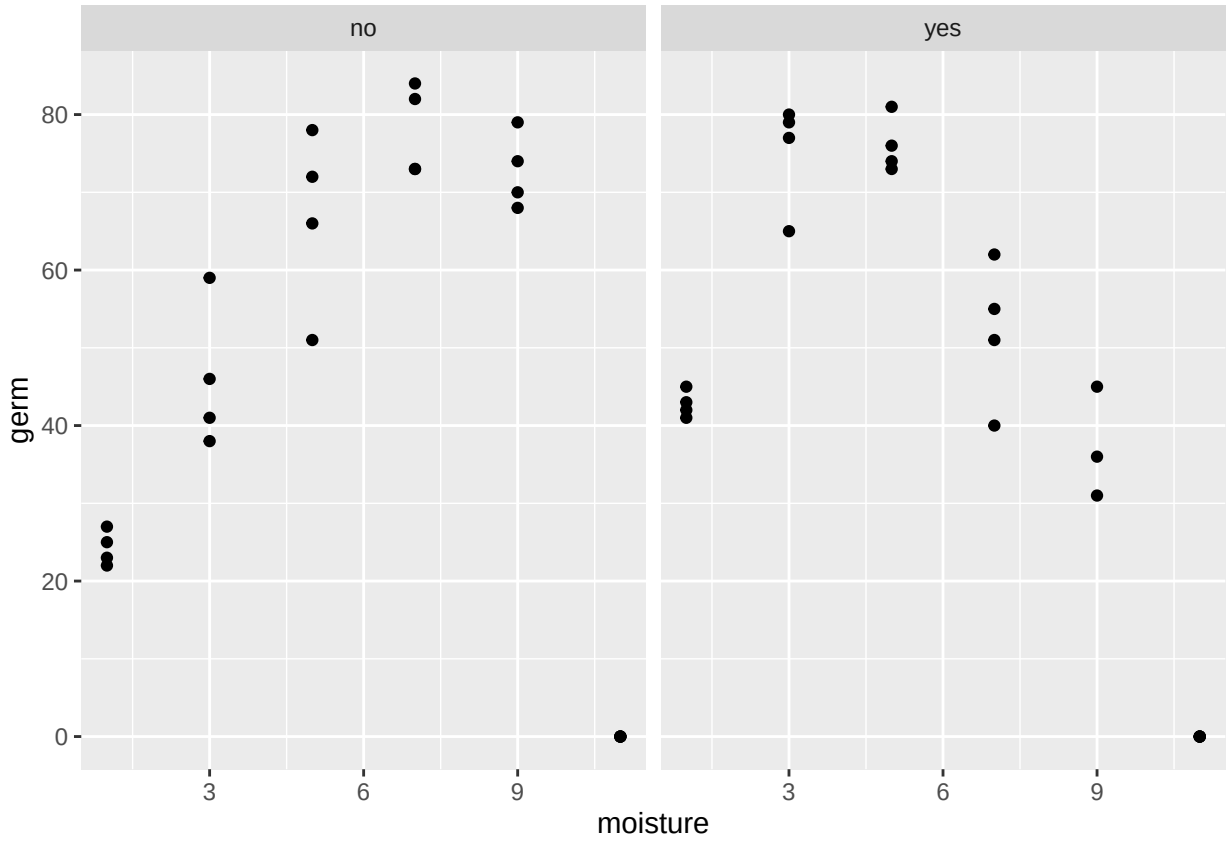
Part (a)

We plot the germination percentage against the moisture level on two side-by-side plots according to the coverage of the box.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

```
data(seeds, package = "faraway")
seeds <- seeds[-47, ] # remove observation 47 which has NA
ggplot(seeds, aes(x = moisture, y = germ)) + geom_point() + facet_grid(~ covered)
```

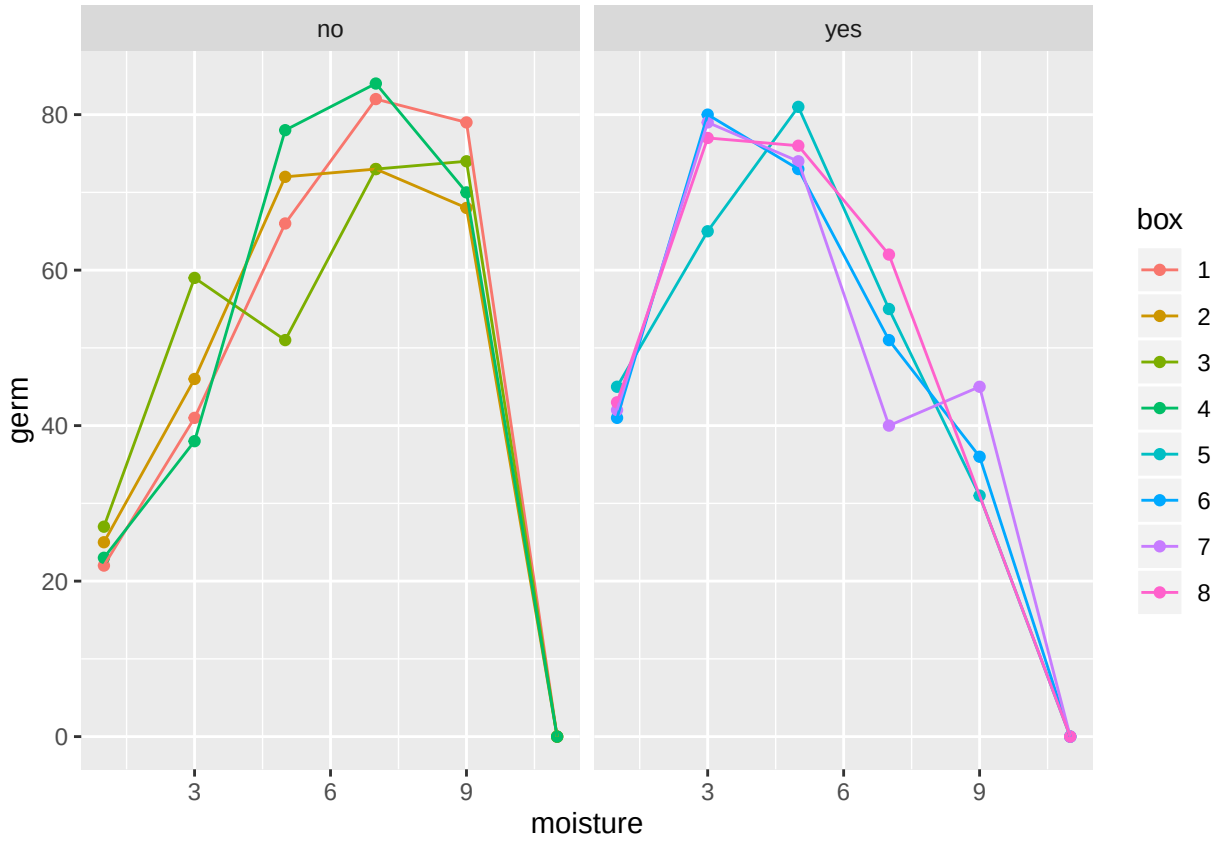


We can observe that as moisture level increases, germination percentage first increases and then decreases for both uncovered boxes and covered boxes.

Part (b)

We create a new factor describing the box and add lines to the previous plot that connects observations from the same box.

```
seeds$box <- gl(8,6)[-47]
ggplot(seeds, aes(x = moisture, y = germ, color = box, group = box)) + geom_point() +
geom_line() + facet_grid(~ covered)
```



There is no clear indication of a box effect.

Part (c)

We fit a binomial response model including the coverage, box and moisture predictors. Since the effect of `moisture` was not linear in (a), we treat the predictor as a factor.

```
glm_1 <- glm(cbind(germ, 100-germ) ~ covered + box + factor(moisture), family=binomial, seeds)
summary(glm_1)
```

```
##
## Call:
## glm(formula = cbind(germ, 100 - germ) ~ covered + box + factor(moisture),
##     family = binomial, data = seeds)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.0511  -1.8750  -0.0003   1.9652   4.4875
##
## Coefficients: (1 not defined because of singularities)
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.68190    0.11575  -5.891 3.83e-09 ***
## coveredyes           0.30967    0.14535   2.130  0.03313 *
## box2                -0.05271    0.13255  -0.398  0.69091
## box3                -0.05271    0.13255  -0.398  0.69091
## box4                 0.02651    0.13295   0.199  0.84193
## box5                -0.42339    0.14492  -2.922  0.00348 **
## box6                -0.38858    0.14503  -2.679  0.00738 **
## box7                -0.39730    0.14500  -2.740  0.00615 **
## box8                      NA         NA      NA       NA
## factor(moisture)3    1.12162    0.10438  10.746  < 2e-16 ***
## factor(moisture)5    1.60530    0.10854  14.790  < 2e-16 ***
## factor(moisture)7    1.30974    0.10562  12.401  < 2e-16 ***
## factor(moisture)9    1.03857    0.10833   9.587  < 2e-16 ***
## factor(moisture)11 -20.62316  905.94969  -0.023  0.98184
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1790.99  on 46  degrees of freedom
## Residual deviance:  314.05  on 34  degrees of freedom
## AIC: 529.79
##
## Number of Fisher Scoring iterations: 16
```

box1 and box8 were considered as the default for uncovered boxes and covered boxes respectively. The plot in part (b) suggests that the model including only the covereage and moisture predictors is appropriate.

Part (d)

We test for the significance of a box effect in your model.

```
glm_s <- glm(cbind(germ, 100-germ) ~ covered + factor(moisture), family=binomial, seeds)
deviance_s <- deviance(glm_s)
deviance_l <- deviance(glm_l)
pchisq(deviance_s - deviance_l, 6, lower = F)
```

```
## [1] 0.07414004
```

We repeat the same test using the Pearson's Chi-squared statistic instead of the deviance.

```
pearson_s <- sum(residuals(glm_s, type = "pearson")^2)
pearson_l <- sum(residuals(glm_l, type = "pearson")^2)
pchisq(pearson_s - pearson_l, 6, lower = F)
```

```
## [1] 0.06299723
```

Since the p-value is larger than 0.05 for both tests, we conclude that a box effect is not significant.

Part (e)

We determine the value of moisture that maximizes the predicted germination.

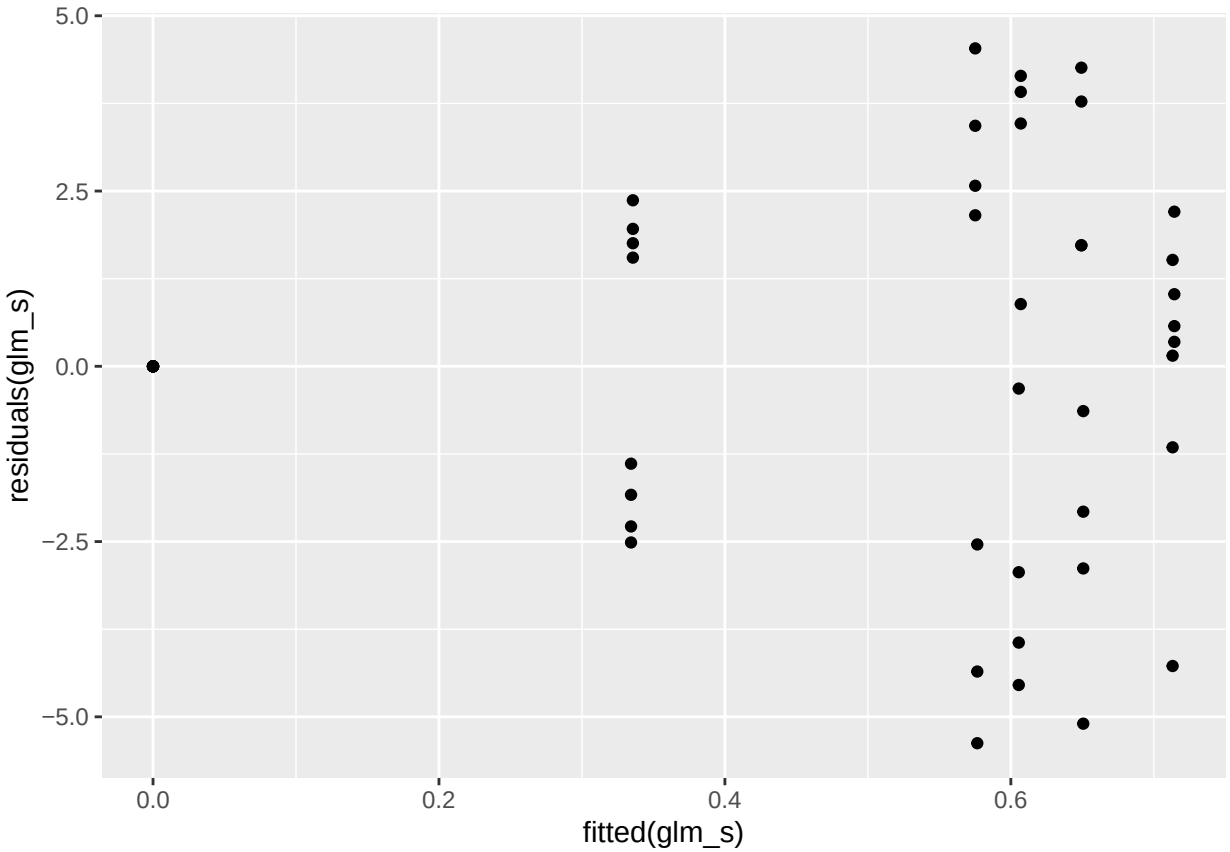
```
seeds$moisture[which.max(fitted(glm_s))]
```

```
## [1] 5
```

Part (f)

We produce a plot of the residuals against the fitted values.

```
ggplot(seeds, aes(x = fitted(glm_s), y = residuals(glm_s))) + geom_point()
```

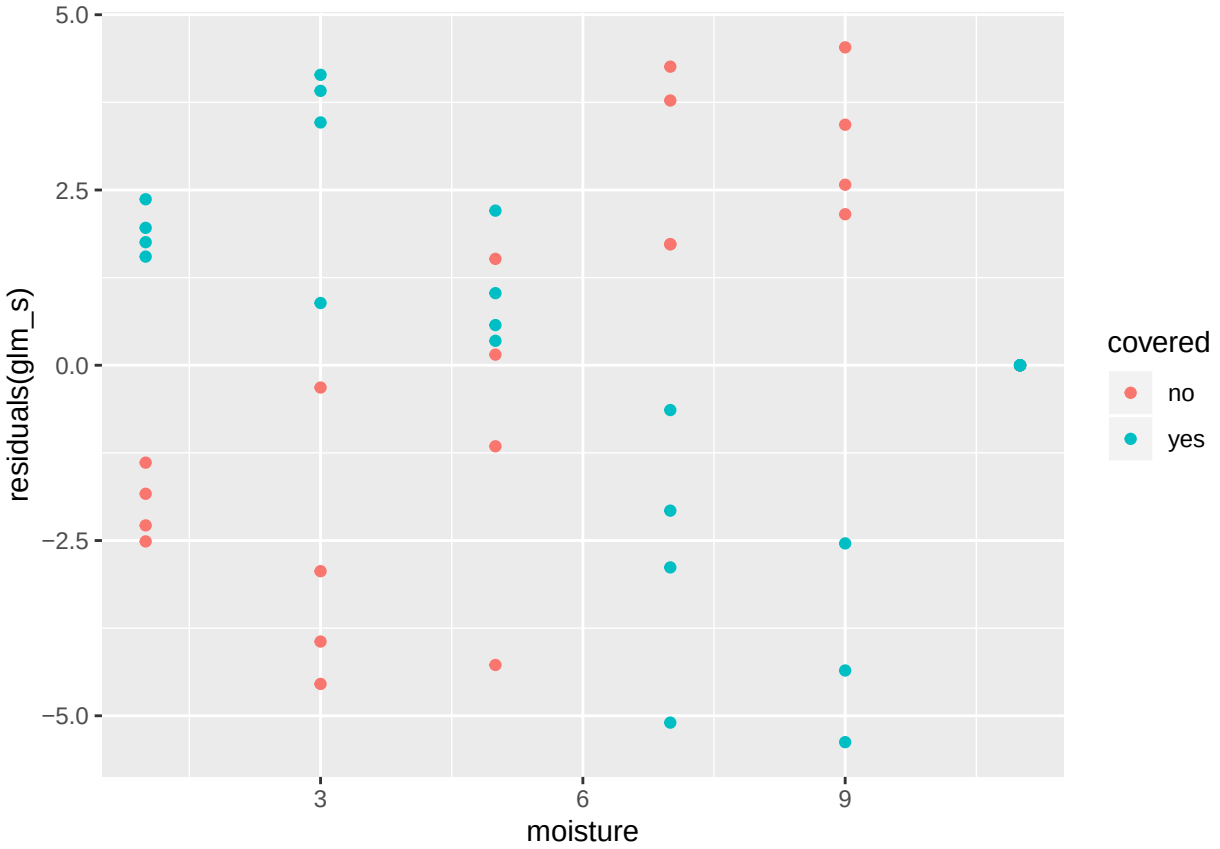


The plot shows a quadratic pattern, implying that our fit is not perfectly reliable.

Part (g)

We plot the residuals against moisture while distinguishing the covering and interpret.

```
ggplot(seeds, aes(x = moisture, y = residuals(glm_s), color = covered)) + geom_point()
```

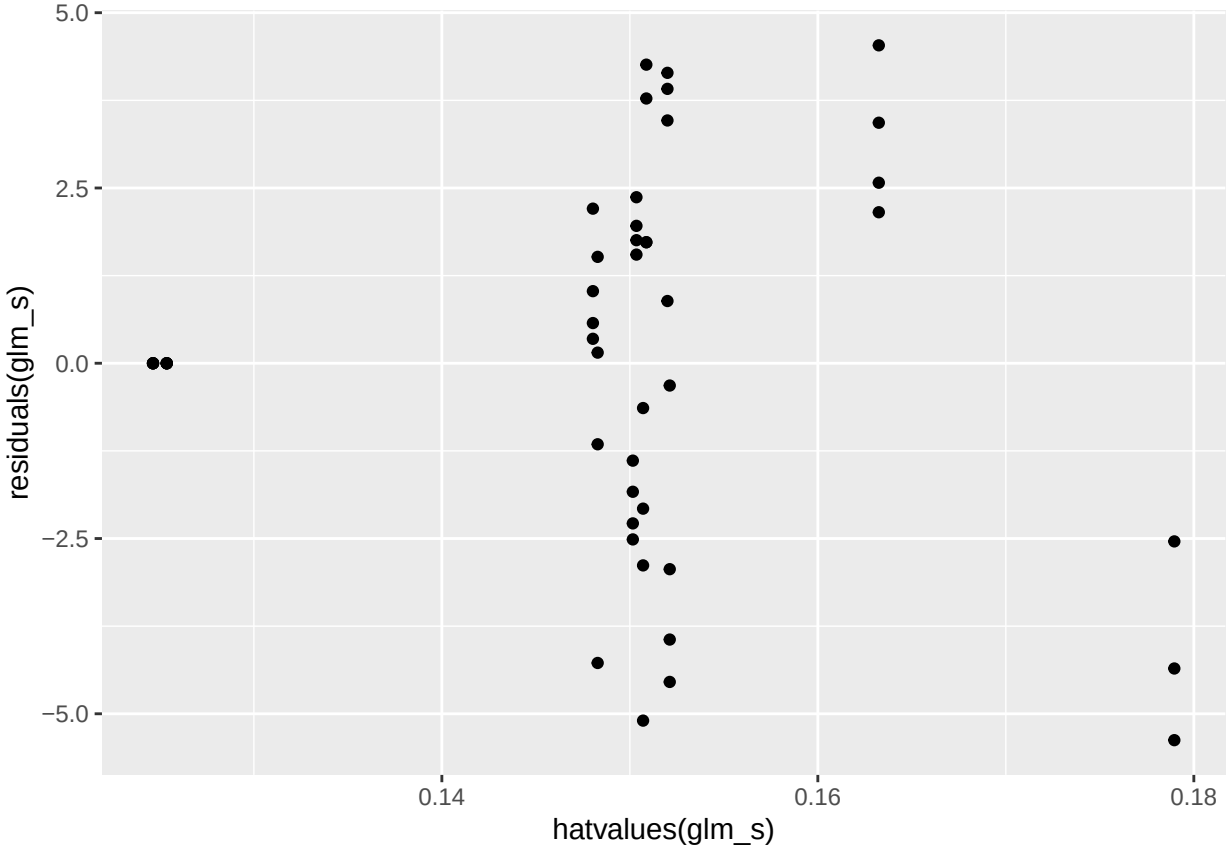


For uncovered boxes, there is an increasing trend of residuals. For covered boxes, it shows a concave downward pattern of residuals. This implies that our model does not fit the data perfectly.

Part (h)

We plot the residuals against the leverages.

```
ggplot(seeds, aes(x = hatvalues(glm_s), y = residuals(glm_s))) + geom_point()
```



There is no sign of influential points.

Page 99 Exercise 2

Part (a)

We plot the data.

```
library(dplyr)
```

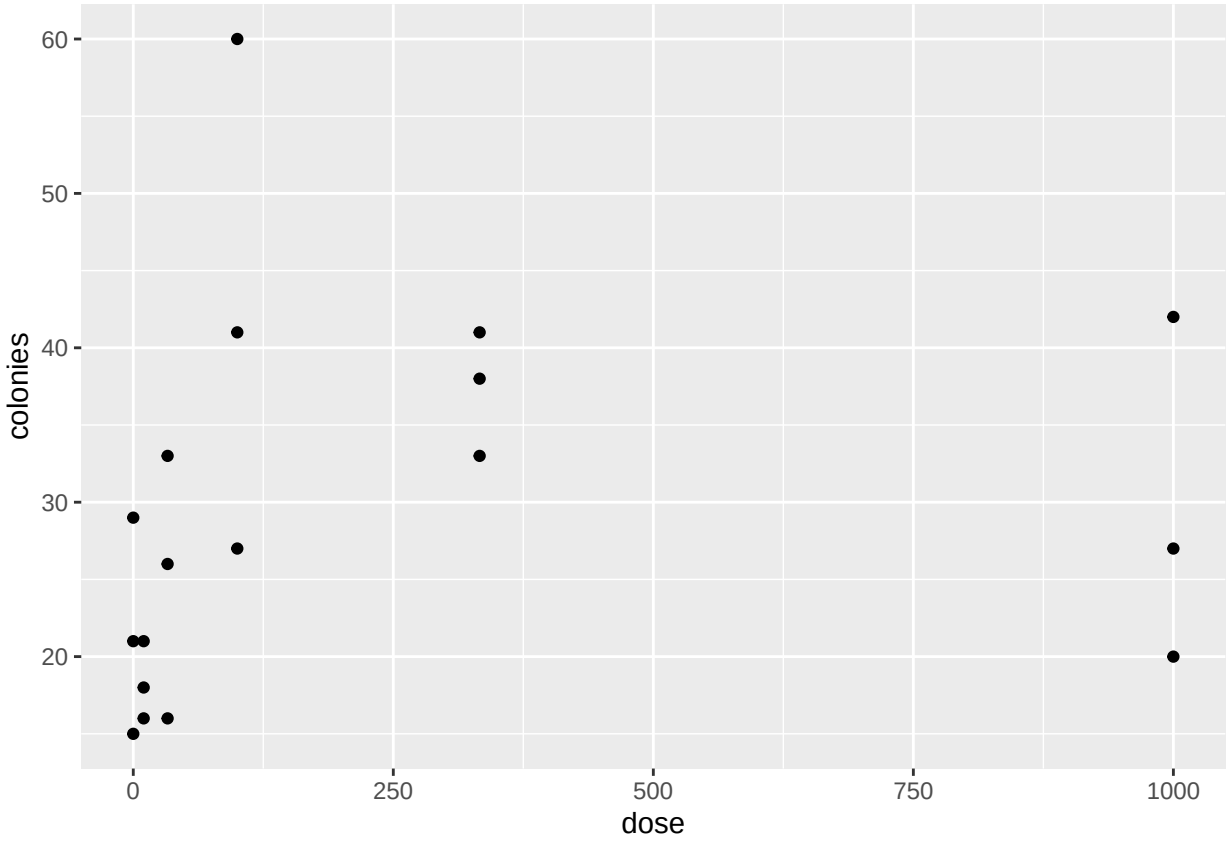
```
## Warning: package 'dplyr' was built under R version 3.6.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data(salmonella, package = "faraway")
ggplot(salmonella, aes(x = dose, y = colonies)) + geom_point()
```



We can observe that there is a positive correlation between the two variables. Note that the variance of `colonies` becomes larger as `dose` increases.

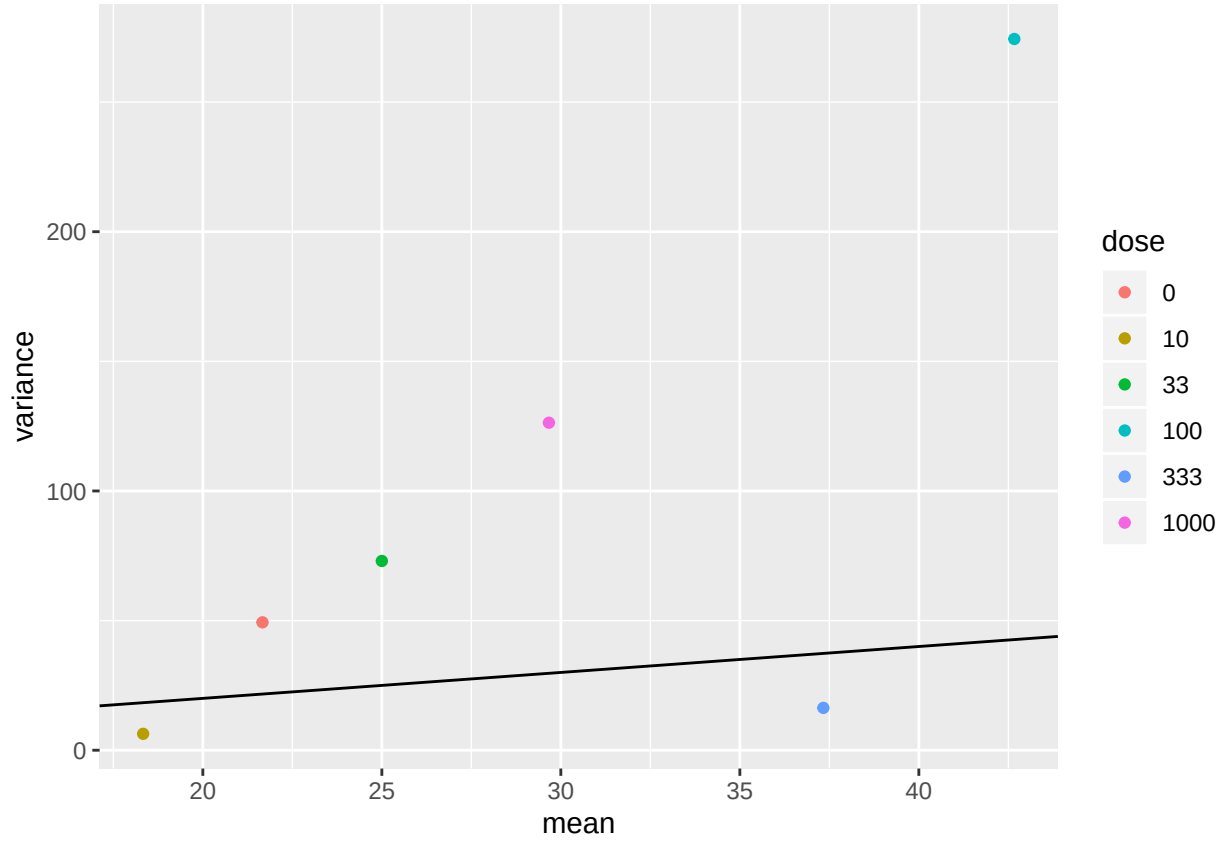
Part (b)

We compute the mean and variance within each set of observations with the same dose. We also plot the variance against the mean.

```
salmonella %>%
group_by(dose) %>%
summarise(mean = mean(colonies),
variance = var(colonies)) %>%
print %>%
ggplot(aes(x = mean, y = variance, color = as.factor(dose))) + geom_point() +
geom_abline(aes(slope = 1, intercept = 0)) + labs(color = "dose")
```

```
## # A tibble: 6 x 3
##    dose  mean variance
##   <int> <dbl>    <dbl>
## 1     0  21.7    49.3
## 2    10  18.3     6.33
```

```
## 3    33  25      73
## 4   100  42.7   274.
## 5   333  37.3    16.3
## 6  1000  29.7   126.
```



The result shows that the ratio of variance to mean is larger than 1 for most of the groups, implying an overdispersion.

Part (c)

We fit a model with dose treated as a six-level factor and check the deviance to determine whether this model fits the data.

```
glm1 <- glm(colonies ~ as.factor(dose), family = poisson, salmonella)
summary(glm1)
```

```
##
## Call:
## glm(formula = colonies ~ as.factor(dose), family = poisson, data = salmonella)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5737  -0.6820  -0.1110   0.6041   2.4989
##
## Coefficients:
```

```
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)           3.0758     0.1240  24.798  < 2e-16 ***
## as.factor(dose)10    -0.1671     0.1832  -0.912 0.361869
## as.factor(dose)33     0.1431     0.1695   0.844 0.398427
## as.factor(dose)100    0.6776     0.1523   4.449 8.62e-06 ***
## as.factor(dose)333    0.5441     0.1559   3.490 0.000484 ***
## as.factor(dose)1000   0.3142     0.1632   1.926 0.054099 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 78.358  on 17  degrees of freedom
## Residual deviance: 33.496  on 12  degrees of freedom
## AIC: 138.03
##
## Number of Fisher Scoring iterations: 4
```

```
pchisq(deviance(glm1), df.residual(glm1), lower = FALSE)
```

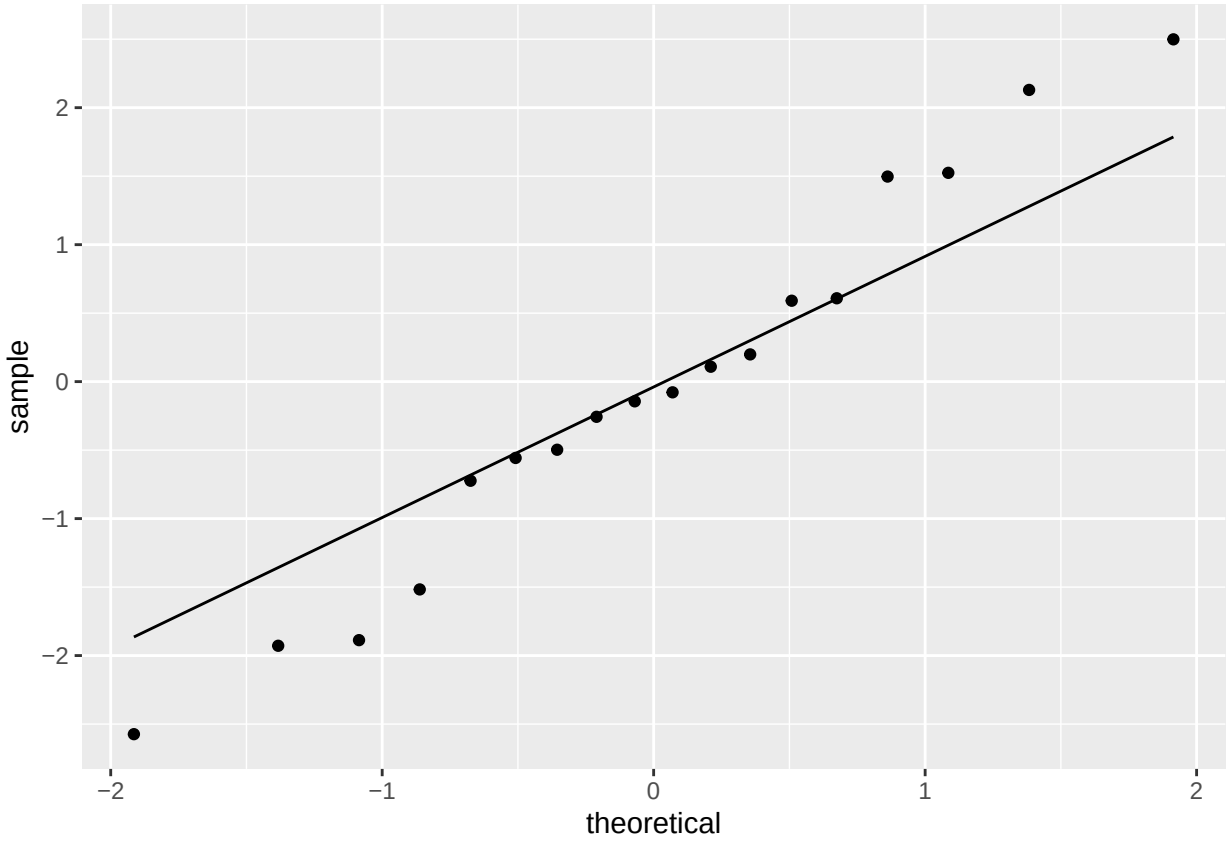
```
## [1] 0.0008096273
```

Since the p-value is so small, this model does not fit the data well. Transforming `dose` will not fix the problem since we have already tried a model with `dose` treated as a factor, which is in the most flexible form.

Part (d)

We make a QQ plot of the residuals from the previous model.

```
ggplot(NULL, aes(sample = residuals(glm1))) + geom_qq() + geom_qq_line()
```



We can see that the residuals does not follow a normal distribution.

Part (e)

We fit a Poisson model that includes an overdispersion parameter and is quadratic in the `dose`.

```
glm2 <- glm(colonies ~ dose + I(dose^2), family = quasipoisson, salmonella)
summary(glm2)
```

```
##
## Call:
## glm(formula = colonies ~ dose + I(dose^2), family = quasipoisson,
##     data = salmonella)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0422  -1.4412  -0.5271   0.8173   4.8797
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.163e+00  1.361e-01  23.237 3.55e-13 ***
## dose         2.507e-03  1.040e-03   2.410   0.0293 *
## I(dose^2)   -2.294e-06  1.003e-06  -2.288   0.0371 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

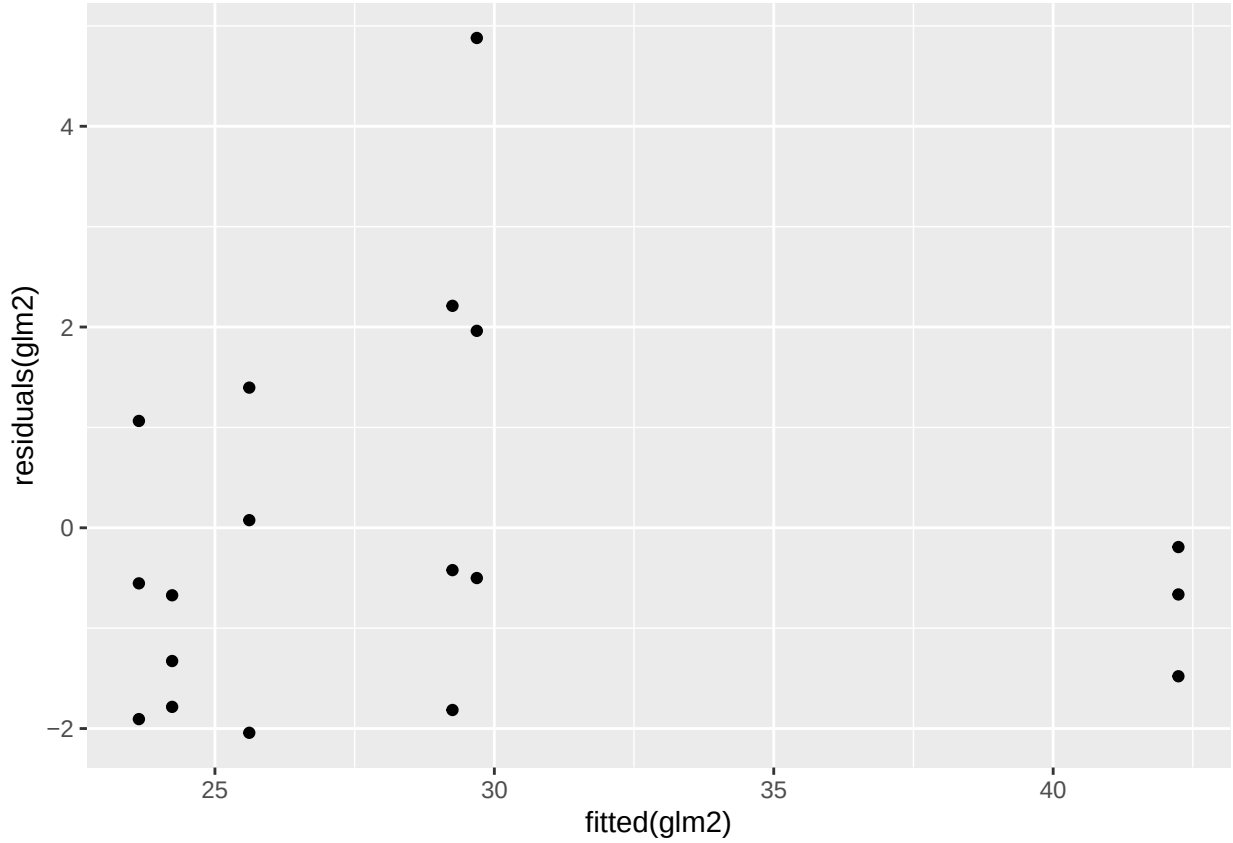
```
##
## (Dispersion parameter for quasipoisson family taken to be 4.126227)
##
##     Null deviance: 78.358  on 17  degrees of freedom
## Residual deviance: 55.535  on 15  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

We cannot determine whether the fit of this model is adequate from the deviance since we have included the overdispersion parameter and the deviance does not follow a chi-squared distribution anymore.

Part (f)

We plot the residuals against the fitted values for the previous model.

```
ggplot(NULL, aes(x = fitted(glm2), y = residuals(glm2))) + geom_point()
```

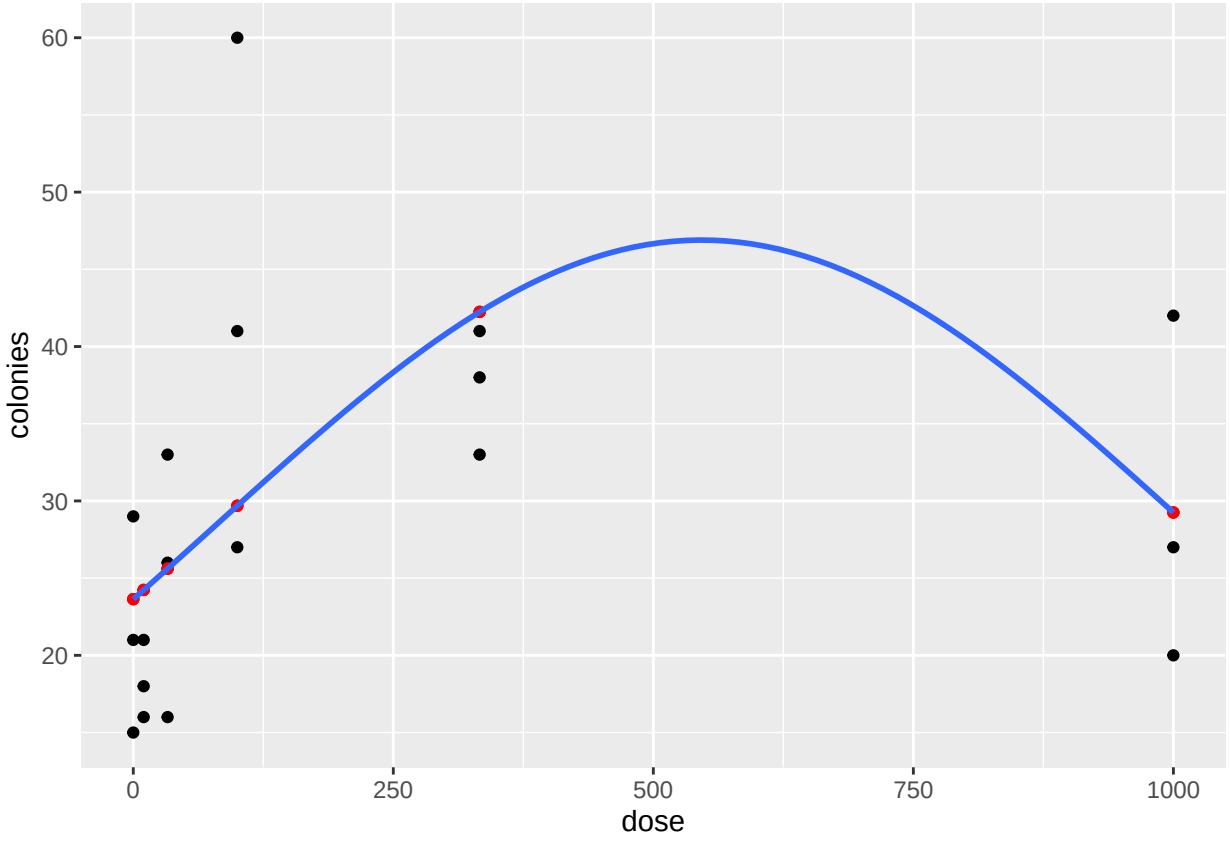


It shows that the residuals are random with constant variance.

Part (g)

We plot the fitted mean response of the model on top of the data.

```
ggplot(salmonella, aes(x = dose, y = colonies)) +
geom_point() +
geom_point(aes(x = dose, fitted(glm2)), color = 'red')+
geom_smooth(method = glm,
formula = y ~ x + I(x^2),
method.args = list(family = quasipoisson),
se = FALSE)
```



Part (h)

We give the predicted mean response for a dose of 500 and compute a 95% confidence interval.

```
pred <- predict(glm2, data.frame(dose = 500), type = "link", se.fit = TRUE)
exp(pred$fit)
```

```
##        1
## 46.66347
```

```
exp(c(pred$fit-1.96*pred$se.fit, pred$fit+1.96*pred$se.fit))
```

```
##        1        1
## 31.14967 69.90377
```

The predicted mean response is 46.6634672 and the 95% confidence interval is [31.1496681, 69.9037679].

Part (i)

We determine the value of `dose` that maximizes the predicted response.

```
salmonella$dose[which.max(fitted(glm2))]
```

```
## [1] 333
```

The predicted response is maximized when `dose` = 333.