Aidan Ashrafi, Valerie Passanisi, Trystan Ng,
Sam Scherkenbach, and Joel Brown

I. Background

The World Happiness Report reviews the state of happiness in the world. The data was collected from a nationally representative sample of 3,000 data points from a survey from the Gallup World Poll. Information from the World Development Indicators and the World Health Organization was also used. We want to create a model to understand what factors influence a nation's happiness.

II. Statistical analyses

II.a. Initial Data Analysis

For each country, there is a happiness score (called life_ladder). There are 8 numerical feature variables: log_GDP_per_capita, social_support, healthy_life_expectancy_at_birth, freedom_to_make_life_choices, generosity, perceptions_of_corruption, positive_affect, and negative_affect.

```
##       country year life_ladder log_GDP_per_capita social_support
## 1 Afghanistan 2018       2.694              7.631          0.508
## 2     Albania 2018       5.004              9.497          0.684
## 3     Algeria 2018       5.043              9.370          0.799
## 4   Argentina 2018       5.793             10.032          0.900
## 5     Armenia 2018       5.062              9.490          0.814
## 6   Australia 2018       7.177             10.801          0.940
##   healthy_life_expectancy_at_birth freedom_to_make_life_choices generosity
## 1                           53.575                         0.374     -0.091
## 2                           69.075                         0.824      0.007
## 3                           66.300                         0.583     -0.151
## 4                           67.050                         0.846     -0.214
## 5                           66.825                         0.808     -0.169
## 6                           70.825                         0.916      0.143
##   perceptions_of_corruption positive_affect negative_affect
## 1                     0.928           0.385           0.405
## 2                     0.899           0.592           0.319
## 3                     0.759           0.534           0.293
## 4                     0.855           0.732           0.321
## 5                     0.677           0.535           0.455
## 6                     0.405           0.706           0.187
```

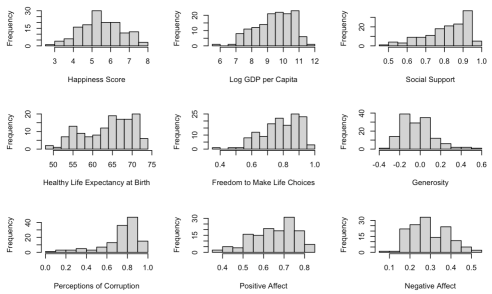
(1)

There is nothing unusual from the summary. There are some missing values, but these do not make up a significant amount of the data.

```
   country              year        life_ladder    log_GDP_per_capita social_support   healthy_life_expectancy_at_birth
 Length:141         Min.   :2018   Min.   :2.694   Min.   : 5.935     Min.   :0.4850   Min.   :49.30
 Class :character   1st Qu.:2018   1st Qu.:4.769   1st Qu.: 8.540     1st Qu.:0.7400   1st Qu.:59.20
 Mode  :character   Median :2018   Median :5.465   Median : 9.552     Median :0.8410   Median :65.21
                    Mean   :2018   Mean   :5.499   Mean   : 9.391     Mean   :0.8122   Mean   :63.89
                    3rd Qu.:2018   3rd Qu.:6.249   3rd Qu.:10.346     3rd Qu.:0.9090   3rd Qu.:68.66
                    Max.   :2018   Max.   :7.858   Max.   :11.645     Max.   :0.9840   Max.   :73.97
                                                                                       NA's   :3
 freedom_to_make_life_choices   generosity       perceptions_of_corruption positive_affect  negative_affect
 Min.   :0.3740               Min.   :-0.33800   Min.   :0.0970            Min.   :0.3790   Min.   :0.0930
 1st Qu.:0.7153               1st Qu.:-0.15100   1st Qu.:0.6910            1st Qu.:0.5770   1st Qu.:0.2155
 Median :0.7955               Median :-0.05500   Median :0.7940            Median :0.6650   Median :0.2850
 Mean   :0.7838               Mean   :-0.02876   Mean   :0.7346            Mean   :0.6526   Mean   :0.2929
 3rd Qu.:0.8762               3rd Qu.: 0.06100   3rd Qu.:0.8520            3rd Qu.:0.7375   3rd Qu.:0.3600
 Max.   :0.9700               Max.   : 0.50900   Max.   :0.9520            Max.   :0.8410   Max.   :0.5440
 NA's   :1                                       NA's   :8                 NA's   :2        NA's   :2
```

(2)

Happiness is normally distributed with an average around 5.5. All other variables are skewed left except generosity which is skewed right.



(3)

II.b Running of the Initial Full Model

The R^2 is 0.7829, meaning that 78.29% of the variability in happiness score is explained by the model. The F-statistic is 54.09 with a very low p-value (< 2.2e-16), indicating that the model is statistically significant.

```
Call:
lm(formula = life_ladder ~ . - country - year, data = happiness)

Residuals:
     Min       1Q   Median       3Q      Max
-1.66321 -0.33003  0.02126  0.29709  1.66671

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                      -4.62111    0.87144  -5.303 5.28e-07 ***
log_GDP_per_capita                0.35544    0.08478   4.192 5.31e-05 ***
social_support                    2.83854    0.72502   3.915 0.000150 ***
healthy_life_expectancy_at_birth  0.03672    0.01448   2.535 0.012536 *
freedom_to_make_life_choices      0.78692    0.55499   1.418 0.158812
generosity                        0.13951    0.34495   0.404 0.686613
perceptions_of_corruption        -0.76970    0.30685  -2.508 0.013462 *
positive_affect                   2.11173    0.56563   3.733 0.000290 ***
negative_affect                   2.47878    0.72805   3.405 0.000901 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5326 on 120 degrees of freedom
  (12 observations deleted due to missingness)
Multiple R-squared:  0.7829,    Adjusted R-squared:  0.7684
F-statistic: 54.09 on 8 and 120 DF,  p-value: < 2.2e-16
```

(4)

Log GDP: This feature variable is statistically significant at a p-value of <0.001. When the log of GDP per capita increases by 1, the happiness score increases by 0.36. This makes intuitive sense as you would expect a country with higher GDP to have happier citizens.

Social Support: This feature variable is statistically significant at a p-value of <0.001. When the social support rating increases by 1, the happiness score increases by 2.84. This makes intuitive sense as you would expect that when people have those they feel like they can rely on, they will be happier.

Life Expectancy: This feature variable is statistically significant at a p-value of 0.05. When the life expectancy increases by 1, the happiness score increases by 0.04. This is slightly surprising as you would expect a higher life expectancy to have a high impact in a happier population. However, it is still a positive coefficient, so it does still lead to happiness, just at a lesser extent.

Freedom: This feature variable is not statistically significant at a p-value of 0.1.

Generosity: This feature variable is not statistically significant at a p-value of 0.1.

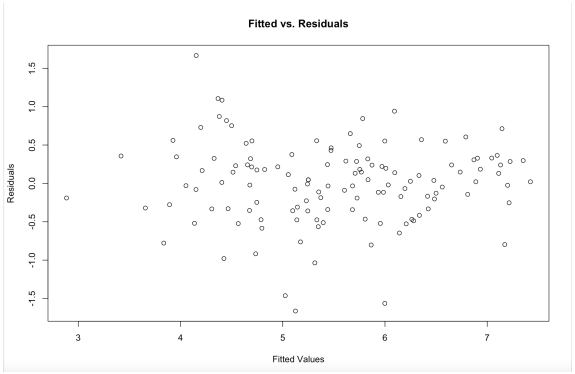
Perceptions of Corruption: This feature variable is statistically significant at a p-value of 0.1. When the perception of corruption score increases by 1, the happiness score decreases by 0.77. This makes intuitive sense as you would expect that when people feel there is corruption within the government and business, they will be less happy.

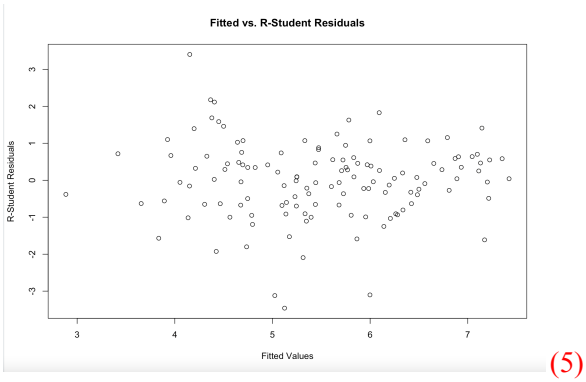
Positive Affect: This feature variable is statistically significant at a p-value of <0.001. When the positive affect score increases by 1, the happiness score increases by 2.11. This makes intuitive sense as you would expect when people have higher averages of laughter and enjoyment, they will be happier.

Negative Affect: This feature variable is statistically significant at a p-value of <0.001. When the negative affect score increases by 1, the happiness score increases by 2.11. This is very surprising since you would expect when people have higher averages of worry, sadness, and anger, they will be less happy.

II.c. Model Diagnostics (including collinearity issues)

We first checked for constant variance of the errors, ensuring the errors are *unbiased* and *homoscedastic*. For biasedness, the points on the graph of fitted values vs. errors should be centered around a horizontal band and not exhibiting any trend such as positive linear, negative linear, etc. For scedasticity, the points should all be evenly spaced around this horizontal band. Pictured below are two graphs: both have the fitted happiness values on the horizontal axis. The first graph has the regular residuals on the vertical axis, while the second has a more precise r-student residual plotted.

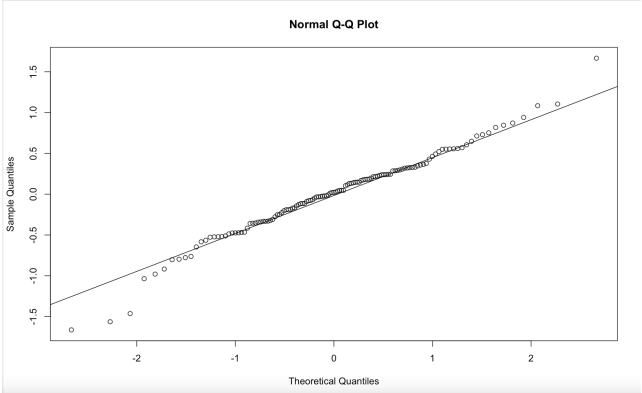




(5)

The graphs are shaped in the same manner, so they will both have the same characteristics. Both clumps of fitted-residual points are centered around a horizontal band at 0. Therefore, the errors seem to be unbiased. The scedasticity is a bit more difficult to interpret. It looks as if the errors in the 4-5 range of the fitted values are a bit more spread out than the errors in the 5-7 range in both cases. However, the spacing does not seem so concerning to make us question the constant variance assumption for the errors.

There are numerous ways to check the normality assumption for the errors. The top way is the Q-Q Plot, where the points should fall right along a hypothetical Q-Q Line, with some deviation in both tails. Attached below is the Q-Q Plot of our residuals.



(6)

Aside from the typical trailing off of the points in the tails, the residuals seem to be normally distributed. There may be a slight S-shape to the graph, which would suggest a short-tailed distribution, but it looks to conform to the normality assumption for the most part. A histogram for the residuals to further confirm the normality assumption can be found in the Appendix (7).

The last way we can check the normality of our errors is by using the Shapiro-Wilk normality test. Once we perform this test, we get a p-value. We want this p-value to be large so we fail to reject the null hypothesis, as the null hypothesis is that the distribution of the errors is normal. Interestingly, we got a p-value of 0.039 for the Shapiro test, which is quite small and would lead us to reject the hypothesis that the errors are normal. Code for the Shapiro Test can be found in the Appendix (8).

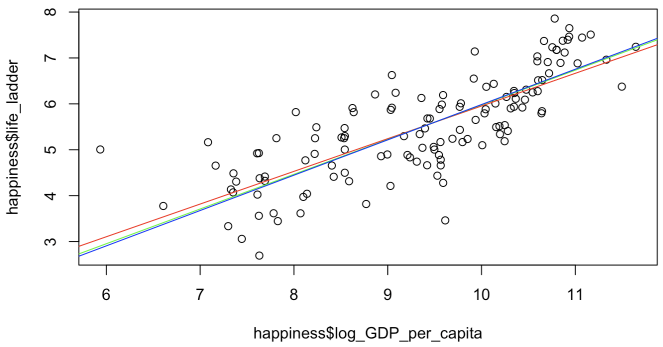
After checking for overall error characteristics, we will now check for any individual points that are skewing our model disproportionately. We will check for large leverage points, outlier points, and see if any of the large leverage points correspond to influential points.

Large leverage points are points that are far outside the "centroid" of all the points, denoted as x-bar. This point would dramatically expand the convex hull that encompasses all of the leverage points and could have too much influence on the model. As a rule of thumb, a "large" leverage point will be greater than 2 * ([k+1] /n). In our case, this value is 2 * (9/129), or 0.139. We found that four points exceeded this rule of thumb. The most egregious point had a leverage of 0.3622 and corresponded to Venezuela. The other three points corresponded to Eswatini, Indonesia, and Rwanda. A plot of all the leverage points, as well as the code and output identifying the four points mentioned, can be found in the Appendix (9, 10).

To check for potential outliers we can use the R-student residuals for each observation (which we also used in 3a to check the error constant variance assumption). A plot for these R-student (aka Jackknife) residuals can be found in the Appendix (11). We can use R to find the R-student residual with the maximum (absolute) value and then measure if that point is an outlier. This value was -3.46, corresponding to Botswana. Now, to make sure we do not erroneously claim that points are outliers, we make our critical value that we compare these residuals to extremely high. The Bonferroni critical value we will use is defined by: t [alpha/(2n), n-p-1]. Given that our n is 129 and our p is 9, our desired critical value will be t [0.05/258, 119]. This returns 3.65, which is larger than the absolute value of -3.46, and therefore we fail to reject the null hypothesis that all the points come from the same model. There are no outliers. The code identifying the maximum R-student residual and getting the Bonferroni critical value can be found in the Appendix (12, 13).

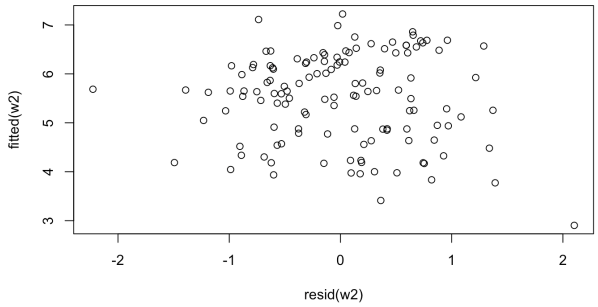
The manner in which we will check for influential points is using Cook's Distance. As a rule of thumb, if the maximum Cook's Distance for a point is under 0.5, there are no significantly influential points. A plot of the Cook's Distance for all our points corresponding to their indices can be found in the Appendix (14). Using R, we find that our maximum Cook's Distance is 0.204, corresponding to Rwanda. This is well within our rule-of-thumb and we can therefore conclude that there are no influential points. Code identifying the maximum Cook's Distance can be found in the Appendix (15).

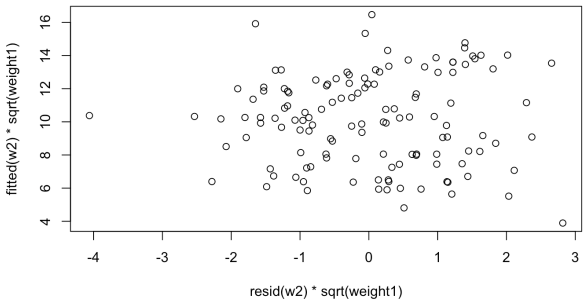
Next, we checked for the possibility of serial correlations. We did this by ordering the predictors and then using the generalized least squares model for each predictor since there was no clear predictor ordered by time or sequence. We concluded that there is no serial correlation in any of the predictors due to the correlation of the errors of the full model and the generalized least squares model being close to 0 or no correlation. Even though in all cases the correlation got smaller, the correlation was not big enough in the first place to need correcting considering the largest correlation was for positive affect at .14. Therefore a generalized least squares model is not necessary for this data. (See Appendix # 35-36 for example Code and output)



We also explored the possibility that the variance of the errors depends linearly on a specific predictor by using iteratively weighted least squares estimation on each predictor. We found that it is unlikely that the variance of errors depends linearly on any specific predictor due to most models changing minimally when using iteratively reweighted least squares as shown by the graph above. The one that changed the

most was log GDP per capita. However, as seen in the plots below, when the residuals vs fitted values of that predictor were plotted for both weighted and unweighted there was not a significant change in the level of homoscedasticity leading us to the aforementioned conclusion that weighted least squares estimate is not necessary for this model. (See Appendix # 37-38 for Code and output)





Another step in model diagnostics is to assess collinearity in the model. A good indicator of potential collinearity in the dataset is observing whether or not small deviations in the dependent variable will change the summary statistics, also known as sensitivity analysis. We will conduct sensitivity analysis to see if small deviations in the dependent variable will drastically change the beta coefficient sign and strength, and overall goodness-of-fit. (See Appendix # 29 for Code)

Model 1

```
Call:
lm(formula = life_ladder + 0.75 * rnorm(141) ~ . - country -
    year, data = happiness)

Residuals:
     Min       1Q   Median       3Q      Max
-2.14765 -0.46281 -0.02271  0.54152  2.33898

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     -4.29460    1.48916  -2.884  0.00466 **
log_GDP_per_capita               0.29794    0.14488   2.056  0.04191 *
social_support                   2.70065    1.23895   2.180  0.03123 *
healthy_life_expectancy_at_birth 0.04268    0.02475   1.724  0.08725 .
freedom_to_make_life_choices     0.30956    0.94839   0.326  0.74469
generosity                       0.35424    0.58947   0.601  0.54901
perceptions_of_corruption       -0.68627    0.52435  -1.309  0.19310
positive_affect                  3.06544    0.96658   3.171  0.00193 **
negative_affect                  1.35918    1.24412   1.092  0.27681
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9102 on 120 degrees of freedom
  (12 observations deleted due to missingness)
Multiple R-squared:  0.5629,    Adjusted R-squared:  0.5337
F-statistic: 19.31 on 8 and 120 DF,  p-value: < 2.2e-16
```

Model 2

```
Call:
lm(formula = life_ladder + 0.5 * rnorm(141) ~ . - country - year,
    data = happiness)

Residuals:
     Min       1Q   Median       3Q      Max
-1.53687 -0.35710 -0.01846  0.38925  1.61976

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     -5.50248    1.08951  -5.050 1.59e-06 ***
log_GDP_per_capita               0.37293    0.10600   3.518 0.000615 ***
social_support                   2.74341    0.90644   3.027 0.003027 **
healthy_life_expectancy_at_birth 0.04702    0.01811   2.597 0.010591 *
freedom_to_make_life_choices     1.39880    0.69386   2.016 0.046038 *
generosity                       0.57250    0.43127   1.327 0.186877
perceptions_of_corruption       -0.84669    0.38363  -2.207 0.029213 *
positive_affect                  1.51603    0.70717   2.144 0.034067 *
negative_affect                  2.94790    0.91023   3.239 0.001554 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6659 on 120 degrees of freedom
  (12 observations deleted due to missingness)
Multiple R-squared:  0.7237,    Adjusted R-squared:  0.7053
F-statistic: 39.29 on 8 and 120 DF,  p-value: < 2.2e-16
```

Original Model

```
Call:
lm(formula = life_ladder ~ . - country - year, data = happiness)

Residuals:
     Min       1Q   Median       3Q      Max
-1.66321 -0.33003  0.02126  0.29709  1.66671

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     -4.62111    0.87144  -5.303 5.28e-07 ***
log_GDP_per_capita               0.35544    0.08478   4.192 5.31e-05 ***
social_support                   2.83854    0.72502   3.915 0.000150 ***
healthy_life_expectancy_at_birth 0.03672    0.01448   2.535 0.012536 *
freedom_to_make_life_choices     0.78692    0.55499   1.418 0.158812
generosity                       0.13951    0.34495   0.404 0.686613
perceptions_of_corruption       -0.76970    0.30685  -2.508 0.013462 *
positive_affect                  2.11173    0.56563   3.733 0.000290 ***
negative_affect                  2.47878    0.72805   3.405 0.000901 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5326 on 120 degrees of freedom
  (12 observations deleted due to missingness)
Multiple R-squared:  0.7829,    Adjusted R-squared:  0.7684
F-statistic: 54.09 on 8 and 120 DF,  p-value: < 2.2e-16
```

As we can see, based on the different models, beta coefficient significance and strength and the overall goodness-of-fit are significantly impacted by minimal changes to the response. Therefore, we have established potential collinearity in the data.

Now we will look at the correlation between variables. (See Appendix # 30 for Code)

	life_ladder	log_GDP_per_capita	social_support	generosity	healthy_life_expectancy_at_birth	freedom_to_make_life_choices	perceptions_of_corruption	positive_affect	negative_affect
life_ladder	1.000	0.778	0.742	-0.051	0.754	0.523	-0.441	0.423	-0.448
log_GDP_per_capita	0.778	1.000	0.768	-0.260	0.840	0.355	-0.326	0.149	-0.589
social_support	0.742	0.768	1.000	-0.167	0.733	0.396	-0.222	0.270	-0.645
generosity	-0.051	-0.260	-0.167	1.000	-0.177	0.239	-0.238	0.255	0.075
healthy_life_expectancy_at_birth	0.754	0.840	0.733	-0.177	1.000	0.331	-0.308	0.138	-0.489
freedom_to_make_life_choices	0.523	0.355	0.396	0.239	0.331	1.000	-0.456	0.592	-0.367
perceptions_of_corruption	-0.441	-0.326	-0.222	-0.238	-0.308	-0.456	1.000	-0.343	0.304
positive_affect	0.423	0.149	0.270	0.255	0.138	0.592	-0.343	1.000	-0.250
negative_affect	-0.448	-0.589	-0.645	0.075	-0.489	-0.367	0.304	-0.250	1.000

Based on this output, negative effect and social support have a relatively strong negative correlation (< -0.5). Furthermore, positive affect and freedom_to_make_life_choice have a relatively strong correlation (>0.5). Finally, life_ladder is strongly positively correlated with log_GDP_per_capita, social_support, and healthy_life_expectancy (> 0.7). Now we have a better idea of what is causing our collinearity.

We will now conduct the Condition Numbers Test and Variance Inflation Factor to dive deeper into the roots of the collinearity issues.
VIF: (See Appendix # 32 for Code)

```
> vif(model)
     log_GDP_per_capita              social_support healthy_life_expectancy_at_birth     freedom_to_make_life_choices                  generosity 
               4.742125                    3.340700                    3.752970                    1.959149                    1.337855 
perceptions_of_corruption              positive_affect              negative_affect 
               1.480502                    1.648568                    1.883040 
```

Condition Numbers: (See Appendix # 31 for Code)

```
> condition_numbers
[1]    1.0000   98.6838  287.9086  369.6094  556.9797  707.9340  933.9028 1146.2604
```

These outputs suggest that none of the VIFs > 10, so using this test, collinearity is not an issue, suggesting that individual variables are not causing the collinearity issues. As several condition numbers are large (> 30), this suggests problems are being caused by more than one linear combination of variables. This leads us to question if the collinearity is associated with individual variables as the VIF assesses or if the collinearity is associated with a combination of variables as the condition numbers assess.

Based on the correlation matrix, and the insights from the VIF and Condition Number Test, we can conclude that the collinearity is associated with a combination of variables. Amputation may not be as effective as intended as individual variables are not causing our problem. Here are some examples to prove Amputation is not a viable solution. (See Appendix #31 for Code)

Example 1:

```
> aic_model<- lm(life_ladder ~ . - country - year - generosity, data=happiness)
> aic_condition_numbers
[1]    1.00000   98.82785  293.78286  428.71191  702.41973  932.88061 1146.25405
```

Example 2:

```
> bic_model<- lm(life_ladder~ . - country - generosity - year - freedom_to_make_life_choices,data=happiness)
> bic_condition_numbers
[1]    1.00000   98.83418  294.09623  545.90357  702.40043 1145.77921
```

Example 3:

```
> corr_model<- lm(life_ladder~ . - country - healthy_life_expectancy_at_birth - year,data=happiness)
> corr_model_condition_numbers
[1]   1.00000  38.97211  49.57137  82.09161 103.98844 137.86215 157.72621
```

Example 4:

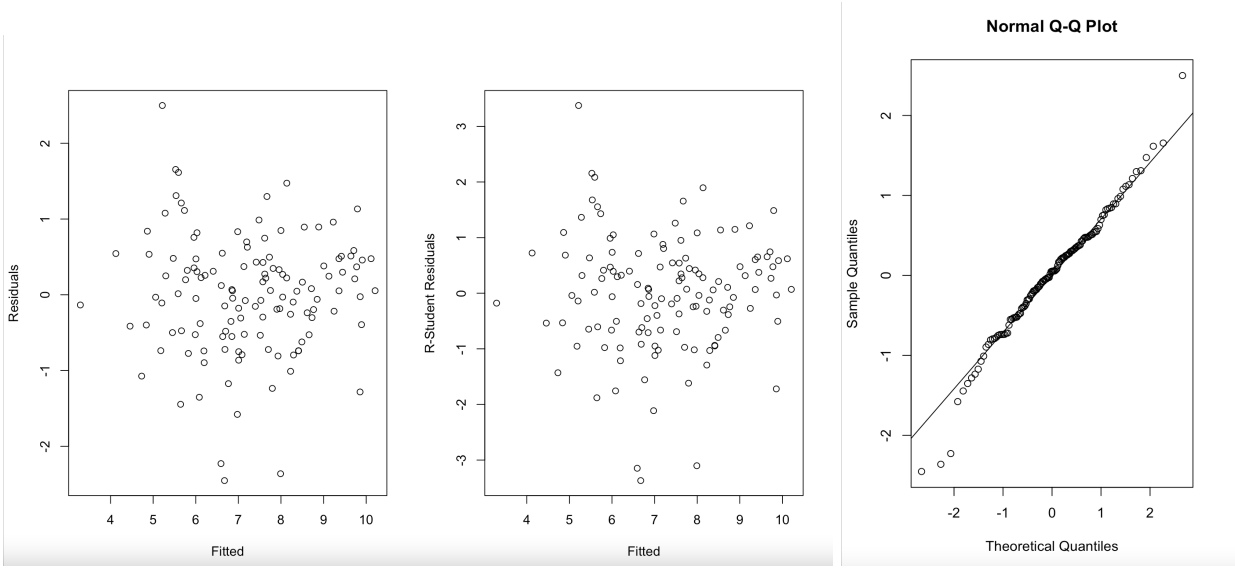
```
> corr2_model<- lm(life_ladder~ . - country - healthy_life_expectancy_at_birth - social_support - year,data=happiness)
> corr2_model_condition_numbers
[1]   1.00000  38.85856  49.54319  82.44814 114.42589 137.68762
```

Amputation may not be as effective as intended as individual variables are not causing our problem. Thus, we have determined the presence of collinearity that is complex in its nature.

II.d. Correcting Model Inadequacies

The first way we can go about modifying our model to correct inadequacies is through a Box-Cox transformation. Because all of our response data is positive, we can apply a Box-Cox transformation. To see what might be best to set our lambda to, we should plot the Box-Cox line and see where the confidence interval falls. Based on where this confidence interval lies, we could try raising the response variable to the power of whatever number is in the middle of the interval. A plot of our Box-Cox interval for lambda can be found in the Appendix (28). The 95% confidence interval seems to lie from 0.87 to 1.45 or so. 1 is within this interval and is therefore plausible, so there is not a huge need to apply a Box-Cox transformation. However, because 1.16 is the middle of this interval, we will create a new model with our response variable raised to the power of 1.16 and see if the error variance and/or normality significantly improves.

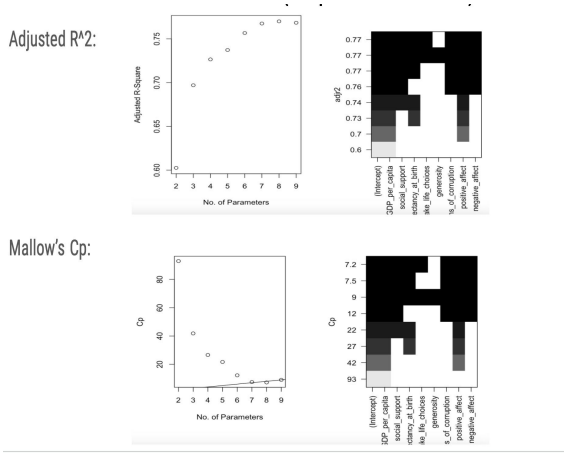
Attached below are the plots of our new model's fitted values up against our new model's residuals and jackknife/R-student residuals, as well as a Q-Q Plot of our new model. The formulation of this model, as well as code for the plots, can be found in the Appendix (39).



Compared to the plots we generated for the initial full model, there does not seem to be a significantly more homoscedastic distribution of the errors here. Additionally, the normality seems to have gotten worse as the S-tailed distribution is more profound here than it was before. Using R we found that a Shapiro-Wilkes Normality Test gave this model a p-value of 0.05229. This p-value is fairly larger than the original p-value of 0.039 from a percentage perspective, but at an alpha = 0.1 level of significance, it would still lead to a rejection of the null hypothesis. Therefore, we conclude that while applying a Box-Cox transformation to our model may make it conform to the Gauss-Markov assumptions slightly better, it is not enough to justify changing it over, which would make the model much harder to interpret.

II.e. Model Selection (Exhaustive Search and Stepwise Methods)

Exhaustive Search: We will apply an exhaustive model selection using the AIC, BIC, Mallow's Cp, Adjusted R-squared, and R-squared. (See Appendix #33, # 34). Based on the outputs below, it is clear the best model through exhaustive search using Mallow's Cp has 7 parameters (6 predictors), amputating generosity and freedom_to_make_life_choices out of the model. The best model through exhaustive search using Adjusted R-squared has 8 parameters (7 predictors), amputating generosity out of the model.



AIC: `step_model_aic <- step(model, k = 2, direction = "both", trace = 1)`

```
Step:  AIC=-155.67
life_ladder ~ log_GDP_per_capita + social_support + healthy_life_expectancy_at_birth +
    freedom_to_make_life_choices + perceptions_of_corruption +
    positive_affect + negative_affect

                                Df Sum of Sq    RSS     AIC
<none>                                        34.091 -155.67
- freedom_to_make_life_choices   1   0.6521 34.744 -155.22
+ generosity                     1   0.0464 34.045 -153.84
- healthy_life_expectancy_at_birth 1   1.8628 35.954 -150.81
- perceptions_of_corruption      1   2.0164 36.108 -150.26
- negative_affect                1   3.2819 37.373 -145.81
- positive_affect                1   4.8824 38.174 -143.08
- social_support                 1   4.3302 38.422 -142.24
- log_GDP_per_capita             1   5.8835 39.175 -139.74
```

BIC: `step_model_bic <- step(model, k = log(nrow(happiness)), direction = "both", trace = 1)`

```
Step:  AIC=-135.21
life_ladder ~ log_GDP_per_capita + social_support + healthy_life_expectancy_at_birth +
    perceptions_of_corruption + positive_affect + negative_affect

                                Df Sum of Sq    RSS     AIC
<none>                                        34.744 -135.21
- healthy_life_expectancy_at_birth 1   1.9164 36.660 -133.14
+ freedom_to_make_life_choices   1   0.6521 34.091 -132.79
+ generosity                     1   0.1282 34.615 -130.82
- perceptions_of_corruption      1   2.8573 37.601 -129.87
- negative_affect                1   3.1218 37.865 -128.97
- social_support                 1   4.6403 39.384 -123.89
- log_GDP_per_capita             1   5.2551 39.999 -121.90
- positive_affect                1   7.7041 42.448 -114.23
```

The best model through exhaustive search using AIC has 8 parameters (7 predictors), amputating generosity out of the model. This result is concurrent with the best model through an exhaustive search using Adjusted R-squared. The best model through exhaustive search using BIC has 7 parameters (6 predictors), amputating generosity and freedom_to_make_life_choices out of the model, this result is concurrent with the best model through exhaustive search using Mallow's Cp. The best model through exhaustive search using R-squared was the full model, with 9 parameters (8 predictors), leaving all variables in the model. (See Appendix # 33, # 34).

Stepwise Methods:

We will run through backward elimination and forward selection to reduce our model. We start with the full model for backwards elimination and iteratively get rid of the least significant predictor, corresponding to the predictor with the greatest p-value that is still greater than 0.05. Below, on the far left, we see the summary of the full model and that the predictor with the greatest p-value is generosity. We update the model to get rid of generosity and repeat the same process. The next iteration is the summary in the middle. We continue until all of the p-values are less than 0.05. The summary on the far right is the completed model (all predictors are significant and less than 0.05). We end with six predictors and get rid of generosity and freedom_to_make_life_choices. (See #16 in the appendix for R code)

```
Call:
lm(formula = life_ladder ~ log_GDP_per_capita + social_support +
    healthy_life_expectancy_at_birth + freedom_to_make_life_choices +
    perceptions_of_corruption + positive_affect + negative_affect,
    data = happiness)

Residuals:
     Min       1Q   Median       3Q      Max
-1.68864 -0.32471  0.03681  0.29617  1.66386

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     -4.58279    0.86328  -5.309 5.08e-07 ***
log_GDP_per_capita               0.34651    0.08158   4.248 4.27e-05 ***
social_support                   2.83170    0.72231   3.920 0.000147 ***
healthy_life_expectancy_at_birth 0.03705    0.01441   2.571 0.011343 *
freedom_to_make_life_choices     0.82753    0.54394   1.521 0.130776
perceptions_of_corruption       -0.79740    0.29807  -2.675 0.008502 **
positive_affect                  2.13475    0.56081   3.807 0.000223 ***
negative_affect                  2.47608    0.72550   3.413 0.000875 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5308 on 121 degrees of freedom
  (12 observations deleted due to missingness)
Multiple R-squared:  0.7826,    Adjusted R-squared:   0.77
F-statistic: 62.22 on 7 and 121 DF,  p-value: < 2.2e-16
```

```
Call:
lm(formula = life_ladder ~ log_GDP_per_capita + social_support +
    healthy_life_expectancy_at_birth + freedom_to_make_life_choices +
    perceptions_of_corruption + positive_affect + negative_affect,
    data = happiness)

Residuals:
     Min       1Q   Median       3Q      Max
-1.68864 -0.32471  0.03681  0.29617  1.66386

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     -4.58279    0.86328  -5.309 5.08e-07 ***
log_GDP_per_capita               0.34651    0.08158   4.248 4.27e-05 ***
social_support                   2.83170    0.72231   3.920 0.000147 ***
healthy_life_expectancy_at_birth 0.03705    0.01441   2.571 0.011343 *
freedom_to_make_life_choices     0.82753    0.54394   1.521 0.130776
perceptions_of_corruption       -0.79740    0.29807  -2.675 0.008502 **
positive_affect                  2.13475    0.56081   3.807 0.000223 ***
negative_affect                  2.47608    0.72550   3.413 0.000875 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5308 on 121 degrees of freedom
  (12 observations deleted due to missingness)
Multiple R-squared:  0.7826,    Adjusted R-squared:   0.77
F-statistic: 62.22 on 7 and 121 DF,  p-value: < 2.2e-16
```

```
Call:
lm(formula = life_ladder ~ log_GDP_per_capita + social_support +
    healthy_life_expectancy_at_birth + perceptions_of_corruption +
    positive_affect + negative_affect, data = happiness)

Residuals:
     Min       1Q   Median       3Q      Max
-1.65280 -0.32754  0.04575  0.30626  1.68566

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     -4.17274    0.83089  -5.022 1.75e-06 ***
log_GDP_per_capita               0.33945    0.07992   4.247 4.23e-05 ***
social_support                   2.99574    0.71504   4.190 5.29e-05 ***
healthy_life_expectancy_at_birth 0.03797    0.01445   2.628  0.00968 **
perceptions_of_corruption       -0.94113    0.28658  -3.284  0.00133 **
positive_affect                  2.52001    0.48783   5.166 9.36e-07 ***
negative_affect                  2.36584    0.72405   3.268  0.00141 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5326 on 123 degrees of freedom
  (11 observations deleted due to missingness)
Multiple R-squared:  0.7775,    Adjusted R-squared:  0.7666
F-statistic: 71.63 on 6 and 123 DF,  p-value: < 2.2e-16
```

Next, we run through forward selection. We start with only the intercept and iteratively add the most significant predictors, which corresponds to the predictor with the smallest p-value that is still less than 0.05 (or choose the predictor with the largest F value). Below on the left, we see that we start with only the intercept (life_ladder~1) and that the predictor with the smallest p-value is log_GDP_per_capita, so we add it to the model. We update the model and repeat the same process. The next iteration is on the right (see life_ladder~log_GDP_per_capita). We continue until all of the p-values exceed 0.05. The summary on the bottom is the completed model. We end with six predictors and get rid of generosity and freedom_to_make_life_choices. Note that it is the same model derived from backward elimination. (See #17 in the appendix for R code)

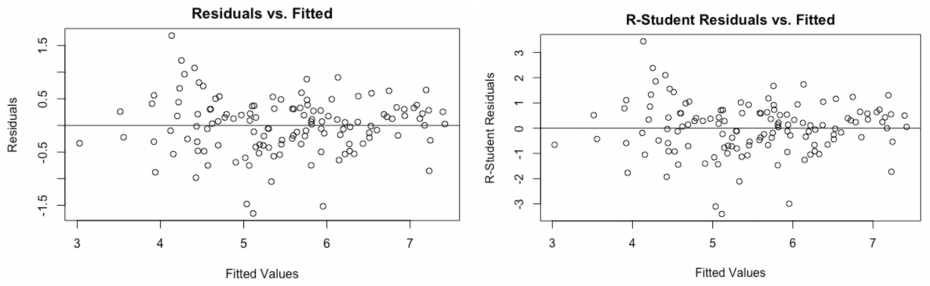
```
Model:
life_ladder ~ 1
                                      Df Sum of Sq     RSS     AIC  F value    Pr(>F)
<none>                                             156.807  27.181
log_GDP_per_capita                     1    94.978  61.829 -90.873 213.5261 < 2.2e-16 ***
social_support                         1    86.234  70.573 -73.808 169.8458 < 2.2e-16 ***
healthy_life_expectancy_at_birth       1    89.058  67.749 -79.076 182.7179 < 2.2e-16 ***
perceptions_of_corruption              1    30.543 126.264   1.234  33.6242 4.304e-08 ***
positive_affect                        1    28.014 128.793   3.793  30.2339 1.779e-07 ***
negative_affect                        1    31.536 125.271   0.216  34.9917 2.450e-08 ***
freedom_to_make_life_choices           1    42.964 113.843 -12.124  52.4583 2.744e-11 ***
generosity                             1     0.409 156.398  28.845   0.3632    0.5477
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Model:
life_ladder ~ log_GDP_per_capita
                                      Df Sum of Sq     RSS      AIC F value    Pr(>F)
<none>                                             61.829  -90.873
social_support                         1    7.8897 53.939 -106.483 20.1855 1.473e-05 ***
healthy_life_expectancy_at_birth       1    5.3046 56.524 -100.444 12.9509 0.0004447 ***
perceptions_of_corruption              1    6.1659 55.663 -102.425 15.2866 0.0001443 ***
positive_affect                        1   15.0617 46.767 -124.888 44.4442 5.786e-10 ***
negative_affect                        1    0.0232 61.805  -88.921  0.0517 0.8204310
freedom_to_make_life_choices           1   10.9631 50.865 -114.051 29.7433 2.211e-07 ***
generosity                             1    3.8578 57.971  -97.184  9.1836 0.0029165 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

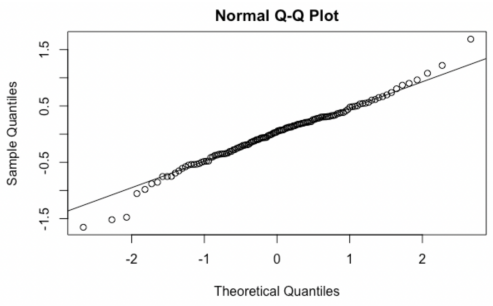
```
Call:
lm(formula = life_ladder ~ log_GDP_per_capita + positive_affect +
    healthy_life_expectancy_at_birth + social_support + negative_affect +
    perceptions_of_corruption, data = happiness)

Residuals:
     Min       1Q   Median       3Q      Max
-1.65280 -0.32754  0.04575  0.30626  1.68566

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                       -4.17274    0.83089  -5.022 1.75e-06 ***
log_GDP_per_capita                 0.33945    0.07992   4.247 4.23e-05 ***
positive_affect                    2.52001    0.48783   5.166 9.36e-07 ***
healthy_life_expectancy_at_birth   0.03797    0.01445   2.628  0.00968 **
social_support                     2.99574    0.71504   4.190 5.29e-05 ***
negative_affect                    2.36584    0.72405   3.268  0.00141 **
perceptions_of_corruption         -0.94113    0.28658  -3.284  0.00133 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5326 on 123 degrees of freedom
  (11 observations deleted due to missingness)
Multiple R-squared:  0.7775,    Adjusted R-squared:  0.7666
F-statistic: 71.63 on 6 and 123 DF,  p-value: < 2.2e-16
```

II.f. Model diagnostics of Selected Reduced or Corrected Model

We are going to run some model diagnostics on the reduced model derived from backward elimination. Note that k=6, p=7, and n=130.



When checking the normal variance assumption, both graphs have fitted happiness values on the horizontal axis. The first graph has regular residuals on the vertical axis, while the second graph has r-student residuals on the vertical axis. Again, both graphs show that the points are centered around the horizontal line at 0, indicating unbiasedness. In terms of scedasticity, they seem to be fairly evenly spaced around the horizontal band. It is pretty safe to assume that the constant variance assumption holds. Note that there may be some larger variance corresponding with smaller fitted values, but we do not think it is enough to be of concern. (See #18 in the appendix for R code)




```
        Shapiro-Wilk normality test

data:  residuals(backwardModel)
W = 0.98181, p-value = 0.07885
```

The residuals seem to be pretty normally distributed, as the Q-Q plot follows a pretty linear path. Again, there seems to be some trailing off in the points at the tails, but nothing of great concern. One may also interpret a slight S-curve, which would indicate a shorter tail distribution, but again does not seem to be too concerning. The Shapiro-Wilk normality test gives a p-value of 0.07885 which is a definite improvement over the 0.039 p-value obtained in the full model above. It is also over the threshold of the normal alpha of 0.05, which is a positive sign. See #19 and #20 in the appendix for R code. See #21 in the appendix for further evidence of normality.

```
[1] 0.1076923
       19        39        45       107       112       137       139
0.1110963 0.1350127 0.1083399 0.1472193 0.1209317 0.3170732 0.1078397
```

We must do some quantitative analysis to see which points have large leverage points. Generally, any point with leverage greater than 2*[(k+1)/n] is considered an outlying X observation (i.e. a point with large leverage). This is the first number. Based on the R output, it seems as though seven points have unusually large leverages. These correspond to countries Burundi, Eswatini, Georgia, Rwanda, Singapore, Venezuela, and Yemen. See appendix #22 for a plot of leverages of all points and #23 for the R code shown above.

```
                12
3.439728   [1] 3.651938
```

We must check if points are outliers using r-student residuals. The value on the left is the largest absolute value jackknife residual. The value on the right is the critical value. We use the Bonferroni correction since we are checking if any points in the data set is an outlier. Since the point with the largest absolute value jack residual is less than the Bonferroni critical value, we fail to reject the null hypothesis that all the points come from the same model. There are no outliers. See appendix #24 to get the largest absolute value jackknife residual and #25 for the Bonferroni critical value in R.

```
       107
0.2219613
```

For influential points, we look at Cook's distance. The point with the largest Cook's distance is the value shown above, and it is less than 0.5 (within our rule of thumb,) so there are no influential points. See appendix #26 for a plot of cook distances of all points and #27 for the R output shown above.

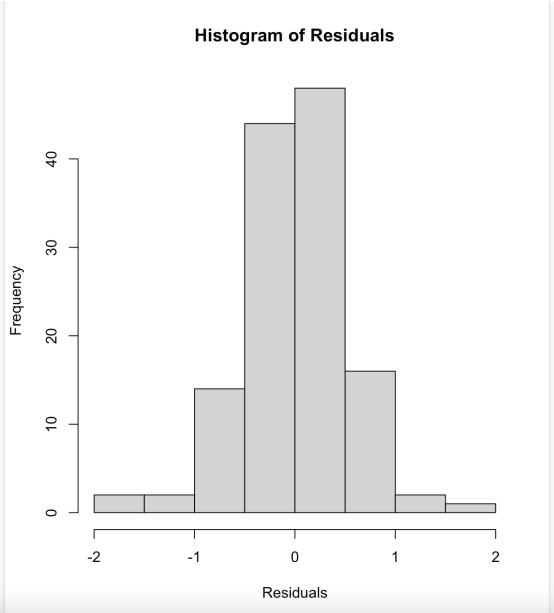
III. Conclusions and Interpretations of the Proposed Fitted Model

The best model was the reduced model found through backward elimination. With complex collinearity, the predictive power of the model is still limited. However, all feature variables are significant and the model has an R^2 of 0.7755, indicating that the model explains much of the variability in happiness score.

From the model, we can tell that all feature variables positively impact happiness, except for perceptions of corruption. The variable with the largest positive beta was social support. As the level of social support increases by one, the happiness score increases by 3. The variable with the largest negative beta was perceptions of corruption. As the level of perceptions of corruption increases by 1, the happiness score decreases by 0.94. These predictions are important since they can inform policymakers, employers, and schools of various factors impacting happiness levels. Fostering strong connections and developing strategies to reduce corruption is crucial to happiness.

Appendix

1	happiness <- read.csv("/Users/valeriepassanisi/Downloads/Happiness_2018.csv") head(happiness)
2	summary(happiness)
3	par(mfrow = c(3, 3)) hist(happiness$life_ladder,xlab="Happiness Score",main="") hist(happiness$log_GDP_per_capita,xlab="Log GDP per Capita",main="") hist(happiness$social_support,xlab="Social Support",main="") hist(happiness$healthy_life_expectancy_at_birth,xlab="Healthy Life Expectancy at Birth",main="") hist(happiness$freedom_to_make_life_choices,xlab="Freedom to Make Life Choices",main="") hist(happiness$generosity,xlab="Generosity",main="") hist(happiness$perceptions_of_corruption,xlab="Perceptions of Corruption",main="") hist(happiness$positive_affect,xlab="Positive Affect",main="") hist(happiness$negative_affect,xlab="Negative Affect",main="")
4	mymodel <- lm(life_ladder~.-country-year, data=happiness) summary(mymodel)
5	par(mfrow = c(1,2)) plot(fitted(mymodel), resid(mymodel), main = "Residuals vs. Fitted", xlab = 'Fitted Values', ylab = 'Residuals') plot(fitted(mymodel), rstudent(mymodel), main = "R-Student Residuals vs. Fitted", xlab = 'Fitted Values', ylab = 'R-Student Residuals')
6	qqnorm(resid(mymodel)) qqline(resid(mymodel))
7	hist(resid(mymodel), main = "Histogram of Residuals", xlab = "Residuals")

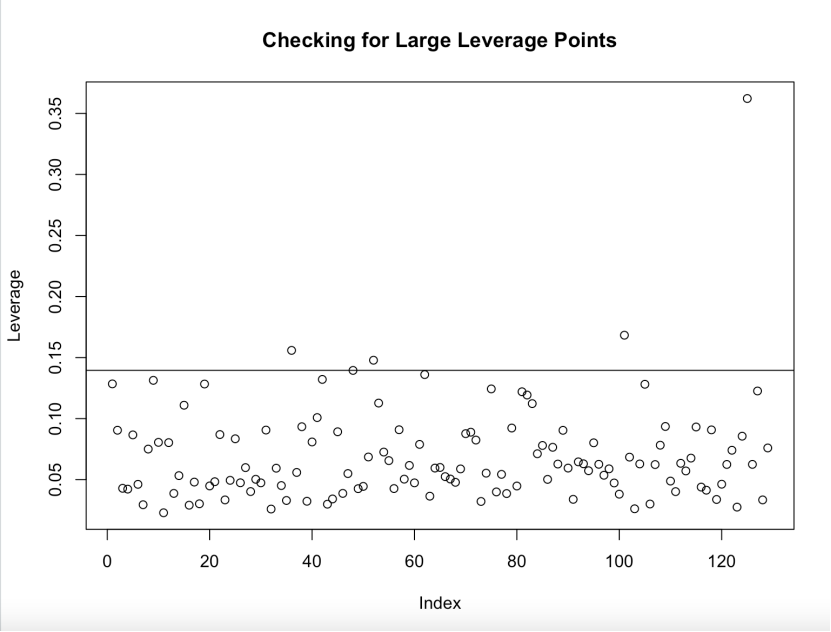


8

```
shapiro.test(resid(mymodel))

Data: resid(mymodel)
W = 0.9786, p-value = 0.039
```

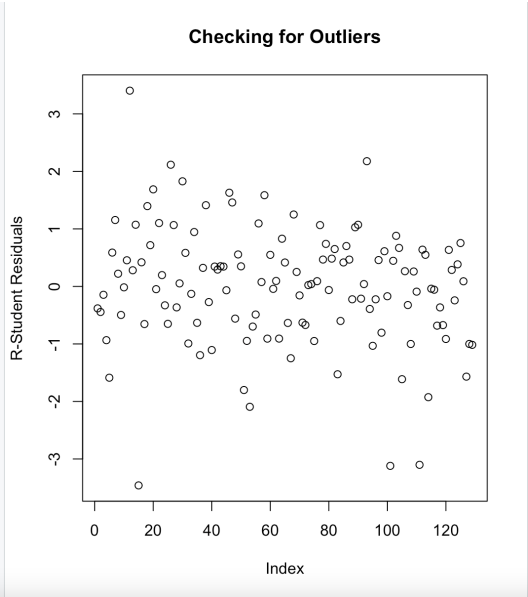
9

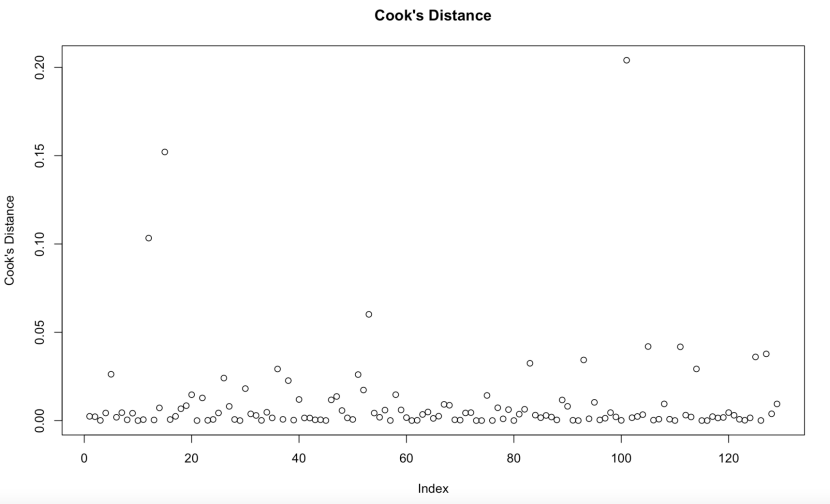
```
plot(lm.influence(mymodel)$hat, main = "Checking for Large Leverage
Points", ylab = "Leverage")
abline(h=2* (9/129))
```

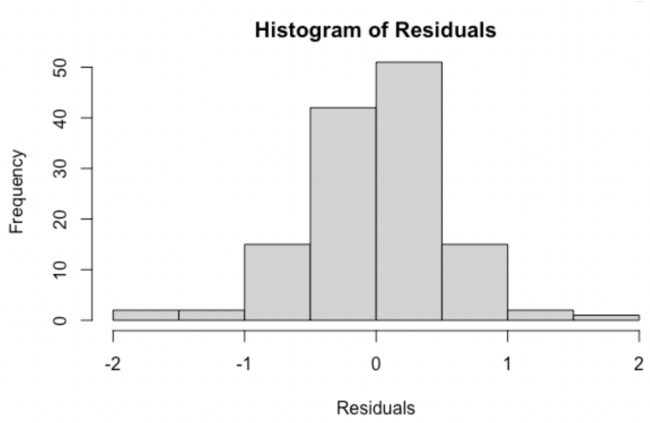
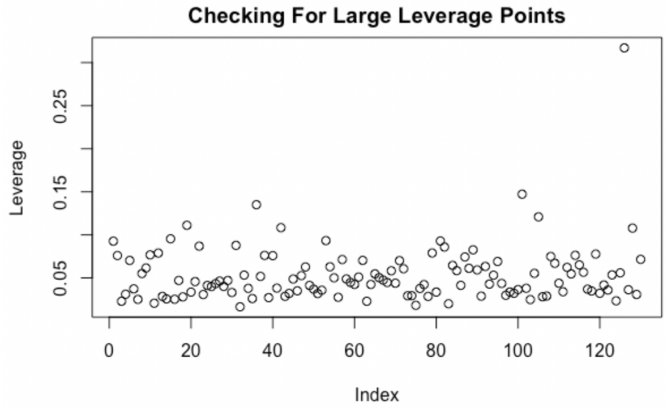


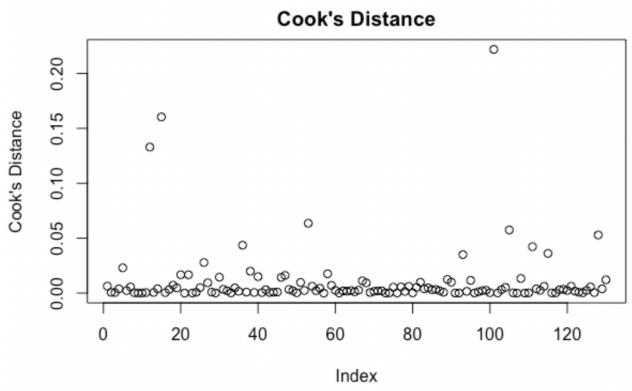
10

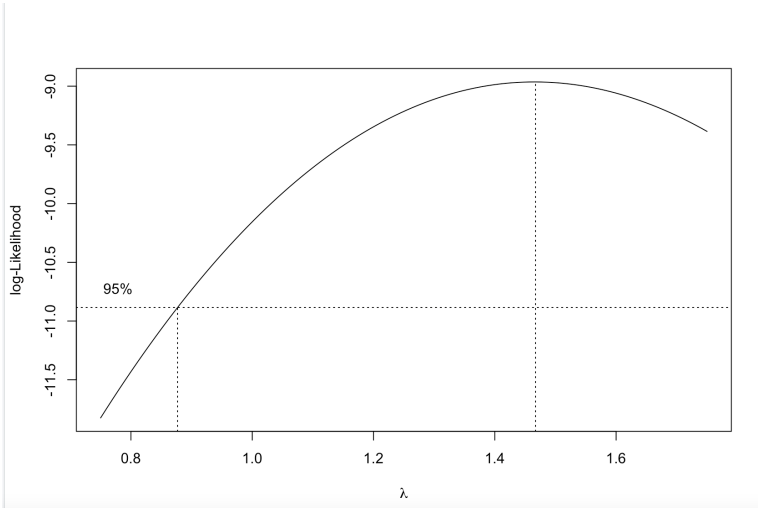
```
leverage <- lm.influence(mymodel)$hat
leverage[which.max(abs(leverage))]
leverage[leverage>(2*(9/129))]
```

	137 0.3622 39, 0.1558946; 55, 0.1478289; 107, 0.1683276; 137, 0.3622000
11	plot(rstudent(mymodel), main = 'Checking for Outliers', ylab = 'R-Student Residuals') 
12	jack <- rstudent(mymodel) jack[which.max(abs(jack))] 15 -3.459753
13	qt(1-(0.05/(2*129)), 119) [1] 3.652358
14	cook <- cooks.distance(mymodel) plot(cook, main = "Cook's Distance", ylab = "Cook's Distance")

	
15	cook[which.max(abs(cook))] 107 0.2040419
16	backwardModel <- update(backwardModel, life_ladder~.-country-year-generosity) summary(backwardModel)
17	forwardModel <- lm(life_ladder~log_GDP_per_capita+positive_affect, data=happiness) add1(forwardModel, ~log_GDP_per_capita+social_support+healthy_life_expectancy_at_birth+perce ptions_of_corruption+positive_affect+negative_affect+freedom_to_make_life _choices+generosity, test="F")
18	plot(fitted(backwardModel), resid(backwardModel), main="Residuals vs. Fitted", xlab="Fitted Values", ylab="Residuals") abline(h=0) plot(fitted(backwardModel), rstudent(backwardModel), main="R-Student Residuals vs. Fitted", xlab="Fitted Values", ylab="R-Student Residuals") abline(h=0)
19	qqnorm(resid(backwardModel)) qqline(resid(backwardModel))
20	shapiro.test(residuals(backwardModel))
21	hist(resid(backwardModel), main="Histogram of Residuals", xlab="Residuals", ylab="Frequency")

	 The histogram is also indicative of the normal nature of the residuals.
22	plot(lm.influence(backwardModel)$hat, main="Checking For Large Leverage Points", xlab="Index", ylab="Leverage") 
23	boundary = 2*(7/130) print(boundary) leverage = lm.influence(backwardModel)$hat leverage[which(leverage>boundary)]
24	jack = rstudent(backwardModel) jack[which.max(abs(jack))]
25	qt(1-(.05/(2*130)), 130-7-1)
26	plot(cooks.distance(backwardModel), main="Cook's Distance", xlab="Index", ylab="Cook's Distance")

	
27	`cook = cooks.distance(backwardModel)` `cook[which.max(abs(cook))]`

28	`library(MASS)`
	`boxcox(mymodel, plotit = T, lambda = seq(0.75, 1.75, by = 0.1))`
	 

29	`# Sensitivity Analysis Code`
	`min<- min(happiness$life_ladder)`
	`max<- max(happiness$life_ladder)`
	`mean<- mean(happiness$life_ladder)`
	`sd<- sd(happiness$life_ladder)`
	`mean - 2*sd >= min`
	`max - 2*sd >= mean`
	`model<- lm(life_ladder~ . - country - year,data=happiness)`
	`sensitivity_model_1<- lm(life_ladder + 0.75*rnorm(141)~ . - country - year, data=happiness)`
	`sensitivity_model_2<- lm(life_ladder + 0.5*rnorm(141)~ . - country - year, data=happiness)`
	`summary(sensitivity_model_1)`
	`summary(sensitivity_model_2)`
	`summary(model)`

30

```
# Correlation Matrix Code
variables_of_interest <- c("life_ladder","log_GDP_per_capita",
"social_support",
"generosity","healthy_life_expectancy_at_birth","freedom_to_make_life_cho
ices","perceptions_of_corruption","positive_affect", "negative_affect")
happiness_subset <- happiness[, variables_of_interest]
# Calculate the correlation matrix, excluding missing values
cor_matrix <- round(cor(happiness_subset, use = "complete.obs"), 3)
print(cor_matrix)

# Correlation Matrix Loop for Max Correlation for Each Variable

# Initialize a vector to store the maximum correlation for each variable
max_cor_for_var <- rep(-Inf, length(variables_of_interest))
# Initialize a vector to store the corresponding variables for max
correlation
max_var_for_cor <- rep("", length(variables_of_interest))
# Loop through each variable
for (j in seq_along(variables_of_interest)) {
  # Loop through each element in the correlation matrix for the current
variable
  for (i in seq_along(cor_matrix[, j])) {
    # Check if the element is not 1 (assuming you want to exclude
self-correlations)
    if (cor_matrix[[i, j]] != 1 && !is.na(cor_matrix[[i, j]])) {
      # Update max_cor_for_var, max_var_for_cor, and max_indices_for_var
if the current correlation is greater
      current_corr_value <- cor_matrix[[i, j]]
      abs_corr_value <- abs(current_corr_value)
      if (abs_corr_value > max_cor_for_var[j]) {
        max_cor_for_var[j] <- abs_corr_value
        max_var_for_cor[j] <- variables_of_interest[i]
        if (current_corr_value < 0) {
          # If the correlation is negative, include the sign in the
output
          sign_indicator <- "-"
        } else {
          sign_indicator <- ""
        }
      }
    }
  }
  # Print the maximum correlation, the corresponding variable on a new
line
  cat(sprintf("Variable: %s, Max Correlation: %s%f, Corresponding
Variable: %s\n",
              variables_of_interest[j], sign_indicator,
max_cor_for_var[j], max_var_for_cor[j]))
}
```

31

```
# Condition Number Code & Amputation Examples Code
x<- model.matrix(model)[,-1];
e<- eigen(t(x)%*%x)
e$val
condition_numbers<-sqrt(e$val[1]/e$val)
#If the condition number is >30, we have a collinearity problem
condition_numbers
#Initialize a numeric variable
condition_numbers_showing_multicollinearity <- numeric()
#Check to see if condition_numbers > 30
for (number in condition_numbers) {
  if (number > 30) {
    condition_numbers_showing_multicollinearity <-
c(condition_numbers_showing_multicollinearity, number)
  }
}
Condition_numbers_showing_multicollinearity

# Amputation Example Code

#Example 1 showing Amputation was not valid solution
aic_model<- lm(life_ladder ~ . - country - year - generosity,
data=happiness)
z<- model.matrix(aic_model)[,-1];
eign<- eigen(t(z)%*%z)
eign$val
aic_condition_numbers<-sqrt(eign$val[1]/eign$val)
aic_condition_numbers

#Example 2 showing Amputation was not valid solution
bic_model<- lm(life_ladder~ . - country - generosity - year -
freedom_to_make_life_choices,data=happiness)
y<- model.matrix(bic_model)[,-1];
eig<- eigen(t(y)%*%y)
eig$val
bic_condition_numbers<-sqrt(eig$val[1]/eig$val)
bic_condition_numbers

#Example 3 showing Amputation was not valid solution
corr_model<- lm(life_ladder~ . - country -
healthy_life_expectancy_at_birth - year,data=happiness)
w<- model.matrix(corr_model)[,-1];
eigw<- eigen(t(w)%*%w)
eigw$val
corr_model_condition_numbers<-sqrt(eigw$val[1]/eigw$val)
corr_model_condition_numbers

#Example 4 showing Amputation was not valid solution
corr2_model<- lm(life_ladder~ . - country -
healthy_life_expectancy_at_birth - social_support - year,data=happiness)
wn<- model.matrix(corr2_model)[,-1];
eigwn<- eigen(t(wn)%*%wn)
eigwn$val
corr2_model_condition_numbers<-sqrt(eigwn$val[1]/eigwn$val)
corr2_model_condition_numbers
```

32	# VIF Code
	#Compute the first variable VIF to make sure it agrees with vif()
	a<-summary(lm(x[,1]~x[,-1]))$r.squared
	1/(1-a)
	#Will check the first variable VIF computation
	#If VIF > 10, this indicates high collinearity in the model
	#Computes all variables VIF in the model
	library(car)
	vif_model<- vif(model)
	#Initialize a numeric variable
	vif_showing_multicollinearity <- numeric()
	vif_under_threshold <- numeric()
	#Check to see if vif > 10
	for (number in vif_model) {
	  if (number > 10) {
	  vif_showing_multicollinearity <- c(vif_showing_multicollinearity,
	number)
	  }else {
	  vif_under_threshold <- c(vif_under_threshold,number)
	  vif_showing_multicollinearity <-
	c(vif_showing_multicollinearity,"No")
	  }
	  }
	vif_showing_multicollinearity
	vif_under_threshold

33	# Exhaustive Search Adjusted R2, R2, Mallow's Cp
	
	# Adjusted R-squared
	#Initialize a model
	
	model<- lm(life_ladder~ . - country - year,data=happiness)
	summary(model)
	# Load the leaps package
	library(leaps)
	#In order for leaps to run, we must omit values that R previously removed
	itself in calculation (12 observations deleted due to missingness)
	happiness<-na.omit(happiness)
	happiness_subset <- happiness[, c("life_ladder","log_GDP_per_capita",
	"social_support", "generosity", "healthy_life_expectancy_at_birth",
	"freedom_to_make_life_choices","perceptions_of_corruption","positive_affe
	ct", "negative_affect")]
	# Create the predictor matrix (excluding the response variable)
	x <- happiness_subset[,c("log_GDP_per_capita", "social_support",
	"generosity", "healthy_life_expectancy_at_birth",
	"freedom_to_make_life_choices","perceptions_of_corruption","positive_affe
	ct", "negative_affect")]
	# Response variable
	y <- happiness_subset$life_ladder
	# Perform subset selection using leaps
	leaps_model<- leaps(x, y, method = "adjr2")
	best_model_index <- which.max(leaps_model$adjr2)
	best_model_adjr2 <- leaps_model$adjr2[best_model_index]
	best_model_variables <- colnames(x)[leaps_model$which[best_model_index,
	]]
	best_model_index

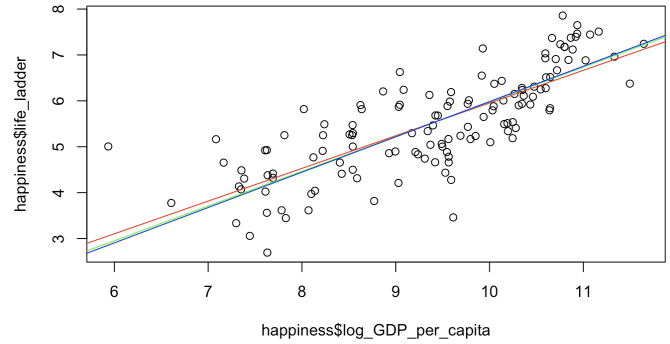
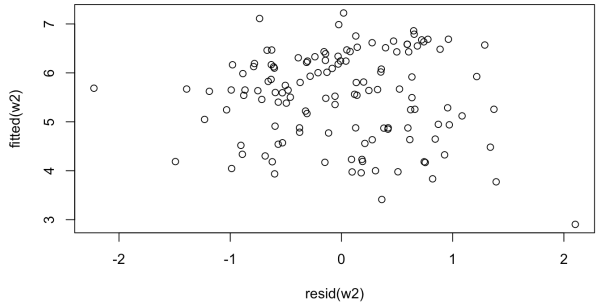
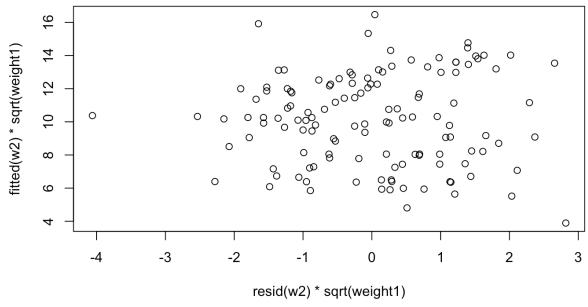
	best_model_adjr2
	best_model_variables
	best_model_1 <- lm(life_ladder ~ . - country - year - generosity,
	data=happiness)
	summary(best_model_1)
	
	# R Squared
	leaps_model_2<- leaps(x, y, method = "r2")
	best_model_2_index <- which.max(leaps_model_2$r2)
	best_model_2_r2 <- leaps_model_2$r2[best_model_2_index]
	best_model_2_variables <-
	colnames(x)[leaps_model_2$which[best_model_2_index,]]
	best_model_2_index
	best_model_2_r2
	best_model_2_variables
	best_model_2 <- lm(life_ladder ~ . - country - year, data=happiness)
	summary(best_model_2)
	# Mallows Cp
	model<- lm(life_ladder~ . - country - year,data=happiness)
	b<- regsubsets(life_ladder~ . - country - year,data=happiness, nbest=1)
	(rs<-summary(b))
	# Graphical Represenation of analysis
	par(mfrow=c(1,2))
	plot(2:9,rs$adjr2,xlab="No. of Parameters", ylab="Adjusted R-Square")
	plot(b,scale="adjr2")
	plot(2:9,rs$cp,xlab="No. of Parameters", ylab="Cp")
	abline(0,1)
	plot(b,scale="Cp")

34	# Exhaustive Search AIC & BIC
	# AIC
	step_model_aic <- step(model, k = 2, direction = "both", trace = 1)
	# BIC
	step_model_bic <- step(model, k = log(nrow(happiness)), direction =
	"both", trace = 1)

35	# Example code of generalized least squares estimator
	ordered_happiness1 = happiness[order(happiness$log_GDP_per_capita),]
	fit1 <- lm(life_ladder~.-country-year, data=ordered_happiness1)
	e <- resid(fit1)
	(rhoest = cor(e[-1],e[-129]))
	V<-matrix(1,129,129)
	V<-rhoest^(abs(row(V)-col(V)))
	C<-chol(V)
	tildey<-solve(t(C))%*% ordered_happiness1$life_ladder
	X<-model.matrix(fit1)
	tildeX<-solve(t(C))%*%X
	fit2<-lm(tildey~tildeX-1)
	e2<-resid(fit2)
	(rhoest2<-cor(e2[-1],e2[-129]))

36	# Output for appendix 35 0.042776 0.004798293

37	# Example code of iteratively weighted least squares estimators
	weight0 = rep(1,129)
	w1 = lm(life_ladder~log_GDP_per_capita, data = happiness, weights = weight0)
	sd1 = lm(abs(w1$resid)~log_GDP_per_capita, data = happiness)
	weight1 = (1/sd1$fitted)^2
	w2 = lm(life_ladder~log_GDP_per_capita, data = happiness, weights = weight1)
	sd2 = lm(abs(w2$resid)~log_GDP_per_capita, data = happiness)
	weight2 = (1/sd2$fitted)^2
	w3 = lm(life_ladder~log_GDP_per_capita, data = happiness, weights = weight2)
	plot(happiness$log_GDP_per_capita, happiness$life_ladder)
	abline(coef(w1), col = "red")
	abline(coef(w2), col = "green")
	abline(coef(w3), col = "blue")
	
	plot(resid(w2), fitted(w2))
	plot(resid(w2)*sqrt(weight1), fitted(w2)*sqrt(weight1))

38	# Output for appendix 37   

39	newmodel <- lm((life_ladder)^1.16 ~ log_GDP_per_capita + social_support + healthy_life_expectancy_at_birth + freedom_to_make_life_choices + generosity + perceptions_of_corruption + positive_affect + negative_affect, data=Data) summary(newmodel) plot(fitted(newmodel), resid(newmodel), xlab = 'Fitted', ylab = 'Residuals') plot(fitted(newmodel), rstudent(newmodel), xlab = 'Fitted', ylab = 'R-Student Residuals') qqnorm(resid(newmodel)) qqline(resid(newmodel))