Exam 2

Netid: ghosh17

Subhankar Ghosh

Question 1

What is this data about?

We will be looking at the Africa dataset in Faraway package, let us know more about the data.

Africa dataset is an outcome of a study on the factors affecting regime stability in Sub-Saharan Africa. This data has 47 observations and 9 variables:

- miltcoup = number of successful military coups from independence to 1989 (*datatype=int*, has to be converted to factor)
- oligarchy = number years country ruled by military oligarchy from independence to 1989 (*datatype=int*)
- pollib = Political liberalization - 0 = no civil rights for political expression, 1 = limited civil rights for expression but right to form political parties, 2 = full civil rights (*datatype=int*, has to be converted to factor)
- parties = Number of legal political parties in 1993 (*datatype=int*)
- pctvote = Percent voting in last election (*datatype=numerical*)
- popn = Population in millions in 1989 (*datatype=numerical*)
- size = Area in 1000 sq km (*datatype=numerical*)
- numelec = Total number of legislative and presidential elections (*datatype=int*)

Let us take a look at the columns in the dataset

```
'data.frame':   47 obs. of  9 variables:
 $ miltcoup : int  0 5 0 6 2 0 1 3 1 2 ...
 $ oligarchy: int  0 7 0 13 13 0 0 14 15 0 ...
 $ pollib   : int  2 1 NA 2 2 2 2 2 2 2 ...
 $ parties  : int  38 34 7 62 10 34 5 14 27 4 ...
 $ pctvote  : num  NA 45.7 20.3 17.5 34.4 ...
 $ popn     : num  9.7 4.6 1.2 8.8 5.3 11.6 0.361 3 5.5 0.458 ...
 $ size     : num  1247 113 582 274 28 ...
 $ numelec  : int  0 8 5 5 3 14 2 6 4 6 ...
 $ numregim : int  1 3 1 3 3 3 1 4 3 2 ...
```

Handling Missing Values

We can see a lot of missing values *(NA)* in the dataset. We would like to know how many incomplete observations do we have?

```
#find number of incomplete cases
sum(!complete.cases(africa))
```

```
## [1] 11
```

There are **11 incomplete observations** in the africa dataset.

We will only be dealing with complete observations that is we would remove the rows having missing values.

Part a

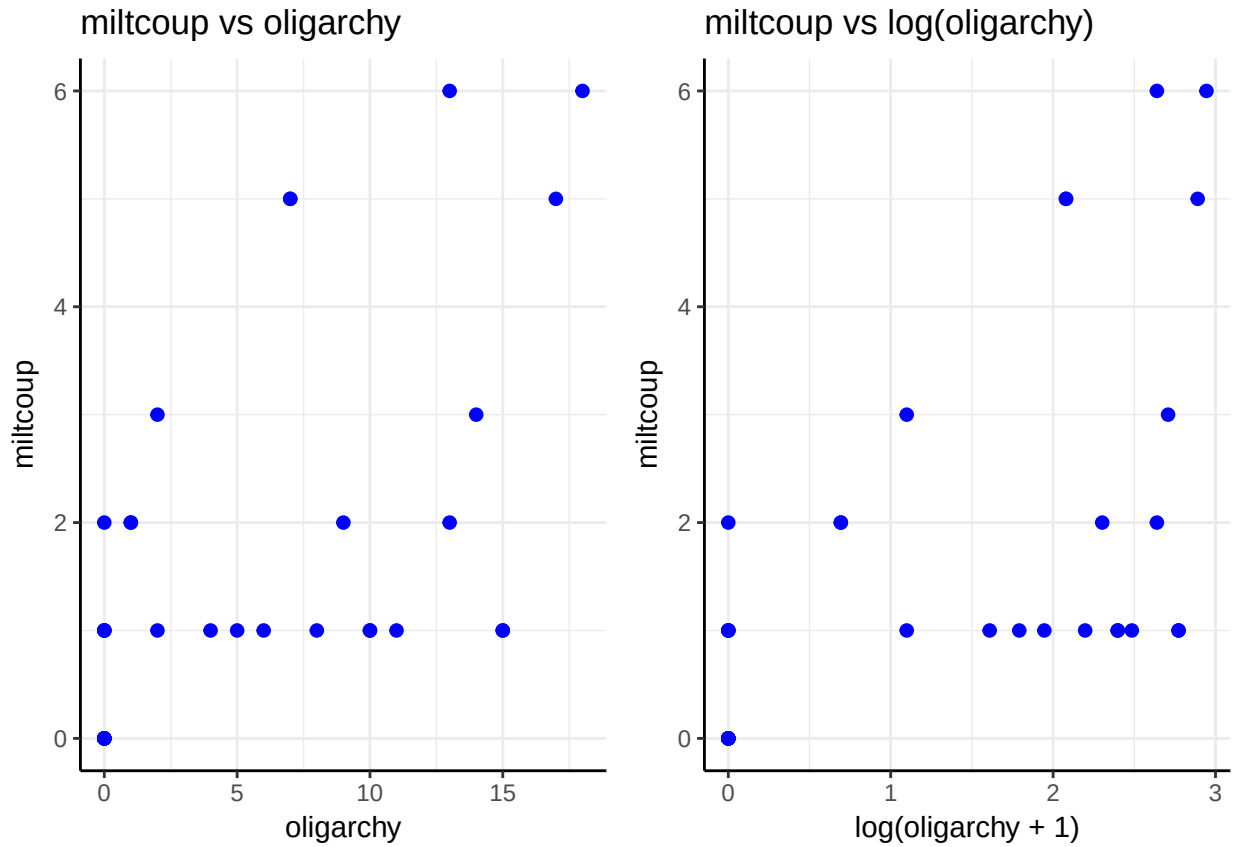
The Initial Model

In this part using an appropriate model selection technique we have to construct a **Poisson regression model** with **miltcoup as the response** and the other variables excluding numregim as possible predictors.

We will first create a full poisson model. A full model is a model that will include all the predictors to predict miltcoup.

We will use the glm function in R, *'family'* parameter is *'poisson'* since this is a poisson model.

Before we fit the model, let us look at the plot of oligarchy vs miltcoup



From the above two plots we can see that log transforming oligarchy predictor brings it in a somewhat linear relationship with miltcoup. Such a relationship was absent in the first plot. Therefore we will use the log transformed oligarchy as a predictor.

```
# Fit poisson model
poisson.mdl = glm(miltcoup ~ log(oligarchy+1) + factor( pollib ) + parties +
                    pctvote + popn + size + numelec,
```

```
                  data = africa2,
                  family = poisson)
```

We will see the summary of the full model we just created.

```
summary(poisson.mdl)
```

```
##
## Call:
## glm(formula = miltcoup ~ log(oligarchy + 1) + factor(pollib) +
##     parties + pctvote + popn + size + numelec, family = poisson,
##     data = africa2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4979  -0.9416  -0.4318   0.6345   1.7041
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -0.2466888  0.9634662  -0.256  0.79792
## log(oligarchy + 1)  0.5069833  0.1817862   2.789  0.00529 **
## factor(pollib)1    -0.7171716  0.6666232  -1.076  0.28200
## factor(pollib)2    -1.2340106  0.6969352  -1.771  0.07662 .
## parties             0.0268510  0.0112127   2.395  0.01663 *
## pctvote             0.0118409  0.0104005   1.138  0.25492
## popn                0.0094725  0.0055567   1.705  0.08825 .
## size               -0.0002410  0.0002644  -0.912  0.36192
## numelec            -0.0094734  0.0665066  -0.142  0.88673
## ---
## 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: 28.536  on 27  degrees of freedom
## AIC: 111.35
##
## Number of Fisher Scoring iterations: 5
```

Analyzing the Initial Model

The formula shows that all the predictors have been used to build the model.

formula = miltcoup ~ oligarchy + factor(pollib) + parties + pctvote + popn + size + numelec

The Residual Deviance or the Deviance of this model is 28.538 on 27 degrees of freedom compared to the Null Deviance of 65.945 on 35 degrees of freedom. The deviance is less but we have to see if we can find a better model than this.

From the $p - values$ of the predictors only *oligarchy*, *pollib* and *parties* look like they are important predictors since they have their p-values less than the significance level α of 0.05.

Feature Selection

We will do a feature selection using backward step-wise method with AIC as our metric of evaluation of the models.We will select a model having lowest AIC value.

```
# Variable selection using aic backward
model.aic = step(poisson.mdl, direction = "backward", trace = 1)
```

We have got the model with reduced number of features after we applied step-wise feature selection method based on AIC metric for model selection. Let us see the summary of our best model "model.aic"

```
# best model summary
summary(model.aic)
```

```
##
## Call:
## glm(formula = miltcoup ~ log(oligarchy + 1) + factor(pollib) +
##     parties, family = poisson, data = africa2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3522  -0.9279  -0.5327   0.6509   1.8902
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.297423   0.500284  -0.595   0.5522
## log(oligarchy + 1)   0.566318   0.144549   3.918 8.93e-05 ***
## factor(pollib)1     -0.201338   0.463686  -0.434   0.6641
## factor(pollib)2     -0.800866   0.457787  -1.749   0.0802 .
## parties              0.019651   0.009349   2.102   0.0356 *
## ---
## 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: 32.132  on 31  degrees of freedom
## AIC: 106.94
##
## Number of Fisher Scoring iterations: 5
```

Interpreting our Best Model

We have a reduced AIC score of 107.63.

Our reduced model has just three predictors and now we will see the interpretaion of the coefficients in our model:

- **log(Oligarchy+1)-** for every unit(year) increase in oligarchy there is an increase in Probability of military coup by a factor of $e^{0.0915}$
- **pollib-** We see that for pollib=0 (No civil rights for political expression) the Probability of military coup increases by a factor of $e^{0.207}$ and decreases in the cases of pollib=1 (limited civil rights for expression but right to form political parties) and further decreases in pollib=2 (full civil rights). **This makes complete sense and strengthens our belief in a society with full civil rights.**

- **parties-** With a unit increase in this predictor we see a small change of $e^{0.0224}$ in the probability of a military coup.

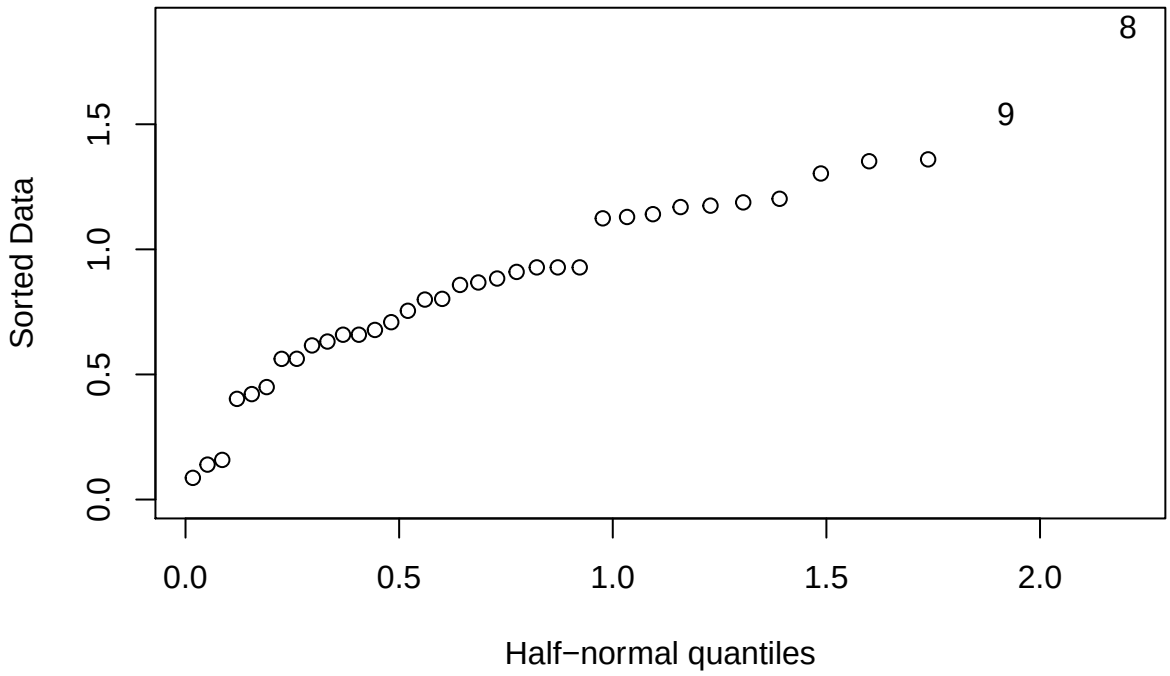
Part b

Here we will discuss model fit, and also look into influence and outliers you might identify

Influence Points and Outliers

Let us try to identify any points of influence and outliers if we can find any.

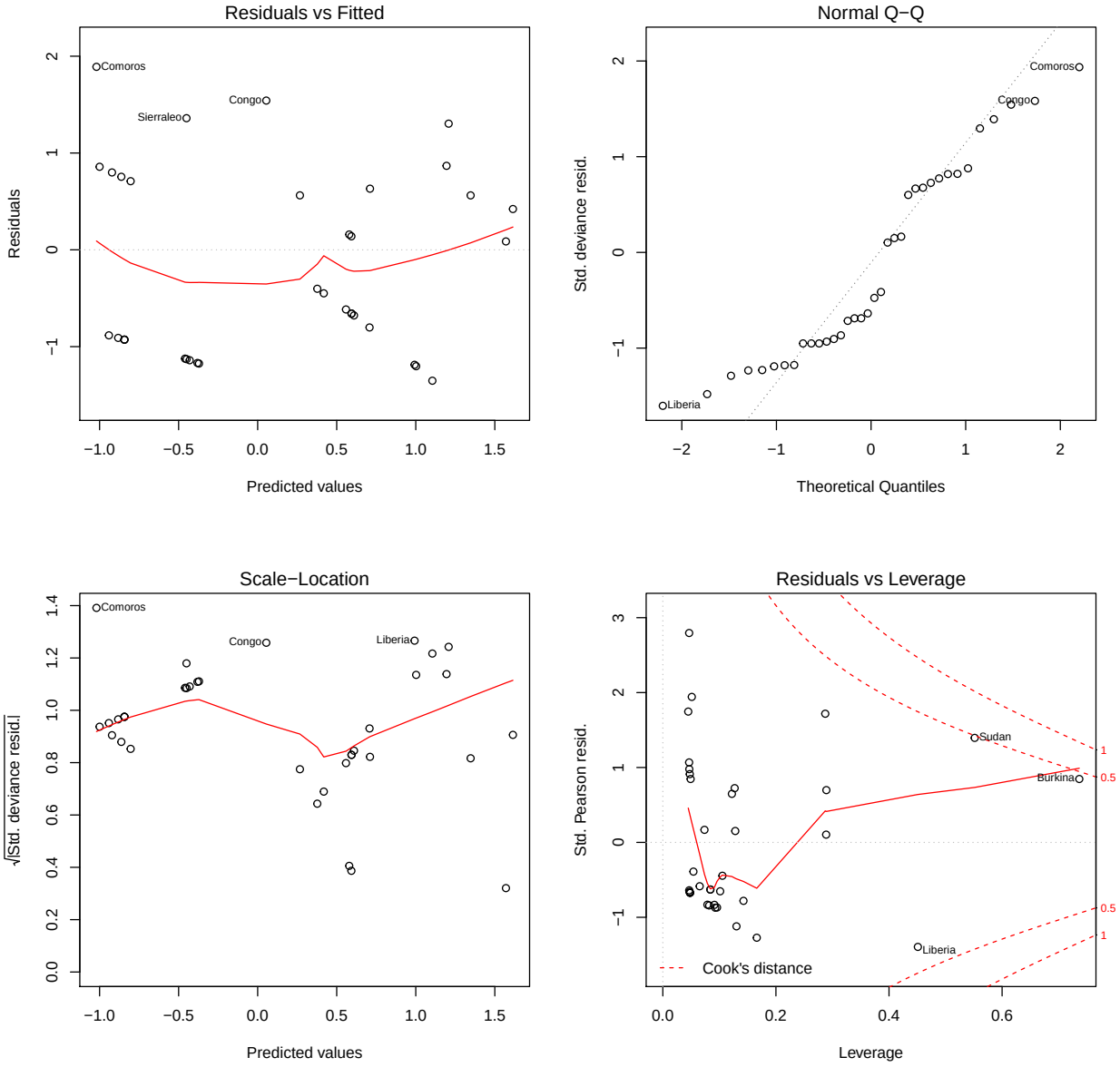
The Half normal plot should show us any outliers



The half normal plot looks fine

- We do not find any outlier
- The points lie almost on the straight line

Let us now look at all the plots of the model. The Residual vs Fitted values, The cooks's distance plot.



In the Residuals vs Fitted value we are mainly concerned with two issues:

- Is there any trend in the plot, in this case we do not notice any trend (linear or non-linear) so we are good.
- Is the variance of the residuals constant. In our case the variance of the residuals look quite constant.

From the Q-Q plot we identify Lineria as a point of concern that is it lies out of the normal line, so we will look at it in the Residual vs Leverage plot

In the residual vs leverage plot we notice that Liberia is within the drawn limits of Cook's distance so we cannot mark it as an outlier.

Model Fit Tests

We know that the deviance of the correct model is approximately chi-square distribution and the deviance for an incorrect model tend to be larger.

Our H_0 null hypothesis is that our model is correct, We will now find the p-value of the chi-square test.

```
# Chi-square test of the model deviance
pchisq(deviance(model.aic), df.residual(model.aic), lower=FALSE)
```

```
## [1] 0.4103609
```

Our p-value is greater than the significance level so we do not have evidence of lack-of fit. Our model fits the data well but at the same time we cannot say just from this that our model is the best model.

We can do a similar lack-of-fit test based on *Pearson chi-square* statistic

```
X.2 <- sum(residuals(model.aic,type="pearson")^2)
pchisq(X.2, df.residual(model.aic), lower=FALSE)
```

```
## [1] 0.3707424
```

Again our p-value is greater than the significance level of 0.05 thus we do not find any evidence of a lack-of fit and we can say that our data fits well.

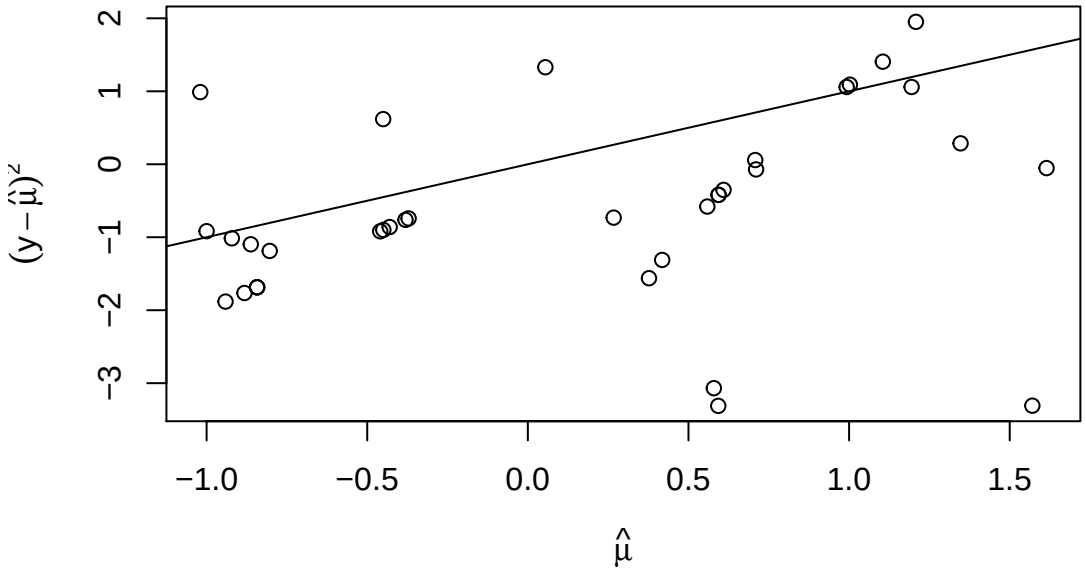
This corroborates our analysis of outliers and influence points wherein we did not find any such points of interest.

To find a better model we can look at a possible over-dispersion

Part c

Investigation of Over Dispersion

For a Poisson distribution, the mean is equal to the variance. Let's investigate this relationship for this model. It is difficult to estimate the variance for a given value of the mean, but $(y-\hat{\mu})^2$ does serve as a crude approximation. We plot this estimated variance against the mean



We see that the variance is proportional to, but larger than, the mean. When the variance assumption of the Poisson regression model is broken but the link function and choice of predictors is correct, the estimates of β are consistent, but the standard errors will be wrong.

Thus, we find the dispersion parameter.

```
## Dispersion parameter (phi-hat) =  1.063668
```

Our dispersion parameter $\hat{\phi} = 1.063668$ is greater than one but very slightly. Still we will go ahead and adjust the standard errors in the summary

```
summary(model.aic, dispersion = phihat)
```

```
##
## Call:
## glm(formula = miltcoup ~ log(oligarchy + 1) + factor(pollib) +
##     parties, family = poisson, data = africa2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3522  -0.9279  -0.5327   0.6509   1.8902
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)        -0.297423   0.515965  -0.576 0.564317
## log(oligarchy + 1)  0.566318   0.149080   3.799 0.000145 ***
## factor(pollib)1    -0.201338   0.478220  -0.421 0.673744
## factor(pollib)2    -0.800866   0.472135  -1.696 0.089836 .
## parties             0.019651   0.009642   2.038 0.041552 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1.063668)
##
##     Null deviance: 65.945  on 35  degrees of freedom
## Residual deviance: 32.132  on 31  degrees of freedom
## AIC: 106.94
##
## Number of Fisher Scoring iterations: 5
```

As expected we do not see any changes in the coefficients but there are changes in the standard errors of the predictors.

When comparing Poisson models with overdispersion, an F-test rather than a χ^2 test should be used. As in normal linear models, the variance, or dispersion parameter in this case, needs to be estimated. This requires the use of the F-test. So to test the significance of each of the predictors relative to the full model, use:

```
drop1(model.aic, test="F")
```

```
## Single term deletions
##
## Model:
## miltcoup ~ log(oligarchy + 1) + factor(pollib) + parties
##                     Df Deviance    AIC F value    Pr(>F)
## <none>                   32.132 106.94
## log(oligarchy + 1)  1   49.458 122.27 16.7152 0.0002855 ***
## factor(pollib)      2   37.431 108.24  2.5558 0.0938804 .
## parties             1   36.086 108.89  3.8148 0.0598797 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 2

In this question we will be working with the **aflatoxin** dataset. So let us understand the dataset.

Description

Aflatoxin B1 was fed to lab animals at vary doses and the number responding with liver cancer recorded.

Data Fields

dose = dose in ppb (datatype : int)

total = number of test animals (datatype : int)

tumor = number with liver cancer (datatype : int)

```
## 'data.frame':    6 obs. of  3 variables:
##  $ dose : int  0 1 5 15 50 100
```

```
##  $ total: int  18 22 22 21 25 28
##  $ tumor: int  0 2 1 4 20 28
```

Are there any missing values?

In this dataset there are no missing values or incomplete cases.

Part a

Use a variety of link functions in a glm for binary regression to model the probability of developing liver cancer depending on the dose of aflatoxin.

We will fit a **logit** link function first and let us see it's summary

```
# Fit logit model
logitmod = glm(formula = cbind(tumor, total - tumor) ~ dose,
               family = binomial,
               data = aflatoxin)
```

```
summary(logitmod)
```

```
##
## Call:
## glm(formula = cbind(tumor, total - tumor) ~ dose, family = binomial,
##     data = aflatoxin)
##
## Deviance Residuals:
##       1        2        3        4        5        6
## -1.2995   0.7959  -0.4814   0.4174  -0.1629   0.3774
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.03604    0.48226  -6.295 3.07e-10 ***
## dose         0.09009    0.01456   6.189 6.04e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 116.524  on 5  degrees of freedom
## Residual deviance:   2.897  on 4  degrees of freedom
## AIC: 17.685
##
## Number of Fisher Scoring iterations: 5
```

From the p-value we can infer that the predictor dose is very significant and the Residual deviance is 2.897 as compared to the Null deviance of 116.524 which is again an indicator of a good fit.

Let us see the **probit** model now.

```
# Fit probit model

probitmod = glm(formula = cbind(tumor, total - tumor) ~ dose,
                family = binomial(link = probit),
                data = aflatoxin)
```

```r
summary(probitmod)
```

```
##
## Call:
## glm(formula = cbind(tumor, total - tumor) ~ dose, family = binomial(link = probit),
##     data = aflatoxin)
##
## Deviance Residuals:
##       1         2         3         4         5         6
## -1.23164   0.89070  -0.47838   0.25563  -0.06575   0.12366
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.736231   0.240449  -7.221 5.17e-13 ***
## dose         0.051933   0.007873   6.596 4.22e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 116.5243  on 5  degrees of freedom
## Residual deviance:   2.6241  on 4  degrees of freedom
## AIC: 17.413
##
## Number of Fisher Scoring iterations: 5
```

From the p-value we can infer that the predictor dose is very significant, infact more significant compared to the "logitmod" model and the Residual deviance is 2.624 as compared to the Null deviance of 116.524 which is again an indicator of a good fit.

We will fit one more link function **complimentary log-log** function.

```r
# Fit complimentary log model

clogmod = glm(formula = cbind(tumor, total - tumor) ~ dose,
              family = binomial(link = cloglog),
              data = aflatoxin)

summary(clogmod)
```

```
##
## Call:
## glm(formula = cbind(tumor, total - tumor) ~ dose, family = binomial(link = cloglog),
##     data = aflatoxin)
##
## Deviance Residuals:
##       1        2        3        4        5        6
## -1.3993   0.6439  -0.5416   0.6345  -0.0724   0.0000
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.91152    0.45356  -6.419 1.37e-10 ***
## dose         0.06811    0.01071   6.361 2.00e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

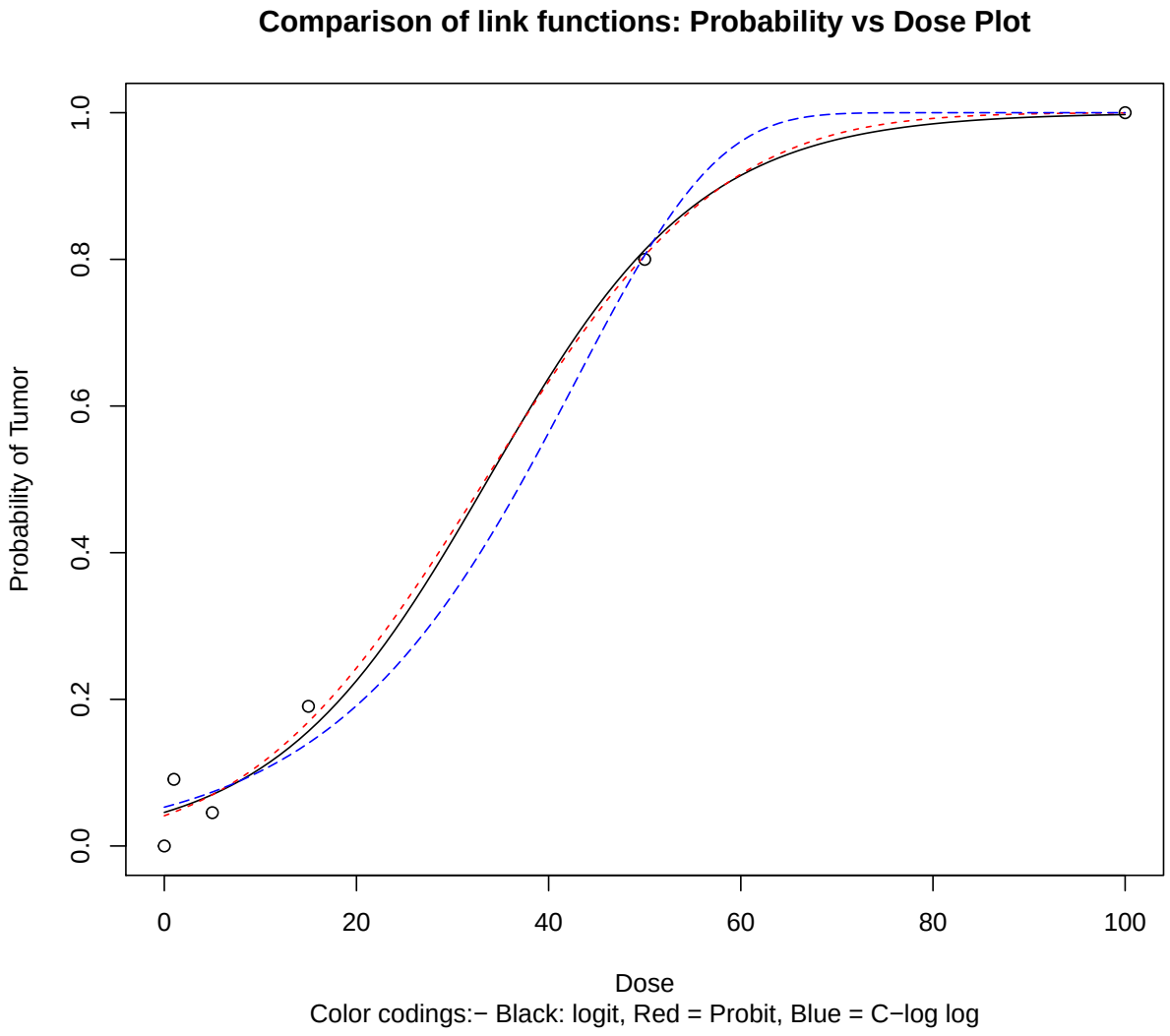
```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 116.5243  on 5  degrees of freedom
## Residual deviance:   3.0739  on 4  degrees of freedom
## AIC: 17.862
##
## Number of Fisher Scoring iterations: 6
```

The residual deviance has increased compared to the other two models but still it is significant.

Part b

In this part we will be comparing for each link function the observed versus expected proportions, and suggest which link function works best.

Before looking at the comparison of observed vs expected proportions in the tabular form we will look at the comparison plot of all the three link functions.



From the plot we can observe that the blue curve (complimentary log link function) does not fit the data as much as the black(logit) or the red(probit) lines fit. This was expected since we had got a higher deviance value for the complimentary log model as compared to the probit or logit model.

Now let us see the comparison of the observed vs expected proportion in a tabular format.

Actual	Logit Model	Probit Model	Complimentary Log
0.0000000	0.0458242	0.0412615	0.0529401
0.0909091	0.0499284	0.0460621	0.0565638
0.0454545	0.0700714	0.0698962	0.0736103
0.1904762	0.1564735	0.1692252	0.1402256
0.8000000	0.8128144	0.8052251	0.8057495
1.0000000	0.9974595	0.9997270	1.0000000

From the comparison table above it looks like **Probit model** is the best link function. On average the values of the probit model are the closest to the actual values compared to the other two models.

To further clarify this point we did a chisquare test on the deviance of the three models and the $p - value$ for the probit link function model was the highest showing the best fit among the three models.

We conclude that the probit link function is the best link function for this data

Part c

In this part we willcompute the dose at which we would expect 50 percent to develop liver cancer for the case of logistic regression.

For logistic regression the equation is as follows:

$$log(\frac{p(x)}{1-p(x)}) = \beta_0 + \beta_1 X$$

Now if we set $p(x) = 0.5$ which is what we require. The left hand side of the above equation would be 0.

Thus we can find the dose at which there is a 50% chance of developing cancer would be:

$$X = -\beta_0/\beta_1$$

At 33.7 ppb dose there is a 50% probability of developing cancer.

Part d

Using your best model (link function), predict the proportion developing liver cancer at a dose of 25, and provide a 95 percent confidence interval for it.

We get the following result:

```
##         1         1         1
## 0.2213407 0.3307299 0.4569486
```

The predicted probability of having cancer is 33.07%

The upper bound of the 95% prediction interval is 45.69% and the lower bound is 22.13%

Appendix

```
knitr::opts_chunk$set(echo = FALSE, warning=FALSE)
# Loading required Libraries
library(faraway)
library(ggplot2)
library(cowplot)
# looking at Africa data
str(africa)
#find number of incomplete cases
sum(!complete.cases(africa))
# Work with only the complete cases
complete.ind = complete.cases(africa)
africa2 = africa[complete.ind, ]
#Plot of miltcoup vs oligarchy and miltcoup vs log(oligarchy)

p1 = ggplot(data = africa2, aes(x = oligarchy, y = miltcoup)) +
  geom_point(pch = 20, size = 3, col = "blue") +
  theme_bw() +
  theme(panel.border = element_blank(), axis.line = element_line(colour = "black")) +
  ggtitle("miltcoup vs oligarchy")


p2 = ggplot(data = africa2, aes(x = log(oligarchy+1), y = miltcoup)) +
  geom_point(pch = 20, size = 3, col = "blue") +
  theme_bw() +
  theme(panel.border = element_blank(), axis.line = element_line(colour = "black")) +
  ggtitle("miltcoup vs log(oligarchy)")

plot_grid(p1, p2, ncol = 2, nrow = 1)
## # Fit poisson model
## poisson.mdl = glm(miltcoup ~ log(oligarchy+1) + factor( pollib ) + parties +
##                     pctvote + popn + size + numelec,
##                   data = africa2,
##                   family = poisson)
poisson.mdl = glm(miltcoup ~ log(oligarchy+1) + factor( pollib ) + parties +
                    pctvote + popn + size + numelec,
                  data = africa2,
                  family = poisson)
summary(poisson.mdl)
## # Variable selection using aic backward
## model.aic = step(poisson.mdl, direction = "backward", trace = 1)
model.aic = step(poisson.mdl, direction = "backward", trace = 1)
# best model summary
summary(model.aic)
# PART B


halfnorm(residuals(model.aic))
#diagnostic plots of the model
par(mfrow=c(2,2))
plot(model.aic)
```

```r
# Chi-square test of the model deviance
pchisq(deviance(model.aic), df.residual(model.aic), lower=FALSE)
X.2 <- sum(residuals(model.aic,type="pearson")^2)
pchisq(X.2, df.residual(model.aic), lower=FALSE)

#Investigation of Over Dispersion

plot(log(fitted(model.aic)),log((africa2$miltcoup - fitted(model.aic))^2),
xlab=expression(hat(mu)),ylab=expression((y-hat(mu))^2))

abline(0,1)
# Dispersion parameter
phihat <- X.2 / df.residual(model.aic)
cat("Dispersion parameter (phi-hat) = ", phihat)
summary(model.aic, dispersion = phihat)
drop1(model.aic, test="F")


# Question 2

# lookin at the aflatoxin data
str(aflatoxin)

#Find incomplete cases

sum(!complete.cases(aflatoxin))
# Fit logit model
logitmod = glm(formula = cbind(tumor, total - tumor) ~ dose,
               family = binomial,
               data = aflatoxin)
summary(logitmod)
# Fit probit model

probitmod = glm(formula = cbind(tumor, total - tumor) ~ dose,
                family = binomial(link = probit),
                data = aflatoxin)

summary(probitmod)

# Fit complimentary log model

clogmod = glm(formula = cbind(tumor, total - tumor) ~ dose,
              family = binomial(link = cloglog),
              data = aflatoxin)

summary(clogmod)

#comparative study of link functions

x <- seq(0,100,1)
pl <- ilogit(logitmod$coef[1]+logitmod$coef[2]*x)
pp <- pnorm(probitmod$coef[1]+probitmod$coef[2]*x)
pc <- 1-exp(-exp((clogmod$coef[1]+clogmod$coef[2]*x)))
```

```r
plot(aflatoxin$dose, aflatoxin$tumor/aflatoxin$total, ylab="Probability of Tumor", xlab="Dose",
     sub="Color codings:- Black: logit, Red = Probit, Blue = C-log log",
     main="Comparison of link functions: Probability vs Dose Plot")
lines(x, pl)
lines(x, pp, lty=2, col="red")
lines(x, pc, lty=5, col="blue")
df = cbind(aflatoxin$tumor/aflatoxin$total, logitmod$fitted.values,
           probitmod$fitted.values, clogmod$fitted.values)
colnames(df) = c("Actual", "Logit Model", "Probit Model", "Complimentary Log")
knitr::kable(df, align = "c")
pchisq(deviance(logitmod), df.residual(logitmod), lower=FALSE)
pchisq(deviance(probitmod), df.residual(probitmod), lower=FALSE)
pchisq(deviance(clogmod), df.residual(clogmod), lower=FALSE)

# Part C

x50 = -logitmod$coefficients[1]/logitmod$coefficients[2]
x50

# Part d

preds = predict(probitmod, data.frame(dose = 25), type="link", se.fit=TRUE)
critval <- 1.96 ## 95% CI
upr <- pnorm(preds$fit + (critval * preds$se.fit))
lwr <- pnorm(preds$fit - (critval * preds$se.fit))
fit <- pnorm(preds$fit)
print(c(lwr, fit, upr))
##
```