Simple Linear Regression and ANOVA

```
#Check R version
R.version
```

```
##                _
## platform       x86_64-w64-mingw32
## arch           x86_64
## os             mingw32
## system         x86_64, mingw32
## status
## major          4
## minor          0.2
## year           2020
## month          06
## day            22
## svn rev        78730
## language       R
## version.string R version 4.0.2 (2020-06-22)
## nickname       Taking Off Again
```

```
library(ggplot2)
library(car)
```

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

```
library(MASS)
```

Part A. ANOVA

Additional Material: ANOVA tutorial

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

Calculate Manualy

```
# Data
control = c(0.53, 0.36, 0.20, -0.37, -0.60, -0.64, -0.68, -1.27)
knees = c(0.73, 0.31, 0.03, -0.29, -0.56, -0.96, -1.61)
eyes = c(-0.78, -0.86, -1.35, -1.48, -1.52, -2.04, -2.83)
SS_Error = 9.415
MS_Treatments = 3.6122

# Find Degrees of Freedom
k = 3
N = 22
df_treatments = k - 1
df_Error = N - k
df_total = N - 1

# Find SS
data = c(0.53, 0.36, 0.20, -0.37, -0.60, -0.64, -0.68, -1.27, 0.73, 0.31, 0.03, -0.29, -0.56, -
0.96, -1.61, -0.78, -0.86, -1.35, -1.48, -1.52, -2.04, -2.83)
grand_mean = mean(data)

control_mean = mean(control)
knees_mean = mean(knees)
eyes_mean = mean(eyes)
means = c(control_mean, knees_mean, eyes_mean)

n = c(8, 7, 7)
SSTR = 0
for (i in c(1,2,3)) {
  SSTR = SSTR + (n[i] * ((means[i] - grand_mean)^2))
  }

MS_Error = SS_Error / df_Error


print(paste("Treatment Df: ", df_treatments))
```

```
## [1] "Treatment Df:  2"
```

```
print(paste("Error Df: ", df_Error))
```

```
## [1] "Error Df:  19"
```

```
print(paste("df_total: ", df_total))
```

```
## [1] "df_total:  21"
```

```
print(paste("Treatment SS: ", SSTR))
```

```
## [1] "Treatment SS:  7.22449172077922"
```

```
print(paste("Total SS: ", SSTR + SS_Error))
```

```
## [1] "Total SS:  16.6394917207792"
```

```
print(paste("Treatments MS: ", SSTR / df_treatments))
```

```
## [1] "Treatments MS:  3.61224586038961"
```

```
print(paste("Error MS: ", MS_Error))
```

```
## [1] "Error MS:  0.495526315789474"
```

```
print(paste("F Stat: ", MS_Treatments / MS_Error))
```

```
## [1] "F Stat:  7.28962294211365"
```

Calculate using the R's aov Function

```
treatment = rep(c("control", "knees", "eyes"), times=c(length(control), length(knees), length(ey
es)))
jet = data.frame(Y= c(control, knees, eyes), treatment =factor(treatment))
jet_anova = aov(Y ~ treatment, data = jet)
summary(jet_anova)
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## treatment    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
```

Answer

Source	Df	Sum of Squares	Mean Squares	F-statistics	p-value
Treatments	2	7.224492	3.6122	7.289623	0.004
Error	19	9.415	0.496		
TOTAL	21	16.639492			

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.

```
mu_1 = control_mean
mu_2 = knees_mean
mu_3 = eyes_mean

print(paste("mu_1 = ", control_mean))
```

```
## [1] "mu_1 =  -0.30875"
```

```
print(paste("mu_2 = ", knees_mean))
```

```
## [1] "mu_2 =  -0.335714285714286"
```

```
print(paste("mu_3 = ", eyes_mean))
```

```
## [1] "mu_3 =  -1.55142857142857"
```

Answer

μ_1 is the average time (in hours) of the control group's, where no treatment was applied, phase shift in the subject's daily cycle of melatonin production. μ_2 is the phase shift in the subject's daily cycle of melatonin production average time (in hours) of the group which was exposed to bright lights applied to the knees only. μ_3 is the phase shift in the subject's daily cycle of melatonin production average time (in hours) of the group which was exposed to bright lights applied to the eyes only.

μ_1 = -0.30875

μ_2 = -0.335714

μ_3 = -1.551429

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

H_0 : $\mu_1 = \mu_2 = \mu_3$ (all the means are equal)

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

Answer

H_A : Not all $\mu's$ are equal or, at least one is different from the others.

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

```
pf = pf(MS_Treatments / MS_Error, df1=df_treatments, df2=df_Error, lower.tail = FALSE)
print(pf)
```

```
## [1] 0.00447183
```

The p-value is 0.004 as indicated in the above code and in the table

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.

Answer

According to the results of the ANOVA F-test, we can REJECT the Null hypothesis (with α = 0.05) stating the means of the three treatments are equals. This means that at least one mean is different from the others, but not necessarily that the Control group is different than the treated groups (it could be that the difference is between the two treated groups and the Control's mean is equal to the mean of one of them). We would need to perform the Tukey's HSD test to test whether the Control's group mean is different from the means of the treated groups in order to determine if the light treatment affect phase shift.

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("machine.csv", head = TRUE, sep = ",")
# Show the first few rows of data
head(data, 3)
```

	vendor <chr>	chmax <int>	performance <int>
1	adviser	128	198
2	amdahl	32	269

	vendor <chr>	chmax <int>	performance <int>
3	amdahl	32	220

3 rows

Question B1: Exploratory Data Analysis - 9 pts

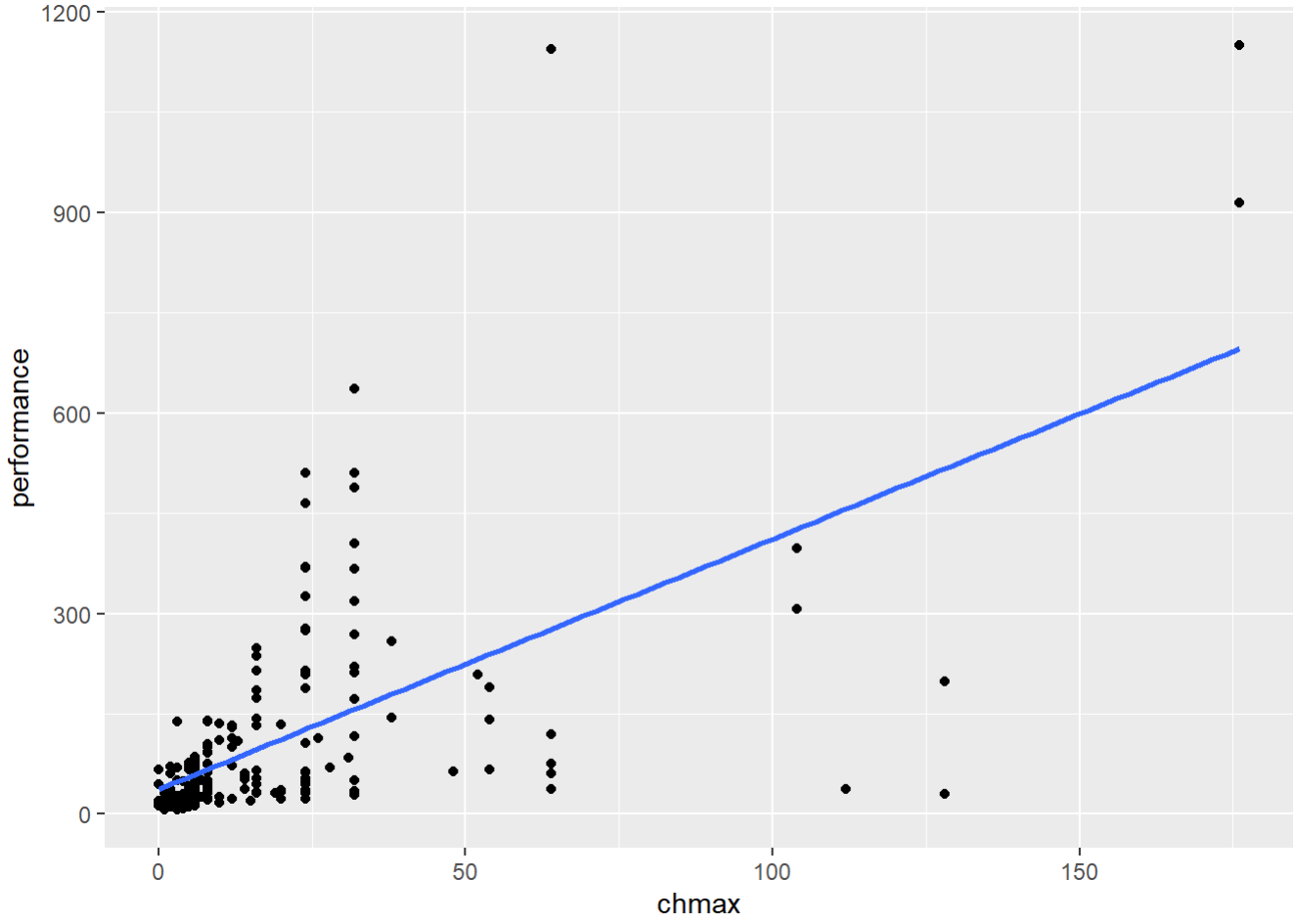
a. **3 pts** Use a scatter plot to describe the relationship between CPU performance and the maximum number of channels. Describe the general trend (direction and form). Include plots and R-code used.

Answer

We can see below a direct relationship between chmax and performance, meaning that as the maximum number of channels increases, performance increases as well. There is some form of linearity, but the goodness of fit should still be investigated. Moreover, it seems like there potentially few outliers which will have to be investigated.

```
# Your code here...
ggplot(data, aes(x=chmax, y=performance)) + geom_point() +
  scale_colour_hue(l=50) + # Use a slightly darker palette than normal
  geom_smooth(method="lm",   # Add linear regression line
              se=FALSE,    # Don't add shaded confidence region
              fullrange=TRUE) # Extend regression lines
```

```
## `geom_smooth()` using formula 'y ~ x'
```



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.

Answer

We can see below that the correlation coefficient is ***0.6052***. The correlation coefficient can be used to measure how strong the relationship between two variables (in other words, how close the response variables to the regression line). Ranging between -1 and 1, we can establish that there is some (but not very strong) positive relationship between performance and chmax, meaning that for every *positive* increase in chmax, there is a *positive* increase in performance.

```
# Your code here...
r = cor(data$chmax, data$performance)
print(paste("The correlation coefficient between performance and chmax is: ", r))
```

```
## [1] "The correlation coefficient between performance and chmax is:  0.605209292812674"
```

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

Answer

Yes, based on the exploratory analysis above, I would recommend to perform a simple linear regression. However, residuals investigation should be performed to make sure assumptions hold and to check if transformation could help address any violation in assumptions.

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

Answer

Yes, it seems like there are few outliers that might affect the linear regression model. We can also see that the variance of the performance variable increases as chmax increases, suggesting that there might be a violation of the constant variance assumption. Hence, I would pursue a transformation of the data to address the heteroscedasticity issue.

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
```

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

Answer

The model parameters and their estimates are: $\hat{\beta}_0$ (intercept) = 37.2252 $\hat{\beta}_1$ (slope) = 3.7441 $\hat{\sigma^2}$ (variance of the error terms) = 128.3^2 = 16,460.89

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

Answer

performance = 37.2252 + 3.7441*chmax

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

Answer

We interpret $\hat{\beta}_1$ parameter such that as maximum number of channels increases by 1 unit, performance increases by 3.7441 units.

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

Answer

We can see below that at the 95% level, the confidence interval for the β_1 parameter is (3.069251, 4.418926). As we see in the model1 output above, β_1 is statistically significant at this level, because 2e-16 < 0.05. In addition, the 95% confidence interval below does not contain the value zero which leads us to reject the null hypothesis β_1 = 0.

```
b_CI = confint(model1, level=0.95)
print("95% confidence interval of b1 parameter is:")
```

```
## [1] "95% confidence interval of b1 parameter is:"
```

```
print(b_CI[2,])
```

```
##    2.5 %   97.5 %
## 3.069251 4.418926
```

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?

Answer

β_1 is statistically significantly positive at an α-level of 0.01. See below that the confidence interval at this level does not contain 0 and the interval comprised of positive values. Also, the approximate p-value of this test is ~ 1.424772*e-22*. The p-value is smaller than 0.01 and hence we REJECT the null hypothesis that $\beta_1 \leq 0$, meaning β_1 is indeed statistically significantly positive.

```
b_CI = confint(model1, level=0.99)
print("99% confidence interval of b1 parameter is:")
```

```
## [1] "99% confidence interval of b1 parameter is:"
```

```
print(b_CI[2,])
```

```
##    0.5 %   99.5 %
## 2.854185 4.633991
```

```
tv = 10.938
df = 207
# H0: b_1 =< 0; H1: b_1 > 0
p = pt(tv, df, lower.tail = FALSE)
print(paste("p_value is approximately: ", p))
```

```
## [1] "p_value is approximately:  1.42477173579908e-22"
```

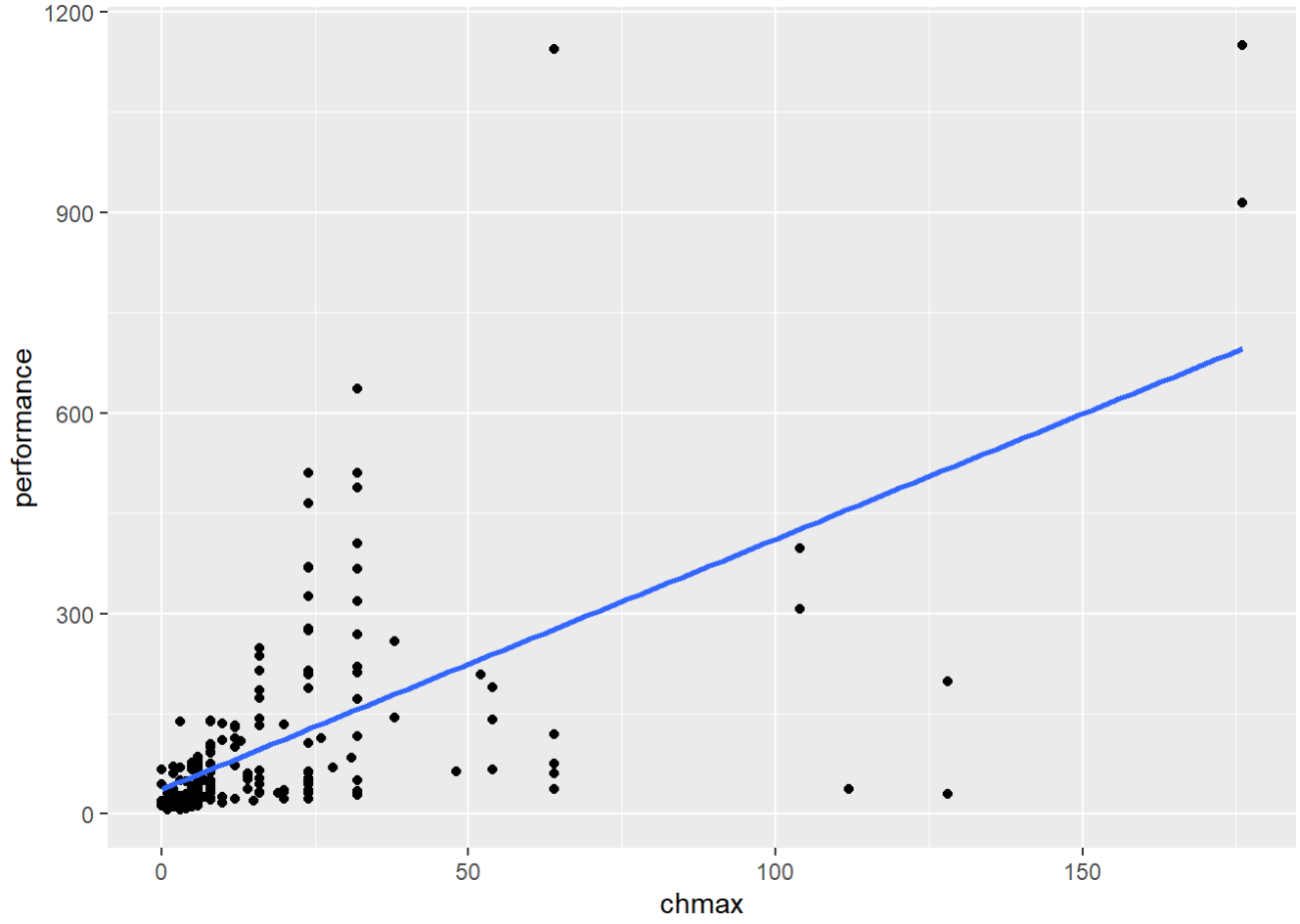
Question B3: Checking the Assumptions of the Model - 8 pts

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...
ggplot(data, aes(x=chmax, y=performance)) + geom_point() +
  scale_colour_hue(l=50) + # Use a slightly darker palette than normal
  geom_smooth(method="lm",   # Add linear regression line
              se=FALSE,    # Don't add shaded confidence region
              fullrange=TRUE) # Extend regression lines
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Model Assumption(s) it checks:

Linearity

Constant variance

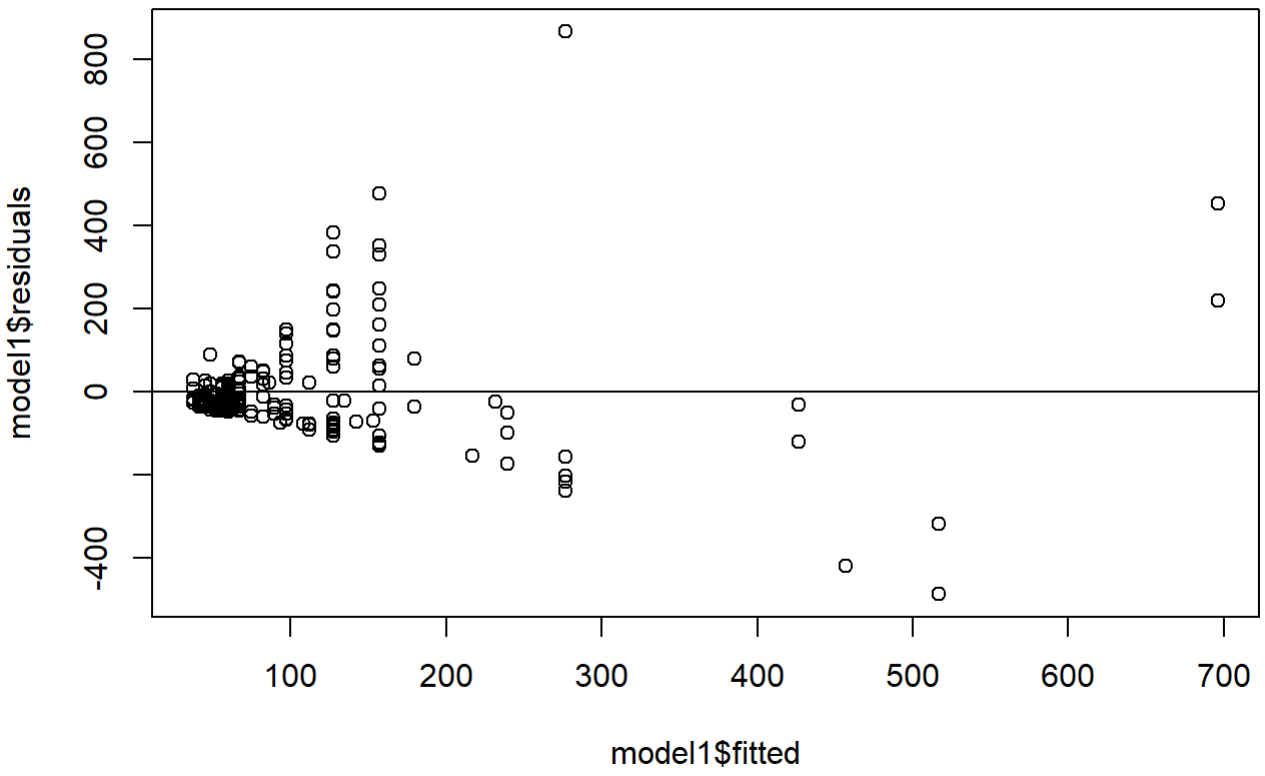
Interpretation:

Linearity - Linearity between the chmax and performance does exist (to some extent). However, it seems like there are few outliers that might affect the linear regression model.

Constant variance - It seems like the constant variance might not hold, because the variability in the performance variable increases as chmax increases.

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

```
# Your code here...
plot(model1$fitted, model1$residuals)
abline(0,0)
```



Model Assumption(s) it checks:

Linearity

Constant variance

Uncorrelated errors (independence assumption)

Interpretation:

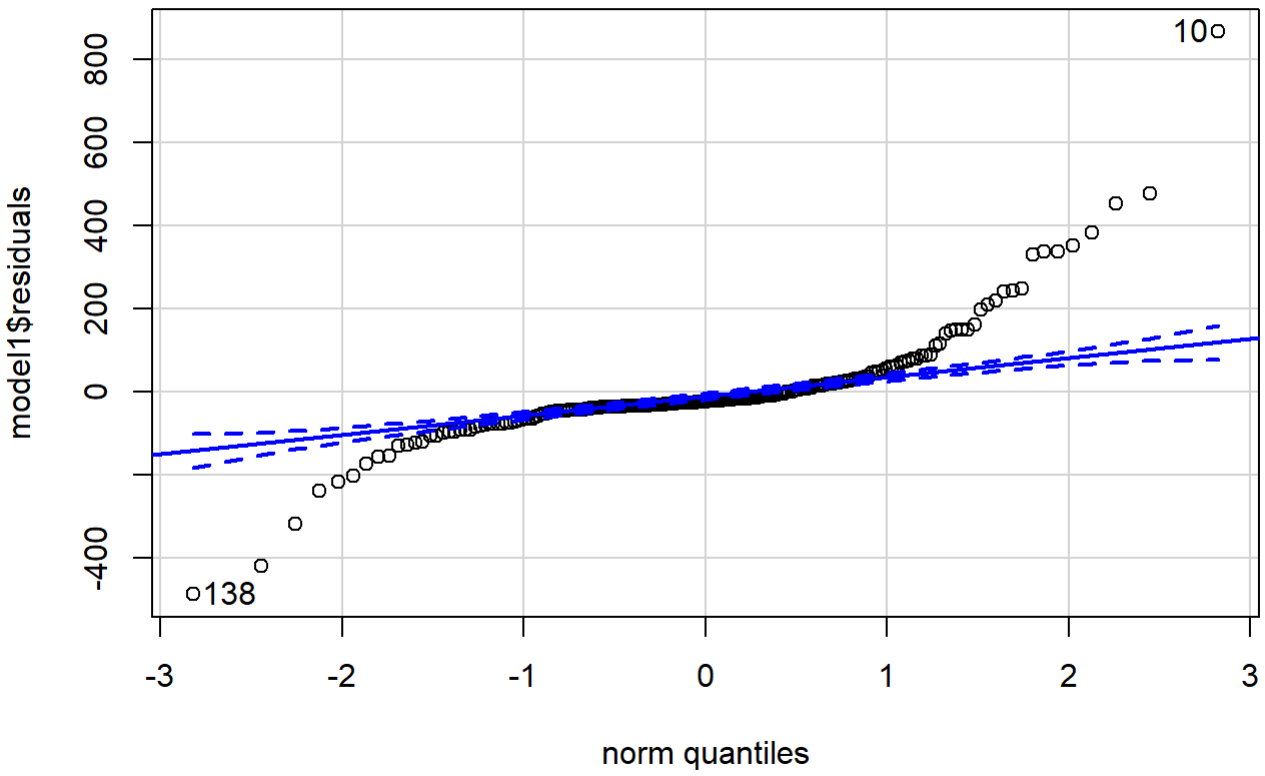
Linearity - there is no pattern in the plot so we can conclude that the linearity holds.

Constant variance - constant variance assumption does not hold because the residuals increase as the fitted values increase.

Uncorrelated errors (independence assumption) - we do not see a grouping of the residuals, meaning that the assumption of uncorrelated error holds.

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

```
# Your code here...
qqPlot(model1$residuals)
```




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

```
hist(model1$residuals)
```

Model Assumption(s) it checks:

Normality

Interpretation:

The Q-Q plot indicated as heavy-tailed. Histogram should have an approximately symmetric distribution with no gaps, which is not presented in our hist plot. Hence, both the Q-Q plot and histogram suggest the normality assumptions does not hold.

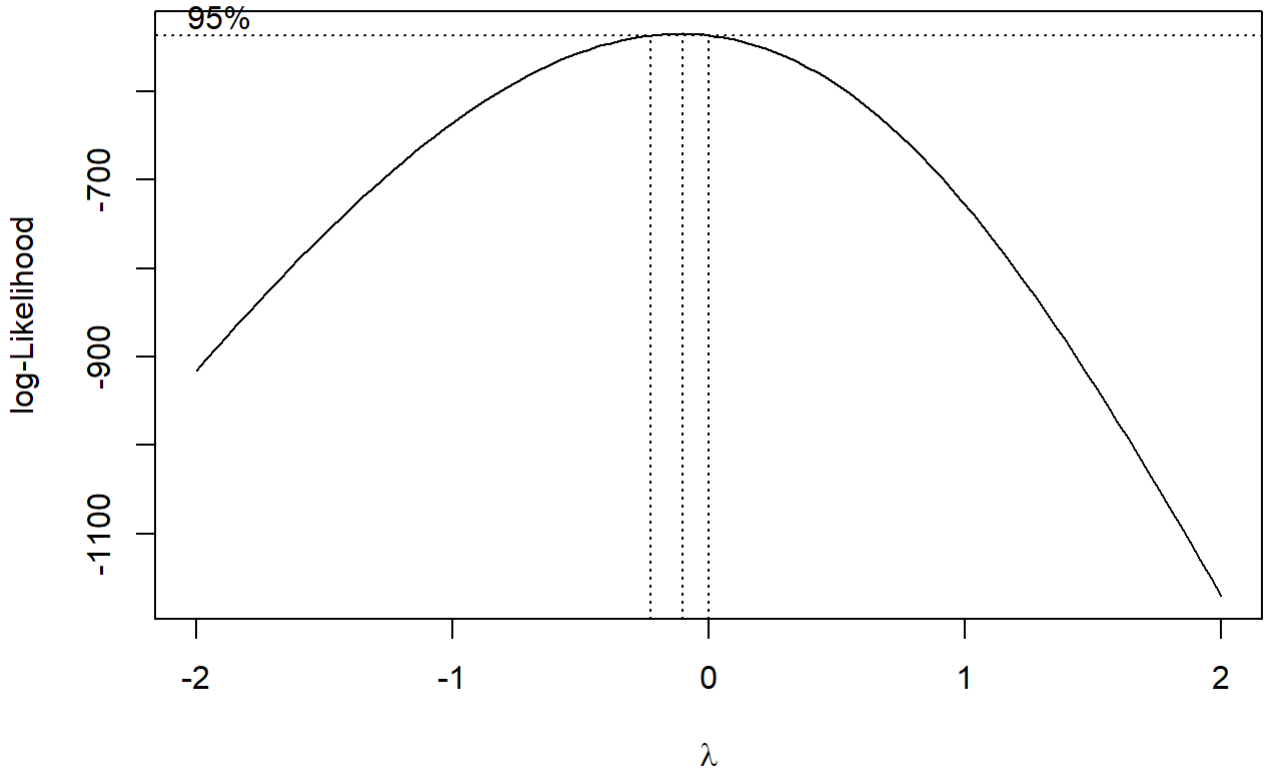
Question B4: Improving the Fit - 10 pts

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?

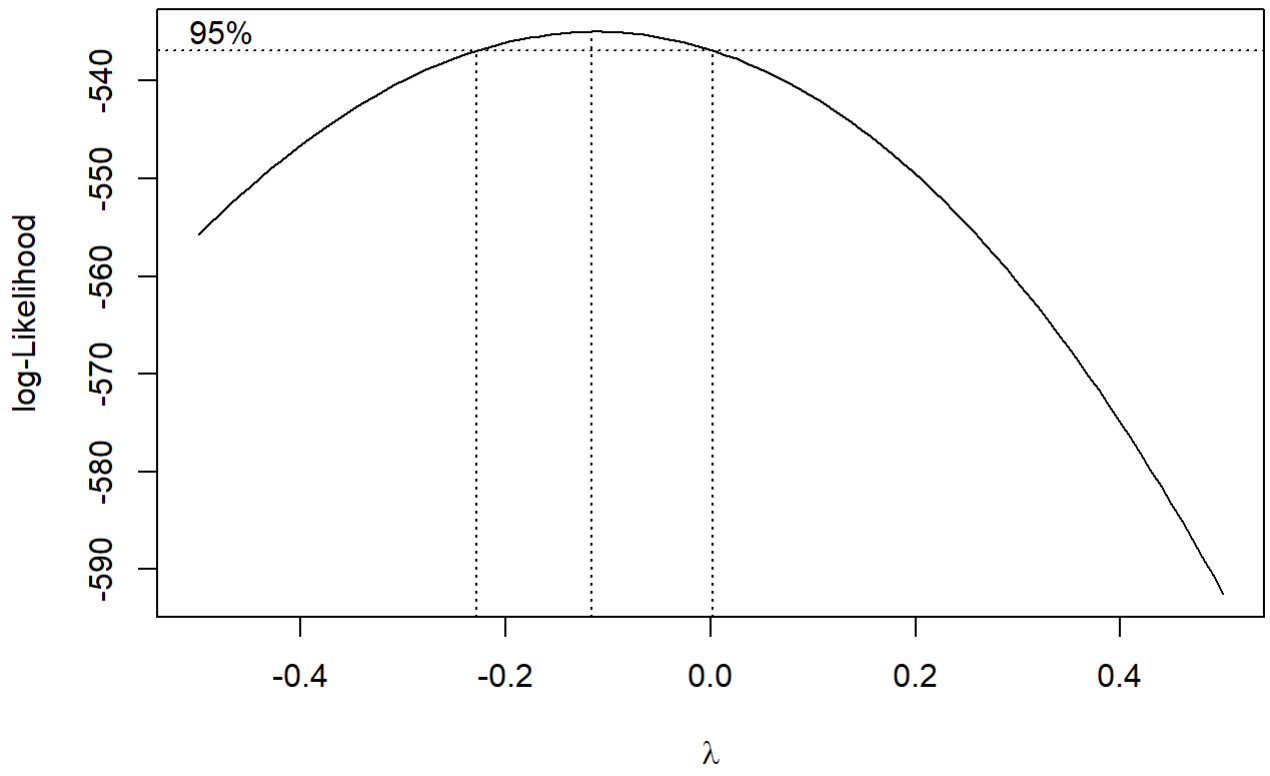
Answer

See below the optimal λ value is -0.1161 and the optimal λ rounded to the nearest half integer is 0, which means to use the normal logarithmic transformation i.e. $log(y)$.

```
# Your code here...
x = data$chmax
y = data$performance
bc = boxcox(y~x, plotit = TRUE)
```




```
bc = boxcox(y~x, plotit = TRUE, lambda = seq(-0.5, 0.5, by = 0.1))
```




```
lambda = bc$x[which.max(bc$y)]
print(paste("Optimal lambda: ", lambda))
```

```
## [1] "Optimal lambda:  -0.116161616161616"
```

```
print(paste("Optimal lambda rounded to the nearest half integer: ", round(2*lambda)/2))
```

```
## [1] "Optimal lambda rounded to the nearest half integer:  0"
```

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(chmax+1), data = data)
summary(model2)
```

```
##
## Call:
## lm(formula = log(performance) ~ log(chmax + 1), 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(chmax + 1)  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?

Answer

Model1's R^2 is 0.3663 and *model2's* R^2 is 0.4103 suggesting the transformation did improve the explanatory power of the model.

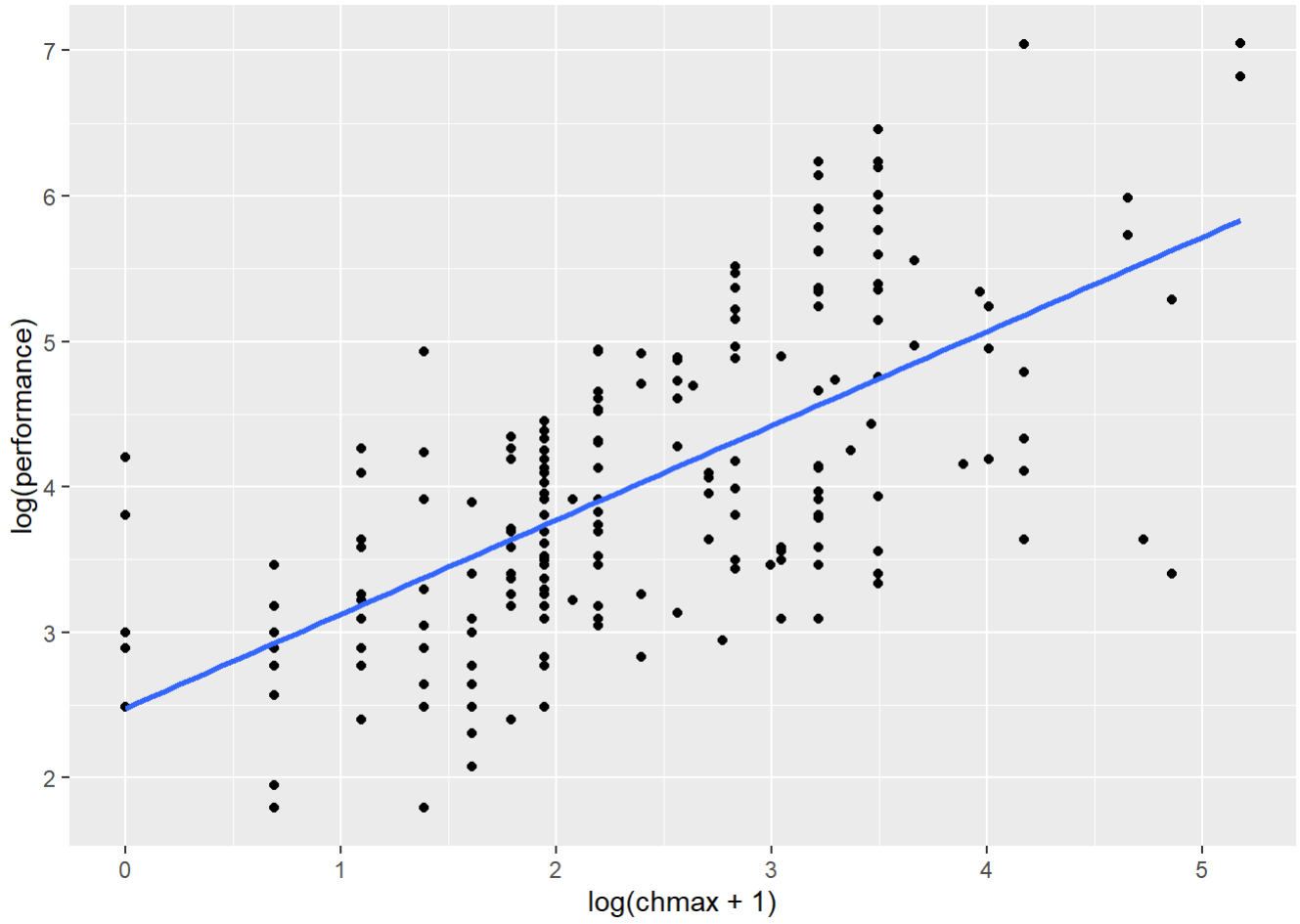
c. **4 pts** Similar to Question B3, assess and interpret all model assumptions of *model2*. A model is considered a good fit if all assumptions hold. Based on your interpretation of the model assumptions, is *model2* a good fit?

Answer

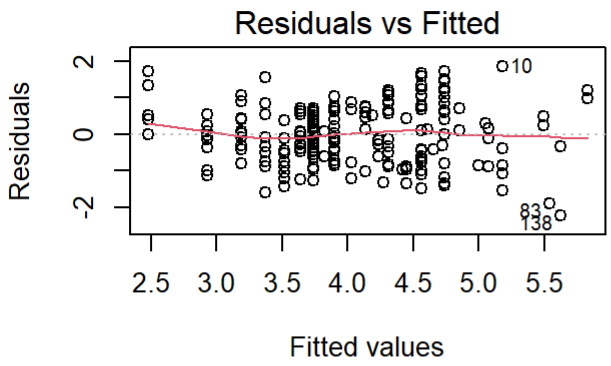
Observe below in the log(performance)-log(chmax+1) plot that we can establish there is linearity and constant variance assumption seems to hold as well. Also, we can determine based on the Residuals vs. Fitted values plot that the linearity, constant variance and uncorrelated errors assumptions hold because the residuals plot show no pattern, variance is constant across the fitted values, and there is no grouping. The Normal Q-Q and histogram plots also suggest normality. Hence, we can conclude that *model2* is considered a good fit because all of the assumptions hold.

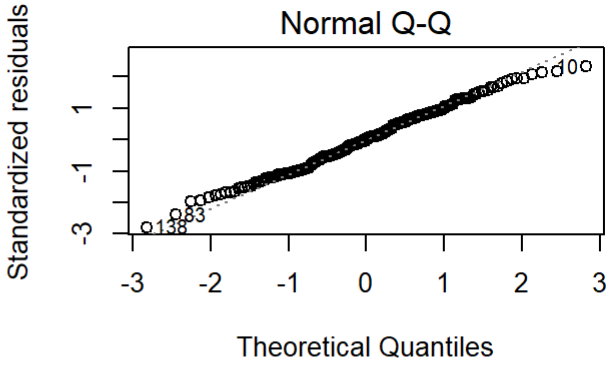
```
# Your code here...
ggplot(data, aes(x=log(chmax+1), y=log(performance))) + geom_point() +
  scale_colour_hue(l=50) + # Use a slightly darker palette than normal
  geom_smooth(method="lm",   # Add linear regression line
              se=FALSE,    # Don't add shaded confidence region
              fullrange=TRUE) # Extend regression lines
```

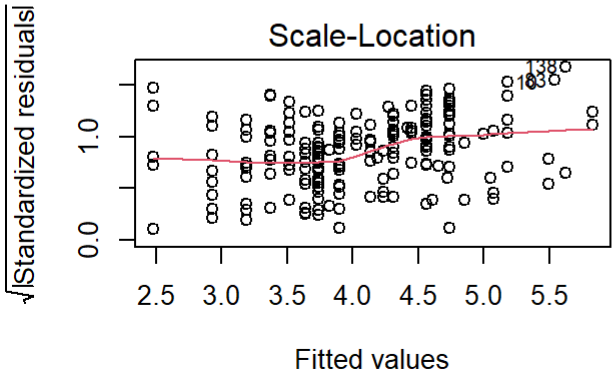
```
## `geom_smooth()` using formula 'y ~ x'
```

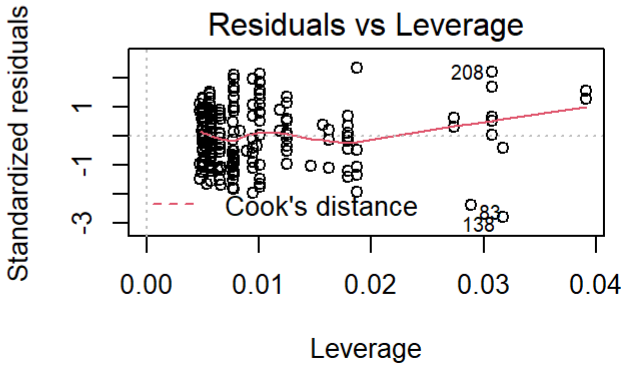



```
par(mfrow = c(2, 2))
plot(model2)
```

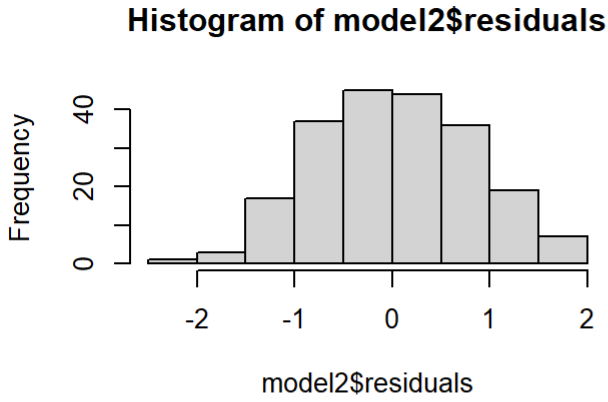









```
hist(model2$residuals)
```



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?

Answer

See below the 95% prediction intervals:

model1

fit: 516.4685

lower value of interval: 252.2519

upper value of interval: 780.6851

model2 (in original scale)

fit: 277.722969952335

lower value of interval: 55.1790730347687

upper value of interval: 1397.81340637138

We can see that the predicted value using *model2* is much smaller than the predicted value using *model1*. Also, the prediction interval under *model2* is much wider than the interval provided by *model1*.

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

print("predict chmax = 128 using model1, including 95% prediction interval: ")
```

```
## [1] "predict chmax = 128 using model1, including 95% prediction interval: "
```

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

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

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

print(paste("predict chmax = 128 using model2, fit value: ", exp(log_pred[1])))
```

```
## [1] "predict chmax = 128 using model2, fit value:  277.722969952335"
```

```
print(paste("predict chmax = 128 using model2, lwr value: ", exp(log_pred[2])))
```

```
## [1] "predict chmax = 128 using model2, lwr value:  55.1790730347687"
```

```
print(paste("predict chmax = 128 using model2, upr value: ", exp(log_pred[3])))
```

```
## [1] "predict chmax = 128 using model2, upr value:  1397.81340637138"
```

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)
head(data2)
```

	vendor <fctr>	chmax <int>	performance <int>
67	hp	10	17
68	hp	10	26
69	hp	24	32

	vendor <fctr>	chmax <int>	performance <int>
70	hp	19	32
71	hp	24	62
72	hp	48	64

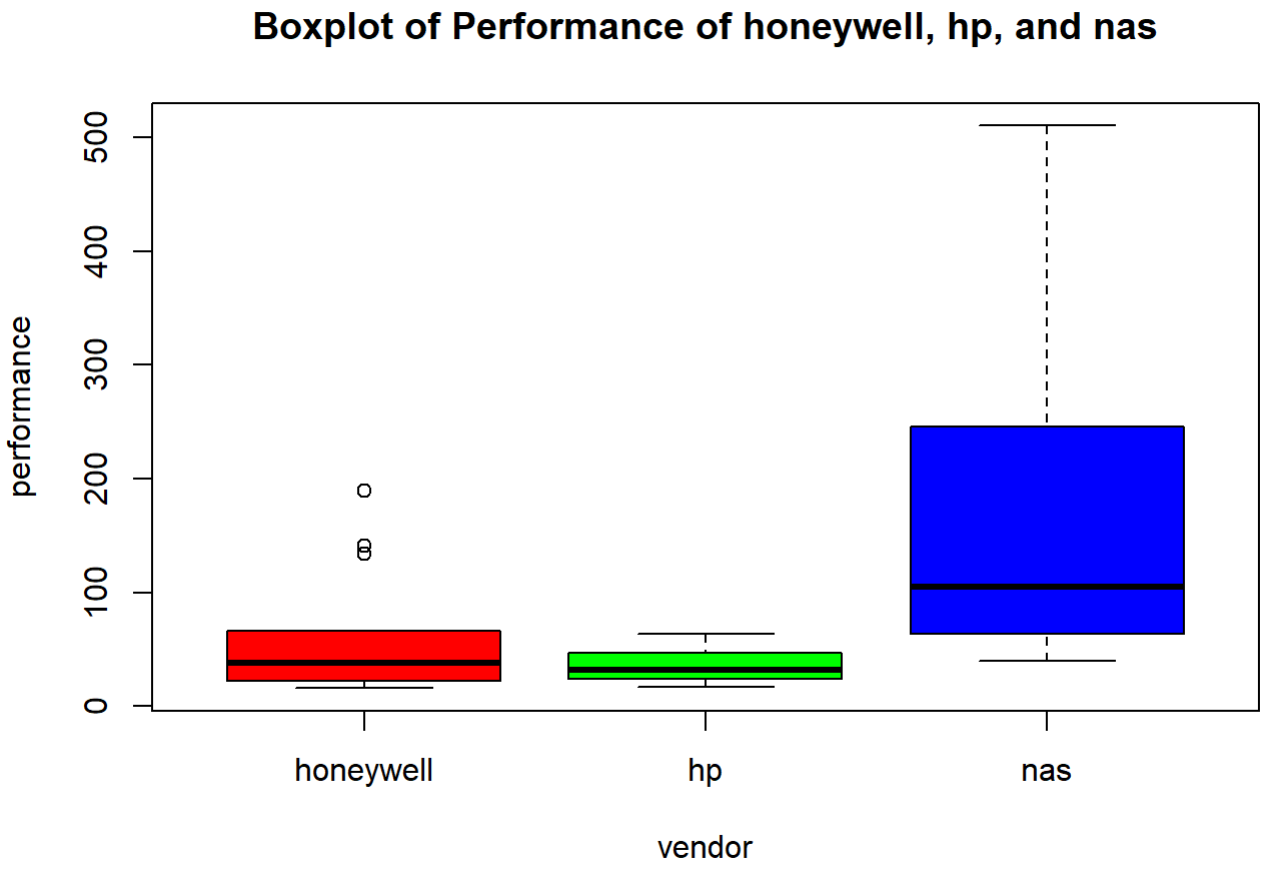
6 rows

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

Answer

We can see in the boxplot below that there is some between-variability as the performance means of honeywell and hp are somewhat the same while nas' mean is higher. Also, regarding the within-variability, hp's performance variability is the smallest, followed by honeywell's and nas with the largest variability. The plot indicates that honeywell might have few outliers.

```
# Your code here...
boxplot(performance ~ vendor, data = data2, main="Boxplot of Performance of honeywell, hp, and n
as", col= rainbow(3))
```



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.

Answer

See below p-value = 0.00553, meaning that at α-level equals 0.05 we can REJECT the null hypothesis stating that the three means are equal. This means that at least one vendor's performance mean is not equal to at least one of the other two means.

```
# Your code here...
anova = aov(performance ~ vendor, data = data2)
summary(anova)
```

```
##             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
```

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?

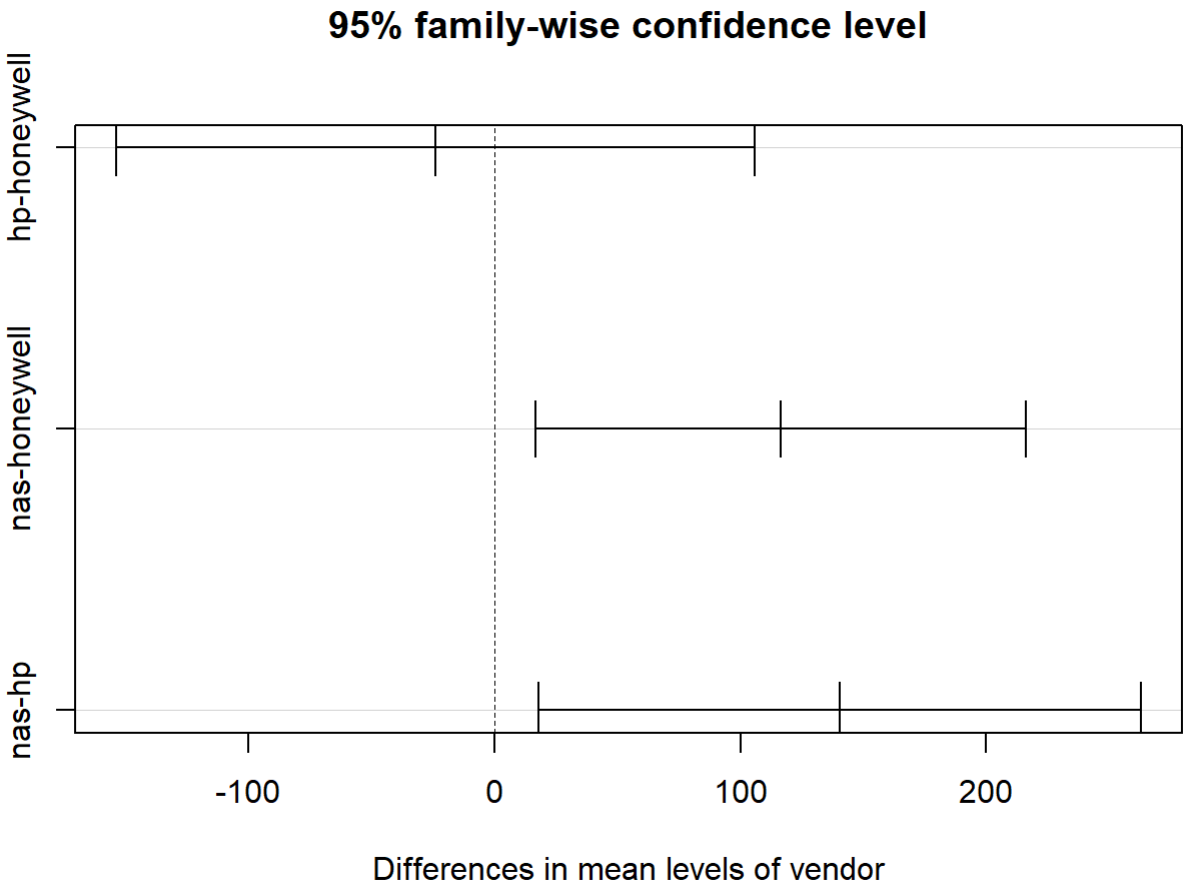
Answer

Observe in the TukeyHSD output and plot below that there is statistically significant difference between the means of nas-honeywell (p-value 0.0188830) and nas-hp (p-value 0.0214092) as indicated by a smaller p-value than 0.05. With such small p_values, we can reject the null hypothesis that these pairs' mean are equal, hence they are different. Note that the 95% confidence intervals of these pairs do not include 0 which is another indication nas-honeywell pair and nas-hp pair are different. However, hp-honeywell pair's confidence interval (95%) contains 0, has a large p_value of 0.8935 and thus, has no statistically significant difference.

```
# Your code here...
pc = TukeyHSD(anova)
print(pc)
```

```
##   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
```

```
plot(pc)
```


95% family-wise confidence level
hp-honeywell
nas-honeywell
nas-hp
-100
0
100
200
Differences in mean levels of vendor