R Notebook assignment 6

Experimental Design and Data Analysis: Assignment 6

Fabio Curi Paix?o & Arash Parna

EXERCISE 1

We first load the file needed for this exercise.

```
fruitflies = read.table('fruitflies.txt',header = TRUE)
thorax=as.numeric(fruitflies$thorax)
longevity=as.numeric(fruitflies$longevity)
activity=as.factor(fruitflies$activity)
```

1

The log of the longevity is given as follows:

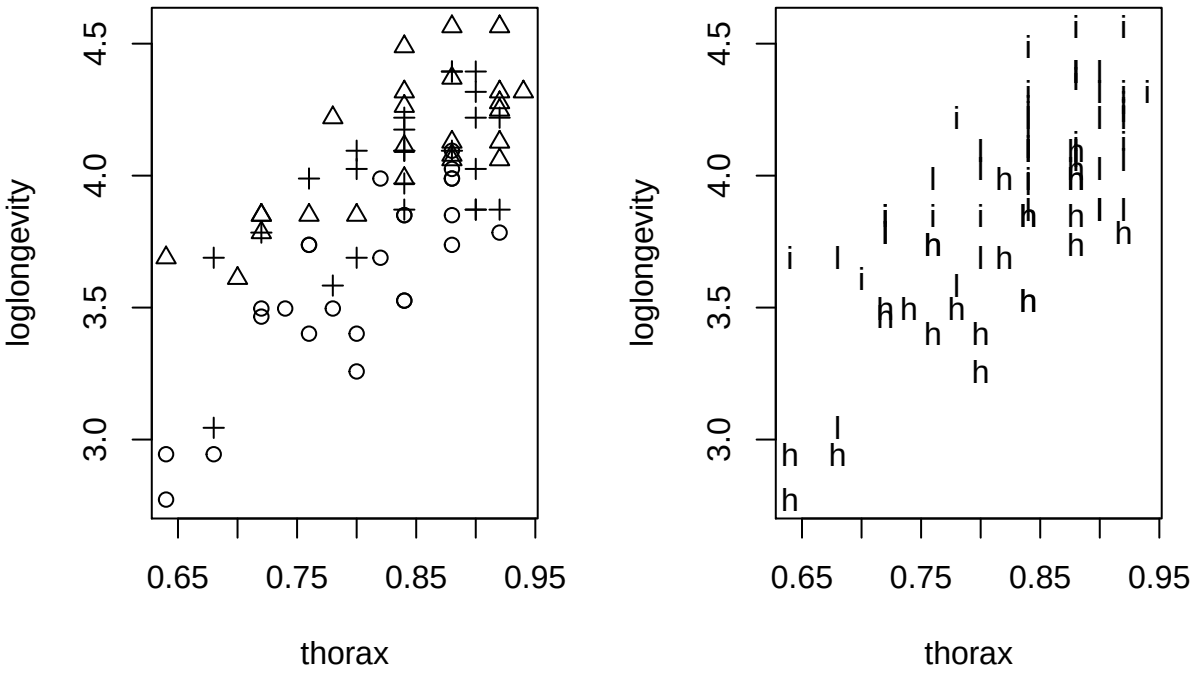
```
fruitflies$loglongevity <- log(as.numeric(fruitflies$longevity))
loglongevity=as.numeric(fruitflies$loglongevity)
```

From now on, the output variable will be "loglongevity", unless specified.

2

Informative plots of the data are given here after.

```
par(mfrow=c(1,2))
plot(loglongevity~thorax,pch=unclass(activity))
plot(loglongevity~thorax,pch=as.character(activity))
```



,
where in the second plot the circles represent "high", triangles are "isolated" and the crosses are "low", which are outputs for "activity".

3

We build the following intercept-free model:

```
fit1 = lm(loglongevity~activity-1,data=fruitflies)
summary(fit1)
```

```
##
## Call:
## lm(formula = loglongevity ~ activity - 1, data = fruitflies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.95531 -0.13338  0.02552  0.20891  0.49222
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## activityhigh      3.60212    0.06145   58.62   <2e-16 ***
## activityisolated  4.11935    0.06145   67.04   <2e-16 ***
## activitylow       3.99984    0.06145   65.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3072 on 72 degrees of freedom
## Multiple R-squared:  0.9941, Adjusted R-squared:  0.9939
## F-statistic:  4056 on 3 and 72 DF,  p-value: < 2.2e-16
```

```
anova(fit1)
```

```
## Analysis of Variance Table
##
```

```
## Response: loglongevity
##           Df Sum Sq Mean Sq F value    Pr(>F)
## activity   3 1148.6  382.86  4055.8 < 2.2e-16 ***
## Residuals 72    6.8    0.09
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fit1
```

```
##
## Call:
## lm(formula = loglongevity ~ activity - 1, data = fruitflies)
##
## Coefficients:
##     activityhigh  activityisolated       activitylow
##            3.602             4.119             4.000
```

By the results obtained from the intercept and the coefficients for isolated and low activities, it is safe to say that sexual activity influences loglongevity. Furthermore, the values in the last column of the summary show very low values, which confirm what has just been said.

4

The final model is given as follows:

loglongevity = 3.60212*activityhigh* + *4.11935*activityisolated + 3.99984*activitylow + error

With these positive coefficients, it is safe to state that sexual activity increase loglongevity for the three cases. The estimated loglongevities in days for the three conditions are the following:

loglongevity(high) = 3.60212*1* + *4.11935*0 + 3.99984*0 + error = 3.60212 + error

loglongevity(isolated) = 3.60212*0* + *4.11935*1 + 3.99984*0 + error = 4.11935 + error

loglongevity(low) = 3.60212*0* + *4.11935*0 + 3.99984*1 + error = 3.99984 + error

5

Now, we will build the model with both explanatory variables.

```
fit2= lm(loglongevity~activity-1+thorax,data=fruitflies)
summary(fit2)
```

```
##
## Call:
## lm(formula = loglongevity ~ activity - 1 + thorax, data = fruitflies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4858 -0.1612  0.0104  0.1510  0.3574
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## activityhigh        1.2189     0.2486   4.902 5.79e-06 ***
## activityisolated    1.6289     0.2595   6.276 2.42e-08 ***
## activitylow         1.5046     0.2600   5.786 1.79e-07 ***
## thorax              2.9790     0.3067   9.715 1.14e-14 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2027 on 71 degrees of freedom
## Multiple R-squared:  0.9975, Adjusted R-squared:  0.9973
## F-statistic:  7010 on 4 and 71 DF,  p-value: < 2.2e-16
```

The results of this new model induce us to think differently. The coefficients for the different sexual activities lowered, while the one for the thorax length has the highest coefficient. From the values in the last column, which are very low, it is still safe to state that both variables are influential in the model.

6

From this new model, sexual activity appears to decrease loglongevity. The final model is the following:

loglongevity = 1.2189$activityhigh$ + 1.6289activityisolated + 1.5046$activitylow$ + 2.9790thorax + error

The average and smallest thorax length in the dataset are the following:

```
mean(thorax)
```

```
## [1] 0.8245333
```

```
min(thorax)
```

```
## [1] 0.64
```

The loglongevity in days for a fly with average thorax length is the following:

loglongevity(high)= 1.2189$1$ + 1.6289$0$ + 1.5046$0$ + 2.9790$0.82$ + error = 3.66 + error

loglongevity(isolated)= 1.2189$0$ + 1.6289$1$ + 1.5046$0$ + 2.9790$0.82$ + error = 4.07 + error

loglongevity(low)= 1.2189$0$ + 1.6289$0$ + 1.5046$1$ + 2.9790$0.82$ + error = 3.95 + error

And for a fly with the smallest thorax length:

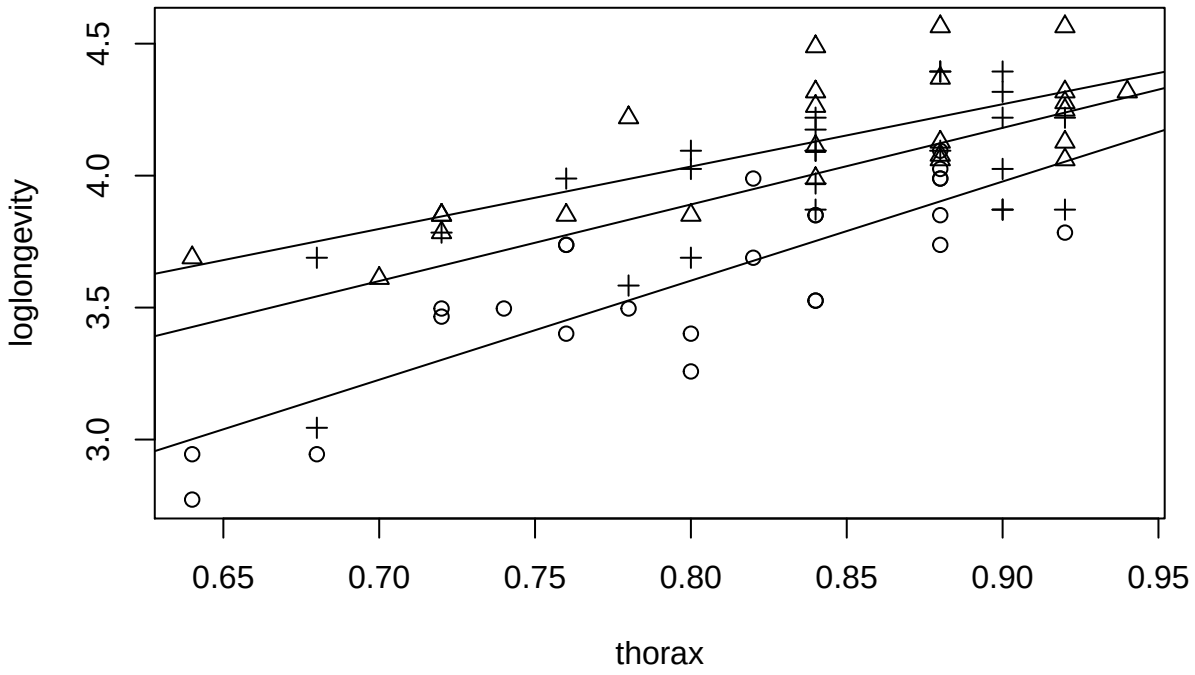
loglongevity(high)= 1.2189$1$ + 1.6289$0$ + 1.5046$0$ + 2.9790$0.64$ + error = 3.12 + error

loglongevity(isolated)= 1.2189$0$ + 1.6289$1$ + 1.5046$0$ + 2.9790$0.64$ + error = 3.54 + error

loglongevity(low)= 1.2189$0$ + 1.6289$0$ + 1.5046$1$ + 2.9790$0.64$ + error = 3.41 + error

7

```
par(mfrow=c(1,1))
plot(loglongevity~thorax,pch=unclass(activity))
abline(lm(loglongevity~thorax,data=fruitflies[fruitflies$activity=='isolated',]))
abline(lm(loglongevity~thorax,data=fruitflies[fruitflies$activity=='low',]))
abline(lm(loglongevity~thorax,data=fruitflies[fruitflies$activity=='high',]))
```



,

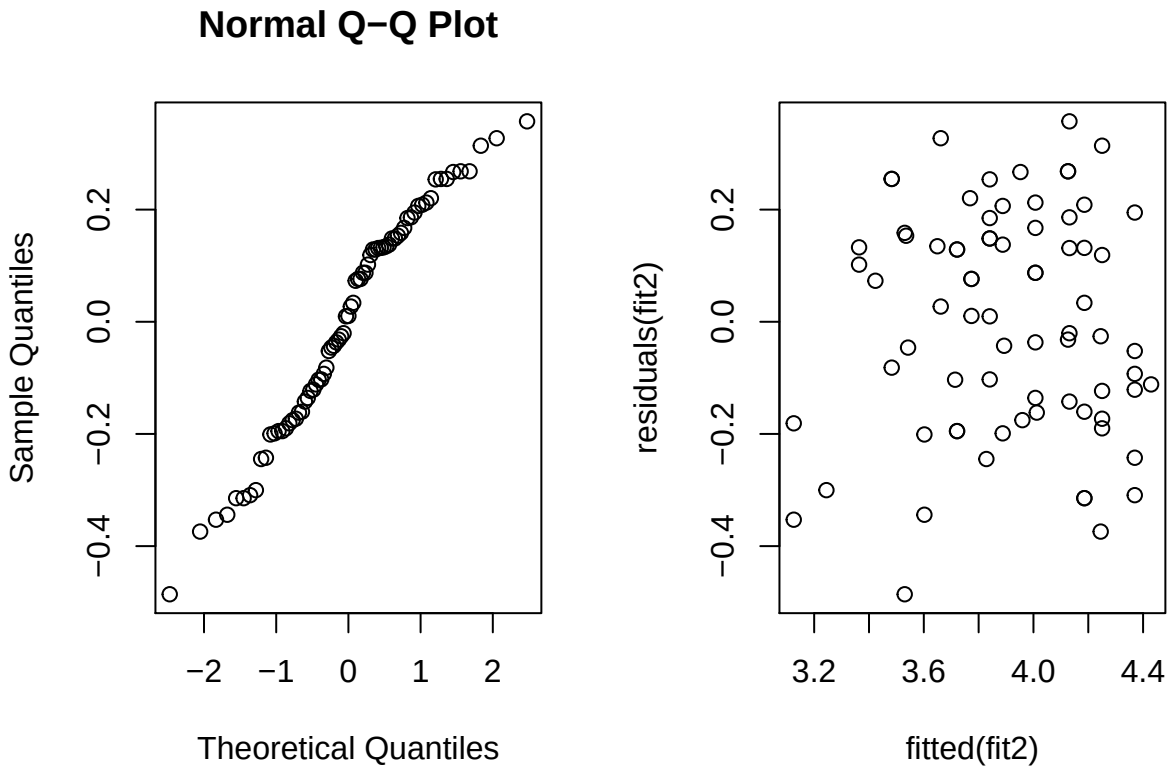
where the abline plots are, from up to bottom, fitting the isolated, low and high sexual activities. Thus, when considering the thorax length, the plot shows three upward linear behaviours for the three cases. Thus, they have simmilar behaviours, however the three conditions have higher abline fit values from isolated towards high sexual activities, passing through low sexual activity. This fact confirms the numerical results obtained in the previous subsection.

8

The second model, considering the thorax length, seems to be more appropriate to the context of the exercise. There are two different ways of seeing this experiment: the first, only the sexual activity is considered, which has in turn always a positive influence towards loglongevity. When the thorax length is added to the model, these three scenarios change their coefficients in the model. We believe that considering the two explanatory variables is the best decision as we remain faithful to the original dataset, whilst respecting the influential aspect of the thorax length. At last, this decision is supported by the fact that the second model showed that the two explanatory variables are significant (see subsection 5). Thus, the first analysis is by us considered inappropriate.

9

```
par(mfrow=c(1,2))
qqnorm(residuals(fit2))
plot(fitted(fit2),residuals(fit2))
```



The results of the first plot, which shows the QQ-plot of the residuals, present normal distribution as the plot follows a rather straight line. Regarding heteroscedasticity, when there is a completely random and equal distribution of points throughout the range of X axis, it does not exist. Visually looking at the plot on the left, we could state that there is little heteroscedasticity since the points distribution around X seem to be rather equal. We will check this statement by the Breush-Pagan test:

```
lmtest::bptest(fit2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fit2
## BP = 2.5333, df = 3, p-value = 0.4693
```

This test gave a p-value higher than the significance level of 0.05, therefore we can not reject the null hypothesis that the variance of the residuals is constant.

10

Now, we will make the same approach without the logarithm.

```
fit3= lm(longevity~activity-1+thorax,data=fruitflies)
summary(fit3)
```

```
##
## Call:
## lm(formula = longevity ~ activity - 1 + thorax, data = fruitflies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.688  -8.622  -1.176   6.790  26.605
```

```
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## activityhigh       -67.37      12.75  -5.284 1.33e-06 ***
## activityisolated   -47.31      13.31  -3.555 0.000678 ***
## activitylow        -54.32      13.33  -4.074 0.000119 ***
## thorax             132.62      15.72   8.434 2.62e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.4 on 71 degrees of freedom
## Multiple R-squared:  0.9673, Adjusted R-squared:  0.9654
## F-statistic: 524.4 on 4 and 71 DF,  p-value: < 2.2e-16
```

```
anova(fit3)
```

```
## Analysis of Variance Table
##
## Response: longevity
##           Df Sum Sq Mean Sq F value    Pr(>F)
## activity   3 219020   73007 675.548 < 2.2e-16 ***
## thorax     1   7687    7687  71.127 2.624e-12 ***
## Residuals 71   7673     108
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now, we will check for normality and heteroscedasticity.

```
lmtest::bptest(fit3)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fit3
## BP = 10.516, df = 3, p-value = 0.01465
```

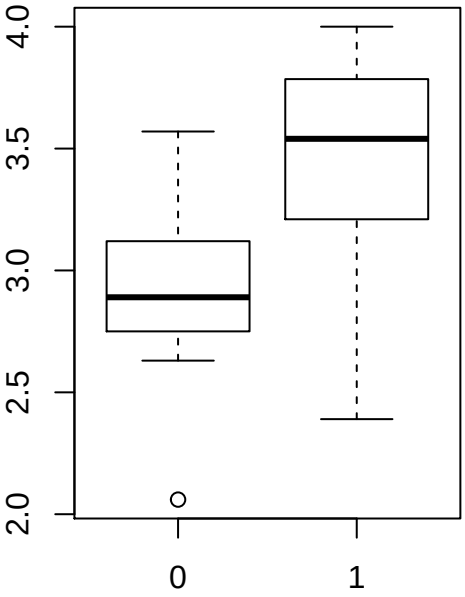
This test has a p-value less that a significance level of 0.05, therefore we can reject the null hypothesis that the variance of the residuals is constant and infer that heteroscedasticity is indeed present. Finally, as heteroscedasticity is not desired, the decision of taking the logarithm was not wise.

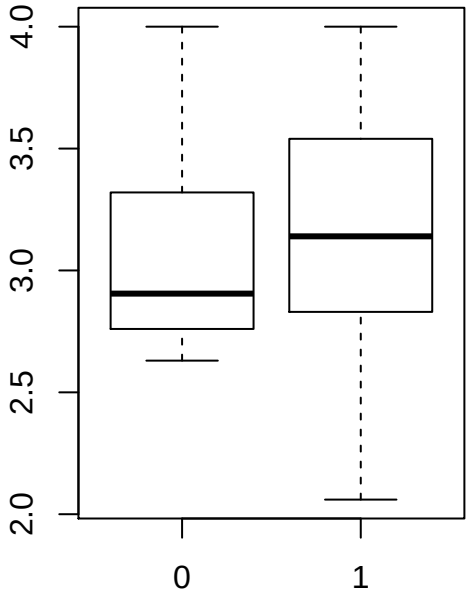
EXERCISE 2

```
psi_table = read.table('psi.txt',header = TRUE)
passed = as.numeric(psi_table$passed)
gpa = as.numeric(psi_table$gpa)
psi = as.numeric(psi_table$psi)
```

1

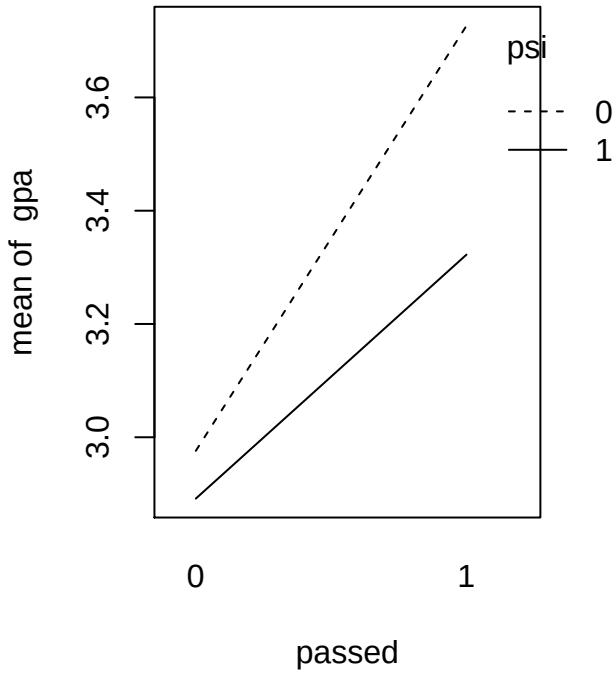
```
par(mfrow=c(1,2))
boxplot(gpa~passed)
boxplot(gpa~psi)
```

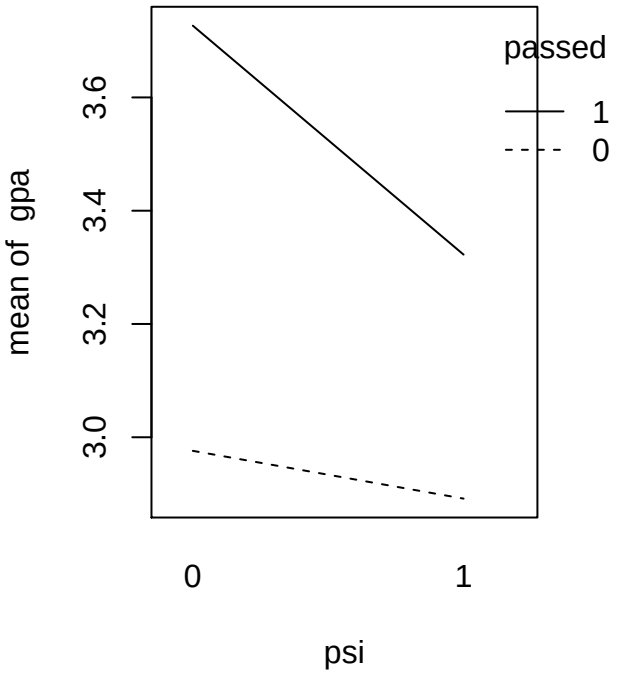




We can observe here that the "gpa" was considerably higher for those who passed the exam, which is expected. Now, a more interesting analysis is done, which indicates that the median in the boxplot for those who obtained the "psi" was higher. Students who obtained training with or without this method managed, nevertheless, to reach the maximum score of 4. Furthermore, within the group of students who were offered the "psi" training, it was observed the lower overall score ("gpa" around 2.1).

```
par(mfrow=c(1,2))
interaction.plot(passed,psi,gpa)
interaction.plot(psi,passed,gpa)
```





These plots are very interesting because they show the influence of the psi methodology on the students' overall score in the assignment. For those having been under the "psi" training, the mean of "gpa" is always observed to be lower than the mean of those who have not been trained with psi. Furthermore, among those who passed the assignment, the mean of "gpa" is lower for those who have had the psi training. The same stands for those who have not passed the assignment.

2

Here we fit a logistic model for this example.

```
psi_table$gpa2=gpa^2
psiglm=glm(passed~gpa+gpa2+psi,data=psi_table,family=binomial)
summary(psiglm)
```

```
##
## Call:
## glm(formula = passed ~ gpa + gpa2 + psi, family = binomial, data = psi_table)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8183  -0.5737  -0.2500   0.4733   2.1584
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   14.050     16.462   0.854   0.3934
## gpa          -13.893     11.048  -1.258   0.2086
## gpa2           2.730      1.830   1.492   0.1356
## psi            2.520      1.179   2.137   0.0326 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 41.183  on 31  degrees of freedom
## Residual deviance: 24.007  on 28  degrees of freedom
## AIC: 32.007
##
## Number of Fisher Scoring iterations: 5
```

```
psiglm2=glm(passed~gpa+psi,data=psi_table,family=binomial)
summary(psiglm2)
```

```
##
## Call:
## glm(formula = passed ~ gpa + psi, family = binomial, data = psi_table)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8396  -0.6282  -0.3045   0.5629   2.0378
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -11.602      4.213  -2.754  0.00589 **
## gpa            3.063      1.223   2.505  0.01224 *
## psi            2.338      1.041   2.246  0.02470 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 41.183  on 31  degrees of freedom
```

```
## Residual deviance: 26.253  on 29  degrees of freedom
## AIC: 32.253
##
## Number of Fisher Scoring iterations: 5
```

The 2 explanatory variables are inserted as numerical variables. The positive signs of the parameter estimates mean that higher values of these variables give higher probability that the psi method was applied. A very interesting observation is that, in general, for higher values of gpa“, the lower is the probability that the”psi" method was applied.

3

Now we can conclude from the two previous subsections that the “psi” method does work with regards to being efficient in enabling the students to pass the exam, however their grades are not as good as those who have not been submitted the “psi” methodology, in general.

4

The probability that a student with a “gpa” equal to 3 who receives “psi” passes the assignment is given as follows:

```
psiglm=glm(passed~gpa+psi,data=psi_table,family=binomial)
gpa3passed=data.frame(psi=as.numeric(1),gpa=3)
predict.glm(psiglm,gpa3passed,type="response")
```

```
##         1
## 0.4815864
```

And for those who have not received “psi”:

```
gpa3fail=data.frame(psi=as.numeric(0),gpa=3)
predict.glm(psiglm,gpa3fail,type="response")
```

```
##          1
## 0.08230274
```

Thus, this confirms that “psi” is a good methodology to make students pass. Students with a “gpa” of 3 who received psi" are at least 6 times more likely to pass the assignment.

5

```
psiglm=glm(passed~psi,data=psi_table,family=binomial)

gpa3passed=data.frame(psi=as.numeric(1))
pass = predict.glm(psiglm,gpa3passed,type="response")
pass
```

```
##         1
## 0.5714286
```

```
gpa3fail=data.frame(psi=as.numeric(0))
fail = predict.glm(psiglm,gpa3fail,type="response")
fail
```

```
##         1
## 0.1666667
```

These final values are not dependent on "gpa" and show higher probability values for each case, in comparison to the last subsection.

6

```
x=matrix(c(3,15,8,6),2,2)
x
```

```
##      [,1] [,2]
## [1,]    3    8
## [2,]   15    6
```

```
fisher.test(x)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  x
## p-value = 0.0265
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.02016297 0.95505763
## sample estimates:
## odds ratio
##  0.1605805
```

The Fisher test's null hypotesis that students in group p1 have the same probibility as students in p2 is 0.0265, which is lower than 0.05 and thus rejected. The conclusion is that there is a different between the students who did recive "psi" and those who did not .

7

Although both experiments produce the same conclusion, the Fisher test experiment does not consider the "gpa" of the participants. The second approach only tests the effectivness of "psi" without considering other attributes such as "gpa", which is not necessarily wrong.

8

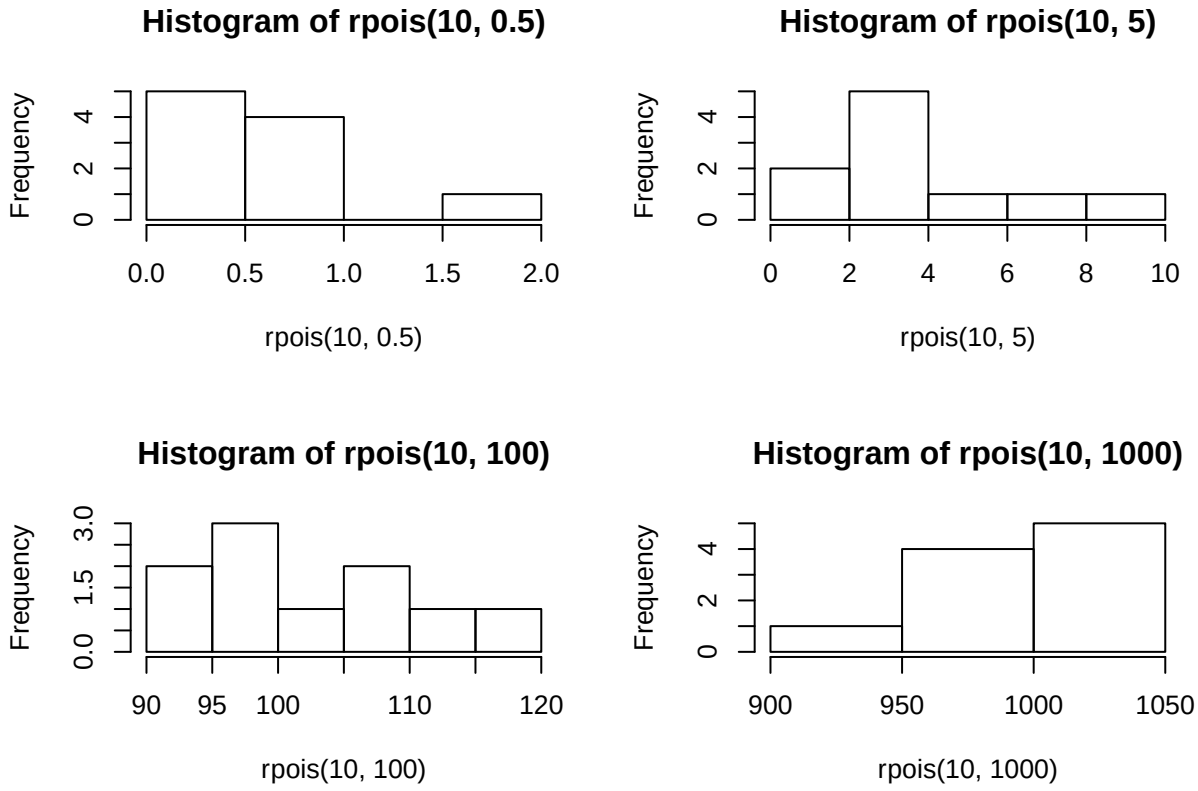
The advantage of using the first method is that if the data set has more than 2 rows and columns we can use the first method, while the second method needs a 2x2 grid. The first method tests the dependency between response variable and other variables, while the second only shows if there is a difference between 2 kinds of classification and is often used in small sample sizes. Finally, the second method is not used for prediction while method 1 can be used for prediction.

EXERCISE 3

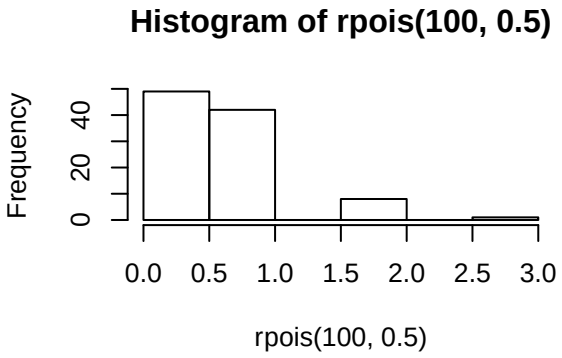
```
africa = read.table('africa.txt',header = TRUE)
```

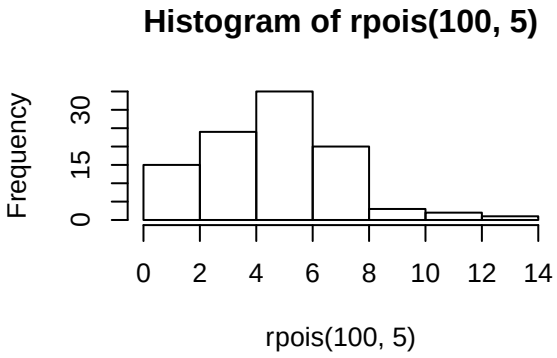
1

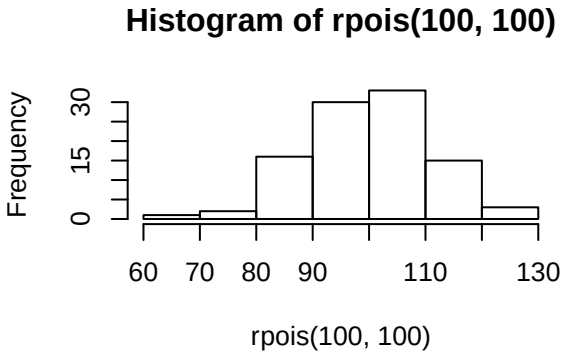
```
par(mfrow=c(2,2))
hist(rpois(10,0.5))
hist(rpois(10,5))
hist(rpois(10,100))
hist(rpois(10,1000))
```

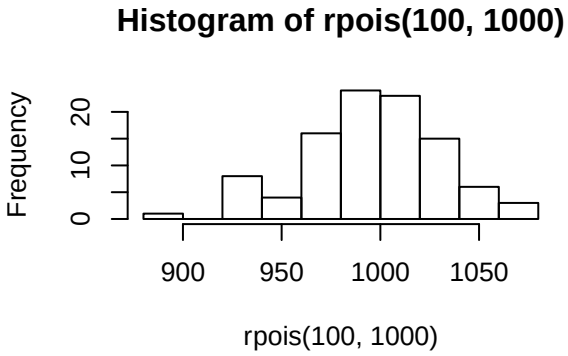



```
par(mfrow=c(2,2))
hist(rpois(100,0.5))
hist(rpois(100,5))
hist(rpois(100,100))
hist(rpois(100,1000))
```

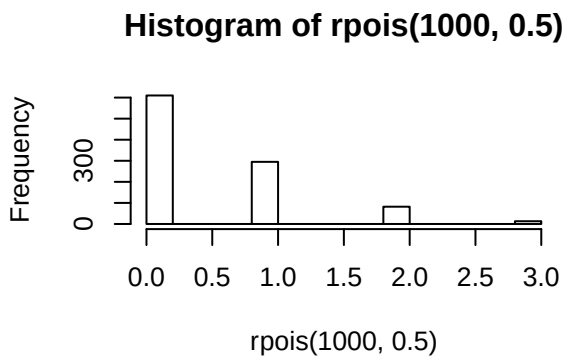


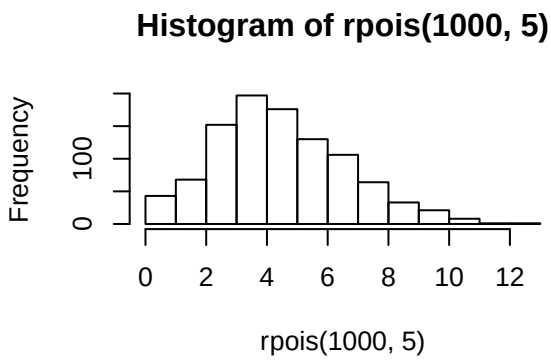


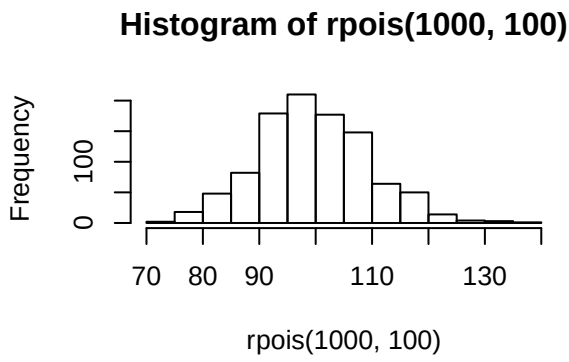


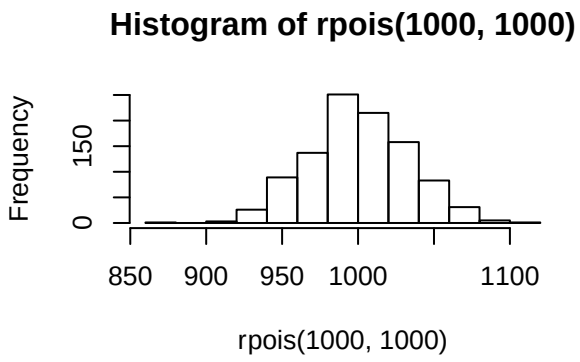



```
par(mfrow=c(2,2))
hist(rpois(1000,0.5))
hist(rpois(1000,5))
hist(rpois(1000,100))
hist(rpois(1000,1000))
```

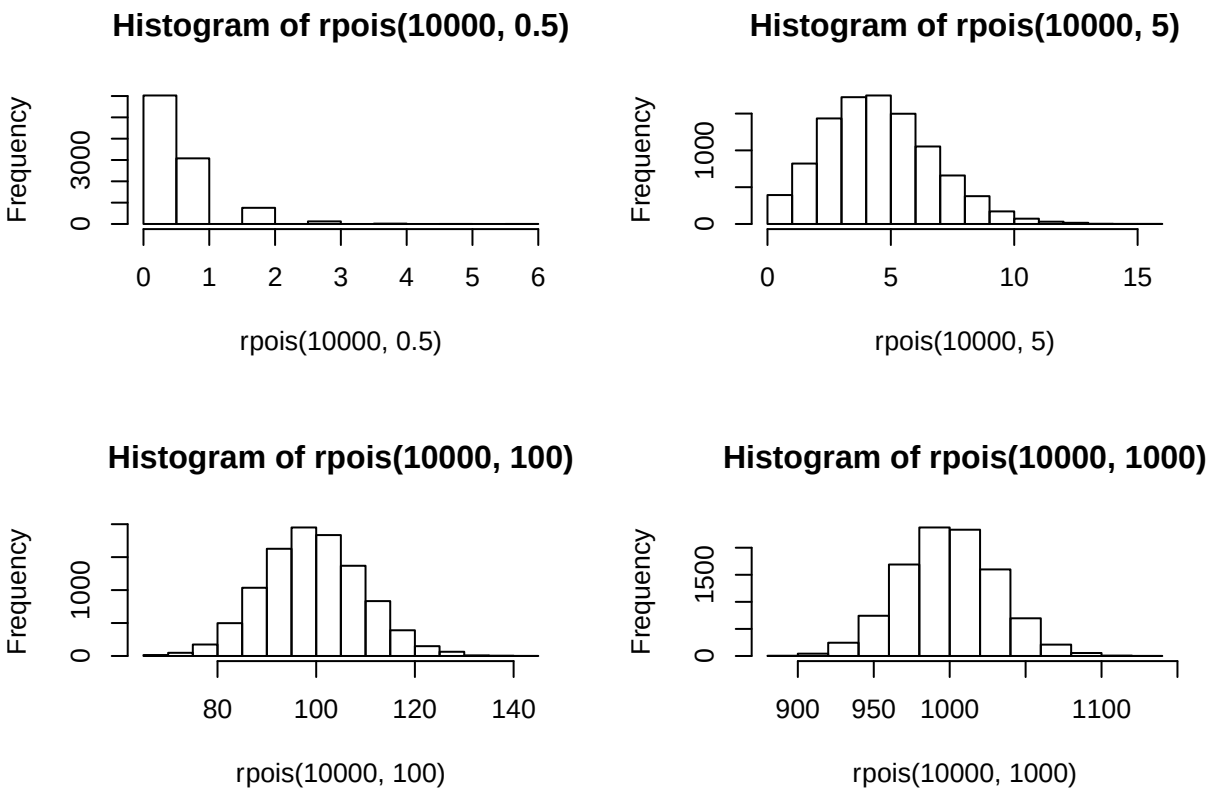









```
par(mfrow=c(2,2))
hist(rpois(10000,0.5))
hist(rpois(10000,5))
hist(rpois(10000,100))
hist(rpois(10000,1000))
```



What can be observed here is that, keeping n fixed and varying lambda, we obtain different histograms. The larger the value of lambda, the larger the values of Y on average and the larger also the spread in the values of Y. For high lambdas, the Poisson(lambda)-distributionis approximately equal to a normal distribution.

Now, keeping lambda fixed and varying n, we obtain again new histograms. We can see that the dimension of the x-axis stays approximately the same, since the lambda is the same. However, a higher number of the population makes the distribution of the values around this same population more equal. This does not mean that the outputs of the rpois function is the same for all elements, only that it is better distributed (more normally distributed).

2

In Poisson regression, the parameter lambda is modeled as follows:

log(lambda) = b0 + b1$x1$ + $b2$x2 + ... + bn*xn

This model states that for each output Y, lambda is modeled in a different way. This is due to the fact that the corresponding explanatory variables x are different. Thus, for each observation, the variances are different. Finally, the residuals do not come from one fixed distribution.

3

```
attach(africa)
miltcoup=as.numeric(africa$miltcoup)
oligarchy=as.numeric(africa$oligarchy)
pollib=as.numeric(africa$pollib)
parties=as.numeric(africa$parties)
pctvote=as.numeric(africa$pctvote)
popn=as.numeric(africa$popn)
```

```r
size=as.numeric(africa$size)
numelec=as.numeric(africa$numelec)
numregim=as.numeric(africa$numregim)

africa_fullmodel=glm(miltcoup~oligarchy+pollib+parties+pctvote+popn+size+numelec+numregim,family=poisso
summary(africa_fullmodel)
```

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties + pctvote +
##     popn + size + numelec + numregim, family = poisson, data = africa)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3443  -0.9542  -0.2587   0.3905   1.6953
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.5102693  0.9053301  -0.564  0.57301
## oligarchy    0.0730814  0.0345958   2.112  0.03465 *
## pollib      -0.7129779  0.2725635  -2.616  0.00890 **
## parties      0.0307739  0.0111873   2.751  0.00595 **
## pctvote      0.0138722  0.0097526   1.422  0.15491
## popn         0.0093429  0.0065950   1.417  0.15658
## size        -0.0001900  0.0002485  -0.765  0.44447
## numelec     -0.0160783  0.0654842  -0.246  0.80605
## numregim     0.1917349  0.2292890   0.836  0.40303
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 65.945  on 35  degrees of freedom
## Residual deviance: 28.668  on 27  degrees of freedom
## AIC: 111.48
##
## Number of Fisher Scoring iterations: 6
```

```r
confint(africa_fullmodel)
```

```
## Waiting for profiling to be done...
```

```
##                      2.5 %       97.5 %
## (Intercept) -2.4335049109  1.148089620
## oligarchy    0.0045915288  0.141483576
## pollib      -1.2570629668 -0.182012570
## parties      0.0080568606  0.052321186
## pctvote     -0.0054171503  0.032940743
## popn        -0.0038404317  0.022244262
## size        -0.0007146351  0.000272539
## numelec     -0.1438197483  0.114689702
## numregim    -0.2632334399  0.643070807
```

```r
coef(africa_fullmodel)
```

```
##   (Intercept)     oligarchy        pollib       parties       pctvote
```

```
## -0.5102692854  0.0730813725 -0.7129778804  0.0307739289  0.0138722128
##          popn          size       numelec      numregim
##  0.0093429334 -0.0001899975 -0.0160783349  0.1917349158
```

The results of this model shows that many variables might not be appropriate for this model. This is explained by the high values on the last column in the summary of the model. For many variables, these values are a lot above 0.05. Thus, a stepwise decrease procedure would be adequate for this model.

4

Now, we must check whether the coefficients are individually equal to zero (hypothesis H0) in the stepwise decrease method. As we can see in the last subsection, the last column in the linear model report has the highest value for the variable 'numelec'. Thus, this last is deleted as it is higher than 0.05. Thus, we perform the test again without this variable.

```
africa_model=glm(miltcoup~oligarchy+pollib+parties+pctvote+popn+size+numregim,family=poisson,data=africa
summary(africa_model)
```

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties + pctvote +
##     popn + size + numregim, family = poisson, data = africa)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3997  -0.9381  -0.2666   0.4220   1.6998
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.6078028  0.8239267  -0.738  0.46070
## oligarchy    0.0781368  0.0277656   2.814  0.00489 **
## pollib      -0.6773897  0.2290130  -2.958  0.00310 **
## parties      0.0296786  0.0102888   2.885  0.00392 **
## pctvote      0.0131290  0.0092895   1.413  0.15756
## popn         0.0089313  0.0063746   1.401  0.16120
## size        -0.0002021  0.0002436  -0.830  0.40682
## numregim     0.1758198  0.2210498   0.795  0.42639
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 65.945  on 35  degrees of freedom
## Residual deviance: 28.728  on 28  degrees of freedom
## AIC: 109.54
##
## Number of Fisher Scoring iterations: 5
```

The same now applies for the variable "numregim". The new test follows.

```
africa_model=glm(miltcoup~oligarchy+pollib+parties+pctvote+popn+size,family=poisson,data=africa)
summary(africa_model)
```

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties + pctvote +
```

```
##     popn + size, family = poisson, data = africa)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3522  -0.9651  -0.1945   0.4833   1.6179
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.1126871  0.5163030  -0.218 0.827228
## oligarchy    0.0859620  0.0259100   3.318 0.000908 ***
## pollib      -0.6894029  0.2278572  -3.026 0.002481 **
## parties      0.0291944  0.0101954   2.863 0.004190 **
## pctvote      0.0141588  0.0091980   1.539 0.123723
## popn         0.0062736  0.0053994   1.162 0.245272
## size        -0.0001950  0.0002425  -0.804 0.421378
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 65.945  on 35  degrees of freedom
## Residual deviance: 29.363  on 29  degrees of freedom
## AIC: 108.17
##
## Number of Fisher Scoring iterations: 5
```

Now, eliminating the variable "size".

```
africa_model=glm(miltcoup~oligarchy+pollib+parties+pctvote+popn,family=poisson,data=africa)
summary(africa_model)
```

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties + pctvote +
##     popn, family = poisson, data = africa)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4109  -0.9943  -0.1399   0.5516   1.6125
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.244466   0.495708  -0.493  0.62190
## oligarchy    0.083168   0.025437   3.270  0.00108 **
## pollib      -0.652830   0.221234  -2.951  0.00317 **
## parties      0.029800   0.010294   2.895  0.00379 **
## pctvote      0.013842   0.009282   1.491  0.13591
## popn         0.005587   0.005378   1.039  0.29883
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 65.945  on 35  degrees of freedom
## Residual deviance: 30.044  on 30  degrees of freedom
```

```
## AIC: 106.85
##
## Number of Fisher Scoring iterations: 5
```

Eliminating "popn":

```
africa_model=glm(miltcoup~oligarchy+pollib+parties+pctvote,family=poisson,data=africa)
summary(africa_model)
```

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties + pctvote,
##     family = poisson, data = africa)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5456  -0.9841  -0.1881   0.5948   1.6705
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.093657   0.463279  -0.202  0.83979
## oligarchy    0.095358   0.022421   4.253 2.11e-05 ***
## pollib      -0.666615   0.217564  -3.064  0.00218 **
## parties      0.025630   0.009502   2.697  0.00699 **
## pctvote      0.012134   0.009056   1.340  0.18031
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 65.945  on 35  degrees of freedom
## Residual deviance: 31.081  on 31  degrees of freedom
## AIC: 105.89
##
## Number of Fisher Scoring iterations: 5
```

and "pctvote":

```
africa_model=glm(miltcoup~oligarchy+pollib+parties,family=poisson,data=africa)
summary(africa_model)
```

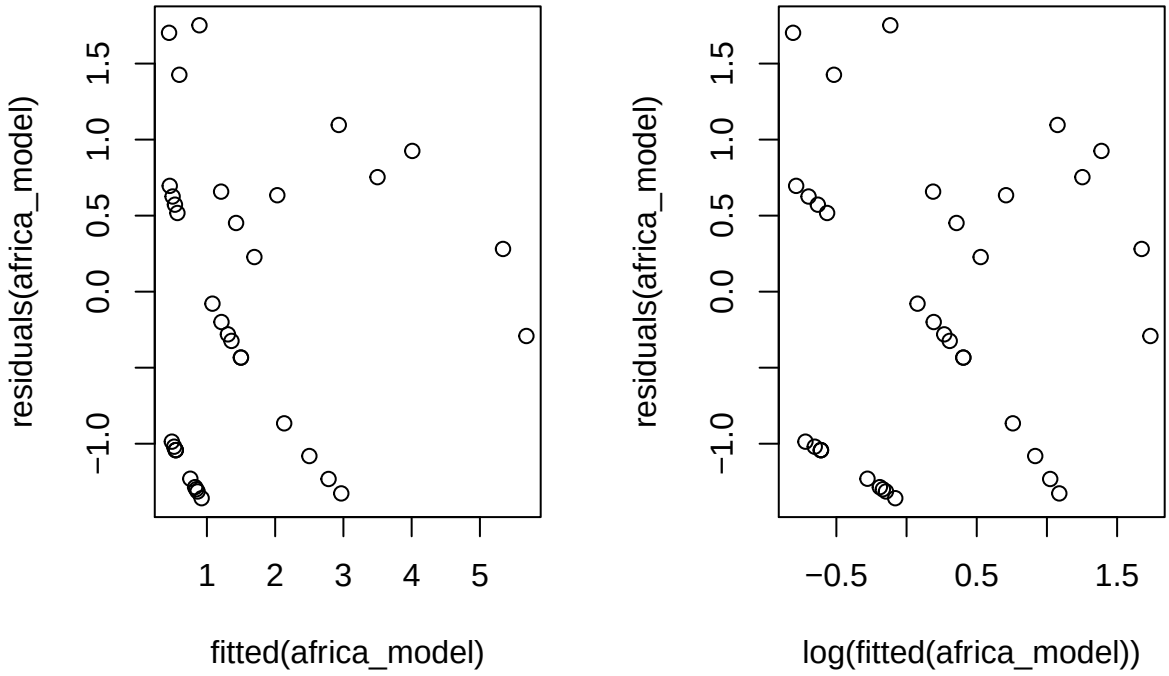
```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties, family = poisson,
##     data = africa)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3583  -1.0424  -0.2863   0.6278   1.7517
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.251377   0.372689   0.674  0.50000
## oligarchy    0.092622   0.021779   4.253 2.11e-05 ***
## pollib      -0.574103   0.204383  -2.809  0.00497 **
## parties      0.022059   0.008955   2.463  0.01377 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 65.945  on 35  degrees of freedom
## Residual deviance: 32.856  on 32  degrees of freedom
## AIC: 105.66
##
## Number of Fisher Scoring iterations: 5
```

Now, all the variables in the model are significant, as their coefficients are lower than 0.05. We end up with the variables "oligarchy", "pollib" and "parties" as explanatory variables for the output "miltcoup". Thus, many variables were deleted from the model.

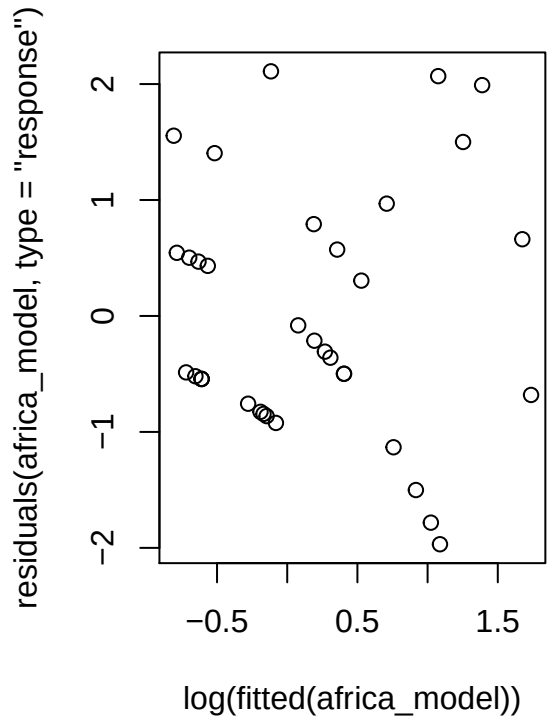
5

```
par(mfrow=c(1,2))
plot(fitted(africa_model),residuals(africa_model))
plot(log(fitted(africa_model)),residuals(africa_model))
```

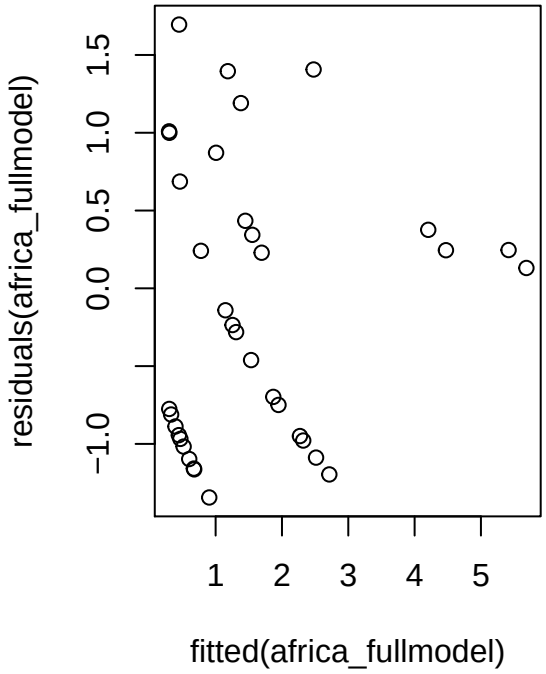


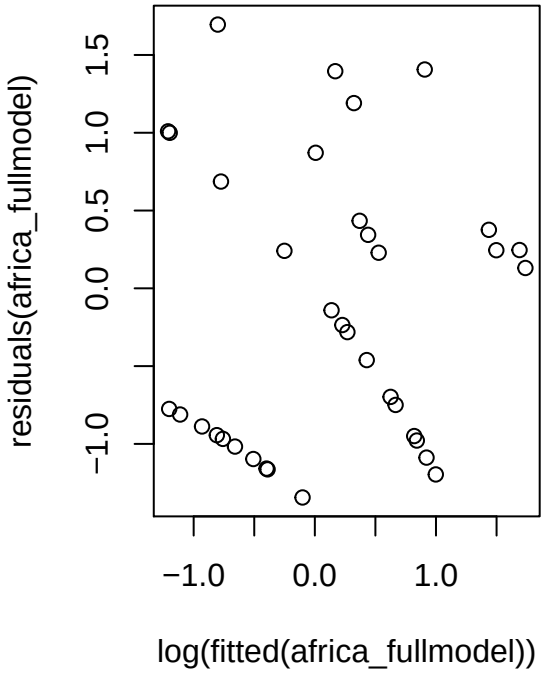

```
par(mfrow=c(1,2))
plot(log(fitted(africa_model)),residuals(africa_model,type="response"))

par(mfrow=c(1,2))
```

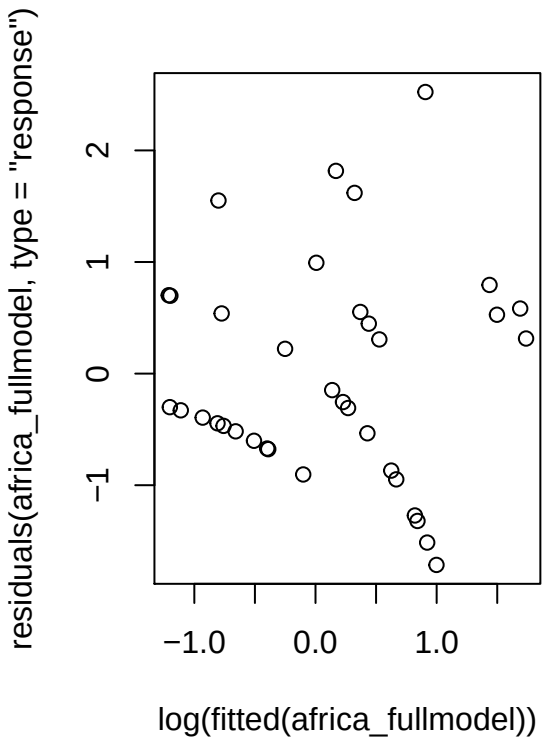



```
plot(fitted(africa_fullmodel),residuals(africa_fullmodel))
plot(log(fitted(africa_fullmodel)),residuals(africa_fullmodel))
```






```
par(mfrow=c(1,2))
plot(log(fitted(africa_fullmodel)),residuals(africa_fullmodel,type="response"))
```



The plots do not have any specific structure, and there does not seem to exist any major changes.