Advanced Predictive Analytics Assignment 1

16343261

13/03/2023

Table of Contents

Before we begin this assignment, I will load in the data and packages as required.

```
library(ISLR2)
library(glmnet)
library(cowplot)
library(ggplot2)
data(Boston)
Boston <- na.omit(Boston)
y <- Boston$medv
x <- model.matrix(medv ~., Boston)[, -1]
```

Question 1A.

To begin this question, we will randomly split the data set into a training and test set.

```
set.seed(116)
train <- sample(1:nrow(x), nrow(x) / 2)
test <- (-train)
y.test <- y[test]
```

Next we will fit our 3 linear models as requested starting with least squares.

```
# Least Squares
ols.mod <- lm(y[train] x̃[train, ])
ols.pred<-cbind(1,x[test, ])%*%ols.mod$coef
```

Next we will calculate our cross validation chosen lambda for Ridge and LASSO regression and fit them accordingly.

```
lambda_ridge<-cv.glmnet(x[train, ],y[train],alpha=0)
lambda_min<-lambda_ridge$lambda.min
lambda_min

[1] 0.657228

lambda_star <- lambda_min

ridge.mod <- glmnet(x[train, ], y[train], alpha = 0, lambda = lambda_star)
ridge.pred <- predict(ridge.mod, s = lambda_star, newx = x[test, ])

lambda_LASSO<-cv.glmnet(x[train, ],y[train],alpha=1)
lambda_min<-lambda_LASSO$lambda.min
lambda_min

[1] 0.01871809

LASSO.mod <- glmnet(x[train, ], y[train], alpha = 1, lambda = lambda_min)
LASSO.pred <- predict(LASSO.mod, s = lambda_min, newx = x[test, ])
```

Now let's report the test error obtained according to these three methods.

```
mean((ols.pred - y.test)^2)

[1] 23.56593

mean((ridge.pred - y.test)^2)

[1] 24.86081

mean((LASSO.pred - y.test)^2)

[1] 23.65745
```

Here we see our Least Squares model has the lowest test error.

Question 1B.

Now let's repeat this procedure 100 times. We will achieve this using a for loop and running our error values into an empty vector.

```
# Empty error vector.
OLS.er <- c()
RIDGE.er <- c()
LASSO.er <- c()
```

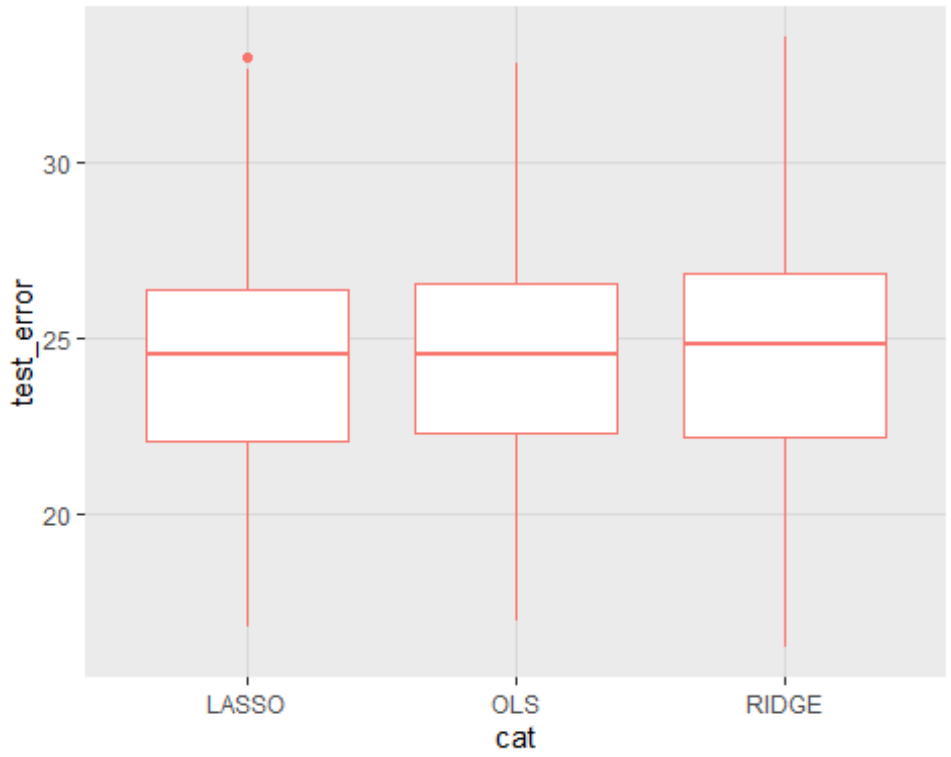
```
# For loop.
for(i in 1:100){
# Split data randomly.
train <- sample(1:nrow(x), nrow(x) / 2)
test <- (-train)
y.test <- y[test]
# Least squares
ols.mod <- lm(y[train] x̃[train, ])
ols.pred<-cbind(1,x[test, ])%*%ols.mod$coef
OLS.er[i] <- mean((ols.pred - y.test)^2)
# Ridge
ridge.mod <- glmnet(x[train, ], y[train], alpha = 0, lambda = lambda_star)
ridge.pred <- predict(ridge.mod, s = lambda_star, newx = x[test, ])
RIDGE.er[i] <- mean((ridge.pred - y.test)^2)
# LASSO
LASSO.mod <- glmnet(x[train, ], y[train], alpha = 1, lambda = lambda_min)
LASSO.pred <- predict(LASSO.mod, s = lambda_min, newx = x[test, ])
LASSO.er[i] <- mean((LASSO.pred - y.test)^2)
}
```

To make our boxplots as requested, we will create a new dataframe with our categories separated.

```
cat <- c()
cat[1:100] <- "OLS"
cat[101:200] <- "RIDGE"
cat[201:300] <- "LASSO"

test_error <- c()
test_error[1:100] <- OLS.er
test_error[101:200] <- RIDGE.er
test_error[201:300] <- LASSO.er

plot <- data.frame(cat,test_error)
plot$cat <- as.factor(plot$cat)
```

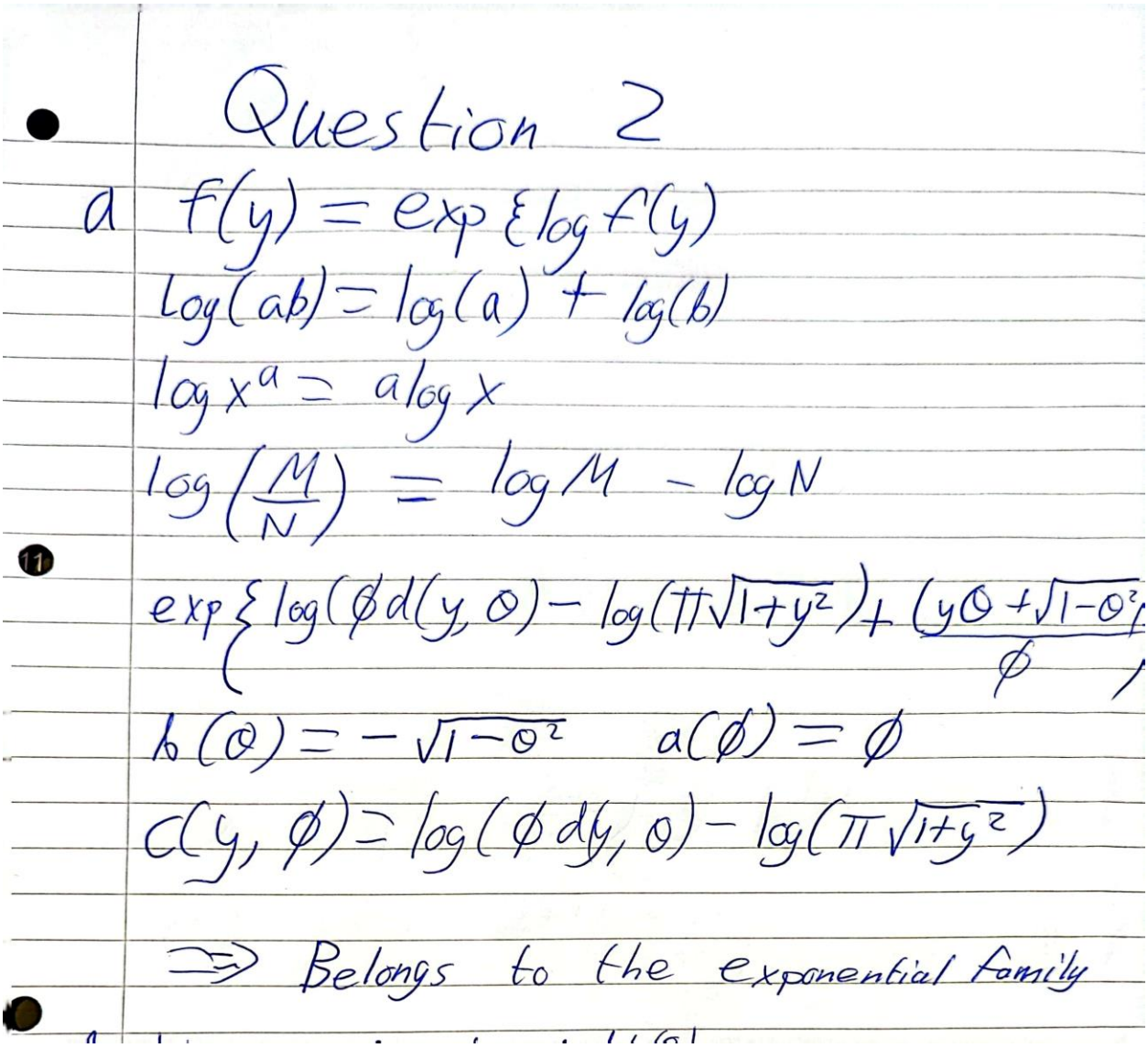
Now lets create our boxplot.

```
# Boxplot
bp <- ggplot(plot, aes(x=cat, y=test_error, color='red')) +
  geom_boxplot() +
  theme(legend.position = "none")
# Add gridlines
bp + background_grid(major = "xy", minor = "none")
```



While all 3 of our models have a very similar test error. It appears from our Boxplot that least squares is performing the best prediction by reducing error slightly more.

Question 2a.



Question 2

a $f(y) = \exp\{\log f(y)\}$

$\log(ab) = \log(a) + \log(b)$

$\log x^a = a \log x$

$\log\left(\frac{M}{N}\right) = \log M - \log N$

$\exp\left\{\log(\phi d(y,\theta) - \log(\pi\sqrt{1+y^2}) + \frac{(y\theta + \sqrt{1-\theta^2}}{\phi}\right.$

$b(\theta) = -\sqrt{1-\theta^2} \quad a(\phi) = \phi$

$c(y,\phi) = \log(\phi d(y,\theta) - \log(\pi\sqrt{1+y^2})$

$\Rightarrow$ Belongs to the exponential family

Question 2b.

Question 2B

Take $\theta = x$

$a(\phi)\, b''(\theta)$ = variance function

$$b'(\theta) = \frac{d}{dx}\left(-\sqrt{(1-x^2)}\right) = -\frac{d}{dx}\left(\sqrt{(1-x^2)}\right)$$

Chain Rule

$$\frac{db}{dx} = \frac{db}{du} \times \frac{du}{dx} \qquad u = 1-x^2$$

$u^{1/2}$

$$\Rightarrow \frac{db}{du} = \frac{1}{2} \cdot \frac{1}{\sqrt{u}} = \frac{1}{2\sqrt{u}} \Rightarrow \frac{db}{dx} = \frac{+x}{\sqrt{1-x^2}}$$

$$\frac{du}{dx} = -2x$$

$$\frac{d^2b}{dx^2} = \frac{x}{\sqrt{1-x^2}} \frac{d}{dx}$$ Quotient Rule

$f(x) = x$, $g(x) = \sqrt{1-x^2}$

$f'(x) = 1$ $g'(x)$ already calculated: $\frac{-x}{\sqrt{1-x^2}}$

$$\Rightarrow \frac{\sqrt{1-x^2} - x\left(\frac{-x}{\sqrt{1-x^2}}\right)}{1-x^2} = \frac{1}{(1-x^2)\sqrt{1-x^2}}$$

$$\Rightarrow \text{Variance function} = \frac{1}{(1-\theta^2)\sqrt{1-\theta^2}}(\phi)$$

Question 2c.

Question 2c

$$E(Y) = \frac{\theta}{\sqrt{1-\theta^2}} = \mu$$

$$\theta^2 = \mu^2(1-\theta^2)$$

$$= \mu^2(1-\theta^2)$$

$$\theta^2 = \mu^2 - \mu^2\theta^2$$

$$\mu^2 = (1+\mu^2)(\theta^2)$$

$$\theta = \sqrt{\frac{\mu^2}{1+\mu^2}} = \frac{\mu}{\sqrt{1+\mu^2}}$$

Deviance

$$D = 2\sum_{i=1}^{n}\{y_i(\tilde{\theta}_i - \hat{\theta}_i) - b(\tilde{\theta}_i) + b(\hat{\theta}_i)\}$$

$$\tilde{\theta}_i = \frac{y_i}{\sqrt{1+y_i^2}} \qquad \hat{\theta}_i = \frac{\hat{\mu}_i}{\sqrt{1+\hat{\mu}_i^2}}$$

$$\Rightarrow 2\sum_{i=1}^{n} y_i\left(\frac{y_i}{\sqrt{1+y_i^2}} - \frac{\hat{\mu}_i}{\sqrt{1+\hat{\mu}_i^2}}\right) - \left(-\sqrt{1-\left(\frac{y_i}{\sqrt{1+y_i^2}}\right)^2}\right) + \left(\sqrt{1-\left(\frac{\hat{\mu}_i}{\sqrt{1+\mu_i^2}}\right)^2}\right)$$

$$= 2\sum_{i=1}^{n}\left\{\frac{y_i^2}{\sqrt{1+y_i^2}} - \frac{y_i\hat{\mu}_i}{\sqrt{1+\hat{\mu}_i^2}} + \sqrt{1-\frac{y_i^2}{1+y_i^2}} - \sqrt{1-\frac{\hat{\mu}_i^2}{1+\mu_i^2}}\right\}$$

Question 3a.

Question 3A

Gamma

$$f(y) = \frac{1}{\Gamma(\alpha)}\left(\frac{\alpha}{\mu}\right)^{\alpha} y^{\alpha-1} e^{-\alpha y/\mu}, \quad y>0$$

$$= \exp\{-\log\Gamma(\alpha) + \alpha\log\alpha - \alpha\log\mu + (\alpha-1)\log y - \alpha y/\mu\}$$

$$= \exp\{\alpha[y(-1/\mu) - \log\mu] + \alpha\log\alpha - \log\Gamma(\alpha) + (\alpha-1)\log y\}$$

$\Rightarrow$ Gamma distribution belongs to the EF with $\theta = -1/\mu$, $\phi = \alpha$,

$$b(\theta) = \log\mu = \log\left(\frac{-1}{\theta}\right) = -\log(-\theta)$$

$a(\phi) = 1/\phi$, and

$$c(y;\phi) = \phi\log\phi - \log\Gamma(\phi) + (\phi-1)\log y$$

$\Rightarrow$ We can use Maximum likelihood estimation

$\Rightarrow$ Associated likelihood equation:

$$\frac{-a'(\phi)}{(a(\phi))^2}\sum_{i=1}^{n}(y_i\hat{\theta}_i - b(\hat{\theta}_i)) + \sum_{i=1}^{n} c'(y_i,\phi) = 0$$

$$(\phi^{-1})\frac{d}{d\phi} = \frac{-1}{\phi^2} = a'(\phi)$$

$$\frac{d}{d\phi} c(y;\phi) = [illegible]$$

$$\frac{d}{d\phi}\left(\phi\log\phi - \log\Gamma(\phi) + (\phi-1)\log y\right)$$

$$= \log(\phi)+1 [illegible] + \ln(y) - \psi(\phi) = c(y_i;\phi)$$

Subing into our equation

$$\frac{-\phi^{-2}}{\phi^{-2}} \sum_{i=1}^{n}\left(y_i\hat{\theta}_i - (-\log(-\hat{\theta}_i))\right) + \sum_{i=1}^{n} c'(y_i,\phi) = 0$$

$$= -1\sum_{i=1}^{n}\left(y_i\hat{\theta}_i + \log(-\hat{\theta}_i)\right) + \sum_{i=1}^{n}\left(\log(\phi)+1+\ln(y_i) - \psi(\phi)\right)$$

$$= 0$$

Question 3b.

Question 3b

Log-Likelihood function

$$l = \sum_{i=1}^{n} \frac{y_i \theta_i - b(\theta_i)}{a(\phi)} + \sum_{i=1}^{n} c(y_i; \phi)$$

$$= \sum_{i=1}^{n} \frac{y_i \theta_i + \log(-\theta)}{\left(\frac{1}{\phi}\right)} + \sum_{i=1}^{n} \left(\phi \log \phi - \log \Gamma(\phi) + (\phi - 1) \log y_i\right)$$

$$= \sum_{i=1}^{n} \left(\phi y_i \theta_i + \phi \log(-\theta)\right) + \sum_{i=1}^{n} \left(\phi \log \phi - \log \Gamma(\phi) + (\phi - 1) \log y_i\right)$$

Remember $\theta = -1/\mu_i$, $\hat{\mu}_i$ MLE

Likelihood-ratio test statistic

$$LR = 2(l_1 - l_0)$$

$$l_0 = \sum_{i=1}^{n} \frac{-y_i}{\mu_i} + \log\left(\frac{1}{\mu_i}\right) + \sum_{i=1}^{n} (0 - 0 + 0)$$

$$= \sum_{i=1}^{n} \frac{-y_i}{\mu_i} + \log\left(\frac{1}{\mu_i}\right)$$

$$l_{\text{over}}1 = \sum_{i=1}^{n} \left(\hat{\phi} y_i / \hat{\mu}_i + \hat{\phi} \log\left(\frac{1}{\hat{\mu}_i}\right) + \sum_{i=1}^{n} \left(\hat{\phi} \log \hat{\phi} - \log \Gamma(\hat{\phi}) + (\hat{\phi} - 1) \log y_i\right)\right.$$

$$\Rightarrow LR =$$

$$2\left(\sum_{i=1}^{n} \left(-\hat{\phi} + 1\right)\left(\frac{y_i}{\hat{\mu}_i}\right) + (\hat{\phi} - 1) \log\left(\frac{1}{\hat{\mu}_i}\right) + \sum_{i=1}^{n} \left(\hat{\phi} \log \hat{\phi} - \log \Gamma(\hat{\phi}) + (\hat{\phi} - 1) \log y_i\right)\right)$$

Question 4a.

To begin to answer this question, we will first create our Table 1 dataset. We will call this AG.

```
WBC <-
c(2300,750,4300,2600,6000,10500,10000,17000,5400,7000,9400,32000,35000,100000
,100000,52000,100000,4000,3000,4000,1500,9000,5300,10000,19000,27000,28000,31
000,26000,21000,79000, 100000,100000)
time <-
c(65,156,100,134,16,108,121,4,39,143,56,26,22,1,1,5,65,56,65,17,7,16,22,3,4,2
,3,8,4,3,30,4,43)
result <-
c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
AG <- data.frame(WBC,time,result)
```

Next we will define our Gamma glm and define our log(WBC) variables.

```
AG$Log_WBC <- log(WBC)
fit<-glm(time ~ result+Log_WBC, family=Gamma, data = AG)
summary(fit)


Call:
glm(formula = time ~ result + Log_WBC, family = Gamma, data = AG)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1982  -1.1746  -0.4314   0.4795   1.3680

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.002075   0.025417  -0.082   0.9355
result      -0.034432   0.014579  -2.362   0.0249 *
Log_WBC      0.006122   0.002309   2.652   0.0127 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Gamma family taken to be 0.9858659)

    Null deviance: 58.138  on 32  degrees of freedom
Residual deviance: 40.002  on 30  degrees of freedom
AIC: 301.18

Number of Fisher Scoring iterations: 6
```

Here we find our line of best fit: Time = (-0.034)xresult + (0.006)xLog_wbc.

Now let's construct our 95% confidence intervals for the parameters.

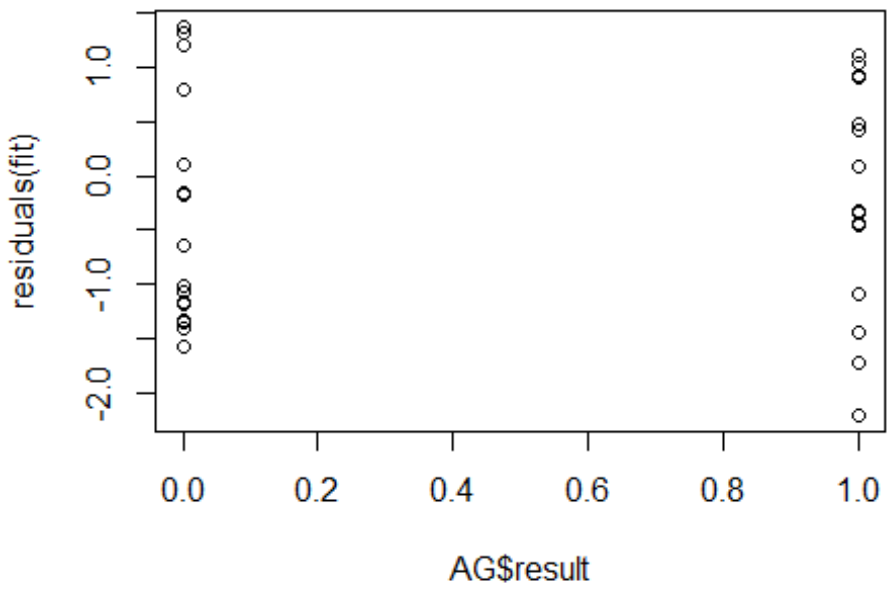
```
SE<-coef(summary(fit))[,2]
# standard errors
inf<-fit$coef-qnorm(1-0.05/2)*SE
# inferior bound
sup<-fit$coef+qnorm(1-0.05/2)*SE
# Superior bound
inf

(Intercept)      result     Log_WBC
-0.05189137 -0.06300612  0.00159719

sup

 (Intercept)       result      Log_WBC
 0.047741428 -0.005857187  0.010647238
```

Here we find our result and log_wbc variables are both significant as zero does not fall in our 95% confidence interval. However here we find our intercept is not significant as zero falls in our interval.

Next we will check the adequacy of our model using residuals. First let's check our assumption of Homoscedasticity.

To check this assumption we must plot the residuals of an explanatory variable and check for constant variance.

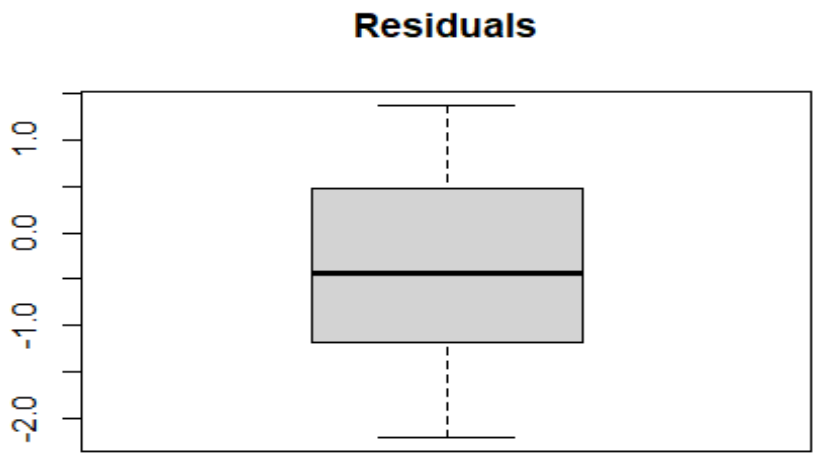
```
plot(AG$result, residuals(fit))
```



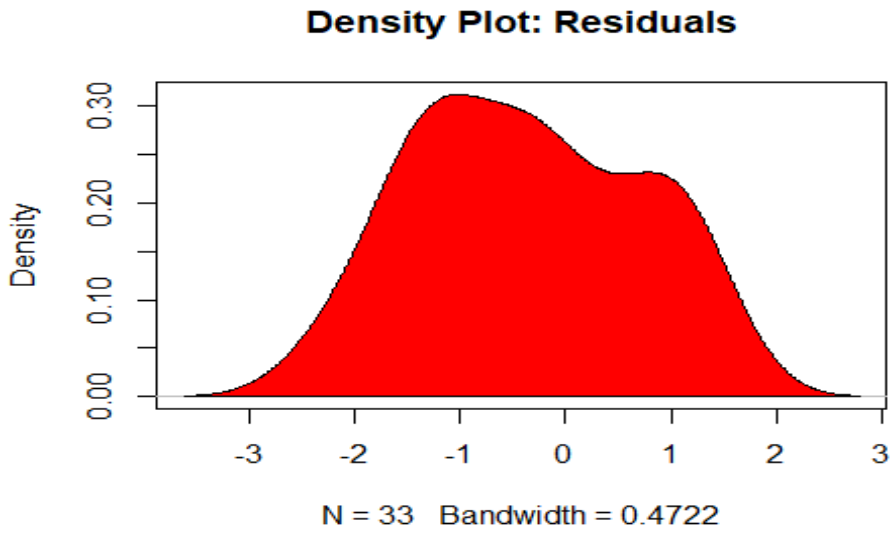
Our variance looks to be constant as the variability of our results observations is similar across our different values.

Next we can use residuals to check our normality assumption. To check this assumption, we will create a boxplot of the residuals and the density of the residuals to see if there is any outliers present in our data or any unusual distribution.

```
boxplot(residuals(fit), main="Residuals")
```



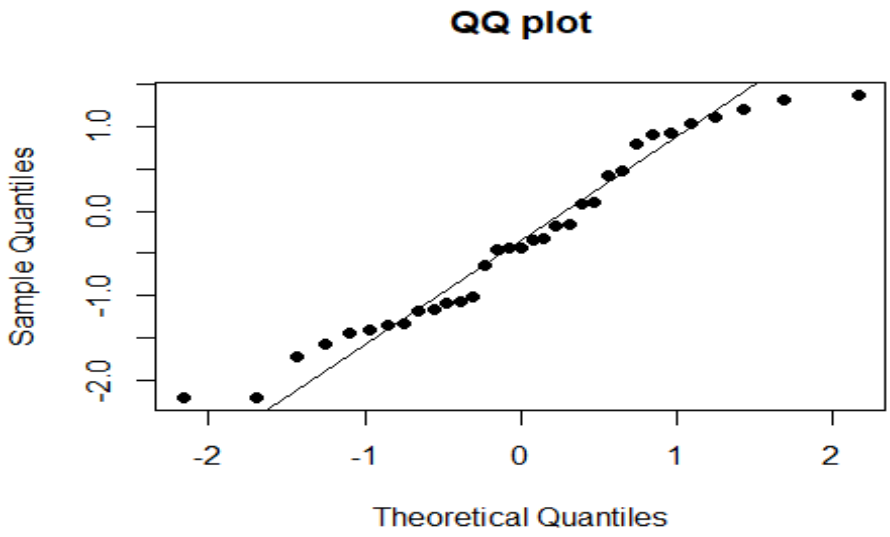

```
plot(density(residuals(fit)),
     main="Density Plot: Residuals")
polygon(density(residuals(fit)), col="red")
```



Here we can see that there is no outliers in our box plot. This is a good sign. However, we can see an unusual bimodal distribution in our density plot. We can say however that our density is relatively normal for now.

If the quantiles of our residuals distribution are relatively similar to normal distribution they will map to that of normal distribution. So we will check this as well.

```
qqnorm(residuals(fit),main="QQ plot",pch=19)
qqline(residuals(fit))
```



Here we can see a very small amount of deviation from our normal distribution. This is indicative of a normal distribution.

Now we will perform a Shapiro-Wilk Normality Test to check if our data is significantly different from normal distribution, which is likely the case.

```
shapiro.test(residuals(fit))


    Shapiro-Wilk normality test

data:  residuals(fit)
W = 0.95137, p-value = 0.1459
```

Our P-Value here is greater than 0.05. So this is confirming that our data is not significantly different from normal distribution. We can assume normality.

Question 4b.

```
fit_inv <-glm(time ~ result+Log_WBC, family=inverse.gaussian, data = AG)
summary(fit_inv)
```

```
Call:
glm(formula = time ~ result + Log_WBC, family = inverse.gaussian,
    data = AG)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.97087  -0.38663  -0.04355   0.08659   0.30586

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.0014721  0.0018295   0.805   0.4274
result      -0.0025954  0.0014300  -1.815   0.0795 .
Log_WBC      0.0001713  0.0001259   1.360   0.1838
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for inverse.gaussian family taken to be 0.04450377)

    Null deviance: 4.6863  on 32  degrees of freedom
Residual deviance: 4.2398  on 30  degrees of freedom
AIC: 314.08

Number of Fisher Scoring iterations: 7
```

```
SE_inv <-coef(summary(fit_inv))[,2]
# standard errors
inf_inv<-fit_inv$coef-qnorm(1-0.05/2)*SE_inv
# inferior bound
sup_inv<-fit_inv$coef+qnorm(1-0.05/2)*SE_inv
# Superior bound
inf_inv
```

```
  (Intercept)        result       Log_WBC
-2.113749e-03 -5.398111e-03 -7.546162e-05
```

```
sup_inv
```

```
 (Intercept)       result      Log_WBC
0.0050579099 0.0002072451 0.0004179780
```

Using an inverse-Gaussian GLM, none of our variables are statistically significant. This is due to zero falling between each of our parameters confidence interval. This would be indicative of our model being inadequate using residuals. We can check this using our Shapiro-Wilk test.

```
shapiro.test(residuals(fit_inv))


    Shapiro-Wilk normality test
```

```
data:  residuals(fit_inv)
W = 0.9191, p-value = 0.0173
```

Our P-Value here is less than 0.05. So this is confirming that our data is significantly different from normal distribution. We cannot assume normality.

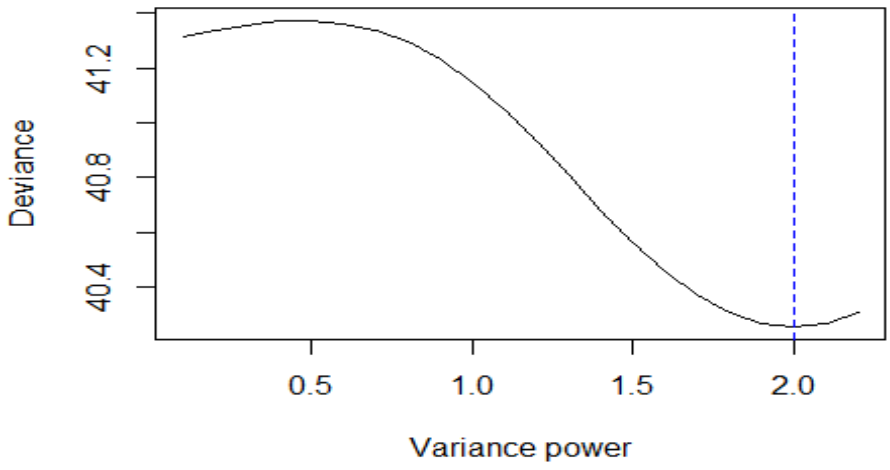
Question 4c.

Let's consider a Quasi-Likelihood model as described in our question.

```
powfam <- quasi(link="log",variance="mu^2")
varpow <- seq(.1,2.2,by=0.1)
devpow <- numeric(length(varpow))
for(i in seq(along=varpow)){
powfam[["variance"]] <- function(mu) mu^varpow[i]
fit.QL <- glm(time ~ result+Log_WBC, family=powfam, data = AG)
devpow[i] <- deviance(fit.QL)}
```

To select the value of P, we will plot our deviance against P and select our minimum P.

```
plot(varpow, devpow, type="l", xlab="Variance power", ylab="Deviance")
min.pow<-which(devpow==min(devpow))
varpow[min.pow]

[1] 2

abline(v=varpow[min.pow], col="blue",lty=2)
```



Here we see our minimum P is 2. We will fit our model and implement this.

```
fit.QL<-glm(time ~ result+Log_WBC, family=quasi(link="log",variance="mu^2"),
data = AG)
summary(fit.QL)


Call:
glm(formula = time ~ result + Log_WBC, family = quasi(link = "log",
    variance = "mu^2"), data = AG)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1744  -1.2638  -0.4219   0.4961   1.9117

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.8127     1.3459   4.319 0.000158 ***
result        1.0214     0.3644   2.803 0.008783 **
Log_WBC      -0.3045     0.1373  -2.218 0.034293 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasi family taken to be 1.088926)

    Null deviance: 58.138  on 32  degrees of freedom
Residual deviance: 40.254  on 30  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 8
```

Here we find our line of best fit: Time = (1.0214)xresult - (0.3045)xLog_wbc + 5.8127.

Now let's provide our 95% confidence interval for the parameters and study their significance.

```
SE.QL<-coef(summary(fit.QL))[,2]
# standard errors
inf.QL<-fit.QL$coef-qnorm(1-0.05/2)*SE.QL
# inferior bound
sup.QL<-fit.QL$coef+qnorm(1-0.05/2)*SE.QL
# Superior bound
inf.QL

(Intercept)      result     Log_WBC
  3.1747678   0.3072608  -0.5736413

sup.QL

(Intercept)      result     Log_WBC
 8.45067627  1.73551638 -0.03539114
```

Here we find all of our parameters are statistically significant in accordance with our confidence interval.