Statistical Data Analysis
Assignment 7

Matt Kuodis & James Zoryk.
2640428 : 2663347
Group: 36
Due Date: 24.05.2022.

Question 1.

Part b.

Before we begin our process of linear regression, we take a look at the data sets we have. We want to get a feel for the distribution and overall shape of the data, and see whether similarities are immediately noticeable. We plot the histograms, and QQ-plots of the data, which gives us the graphs below (2 and 1).

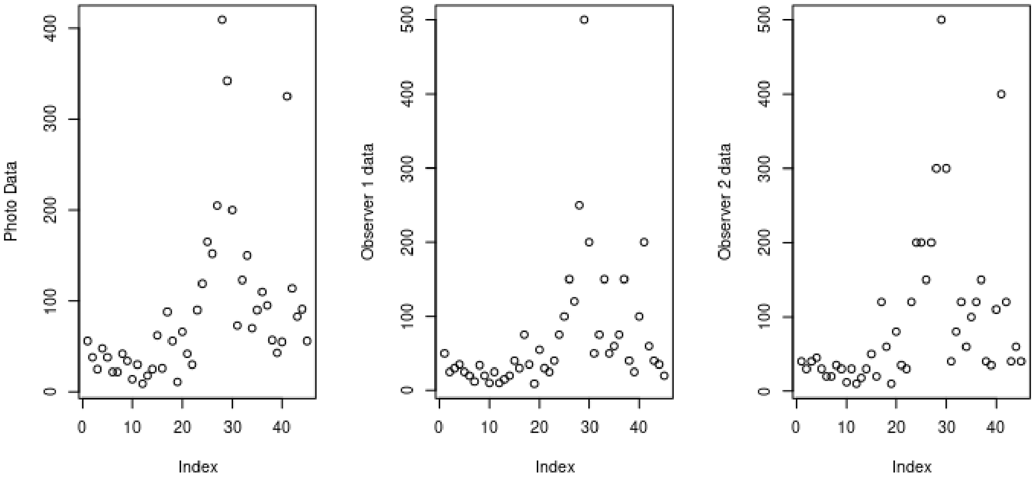

Figure 1: QQ-plots of the data sets.

We immediately take note that the data looks similar. To quantify the similarities, we try to express our observed variable (photo count) as a linear function of the dependent variable (observer counts), for both observers. To do this, we use the built in `lm()` function, with $y =$ photo count, and $x =$ observer1 or observer2.

We do this separately for the two dependent variables, and get the following value tables,

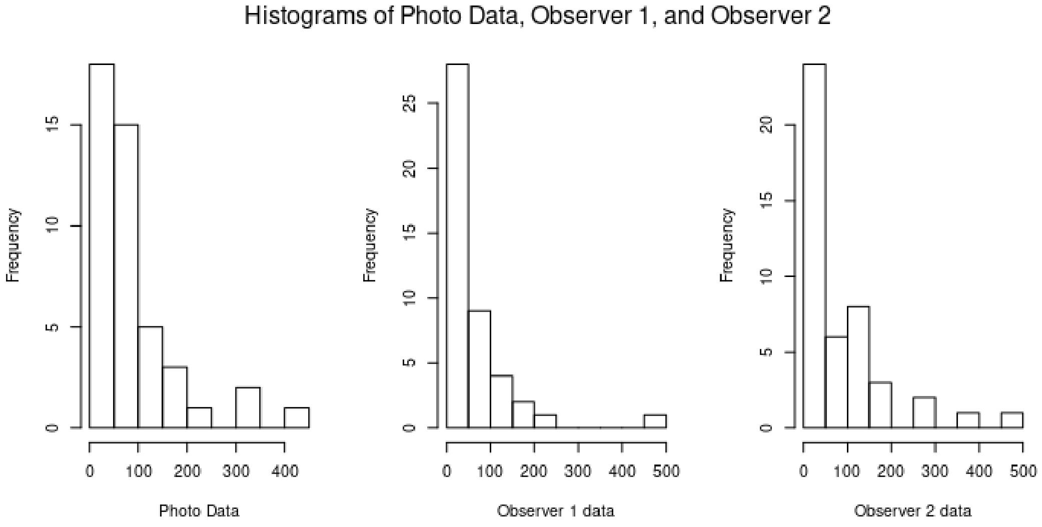

Figure 2: Histograms of the data sets.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	26.64957	8.61448	3.094	0.00347
observer1	0.88256	0.07764	11.367	1.5×10^{-14}

and

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	16.16428	6.82973	2.367	0.0225
observer2	0.76907	0.04835	15.905	2×10^{-16}

for observer 1, and observer 2 respectively, but in addition we also get the R^2 and F values for both models, namely $R^2_{obs1} = 0.7503$ with $F_{obs1} = 129.2$, and $R^2_{obs2} = 0.8547$ with $F_{obs2} = 253$.

For the test with $H_0 : \beta_1 = 0$ against $H_a : \beta_1 \neq 0$, at significance level $\alpha = 0.05$, we see from the p-values in the tables above (1.5 $\times 10^{-14}$, and 2 $\times 10^{-16}$ respectively) that we can safely reject the null hypothesis, and accept the alternative hypothesis that there is in fact a linear relation between the photo count and the observer counts for both cases.

If we plug in the values computed above, we can express photo count (P) in terms of the observer data (O_1 and O_2) as the following:

$$P = 26.64957 + 0.88256 \times O_1 + \varepsilon \qquad P = 16.16428 + 0.76907 \times O_2 + \varepsilon$$

where ϵ are the residuals. If we plot these lines, we get the graphs below (3) where the red line is given by the functions above.

Part c.

Here we plot the residuals of the two models computed in the earlier sub-question. This gives us the following plots:

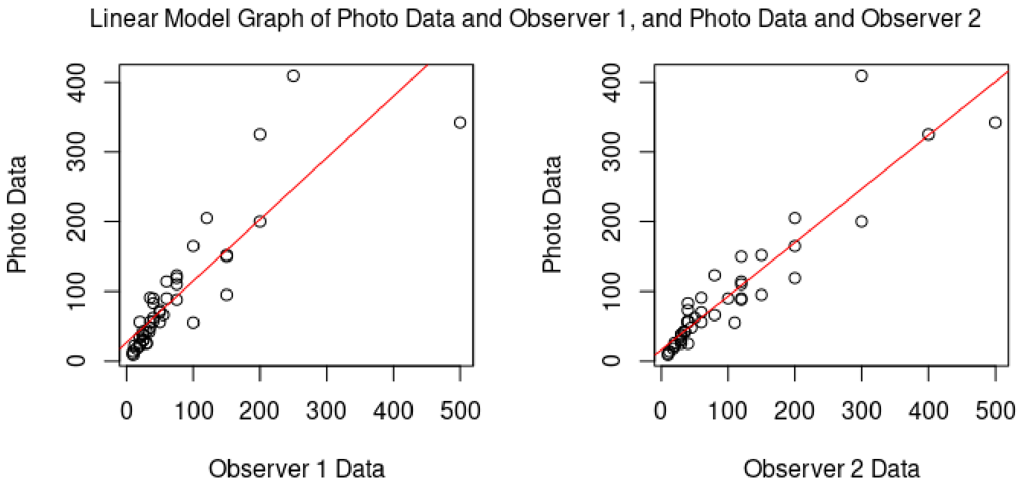

Figure 3: Linear models for the data sets.

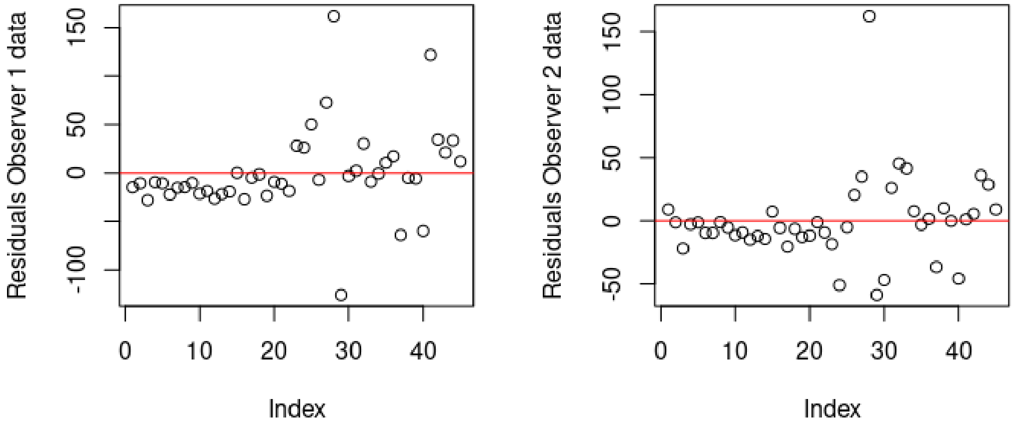

Figure 4: Residuals around the line Y=0.

Now generally, it is assumed that residuals are distributed normally with a distribution $N(0, \sigma^2)$, but from the plots it seems that the outliers in the data skew the residual distribution in a way that may revoke our assumptions of normality. We investigate this more thoroughly in the following sub-question.

Part d.

We begin our investigation on the normality of the residuals by setting up our hypotheses as follows, $H_0 : \varepsilon \sim N(0, \sigma^2)$ against $H_a : \varepsilon \not\sim N(0, \sigma^2)$.

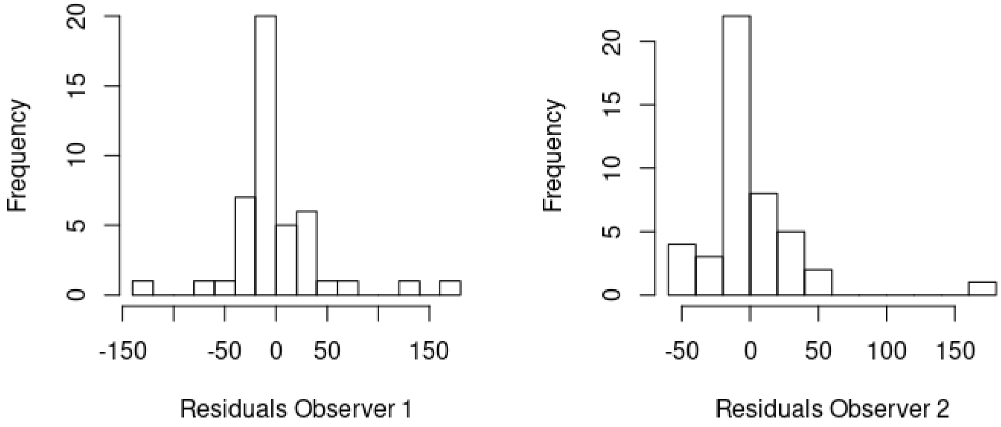

Figure 5: Histograms of residuals.

We first compute the standard deviation of both residual samples, and use the `lm.norm.test()` function to investigate normality. Before we do this, we perform the Kolmogorov-Smirnov test on our residuals compared to an actually normal error sample that shares the standard deviation of our residuals. This gives us the test statistics $K_1 = 0.194$ and $K_2 = 0.204$. We then use `lm.norm.test()` to get a thousand K values. The function creates a thousand linear models with the coefficients generated by the given data sets (same as those we denoted in sub-question **b.**, and padded with truly normally distributed residuals. Then the Kolmogorov-Smirnov test is performed on the residuals of these linear models, and a truly normal sample. For each linear model, we get a corresponding K value.

Subsequently we compute how many of these generated K values are greater than the test K_1 (or K_2 if we repeat this for observer 2 data). We compute our p-values as

$$p_1 = \frac{|\{K > K_1\}|}{1000}$$

and similarly for p_2. In either case we get that $p_1 = p_2 = 0$, meaning that our values K_1 and K_2 are far larger than what would be expected from a normally distributed residual sample. Thus we reject the null hypothesis, and conclude that the residuals are not normally distributed.

Part e.

For this sub-question we transform all our data with the `log()` function. This gives us the graphs below.

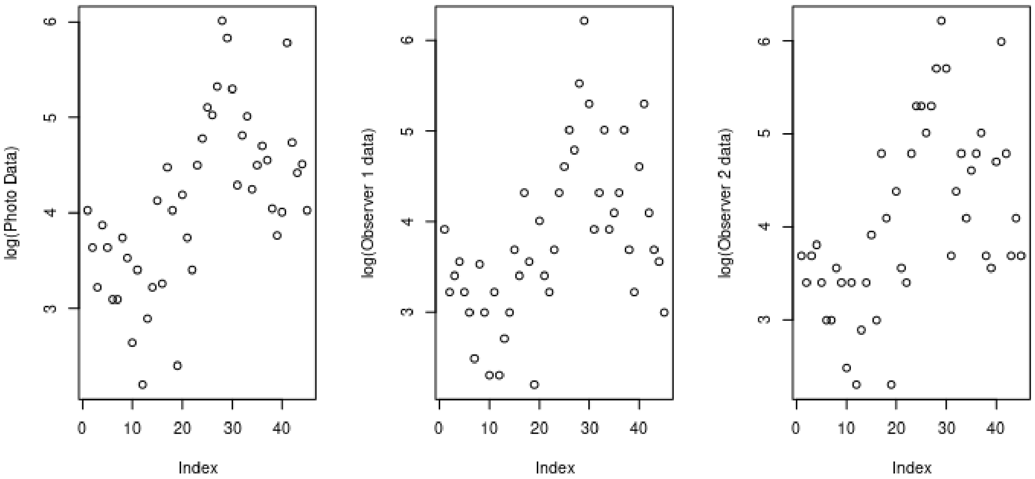

Figure 6: Log transformed data.

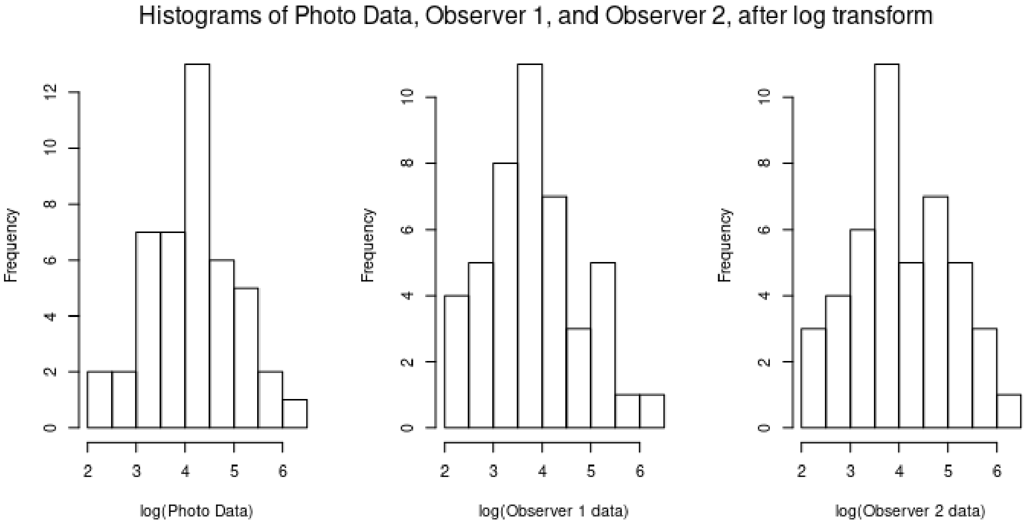

Figure 7: Histograms of the log(data).

We note that the data sets overlap much more, now that the severity of the outliers has been reduced. Data points that over or underestimated the amount of geese by a hundred or more now stray far less from the $y = x$ line that we'd see if the photo counts and the observer counts were (near) identical.

Furthermore we compute the following tables as earlier:

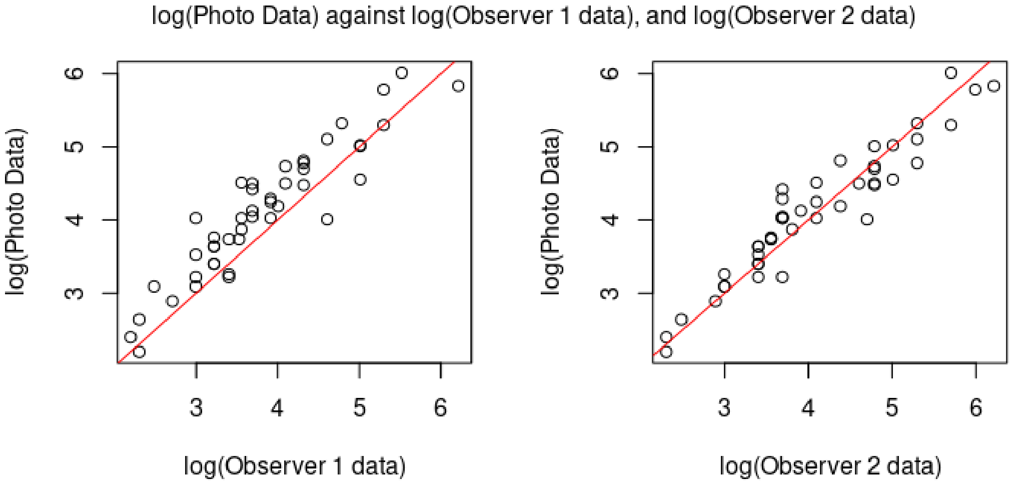

Figure 8: Comparisons between the log transformed data sets.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.65115	0.21348	3.05	0.00391
log(observer1)	0.90680	0.05444	16.66	2×10^{-16}

and

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.57016	0.17970	3.173	0.00279
log(observer2)	0.86778	0.04285	20.252	2×10^{-16}

As earlier, we see from the p-values that given the hypotheses $H_0 : \beta_1 = 0$ against $H_a : \beta_1 \neq 0$, at significance level $\alpha = 0.05$, we can reject H_0 and conclude that there is a relation between the two data sets, for `log`(photo data) and `log`(observer 1 data), and `log`(photo data) and `log`(observer 2 data).

Similarly we also analyze the residuals of this transformed data set, in the same way as we did in **d.**. Retracing the same steps we get the following graphs for the residuals.

We investigate whether the residuals are distributed normally, in the same way as we did earlier, using the `lm.norm.test()` function. We set up our test hypotheses as $H_0 : \varepsilon \sim N(0, \sigma^2)$ against $H_a : \varepsilon \not\sim N(0, \sigma^2)$, with significance $\alpha = 0.05$. This time, however, we get $p_{1,log}$-value $= 0.559$, and $p_{2,log}$-value $= 0.64$, both of which are larger than 0.05, and thus we do not reject the null hypothesis that the residuals are in fact distributed normally.

Part f.

With the linear models we've computed so far, we note that the `log` transformed data fits better overall, and specifically the data from observer 2 does well in approximating the photo data. However whether it is wise to use transformed data remains an unanswered issue.

In this case, we need to look at the context of the data. The data compared the counts of geese made by two human observers, and compared their estimates to the actual amount of geese, which we know from the photos taken of the flocks. The human observers can obviously not count each goose individually, so they resort to educated guesses when it comes to larger

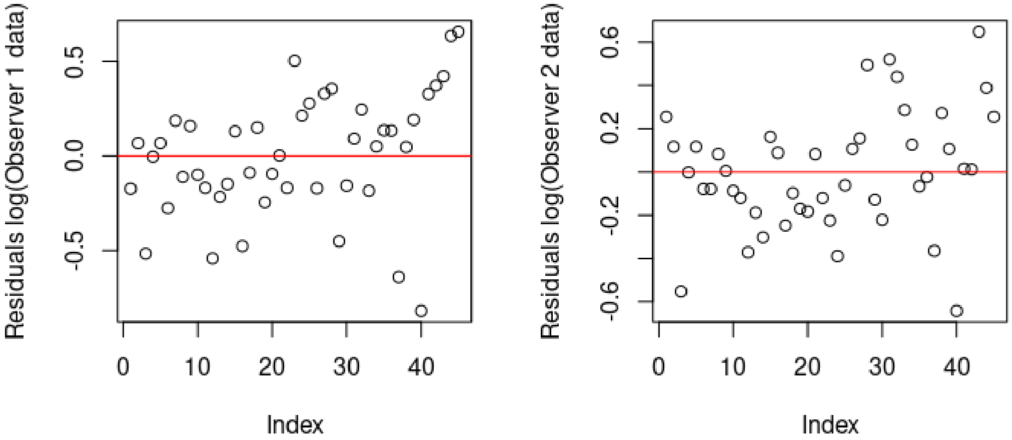

Figure 9: log(data) residuals.

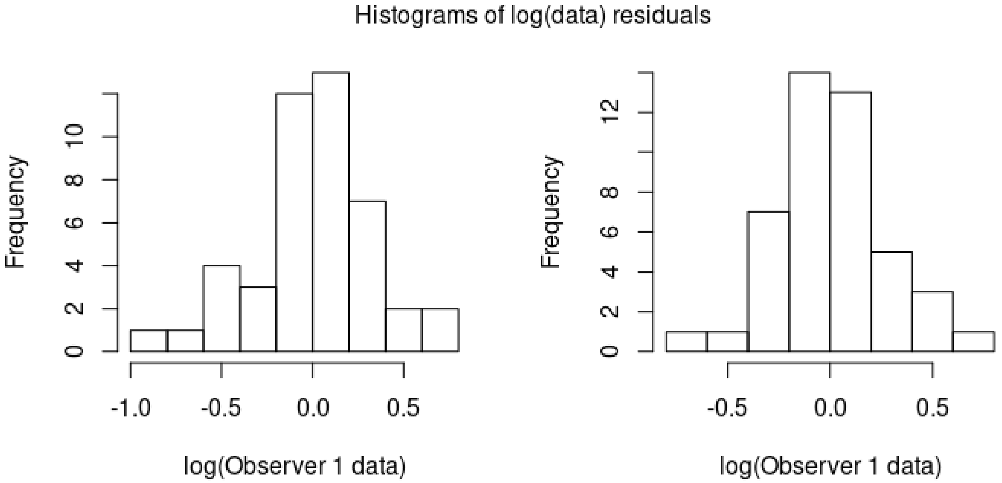

Figure 10: Histograms of log(data) residuals.

flocks. We see from the original data in 1 that at most the humans estimate the largest flocks at around 500 geese, while the photo count tells us it is in fact around 400 geese. At times the human observers also underestimate the number of geese.

Overall, since when it comes to smaller flock sizes the guesses are good enough for us to comfortably conclude a linear relation between the observers and the photo count, we can consider large outliers in the residual data as expected errors when estimating large flock sizes by making educated guesses. In short, the larger the flock size, the larger the error that the human observers make. Taking the `log` transform of the data allows us to minimize the influence of these outliers, and take the transformed data models as our preferred models, out of which the one of observer 2 fits the best.

With these considerations in mind, we conclude that the transformed data from our second observer is the most trustworthy.

Part g.

As there is a noticeable positive correlation between the photo count and the observer counts, we can say that the photo count does a good job at reflecting the observer counts. If considerations about errors in large flock sizes are regarded as in sub-question **f.**, then the transformed data from the photo count is a near perfect reflection of the transformed data from observer 2, as shown in the right graph in figure 8.

Question 7.2.

Part b. For this task we shall use the 'step up' method to create our proper multiple linear regression model. To achieve this we use the various t-test with the following null and alternative hypothesis:

$$\begin{aligned} H_0 :&\beta_k = 0;\ \beta_j \text{arbitrary for } 0 \leq j \leq p,\ j \neq k. \\ H_1 :&\beta_k \neq 0;\ \beta_j \text{arbitrary for } 0 \leq j \leq p,\ j \neq k. \end{aligned}$$

Here out test statistic is defined as

$$|T_k| = \frac{|\hat{\beta}_k|}{\sqrt{c\hat{o}v(\hat{\beta}_{kk}}}.$$

with n being the number of observations and p the explanatory variables This give us $n-p-1$ degrees of freedom. The test significance level is set to $\alpha = 0.05$. We shall compute a simple regression model for each of the explanatory variables given in the pollen data, then we shall compare the 'multiple R-squared' values of each of the linear models, which are shown in Table 1.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	45.3171	4.8976	9.25	0.0000
wind	-0.6331	0.1005	-6.30	0.0000
$\mathcal{R}^2 =$	0.586			

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-46.4292	9.9542	-4.66	0.0001
temperature	0.7850	0.1273	6.17	0.0000
$\mathcal{R}^2 =$	0.576			

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	27.4446	6.4368	4.26	0.0002
humidity	-0.2088	0.1049	-1.99	0.0563
$\mathcal{R}^2 =$	0.124			

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-1.4328	5.3297	-0.27	0.7900
insolation	0.2299	0.0742	3.10	0.0044
$\mathcal{R}^2 =$	0.255	p-value =		

Table 1: Computed summary for linear models of a single explanatory variable.{wind, temperature, humidity and insolation}

We order the 'multiple R-squared' values for the explanatory variables to get wind(0.586) > temperature(0.576) > humidity(0.124) > insolation(0.2552). Therefore wind has the largest determination coefficients, meaning it contributes the most to the variance of oxidant levels measured. Furthermore we need to check the t-test of the wind and the associated p-level to see wind is significant to be added to the model. Reading off the table we see that it has a p-value of $0 < \alpha$, hence there is enough evidence to include wind speed into our model.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-5.2033	11.1181	-0.47	0.6435
wind	-0.4271	0.0864	-4.94	0.0000
temperature	0.5204	0.1081	4.81	0.0001
$\mathcal{R}^2 =$	0.777			

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	46.9157	5.6857	8.25	0.0000
wind	-0.6096	0.1097	-5.56	0.0000
humidity	-0.0452	0.0787	-0.57	0.5706
$\mathcal{R}^2 =$	0.5913			

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	32.3262	6.9710	4.64	0.0001
wind	-0.5564	0.0978	-5.69	0.0000
insolation	0.1316	0.0538	2.45	0.0213
$\mathcal{R}^2 =$	0.6613			

Table 2: Step 2, of the step up method.

We now follow the process stated above, but now we include wind speed into our model. Here we obtain the following data for the given models: Now ordering the multiple R-squared values we have; temperature(0.777) > humidity(0.5913) > insolation(0.6613). Moreover we check that the p-values for temperature which we see is $0.0001 < \alpha$. Therefore, there is enough evidence to include temperature in out model.

We repeat the process again, this time we have wind speed and temperature included in the model. Doing so we obtain the following data:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-16.6070	13.0715	-1.27	0.2152
wind	-0.4462	0.0851	-5.24	0.0000
temperature	0.6019	0.1176	5.12	0.0000
humidity	0.0985	0.0632	1.56	0.1310
$\mathcal{R}^2 =$	0.7964			

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-4.4550	11.2671	-0.40	0.6958
wind	-0.4235	0.0874	-4.85	0.0001
temperature	0.4756	0.1256	3.79	0.0008
insolation	0.0365	0.0507	0.72	0.4786
$\mathcal{R}^2 =$	0.7816			

Table 3: Step 3 of the step up method.

However in both cases we see that the p-value humidity(0.1310), insolation(0.4786) is greater than the significance level α and thus we shall not include these in the model.

We can conclude the step up process for our multiple linear model as

$$LM : Y_{\text{oxidant}} := -5.203 - 0.427 \cdot V_W + 0.52 \cdot V_T + e_{\text{error}}$$

Part c. The full linear regression model is give by;

$$LM : Y_{\text{oxidant}} = -15.493 - 0.442 \cdot V_W + 0.569 \cdot V_T + 0.092 \cdot V_H + 0.022 \cdot V_I + e.$$

To analyse whether we should include one or more explanatory variables in the linear regression model let us consider the results of a $\mathcal{F}$-test with the following null and alternative hypotheses

$$H_0 : \ \beta_1 = \beta_2 = \beta_3 = \beta_4 = 0,$$
$$H_1 : \beta_j \neq 0 \text{ for some} \, 0 \leq j \leq 4.$$

For this test the significance level is $\alpha = 0.05$ and the test statistic is defined as

$$\mathcal{F} = \frac{(n-p-1)(SSY - RSS)}{pRSS} = \frac{(30-4-1)(SSY - RSS)}{4 \times RSS}$$

Here RSS is the residual sum of squares and SSY is the sum of squares for Y. Computing our $\mathcal{F} = 24.69$ with the associated p-value $= 2.279 \times 10^{-8} < \alpha$. Thus there is enough evidence to reject the null hypothesis and accept the alternative hypothesis. In other words the test suggests that we should consider one or more explanatory variables for our linear regression model.

Part d. For this task we shall preform a 'step down' method to construct our linear regression models. We shall use the null and alternative hypothesis as described in part **b.**, along with the same test statistic and significance level. We begin with all the explanatory variables added to the model and we compute the summary of the linear model as (Table 4);

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-15.4937	13.5065	-1.15	0.2622
temperature	0.5693	0.1398	4.07	0.0004
wind	-0.4429	0.0868	-5.10	0.0000
humidity	0.0929	0.0653	1.42	0.1674
insolation	0.0228	0.0507	0.45	0.6573

Table 4: Step down, first stage computed summary.

Here we compute the p-values for the models. Furthermore we see that Insolation has the largest p-value (0.6573) and that it is greater than α.Therefore we shall remove Insolation from the model and repeat the process. Doing so leads use to compute the following model summary (Table 5);

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-16.6070	13.0715	-1.27	0.2152
temperature	0.6019	0.1176	5.12	0.0000
wind	-0.4462	0.0851	-5.24	0.0000
humidity	0.0985	0.0632	1.56	0.1310

Table 5: Step down stage 2 computed summary.

Here we see that Humidity has a p-value of 0.1310 which is again greater than α. As such we can remove Humidity from our linear regression model. We then repeat computing the summary for the model as (Table 6);

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-5.2033	11.1181	-0.47	0.6435
temperature	0.5204	0.1081	4.81	0.0001
wind	-0.4271	0.0864	-4.94	0.0000

Table 6: Step down stage 3 computed summary.

In the Table 6 we see that all the explanatory variables have p-values greater than our set α and therefore we shall keep them in our linear regression model. Thus we can conclude that via the step down method we obtain the model

$$LM : -5.203 - 0.427 \cdot V_W + 0.52 \cdot V_T + e_{\text{error}}.$$

Part e. The linear regression model that we construct in part **b.** via the 'Step up' method and in **d.** via the 'Step down' method both arrive to the same linear model. Therefore we can conclude that this is the best linear regression model to use for the given data. These results are not surprising as we made this prediction while analysing the figures and data in part **a.**.

Part f. First we plot the added variable plots for both the Insolation and the Humidity which gives us a visualisation of partial correlation. As we can see from Figure 11 there are no linear trends either upwards or downwards. Hence we can say that there is no strong influence of the explanatory variables in the response variable, namely in the oxidant levels.

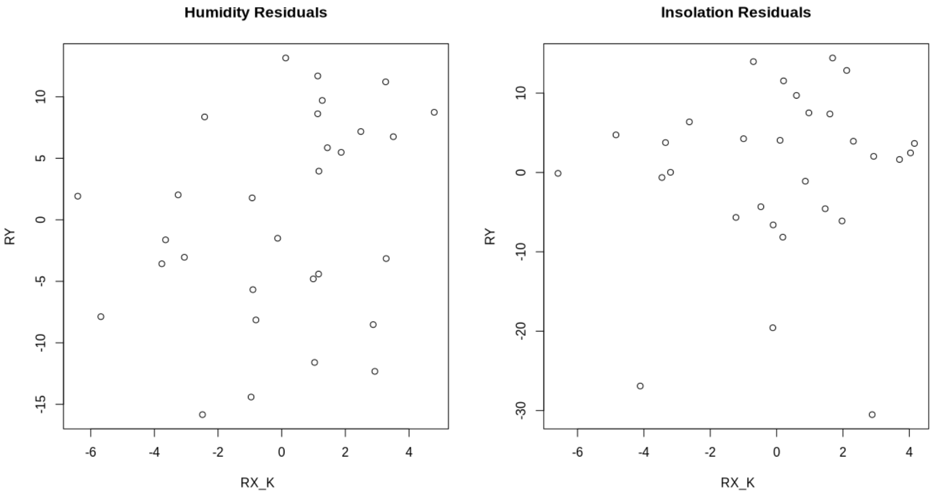

Figure 11

Here we plot the scatter plot of the vector of residuals of the regression against the vector of residuals of the regression of the explanatory variable, for both humidity and Insolation, which are shown in Figure 12. In the Figure 12 we see that there is no linear pattern to the data, thus we can conclude that we should not add Humidity and/or Insolation to the model.

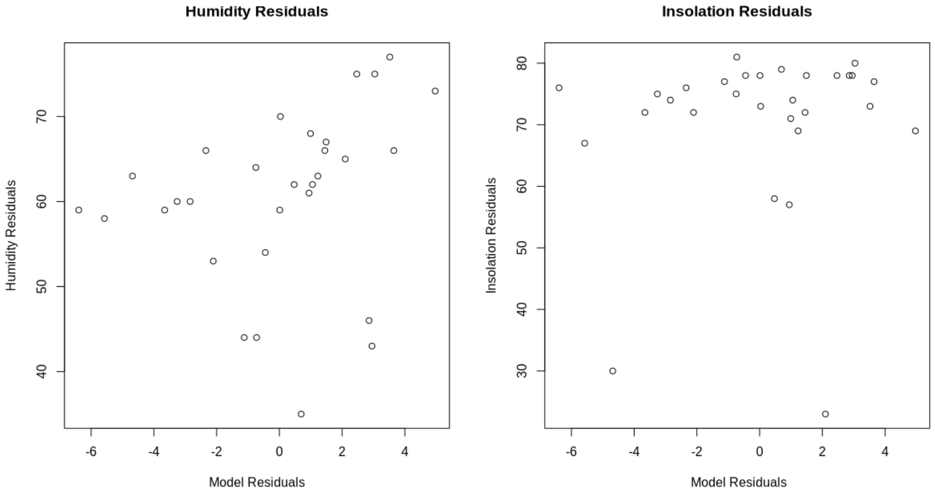

Figure 12

Part g. We set up a t-test with the following null and alternative hypotheses

$$H_0 : \sigma = 0,\ \beta \text{ arbitrary},$$
$$H_1 : \sigma \neq 0,\ \beta \text{ arbitrary}.$$

We set the significance level to $\alpha = 0.05$ and note that the test statistic in this case is

$$t_{p+1} = \frac{\hat{\sigma}}{\sqrt{c\hat{o}v(\hat{\beta})_{p+1,p+1}}} \leq t_{(}n - p - 2)$$

with $p = 2$, we see that $t_{(n-p-2)} = t_{26}$ which yields the following test statistic of 2.036. The associated p-value is computed as 0.052, which we see is larger than α. Therefore, there is not enough evidence to reject the null hypothesis. In other words, the fourth observation is not an outlier to the given data set.

Part h. In order to obtain the leverage points of the given data, we must first compute the hat-matrix, then the leverage points are along the diagonals of this matrix. Once computed we can plot them as seen in Figure 13, where we should note that the closer to 0 in the y-axis, the better. Moreover, we see in Figure 13 that all the points are bound by approximately 0.3, and thus are not close to 1. Numerically we can check this with

$$\frac{2(p+1)}{n} = \frac{6}{30} = 0.2.$$

We see that there are 5 points which are above 0.2, thus we have 5 leverage points.

Now for influence points, this is achieved via computing the Cook's distance for every observations in our data set, we plot the results of this computation in Figure 14. From this analysis, we see that all the point are below 0.4 on the y-axis. Thus we can conclude that none of the observations are influence points in our data set.

In order to conduct analysis of collinearity, we must first compute the correlation between the explanatory variables in our data set. Doing this yields the Table 7. We observe that

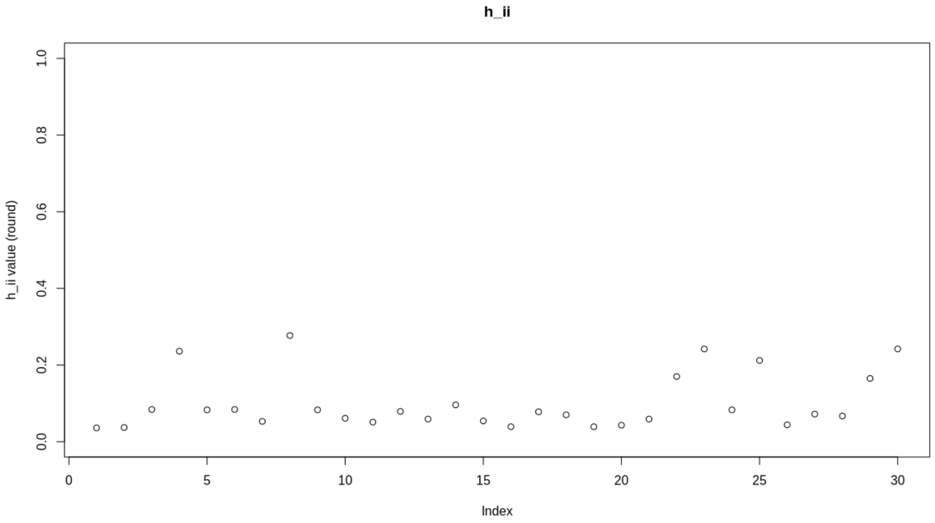

Figure 13: Plot of the h$_{ii}values$.

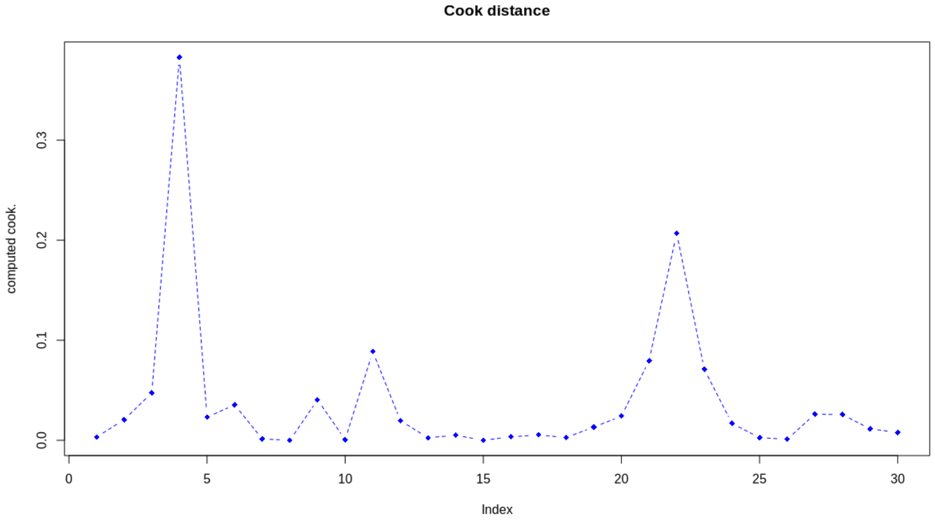

Figure 14: Cook distance values.

	wind	temperature	humidity	insolation	oxidant
wind	1.00	-0.50	0.37	-0.32	-0.77
temperature	-0.50	1.00	-0.54	0.57	0.76
humidity	0.37	-0.54	1.00	-0.18	-0.35
insolation	-0.32	0.57	-0.18	1.00	0.51
oxidant	-0.77	0.76	-0.35	0.51	1.00

Table 7: Computed correlation between the explanatory variables.

the correlation between the temperature and the wind speed is -0.5, however this does not yield a lot of information, thus we compute the variance inflation factors, which are $\{1.36, 2.26, 1.5, 1.53\}$. As we can see the values range between $(1, 3)$, since these are small values there is no evidence to distrust the estimates of β for our model.

The last step is to compute and analyse the variance decomposition matrix which is given in Table 8.

	conditionindices	0	1	2	3	4
1	1.00	0.00	0.00	0.00	0.00	0.00
2	9.62	0.00	0.04	0.00	0.14	0.24
3	17.07	0.00	0.09	0.09	0.33	0.39
4	22.63	0.00	0.61	0.11	0.20	0.34
5	3317.32	1.00	0.26	0.80	0.33	0.03

Table 8: Variance decomposition matrix.

Part i. To check for normality in our linear regression model, we plot the computed residues of the model against the thoeritical values of a normal distribution, this is achieved by using an qqplot, the results can be found in Figure 15. As we can see from Figure 15 we have good straight line which indicates our data comes from a normal distribution. This confirms our assumptions.

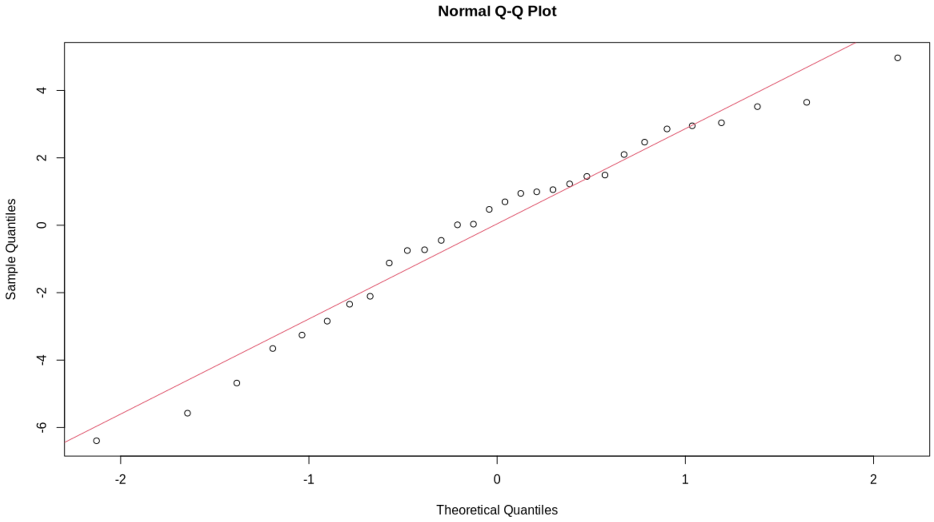

Figure 15: QQ-plot to check for normality.

Part j. Here we finally conclude on the linear regression model for the pollen data is

$$LM : -5.203 - 0.427 \cdot V_W + 0.52 \cdot V_T + e_{\text{error}}.$$

Our estimated values for $\beta_0 = -5.203$, then for wind speed $\beta_1 = -0.427$ and temperature $\beta_2 = 0.52$. We have see that the estimated variance of the error is computes as 2.95 and we have an determination coefficient of 0.777. We can conclude this model is a good approximation for the data set. Moreover we see that we arrive at the same model from both the step up method and also the step down method. We have done analysis to check for outlier, leverage and influence points and collinearity. Therefore we have good evidence that this linear regression model is suitable for the data set.

Question 7.3.

Part b.

In this exercise we tried to find a multiple linear regression model between the given variables. Much of the data seemed unlikely to be related, and we sought out a reasonable model by iterating over the variables and computing the R^2 values for each pair. As the objective was to find a multiple linear model, we looked for a variable that had a relatively high R^2 value for at least two other variables. We found this to be the case for Glycerine, which had an R^2 value of 0.8904632 with the variable Fatty Acid, and an R^2 value of 0.5843941 with the variable Operating Days. Though 0.58 is not a very high R^2 value, it is better than the majority of other R^2 values between the other pairs, and Glycerine was the only variable to have more than one pairing with another variable that had an R^2 above 0.5. Overall, much of the data we had was uncorrelated, save for a few pairs.

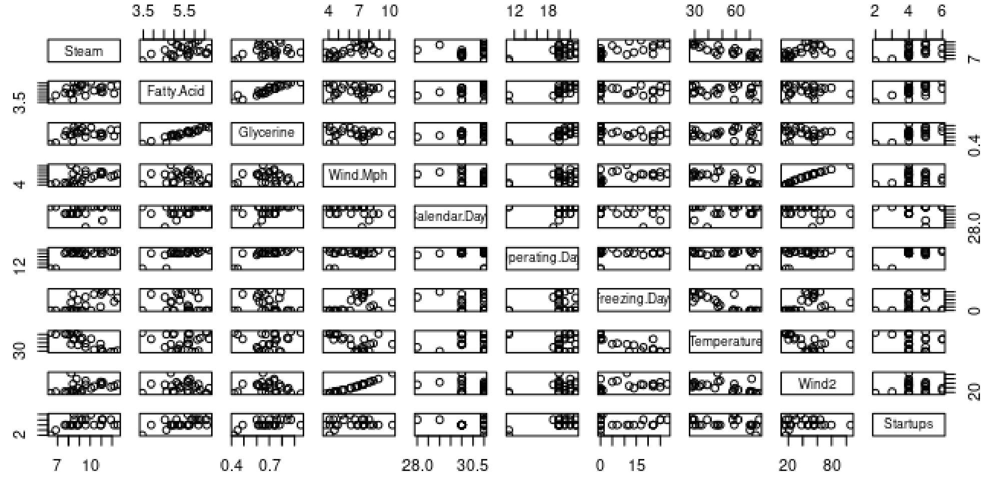

Figure 16: A table to paired plots between different variables.

As in earlier sub-questions, we set up our test hypotheses as $H_0 : \beta = 0$ against $H_a : \beta \neq 0$, with significance level $\alpha = 0.05$, and perform the `lm()` function on the two pairs, namely Glycerine and Fatty Acid, and Glycerine and Operating Days.

In the case of the Fatty Acid, we return a p-value of 1.56 $\times 10^{-12}$, which allows us to reject the null hypothesis that the two variables are not linearly related.

In the case of Operating Days, we return a p-value of 8.63 $\times 10^{-06}$, which again allows us to reject the null hypothesis.

We also checked the relation between the two dependent variables, namely Fatty Acid and Operating Days, and found a striking relation between the two, with an R^2 value of 0.4693514 and a p-value of 0.000158 which allows us to reject the null hypothesis for these two variables.

Combining the two dependent variables in our model for Glycerine gives us the linear model

$$Glycerine = -0.156023 + 0.121921 \times FattyAcid + 0.009273 \times OperatingDays + \varepsilon$$

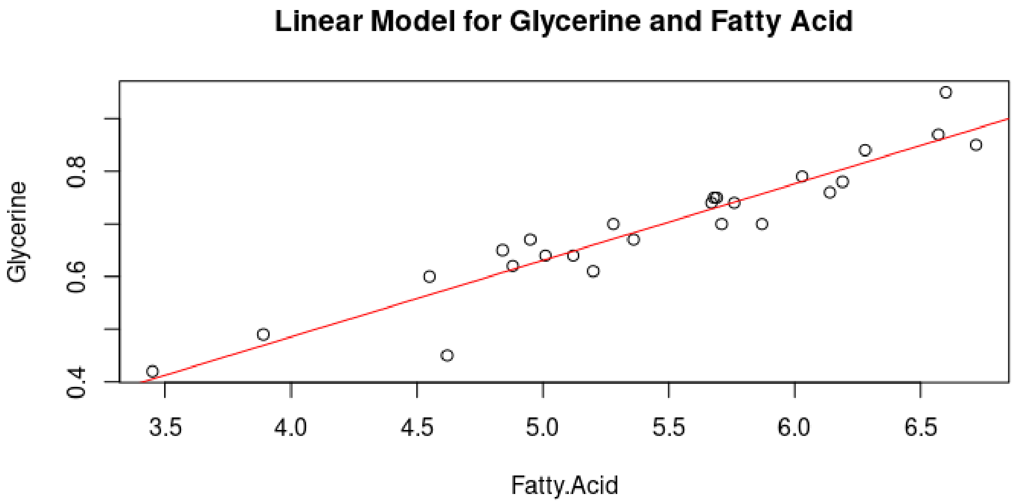

Figure 17: Linear model of Glycerine related to Fatty Acid.

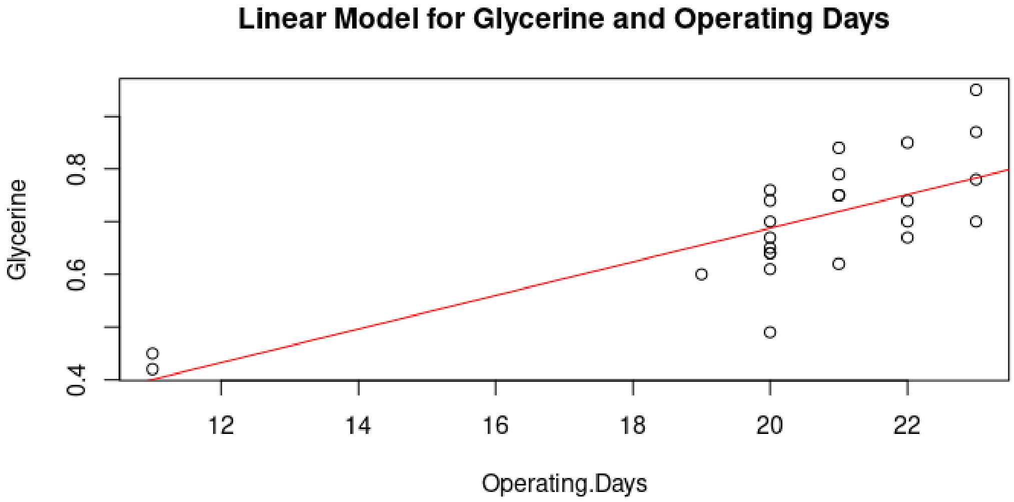

Figure 18: Linear model of Glycerine related to Operating Days.

where $\varepsilon \sim N(0, \sigma^2)$ with the full table of values given by the summary in RStudio as

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.156023	0.056915	-2.741	0.0119
Fatty.Acid	0.121921	0.013015	9.368	3.91×10^{-9}
Operating.Days	0.009273	0.003523	2.632	0.0152

Part c.

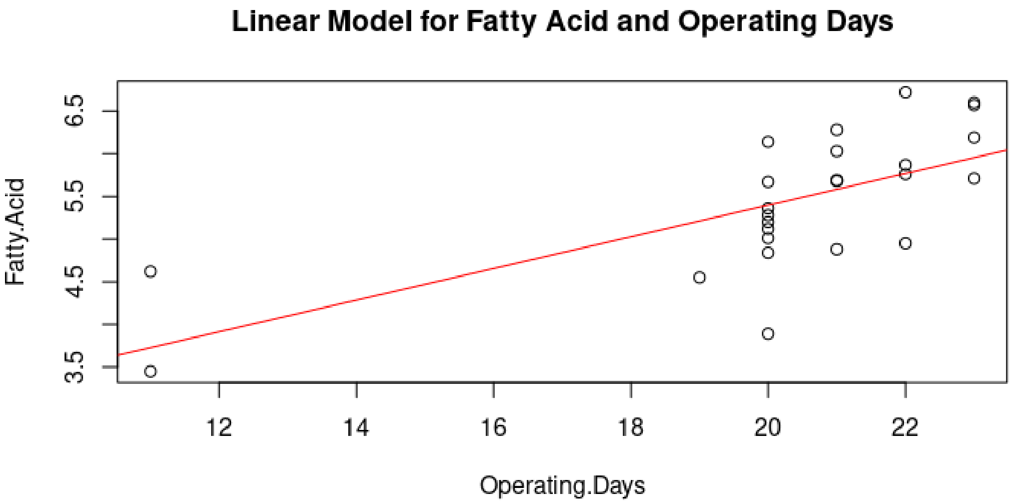

Figure 19: Linear model of Fatty Acid related to Operating Days.

We now turn to investigating whether the data is in truth only heavily swayed by a few data points that overlap between the data sets. We first inspect this visually by observing the data plots of the three variables.

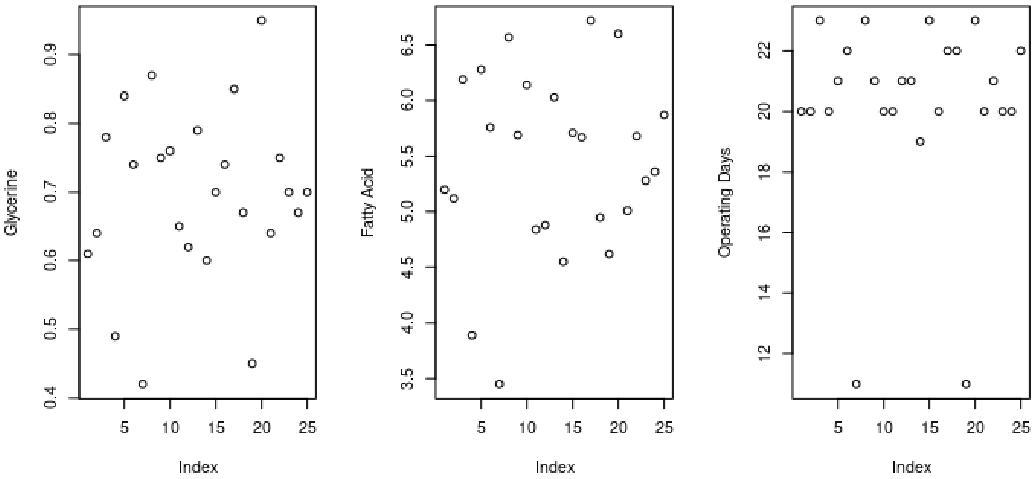

Figure 20: Data plots for Glycerine, Fatty Acid, and Operating Days.

We note that the two lowest data points in the Operating Days data are distinctly noticeable as points of similarity between its plot with that of Glycerine. If we remove these two points (and to keep the data sets equal in size, remove the two lowest points of the other data sets), we observe that the linear relation between Glycerine and Operating days decreases, with an R^2 value of only around 0.396, and a p-value of 0.001296, a far less certain rejection of the null hypothesis than the one we had earlier.

Similarly, the relation between the adjusted data of Glycerine and Fatty Acid decreases rapidly, with the linear model having an R^2 of only 0.01852, and a p-value of 0.5358, which means that we do not reject the null hypothesis, and the linear relation between these two variables is put in high doubt. This is a rather remarkable change, as these two variables showed a great amount of linearity in the initial pair plot.

Ironically, the adjusted data of Fatty Acids and Operating Days gives us an R^2 value of 0.07073 with a p-value of 0.22. These two data sets seemed highly unrelated, but the removal of the two smallest data points has somehow made these variables more linear to each other than the adjusted data of Glycerine and Fatty Acid, even though we still cannot reject the null hypothesis that $\beta = 0$.

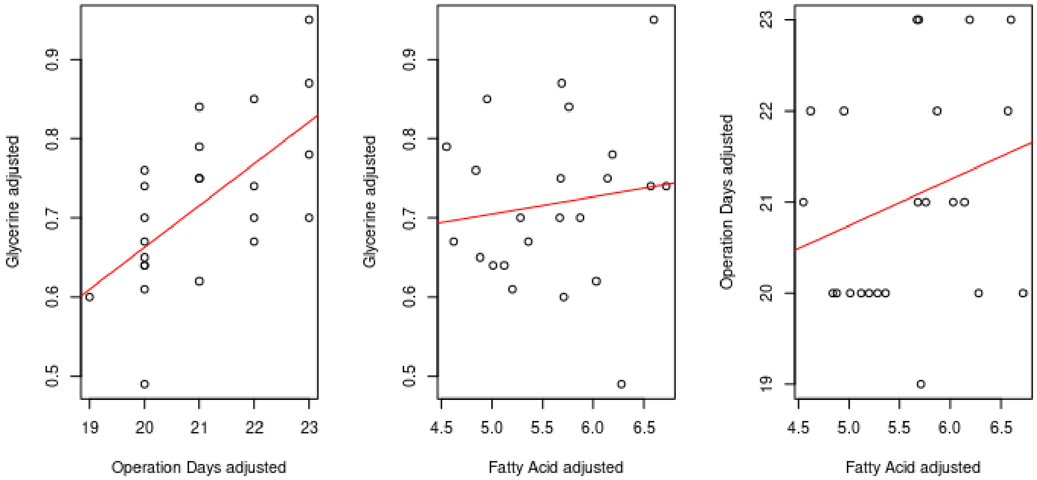

Figure 21: Adjusted pairings with the smallest two values removed for each data set.

Part d.

For this question, we perform the same test and operations as we did in **7.1 d.**. We use the *residuals* function built into `lm()`, and generate a test statistic against which we compare the values we obtain by performing `lm.norm.test()`.

Just to compare the results of the two tests, we also performed a Shapiro-Wilk test on the residuals, which resolves with a p-value of 0.4212 in support of normality, far larger than the standard significance value required of $\alpha = 0.05$. Our own p-value, computed in the same way as it was computed in **7.1 d.** is 0.549, indicating that the residuals strongly seem to be distributed normally.

Part e.

Considering the observations made in **c.**, and the overall low levels of linear correlation between the data sets, other than a few selective pairs, it seems that assuming, or rather projecting, any kind of linearity between certain ”highly correlated” variables revolves around a few data points shifting the values in favour of linearity, but aside from those points the rest of the data hardly manages to construct a multiple linear regression model. Maybe we

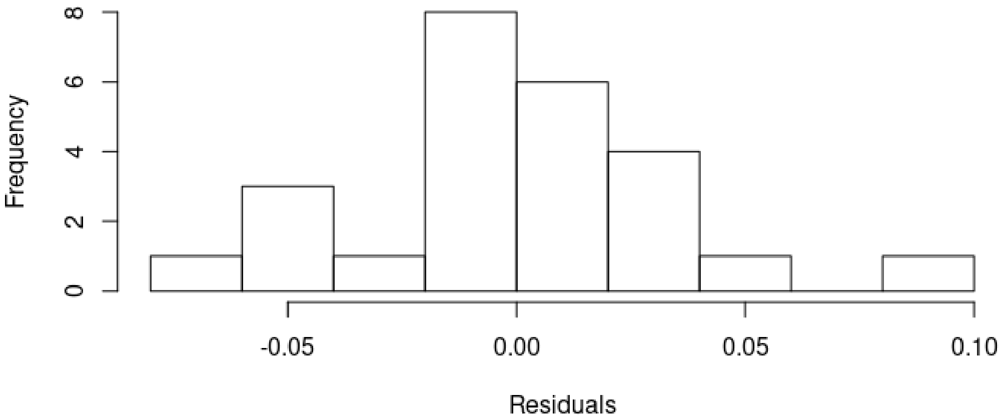

Figure 22: Histogram of residuals for the multiple linear regression model.

selected an unfortunate set of variables to perform this exercise on, but we can certainly conclude that lending too much credit to R^2 values can lead to far reaching assumptions about linearity and correlation.

R code

```
# Assigment 7 - Main code
# Group number: 36
# Authors: Matt and James
# Date : 13/05/22


################################################################################
# Question : 7.1
# set working dir
setwd("~/Desktop/P4_and_P5_2022/SDA_Assignments/SDA7") #Matt
source("functions_Ch8.txt")

#read the data
geese_data <- read.table("geese.txt", header = TRUE)

#part b:

#make data vectors for quick referencing
photo_data = geese_data$photo
observer1 = geese_data$observer1
observer2 = geese_data$observer.2

#compute the linear models for the pairs and the total
geese_lm = lm(photo ~ observer1 + observer.2, data=geese_data)
geese_lm1 = lm(photo ~ observer1, data=geese_data)
geese_lm2 = lm(photo ~ observer.2, data=geese_data)

#plot said linear models
par(mfcol=c(1,2))
plot(observer1, photo_data, ylab = 'Photo Data', xlab = 'Observer 1 Data');
abline(geese_lm1$coefficients[1], geese_lm1$coefficients[2], col='red');
plot(observer2, photo_data, ylab = 'Photo Data', xlab = 'Observer 2 Data');
abline(geese_lm2$coefficients[1], geese_lm2$coefficients[2], col='red')

mtext("Linear Model Graph of Photo Data and Observer 1, and Photo Data and
    Observer 2",                   # Add main title
      side=3,
      line=-3,
      outer=TRUE)

#quick summary of both linear models
summary(geese_lm1);
summary(geese_lm2)

#part c:

#QQ-plots of data
par(mfrow=c(1,3))
plot(photo_data, ylab = 'Photo Data');
plot(observer1, ylab = 'Observer 1 data');
plot(observer2, ylab = 'Observer 2 data'):

#histograms of data
par(mfrow=c(1,3))
hist(photo_data, main='', xlab='Photo Data');
hist(observer1, breaks=10, main='', xlab='Observer 1 data');
hist(observer2, main='', xlab='Observer 2 data')
mtext("Histograms of Photo Data, Observer 1, and Observer 2",
                        # Add main title
      side=3,
      line=-3,
      outer=TRUE)
```

```
#plots of photo data vs observer1 and observer2 and the line y=x
par(mfrow=c(1, 2))
plot(observer1, photo_data, ylab='Photo Data', xlab='Observer 1 data');
abline(0,1, col='red')
plot(observer2, photo_data, ylab='Photo Data', xlab='Observer 2 data');
abline(0,1, col='red')
mtext("Photo Data against Observer 1, and Observer 2",                  # Add
     main title
       side=3,
       line=-3,
       outer=TRUE)

#QQ plot of residuals
par(mfrow=c(1,2))
plot(geese_lm1$residuals, ylab = 'Residuals Observer 1 data');
abline(0,0, col='2');
plot(geese_lm2$residuals, ylab = 'Residuals Observer 2 data');
abline(0,0, col='2');

hist(geese_lm1$residuals, xlab="Residuals Observer 1",
     ylab="Frequency", main=" ", breaks=15)
hist(geese_lm2$residuals, xlab="Residuals Observer 2",
     ylab="Frequency", main=" ", breaks=15)


#part d:

#generate residuals and their mean + sd
residuals_obs1 = as.numeric(geese_lm1$residuals)
res_obs1_sd = sd(residuals_obs1)
res_obs1_mean = mean(residuals_obs1)

residuals_obs2 = as.numeric(geese_lm2$residuals)
res_obs2_sd = sd(residuals_obs2)
res_obs2_mean = mean(residuals_obs2)

#create a test statistic to compare residuals to
#parametric normal distribution under same mean and sd
#This is not overfitted since ks test checks the
#empirical cumulative distribution compared to said
#parametric normal sample, i.e. we want to see what
#statistic we get comparing our residuals to
#a normal distribution


test_stat_obs1 = ks.test(residuals_obs1, 'pnorm',
                         mean = res_obs1_mean,
                         sd = res_obs1_sd)

test_stat_obs2 = ks.test(residuals_obs2, 'pnorm',
                         mean = res_obs2_mean,
                         sd = res_obs2_sd)

#here we generate a lot of ks test values. The function does this by
#generating a random normal sample with the parameters of our residuals
#and then performing the ks test on the residuals of that data against
#a parametric normal distribution. If the test values are
#similar to the ones we computed earlier, then our residuals are
#very likely to be normal

res_norm_obs1 = lm.norm.test(photo_data, observer1, B=1000)
res_norm_obs2 = lm.norm.test(photo_data, observer2, B=1000)
```

```
#p value is in this case the probability of observing a ks test value larger
#than our ks test value. If more than 95% of the bootstrapped ks values are
#below our test statistic, then our residuals deviate significantly
#(at alpha = 0.05) from error terms which would be truly normal

p_val_obs1 = length(res_norm_obs1[res_norm_obs1>test_stat_obs1$statistic])/
    1000
p_val_obs2 = length(res_norm_obs2[res_norm_obs2>test_stat_obs2$statistic])/
    1000

#both p-values at 0 means that the ks test statistic for our residuals
    compared
#to a normal parametric distribution is larger than any ks statistic value
    that
#is found when comparing actually normal residuals against the normal
    parametric
#distribution. Thus the residuals are not normal!

#part e:

#We take the log of the data sets
log_geese_data = log(geese_data)
log_photo_data = log_geese_data$photo
log_observer1 = log_geese_data$observer1
log_observer2 = log_geese_data$observer.2

#compute the residuals of the log transformed data

lm_log_obs1 = lm(log_photo_data ~ log_observer1, data = log_geese_data)
log_res_obs1 = lm_log_obs1$residuals

lm_log_obs2 = lm(log_photo_data ~ log_observer2, data = log_geese_data)
log_res_obs2 = lm_log_obs2$residuals

#all the plots from b-c

summary(lm_log_obs1)
summary(lm_log_obs2)

hist(log_res_obs1, xlab="log(Observer 1 data)", main=" ")
hist(log_res_obs2, xlab="log(Observer 1 data)", main=" ")
mtext("Histograms of log(data) residuals",
      side=3,
      line=-3,
      outer=TRUE)

par(mfrow=c(1,3))
plot(log_photo_data, ylab='log(Photo Data)');
plot(log_observer1, ylab='log(Observer 1 data)');
plot(log_observer2, ylab='log(Observer 2 data)'):

  par(mfrow=c(1,3))
hist(log_photo_data, main='', xlab='log(Photo Data)');
hist(log_observer1, breaks=10, main='', xlab='log(Observer 1 data)');
hist(log_observer2, main='', xlab='log(Observer 2 data)')
mtext("Histograms of Photo Data, Observer 1, and Observer 2, after log
    transform",
      side=3,
      line=-3,
      outer=TRUE)

par(mfrow=c(1, 2))
```

```
plot(log_observer1, log_photo_data, ylab='log(Photo Data)', xlab='log(Observer
     1 data)');
abline(0,1, col='red')
plot(log_observer2, log_photo_data, ylab='log(Photo Data)', xlab='log(Observer
     2 data)');
abline(0,1, col='red')
mtext("log(Photo Data) against log(Observer 1 data), and log(Observer 2 data)"
    ,                     # Add main title
      side=3,
      line=-3,
      outer=TRUE)

log_geese_lm1 = lm(log_photo_data ~ log_observer1, data=log_geese_data)
log_geese_lm2 = lm(log_photo_data ~ log_observer2, data=log_geese_data)

par(mfrow=c(1,2))
plot(log_geese_lm1$residuals, ylab = 'Residuals log(Observer 1 data)');
abline(0,0, col='2');
plot(log_geese_lm2$residuals, ylab = 'Residuals log(Observer 2 data)');
abline(0,0, col='2');

#now we do the tests as in d.
#first get the sd and the mean of the log_residuals
log_res_obs1_sd = sd(log_res_obs1)
log_res_obs1_mean = mean(log_res_obs1)

log_res_obs2_sd = sd(log_res_obs2)
log_res_obs2_mean = mean(log_res_obs2)

#create the two test statistics
log_test_stat_obs1 = ks.test(log_res_obs1, 'pnorm',
                             mean = log_res_obs1_mean,
                             sd = log_res_obs1_sd)

log_test_stat_obs2 = ks.test(log_res_obs2, 'pnorm',
                             mean = log_res_obs2_mean,
                             sd = log_res_obs2_sd)

#perform the bootstrap

log_res_norm_obs1 = lm.norm.test(log_photo_data, log_observer1, B=1000)
log_res_norm_obs2 = lm.norm.test(log_photo_data, log_observer2, B=1000)

#compute the p-values

log_p_val_obs1 = length(log_res_norm_obs1[log_res_norm_obs1>log_test_stat_obs1
    $statistic])/1000
log_p_val_obs2 = length(log_res_norm_obs2[log_res_norm_obs2>log_test_stat_obs2
    $statistic])/1000

#part f:

#log makes the data more compact, outliers become normalized, can be preferred
     if
#assumption that larger errors in counting large flocks of geese are to be
#expected

#part g:

#photo count reflects both observers very well for the initial values, but for
#larger flocks the observer estimates start to deviate. Observer 2 made
    overall
#the most accurate observations, and the photo count reflects their
```

```
    observation
#the best overal.
```

code/sda_hw7_ex1.R

```
# Assigment 7 - Main code
# Group number: 36
# Authors: Matt and James
# Date : 13/05/22


##############################################################################
# Question : 7.1
# set working dir
setwd("~/Maths/SDA/Assignment_7")
# part a.

geese_data <- read.table("geese.txt", header = TRUE)

par(mfrow=c(1,3))
plot(geese_data$photo); plot(geese_data$observer1);plot(geese_data$observer.2)
hist(geese_data$photo); abline(); hist(geese_data$observer1); hist(geese_data$
    observer.2)
par(mfrow=c(1,2))
plot(geese_data$observer1,geese_data$photo);abline(0,1, col='red');plot(geese_
    data$observer.2,geese_data$photo);abline(0,1, col='red')

# In both case we see that there is evdence for a postive linear model.
    However, we should note as the x-value is increasesed we see
# a larger dievation (eucliean distance)from the linear line (in red)
    increases. However, most of the data points occur around the orgin
# and we can see the deviation at the tails only effects thte data in a small
    amount.


# part b
is.data.frame(geese_data)
geese_lm = lm( photo ~ observer1 + observer.2, data=geese_data)
lm1 = lm(photo ~ observer1, data=geese_data)
lm2 = lm(photo ~ observer.2, data=geese_data)
plot(geese_data$observer1,geese_data$photo);abline(lm1$coefficients[1],lm1$
    coefficients[2], col='red');plot(geese_data$observer.2,geese_data$photo);
    abline(lm1$coefficients[1],lm1$coefficients[2], col='red')
summary(lm1);summary(lm2)


#part c
plot(lm1$residuals, main = 'Residuals geese observer 1');abline(0,0, col='2');
    plot(lm2$residuals, main='Residuals geese observer 2');abline(0,0, col='2')

# part d
source('functions_Ch8.txt')
B=1000
res_lm1 = as.numeric(lm1$residuals)
D_ts = numeric(B)
d_norm = lm.norm.test(geese_data$photo, geese_data$observer1, B=B)
sd_res = sd(res_lm1)
for(i in 1:B){
  test_norm = rnorm(B, mean=0, sd=sd(res_lm1))
  D_ts[i] = ks.test(res_lm1, test_norm, alt='t',)$statistic
}
test_D = mean(D_ts)
p_val = length(d_norm[d_norm<test_D])/B
```

```
p_val



#
   ################################################################################

# question 2
polls <- read.table("airpollution.txt")
poll = data.frame(polls)

#a
pairs(polls)
is.data.frame(polls)
# linear correralation between `temperature & oxidant` (postive)
# negative correraltion `wind & oxidant`
# humidity no correraltion
# insolation no correraltion

poll_lm1 = lm(oxidant ~ temperature+ wind + humidity + insolation, data=polls)
summary(poll_lm1)
poll_lm2 = lm(oxidant ~ temperature + wind, data=polls )
poll_lm3 = lm(oxidant ~ temperature, data=polls )
poll_lm4 = lm(oxidant ~ wind, data=polls )
mlist = list(poll_lm1,poll_lm2, poll_lm3, poll_lm4)
for( x in mlist){ print(summary(x))}

main_lm = lm(oxidant ~ wind +temperature, data=polls )

library(xtable)
options(xtable.floating=FALSE)
options(xtable.timestamp = '')

#part b
 #step 1
lm_b_w <- lm(oxidant ~ wind, data=polls)
lm_b_t <- lm(oxidant ~ temperature, data=polls )
lm_b_h <- lm(oxidant ~ humidity, data=polls )
lm_b_i <- lm(oxidant ~ insolation, data=polls )
b1_list= list(lm_b_w, lm_b_t,lm_b_h, lm_b_i)
for( x in b1_list ){ print(summary(x))}
# include wind into the model.
# step 2
lm_b_w_t <- lm(oxidant ~ wind+ temperature, data=polls )
lm_b_w_h <- lm(oxidant ~ wind + humidity, data=polls )
lm_b_w_i <- lm(oxidant ~ wind + insolation, data=polls )
b2_list= list(lm_b_w_t,lm_b_w_h, lm_b_w_i)
for( x in b2_list ){ print(summary(x))}
for( x in b2_list ){ print(xtable(summary(x)))}
# include wind + temperature
# step 3
lm_b_w_t_h <- lm(oxidant ~ wind + temperature + humidity, data=polls )
lm_b_w_t_i <- lm(oxidant ~ wind + temperature + insolation, data=polls )
b3_list= list(lm_b_w_t_h, lm_b_w_t_i)
for( x in b3_list ){ print(summary(x))}
for( x in b3_list ){ print(xtable(summary(x)))}
# p-values in both cases are larger than \alpha, so we dont add them

# part c
full_lm <- lm(oxidant ~ temperature+ wind + humidity + insolation, data=polls)

#part d
```

```
summary(full_lm)
xtable(summary(full_lm))
# insolation Pr>\alpha, thus we remove.
d1_lm <- lm(oxidant ~ temperature+ wind + humidity, data=polls)
summary(d1_lm)
xtable(summary(d1_lm))
# humidity Pr > \alpha, thus remove.
d2_lm <- lm(oxidant ~ temperature+ wind, data=polls)
summary(d2_lm)
xtable(summary(d2_lm))
# all variables are less than \alpha, thus we finised.


# part e
# no code.

# part f Diagnostics
RY_hum <- lm(oxidant ~ wind + temperature + insolation, data=polls)
RX_K_hum <- lm(humidity ~ wind + temperature + insolation, data=polls)
par(mfrow=c(2,2))
par(pty='s')
plot(RY_hum$residuals, RX_K_hum$residuals)

RY_ins <- lm(oxidant ~ wind + temperature + humidity, data=polls)
RX_K_ins <- lm(insolation ~ wind + temperature + humidity, data=polls)
plot(RY_ins$residuals, RX_K_ins$residuals)


plot(main_lm$residuals, polls$humidity)
plot(main_lm$residuals, polls$insolation)
source('functions_Ch8.txt')
source('functions_Ch3.txt')



# part g.
fourth <- c( rep(0,3), 1 , rep(0,26))
model_g= lm(oxidant ~ wind +temperature + fourth, data=polls)
summary(model_g)
# see that the fourth t-value = 2.036 with p= 0.052 > \alpha


#part h.
model_h = lm(oxidant ~ wind +temperature , data=polls)
hat_rnd = round(hatvalues(model_h),3)
par(mfrow=c(1,1))
plot(hat_rnd, ylim=c(0,1))

cook_rnd = cooks.distance(model_h)
plot(cook_rnd, pch = 18, col = "blue", type = "b", lty = 2)

cor_table =round(cor(polls[,2:6]),2 )
var_inf_table = varianceinflation(polls[,2:5])
var_dec_table=round(vardecomposition(polls[,2:5]),3)
xtable(cor_table)
xtable(var_inf_table)
xtable(var_dec_table)

# We conclude that there
# might be a small collinearity among the explanatory variables,
# but as it is not very strong it

#part i - QQplot
```

```
qqnorm(model_h$residuals)
qqline(model_h$residuals, col='2')
```

code/sda_hw7_main_code.R

```
# Assigment 7 - Main code
# Group number: 36
# Authors: Matt and James
# Date : 13/05/22


###############################################################################
# Question : 7.3
# set working dir
setwd("~/Desktop/P4_and_P5_2022/SDA_Assignments/SDA7") #Matt
source("functions_Ch8.txt")

#part a:

steam = read.table("steamtable.txt", header=TRUE)
pairs(steam)

t(round(cor(steam[,1], steam[,-1]),3))
attach(steam)
summary(lm(Steam~Fatty.Acid))$r.squared
#summaries Glycerine, Wind.Mph, Calendar.Days, Operating.Days, Freezing.Days,
    Temperature, Wind2, Startups
#The closer to 1, the better
summary(lm(Steam~Temperature))$r.squared
cor(Steam, Temperature)^2

###

variable_list = list(Steam, Fatty.Acid, Glycerine, Wind.Mph, Calendar.Days,
                     Operating.Days, Freezing.Days, Temperature, Wind2,
                     Startups)

for (variable in variable_list) {


  for (variable2 in variable_list) {

    print(summary(lm(variable~variable2))$r.squared)

  }

}

#manually scan through to find that Glycerine has a nice r^2 value for Fatty.
    Acid
#and Operating Days.

lm_glys_fat = lm(Glycerine~Fatty.Acid)
summary(lm_glys_fat)
summary(lm(Glycerine~Fatty.Acid))$r.squared

par(mfcol=c(1,1))
plot(Fatty.Acid, Glycerine, main="Linear Model for Glycerine and Fatty Acid");
abline(lm_glys_fat$coefficients[1], lm_glys_fat$coefficients[2], col="red")

lm_glys_oper = lm(Glycerine~Operating.Days)
summary(lm_glys_oper)
summary(lm(Glycerine~Operating.Days))$r.squared
```

```
plot(Operating.Days, Glycerine, main="Linear Model for Glycerine and Operating
     Days");
abline(lm_glys_oper$coefficients[1], lm_glys_oper$coefficients[2], col="red")

#check for a relationship between fatty acid and operating days
lm_fat_oper = lm(Fatty.Acid~Operating.Days)
summary(lm_fat_oper)
summary(lm(Fatty.Acid ~ Operating.Days))$r.squared

plot(Operating.Days, Fatty.Acid, main="Linear Model for Fatty Acid and
    Operating Days");
abline(lm_fat_oper$coefficients[1], lm_fat_oper$coefficients[2], col="red")

lm_glycerine = lm(Glycerine ~ Fatty.Acid + Operating.Days)
summary(lm_glycerine)

#part c:

#Note that there are a few data points (outliers) in which a lot of the
#similarity lies

par(mfcol=c(1,3))
plot(Glycerine, ylab="Glycerine")
plot(Fatty.Acid, ylab="Fatty Acid")
plot(Operating.Days, ylab="Operating Days")

#remove the two lowest data points
par(mfcol=c(1,1))
op_days_adjusted = Operating.Days[Operating.Days>12]
plot(op_days_adjusted)

length(op_days_adjusted)
length(Glycerine)

#match the length of the data sets

glys_adjusted_initial = Glycerine[Glycerine>min(Glycerine)]
glys_adjusted = glys_adjusted_initial[glys_adjusted_initial>min(glys_adjusted_
    initial)]

fat_adjusted_initial = Fatty.Acid[Fatty.Acid>min(Fatty.Acid)]
fat_adjusted = fat_adjusted_initial[fat_adjusted_initial>min(fat_adjusted_
    initial)]

length(glys_adjusted)
length(fat_adjusted)

summary(lm(glys_adjusted~op_days_adjusted))
plot(op_days_adjusted, glys_adjusted, ylab="Glycerine adjusted",
     xlab="Operation Days adjusted")
abline(lm(glys_adjusted~op_days_adjusted)$coefficients[1],
       lm(glys_adjusted~op_days_adjusted)$coefficients[2], col="red")


#verify if something similar is happening between glys and fat
summary(lm(glys_adjusted~fat_adjusted))
plot(fat_adjusted, glys_adjusted, ylab="Glycerine adjusted",
     xlab="Fatty Acid adjusted")
abline(lm(glys_adjusted~fat_adjusted)$coefficients[1],
       lm(glys_adjusted~fat_adjusted)$coefficients[2], col="red")

summary(lm(op_days_adjusted~fat_adjusted))
```

```
plot(fat_adjusted, op_days_adjusted, ylab="Operation Days adjusted",
     xlab="Fatty Acid adjusted")
abline(lm(op_days_adjusted~fat_adjusted)$coefficients[1],
       lm(op_days_adjusted~fat_adjusted)$coefficients[2], col="red")

lm_glycerine_adjusted = lm(glys_adjusted ~ fat_adjusted + op_days_adjusted)
summary(lm_glycerine_adjusted)

#Those two points account for a lot of the similarity. removing them has
#plummeted the R^2 values, and generally destabilized the model by a lot.

#part d:
par(mfcol=c(1,1))
hist(lm_glycerine$residuals, main=" ", xlab="Residuals")
shapiro.test(lm_glycerine$residuals)
#should be normal, by the shapiro-wilk test

residual_sd = sd(lm_glycerine$residuals)
residual_mean = mean(lm_glycerine$residuals)

test_stat_residuals = ks.test(lm_glycerine$residuals, 'pnorm',
                         mean = residual_mean,
                         sd = residual_sd)

res_norm = lm.norm.test(Glycerine, Fatty.Acid + Operating.Days, B=1000)

p_value = length(res_norm[res_norm>test_stat_residuals$statistic])/1000
#supports normality, as assumed earlier with the shapiro-wilk


#part e:

#conclusive remarks and observations
```

code/sda_hw7_ex3.R