HW1 Peer Assessment Solution

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	21	?			

Fill in the missing values in the analysis of the variance table. Note: Missing values can be calculated using the corresponding formulas provided in the lectures, or you can build the data frame in R and generate the ANOVA table using the aov() function. Either approach will be accepted.

Source	Df	Sum of Squares	Mean Squares	F-statistics	p-value
Treatments	2	7.224	3.6122	7.29	0.004

Source	Df	Sum of Squares	Mean Squares	F-statistics	p-value
Error	19	9.415	0.4955		
TOTAL	21	16.639			

$Df_{Treatments} = k - 1 = 2$

$Df_{Error} = N - k = 22 - 3 = 19$

$Df_{Total} = Df_{Treatments} + Df_{Erorr} = (N - k) + (k - 1) = N - 1 = 21$

$SST_R = MST_R \times (k - 1) = 3.6122 \times 2 = 7.224$

$SST = SSE + SST_R = 7.224 + 9.415 = 16.639$

$MSE = SSE/(N - k) = 9.415/19 = 0.4955$

$F\text{-}test = MST_R/MSE = 3.6122/0.4955 = 7.29$

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 (i.e. explain what μ_1, μ_2, and μ_3 mean in words in the context of this problem). Find the estimates of these parameters.

- μ_1: true mean phase shift for subjects in Control group. Its estimate, $\hat{\mu}_1$, is -0.3088
- μ_2: true mean phase shift for subjects in Knees group. Its estimate, $\hat{\mu}_2$, is -0.3357
- μ_3: true mean phase shift for subjects in Eyes group. Its estimate, $\hat{\mu}_3$, is -1.5514

Question A3 - 5 pts

Use the ANOVA table in Question A1 to write the:

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

H_0: μ_1=μ_2=μ_3

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

H_A: At least 2 of the means are not equal (μ_1≠μ_2 and/or μ_1≠μ_3 and/or μ_3≠μ_2)

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

$F(k - 1, N - k) = F(2, 19)$

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

The p-value is 0.004, as given in the ANOVA table.

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

We reject the null hypothesis that all three means are equal because the p-value is much smaller than 0.05. Therefore, the mean of the phase shift is not the same for all three treatment groups, and we conclude that light treatment does 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()`.

```
# Import libraries
library(ggplot2)
```

```
## Warning: package ’ggplot2’ was built under R version 4.0.5
```

```
library(ggpubr)
library(car)
```

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

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

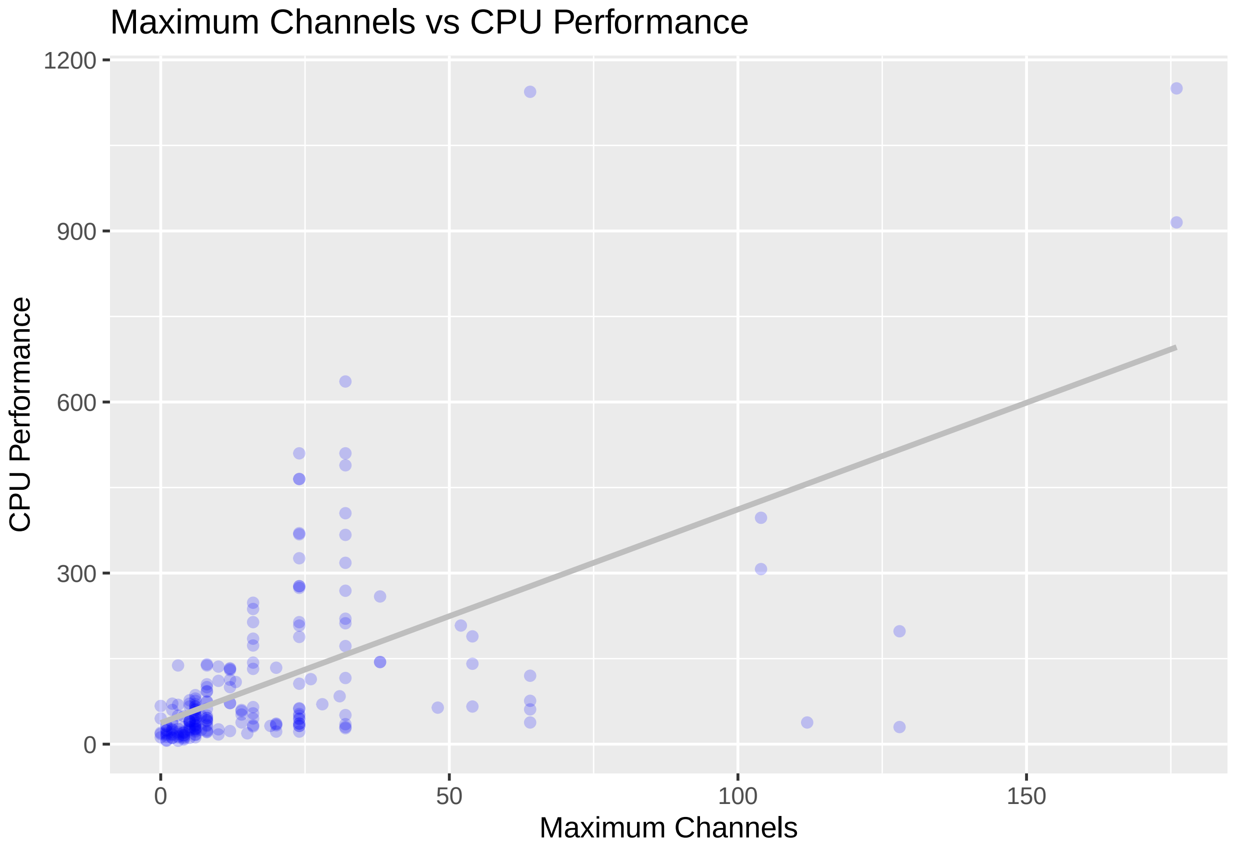
```
##    vendor chmax performance
## 1 adviser   128         198
## 2  amdahl    32         269
## 3  amdahl    32         220
```

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...

ggplot(data=data, aes(x=chmax, y=performance)) +
  geom_point(alpha=I(0.2),color='blue') +
  xlab('Maximum Channels') +
  ylab('CPU Performance') +
  ggtitle('Maximum Channels vs CPU Performance')+
  geom_smooth(method="lm",color='gray', se=FALSE)
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



There seems to be a positive, linear relationship of moderate strength between CPU performance and the maximum number of channels. There is a general increasing trend in CPU performance as the maximum channels increases. As the maximum number of channels increases the variance of CPU performance appears to increase as well.

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(data$performance, data$chmax)
```

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

The correlation coefficient of 0.6052093 suggests that we have a moderate positive linear relationship between *chmax* and *performance*.

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

I would recommend attempting a simple linear regression model because it is easy to interpret, but we are likely going to want to attempt a Box-Cox transformation to reduce the heteroskedasticity.

Note: Any other logical answer is acceptable for full credit.

d. **1 pts** Based on the analysis above, would you pursue a transformation of the data?

Yes, I would recommend transforming the data using a Box-Cox transformation because of the heteroskedasticity in CPU performance as maximum channels increases.

Note: Any other logical answer is acceptable for full credit.

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)
```

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

```
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
```

```
sigsq = summary(model1)$sigma ** 2
```

The model parameters are:

- Intercept, β_0, and its estimate, $\hat{\beta}_0$, is 37.2252
- Slope, β_1, and its estimate, $\hat{\beta}_1$, is 3.7441
- Variance of the error terms, σ^2, and its estimate, $\hat{\sigma}^2$, is $128.3408^2 = 16471.37$

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

$\widehat{performance} = 37.2252 + 3.7441 \times chmax$

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

A one unit increase in maximum channels increases CPU performance by 3.7441 units on average.

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

```
confint(model1)['chmax',]
```

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

The 95% confidence interval has a lower bound of 3.069251 and an upper bound of 4.418926. Given that the confidence interval does not include zero, β_1 is statistically significant at this level.

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?

- $H_0 : \beta_1 \leq 0$
- $H_A : \beta_1 > 0$

We need to conduct a one-sided t-test on β_1. We can extract the t-value from the summary table of *model1* and the degrees of freedom from *model1*. We then calculate the distribution function on the upper tail, since we are testing if β_1 is positive.

```
tval = summary(model1)$coefficients['chmax','t value']
df = model1$df.residual
pval = pt(tval, df, lower=F)
pval
```

```
## [1] 1.423882e-22
```

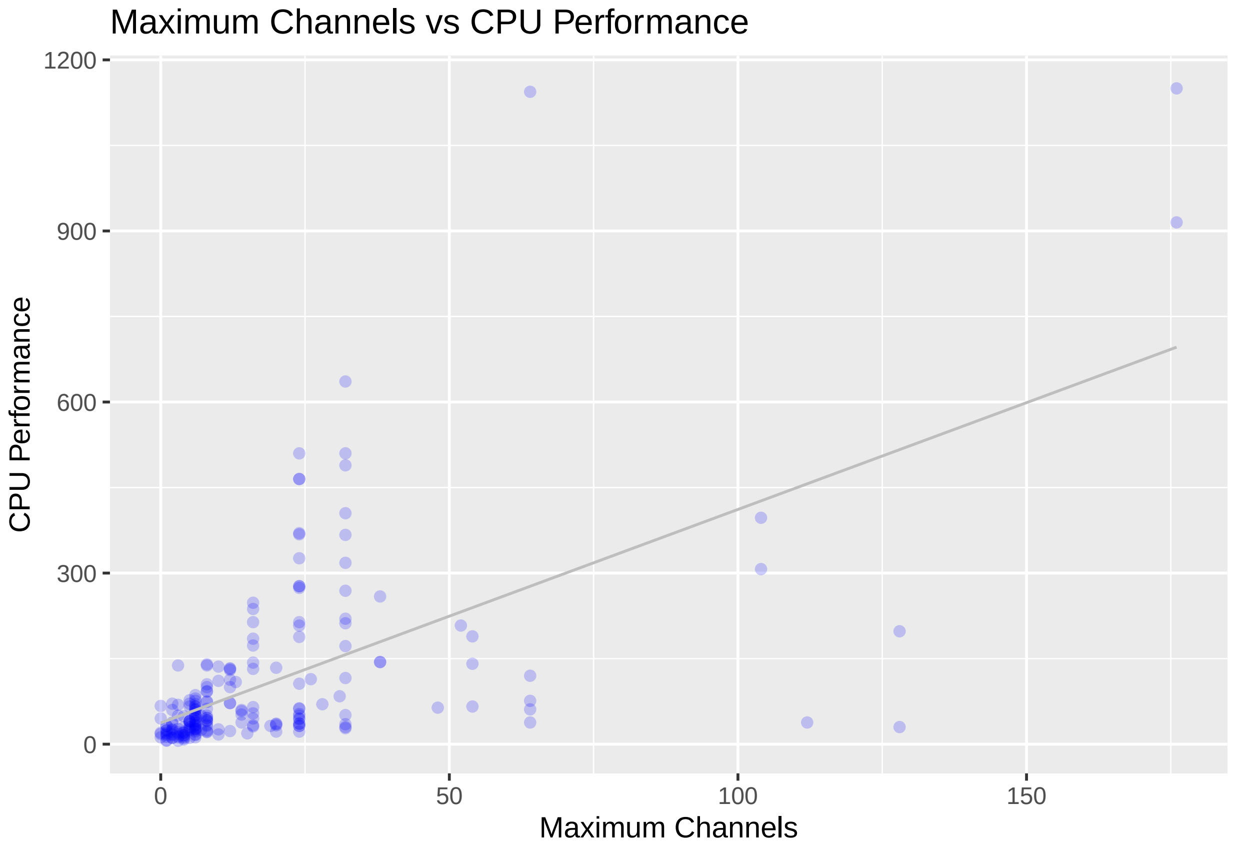
The p-value is 1.423882×10^{-22}, which is approximately equal to zero. Since this value is less than the α-level of 0.01, we conclude that β_1 is statistically significantly positive.

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=data, aes(x=chmax, y=performance)) +
  geom_point(alpha=I(0.2),color='blue') +
  xlab('Maximum Channels') +
  ylab('CPU Performance') +
  ggtitle('Maximum Channels vs CPU Performance') +
  geom_line(data=data, aes(x=chmax, y=model1$fitted.values), color="grey")
```



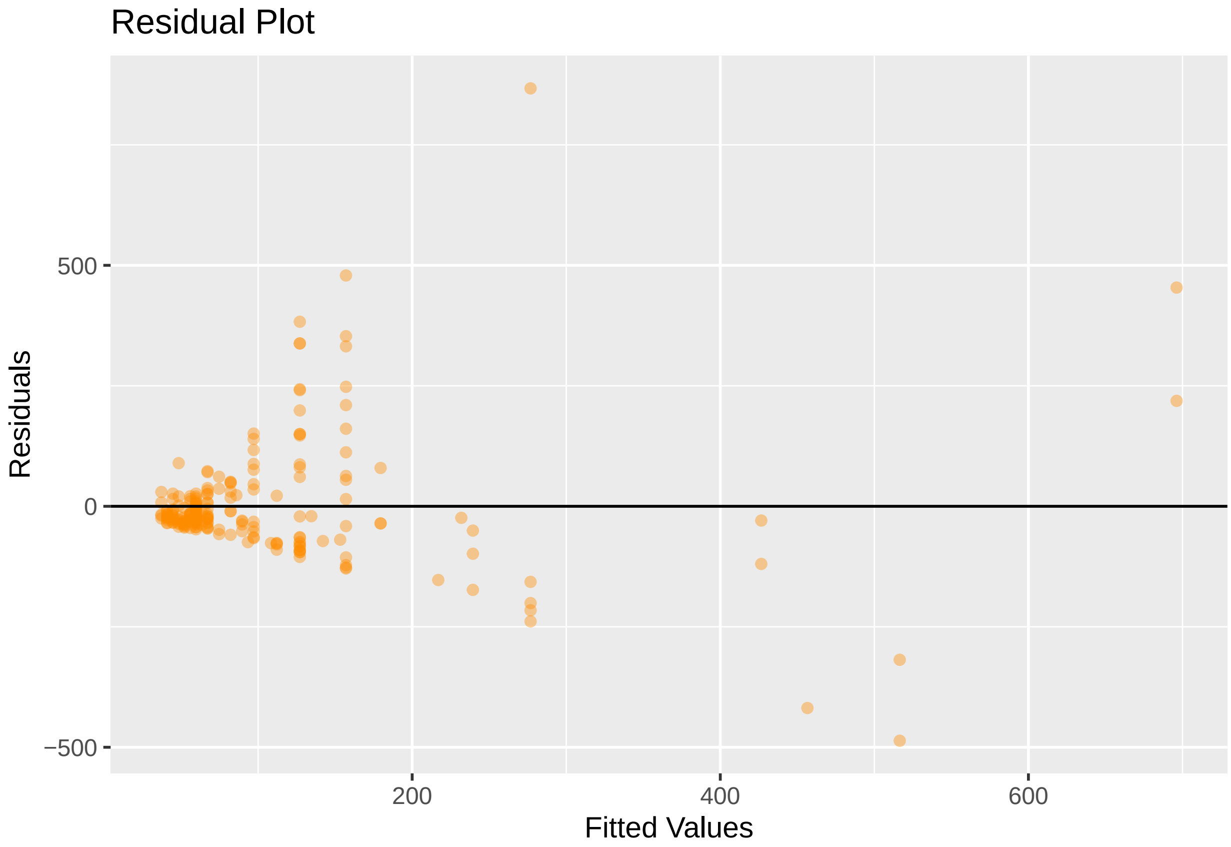
Model Assumption(s) it checks: Linearity/Mean Zero, Independence ("Uncorrelated errors") and Constant Variance.

Note: Any one of the assumptions above is acceptable for full credit.

Interpretation: There seems to be issues with large values of *chmax* not evenly distributed across the model line. For instance, between 40 and 150 maximum channels nearly all of the data points fall below the model line. This suggests that the linearity assumption may not strongly hold.

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

```
# Your code here...

ggplot(data=data, aes(x=model1$fitted.values, y=model1$residuals)) +
  geom_point(alpha=I(0.4),color='darkorange') +
  xlab('Fitted Values') +
  ylab('Residuals') +
  ggtitle('Residual Plot') +
  geom_hline(yintercept=0)
```



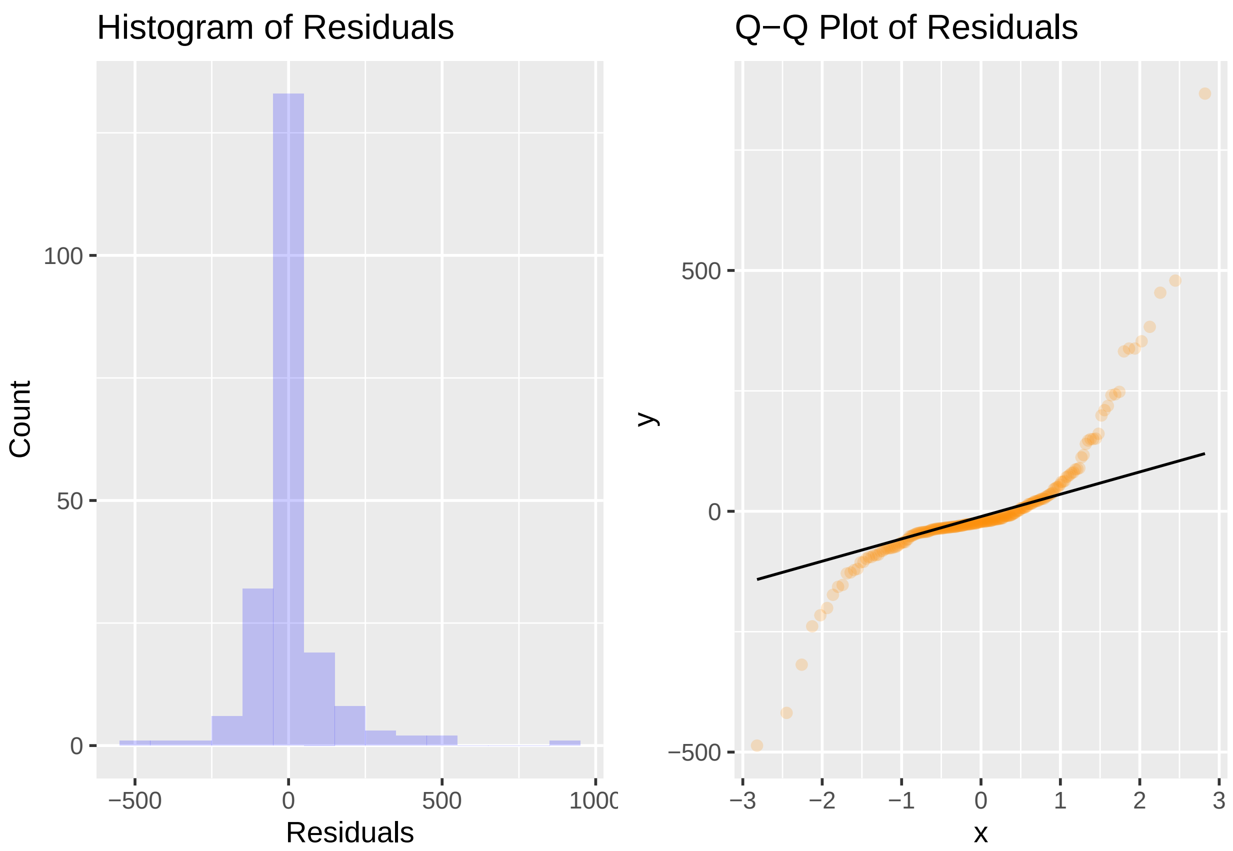
Model Assumption(s) it checks: Independence ("Uncorrelated errors") and Constant Variance

Interpretation: Although at first glance, there appear to be clusters at the lower ranges, this is a feature of the data set. Most of the data points have small predictor values, which is why most of the fitted values are at the lower range. Since most of the data points have small maximum channel values, most of the fitted values for performance are similarly small. You may also notice two vertical bands between 100 and 200. These are not clusters. The predictor, *chmax*, can only take discrete integer values, and many of the data points have identical maximum channel values. This leads to the vertical bands that we see in the residual plot above. Based on this analysis there do not appear to be correlated errors.

The plot does show heteroscedasticity with variance increasing as the fitted values increase. This suggests that the constant variance assumption may not hold.

c. **3 pts** Histogram and q-q plot

```r
# Your code here...

hp = qplot(model1$residuals,
           geom="histogram",
           binwidth=100,
           main = "Histogram of Residuals",
           xlab = "Residuals",
           ylab = "Count",
           fill=I("blue"),
           alpha=I(0.2))

qqp = ggplot(data, aes(sample=model1$residuals)) +
  stat_qq(alpha=I(0.2),color='darkorange') +
  stat_qq_line() +
  ggtitle("Q-Q Plot of Residuals")

ggarrange(hp, qqp, ncol=2, nrow=1)
```



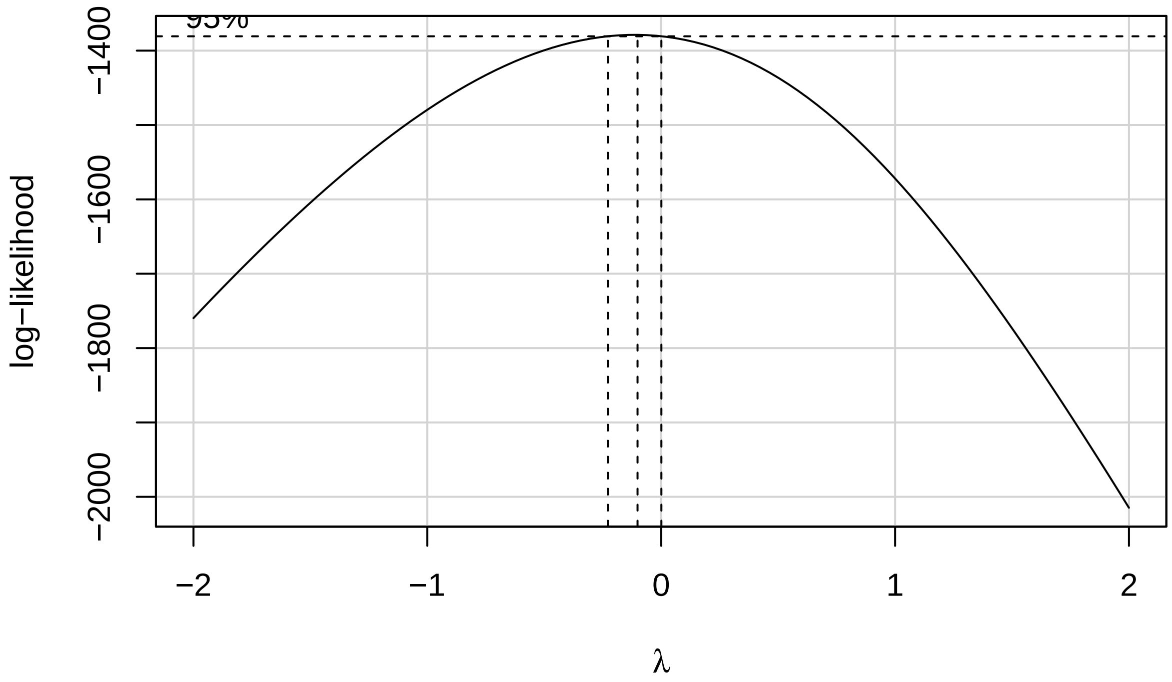
Model Assumption(s) it checks: Normality

Interpretation: Both the histogram and the quantile-quantile plot indicate that the residuals have heavy tails. This suggests that the normality assumption may not hold.

Question B4: Improving the Fit - 10 pts

a. **2 pts** Use a Box-Cox transformation (`boxCox()`) in `car()` package or (`boxcox()`) in `MASS()` package 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...

# Perform Box-Cox transformation
bc = boxCox(model1)
```



```
# Find the optimal lambda
opt.lambda = bc$x[which.max(bc$y)]
# Round it to the nearest 0.5
cat("Optimal lambda:", round(opt.lambda/0.5)*0.5, end="\n")
```

```
## Optimal lambda: 0
```

The optimal lambda value is zero, suggesting that the log of the response may improve normality and/or constant variance.

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.

```
# Your code here...

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

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

```
r2m1 = summary(model1)$r.squared
r2m2 = summary(model2)$r.squared

cat("R-squared of model1 is:", r2m1, end="\n")
```

```
## R-squared of model1 is: 0.3662783
```

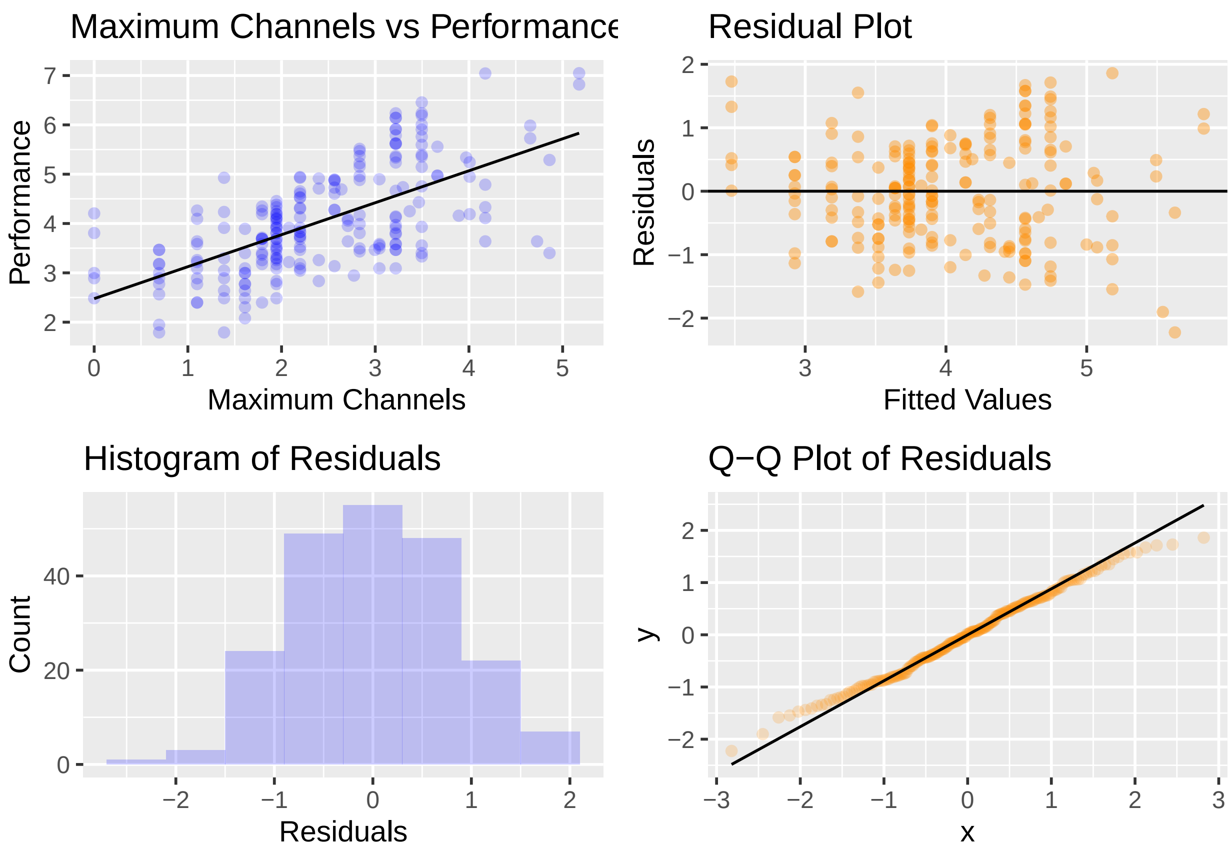
```r
cat("R-squared of model2 is:", r2m2, end="\n")
```

```
## R-squared of model2 is: 0.4102926
```

The R^2 value of *model1* is 0.366, and the R^2 value of *model2* is 0.410. This indicates that there is an improvement in 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?

```r
# Your code here...

sp2 = ggplot(data=data, aes(x=log(chmax+1), y=log(performance))) +
  geom_point(alpha=I(0.2),color='blue') +
  xlab('Maximum Channels') +
  ylab('Performance') +
  ggtitle('Maximum Channels vs Performance') +
  geom_line(data=data, aes(x=log(chmax+1), y=model2$fitted.values))

rp2 = ggplot(data=data, aes(x=model2$fitted.values, y=model2$residuals)) +
  geom_point(alpha=I(0.4),color='darkorange') +
  xlab('Fitted Values') +
  ylab('Residuals') +
  ggtitle('Residual Plot') +
  geom_hline(yintercept=0)

hp2 = qplot(model2$residuals,
            geom="histogram",
            binwidth=0.6,
            main = "Histogram of Residuals",
            xlab = "Residuals",
            ylab = "Count",
            fill=I("blue"),
            alpha=I(0.2))

qqp2 = ggplot(data, aes(sample=model2$residuals)) +
  stat_qq(alpha=I(0.2),color='darkorange') +
  stat_qq_line() +
  ggtitle("Q-Q Plot of Residuals")

ggarrange(sp2, rp2, hp2, qqp2, ncol=2, nrow=2)
```



CPU performance appears to be evenly distributed across the model line for all maximum channel values. This suggests that the linearity/mean zero assumption holds.

There appears to be homoskedasticity in the residual plot. This suggests that the constant variance assumption holds.

There also does not appear to be any clear pattern or clustering in the residuals. This suggests that the errors are uncorrelated. We cannot definitely state that the errors are independent because the data came from an observational study.

Both the histogram and the quantile-quantile plot of the residuals suggests that the normality assumption holds.

All model assumptions appear to hold using the log-transformed data!

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?

```r
# Your code here...

newcpu = data.frame(chmax=128)

cat("model1:", end="\n")
```

```
## model1:
```

```
predict(model1, newcpu, interval="prediction", level=0.95)
```

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

```
cat("model2:", end="\n")
```

```
## model2:
```

```
exp(predict(model2, newcpu, interval="prediction", level=0.95))
```

```
##       fit      lwr      upr
## 1 277.723 55.17907 1397.813
```

When there are a maximum of 128 channels in CPU, model1 predicts a CPU performance of 516.4685 with a lower bound of 252.2519 and an upper bound of 780.6851 for the 95% prediction interval, while model2 predicts a CPU performance of 277.723 with a lower bound of 55.17907 and an upper bound of 1397.813 for the 95% prediction interval.

We can see that model2, which uses the log transformation, has a larger prediction interval than model1, the model using the untransformed data. We can also see that the predicted value is much lower using Model2 than Model1. With that said, both predicted values fall within the prediction intervals of the other model.

Based on the goodness of fit assessments, model1's prediction interval is likely to be inaccurate. Hence, model2's prediction interval seems to be much more reliable than model1's. However, we might need to split our data set into training and testing sets and calculate prediction accuracy measurements in order to further evaluate the prediction accuracy of the models.

Note: Any other logical answer is acceptable for full credit.

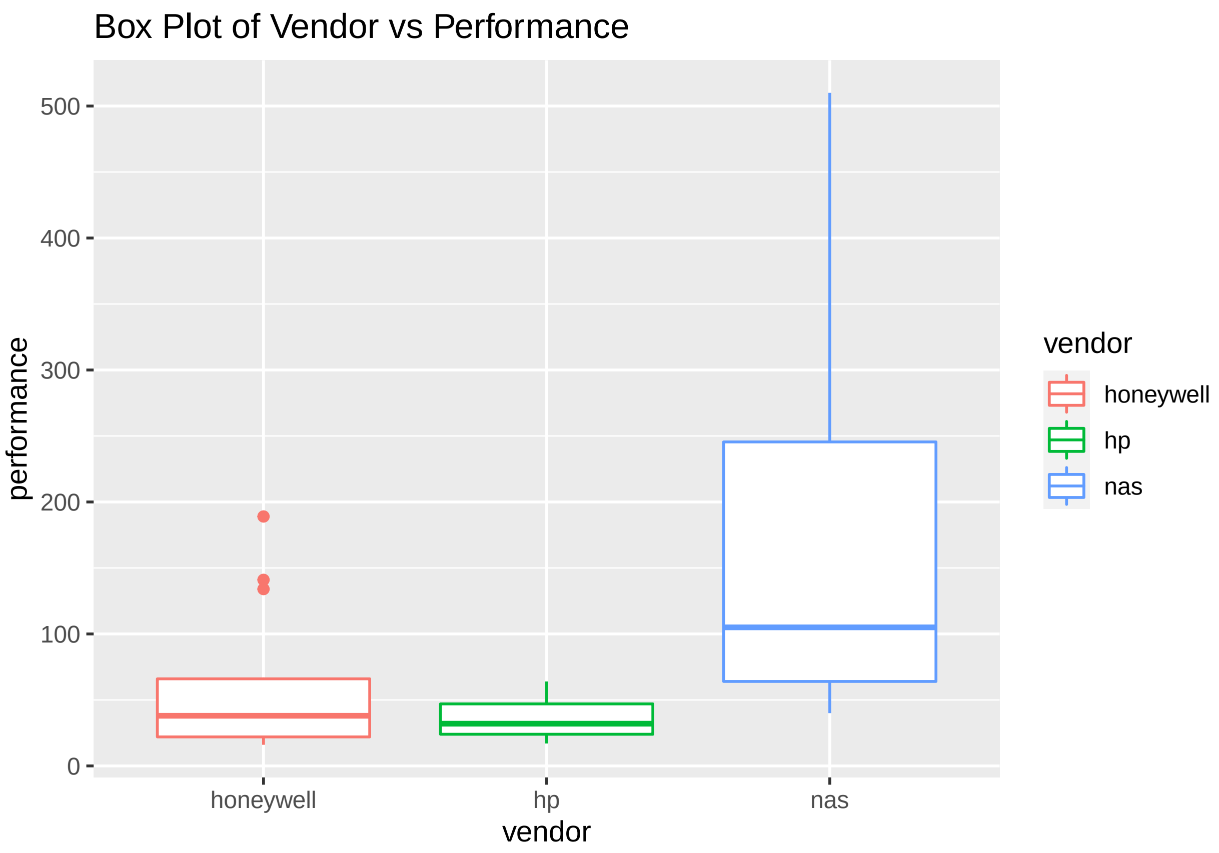
Part C. ANOVA - 8 pts

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...

ggplot(data2, aes(x=vendor, y=performance, color=vendor)) +
  geom_boxplot() +
  ggtitle("Box Plot of Vendor vs Performance")
```



The box plot above suggests that CPU performance differs between the vendors. The vendor nas appears to have CPUs with higher performance than either honeywell or hp.

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...
model3 = aov(performance ~ vendor, data2)
summary(model3)
```

```
##              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
```

The p-value of the F-test is 0.00553, which is less than the α-level of 0.05. We reject the null hypothesis that the mean CPU performance off all three vendors is equal, and conclude that at least two means statistically significantly differ from each other.

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

```r
# Your code here...
TukeyHSD(model3, "vendor", conf.level = 0.95 )
```

```
##   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
```

Nas-honeywell and nas-hp are the two pairs of vendors that have statistically significantly different means at the significance level of 0.05 since the p-values of the pairwise comparisons are smaller than the α-level of 0.05; In fact, the intervals fall completely on the positive side and don't include zero. In the context of the problem, we can conclude that the mean CPU performance of nas is significantly higher than the mean CPU performance of the other two vendors honeywell and hp.