Assignment 4

Tanjina Islam, Miguel Morales Expósito and Carlos Perales Liñan, group 12

22 March 2018

Exercise 1

We load the data from the data source.

```
flies = read.table("fruitflies.txt",header=TRUE);
```

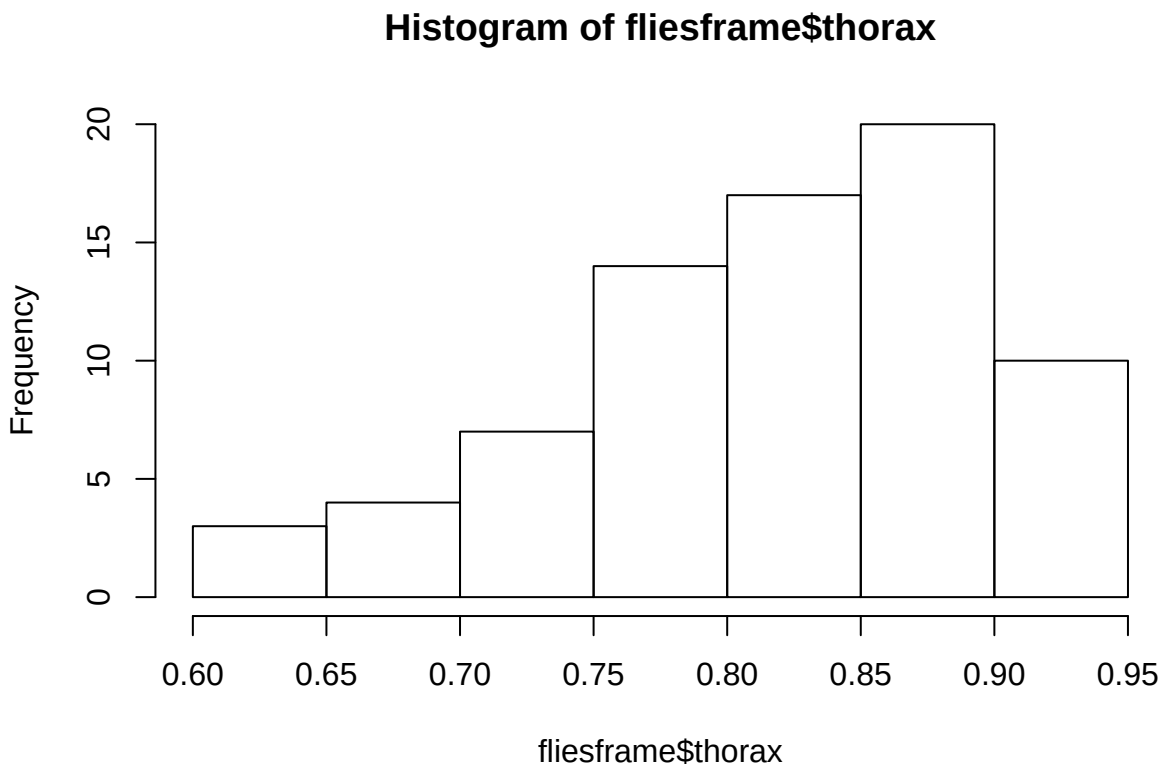
Task 1

We add the logarithm of longevity in the data frame.

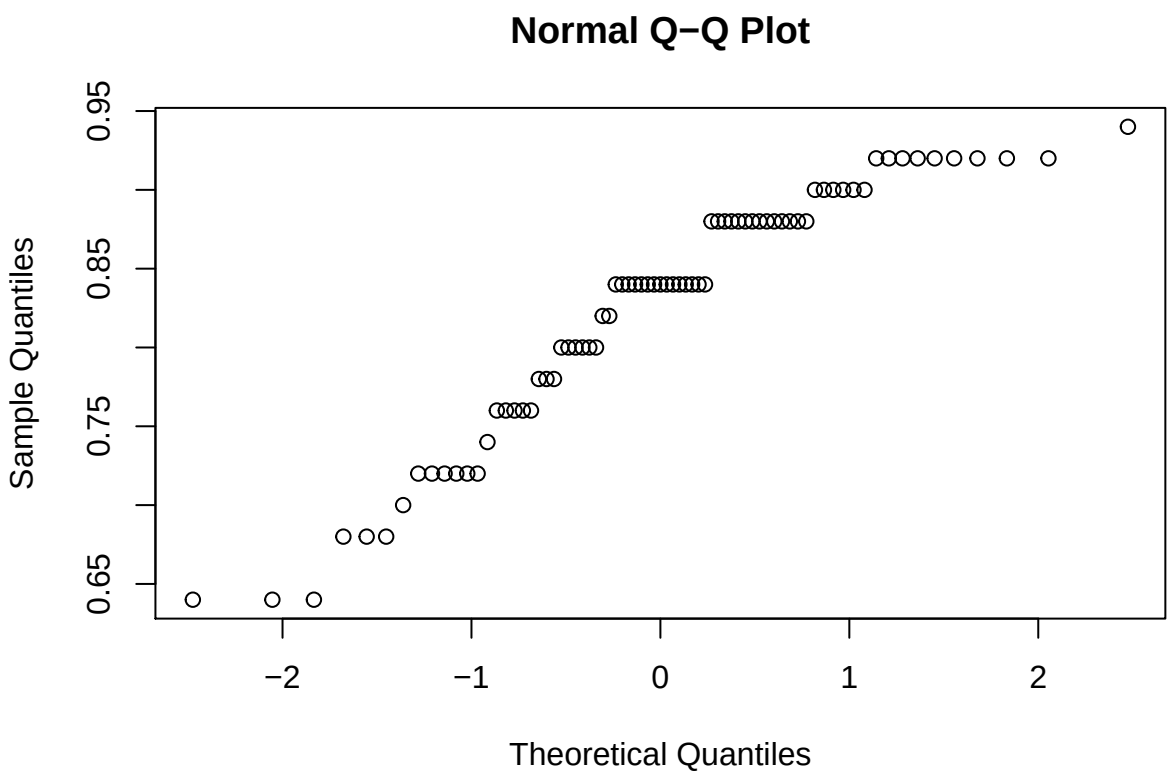
```
fliesframe = data.frame(thorax=flies$thorax,longevity=flies$longevity,activity=flies$activity,l
```

Task 2

```
hist(fliesframe$thorax)
```

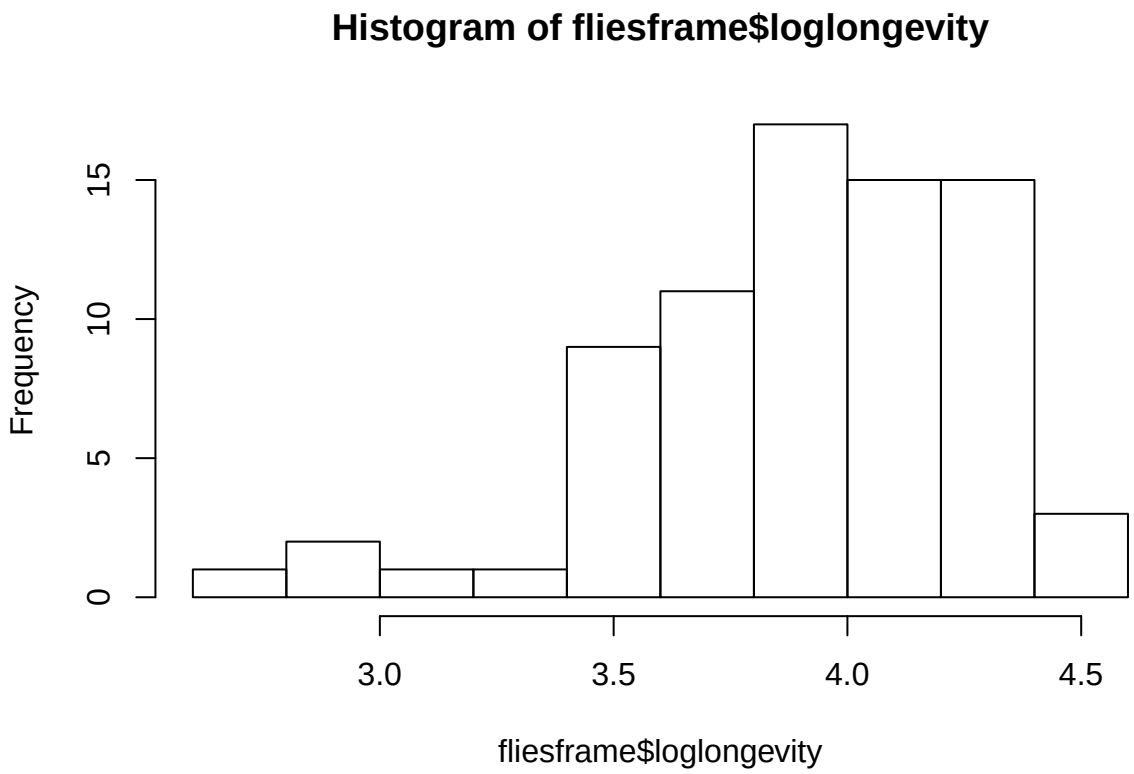


```
qqnorm(fliesframe$thorax)
```

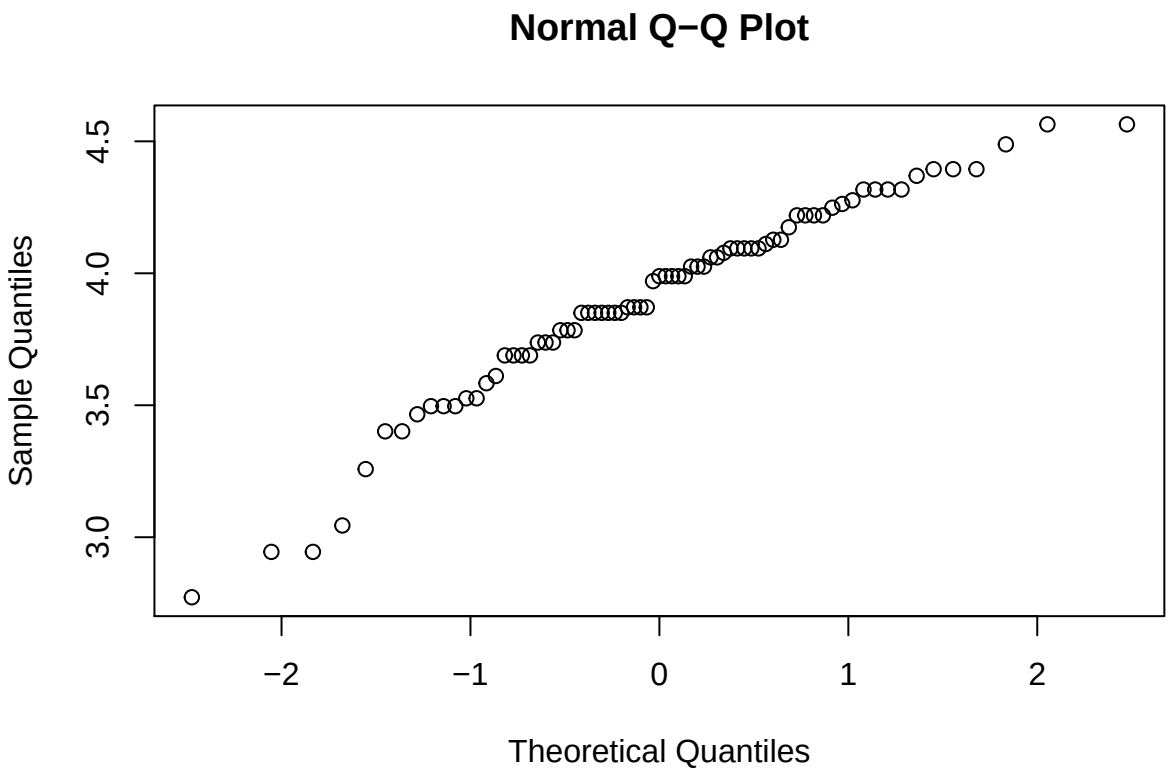


The population of thorax does not seem normal.

```
hist(fliesframe$loglongevity)
```




```
qqnorm(fliesframe$loglongevity)
```



The population of loglongevity does not seem normal.

Task 3

We perfom an anova just considering the sexual activity

```
fliesanova = lm(loglongevity~activity,data=fliesframe)
anova(fliesanova)
```

```
## Analysis of Variance Table
##
## Response: loglongevity
##           Df Sum Sq Mean Sq F value  Pr(>F)
## activity   2   3.67   1.833    19.4 1.8e-07 ***
## Residuals 72   6.80   0.094
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We get a p-value 1.798e-07, therefore, H0 is reject so we can say that sexual activity affects the longevity of the flies.

Task 4

```
summary(fliesanova)
```

```
##
## Call:
## lm(formula = loglongevity ~ activity, data = fliesframe)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9553 -0.1334  0.0255  0.2089  0.4922
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3.6021     0.0614   58.62  < 2e-16 ***
## activityisolated    0.5172     0.0869    5.95  8.8e-08 ***
## activitylow         0.3977     0.0869    4.58  1.9e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.307 on 72 degrees of freedom
## Multiple R-squared:  0.35,   Adjusted R-squared:  0.332
## F-statistic: 19.4 on 2 and 72 DF,  p-value: 1.8e-07
```

According to the estimates longevity increases more when the sexual activity is low. ..* High activity = 3.602 ..* Low activity = 3.602 + 0.517 = 4.119 ..* Isolated = 3.602 + 0.397 = 3.999

Task 5

We make a 2-way anova.

```
fliesframe$activity = as.factor(fliesframe$activity)
fliesframe$thorax = as.factor(fliesframe$thorax)

fliesanova2 = lm(loglongevity~activity+thorax,data = fliesframe)
anova(fliesanova2)
```

```
## Analysis of Variance Table
##
## Response: loglongevity
##           Df Sum Sq Mean Sq F value  Pr(>F)
## activity   2   3.67   1.833    51.5 1.1e-13 ***
## thorax    13   4.70   0.361    10.2 9.5e-11 ***
## Residuals 59   2.10   0.036
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We obtain a p-value of 1.141e-13, therefore, we can say that activity has a main effect on longevity when we consider the thorax length.

Task 6

First we calculate the average thorax length.

```
fliesframe$thorax = as.numeric(fliesframe$thorax)
average_thorax = mean(fliesframe$thorax)
fliesframe$thorax = as.factor(fliesframe$thorax)
```

We obtain an average of 8.867.

We obtain the summary.

```
contrasts(fliesframe$thorax)=contr.sum
contrasts(fliesframe$activity)=contr.sum
fliesanova2 = lm(loglongevity~activity+thorax,data = fliesframe)
summary(fliesanova2)
```

```
##
## Call:
## lm(formula = loglongevity ~ activity + thorax, data = fliesframe)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.303 -0.148  0.006  0.129  0.341
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.8026     0.0317  119.97  < 2e-16 ***
## activity1    -0.2758     0.0348   -7.93  7.1e-11 ***
## activity2     0.1865     0.0349    5.34  1.5e-06 ***
## thorax1      -0.5457     0.1070   -5.10  3.8e-06 ***
## thorax2      -0.5443     0.1082   -5.03  4.9e-06 ***
## thorax3      -0.3782     0.1807   -2.09  0.04064 *
## thorax4      -0.1137     0.0783   -1.45  0.15203
## thorax5      -0.0304     0.1803   -0.17  0.86687
## thorax6       0.0508     0.0850    0.60  0.55214
## thorax7      -0.0361     0.1057   -0.34  0.73374
## thorax8      -0.0668     0.0791   -0.84  0.40202
## thorax9       0.3121     0.1314    2.38  0.02082 *
## thorax10      0.1961     0.0556    3.53  0.00081 ***
## thorax11      0.3561     0.0564    6.32  3.8e-08 ***
## thorax12      0.2247     0.0876    2.56  0.01293 *
## thorax13      0.2471     0.0685    3.61  0.00064 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.189 on 59 degrees of freedom
## Multiple R-squared:  0.799,  Adjusted R-squared:  0.748
## F-statistic: 15.7 on 15 and 59 DF,  p-value: 2.69e-15
```

With the values thorax9 = 0.312 (for the average thorax length) and isolated-activity = -0.275, low-activity = 0.186 and high-activity = -(-0.275+0.186) = 0.089, we calculate the estimates for flies with average thorax..

$Y_{isolated,thorax9} = 3.802 + 0.312 - 0.275 = 3.839$ $Y_{low,thorax9} = 3.802 + 0.312 + 0.186 = 4.3$ $Y_{high,thorax9} = 3.802 + 0.312 + 0.089 = 4.203$

For the flies with smallest thorax we use the same estimates for activity but we use thorax1 = -0.545.

$Y_{isolated,thorax1} = 3.802 - 0.545 - 0.275 = 2.982$ $Y_{low,thorax1} = 3.802 - 0.545 + 0.186 = 3.443$ $Y_{high,thorax1} = 3.802 - 0.545 + 0.089 = 3.346$

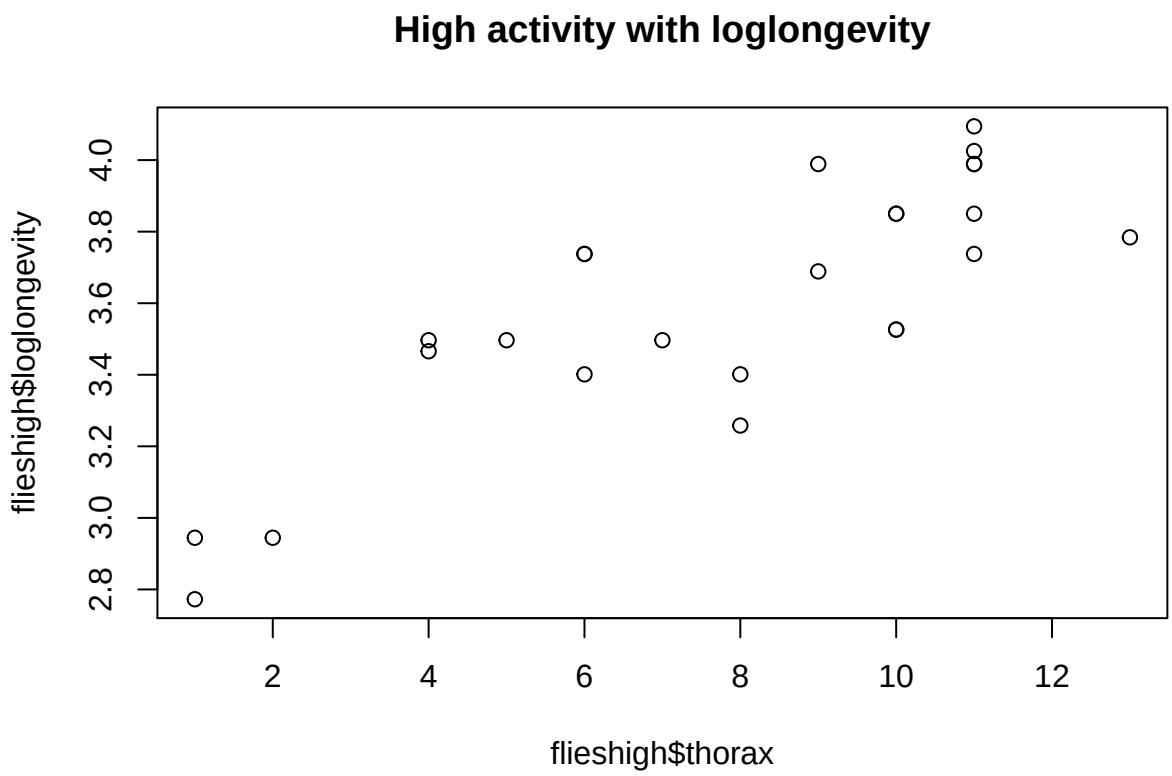
Task 7

In order to investigate graphically how does thorax length influence longevity, we separate in three different variables the data depending on the sexual activity.

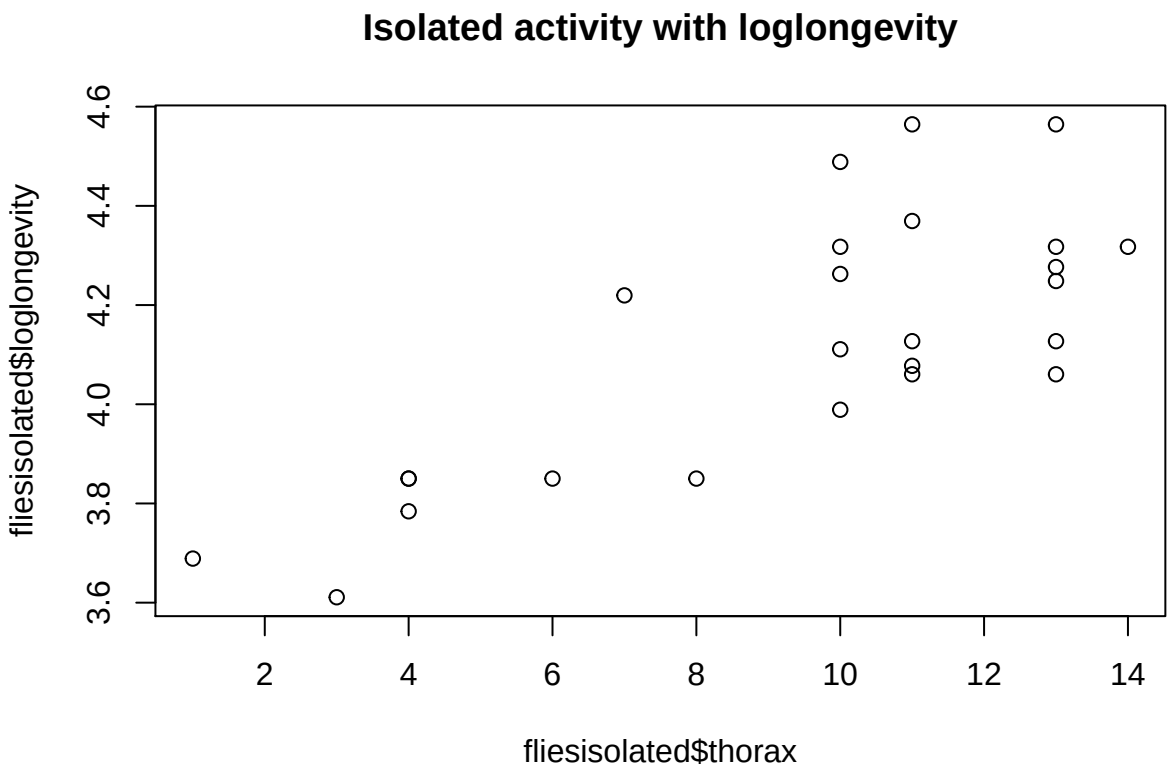
```
fliesframe$thorax = as.numeric(fliesframe$thorax)
flieshigh = fliesframe[which(fliesframe$activity=="high"),]
fliesisolated = fliesframe[which(fliesframe$activity=="isolated"),]
flieslow = fliesframe[which(fliesframe$activity=="low"),]
```

Now we plot the thorax length against the loglongevity.

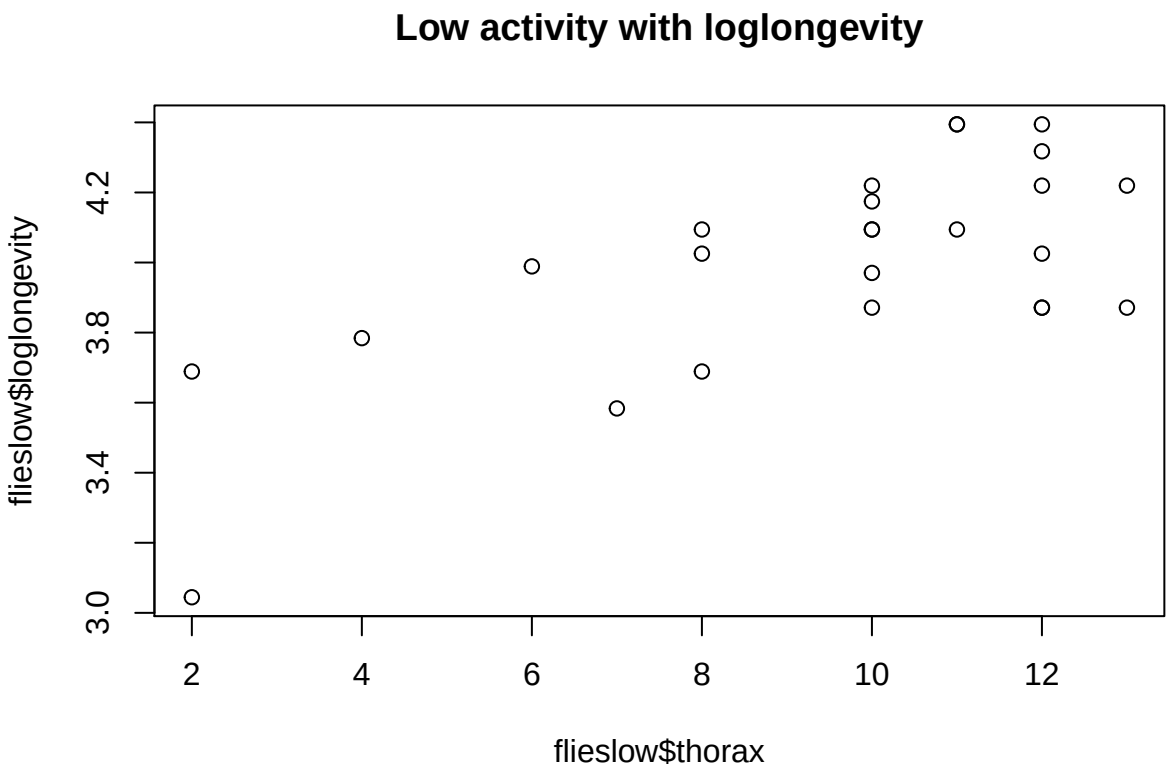
```
plot(flieshigh$thorax,flieshigh$loglongevity,main="High activity with loglongevity")
```




```
plot(fliesisolated$thorax,fliesisolated$loglongevity,main="Isolated activity with loglongevity"
```




```
plot(flieslow$thorax,flieslow$loglongevity,main="Low activity with loglongevity")
```



We see that longevity increases with thorax length. We get higher longevity values for flies with low sexual activity and also for high sexual activity. The values seem to be lower when the flies have been isolated.

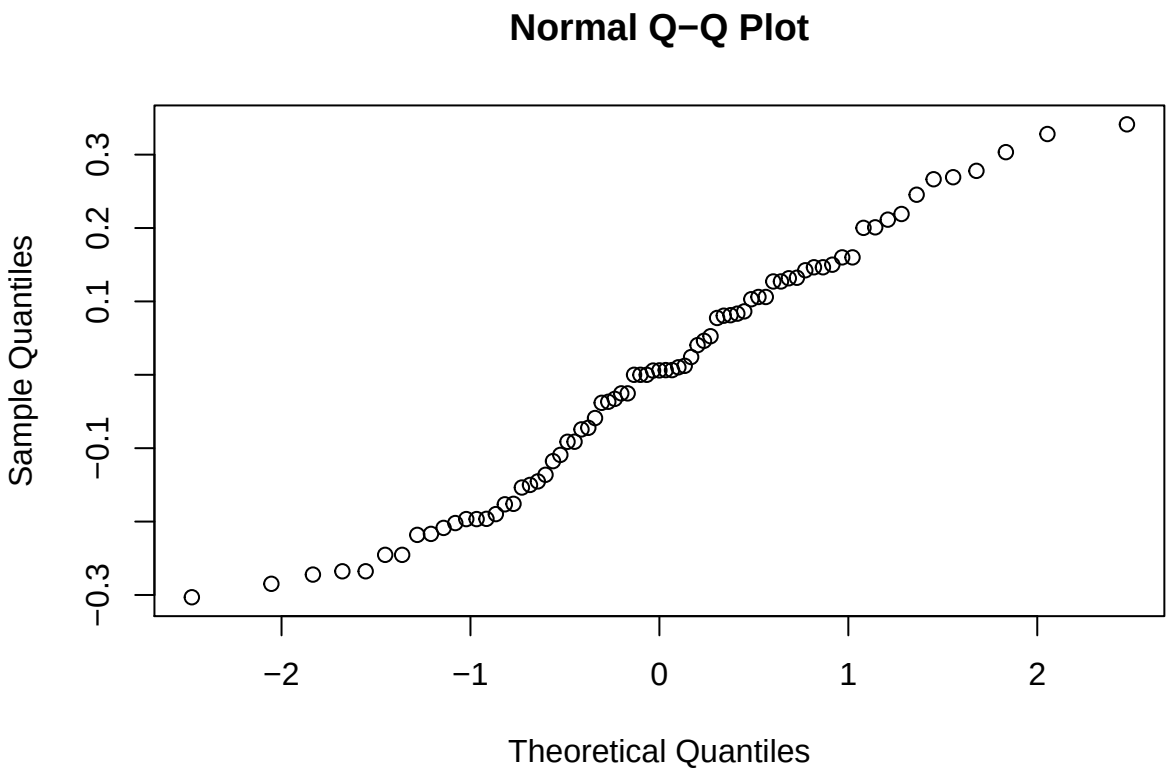
Task 8

The analysis with the thorax length should not be included because there is no real interaction between the sexual activity and the thorax length.

This is because the experimenters randomly chose the sexual activity that the flies were going to have. Thus, we cannot conclude whether the thorax length will really influence in the sexual activity of the fly.

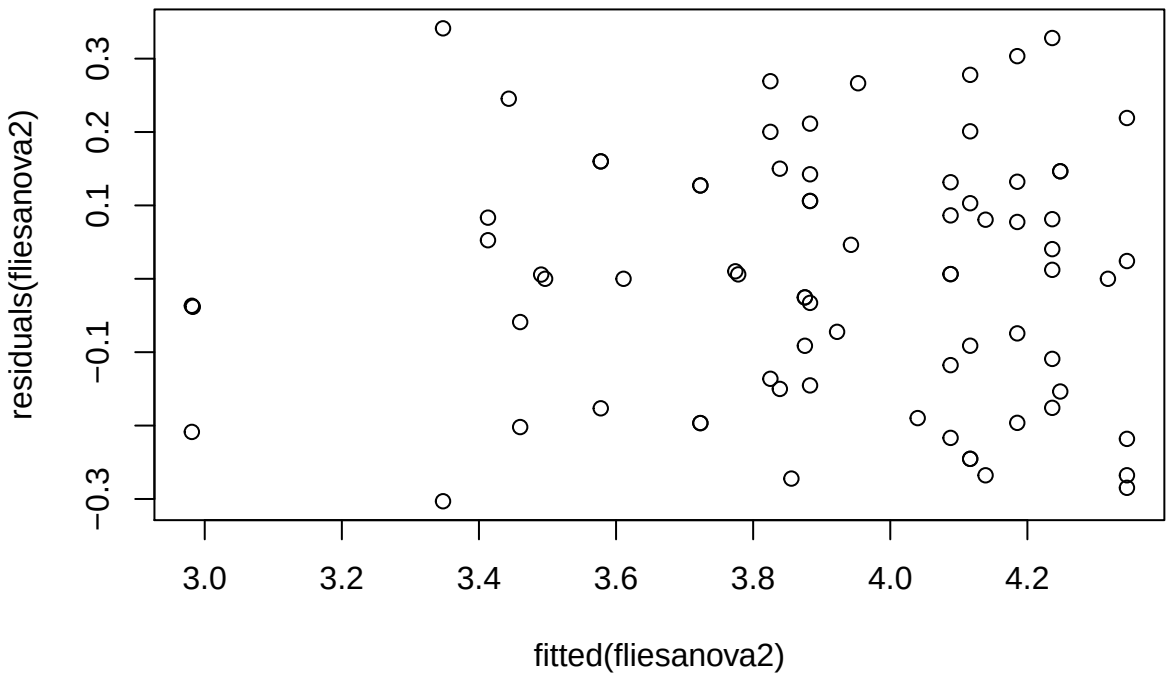
Task 9

```
qqnorm(residuals(fliesanova2))
```



The normality seems doubtful according to the qq-plot.

```
plot(fitted(fliesanova2),residuals(fliesanova2))
```



The spread in the residuals seem to be bigger with bigger fitted values.

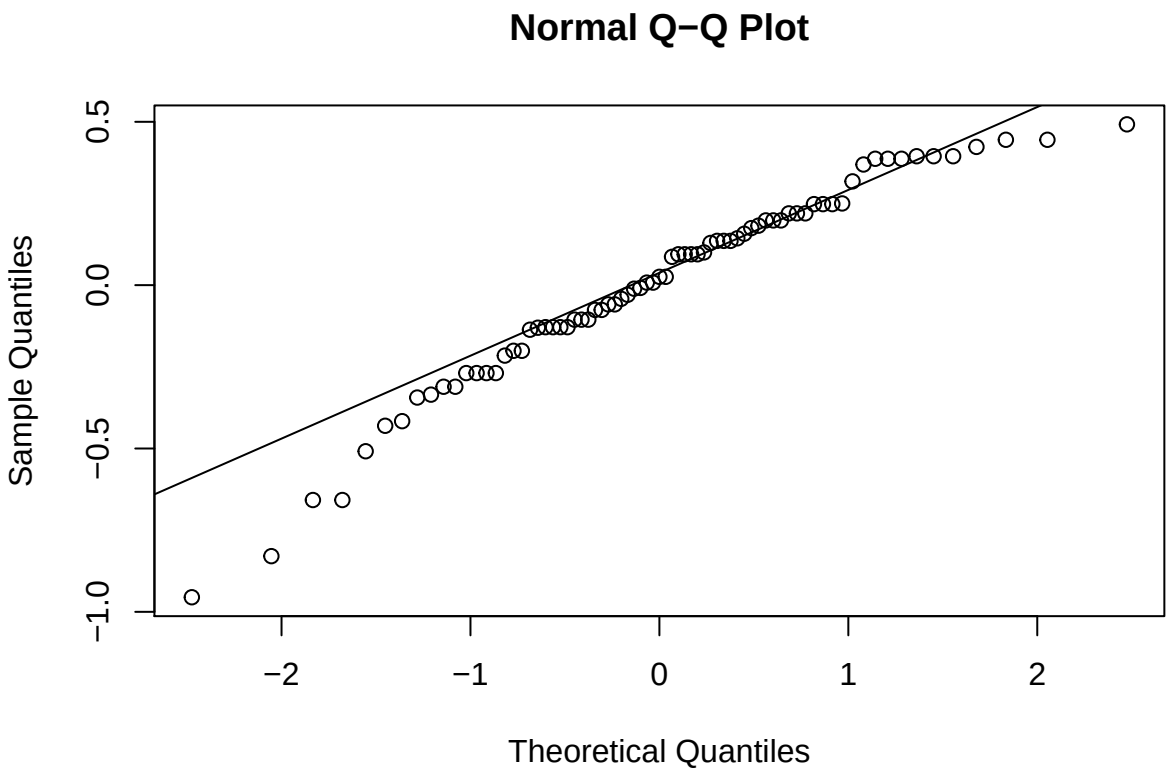
Task 10

```
fliesframe$activity=as.factor(fliesframe$longevity)
fliesaov = lm(longevity~thorax+activity,data=fliesframe)
drop1(fliesaov,type="F")
```

```
## Warning: attempting model selection on an essentially perfect fit is
## nonsense
```

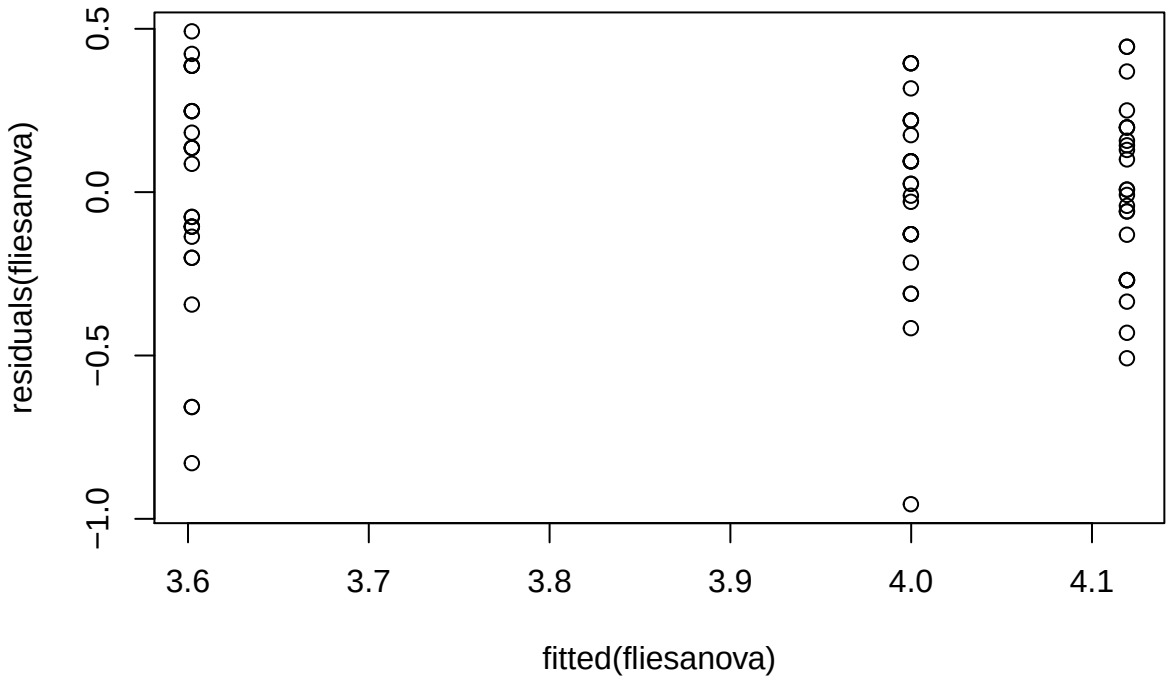
```
## Single term deletions
##
## Model:
## longevity ~ thorax + activity
##          Df Sum of Sq   RSS   AIC
## <none>                     0 -4660
## thorax    1         0     0 -4620
## activity 32     12728 12728   389
```

```
qqnorm(residuals(fliesanova))
qqline(residuals(fliesanova))
```



The normality is doubtful. However, it could be normal. It seems more normal than when using the logaritmic value of longevity.

```
plot(fitted(fliesanova),residuals(fliesanova))
```



We cannot judge becasue the values are not spread in the x axis. This is a prove that using the logarithmic value was a good idea.

Exercise 2

We load the data from the data source.

```
expsi = read.table("psi.txt", header=TRUE)
```

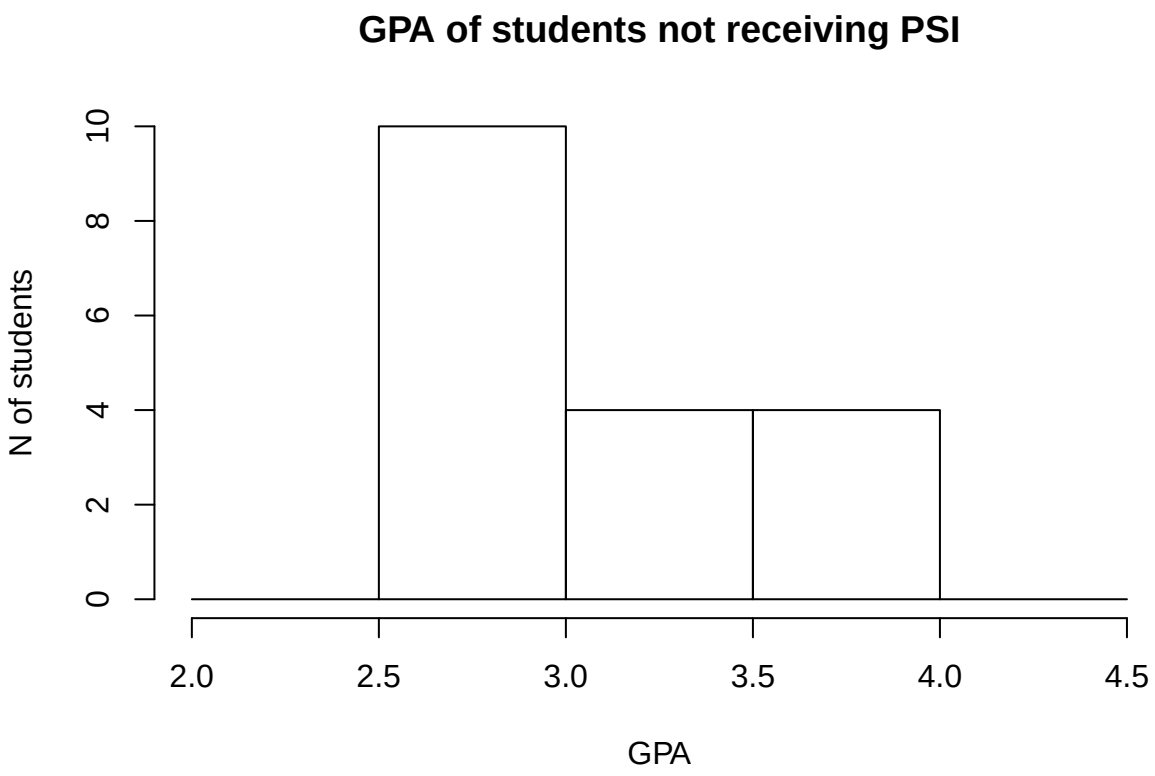
Task 1

To help us on making some summaries, we divide the data in 2 groups: the students that received psi and the ones that didn't.

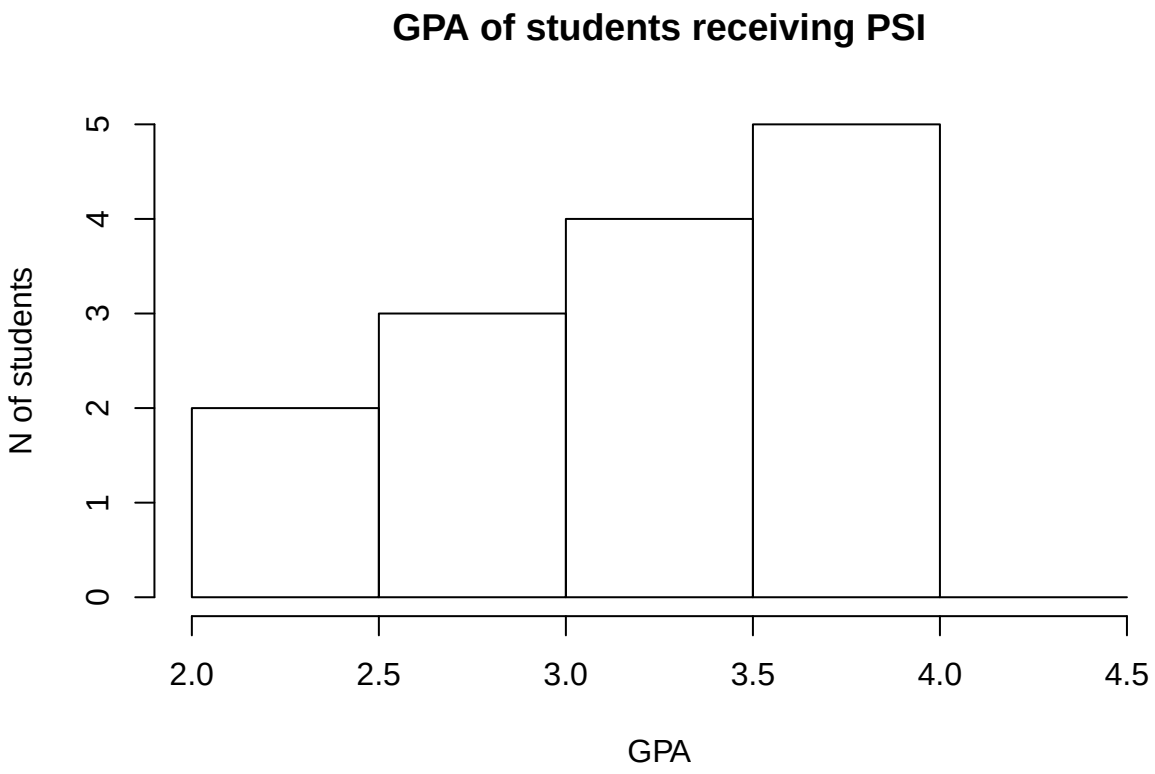
```
nousepsi = expsi[which(expsi$psi == "0"),]
usepsi = expsi[which(expsi$psi == "1"),]
```

Next, we show some data summaries:

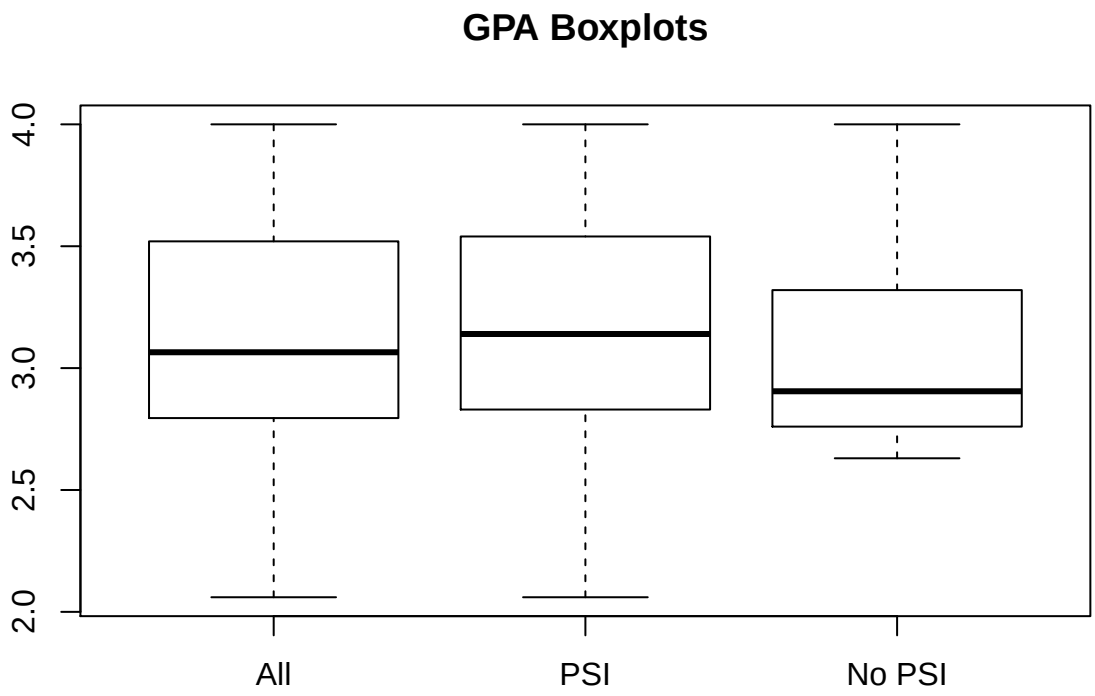
```
hist(nousepsi$gpa, breaks = c(2.0, 2.5, 3.0, 3.5, 4.0, 4.5), xlab="GPA", ylab="N of students",
```



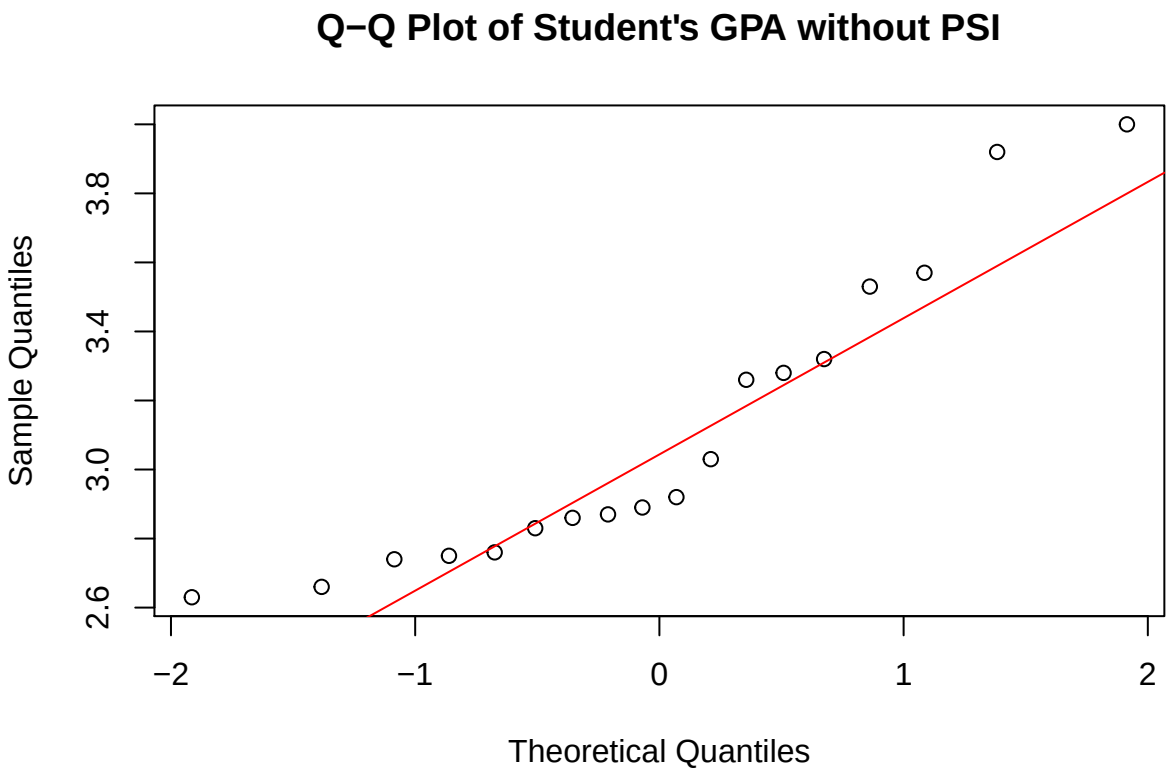

```
hist(usepsi$gpa, breaks = c(2.0, 2.5, 3.0, 3.5, 4.0, 4.5), xlab="GPA", ylab="N of students", ma
```



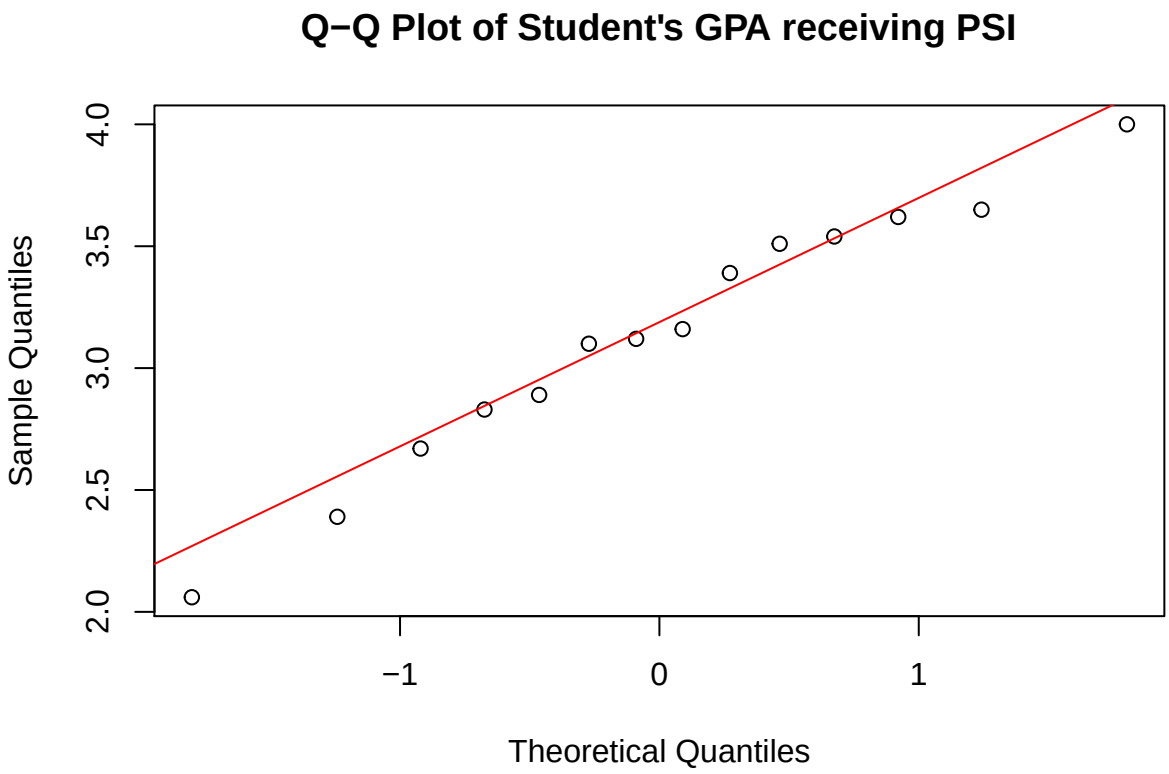

```
boxplot(expsi$gpa, usepsi$gpa, nousepsi$gpa, main="GPA Boxplots", names=c("All", "PSI", "No PSI
```




```
qqnorm(nousepsi$gpa,  main="Q-Q Plot of Student's GPA without PSI") # doesn't look normal
qqline(nousepsi$gpa, col="red")
```




```
qqnorm(usepsi$gpa, main="Q-Q Plot of Student's GPA receiving PSI") # looks normal
qqline(usepsi$gpa, col="red")
```



Task 2

We fit them into a linear regression model.

```
expsiglm=glm(passed~psi+gpa,data=expsi,family=binomial)
summary(expsiglm)
```

```
##
## Call:
## glm(formula = passed ~ psi + gpa, family = binomial, data = expsi)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.840  -0.628  -0.304   0.563   2.038
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -11.60       4.21   -2.75   0.0059 **
## psi             2.34       1.04    2.25   0.0247 *
## gpa             3.06       1.22    2.51   0.0122 *
## ---
## 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: 32.25
##
## Number of Fisher Scoring iterations: 5
```

Task 3

As we can see in the summary of the linear model, the usage of psi increases the linear predictor by 2.338.

Knowing this, we can calculate that it increases odds of passing by exp^2.338 = 10.3604948382

Task 4

With the summary from task 2, we can see the values that we need to estimate these probabilities.

With psi and gpa = 3.0:

-11.602 + 2.338 + (3.063 * 3.0) = -0.075 Probability = 1/(1 + e^0.075) = 0.4812588

Without psi and gpa = 3.0:

-11.602 + (3.063 * 3.0) = -2.408 Probability = 1/(1 + e^2.408) = 0.08256469

Task 5

We can also get these probabilities with the help of the summary in task 2.

With psi:

Probability = 1/(1 + e^11.602) = 9.15e-06

Without psi:

-11.602 + 2.34 = -9.26 Probability = 1/(1 + e^9.26) = 9.51e-05

Difference of probabilities: 9.51e-05 - 9.15e-06 = 8.59e-05

Task 6

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

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

The number 15 are the students that didn't show improvement from the 18, whereas 3 are those who did show improvement. Similarly with the second column, only 8 out of 14 students showed improvement and the remaining 6 didn't.

With the observations above, we can claim that the first column contains the students not receiving psi and the second one shows the students receiving it. Furthermore, row 1 shows the students that improved and row 2 the number of students that didn't.

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

After running Fisher's test, we can conclude that with a p-value of [0.027], we can reject the null hypothesis, which claims that the students receiving psi and the ones not receiving it have the same probablity of improvement.

Task 7

We don't believe this approach is wrong, Fisher's test is a right approach for 2x2 tables with not too big numbers. We believe that this approach is valid since the experiment meet the requirements for its use.

Task 8

Fishers is good for small counts in each 2x2 table cell. More exact than the other approach, but it doesn't work with big counts.

Exercise 3

First of all, we load the data from the data source.

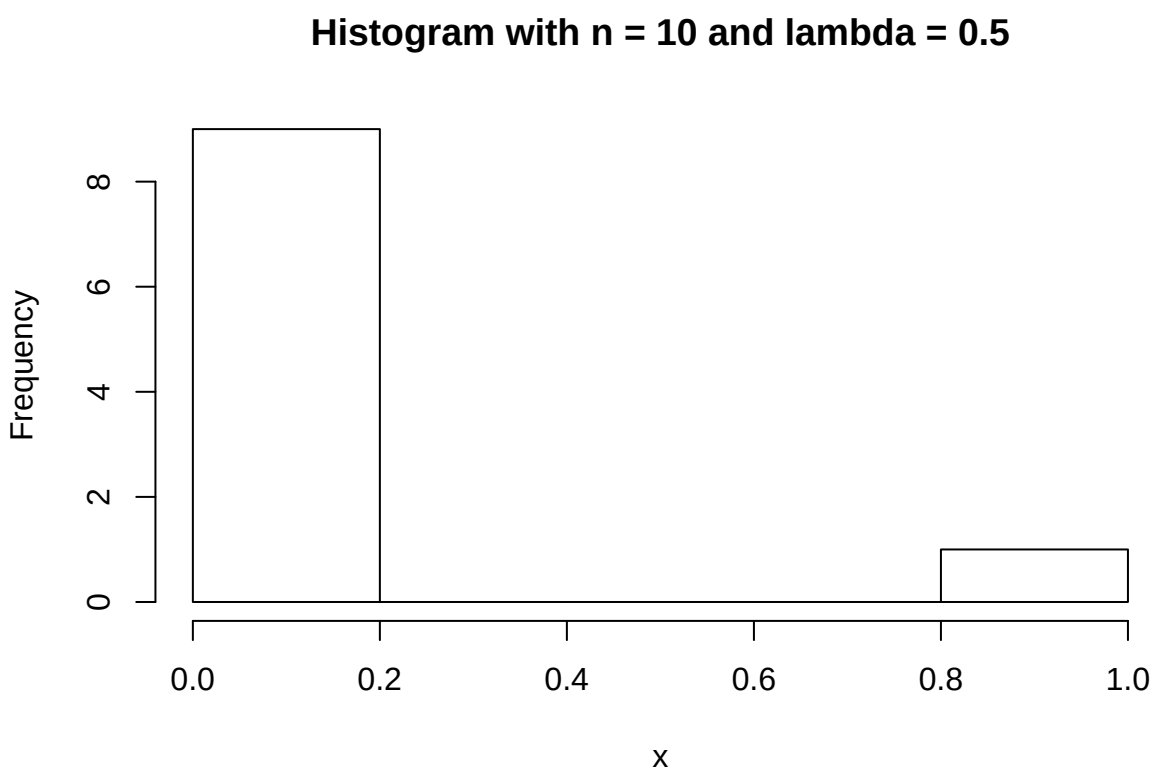
```
africa_data = read.table("africa.txt", header = TRUE)
```

Task 1

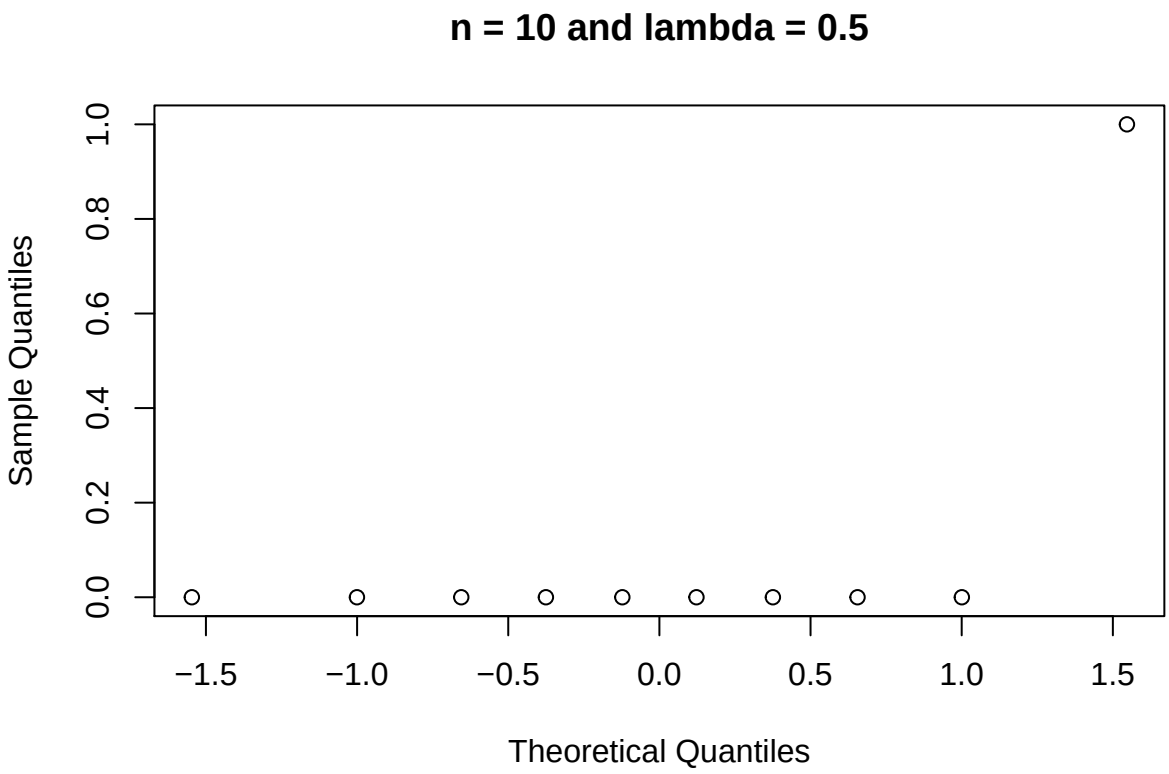
We are going to create an array with 12 different poisson distributions. The distributions are combinations of the values n={10,100,1000} with lambda values l={0.5,1,10,100}.

```
n = c(10, 100, 1000)
lambda = c(0.5, 1, 10, 100)

poisson_st = list()

for(i in n){
  for(j in lambda){
    poisson_st = c(poisson_st,list(rpois(i,j)))
  }
}
```

```r
x = poisson_st[[1]]
hist(x, main = "Histogram with n = 10 and lambda = 0.5")
```

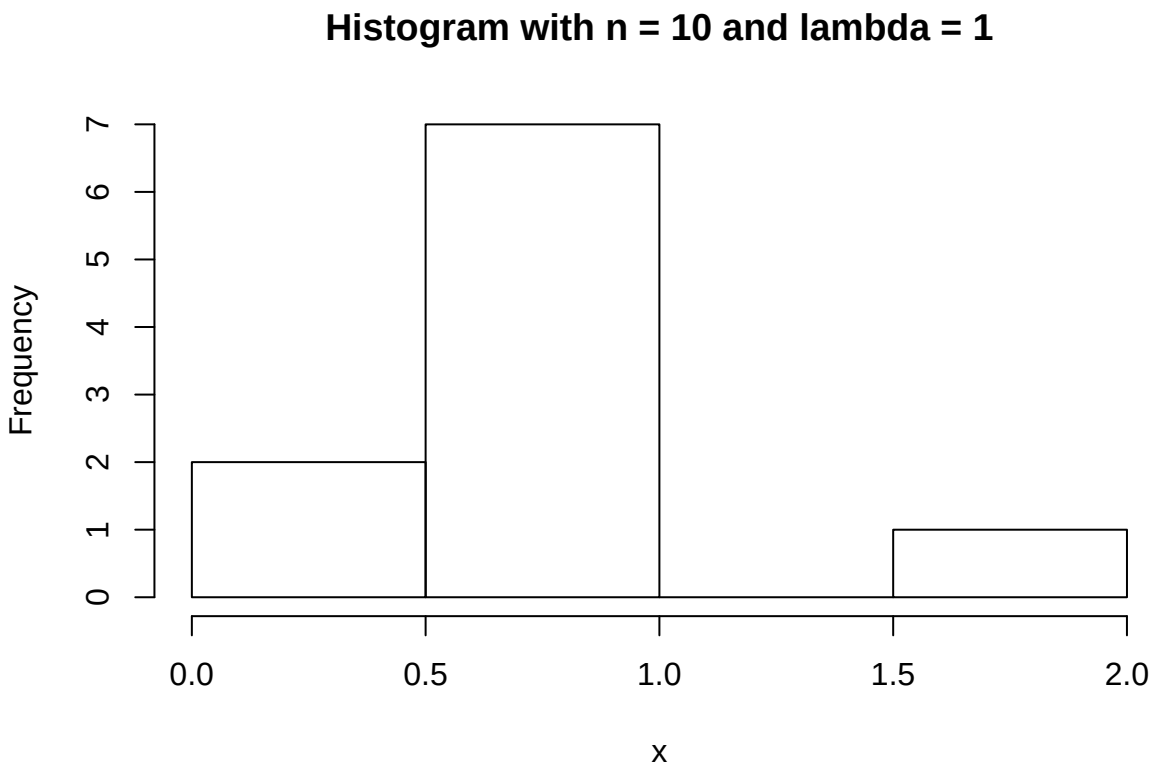



```r
qqnorm(x, main = "n = 10 and lambda = 0.5")
```

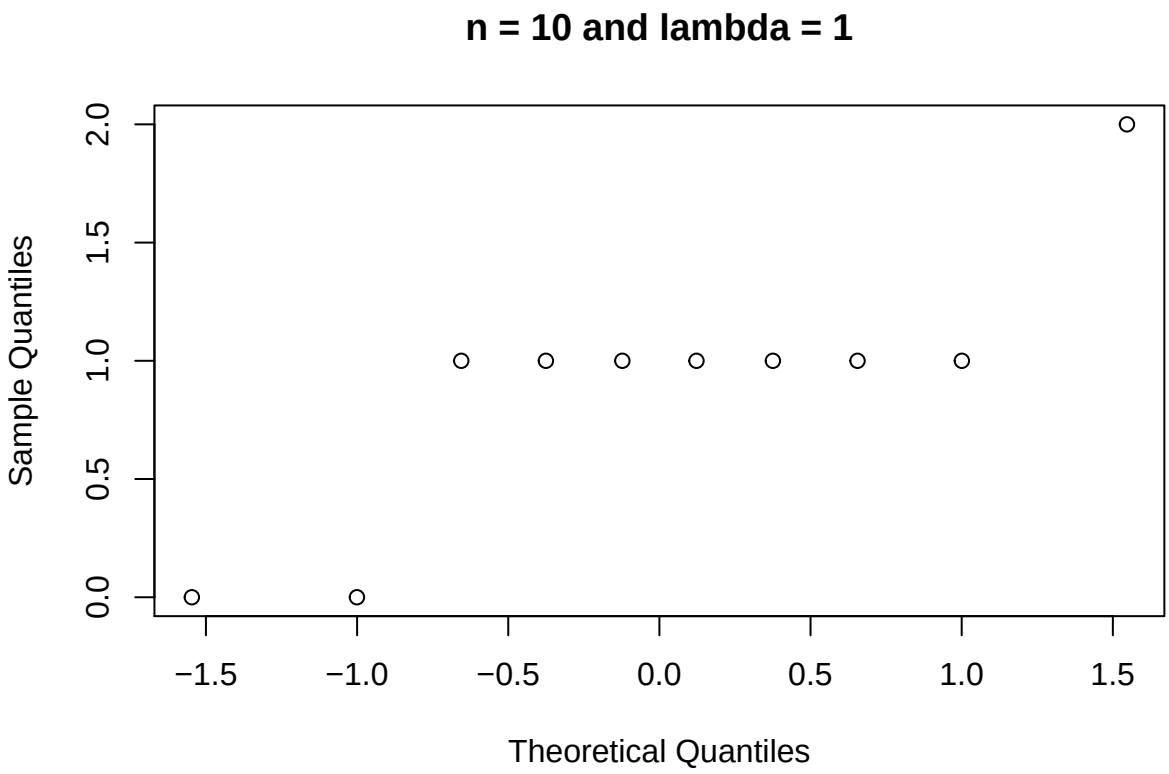


The distribution doesn't look normal.

```
x = poisson_st[[2]]
hist(x, main = "Histogram with n = 10 and lambda = 1")
```

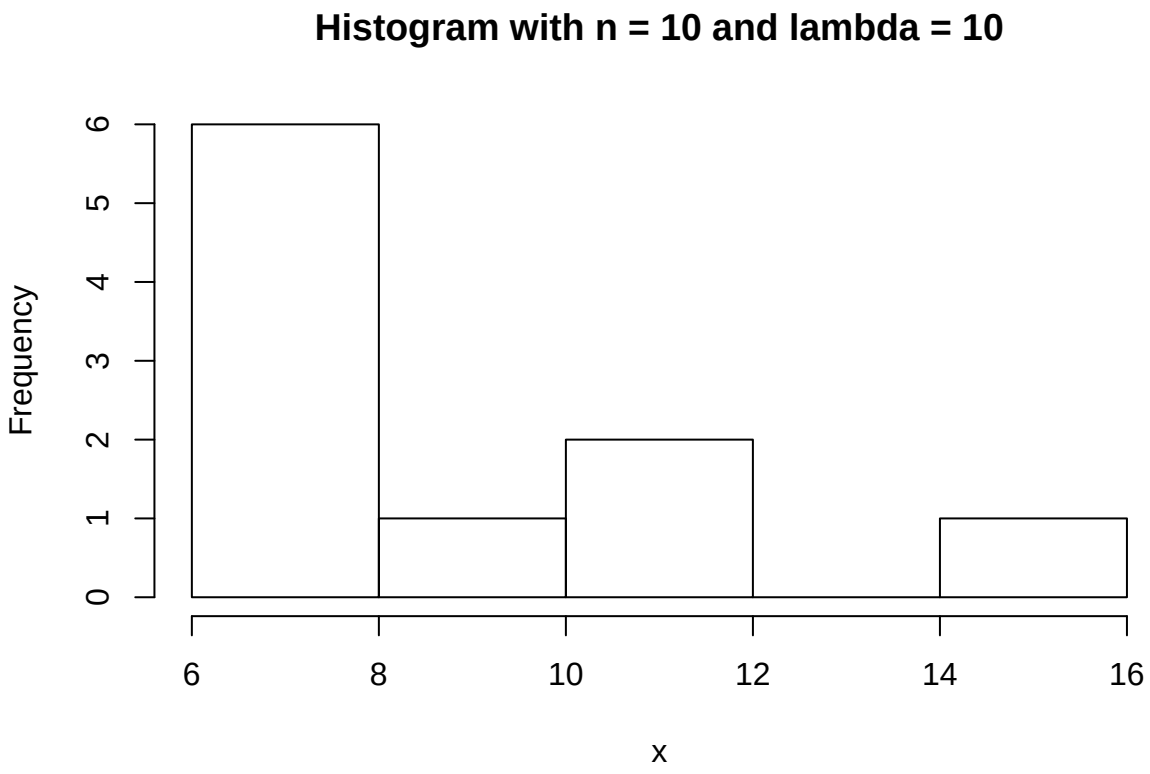



```
qqnorm(x, main = "n = 10 and lambda = 1")
```

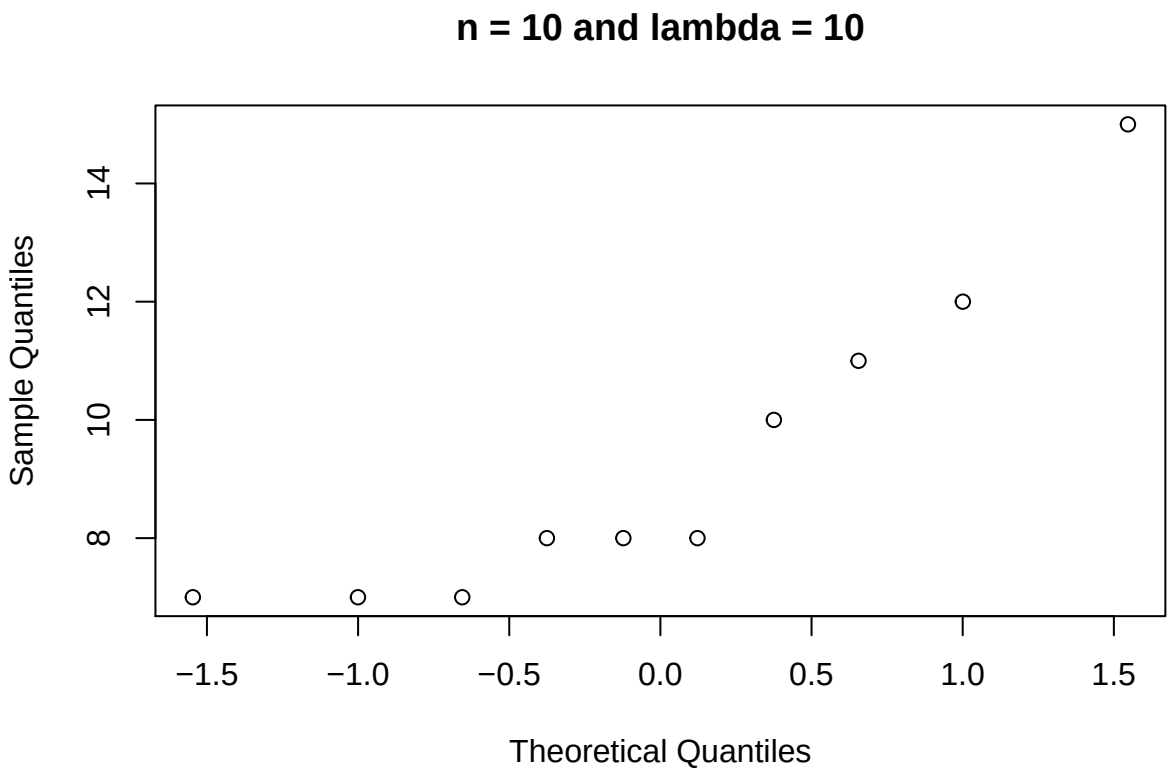


The distribution doesn't look normal.

```
x = poisson_st[[3]]
hist(x, main = "Histogram with n = 10 and lambda = 10")
```




```r
qqnorm(x, main = "n = 10 and lambda = 10")
```

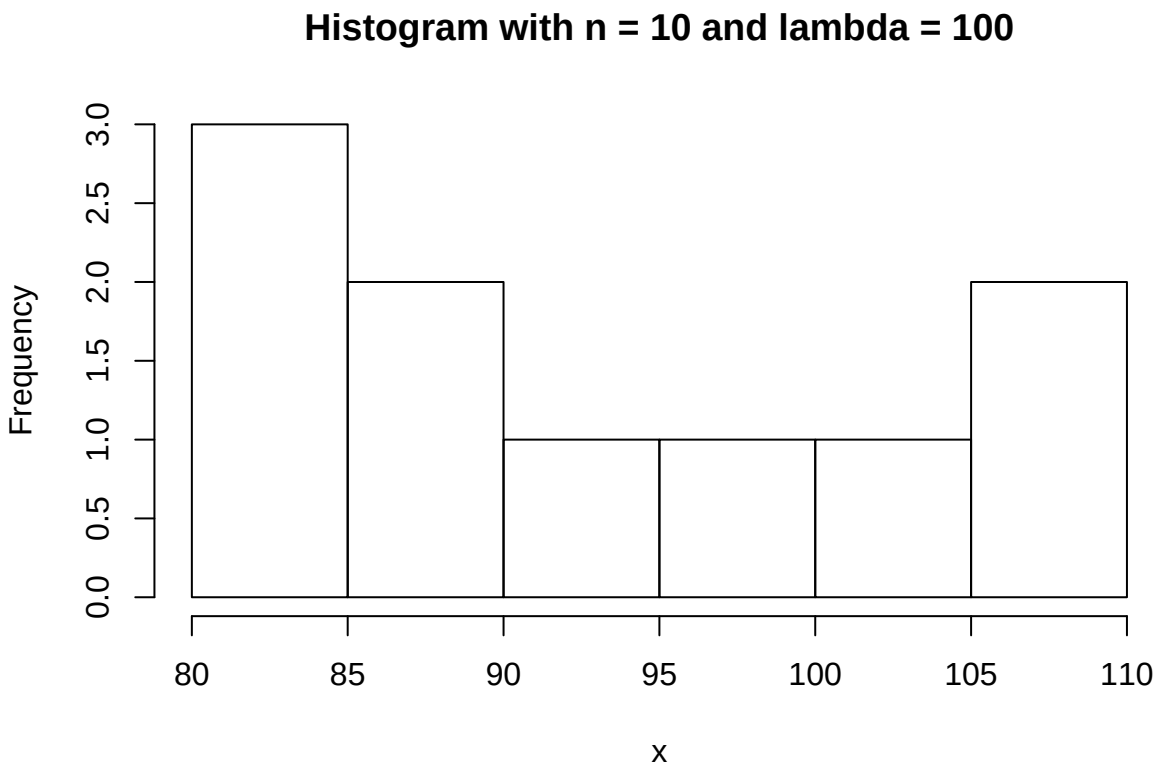


The distribution doesn't look normal.

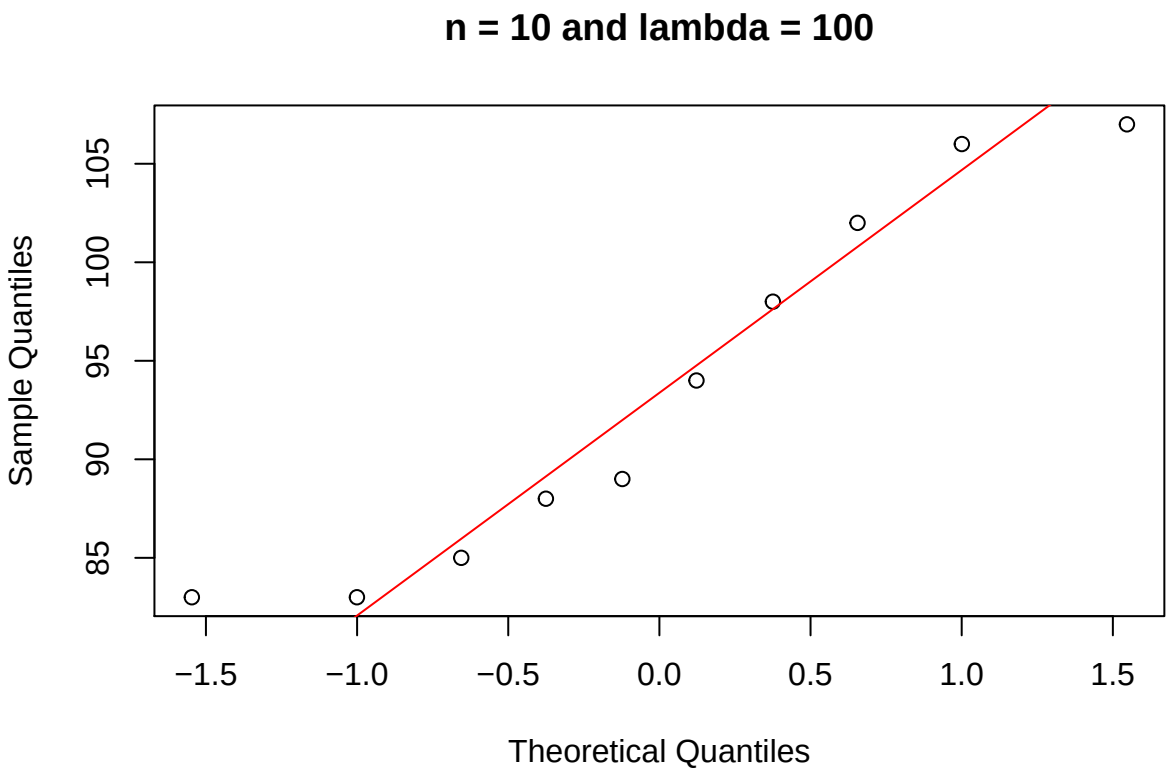
```
x = poisson_st[[4]]
x
```

```
##  [1]  98 107  85  88  94  83 102 106  83  89
```

```
hist(x, main = "Histogram with n = 10 and lambda = 100")
```

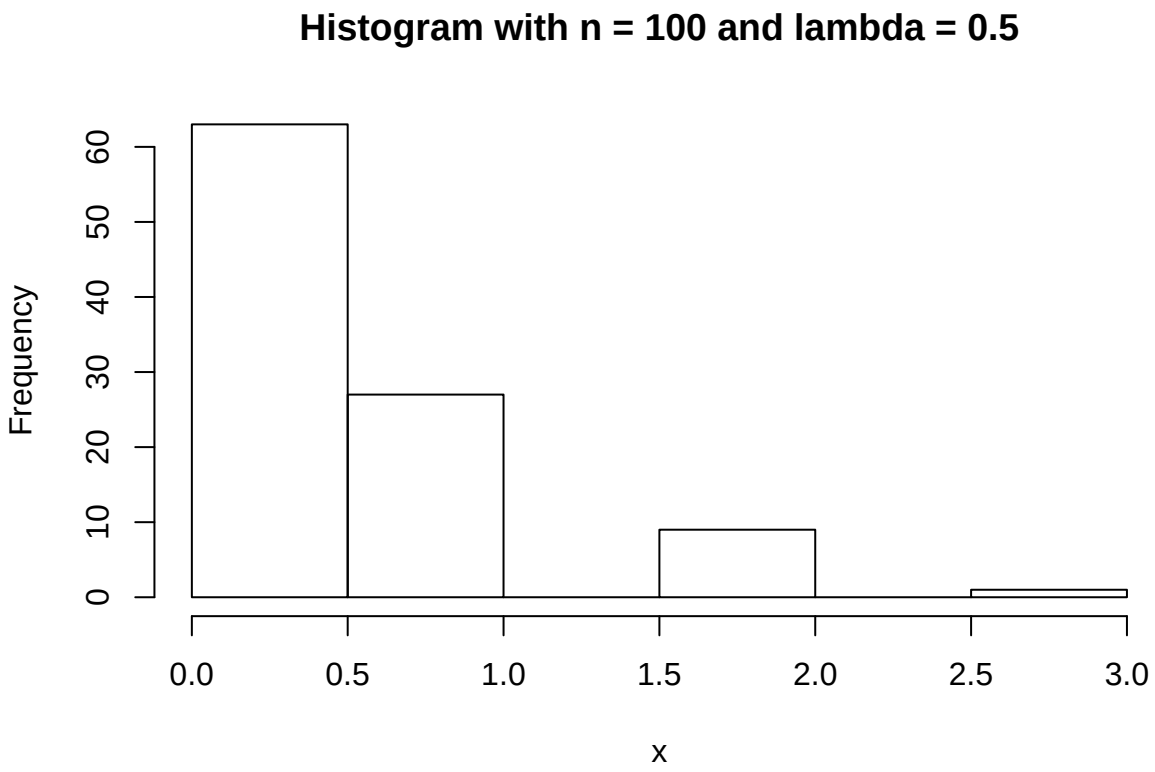



```
qqnorm(x, main = "n = 10 and lambda = 100")
qqline(x, col= "red")
```

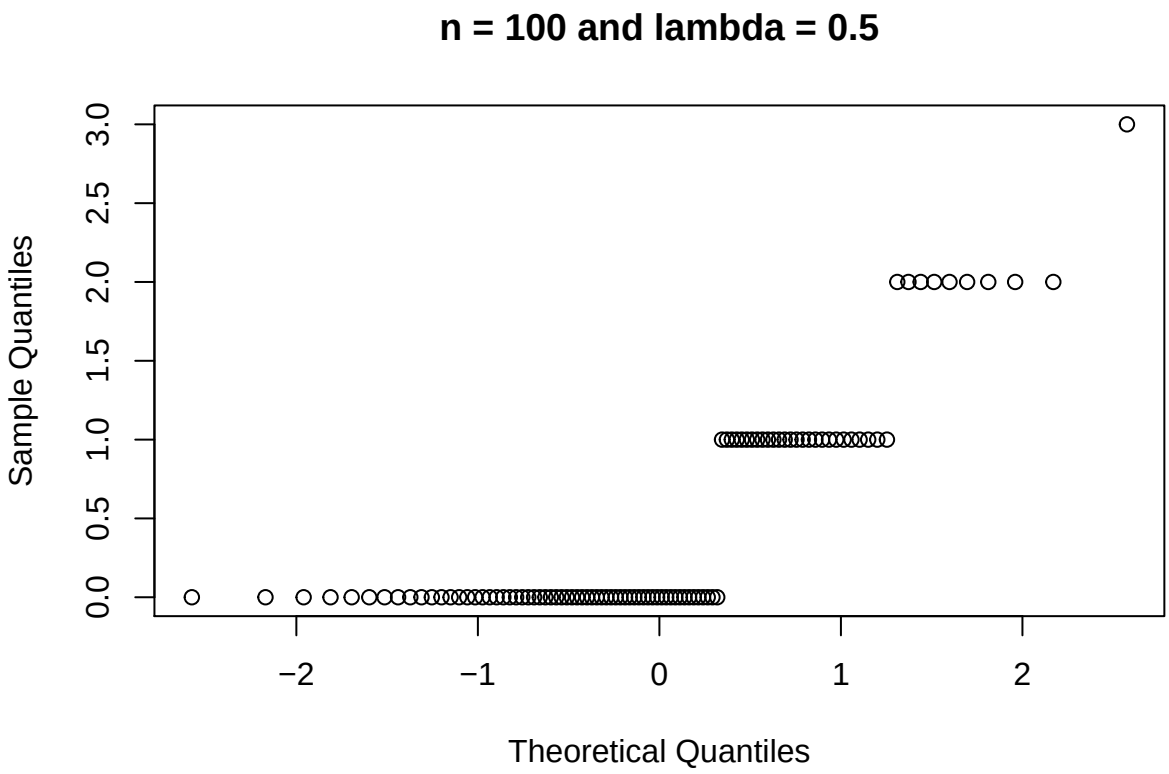


The histogram and QQ-plot resemble normality.

```
x = poisson_st[[5]]
hist(x, main = "Histogram with n = 100 and lambda = 0.5")
```

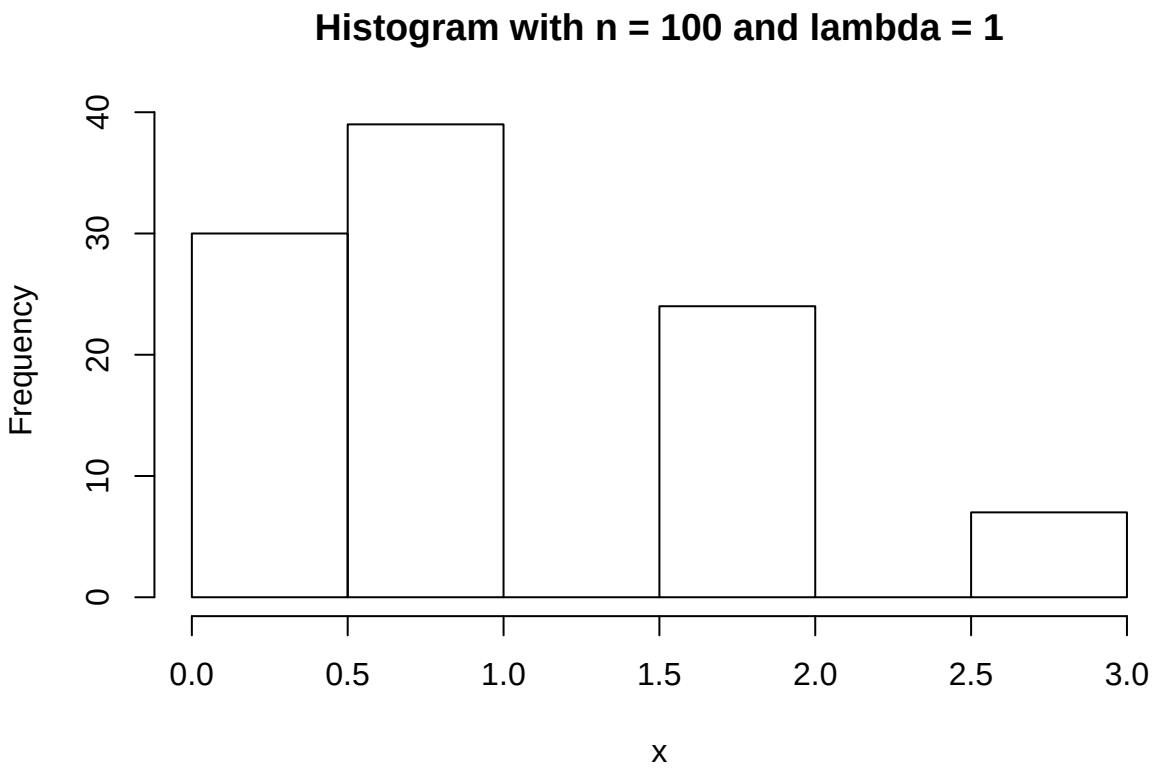



```r
qqnorm(x, main = "n = 100 and lambda = 0.5")
```

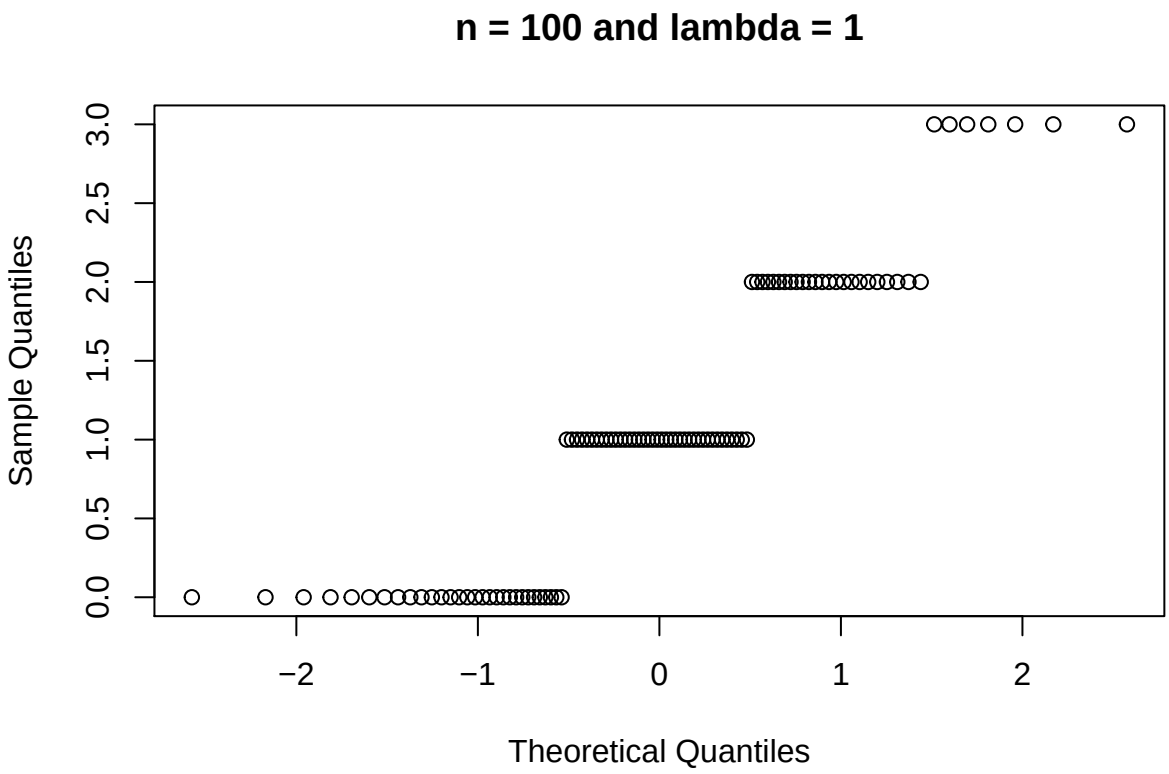


The distribution doesn't look normal.

```
x = poisson_st[[6]]
hist(x, main = "Histogram with n = 100 and lambda = 1")
```

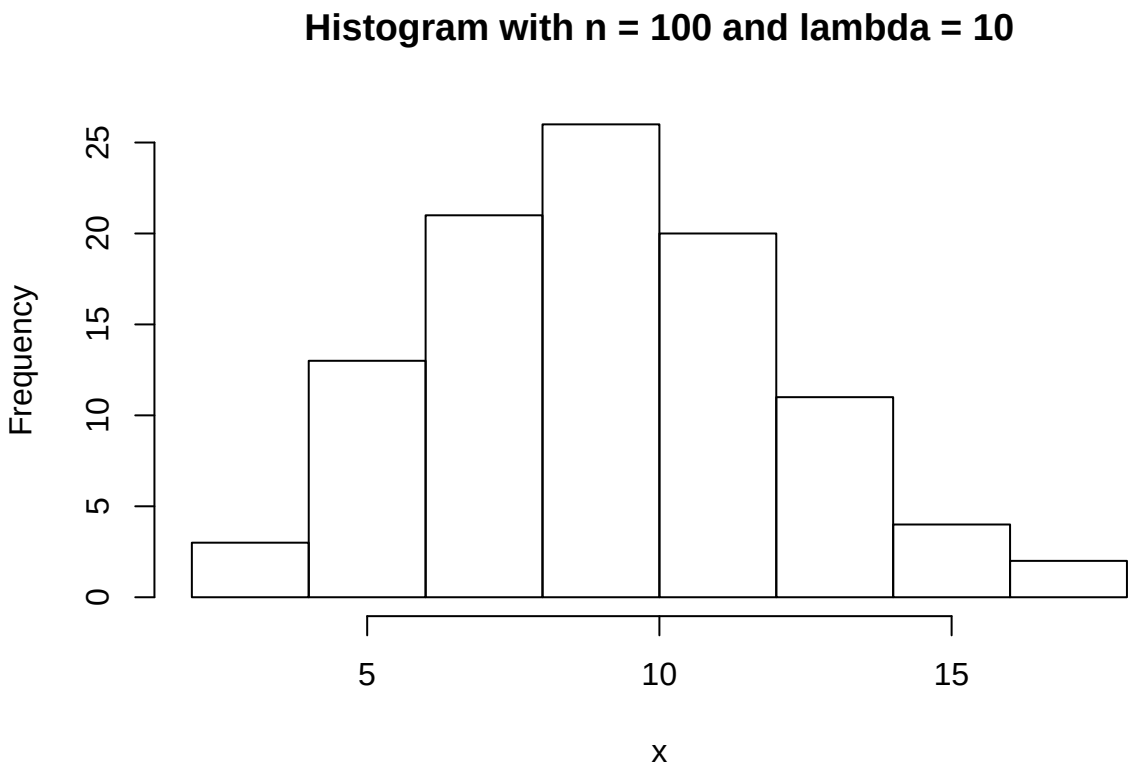



```
qqnorm(x, main = "n = 100 and lambda = 1")
```

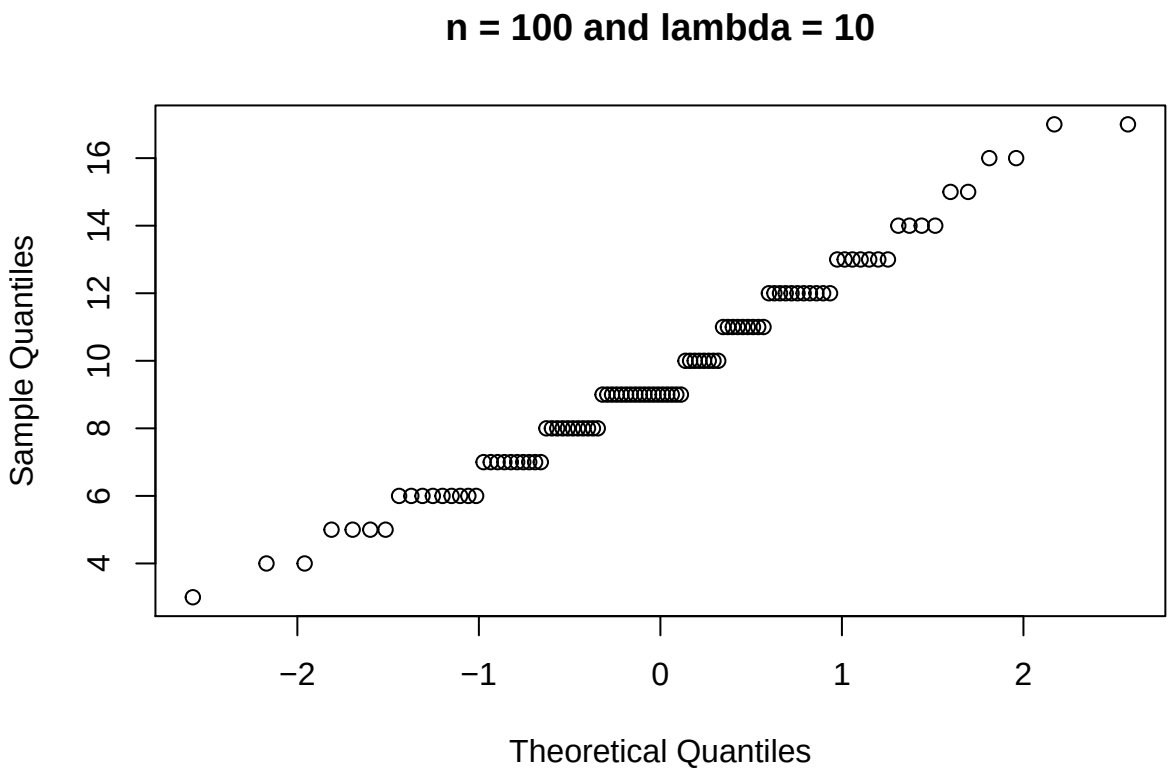


The distribution doesn't look normal.

```
x = poisson_st[[7]]
hist(x, main = "Histogram with n = 100 and lambda = 10")
```

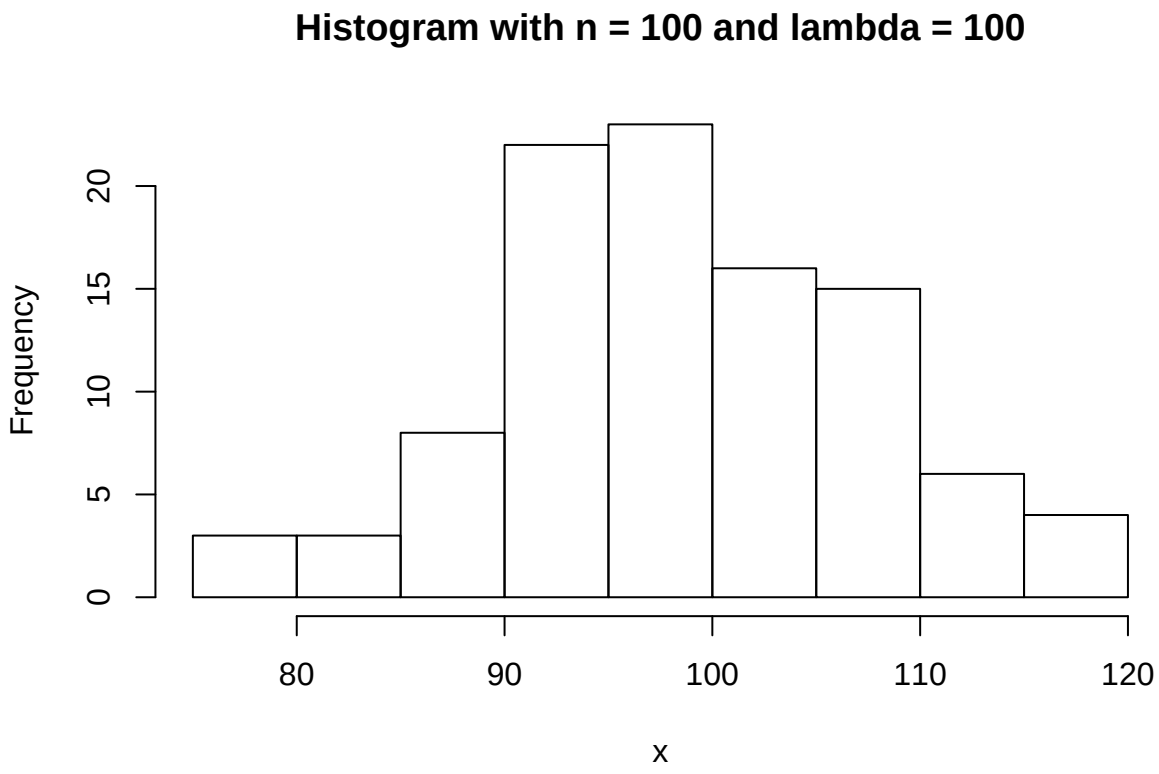



```
qqnorm(x, main = "n = 100 and lambda = 10")
```

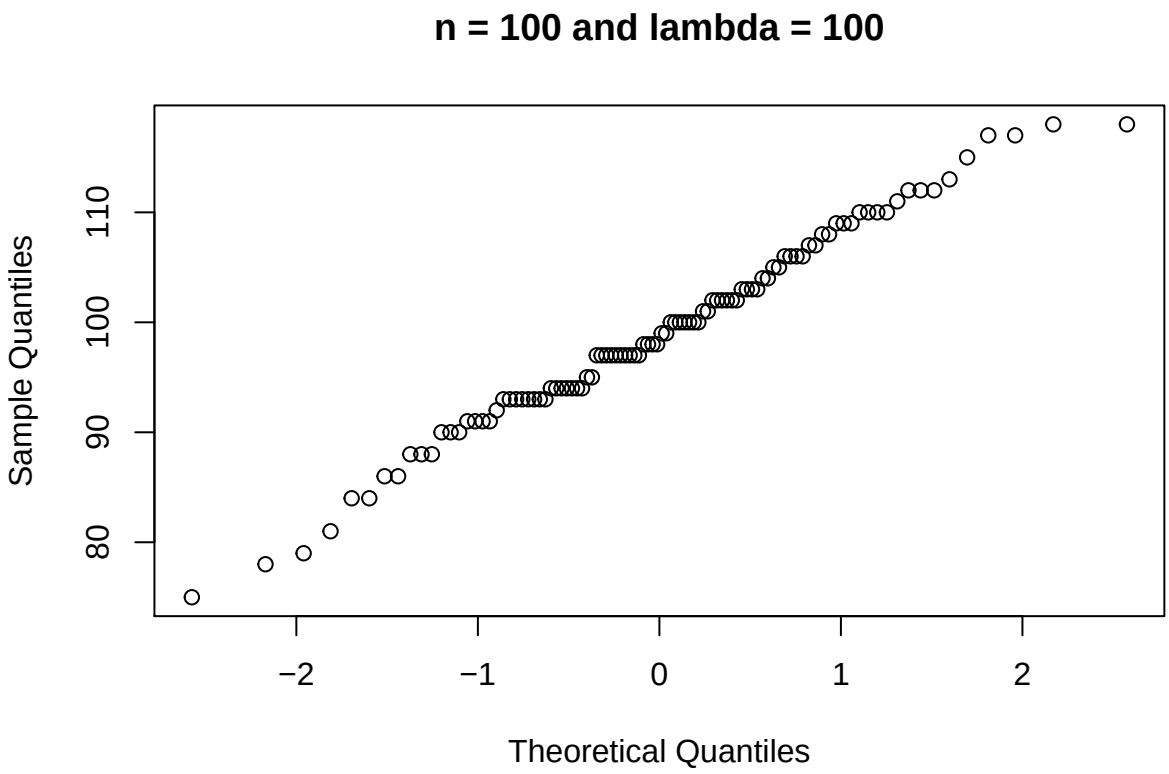


The histogram could be close to normal but the QQ-plot is not.

```
x = poisson_st[[8]]
hist(x, main = "Histogram with n = 100 and lambda = 100")
```

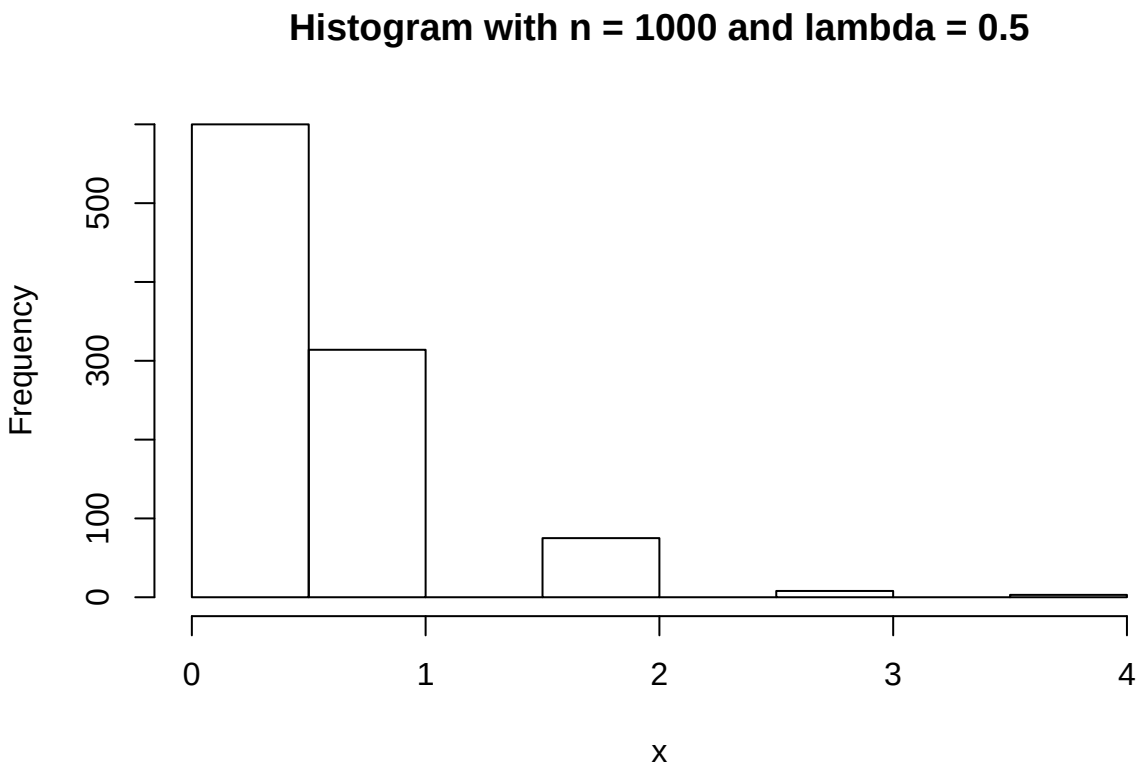



```
qqnorm(x, main = "n = 100 and lambda = 100")
```

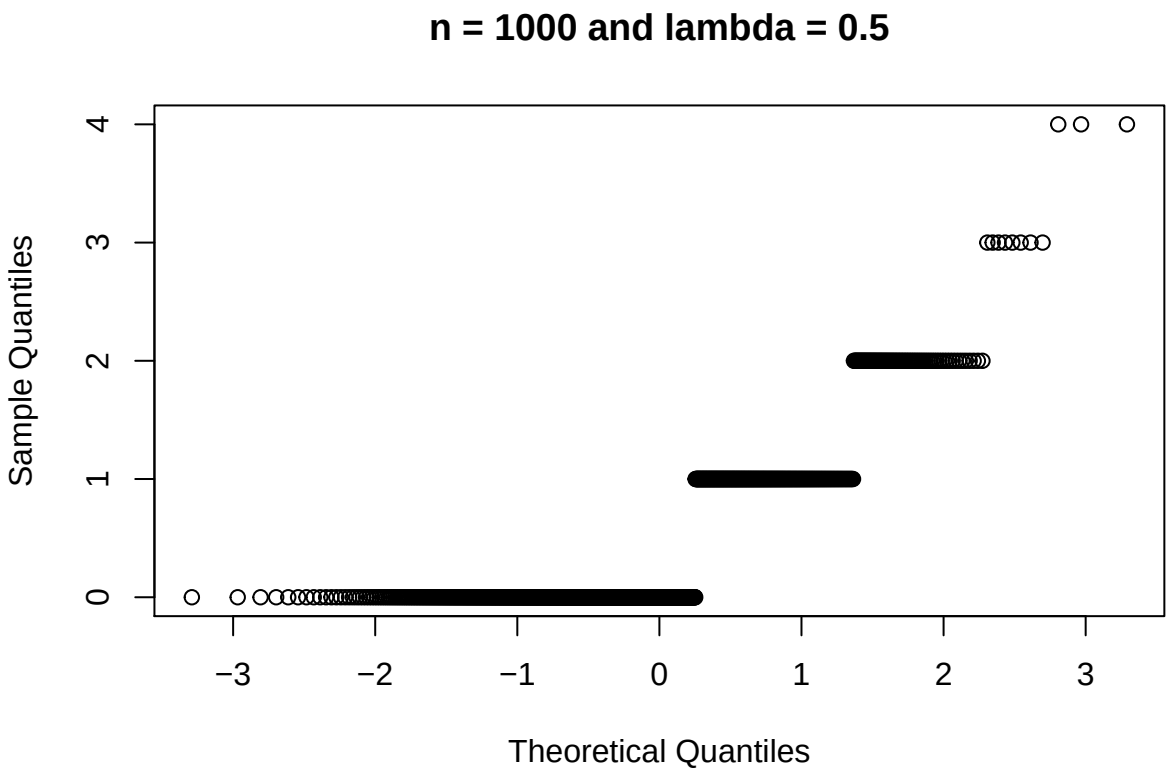


The histogram looks normal but the QQ-plot seems to follow a stepped pattern.

```
x = poisson_st[[9]]
hist(x, main = "Histogram with n = 1000 and lambda = 0.5")
```

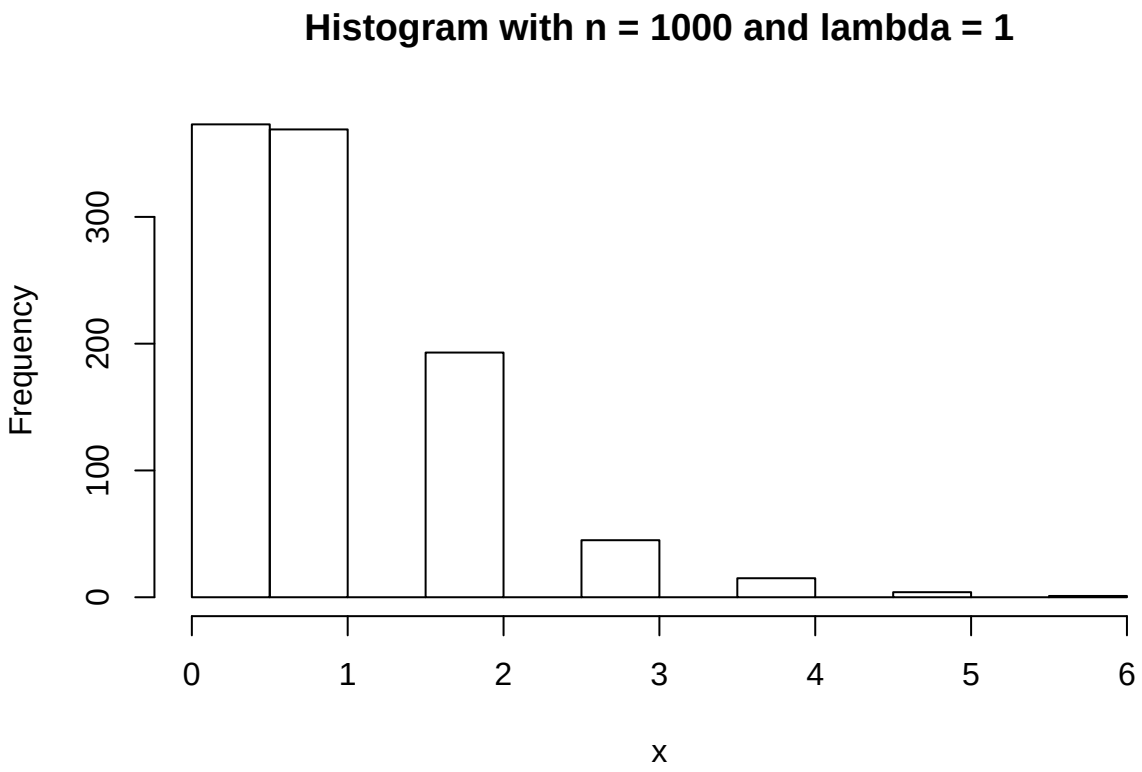



```
qqnorm(x, main = "n = 1000 and lambda = 0.5")
```

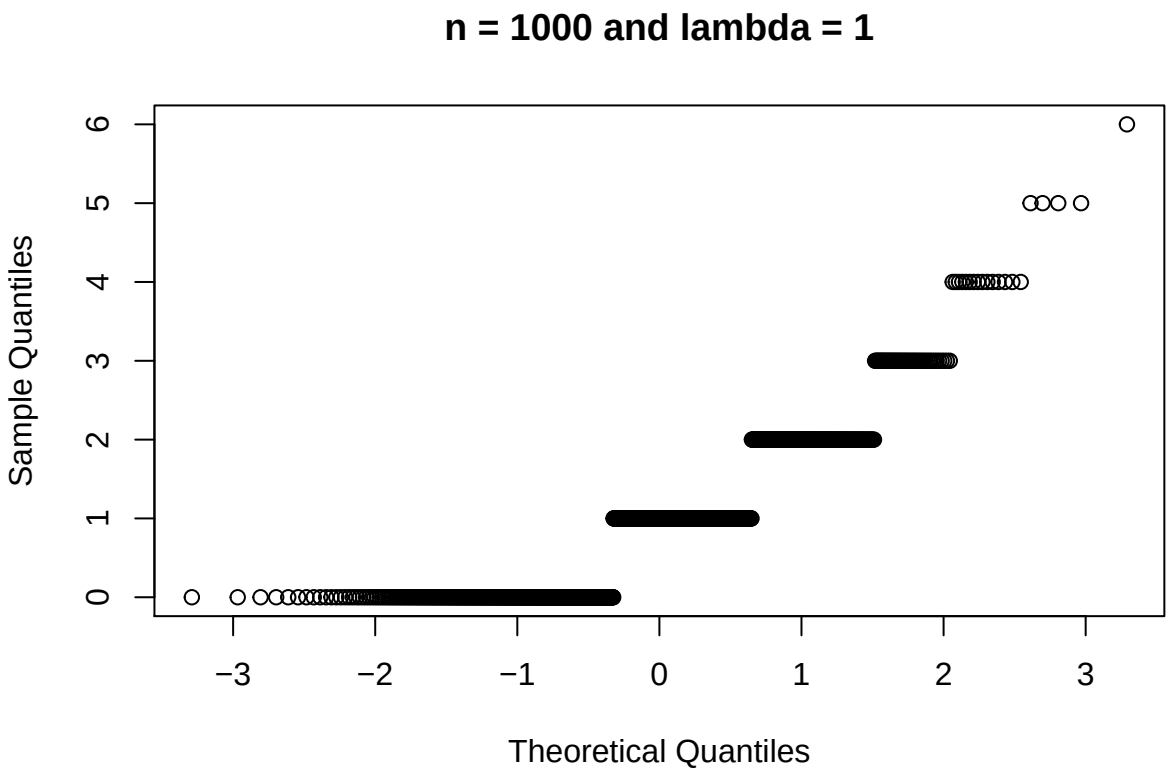


The distribution doesn't look normal.

```
x = poisson_st[[10]]
hist(x, main = "Histogram with n = 1000 and lambda = 1")
```

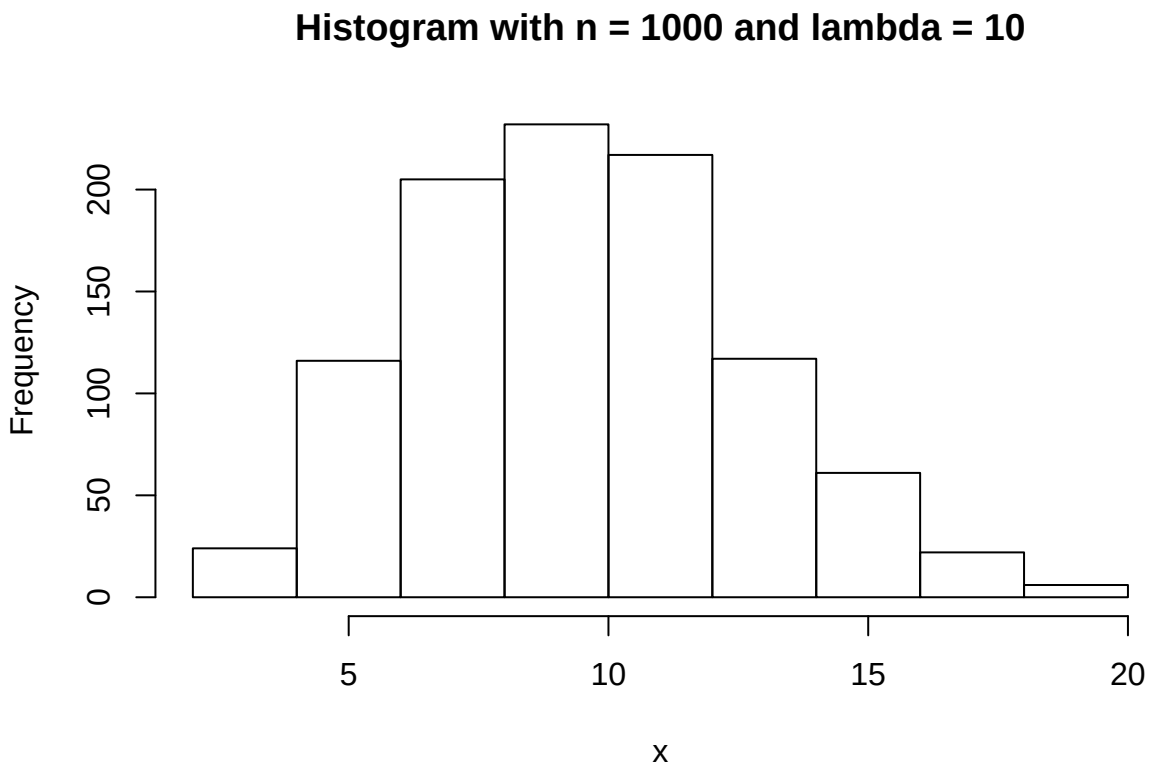



```
qqnorm(x, main = "n = 1000 and lambda = 1")
```

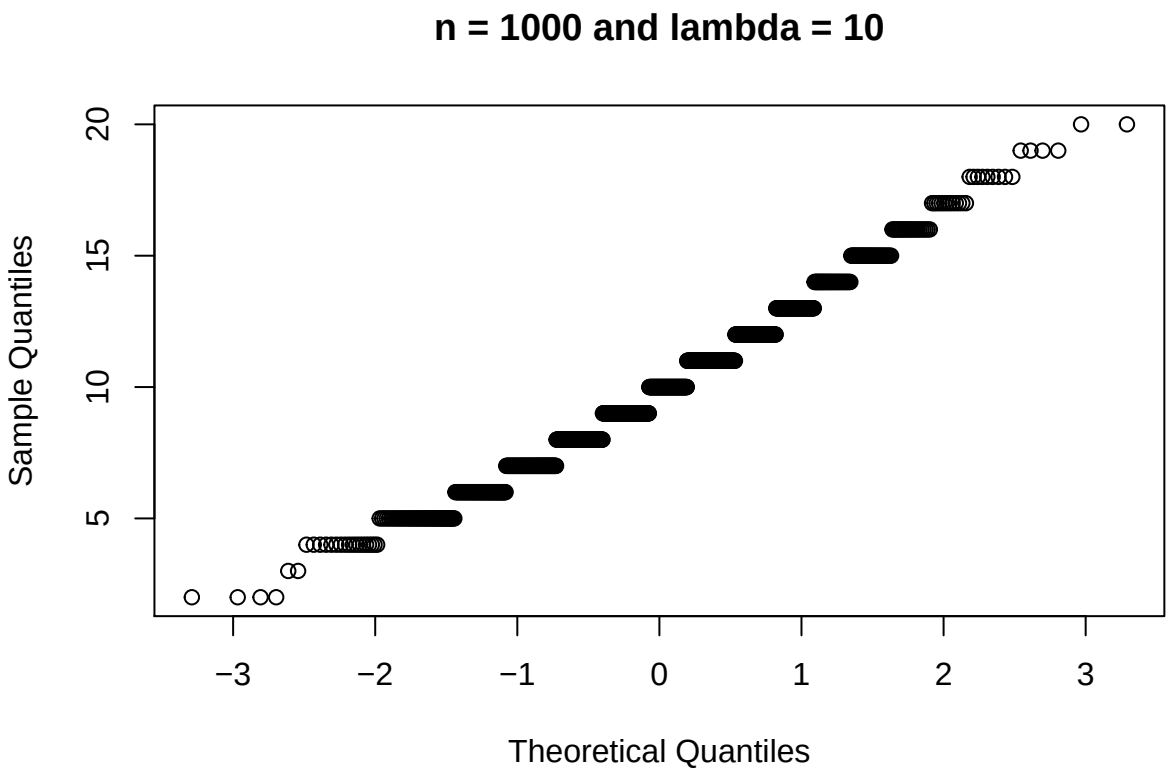


The distribution doesn't look normal.

```
x = poisson_st[[11]]
hist(x, main = "Histogram with n = 1000 and lambda = 10")
```

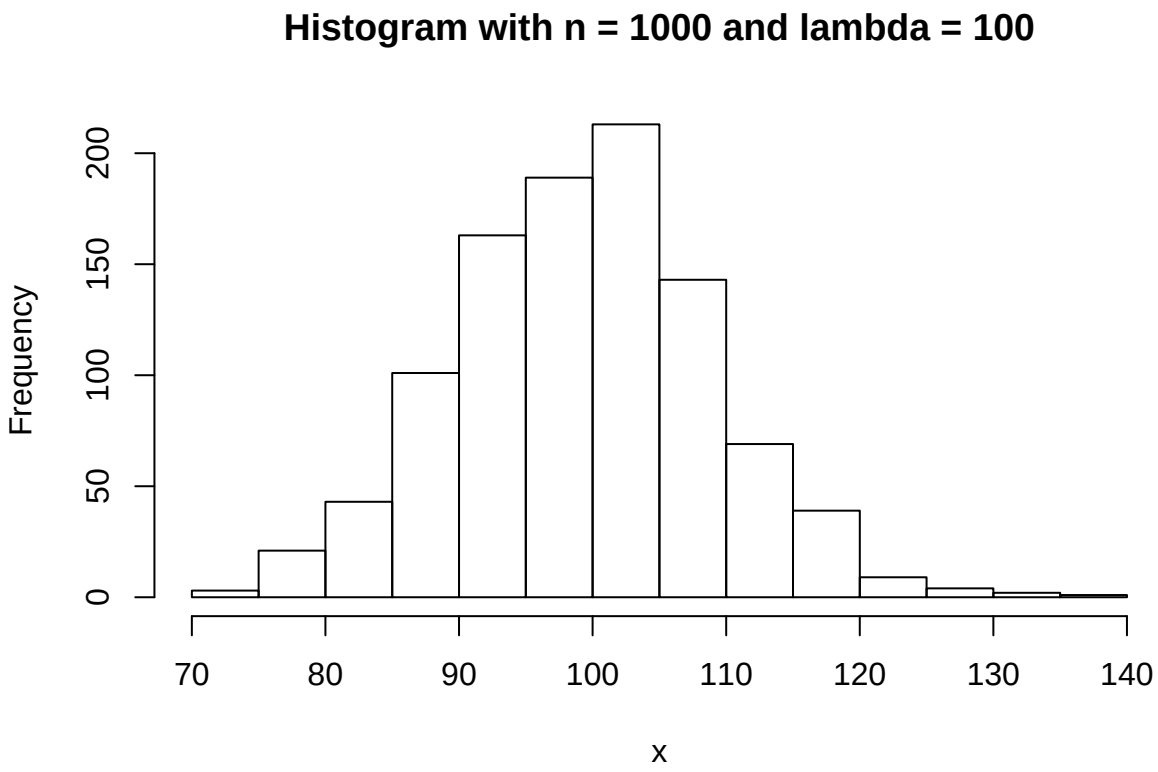



```
qqnorm(x, main = "n = 1000 and lambda = 10")
```

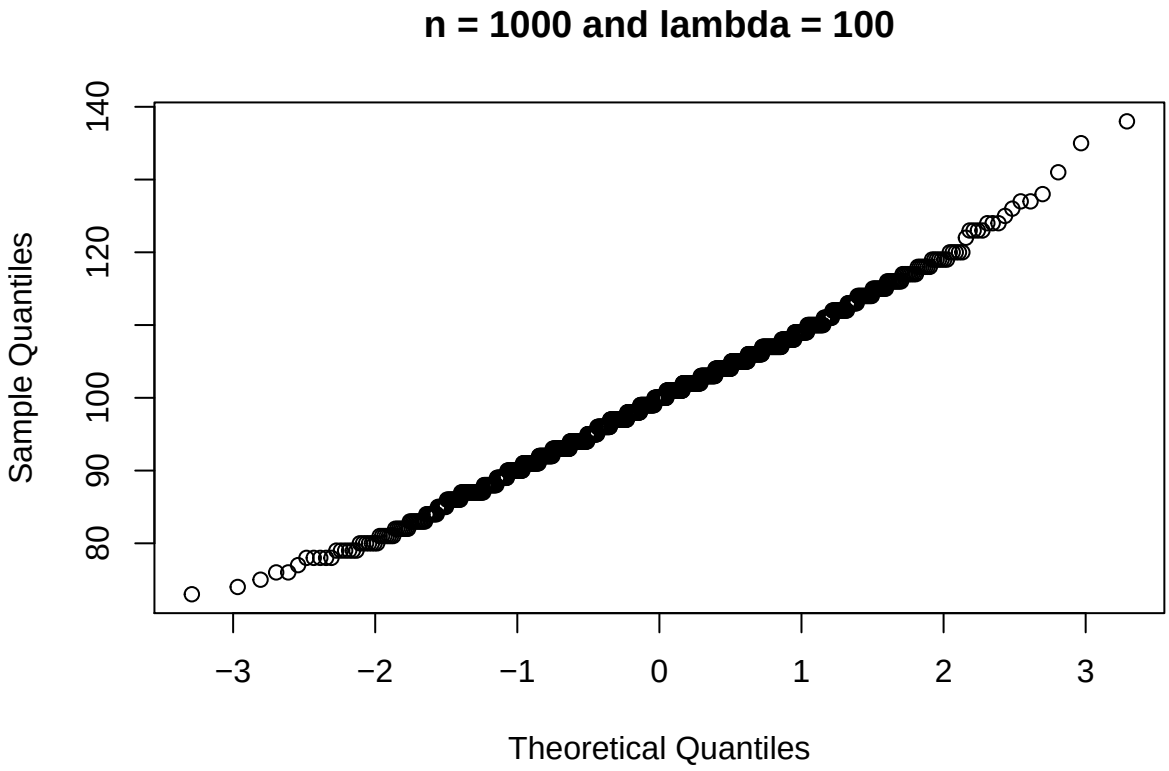


Histogram shows normality but the QQ-plot shows a stepped pattern.

```
x = poisson_st[[12]]
hist(x, main = "Histogram with n = 1000 and lambda = 100")
```




```
qqnorm(x, main = "n = 1000 and lambda = 100")
```



Both histogram and QQ-plot show normality.

Findings

We can see from the previous histograms and QQ-plots that, when increasing the number of samples and lambda values of the Poisson distribution, the histograms look aproximately equal to normal distributions. However, looking at the QQ-plot we see that for some combinations of n and lambda (i.e n=10 l=100) they look normal but in most of the other combinations we see a stepped pattern.

Task 2

The mean and variance of the Poisson distribution both equal lambda. Hence, the larger the parameter, the larger the values of Y on average and the larger the spread in the values of Y.

If we look into the Poisson distributions that we generated above in task 1, we can see that for larger lambda values we get a distribution which seems more like a normal distribution.

In poisson regression, for each observation Y the parameter lambda is modelled differently, since the corresponding values will differ in general. The variances are different as well.

Therefore, we believe that they're not in the same location-scale family.

Task 3

We perform Poisson regression on the data.

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

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties + pctvote +
##     popn + size + numelec + numregim, family = poisson, data = africa_data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.344  -0.954  -0.259   0.391   1.695
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.510269   0.905330   -0.56   0.5730
## oligarchy    0.073081   0.034596    2.11   0.0346 *
## pollib      -0.712978   0.272563   -2.62   0.0089 **
## parties      0.030774   0.011187    2.75   0.0059 **
## pctvote      0.013872   0.009753    1.42   0.1549
## popn         0.009343   0.006595    1.42   0.1566
## size        -0.000190   0.000248   -0.76   0.4445
## numelec     -0.016078   0.065484   -0.25   0.8060
## numregim     0.191735   0.229289    0.84   0.4030
## ---
## 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: 111.5
##
## Number of Fisher Scoring iterations: 6
```

Task 4

Using the step-down approach, we will reduce the number of variables in our model.

```
africa_glm_sd = glm(miltcoup~oligarchy+pollib+parties+pctvote+popn+size+numelec+numregim, famil
summary(africa_glm_sd)
```

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

```
##     popn + size + numelec + numregim, family = poisson, data = africa_data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.344  -0.954  -0.259   0.391   1.695
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.510269   0.905330   -0.56   0.5730
## oligarchy    0.073081   0.034596    2.11   0.0346 *
## pollib      -0.712978   0.272563   -2.62   0.0089 **
## parties      0.030774   0.011187    2.75   0.0059 **
## pctvote      0.013872   0.009753    1.42   0.1549
## popn         0.009343   0.006595    1.42   0.1566
## size        -0.000190   0.000248   -0.76   0.4445
## numelec     -0.016078   0.065484   -0.25   0.8060
## numregim     0.191735   0.229289    0.84   0.4030
## ---
## 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: 111.5
##
## Number of Fisher Scoring iterations: 6
```

```
summary(africa_glm_sd)[[12]][35]
```

```
## [1] 0.806
```

As we can see, the variable numelec has the highest p-value and since it is > 0.05, we discard it for the next iteration.

```
africa_glm_sd = glm(miltcoup~oligarchy+pollib+parties+pctvote+popn+size+numregim, family = pois
summary(africa_glm_sd)
```

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties + pctvote +
##     popn + size + numregim, family = poisson, data = africa_data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.400  -0.938  -0.267   0.422   1.700
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.607803   0.823927   -0.74   0.4607
```

```
## oligarchy    0.078137   0.027766    2.81   0.0049 **
## pollib      -0.677390   0.229013   -2.96   0.0031 **
## parties      0.029679   0.010289    2.88   0.0039 **
## pctvote      0.013129   0.009289    1.41   0.1576
## popn         0.008931   0.006375    1.40   0.1612
## size        -0.000202   0.000244   -0.83   0.4068
## numregim     0.175820   0.221050    0.80   0.4264
## ---
## 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: 109.5
##
## Number of Fisher Scoring iterations: 5
```

```
summary(africa_glm_sd)[[12]][32]
```

```
## [1] 0.426
```

In this iteration, numregim has the highest p-value and it is > 0.05. Hence, we discard it for the next iteration.

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

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties + pctvote +
##     popn + size, family = poisson, data = africa_data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.352  -0.965  -0.195   0.483   1.618
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.112687   0.516303   -0.22  0.82723
## oligarchy    0.085962   0.025910    3.32  0.00091 ***
## pollib      -0.689403   0.227857   -3.03  0.00248 **
## parties      0.029194   0.010195    2.86  0.00419 **
## pctvote      0.014159   0.009198    1.54  0.12372
## popn         0.006274   0.005399    1.16  0.24527
## size        -0.000195   0.000242   -0.80  0.42138
## ---
## 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: 108.2
##
## Number of Fisher Scoring iterations: 5
```

```
summary(africa_glm_sd)[[12]][28]
```

```
## [1] 0.421
```

Here, size has the highest p-value . As we can see, this value is > 0.05. This means we will discard it for our model and proceed to the next iteration.

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

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties + pctvote +
##     popn, family = poisson, data = africa_data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.411  -0.994  -0.140   0.552   1.613
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.24447    0.49571   -0.49   0.6219
## oligarchy    0.08317    0.02544    3.27   0.0011 **
## pollib      -0.65283    0.22123   -2.95   0.0032 **
## parties      0.02980    0.01029    2.89   0.0038 **
## pctvote      0.01384    0.00928    1.49   0.1359
## popn         0.00559    0.00538    1.04   0.2988
## ---
## 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: 106.9
##
## Number of Fisher Scoring iterations: 5
```

```
summary(africa_glm_sd)[[12]][24]
```

```
## [1] 0.299
```

We pick the highest p-value, the one of the variable popn. This value is > 0.05, so discard it for the next iteration.

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

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties + pctvote,
##     family = poisson, data = africa_data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.546  -0.984  -0.188   0.595   1.671
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.09366    0.46328   -0.20   0.8398
## oligarchy    0.09536    0.02242    4.25  2.1e-05 ***
## pollib      -0.66661    0.21756   -3.06   0.0022 **
## parties      0.02563    0.00950    2.70   0.0070 **
## pctvote      0.01213    0.00906    1.34   0.1803
## ---
## 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: 105.9
##
## Number of Fisher Scoring iterations: 5
```

```r
summary(africa_glm_sd)[[12]][20]
```

```
## [1] 0.18
```

Of all the variables, pctvote has the highest p-value. It is > 0.05. So we discard it for the next iteration.

```r
africa_glm_sd = glm(miltcoup~oligarchy+pollib+parties, family = poisson, data = africa_data)
summary(africa_glm_sd)
```

```
##
## Call:
## glm(formula = miltcoup ~ oligarchy + pollib + parties, family = poisson,
##     data = africa_data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.358  -1.042  -0.286   0.628   1.752
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.25138    0.37269    0.67    0.500
## oligarchy    0.09262    0.02178    4.25  2.1e-05 ***
## pollib      -0.57410    0.20438   -2.81    0.005 **
## parties      0.02206    0.00896    2.46    0.014 *
## ---
## 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: 105.7
##
## Number of Fisher Scoring iterations: 5
```

As we can see, all the p-values are now smaller than 0.05. Meaning that all the variables are significant for our model.

The resulting model of the step-down method is:

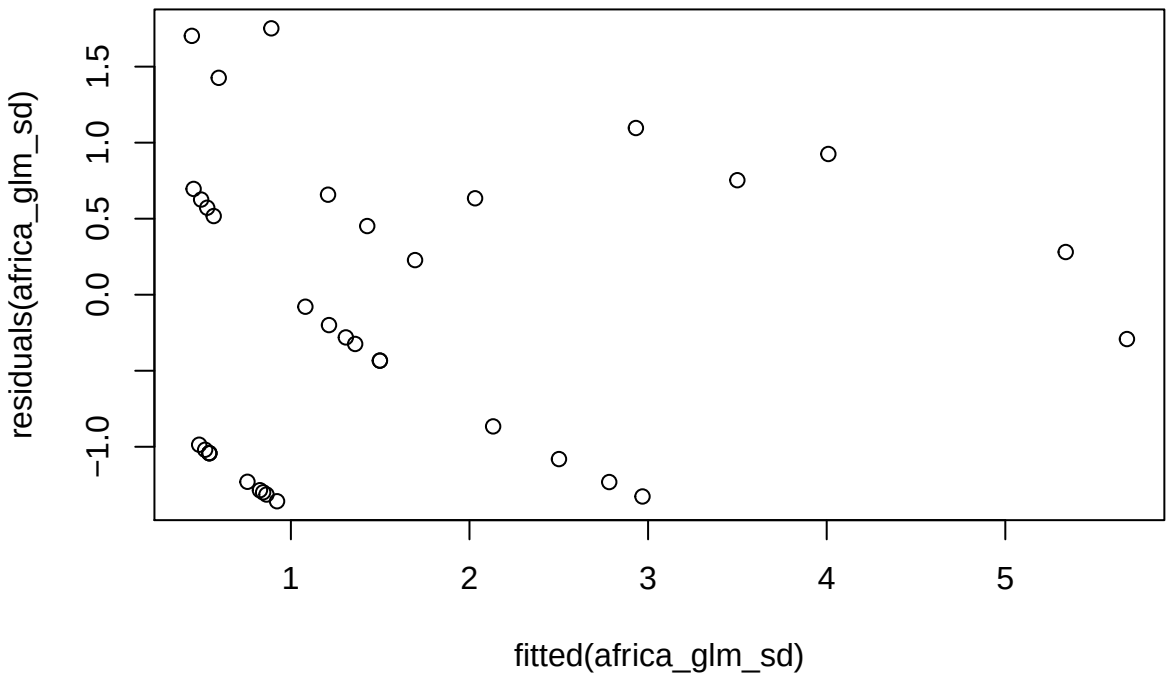
miltcoup = 0.25138 + 0.09262*oligarchy* - *0.57410*pollib + 0.02206*parties + error

Task 5

Next, we will show some diagnostic plots for our model.

Plots: Fitted vs Residuals

```
plot(fitted(africa_glm_sd), residuals(africa_glm_sd))
```

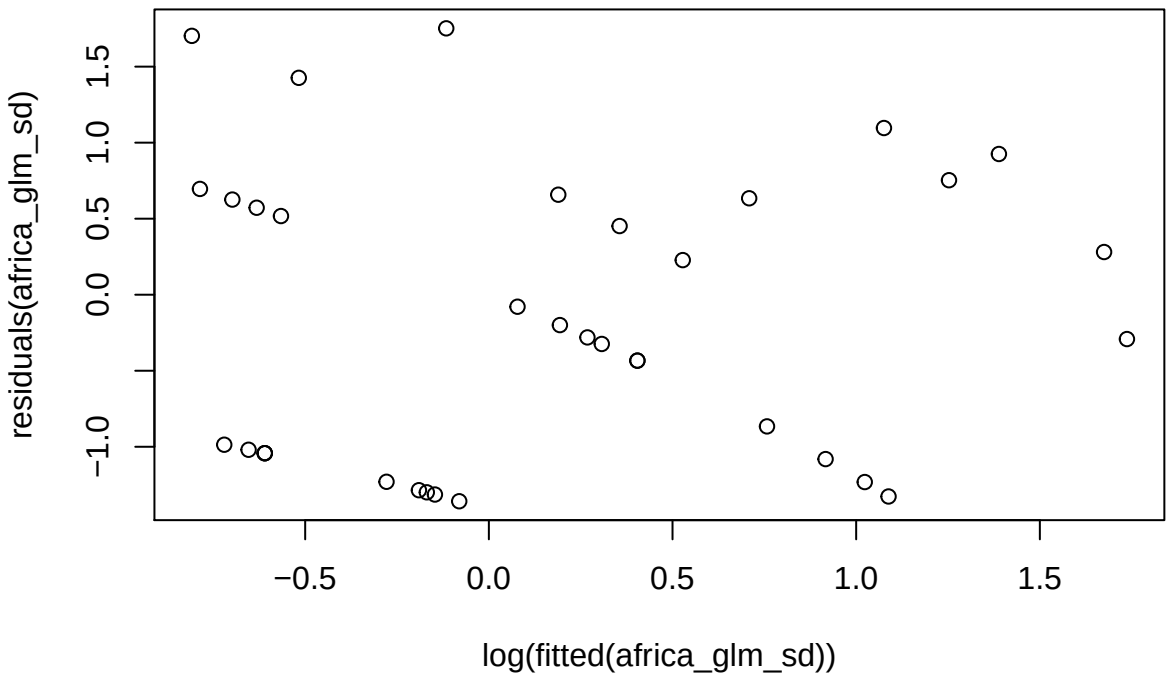


We can't recognize any specific pattern. Data is scattered, and it's not visually good as it would suppose to be in linear regression model.

Because of this, we will take the logarithm to make the x-values fitted by a linear function in the plot.

Plots: Logarithmic-Fitted values vs Residuals

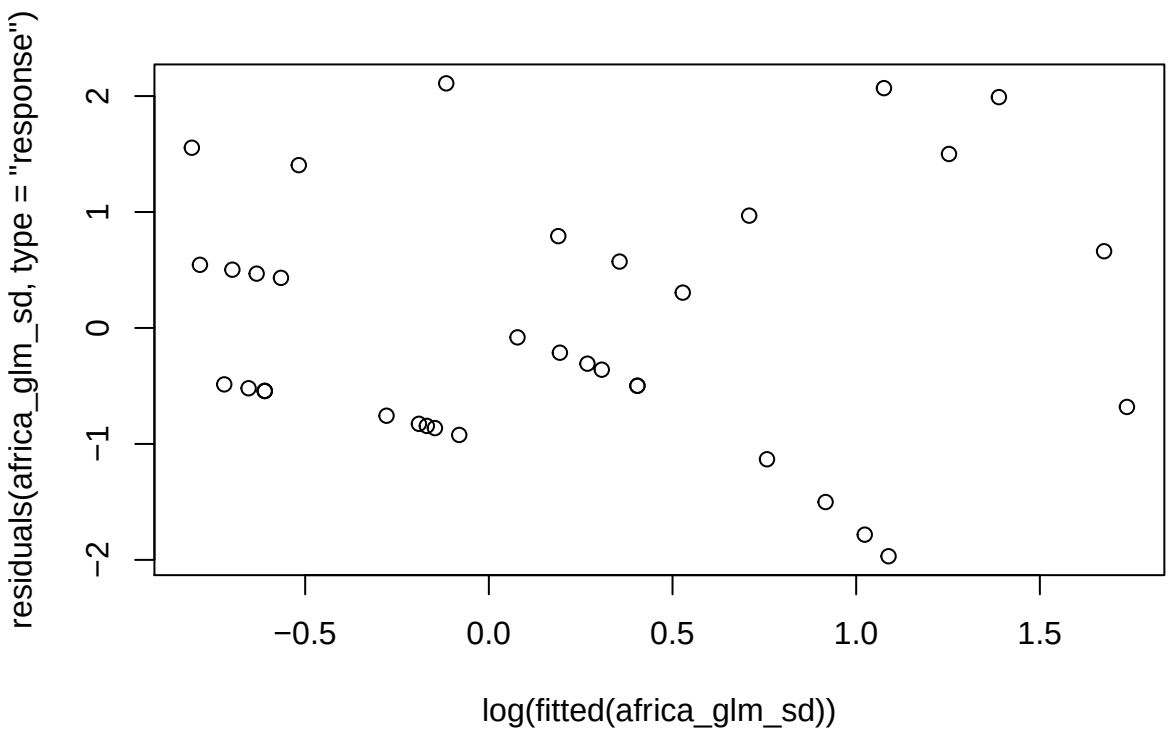
```
plot(log(fitted(africa_glm_sd)), residuals(africa_glm_sd))
```



The plot seems OK yet still no specific pattern. And still looks scattered.

Plots: Logarithmic-Fitted values vs Residuals with type = Response

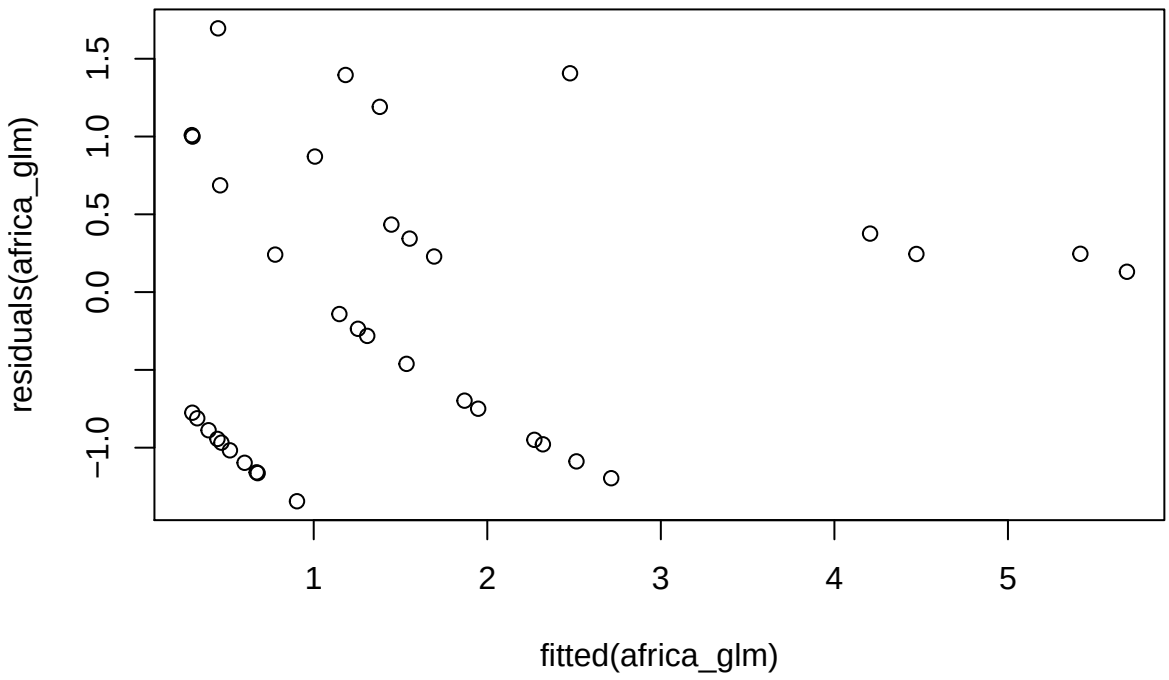
```
plot(log(fitted(africa_glm_sd)), residuals(africa_glm_sd, type = "response"))
```



Check Lec 10 slide: 41-43

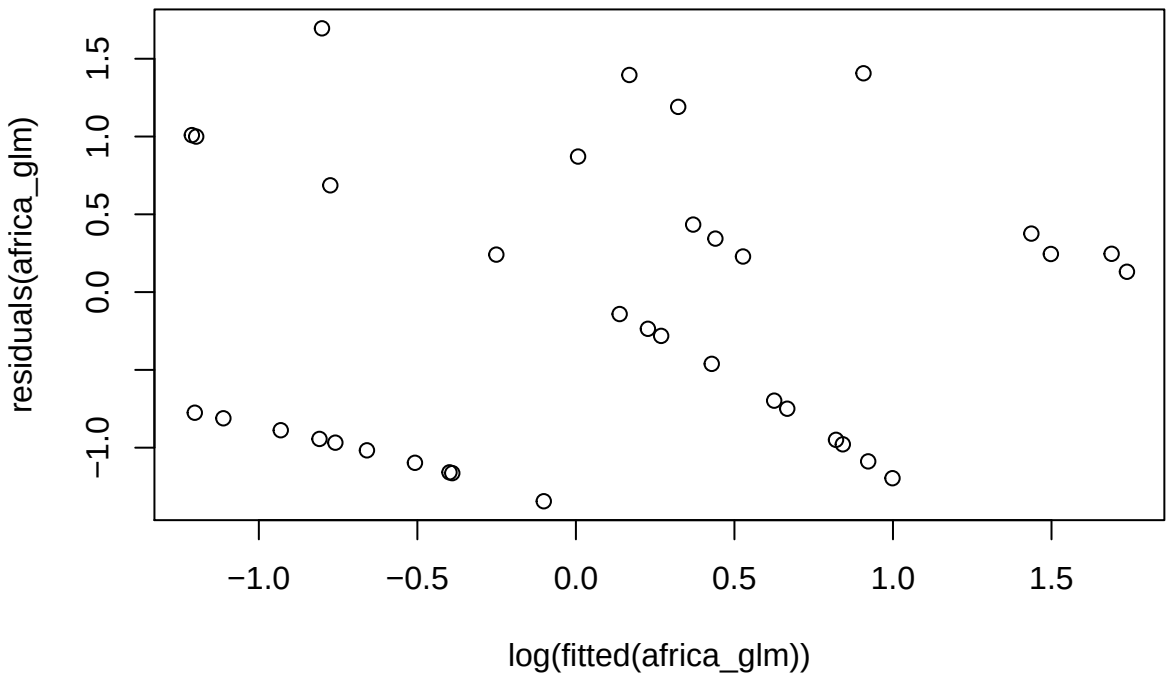
Plots: Fitted vs Residuals in the Full Model

```
plot(fitted(africa_glm), residuals(africa_glm))
```



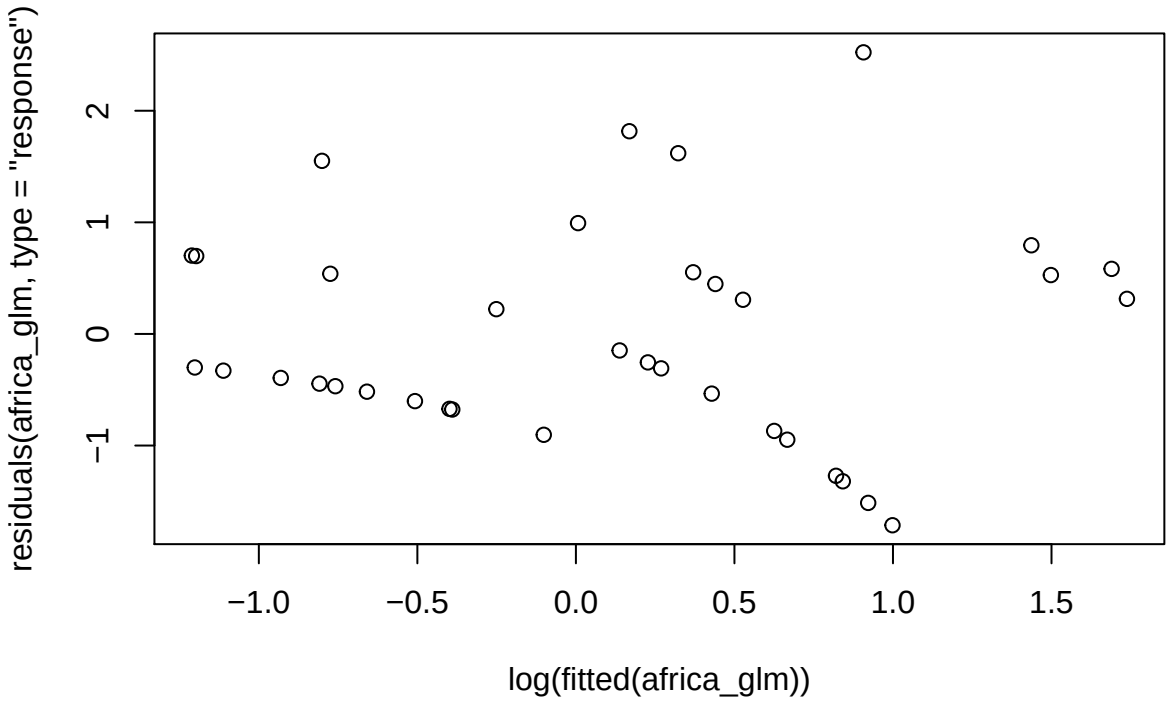
Plots: Logarithmic-Fitted values vs Residuals in the Full Model

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



Plots: Logarithmic-Fitted values vs Residuals with type = Response in the Full Model

```
plot(log(fitted(africa_glm)), residuals(africa_glm, type = "response"))
```



The plots look scattered. All of them follow a certain pattern in our model from task-4. After ploting using the full model, we found the same pattern there too.

Therefore, we can discard that the readon is that we deleted too many variables.

Now we will check normality assumption.

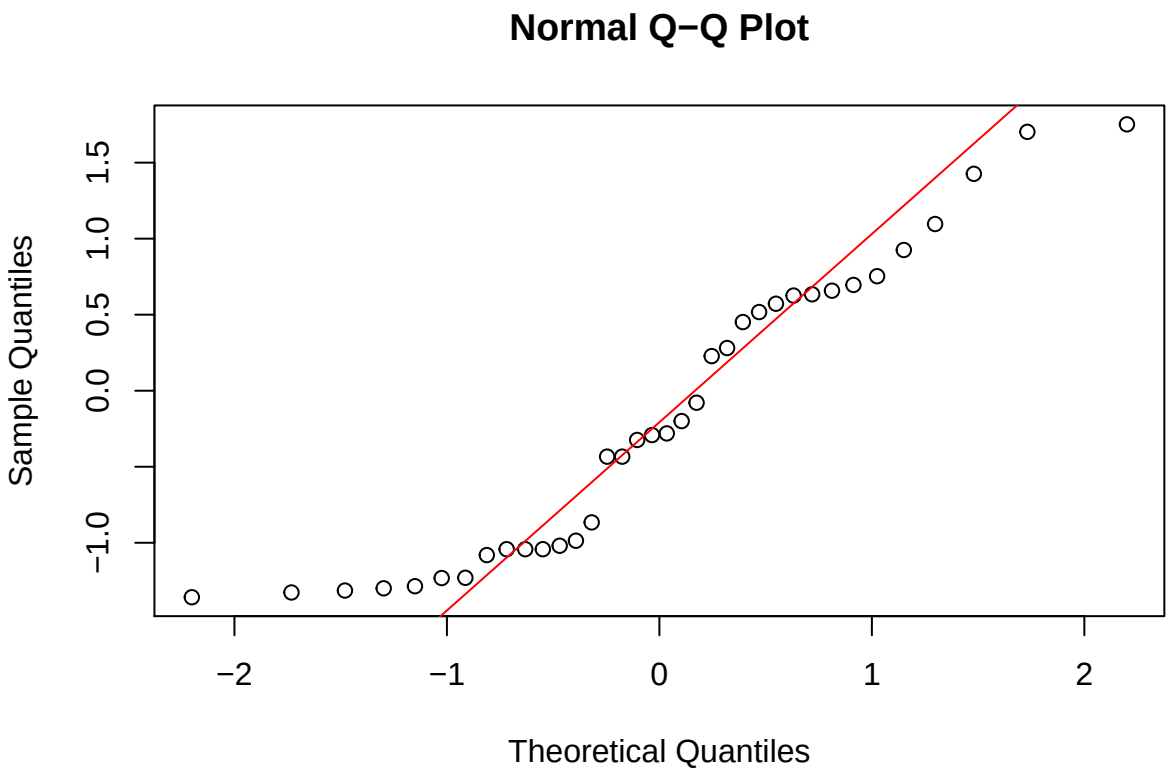
```
shapiro.test(residuals(africa_glm_sd)) # p-value = 0.01 < 0.05 # Normality is doubtful
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(africa_glm_sd)
## W = 0.9, p-value = 0.01
```

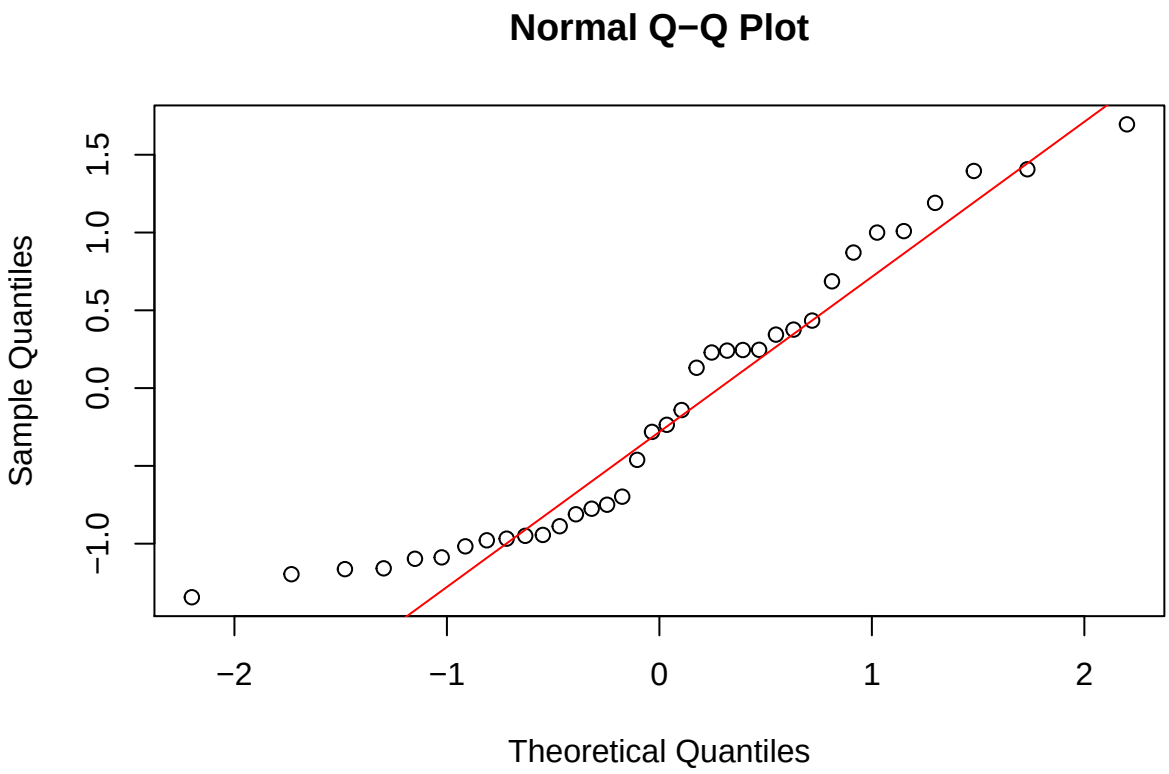
```
shapiro.test(residuals(africa_glm)) # p-value = 0.01 < 0.05 # Normaility is doubtful
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(africa_glm)
## W = 0.9, p-value = 0.01
```

```
qqnorm(residuals(africa_glm_sd)) # doesn't seem normal
qqline(residuals(africa_glm_sd), col="red")
```




```
qqnorm(residuals(africa_glm)) # doesn't seem normal
qqline(residuals(africa_glm), col="red")
```



In both models we got the same results. The plots that we generated for the model that we found in task-4 and for the full model with all explanatory variables followed same type of pattern. For these reasons, the assumption of normality (if any) is doubtful.

This means that the sample might not come from a normal distribution.