HW1 Peer Assessment

Part A. ANOVA

Additional Material: ANOVA tutorial

https://datascienceplus.com/one-way-anova-in-r/

Jet lag is a common problem for people traveling across multiple time zones, but people can gradually adjust to the new time zone since the exposure of the shifted light schedule to their eyes can resets the internal circadian rhythm in a process called "phase shift". Campbell and Murphy (1998) in a highly controversial study reported that the human circadian clock can also be reset by only exposing the back of the knee to light, with some hailing this as a major discovery and others challenging aspects of the experimental design. The table below is taken from a later experiment by Wright and Czeisler (2002) that re-examined the phenomenon. The new experiment measured circadian rhythm through the daily cycle of melatonin production in 22 subjects randomly assigned to one of three light treatments. Subjects were woken from sleep and for three hours were exposed to bright lights applied to the eyes only, to the knees only or to neither (control group). The effects of treatment to the circadian rhythm were measured two days later by the magnitude of phase shift (measured in hours) in each subject's daily cycle of melatonin production. A negative measurement indicates a delay in melatonin production, a predicted effect of light treatment, while a positive number indicates an advance.

Raw data of phase shift, in hours, for the circadian rhythm experiment

Treatment	Phase Shift (hr)
Control	0.53, 0.36, 0.20, -0.37, -0.60, -0.64, -0.68, -1.27
Knees	0.73, 0.31, 0.03, -0.29, -0.56, -0.96, -1.61
Eyes	-0.78, -0.86, -1.35, -1.48, -1.52, -2.04, -2.83

Question A1 - 3 pts

Consider the following incomplete R output:

Source	Df	Sum of Squares	Mean Squares	F-statistics	p-value
Treatments	?	?	3.6122	?	0.004
Error	?	9.415	?		
TOTAL	?	?			

Fill in the missing values in the analysis of the variance table.

Answer:

The complete R output is:

Source	Df	Sum of Squares	Mean Squares	F-statistics	p-value
Treatments	2	7.224	3.6122	7.289	0.004
Error	19	9.415	0.496		
TOTAL	21	16.639			

Which came from applying the following R code:

```
#data
data_anova = read.table(
  "C:\\Users\\adri_\\Documents\\Gatech\\ISYE6414\\Homeworks\\1_Simple_linear_regression_ANOVA\\jetlag1.t
  header = TRUE)
head(data_anova,4)
```

```
##   Shift Treatment
## 1  0.53   Control
## 2  0.36   Control
## 3  0.20   Control
## 4 -0.37   Control
```

```
shift = data_anova$Shift
treatment_type = data_anova$Treatment

#Fitting the model
anova_model = aov(shift ~ treatment_type)
summary(anova_model)
```

```
##                Df Sum Sq Mean Sq F value  Pr(>F)
## treatment_type  2  7.224   3.612   7.289 0.00447 **
## Residuals      19  9.415   0.496
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question A2 - 3 pts

Use μ_1, μ_2, and μ_3 as notation for the three mean parameters and define these parameters clearly based on the context of the topic above. Find the estimates of these parameters.

Answer

The value of μ_1 is -0.3087, which represents the mean of Control, the value of μ_2 is -1.551 which represents the mean of Eyes and the value of μ_3 is -0.3357 which represents the mean of Knees. It means that this is the average time (in hours) that it takes individuals to have a phase shift when no light is applied (control experiment), when light is applied in the eyes and when ligth is applied on the knees with the purpose of adjusting to a new time zone when jetlag is present.

These parameters were calculated using the following code:

```
## Obtain estimated means
model.tables(anova_model, type="means")
```

```
## Tables of means
## Grand mean
##
## -0.7127273
##
##  treatment_type
##     Control   Eyes   Knees
##     -0.3087 -1.551 -0.3357
## rep  8.0000  7.000  7.0000
```

Question A3 - 5 pts

Use the ANOVA table in Question A1 to answer the following questions:

a. **1 pts** Write the null hypothesis of the ANOVA F-test, H_0

Answer

All means are equal.

b. **1 pts** Write the alternative hypothesis of the ANOVA F-test, H_A

Answer At least one pair of means are different.

c. **1 pts** Fill in the blanks for the degrees of freedom of the ANOVA F-test statistic: F(_____, _____)

Answer
$F(2, 19)$

d. **1 pts** What is the p-value of the ANOVA F-test?

Answer

P-value for the F-test = P(F > F_0), where F ~ $F_{(k-1,N-k)}$

The p-value is 0.00447 as shown in the R code above.

e. **1 pts** According the the results of the ANOVA F-test, does light treatment affect phase shift? Use an α-level of 0.05.

Yes, light treatment affects phase shift because the p-value is very small and less than 0.05

Part B. Simple Linear Regression

We are going to use regression analysis to estimate the performance of CPUs based on the maximum number of channels in the CPU. This data set comes from the UCI Machine Learning Repository.

The data file includes the following columns:

- *vendor*: vendor of the CPU

- *chmax*: maximum channels in the CPU
- *performance*: published relative performance of the CPU

The data is in the file "machine.csv". To read the data in R, save the file in your working directory (make sure you have changed the directory if different from the R working directory) and read the data using the R function `read.csv()`.

```
# Read in the data
data = read.csv(
  "C:\\Users\\adri_\\Documents\\Gatech\\ISYE6414\\Homeworks\\1_Simple_linear_regression_ANOVA\\machine.
  head = TRUE, sep = ",")
# Show the first few rows of data
head(data, 3)
```

```
##    vendor chmax performance
## 1 adviser   128         198
## 2  amdahl    32         269
## 3  amdahl    32         220
```

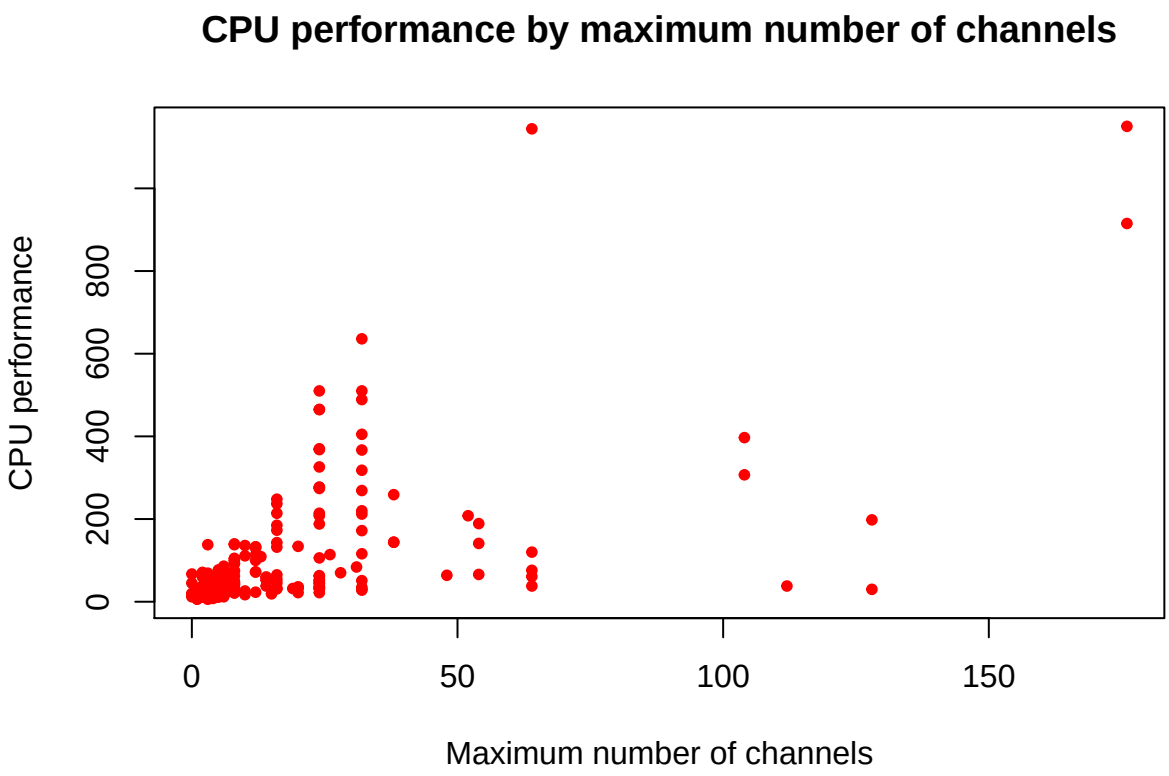
Question B1: Exploratory Data Analysis - 9 pts

a. **3 pts** Use a scatter plot to describe the relationship between CPU performance and the maximum number of channels. Describe the general trend (direction and form). Include plots and R-code used.

```
# Your code here...

performance <- data[,3]   # performance is the response variable
chmax <- data[,2]

plot(chmax,performance,
     xlab="Maximum number of channels",
     ylab="CPU performance",
     main="CPU performance by maximum number of channels",
     col = "red",
     pch=20)
```



The scatter plot shows a lot of "noise" (or dispersion) that presents itself as a big "cone" shape to the right of the data in the relationship between maximum number of channels and CPU performance. We can also identify some observations that may be outliers.

In the general sense, there seems to be a positive trend. The bigger the maximum number of channels is in the equipment, the bigger the CPU performance is.

b. **3 pts** What is the value of the correlation coefficient between *performance* and *chmax*? Please interpret the strength of the correlation based on the correlation coefficient.

```
# Your code here...

cor(chmax,performance)
```

```
## [1] 0.6052093
```

The maximum number of channels and the CPU performance observations are moderately linearly correlated, with a correlation coefficient of 0.6. The correlation is positive, meaning that as the number of channels increase, the value of the CPU performance variable increases too.

c. **2 pts** Based on this exploratory analysis, would you recommend a simple linear regression model for the relationship?

No, I would not recommend a simple linear regression model with the raw variables, as the eploratory analysis shows that we might be violating the assumptions for a valid model, namely linearity (which is not clear) and constant variance (as the data clearly shows larger variation for larger values of number of channels).

d. **1 pts** Based on the analysis above, would you pursue a transformation of the data? *Do not transform the data.*

Yes, I would definitely pursue transforming the data to improve the linear correlation between the independent and dependent variable, and make the variance constant, thus satisfying the assumptions for linear regression models.

Question B2: Fitting the Simple Linear Regression Model - 11 pts

Fit a linear regression model, named *model1*, to evaluate the relationship between performance and the maximum number of channels. *Do not transform the data.* The function you should use in R is:

```
# Your code here...
model1 = lm(performance ~ chmax, data)
summary(model1)
```

```
##
## Call:
## lm(formula = performance ~ chmax, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -486.47  -42.20  -22.20   20.31  867.15
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.2252    10.8587   3.428 0.000733 ***
## chmax         3.7441     0.3423  10.938  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 128.3 on 207 degrees of freedom
## Multiple R-squared:  0.3663, Adjusted R-squared:  0.3632
## F-statistic: 119.6 on 1 and 207 DF,  p-value: < 2.2e-16
```

```
tvalue = 10.938

pvalue = 1-pt(tvalue,207)
pvalue
```

```
## [1] 0
```

```
confint(model1, level = 0.95)
```

```
##                 2.5 %    97.5 %
## (Intercept) 15.817392 58.633048
## chmax        3.069251  4.418926
```

```
confint(model1, level = 0.99)
```

```
##                0.5 %    99.5 %
## (Intercept) 8.994891 65.455549
## chmax       2.854185  4.633991
```

a. **3 pts** What are the model parameters and what are their estimates?

β_0 is the intercept and was estimated as 37.2252 and β_1 is the slope and was estimated as 3.7441

b. **2 pts** Write down the estimated simple linear regression equation.

performance = 37.2252 + 3.7441*(chmax)

c. **2 pts** Interpret the estimated value of the β_1 parameter in the context of the problem.

The CPU performance increases by 3.7441 with each 1 additional channel.

d. **2 pts** Find a 95% confidence interval for the β_1 parameter. Is β_1 statistically significant at this level?

The 95% confidence interval for the β_1 parameter is (3.069251 , 4.418926).

Yes, β_1 is statistically significant, as evidenced by its p-value of 2e-16, which is much smaller than 0.05 (significance threshold at 95% confidence).

e. **2 pts** Is β_1 statistically significantly positive at an α-level of 0.01? What is the approximate p-value of this test?

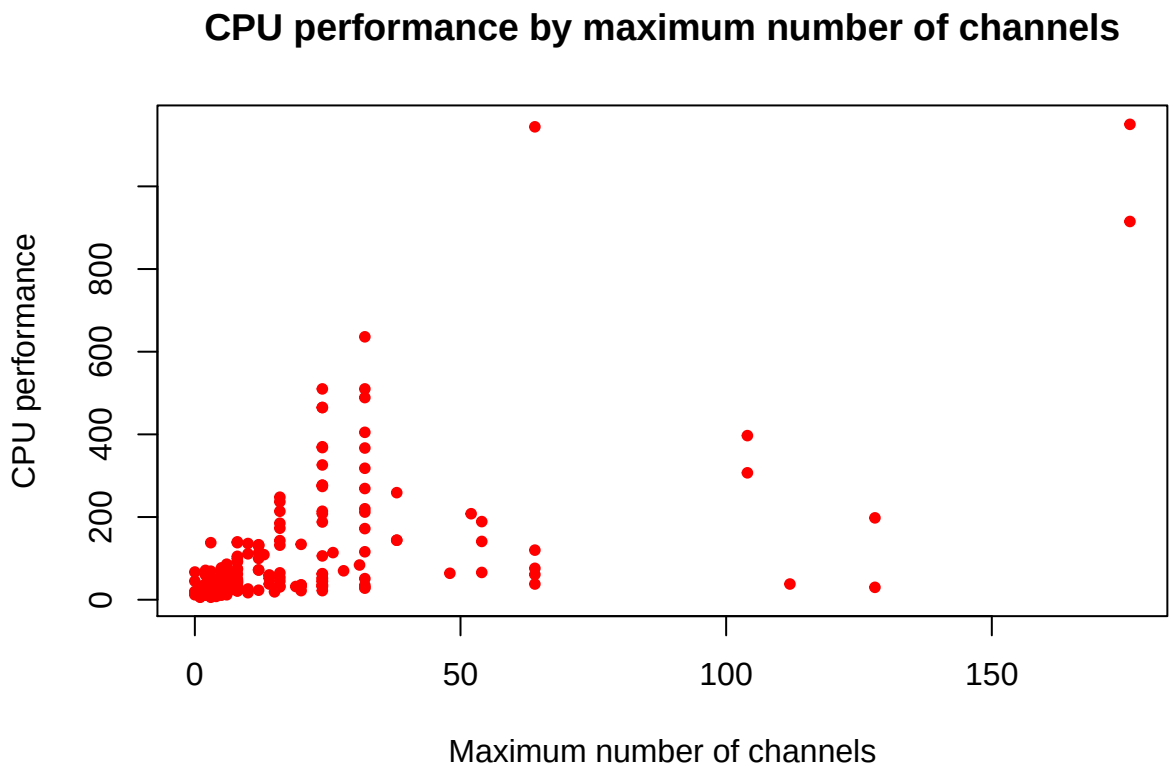
β_1 is statistically significantly positive because the p-value of this test is 0 (e.g. a very small number, which R probably rounded to zero), smaller than 0.01. The lower confidence bound value of β_1 with 0.01 alpha is 2.85, still a positive value.

Question B3: Checking the Assumptions of the Model - 8 pts

Create and interpret the following graphs with respect to the assumptions of the linear regression model. In other words, comment on whether there are any apparent departures from the assumptions of the linear regression model. Make sure that you state the model assumptions and assess each one. Each graph may be used to assess one or more model assumptions.

a. **2 pts** Scatterplot of the data with *chmax* on the x-axis and *performance* on the y-axis

```
# Your code here...
plot(chmax,performance,
     xlab="Maximum number of channels",
     ylab="CPU performance",
     main="CPU performance by maximum number of channels",
     col = "red",
     pch=20)
```

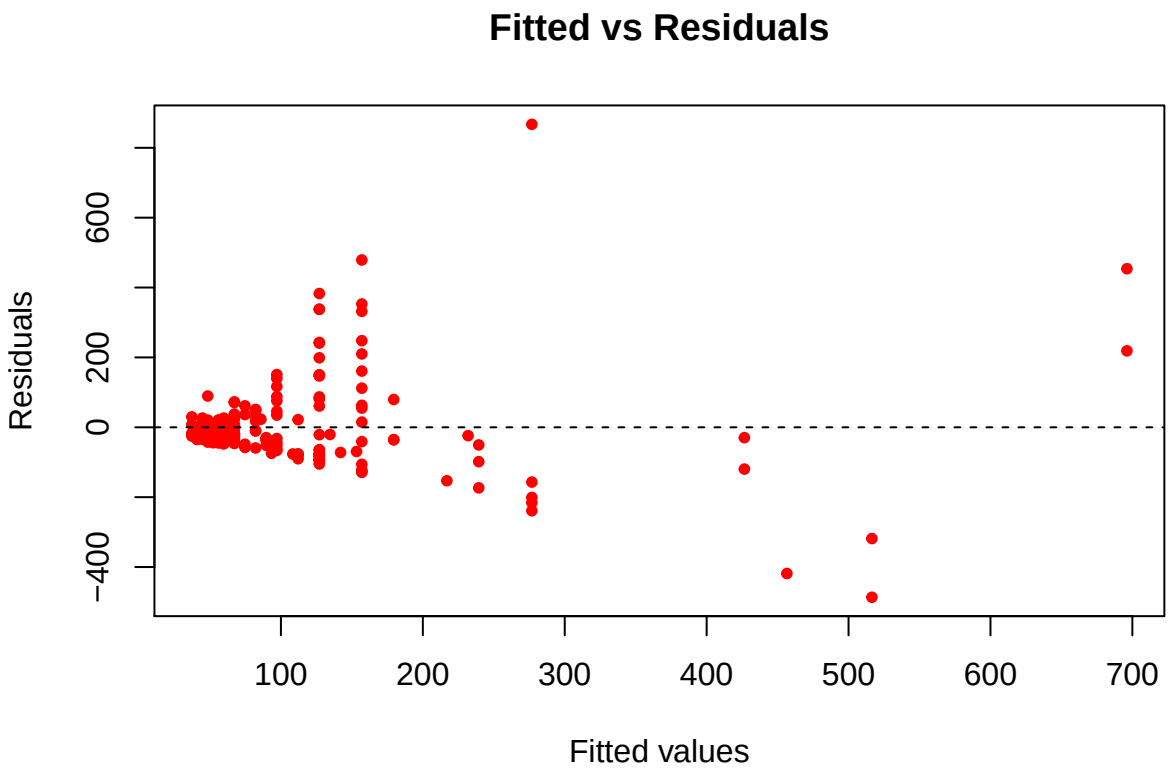


Model Assumption(s) it checks: Linearity

Interpretation: Here we can see that the data is linearly correlated, so this assumption holds. However, the data is very noisy, and it is not clear if there would be a non-linearity (for example, in large values of number of channels). Transforming the data would help making the linearity assumption stronger, as mentioned in an answer above.

b. **3 pts** Residual plot - a plot of the residuals, $\hat{\epsilon}_i$, versus the fitted values, $\hat{y}_i$

```
# Your code here...
plot(fitted(model1),residuals(model1),
     xlab="Fitted values", ylab="Residuals",
     main="Fitted vs Residuals",
     col = "red",
     pch=20)
abline(h=0,lty=2)
```

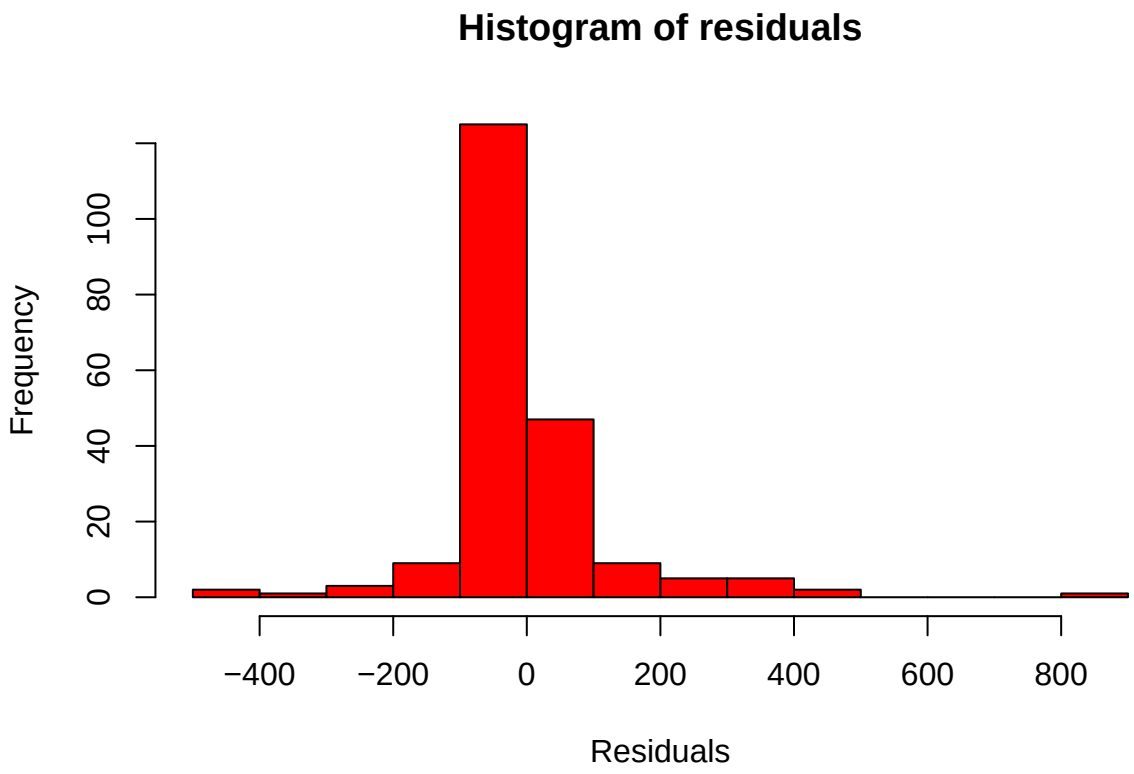


Model Assumption(s) it checks: Constant Variance and independence

Interpretation: The residuals show larger variance as the fitted values increase, showing a characteristic coned shape. The constant variance assumption does not hold. It is difficult to fully validate the independence assumption, as we don't know how the data was collected (e.g. how randomized the data collection was, and if there are any patterns in time or space) but there is no evidence to suggest this assumption is violated.

c. **3 pts** Histogram and q-q plot of the residuals

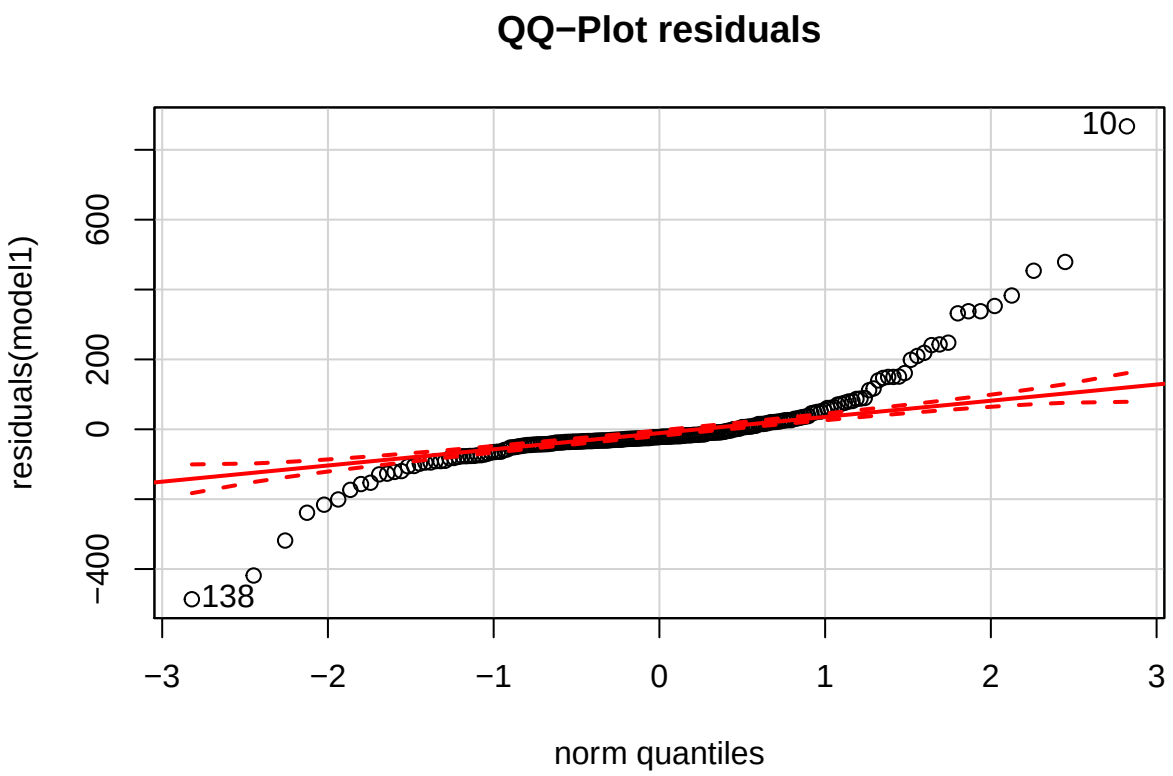
```
# Your code here...
hist(residuals(model1),main="Histogram of residuals",xlab="Residuals",col = "red", breaks=12)
```




```
library(car)
```

```
## Loading required package: carData
```

```
qqPlot(residuals(model1), main = "QQ-Plot residuals", col.lines = "red")
```




```
## [1]  10 138
```

Model Assumption(s) it checks: Normality

Interpretation: The histogram and QQ plots show the residuals are not perfectly normal, as the residuals distribution is a bit asymmetrical (the mean is smaller than zero) anda bit skewed (evidenced by the long tail on the right). However, from a practical perspective, we could assume normality as the distribution shape is not so far off from a normal distribution.

Question B4: Improving the Fit - 10 pts

a. **2 pts** Use a Box-Cox transformation (`boxCox()`) to find the optimal λ value rounded to the nearest half integer. What transformation of the response, if any, does it suggest to perform?

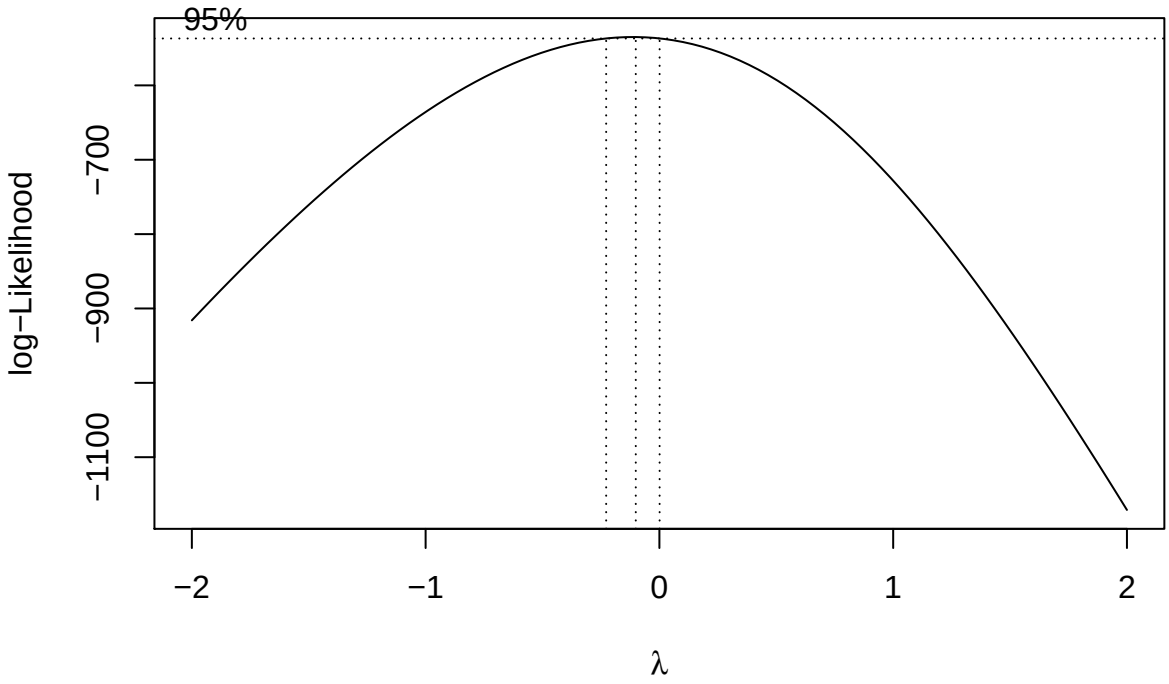
```
# Your code here...

#library(moments)
#skewness(model1$residuals)

library("MASS")
```

```
## Warning: package 'MASS' was built under R version 4.0.3
```

```
boxcox_model = boxcox(performance ~ chmax, data=data)
```




```
boxcox_model
```

```
## $x
##   [1] -2.00000000 -1.95959596 -1.91919192 -1.87878788 -1.83838384 -1.79797980
##   [7] -1.75757576 -1.71717172 -1.67676768 -1.63636364 -1.59595960 -1.55555556
##  [13] -1.51515152 -1.47474747 -1.43434343 -1.39393939 -1.35353535 -1.31313131
##  [19] -1.27272727 -1.23232323 -1.19191919 -1.15151515 -1.11111111 -1.07070707
##  [25] -1.03030303 -0.98989899 -0.94949495 -0.90909091 -0.86868687 -0.82828283
##  [31] -0.78787879 -0.74747475 -0.70707071 -0.66666667 -0.62626263 -0.58585859
##  [37] -0.54545455 -0.50505051 -0.46464646 -0.42424242 -0.38383838 -0.34343434
##  [43] -0.30303030 -0.26262626 -0.22222222 -0.18181818 -0.14141414 -0.10101010
##  [49] -0.06060606 -0.02020202  0.02020202  0.06060606  0.10101010  0.14141414
##  [55]  0.18181818  0.22222222  0.26262626  0.30303030  0.34343434  0.38383838
##  [61]  0.42424242  0.46464646  0.50505051  0.54545455  0.58585859  0.62626263
##  [67]  0.66666667  0.70707071  0.74747475  0.78787879  0.82828283  0.86868687
##  [73]  0.90909091  0.94949495  0.98989899  1.03030303  1.07070707  1.11111111
##  [79]  1.15151515  1.19191919  1.23232323  1.27272727  1.31313131  1.35353535
##  [85]  1.39393939  1.43434343  1.47474747  1.51515152  1.55555556  1.59595960
##  [91]  1.63636364  1.67676768  1.71717172  1.75757576  1.79797980  1.83838384
##  [97]  1.87878788  1.91919192  1.95959596  2.00000000
##
## $y
##   [1]  -915.9634  -902.5761  -889.3191  -876.1973  -863.2158  -850.3797
```

```
##   [7]  -837.6941  -825.1644  -812.7964  -800.5959  -788.5689  -776.7216
##  [13]  -765.0604  -753.5920  -742.3231  -731.2608  -720.4121  -709.7846
##  [19]  -699.3856  -689.2230  -679.3047  -669.6386  -660.2329  -651.0961
##  [25]  -642.2364  -633.6625  -625.3831  -617.4069  -609.7426  -602.3992
##  [31]  -595.3855  -588.7105  -582.3829  -576.4117  -570.8057  -565.5736
##  [37]  -560.7241  -556.2657  -552.2070  -548.5562  -545.3216  -542.5111
##  [43]  -540.1326  -538.1938  -536.7023  -535.6653  -535.0900  -534.9834
##  [49]  -535.3521  -536.2027  -537.5415  -539.3746  -541.7076  -544.5462
##  [55]  -547.8955  -551.7602  -556.1448  -561.0530  -566.4885  -572.4536
##  [61]  -578.9507  -585.9810  -593.5447  -601.6415  -610.2697  -619.4269
##  [67]  -629.1092  -639.3117  -650.0286  -661.2522  -672.9746  -685.1859
##  [73]  -697.8757  -711.0324  -724.6434  -738.6956  -753.1751  -768.0673
##  [79]  -783.3575  -799.0303  -815.0706  -831.4628  -848.1918  -865.2422
##  [85]  -882.5991  -900.2479  -918.1745  -936.3649  -954.8057  -973.4844
##  [91]  -992.3884 -1011.5061 -1030.8264 -1050.3385 -1070.0324 -1089.8983
##  [97] -1109.9276 -1130.1121 -1150.4432 -1170.9124
```

```
lambda = boxcox_model$x
lik = boxcox_model$y
combination = cbind(lambda,lik)
combination
```

```
##             lambda        lik
##   [1,] -2.00000000  -915.9634
##   [2,] -1.95959596  -902.5761
##   [3,] -1.91919192  -889.3191
##   [4,] -1.87878788  -876.1973
##   [5,] -1.83838384  -863.2158
##   [6,] -1.79797980  -850.3797
##   [7,] -1.75757576  -837.6941
##   [8,] -1.71717172  -825.1644
##   [9,] -1.67676768  -812.7964
##  [10,] -1.63636364  -800.5959
##  [11,] -1.59595960  -788.5689
##  [12,] -1.55555556  -776.7216
##  [13,] -1.51515152  -765.0604
##  [14,] -1.47474747  -753.5920
##  [15,] -1.43434343  -742.3231
##  [16,] -1.39393939  -731.2608
##  [17,] -1.35353535  -720.4121
##  [18,] -1.31313131  -709.7846
##  [19,] -1.27272727  -699.3856
##  [20,] -1.23232323  -689.2230
##  [21,] -1.19191919  -679.3047
##  [22,] -1.15151515  -669.6386
##  [23,] -1.11111111  -660.2329
##  [24,] -1.07070707  -651.0961
##  [25,] -1.03030303  -642.2364
##  [26,] -0.98989899  -633.6625
##  [27,] -0.94949495  -625.3831
##  [28,] -0.90909091  -617.4069
##  [29,] -0.86868687  -609.7426
##  [30,] -0.82828283  -602.3992
##  [31,] -0.78787879  -595.3855
```

```
##  [32,] -0.74747475  -588.7105
##  [33,] -0.70707071  -582.3829
##  [34,] -0.66666667  -576.4117
##  [35,] -0.62626263  -570.8057
##  [36,] -0.58585859  -565.5736
##  [37,] -0.54545455  -560.7241
##  [38,] -0.50505051  -556.2657
##  [39,] -0.46464646  -552.2070
##  [40,] -0.42424242  -548.5562
##  [41,] -0.38383838  -545.3216
##  [42,] -0.34343434  -542.5111
##  [43,] -0.30303030  -540.1326
##  [44,] -0.26262626  -538.1938
##  [45,] -0.22222222  -536.7023
##  [46,] -0.18181818  -535.6653
##  [47,] -0.14141414  -535.0900
##  [48,] -0.10101010  -534.9834
##  [49,] -0.06060606  -535.3521
##  [50,] -0.02020202  -536.2027
##  [51,]  0.02020202  -537.5415
##  [52,]  0.06060606  -539.3746
##  [53,]  0.10101010  -541.7076
##  [54,]  0.14141414  -544.5462
##  [55,]  0.18181818  -547.8955
##  [56,]  0.22222222  -551.7602
##  [57,]  0.26262626  -556.1448
##  [58,]  0.30303030  -561.0530
##  [59,]  0.34343434  -566.4885
##  [60,]  0.38383838  -572.4536
##  [61,]  0.42424242  -578.9507
##  [62,]  0.46464646  -585.9810
##  [63,]  0.50505051  -593.5447
##  [64,]  0.54545455  -601.6415
##  [65,]  0.58585859  -610.2697
##  [66,]  0.62626263  -619.4269
##  [67,]  0.66666667  -629.1092
##  [68,]  0.70707071  -639.3117
##  [69,]  0.74747475  -650.0286
##  [70,]  0.78787879  -661.2522
##  [71,]  0.82828283  -672.9746
##  [72,]  0.86868687  -685.1859
##  [73,]  0.90909091  -697.8757
##  [74,]  0.94949495  -711.0324
##  [75,]  0.98989899  -724.6434
##  [76,]  1.03030303  -738.6956
##  [77,]  1.07070707  -753.1751
##  [78,]  1.11111111  -768.0673
##  [79,]  1.15151515  -783.3575
##  [80,]  1.19191919  -799.0303
##  [81,]  1.23232323  -815.0706
##  [82,]  1.27272727  -831.4628
##  [83,]  1.31313131  -848.1918
##  [84,]  1.35353535  -865.2422
##  [85,]  1.39393939  -882.5991
```

```
##  [86,]  1.43434343  -900.2479
##  [87,]  1.47474747  -918.1745
##  [88,]  1.51515152  -936.3649
##  [89,]  1.55555556  -954.8057
##  [90,]  1.59595960  -973.4844
##  [91,]  1.63636364  -992.3884
##  [92,]  1.67676768 -1011.5061
##  [93,]  1.71717172 -1030.8264
##  [94,]  1.75757576 -1050.3385
##  [95,]  1.79797980 -1070.0324
##  [96,]  1.83838384 -1089.8983
##  [97,]  1.87878788 -1109.9276
##  [98,]  1.91919192 -1130.1121
##  [99,]  1.95959596 -1150.4432
## [100,]  2.00000000 -1170.9124
```

```
combination[order(-lik),]
```

```
##             lambda        lik
##   [1,] -0.10101010  -534.9834
##   [2,] -0.14141414  -535.0900
##   [3,] -0.06060606  -535.3521
##   [4,] -0.18181818  -535.6653
##   [5,] -0.02020202  -536.2027
##   [6,] -0.22222222  -536.7023
##   [7,]  0.02020202  -537.5415
##   [8,] -0.26262626  -538.1938
##   [9,]  0.06060606  -539.3746
##  [10,] -0.30303030  -540.1326
##  [11,]  0.10101010  -541.7076
##  [12,] -0.34343434  -542.5111
##  [13,]  0.14141414  -544.5462
##  [14,] -0.38383838  -545.3216
##  [15,]  0.18181818  -547.8955
##  [16,] -0.42424242  -548.5562
##  [17,]  0.22222222  -551.7602
##  [18,] -0.46464646  -552.2070
##  [19,]  0.26262626  -556.1448
##  [20,] -0.50505051  -556.2657
##  [21,] -0.54545455  -560.7241
##  [22,]  0.30303030  -561.0530
##  [23,] -0.58585859  -565.5736
##  [24,]  0.34343434  -566.4885
##  [25,] -0.62626263  -570.8057
##  [26,]  0.38383838  -572.4536
##  [27,] -0.66666667  -576.4117
##  [28,]  0.42424242  -578.9507
##  [29,] -0.70707071  -582.3829
##  [30,]  0.46464646  -585.9810
##  [31,] -0.74747475  -588.7105
##  [32,]  0.50505051  -593.5447
##  [33,] -0.78787879  -595.3855
##  [34,]  0.54545455  -601.6415
##  [35,] -0.82828283  -602.3992
```

```
##  [36,] -0.86868687  -609.7426
##  [37,]  0.58585859  -610.2697
##  [38,] -0.90909091  -617.4069
##  [39,]  0.62626263  -619.4269
##  [40,] -0.94949495  -625.3831
##  [41,]  0.66666667  -629.1092
##  [42,] -0.98989899  -633.6625
##  [43,]  0.70707071  -639.3117
##  [44,] -1.03030303  -642.2364
##  [45,]  0.74747475  -650.0286
##  [46,] -1.07070707  -651.0961
##  [47,] -1.11111111  -660.2329
##  [48,]  0.78787879  -661.2522
##  [49,] -1.15151515  -669.6386
##  [50,]  0.82828283  -672.9746
##  [51,] -1.19191919  -679.3047
##  [52,]  0.86868687  -685.1859
##  [53,] -1.23232323  -689.2230
##  [54,]  0.90909091  -697.8757
##  [55,] -1.27272727  -699.3856
##  [56,] -1.31313131  -709.7846
##  [57,]  0.94949495  -711.0324
##  [58,] -1.35353535  -720.4121
##  [59,]  0.98989899  -724.6434
##  [60,] -1.39393939  -731.2608
##  [61,]  1.03030303  -738.6956
##  [62,] -1.43434343  -742.3231
##  [63,]  1.07070707  -753.1751
##  [64,] -1.47474747  -753.5920
##  [65,] -1.51515152  -765.0604
##  [66,]  1.11111111  -768.0673
##  [67,] -1.55555556  -776.7216
##  [68,]  1.15151515  -783.3575
##  [69,] -1.59595960  -788.5689
##  [70,]  1.19191919  -799.0303
##  [71,] -1.63636364  -800.5959
##  [72,] -1.67676768  -812.7964
##  [73,]  1.23232323  -815.0706
##  [74,] -1.71717172  -825.1644
##  [75,]  1.27272727  -831.4628
##  [76,] -1.75757576  -837.6941
##  [77,]  1.31313131  -848.1918
##  [78,] -1.79797980  -850.3797
##  [79,] -1.83838384  -863.2158
##  [80,]  1.35353535  -865.2422
##  [81,] -1.87878788  -876.1973
##  [82,]  1.39393939  -882.5991
##  [83,] -1.91919192  -889.3191
##  [84,]  1.43434343  -900.2479
##  [85,] -1.95959596  -902.5761
##  [86,] -2.00000000  -915.9634
##  [87,]  1.47474747  -918.1745
##  [88,]  1.51515152  -936.3649
##  [89,]  1.55555556  -954.8057
```

```
##  [90,]  1.59595960  -973.4844
##  [91,]  1.63636364  -992.3884
##  [92,]  1.67676768 -1011.5061
##  [93,]  1.71717172 -1030.8264
##  [94,]  1.75757576 -1050.3385
##  [95,]  1.79797980 -1070.0324
##  [96,]  1.83838384 -1089.8983
##  [97,]  1.87878788 -1109.9276
##  [98,]  1.91919192 -1130.1121
##  [99,]  1.95959596 -1150.4432
## [100,]  2.00000000 -1170.9124
```

The optimal lambda value is -0.10101010 corresponding to a Log-likelihood of -534.9834, but since we will round it to the nearest half integer, we will say that lambda is 0. This suggest a transformation of the response from y to ln(y).

b. **2 pts** Create a linear regression model, named *model2*, that uses the log transformed *performance* as the response, and the log transformed *chmax* as the predictor. Note: The variable *chmax* has a couple of zero values which will cause problems when taking the natural log. Please add one to the predictor before taking the natural log of it

```
# Your code here...

model2 = lm(log(performance) ~ log(1+chmax), data)
summary(model2)
```

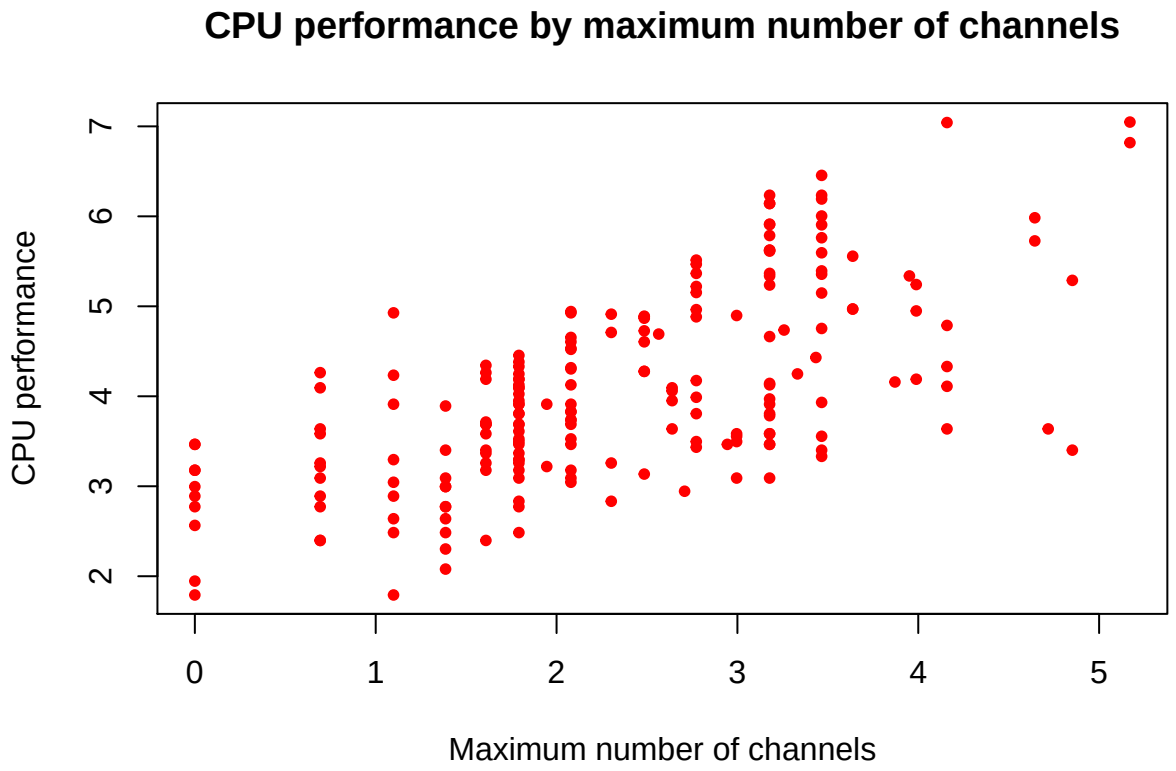
```
##
## Call:
## lm(formula = log(performance) ~ log(1 + chmax), data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.22543 -0.59429  0.01065  0.59287  1.85995
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2.47655    0.14152    17.5   <2e-16 ***
## log(1 + chmax)  0.64819    0.05401    12.0   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.807 on 207 degrees of freedom
## Multiple R-squared:  0.4103, Adjusted R-squared:  0.4074
## F-statistic:   144 on 1 and 207 DF,  p-value: < 2.2e-16
```

e. **2 pts** Compare the R-squared values of *model1* and *model2*. Did the transformation improve the explanatory power of the model?

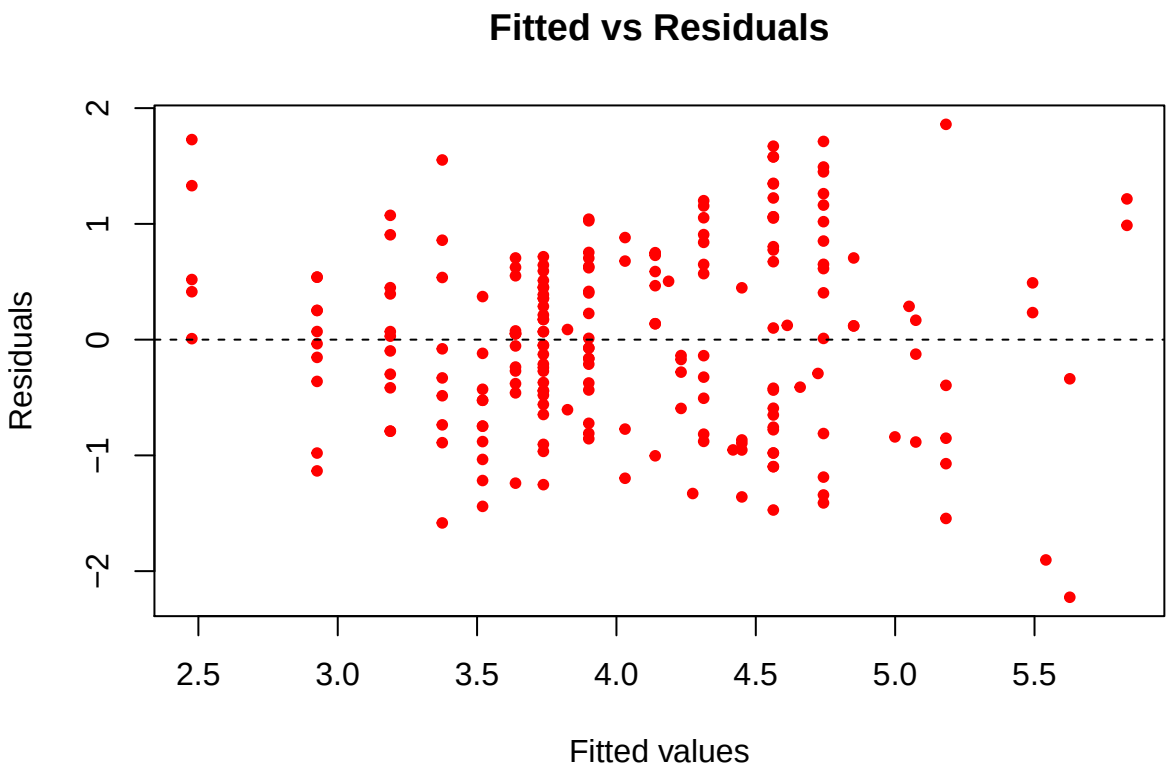
The R-squared value of *model1* is 0.3663,and the R-squared value of *model2* is 0.4103. We can see that the transformation improved the explanatory power by increasing the proportion of the total variability in Y that can be explained by the regression that uses X. For This case, we are referring to the variability of the CPU performance that can be explained by the maximum number of channels. One observation, it is risky to directly compare R-squared from models where the dependent variable has been transformed, as the errors are not in the same scale.

c. **4 pts** Similar to Question B3, assess and interpret all model assumptions of *model2*. A model is considered a good fit if all assumptions hold. Based on your interpretation of the model assumptions, is *model2* a good fit?

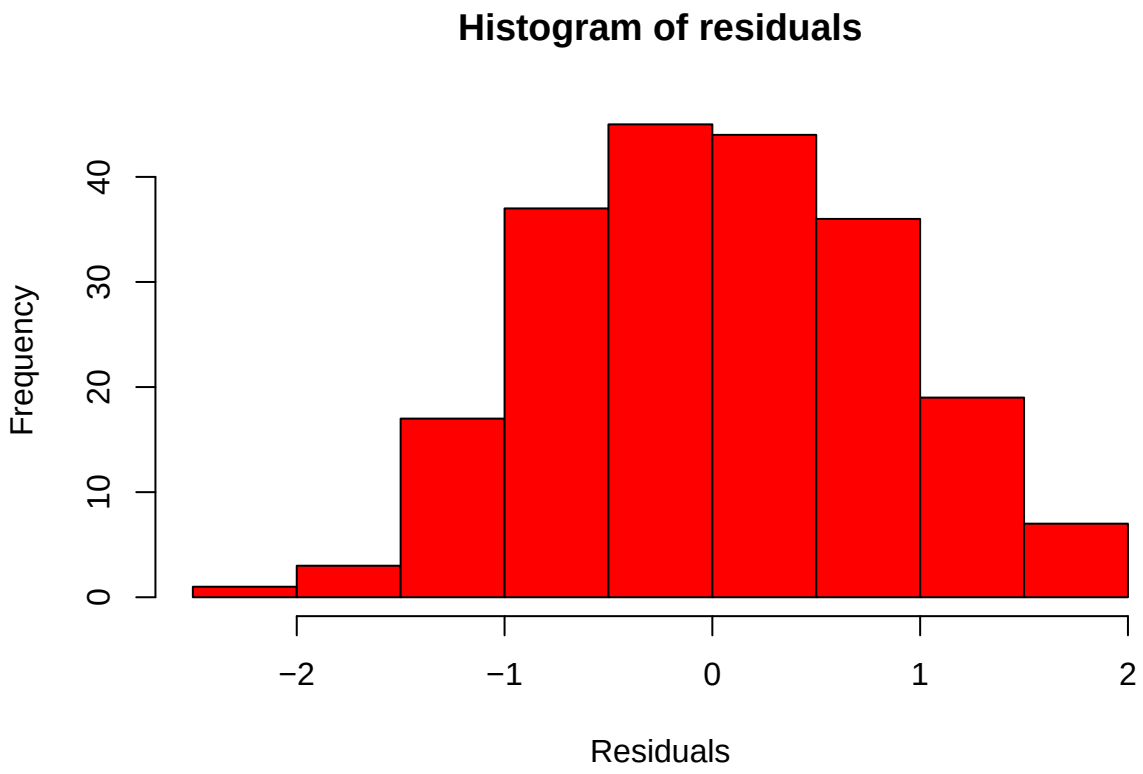
```
# Your code here...
plot(log(chmax),log(performance),
     xlab="Maximum number of channels",
     ylab="CPU performance",
     main="CPU performance by maximum number of channels",
     col = "red",
     pch=20)
```



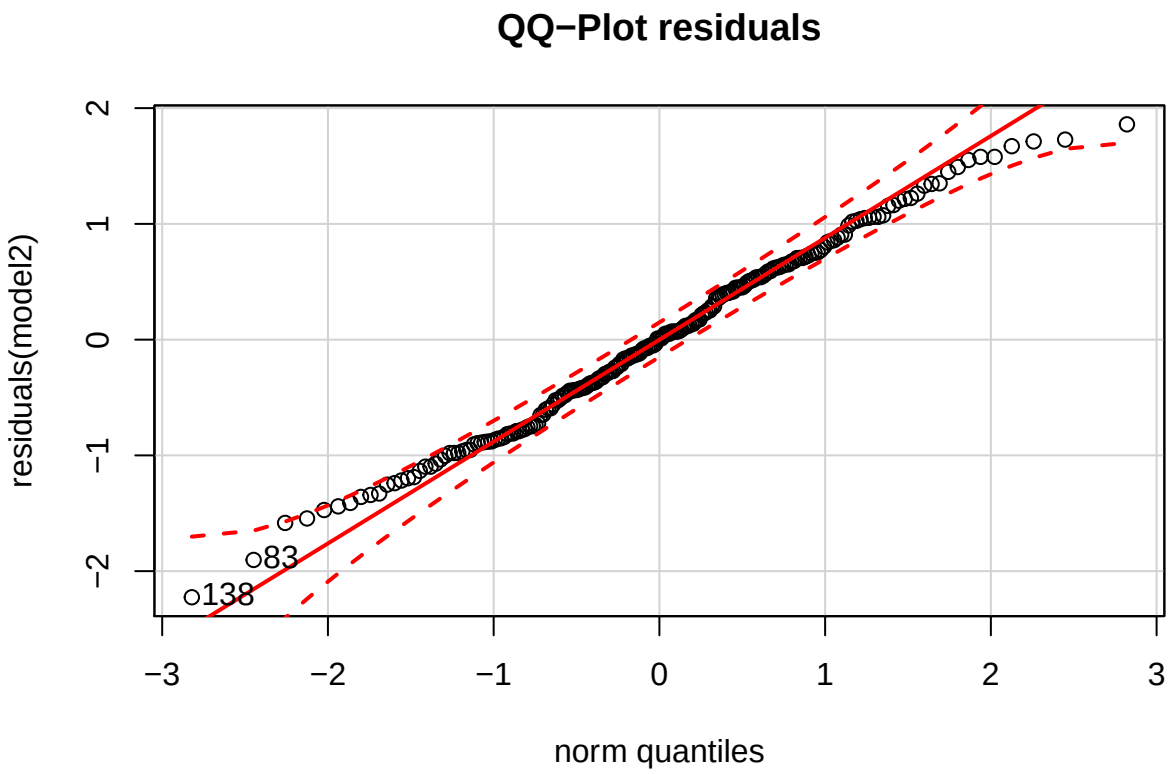

```
plot(fitted(model2),residuals(model2),
     xlab="Fitted values", ylab="Residuals",
     main="Fitted vs Residuals",
     col = "red",
     pch=20)
abline(h=0,lty=2)
```




```
hist(residuals(model2),main="Histogram of residuals",xlab="Residuals",col = "red")
```




```
library(car)
qqPlot(residuals(model2), main = "QQ-Plot residuals", col.lines = "red")
```




```
## [1] 138  83
```

Assumption Linearity

Interpretation Compared to the untransformed variables, now the linear relationship is more clear. The linearity assumption holds.

Assumption Constant variance

Interpretation Fitted values versus residuals present themselves as a cloud of points with no distinct form, and we do not see the cone shape anymore. This is a sign that the variance in constant. The assumption hold.

Assumption Independence

Interpretation As mentioned above for the untransformed model, it is difficult to assess independence without knowing how the data was collected, but the assuptiom appears to hold as there is no evidence of data dependency.

Assumption Normality

Interpretation The histogram and the QQplots indicate that the residuals follow a normal distribution. The normality assumptions hold.

Question B5: Prediction - 3 pts

Suppose we are interested in predicting CPU performance when `chmax = 128`. Please make a prediction using both *model1* and *model2* and provide the 95% prediction interval of each prediction on the original

scale of the response, *performance*. What observations can you make about the result in the context of the problem?

```
# Your code here...
new_data = data.frame(chmax = 128)

predict.lm(model1, new_data , interval = "predict", level = 0.95)
```

```
##        fit      lwr      upr
## 1 516.4685 252.2519 780.6851
```

```
predict.lm(model2, new_data, interval = "predict", level = 0.95)
```

```
##        fit      lwr      upr
## 1 5.626624 4.010584 7.242664
```

```
#Converting model2 results to e^(n)
exp(5.626624)
```

```
## [1] 277.7229
```

```
exp(4.010584)
```

```
## [1] 55.17909
```

```
exp(7.242664)
```

```
## [1] 1397.813
```

With model1, we predict 516.46 for CPU performance with a range between 252.25 - 780.68, and for model2 we predict CPU performance as 5.626624 with a range between 4.010584 and 7.242664.

However,we know that model2 was transformed using the natural logarithm, so we can use the exponential to calculate the actual numbers for the prediction and the range, which are: CPU performance 277.7229, with a range between 55.17909 and 1397.813

Part C. ANOVA - 8 pts

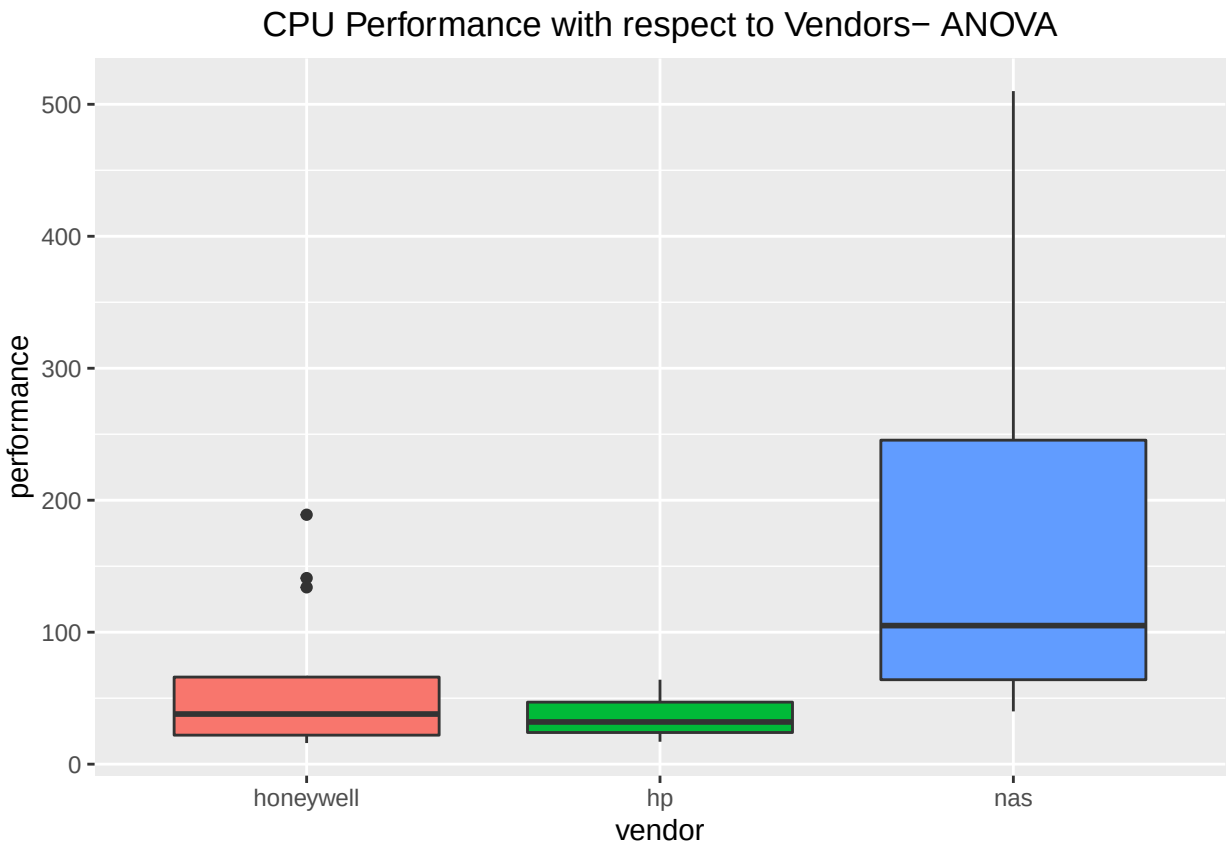
We are going to continue using the CPU data set to analyse various vendors in the data set. There are over 20 vendors in the data set. To simplify the task, we are going to limit our analysis to three vendors, specifically, honeywell, hp, and nas. The code to filter for those vendors is provided below.

```
# Filter for honeywell, hp, and nas
data2 = data[data$vendor %in% c("honeywell", "hp", "nas"), ]
data2$vendor = factor(data2$vendor)
```

1. **2 pts** Using `data2`, create a boxplot of *performance* and *vendor*, with *performance* on the vertical axis. Interpret the plots.

```
# Your code here...

library(ggplot2)
ggplot(data2, aes(x=vendor, y=performance, fill=vendor)) + geom_boxplot(alpha = 3) + theme(legend.posit
  ggtitle("CPU Performance with respect to Vendors- ANOVA") + theme(plot.title = element_text(hjust = 0
```



Just by looking at the plot, it seems that the sample means are different from each other, perhaps we could think that that the CPU's from **nas** exhibit a higher average performance than **hp** and **honeywell** equipment. We can also see that the **nas** data is more wide than the other 2 vendors data. One more observation is that the honeywell data has outliers.

In general, we can say that there is some within-variability because some groups have higher variability than others, such is the case of the vendor **nas**,and we can observe also that there is between-variability because there is variability on the means of the three groups.

2. **3 pts** Perform an ANOVA F-test on the means of the three vendors. Using an α-level of 0.05, can we reject the null hypothesis that the means of the three vendors are equal? Please interpret.

```
# Your code here...
anova_model = aov(performance ~ vendor, data=data2)
model.tables(anova_model, type = "means")
```

```
## Tables of means
## Grand mean
##
```

```
## 112.8718
##
##  vendor
##     honeywell    hp   nas
##         60.46 36.43 176.9
## rep     13.00  7.00  19.0
```

```
summary(anova_model)
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## vendor       2 154494   77247   6.027 0.00553 **
## Residuals   36 461443   12818
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on the analysis above, we can say with 95% confidence that the CPU performance means between the 3 vendors are different because the P-value is very small and is close to 0. Therefore, we reject the null hypothesis that the 3 means are equal.

3. **3 pts** Perform a Tukey pairwise comparison between the three vendors. Using an α-level of 0.05, which means are statistically significantly different from each other?

```
# Your code here...
TukeyHSD(anova_model)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = performance ~ vendor, data = data2)
##
## $vendor
##                     diff        lwr      upr     p adj
## hp-honeywell  -24.03297 -153.76761 105.7017 0.8934786
## nas-honeywell 116.43320   16.82659 216.0398 0.0188830
## nas-hp        140.46617   18.11095 262.8214 0.0214092
```

Based on the analysis above, we can say with 95% confidence that **nas and honeywell** as well as **nas and hp** have CPU performance means which are statistically significant different because the p-value smaller than 0.05 in both cases and their mean difference confidence intervals does not include 0 (there is always a difference).