Statistics Exam

January 15, 2020

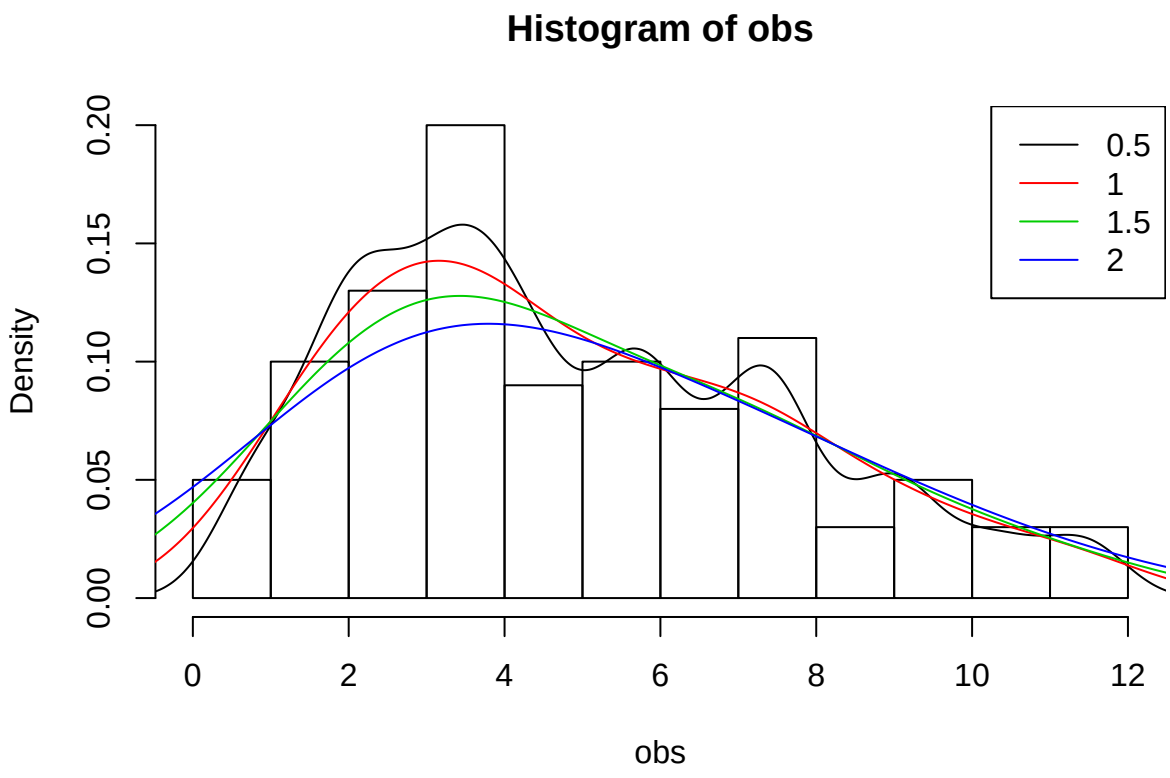
Problem 1

Question 1.1

Plot the histogram of the values and the kernel density estimation (you can try different bandwith and bandwith rules and/or different type of kernel, report the one that you think obtain a good result).

Moreover plot the empirical distribution function and compute and report the first, second and third empirical quantile (Hint: the quantile function in R).

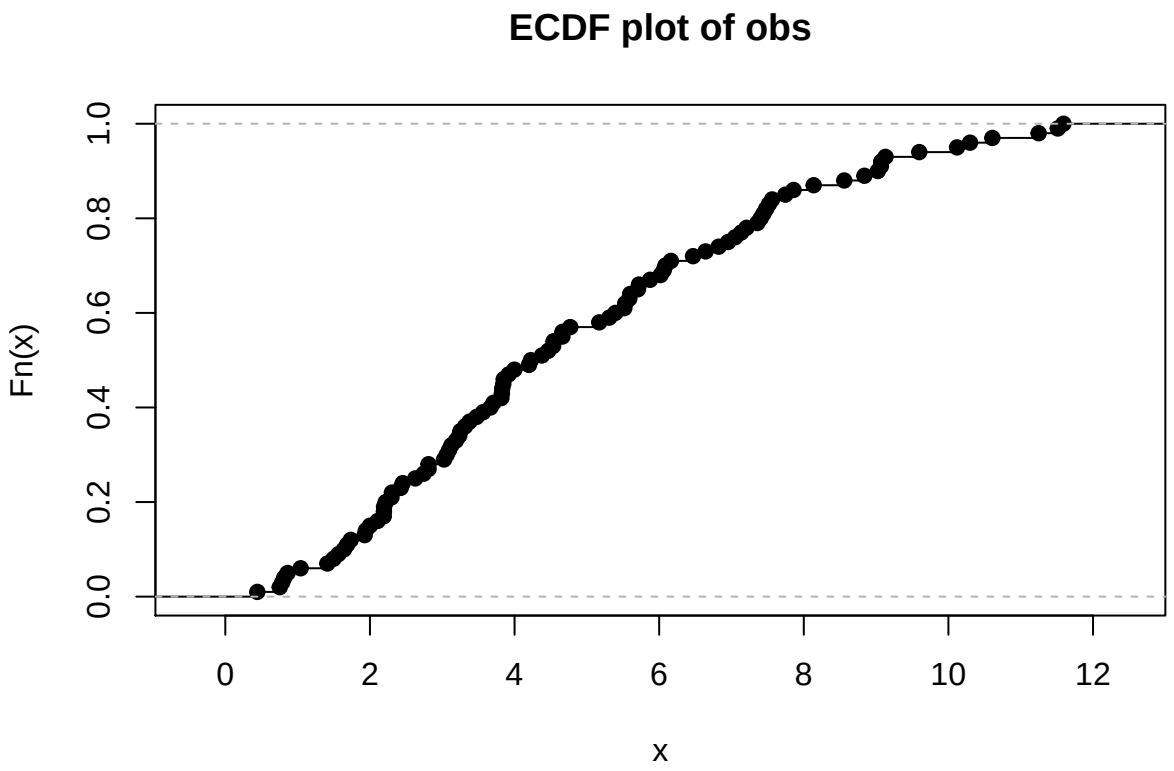
```
hist(obs, breaks = 10, probability = TRUE)
for (bw in seq(0.5,2, 0.5)){
  lines(density(obs, bw = bw), col = bw*2  ) #plot different kernel dnsity functions
}
legend("topright", c("0.5","1", "1.5", "2"), col = seq(0.5,2,0.5)*2, lty = 1)
```



The plot shows kernel density functions with different bandwidths. The default value of 1 (the red line) looks

like the best result.

```
plot(ecdf(obs), main = "ECDF plot of obs")
```




```
q <- data.frame(Quantile = c("1st quantile", "2nd quantile", "3rd quantile"),
                Value = quantile(sort(obs))[2:4])
q
```

```
##         Quantile    Value
## 25% 1st quantile 2.714263
## 50% 2nd quantile 4.302654
## 75% 3rd quantile 6.978805
```

Question 1.2

Fit the following models to the data in obs:

- **Gaussian distribution.**
- **Log-normal distribution.**
- **Gamma distribution.**
- **Exponential distribution.**

Summarize the results in a single plot, showing the histogram of the observations and the fitted densities with different colors, use a legend. Hint: We saw all these models and how to fit them to data during the course, for some of them we know closed form expression of the maximum-likelihood estimators, otherwise numerical method must be used.

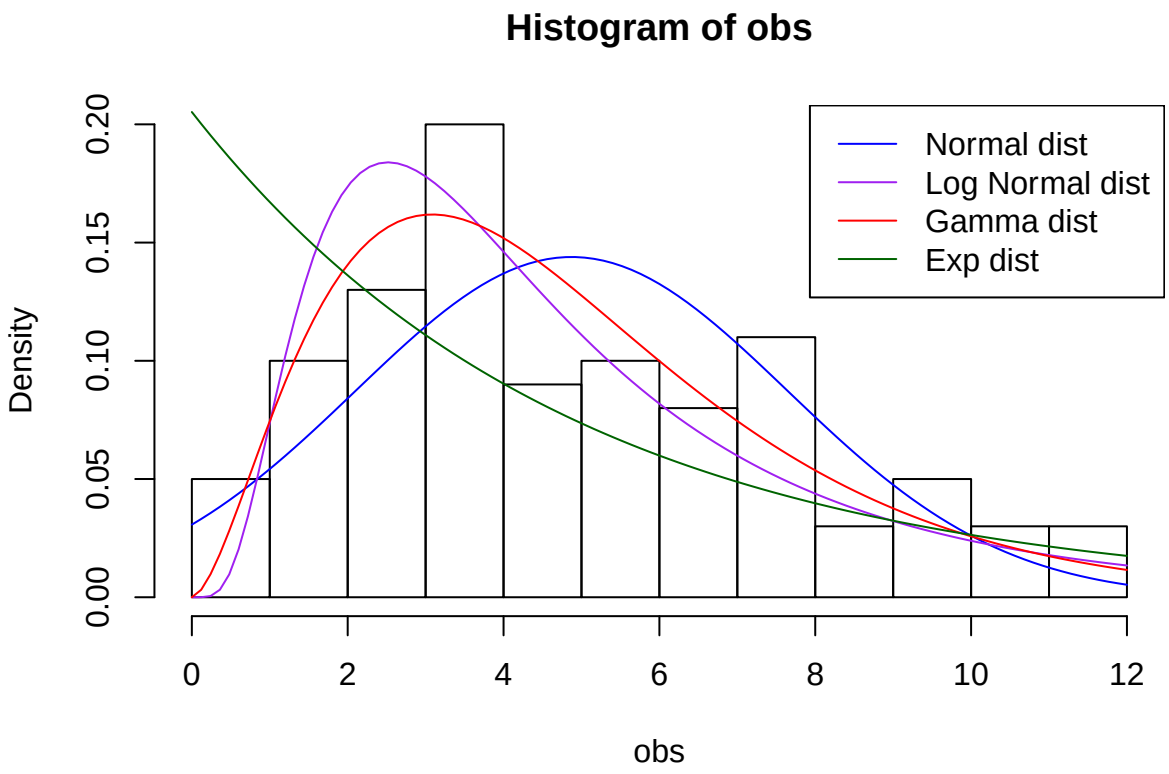
The Gaussian distribution has two parameters, μ and σ, with closed form solutions:

$$\hat{\mu} = \bar{X} \quad \hat{\sigma} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{\mu})^2}$$

This is also true for the log normal distribution.
The Gamma distribution has no closed form solution, so parameters must be obtained numerically.
The exponential distribution has one parameter, λ, with a closed form solution:

$$\hat{\lambda} = \frac{1}{\bar{X}}$$

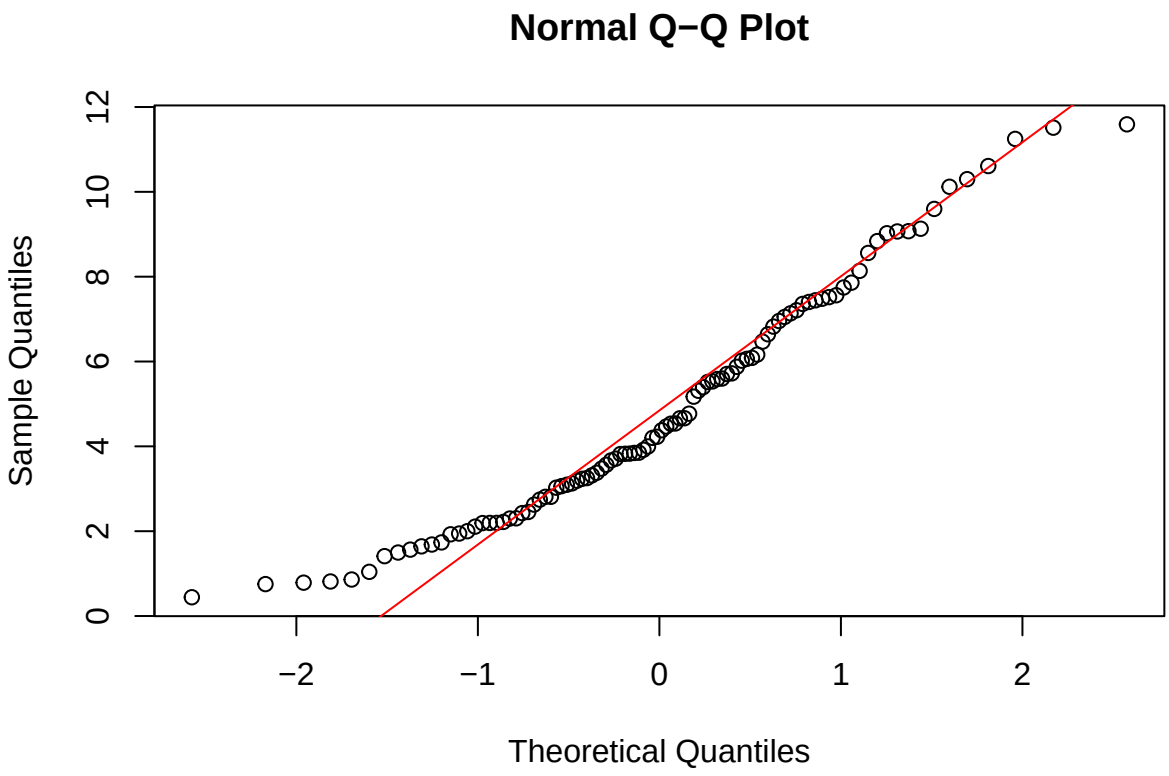
```r
#Maximum likelihood function for gamma distribution
gammaMLH <- function(parameters, xvals){
  return(-sum(dgamma(xvals, shape = parameters[1], rate = parameters[2], log = TRUE)))
}
#store the parameters
parametersGamma <- optim(par = c(1, 1), fn = gammaMLH, xvals = obs)$par
#store the value
valueGamma <- optim(par = c(1, 1), fn = gammaMLH, xvals = obs)$value

#plot
hist(obs, breaks = 10, probability = TRUE)
curve(dnorm(x, mean = mean(obs), sd = sd(obs)),
      add = TRUE, col = "blue") #parameters obtained from formula
curve(dlnorm(x, meanlog = mean(log(obs)), sd = sd(log(obs))),
      add = TRUE, col = "purple") #parameters obtained from formula
curve(dgamma(x, shape = parametersGamma[1], rate = parametersGamma[2]),
      add = TRUE, col = "red") #parameters obtained from numerical method
curve(dexp(x, rate = 1/mean(obs)),
      add = TRUE, col = "darkgreen") #parameters obtained from formula
legend("topright", c("Normal dist", "Log Normal dist", "Gamma dist", "Exp dist"),
       col = c("blue", "purple", "red", "darkgreen"), lty = 1)
```



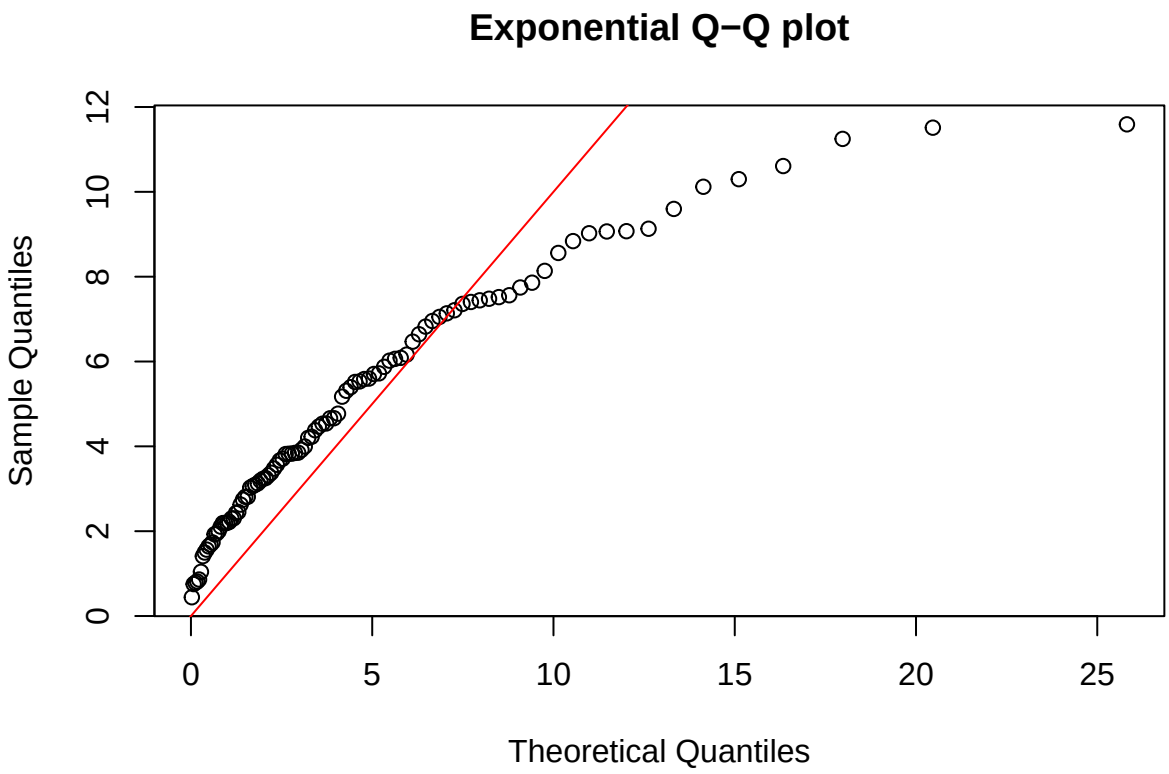
Question 1.3

Judge how the Gaussian distribution and the exponential distribution fit the data using Q-Q plots; What can you observe? Do you think the Gaussian distribution is a good candidate model? Why? Do you think the exponential model fit well the data judging from the Q-Q plot?

```
#qqplot of normal distribution
qqnorm(obs)
qqline(obs, col = "red")
```




```r
#qqplot of exponential distribution
qqplot(qexp(ppoints(obs), rate = 1/mean(obs)), sort(obs),
       xlab = "Theoretical Quantiles", ylab = "Sample Quantiles",
       main = "Exponential Q-Q plot")
abline(0, 1, col = "red")
```



The Gaussian Q-Q plot fits reasonably well since the plot is close to the indentity line, so it can be considered a good candidate model. The exponential model is a much worse fit to the identity line, specially towards the end where the points are very far from the identity line.

Question 1.4

Can we use the likelihood-ratio test to compare the Exponential and Gamma model? Justify your answer and eventually perform the likelihood ratio test, report the obtained p-value and comment the results.

Yes, we can use the likelihood ratio test. When the shape parameter of the gamma distribution is 1, it is identical to the exponential distribution. This means the exponential distribution is nested in the gamma distribution and the likelihood ratio test is valid for nested sitributions.

```
loglikeExp <- sum(dexp(obs, rate = 1/mean(obs), log = TRUE))
loglikeGamma <- -valueGamma
delta <- -2 * (loglikeExp - loglikeGamma)
pvalue <- pchisq(delta, lower.tail = FALSE, df = 1)
pvalue
```

```
## [1] 4.973682e-11
```

The p-value is very small (4.973682e-11) so at $\alpha = 0.01$, we reject the null hypothesis that the simpler model (exponential model) is sufficient to describe the data. Therefore the likelihood-ratio test prefers the gamma distribution model.

Quesion 1.5

Use AIC to select among all the models fitted in Question 1.2, report the computed AIC scores and report the selected model.

```
#create list of models
models <- list(Normal = dnorm(obs, mean = mean(obs), sd = sd(obs), log= TRUE),
               LogNormal = dlnorm(obs, meanlog = mean(log(obs)), sd = log(sd(obs)),
                                  log = TRUE),
               Gamma = dgamma(obs, shape = parametersGamma[1], rate = parametersGamma[2],
                              log = TRUE),
               Exponential = dexp(obs, rate = 1/mean(obs), log = TRUE))
#calculate AIC
AIC <- sapply(models, function(x) -2*sum(x) + 2*2)
AIC[4] <- AIC[4] - 1 #Exponential dist only has one parameter, so adjust
AIC
```

```
##      Normal   LogNormal       Gamma Exponential
##    490.7249    513.8526    477.5539    519.7414
```

We select the Gamma distribution because it has the lowest AIC.

Question 1.6

Consider now the exponential model for the data in obs, build a 95% confidence interval for the rate parameter λ and answer the following question:

Is there enough evidence to say that $\lambda = 0.5$?

Moreover compute and report the p-value for the corresponding Wald test.

We know that the exponential distribution has one parameter, λ, with a closed form solution:

$$\hat{\lambda} = \frac{1}{\bar{X}}$$

The standard error of $\hat{\lambda}$ is:

$$\hat{se}(\hat{\lambda}) = \frac{\hat{\lambda}}{\sqrt{n}}$$

So we can compute the confidence interval analytically:

```
#calculate values
alpha <- 0.05
z <- qnorm(1 - alpha/2)
lambda_est <- 1/(mean(obs))
n <- length(obs)
se_est <- lambda_est/sqrt(n)

#apply the formula
CI <- cbind("2.5%" = lambda_est - z * se_est,
      "97.5%" = lambda_est + z * se_est)
CI #Confidence interval
```

```
##           2.5%     97.5%
## [1,] 0.1649919 0.2454339
```

There is a 95% chance that the true value of λ is in our confidence interval. Since our confidence interval does not include the value of 0.5, we can say there is not enough evidence to suggest that $\lambda = 0.5$.

We want to test $H_0 : \lambda = 0.5$ versus $H_1 : \lambda \neq 0.5$. We can do this using there wald test, where the wald statistic W is:

$$W = \frac{\hat{\theta} - \theta_0}{\hat{se}(\hat{\theta})}$$

So our W value is:

$$W = \frac{\hat{\lambda} - \lambda_0}{\hat{se}(\hat{\lambda})}$$

We calculate the p value using:

$$pvalue = 2F_Z(-|W|)$$

```
wald <- (lambda_est - 0.5) / se_est
pvalue <- 2 * pnorm(-abs(wald))
pvalue
```

```
## [1] 8.588516e-47
```

Question 1.7

Obtain 95% confidence intervals for the parameters of the Gamma distribution using non-parametric bootstrap.

```
#create vector of 1000 parameters
gamma_parameters_bs <- replicate(1000, {
  obs_bs <- sample(obs, replace = TRUE)
  para_bs <- optim(par = c(1,1), fn = gammaMLH, xvals = obs_bs)$par
  return(c(shape = para_bs[1], rate = para_bs[2]))
})

#calculate standard deviation of each parameter
se <- apply(gamma_parameters_bs, MARGIN = 1, function(x) sd(x))

#find the confidence interval of each parameter
z <- qnorm(1 - 0.05 / 2)
se <- apply(gamma_parameters_bs, MARGIN = 1, function(x) sd(x))
CI <- matrix(parametersGamma + z * se %*% t(c(-1, +1)),
       dimnames = list(c("shape", "rate"), c("2.5%", "97.5%")), ncol = 2)
CI #confidence intervals
```

```
##            2.5%     97.5%
## shape 2.0193710 3.4269843
## rate  0.4129892 0.7045564
```

Problem 2

Question 2.1

Using the data in applejuice fit a simple logistic regression model for the variable growth using all the other variables as covariates. Report the estimated coefficients, the standard errors and comment on which coefficient is significant at a level 0.01.

```
applejuice$growth <- as.factor(applejuice$growth)
apple_simple <- glm(growth ~ ., family = "binomial", data = applejuice)
summary(apple_simple)
```

```
##
## Call:
## glm(formula = growth ~ ., family = "binomial", data = applejuice)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3614  -0.3990  -0.1585   0.6306   1.6200
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.24633    3.21864  -2.251 0.024362 *
## ph           1.88595    0.54123   3.485 0.000493 ***
## nisin       -0.06628    0.01905  -3.479 0.000503 ***
## temp         0.11042    0.04769   2.316 0.020585 *
## brix        -0.31173    0.14317  -2.177 0.029458 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 95.945  on 73  degrees of freedom
## Residual deviance: 52.331  on 69  degrees of freedom
## AIC: 62.331
##
## Number of Fisher Scoring iterations: 6
```

At a significance level of 0.01, only the pH and nisin coefficients are significant since they have a p value smaller than 0.01.

Question 2.2

Using the data in applejuice, fit the logistic regression model where we add three more terms corresponding to the products between brix and the remaining variables. Report the estimated coefficients and the corresponding standard errors.

```
apple_advanced <- update(apple_simple,
                         . ~ . + I(brix * ph) + I(brix * temp) + I(brix * nisin))
summary(apple_advanced)
```

```
##
## Call:
## glm(formula = growth ~ ph + nisin + temp + brix + I(brix * ph) +
##     I(brix * temp) + I(brix * nisin), family = "binomial", data = applejuice)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.02710  -0.27399  -0.00002   0.00207   2.44630
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -260.74530  129.81297  -2.009   0.0446 *
## ph                43.00190   21.91533   1.962   0.0497 *
## nisin             -1.90315    0.95384  -1.995   0.0460 *
## temp               3.18523    1.53655   2.073   0.0382 *
## brix              13.41225    6.82054   1.966   0.0492 *
```

```
## I(brix * ph)      -2.21485    1.14186  -1.940   0.0524 .
## I(brix * temp)    -0.16884    0.08321  -2.029   0.0425 *
## I(brix * nisin)    0.10020    0.05028   1.993   0.0463 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 95.945  on 73  degrees of freedom
## Residual deviance: 21.953  on 66  degrees of freedom
## AIC: 37.953
##
## Number of Fisher Scoring iterations: 10
```

Question 2.3

Perform model selection using AIC and BIC between the two models obtained in Questions 2.1 and 2.2. Can we use the likelihood ratio test to select between the two models ? If yes perform the likelihood ratio test and comment the results, otherwise justify your answer.

```
AIC(apple_simple, apple_advanced)
```

```
##                df      AIC
## apple_simple    5 62.33065
## apple_advanced  8 37.95315
```

```
BIC(apple_simple, apple_advanced)
```

```
##                df      BIC
## apple_simple    5 73.85098
## apple_advanced  8 56.38567
```

The advanced model has lower values for both AIC and BIC so it should be selected.

Since all the parameters of the simple model are also parameters of the advanced model, the simple model is nested in the advanced model. Since the models are nested, Likelihood ratio test can be performed.

```
anova(apple_simple, apple_advanced, test="LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: growth ~ ph + nisin + temp + brix
## Model 2: growth ~ ph + nisin + temp + brix + I(brix * ph) + I(brix * temp) +
##     I(brix * nisin)
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        69     52.331
## 2        66     21.953  3   30.378 1.149e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is very small (1.149e-06) so at $\alpha = 0.01$, we reject the null hypothesis that the simpler model is sufficient to describe the data. Therefore the likelihood-ratio test prefers the advanced model. This is in agreement with the AIC and BIC results.

Question 2.4

Compute 99% confidence intervals for the coefficients in the simple logistic regression model fitted in Question 2.1. (You can use the built-in function confint). Then, based on the computed confidence intervals, answer the following questions:

- **Can we state, with 99% confidence, that the ph coeficient is greater than zero?**
- **What can we say about the brix coeffcient?**
- **Can we say, that an higher value of ph will result in a predicted higher probability of growth of Alicyclobacillus?**

```
confint(apple_simple, level = 0.99)
```

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

```
##                       0.5 %      99.5 %
## (Intercept) -17.281191477  0.03982106
## ph            0.666526559  3.56603330
## nisin        -0.127816521 -0.02548041
## temp          0.001884823  0.25743121
## brix         -0.744106674  0.01941213
```

- There is a 99% chance that the true coefficient of the pH parameter is in our confidence interval. Since our confidence interval does not include the value of 0, we can say with 99% confidencethat the pH coefficient is greater than 0.
- The brix coefficient ranges between -0.744 and 0.019, so it is possible for the coefficient to be zero and not affect the regression line, but it is more likely that the brix coefficient is a small negative number which would mean the nisin concentration negatively affects the growth, but only slightly.
- Since the 99% confidence interval indicates the coefficient of pH will be a postiive value, we can say that a higher value of pH will result in a predicted higher probability of growth.

Question 2.5

For the two models fitted in Questions 2.1 and 2.2 compute the predicted probabilities of growth for the data in applejuice test. Compare the predicted probabilities obtained from the two models with the probabilities estimated (growth p in the data frame applejuice test). Which model performs better in estimating the probability of growth under the conditions in the applejuice test data? Comment the results also with respect to the model selection performed in Question 2.3.

```
a1_pred <- predict(apple_simple, newdata = applejuice_test, type = "response")
a2_pred <- predict(apple_advanced, newdata = applejuice_test, type = "response")
apple_predictions <- data.frame(apple_simple = a1_pred,
           apple_advanced = a2_pred,
           test_data = applejuice_test$growth_p)
apple_predictions
```

```
##   apple_simple apple_advanced test_data
## 1   0.29484940   3.072582e-04       0.0
## 2   0.42405198   5.582004e-02       0.2
## 3   0.82917472   9.999996e-01       1.0
## 4   0.03948246   2.220446e-16       0.0
## 5   0.06749440   2.287283e-13       0.0
## 6   0.32303048   9.396739e-06       0.0
```

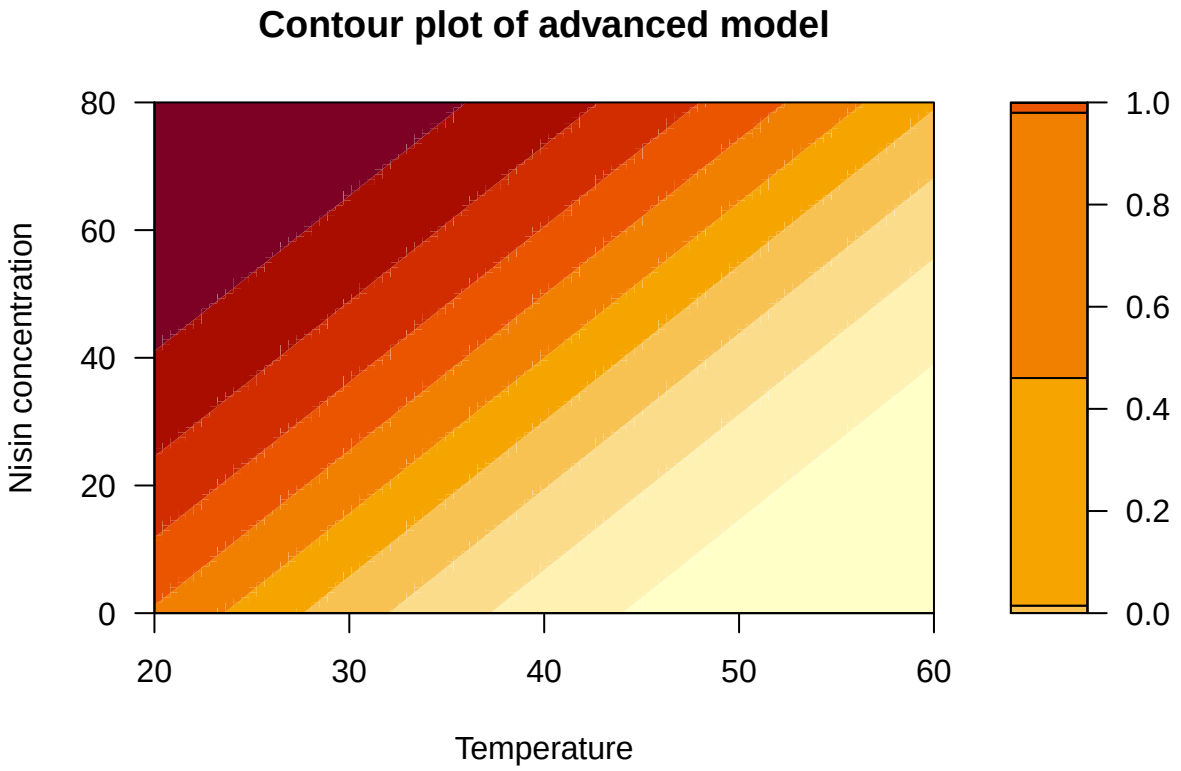
We can see that the advanced model predicts values much closer to 0 or 1, which matches the test data more than the simple model, therefore the advanced model is better. This agrees with our previous results, where

we found the advanced model was chosen during AIC, BIC selection and when the Likelihood ratio test was used.

Question 2.6

Use the model obtained in Question 2.2 to estimate the probability of growth of Alicyclobacillus Acidoterrestris as a function of the temperature and the Nisin concentration when the Brix concentration is fixed at 14 and the ph is equal to 5. In particular you can obtain the estimated probabilities of growth for temperatures varying between 20 and 60, and Nisin concentrations between 0 and 80 (You will thus obtain estimated probabilities for a grid of temperature and nisin concentration values). Resume the different estimated probabilities in a plot. You can use a contour plot or different colours.

```
#create dataframe of combinations
temp <- seq(20, 60, length.out = 100)
nisin <- seq(0, 80, length.out = 100)
data_grid <- expand.grid(temp = temp, nisin = nisin)
data_grid$ph <- 5
data_grid$brix <- 14

#use datafram to make predictions
pred_grid <- predict(apple_advanced, newdata = data_grid, type = "response")
#convert vector of predictions to matrix
pred_mat <- matrix(pred_grid, nrow=100, byrow = TRUE)
#use matrix to create contour plot
filled.contour(temp, nisin, pred_mat,
               levels = quantile(pred_mat, probs = seq(0, 1, 0.1)), #create contour lines
               xlab= "Temperature", ylab = "Nisin concentration",
               main = "Contour plot of advanced model")
```

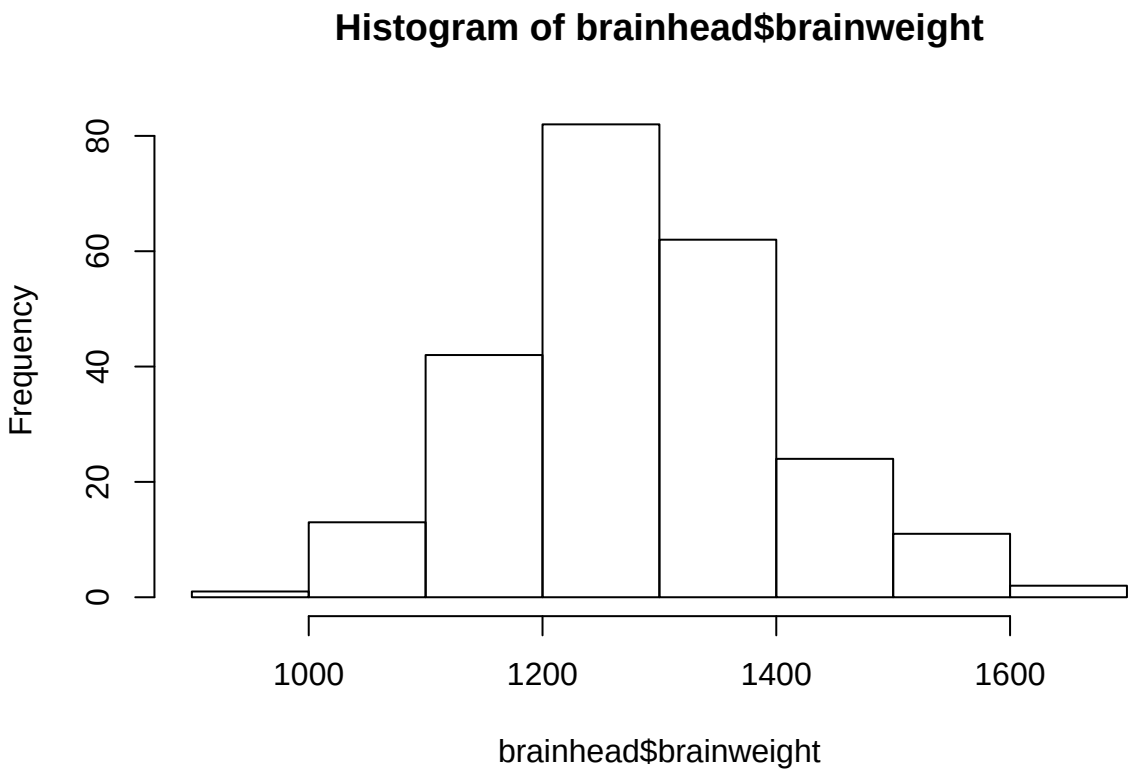


Problem 3

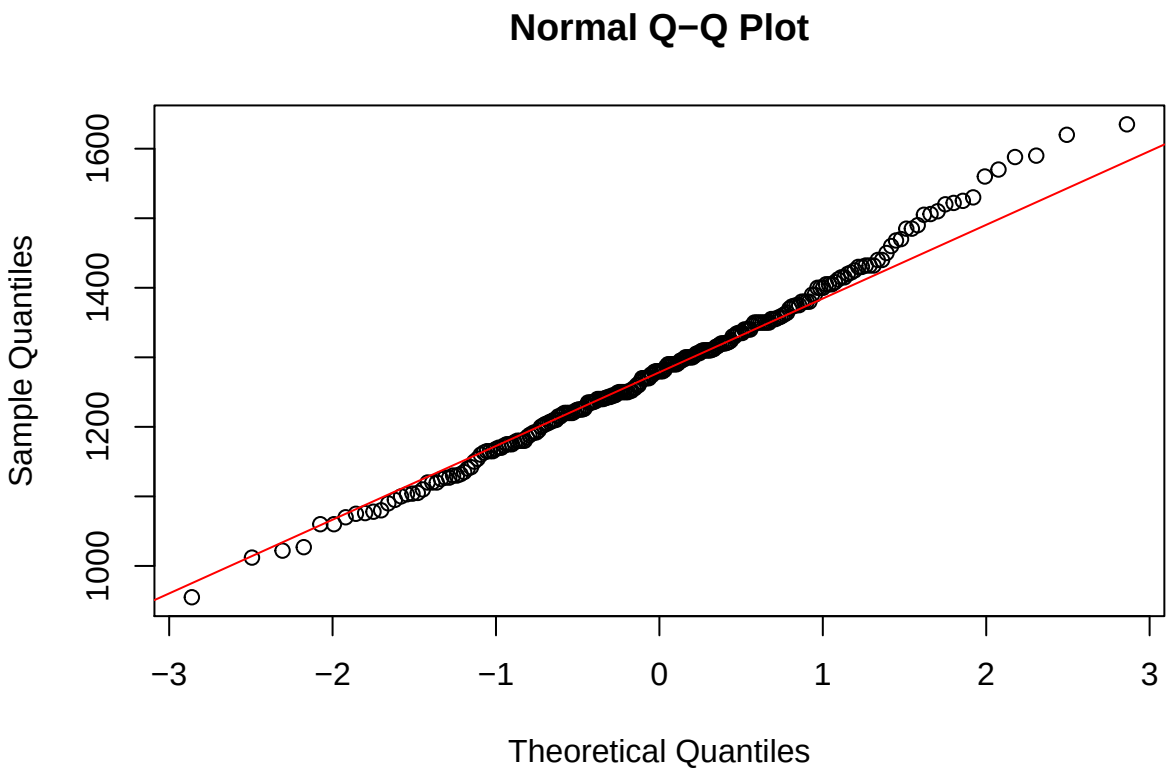
Question 3.1

Check (and comment) if the distribution of the brain weight can be assumed Gaussian, use the tools we have seen in the course (Q-Q plot,histogram).

```
hist(brainhead$brainweight)
```




```
qqnorm(brainhead$brainweight)
qqline(brainhead$brainweight, col = "red")
```



The histogram has many values in the center and fewer near the ends so it appears to follow a Gaussian distribution. The Gaussian Q-Q plot fits well since the plot is close to the indentity line, so the Gaussian distribution can be considered a good approximation of the data.

Question 3.2

Fit now the Gaussian distribution to the brain weight values using all the observations in the data frame brainhead. Report the estimated parameters and plot the obtained density on top of the histogram of the data. Obtain and report a 95% confidence interval for the mean parameter, use the method you think is the most appropriate and comment the choice (you can also use different methods and comment the differences).

The Gaussian distribution has two parameters, μ and σ, with closed form solutions:

$$\hat{\mu} = \bar{X} \quad \hat{\sigma} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{\mu})^2}$$

Therefore the confidence interval can be calculated analytically:

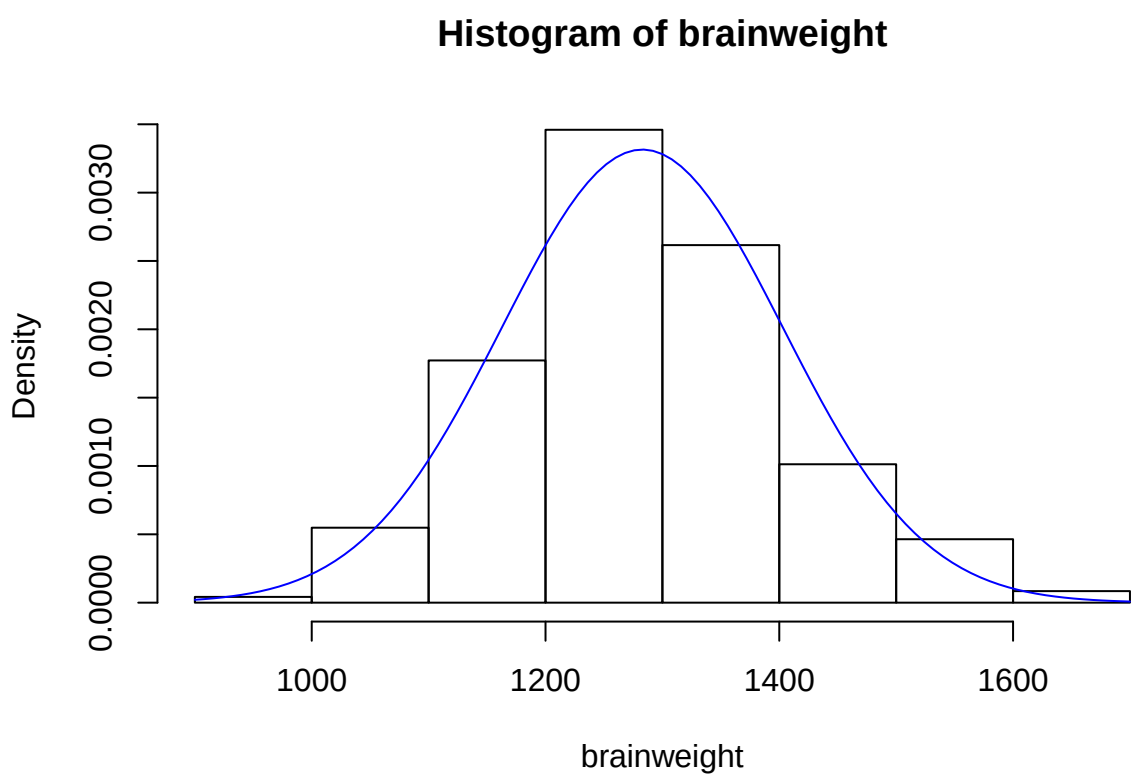
```
brainweight <- brainhead$brainweight

#report estimated parameters
paste("Mu estimate =", mean(brainweight))
```

```
## [1] "Mu estimate = 1282.87341772152"
```

```
paste("Sigma estimate =", sd(brainweight))
```

```
## [1] "Sigma estimate = 120.340445786457"
```

```r
#fit distribution on histogram
hist(brainweight, probability = TRUE)
curve(dnorm(x, mean = mean(brainweight), sd = sd(brainweight)),
      add = TRUE, col = "blue")
```




```r
#calculate values for confidence interval (analytical method)
alpha <- 0.05
z <- qnorm(1 - alpha/2)
n <- length(brainweight)
sem <- mean(brainweight)/ sqrt(n)

#apply the formula
CI_mean <- cbind("2.5%" = mean(brainweight) - z * sem,
      "97.5%" =  mean(brainweight) + z * sem)
CI_mean #Confidence interval
```

```
##          2.5%  97.5%
## [1,] 1119.547 1446.2
```

The 95% confidence interval of the mean can be calculated by bootstrapping or by using the analytical method. Since a closed form solution exists for the μ parameter of a Gaussian distribution, it is better to use the analytical method since it is less computationally expensive and the result is accurate.

Question 3.3

Is there a significant difference between the mean value of the brain weight for old (age > 46) and young (age < 46) subjects? Answer the question with an appropriate statistical procedure and comment the results.

We want to test $H_0 : \mu_1 = \mu_2$ versus $H_1 : \mu_1 \neq \mu_2$. We can use a t-test to test if there is a statistically significant difference in means between two things.

```
brainweight_young <- brainhead$brainweight[brainhead$agerange == 1]
brainweight_old <- brainhead$brainweight[brainhead$agerange == 2]
t.test(brainweight_young, brainweight_old, alternative = "two.sided")
```

```
##
##  Welch Two Sample t-test
##
## data:  brainweight_young and brainweight_old
## t = 2.6428, df = 232.37, p-value = 0.008782
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  10.38279 71.21592
## sample estimates:
## mean of x mean of y
##  1304.736  1263.937
```

The t-test returns a p value of 0.008782 which is very low so at $\alpha = 0.01$, we can reject the null hypothesis. So we accept the alternate hypothesis that there is a difference in the mean value of the brain weight between young subjects and old subjects.

Question 3.4

Fit the simple linear regression model using three different data sets: (1) the full data in brainhead, (2) the observations from the young subjects (agerange = 1) and (3) the observations from the old subjects (agerange = 2).

Resume the three linear fitted models in a single plot: draw a scatter plot of the observations differentiating with two colors young and old subjects; add then the plot of the linear regressions obtained (use the same colors to differentiate the regressions for old and young subjects).

```
#subset data
headsize_young <- brainhead$headsize[brainhead$agerange == 1]
headsize_old <- brainhead$headsize[brainhead$agerange == 2]
young_df <- brainhead[brainhead$agerange == 1,]
old_df <- brainhead[brainhead$agerange == 2,]

#create models
brain_model <- lm(brainweight ~ headsize, data= brainhead)
summary(brain_model)
```

```
##
## Call:
## lm(formula = brainweight ~ headsize, data = brainhead)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -175.98  -49.76   -1.76   46.60  242.34
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 325.57342   47.14085   6.906 4.61e-11 ***
## headsize      0.26343    0.01291  20.409  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 72.43 on 235 degrees of freedom
## Multiple R-squared:  0.6393, Adjusted R-squared:  0.6378
## F-statistic: 416.5 on 1 and 235 DF,  p-value: < 2.2e-16
```

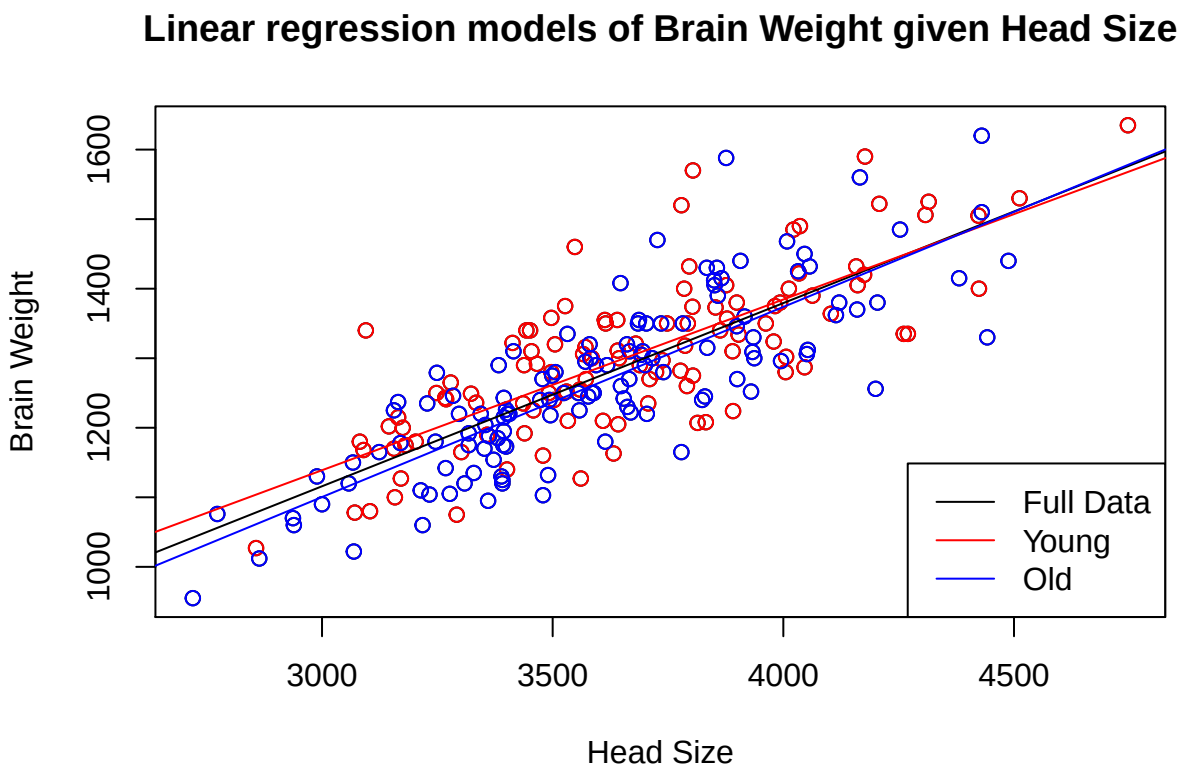
```
brain_model_young <- lm(brainweight ~ headsize, data= young_df)
summary(brain_model_young)
```

```
##
## Call:
## lm(formula = brainweight ~ headsize, data = young_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -149.677  -47.991   -0.128   38.614  233.659
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 402.34144   72.60815   5.541 2.14e-07 ***
## headsize      0.24553    0.01966  12.489  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 74.8 on 108 degrees of freedom
## Multiple R-squared:  0.5909, Adjusted R-squared:  0.5871
## F-statistic:   156 on 1 and 108 DF,  p-value: < 2.2e-16
```

```
brain_model_old <- lm(brainweight ~ headsize, data= old_df)
summary(brain_model_old)
```

```
##
## Call:
## lm(formula = brainweight ~ headsize, data = old_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -172.458  -49.221    3.758   39.716  248.122
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 280.20329   61.24916   4.575 1.13e-05 ***
## headsize      0.27339    0.01694  16.142  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 69.13 on 125 degrees of freedom
## Multiple R-squared:  0.6758, Adjusted R-squared:  0.6732
## F-statistic: 260.6 on 1 and 125 DF,  p-value: < 2.2e-16
```

```r
#plot scatterplot
plot(brainhead$headsize, brainhead$brainweight,
     xlab = "Head Size", ylab = "Brain Weight",
     main = "Linear regression models of Brain Weight given Head Size")
points(headsize_young, brainweight_young, col = "red")
points(headsize_old, brainweight_old, col = "blue")
abline(brain_model)
abline(brain_model_young, col = "red")
abline(brain_model_old, col = "blue")
legend("bottomright", c("Full Data", "Young", "Old"), col = c(1,2, "blue"), lty = 1)
```



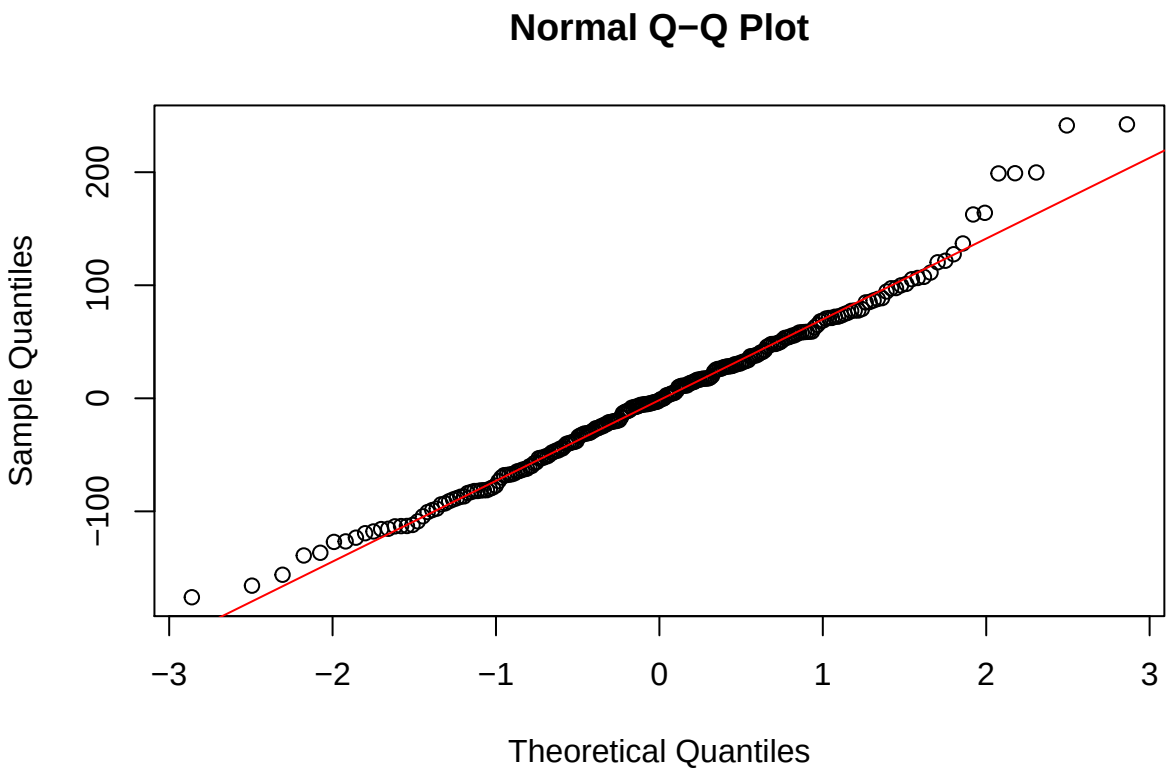
Question 3.5

Investigate if the linear model with Gaussian noise is appropriate by plotting

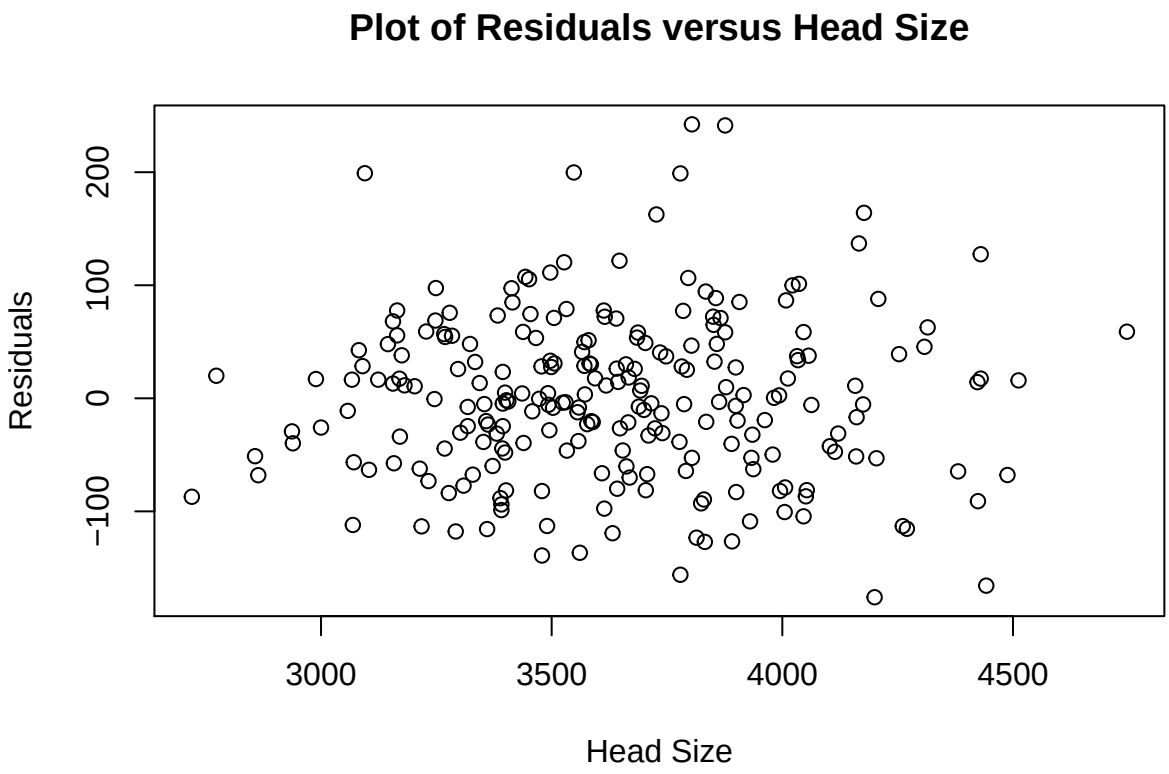
- **the QQ-plot of the residuals against the normal distribution and**
- **the residuals versus the headsize variable.**

Comment the obtained plots.

```r
##qqplot
qqnorm(residuals(brain_model))
qqline(residuals(brain_model), col = "red")
```




```
#residuals versus headsize
plot(brainhead$headsize, residuals(brain_model),
     xlab = "Head Size", ylab = "Residuals",
     main = "Plot of Residuals versus Head Size")
```



The residuals fit the Q-Q plot very well since the plot is close to the identity line. This suggests the error in our model is Gaussian. The scatter plot of headsize versus residuals also seems random. Since no pattern is observed this also suggests the error in our model is Gaussian in nature.

Question 3.6

Consider the linear regression in Question 3.4 using all the data (you can forget the agerange variable here). Obtain an estimation of the mean square error using 10-fold cross validation.

```
#this function takes in a dataframe and splits it into k parts
kfold <- function(k, data){
  groups <- list()
  m <- nrow(data) %/% k #group size
  for(i in 1:k){
    if(i == k){ # if last group, include all remaining rows
      groups[[i]] <- ((i-1)*m+1):(nrow(data))
    }
    else{
      groups[[i]] <- ((i-1)*m + 1):(i*m)
    }
  }
  return(groups)
}

#this function performs k fold cross validation and returns the mean squared error
crossvalid <- function(groups, data){
  output <- c()
```

```
  k <- length(groups)
  for(i in 1:k){
    #fit model with all but i row
    model_i <- lm("brainweight ~ headsize", data = brainhead[-groups[[i]],])
    #predict value for row i
    prediction_i <- predict(model_i, newdata = brainhead[groups[[i]],])
    #calculate list of mean squared errors
    mse <- (prediction_i - brainhead$brainweight[groups[[i]]])^2
    #calculate mean of list and and save it
    output[i] <- mean(mse)
  }
  return(mean(output))
}

bh_groups <- kfold(10, brainhead)
mse <- crossvalid(bh_groups, brainhead)
paste("Estimation of Mean Square Error:", mse)
```

```
## [1] "Estimation of Mean Square Error: 5413.53111759661"
```

Question 3.7

Consider the polynomial regression model. Perform the appropriate model selection methods between the above polynomial regression and the simple linear regression, is the simpler model sufficient?

```
poly_brain <- lm(brainweight ~ headsize + I(headsize^2), data = brainhead)
anova(brain_model, poly_brain, test="F")
```

```
## Analysis of Variance Table
##
## Model 1: brainweight ~ headsize
## Model 2: brainweight ~ headsize + I(headsize^2)
##   Res.Df     RSS Df Sum of Sq     F Pr(>F)
## 1    235 1232728
## 2    234 1224803  1    7924.7 1.514 0.2198
```

```
AIC(brain_model, poly_brain)
```

```
##             df      AIC
## brain_model  3 2706.510
## poly_brain   4 2706.982
```

```
BIC(brain_model, poly_brain)
```

```
##             df      BIC
## brain_model  3 2716.914
## poly_brain   4 2720.854
```

The anova test gives a p value of 0.2198. At $\alpha = 0.05$, this is not suffcient to reject the null hypothesis. So the simpler model is sufficient to describe the data. The AIC and BIC both return a lower value for the simpler model, so we must select it. This aligns with our anova result so the simpler model is sufficient.