Laptop Prices in Relation to Technical Specifications – A Bayesian Perspective of Linear Regression via JAGS in R

Andrew J. Otis & Hsing Yu Chen

COMP 4442-2

June 9th , 2022

Table of Contents

EXECUTIVE SUMMARY

There exist two established frameworks regarding the field of statistics, Frequentist and Bayesian. utilizing the Bayesian framework, which results in a distribution**.** As opposed to the Frequentist framework, where a point estimate for variable of interest as a result. Under the Bayesian framework and with data of laptop prices in relation to their specifications, what is the probability of a laptop's market price, given its specifications? Specifically, the model of linear regression under a Bayesian perspective is utilized. Two similar Bayesian models are constructed and explored, one with all relevant predictors and another with all the relevant predictors; excluding brand factors.

The Bayesian regression model with brands included do not fit our data well. However, the Bayesian model with brands excluded did fit the data well, and therefore would result in more valid results/interpretations.

It is possible that the brand portion of the data is oversaturated with one type of response (e.g. majority of responses consisting of the most popular products at the time).

Thus, it is believed the model with brands included could be improved by the removal of brand factors within the model that do not have a significant effect, re-specifying parameters of the prior distribution with industry expert input or wait for more data to be collected.

While the analysis does not answer this question, it does provide a great starting point in the form of a model that fits the data. Follow up studies on the topic, it is recommended that predictions be made using the constructed Bayesian Model or attempt any of the following recommended adjustments to correct the model including "brand" variables.

Data Source and Definitions

The data set can be found and accessed on Kaggle under the "References" section in the category "Data". The data was originally sourced from the web domain "flipkart.com" for a chrome extension application called "Instant Data Scrapper". Luckily, the secondary source utilized had already done a majority of the data cleaning required for analysis, having 896 observations and 34 attributes.

The data consists of factors considered to be relevant when it comes to a laptop's market price, suggested to affect laptop prices, such as company name and owned laptop brands, the price of the laptop when first released and later in product's life, and the hardware that comes with it.

Through exploratory analysis, it was discovered that the response variable being continuous, and all predictor variables are categorical.

Exploratory Analysis

As part of exploratory analysis, two major data transformations are performed. First, the responses of columns are transformed in to multivariate (i.e. 1, 2, 3,…, etc.) form.

Second, another column is added for the logistic transformation of the response variable “latest_price”. This is due to the original form of the response variable produces a distribution skewed towards the left, the result is a response After these transformations, the data set ready for analysis with 896 observations and 35 attributes.

To determine whether or not he had continuous or categorical predictors, histograms of each of them individually were constructed.

Figure 1.1 – Histogram of Non-Transformed Response Variable “latest_price”

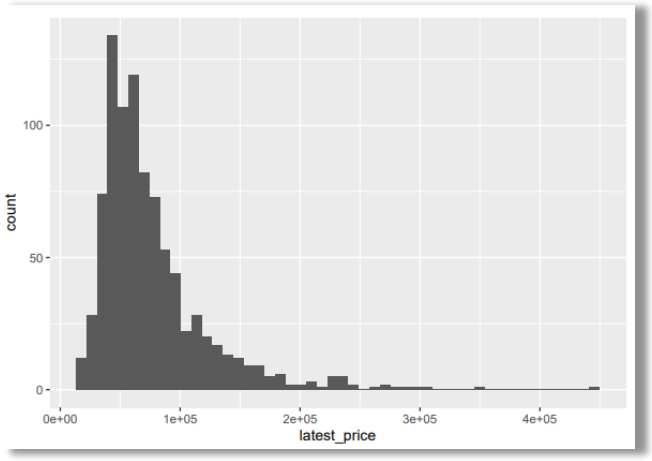


Figure 1.2 – Histogram of Log Transformed Response Variable “latest_price” (a.k.a. “log_price”)

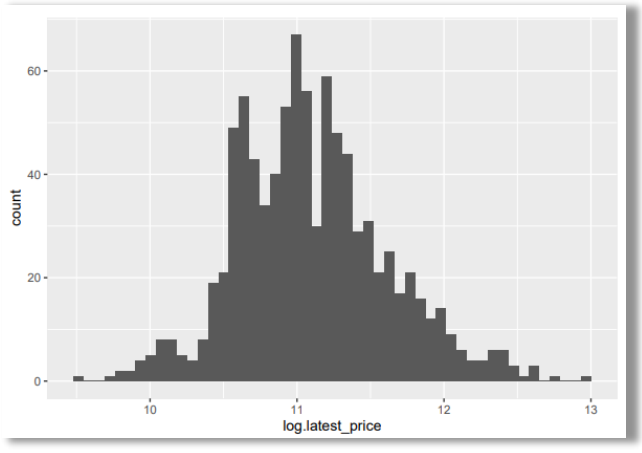


Figure 2.1 – Histogram of Predictor "ram_gb"

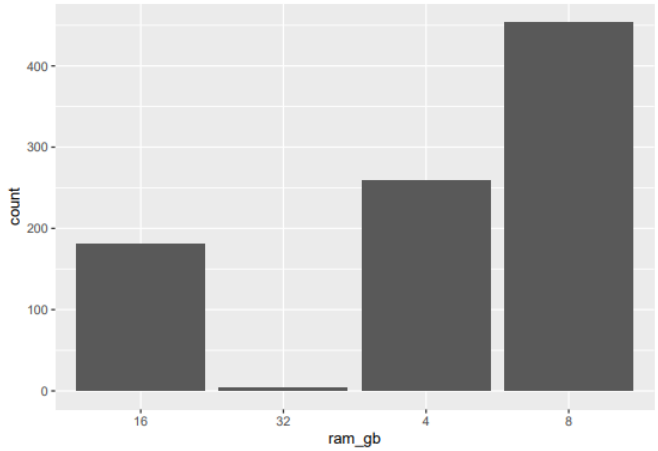


Figure 2.2 – Histogram of Predictor "ssd"

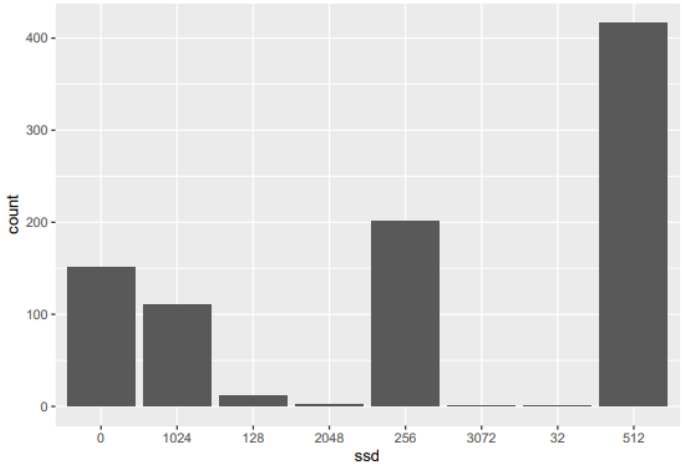
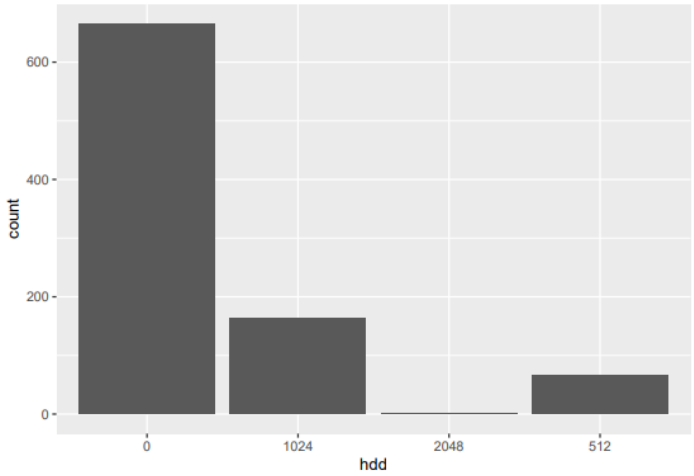


Figure 2.3 – Histogram of Predictor "hdd"



Bayesian Data Analysis, In Principle

Recall, under the Bayesian Framework, the model produces a distribution. How? Bayes Theorem.

$$\boldsymbol{Bayes\ Theorem}\text{: } \boldsymbol{P(A|B)\ \alpha\ P(B|A)P(A)}$$

$Where,$

$P(A|B) \rightarrow Posterior\ Distribution$

$P(B|A) \rightarrow Likelihood\ Distribution, \qquad given\ event\ A\ has\ occured,\ what\ is\ probability\ of\ event\ B?$

$P(A) \rightarrow Prior\ Distribution, \qquad Inital\ guess\ of\ the\ variable\ of\ interest$

The resulting posterior distribution can be interpreted as a report on both the level certainty and uncertainty (i.e. $\pm 0,\ \pm 1,\ \ldots, \pm n\ standard\ deviations$)regarding the probability of an event and model parameters

Bayesian Regression

A linear model can be utilized for data analysis under the Bayesian Framework. To get our Prior Distribution, we will simply run a regular multiple regression. Without industry expert input, this type of prior is commonly referred to as a "non-informative prior".

$$y_i = (\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n) + (\varepsilon_i) \quad , for\ data\ (x_n,\ y_n)$$

$Where,$

$\varepsilon_i \leftarrow Noise$

$\beta_0 \leftarrow y - intercept$

To get our Likelihood Distribution the model runs thousands of samples, each with their own likelihood parameter, which together create a distribution of the likelihood parameters.

These calculations can be accomplished by utilizing the JAGS package in statistical software like R

Applied Bayesian Regression

Through the JAGS R package, the Markov chain Monte Carlo (MCMC) algorithm is applied

Imagine a target distribution you want to analyze, have data on it, but can no longer collect the data? The MCMC algorithm would be a viable solution. Thus, MCMC is simply an algorithm for sampling from a distribution.

One of the most common uses of the MCMC algorithm is to sample the posterior probability distribution of some model in Bayesian inference

Regarding Analysis, four Markov Chains are set up for the predictors, “brand”, “ram_gb”, “hdd”, “ssd”

With the formal equation

$$\log\_price = \left(\beta_0 + \text{brand} * x_1 + \text{ram}_{\text{gb}} * \text{x}_2 + \text{hdd} * \text{x}_3 + ssd * x_4\right) \quad , for\ data\ (x_n,\ y_n)$$

Data Satisfaction

For results of the created Bayesian model to be considered valid, four data requirements must be met by the Bayesian model’s predicted values to determine to see how well it fits the data.

The four data considerations are that there exists a linear relationship between predictors and response, homoscedasticity across predictor values, independence in responses (i.e. residuals demonstrate random dispersal across a horizontal trendline), and normality (i.e. a well fit QQ plot of the residuals plotted against predicted values).

Supporting Visualizations

Table 1.1 - Prior Distribution Results (Brand Factor included)

```
##
## Call:
## lm(formula = log.latest_price ~ ., data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.04228 -0.19797 -0.02481  0.16757  1.48634
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    10.82797    0.08153 132.816  < 2e-16 ***
## brandALIENWARE  1.38371    0.17636   7.846 1.27e-14 ***
## brandAPPLE      0.48734    0.07855   6.204 8.53e-10 ***
## brandASUS      -0.11435    0.04984  -2.294 0.022020 *
## brandAvita     -0.20211    0.09192  -2.199 0.028158 *
## brandDELL      -0.08696    0.05261  -1.653 0.098676 .
## brandHP        -0.08797    0.05326  -1.652 0.098954 .
## brandiball     -1.33530    0.33688  -3.964 7.99e-05 ***
## brandInfinix   -0.43477    0.17271  -2.517 0.012003 *
## brandlenovo     0.54742    0.20068   2.728 0.006505 **
## brandLenovo    -0.09873    0.05286  -1.868 0.062136 .
## brandLG         0.31098    0.15705   1.980 0.048004 *
## brandMi        -0.25140    0.23959  -1.049 0.294322

## brandMICROSOFT  0.51543    0.23002   2.241 0.025295 *
## brandMSI        0.04856    0.06470   0.751 0.453067
## brandNokia     -0.39540    0.17245  -2.293 0.022098 *
## brandrealme    -0.12936    0.17267  -0.749 0.453948
## brandRedmiBook -0.28919    0.19713  -1.467 0.142741
## brandSAMSUNG   -0.20628    0.33645  -0.613 0.539963
## brandSmartron  -0.07215    0.21207  -0.340 0.733785
## brandVaio      -0.21399    0.15617  -1.370 0.170956
## ram_gb32        0.13201    0.23983   0.550 0.582172
## ram_gb4        -0.40891    0.03627 -11.275  < 2e-16 ***
## ram_gb8        -0.23916    0.03294  -7.260 8.65e-13 ***
## ssd1024         1.14024    0.07083  16.098  < 2e-16 ***
## ssd128          0.43806    0.13248   3.307 0.000984 ***
## ssd2048         1.40215    0.24569   5.707 1.58e-08 ***
## ssd256          0.26569    0.05174   5.135 3.49e-07 ***
## ssd3072         2.28541    0.33959   6.730 3.09e-11 ***
## ssd32          -0.67217    0.41514  -1.619 0.105778
## ssd512          0.61682    0.06172   9.994  < 2e-16 ***
## hdd1024         0.22380    0.04761   4.701 3.02e-06 ***
## hdd2048         0.53813    0.33801   1.592 0.111736
## hdd512          0.46233    0.07210   6.412 2.36e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3318 on 862 degrees of freedom
## Multiple R-squared:  0.599,  Adjusted R-squared:  0.5837
## F-statistic: 39.02 on 33 and 862 DF,  p-value: < 2.2e-16
```

Table 1.2 – Posterior Distribution Results (Brand Factor included)

```
##
## Iterations = 8001:23000
## Thinning interval = 1
## Number of chains = 1


## Sample size per chain = 15000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                      Mean       SD  Naive SE Time-series SE
## beta.brand[10]  1.304e-01 31.58900 0.2579231      0.2579231
## beta.brand[11] -1.635e-01 31.75790 0.2593021      0.2593021
## beta.brand[12] -4.568e-01 31.56349 0.2577148      0.2577148
## beta.brand[13] -7.314e-02 31.57584 0.2578157      0.2539222
## beta.brand[14] -2.460e-01 31.56259 0.2577075      0.2577075
## beta.brand[15] -4.205e-01 31.34406 0.2559232      0.2559232
## beta.brand[16] -1.843e-01 31.58092 0.2578571      0.2578571
## beta.brand[17]  8.253e-02 32.21096 0.2630014      0.2669741
## beta.brand[18]  1.440e-01 31.89220 0.2603988      0.2640875
## beta.brand[19] -2.278e-01 31.72398 0.2590252      0.2623149
## beta.brand[1]   0.000e+00  0.00000 0.0000000      0.0000000
## beta.brand[2]   2.229e-02 31.58031 0.2578521      0.2578521
## beta.brand[3]  -2.489e-01 31.60955 0.2580909      0.2580909
## beta.brand[4]   2.324e-01 31.65698 0.2584782      0.2536062
## beta.brand[5]   2.821e-02 31.57080 0.2577745      0.2577745
## beta.brand[6]   1.005e-01 31.61906 0.2581686      0.2581686
## beta.brand[7]  -3.138e-02 31.54855 0.2575929      0.2622506
## beta.brand[8]  -1.507e-01 31.34788 0.2559543      0.2559543
## beta.brand[9]   6.492e-05 31.50632 0.2572480      0.2539852
## beta.hdd[1]     0.000e+00  0.00000 0.0000000      0.0000000
## beta.hdd[2]     1.803e-01  0.05069 0.0004139      0.0022697
## beta.hdd[3]     4.619e-01  0.37017 0.0030224      0.0046024
## beta.hdd[4]     5.451e-01  0.07542 0.0006158      0.0032752
## beta.ram_gb[1]  0.000e+00  0.00000 0.0000000      0.0000000
## beta.ram_gb[2]  4.025e-01  0.26254 0.0021436      0.0031287
## beta.ram_gb[3] -4.911e-01  0.03939 0.0003216      0.0011315
## beta.ram_gb[4] -2.527e-01  0.03600 0.0002939      0.0009988
## beta.ssd[1]     0.000e+00  0.00000 0.0000000      0.0000000
## beta.ssd[2]     1.108e+00  0.07540 0.0006157      0.0042029
## beta.ssd[3]     5.478e-01  0.12237 0.0009992      0.0032622
## beta.ssd[4]     1.338e+00  0.27037 0.0022076      0.0046263
## beta.ssd[5]     2.441e-01  0.05639 0.0004605      0.0026966
## beta.ssd[6]     2.168e+00  0.37115 0.0030304      0.0052575
## beta.ssd[7]    -1.037e+00  0.45562 0.0037202      0.0059422
## beta.ssd[8]     5.600e-01  0.06662 0.0005440      0.0038322
## beta0           1.083e+01  0.07352 0.0006003      0.0047942
```

```
## 2. Quantiles for each variable:
##
##                      2.5%      25%      50%     75%   97.5%
## beta.brand[10] -60.64205 -21.7612  0.18835 21.8100 62.3224
## beta.brand[11] -62.09264 -21.3495 -0.36896 21.0104 63.2653
## beta.brand[12] -61.78271 -22.1448 -0.09921 20.6179 60.6227
## beta.brand[13] -61.97787 -21.3050 -0.18752 21.4385 61.5337
## beta.brand[14] -61.81379 -21.4454 -0.27259 21.0986 61.5894
## beta.brand[15] -62.32082 -21.4856 -0.18614 20.9012 60.4430
## beta.brand[16] -61.44835 -21.4306  0.00365 20.7785 62.0441
## beta.brand[17] -62.84561 -21.5619  0.10729 21.7497 63.1752

## beta.brand[18] -63.26452 -21.2250  0.14058 21.8235 61.9386
## beta.brand[19] -62.05492 -21.9782 -0.10861 21.2982 61.4872
## beta.brand[1]    0.00000   0.0000  0.00000  0.0000  0.0000
## beta.brand[2]  -62.33308 -21.0368  0.02044 20.9577 63.1975
## beta.brand[3]  -61.65073 -21.8953 -0.12976 21.0872 61.8293
## beta.brand[4]  -61.72233 -20.9590 -0.09551 21.7605 62.3341
## beta.brand[5]  -61.65817 -21.4012  0.13616 21.3258 62.0973
## beta.brand[6]  -60.90135 -21.1293 -0.29416 21.4193 63.2433
## beta.brand[7]  -61.81092 -21.5836  0.03053 21.1775 61.7586
## beta.brand[8]  -60.63262 -21.4982 -0.07979 21.1368 60.4678
## beta.brand[9]  -61.36038 -21.1394  0.06924 21.2591 62.4410
## beta.hdd[1]      0.00000   0.0000  0.00000  0.0000  0.0000
## beta.hdd[2]      0.08088   0.1464  0.18024  0.2143  0.2787
## beta.hdd[3]     -0.26224   0.2138  0.46124  0.7124  1.1814
## beta.hdd[4]      0.39745   0.4935  0.54559  0.5963  0.6904
## beta.ram_gb[1]   0.00000   0.0000  0.00000  0.0000  0.0000
## beta.ram_gb[2]  -0.10898   0.2222  0.40316  0.5786  0.9204
## beta.ram_gb[3]  -0.56846  -0.5176 -0.49087 -0.4652 -0.4140
## beta.ram_gb[4]  -0.32478  -0.2769 -0.25291 -0.2285 -0.1823
## beta.ssd[1]      0.00000   0.0000  0.00000  0.0000  0.0000
## beta.ssd[2]      0.96408   1.0568  1.10718  1.1599  1.2563
## beta.ssd[3]      0.30870   0.4654  0.54714  0.6314  0.7850
## beta.ssd[4]      0.80112   1.1580  1.33737  1.5188  1.8689
## beta.ssd[5]      0.13419   0.2065  0.24379  0.2817  0.3548
## beta.ssd[6]      1.43359   1.9220  2.16687  2.4095  2.8942
## beta.ssd[7]     -1.92713  -1.3405 -1.03629 -0.7291 -0.1490
## beta.ssd[8]      0.43217   0.5157  0.55976  0.6054  0.6910
## beta0           10.68880  10.7841 10.83351 10.8840 10.9721
```

Table 2.1 - Prior Distribution Results (Brand Factor NOT included)

```
##
## Call:
## lm(formula = log.latest_price ~ ., data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.34144 -0.21654 -0.02018  0.18700  1.62656
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.83122    0.07417 146.031  < 2e-16 ***
## ram_gb32     0.40272    0.26119   1.542 0.123463
## ram_gb4     -0.49079    0.03905 -12.568  < 2e-16 ***
## ram_gb8     -0.25245    0.03576  -7.060 3.37e-12 ***
## ssd1024      1.10996    0.07617  14.573  < 2e-16 ***

## ssd128       0.54952    0.12136   4.528 6.77e-06 ***
## ssd2048      1.34173    0.26885   4.991 7.25e-07 ***
## ssd256       0.24534    0.05653   4.340 1.59e-05 ***
## ssd3072      2.16783    0.37291   5.813 8.56e-09 ***
## ssd32       -1.03409    0.45403  -2.278 0.022989 *
## ssd512       0.56173    0.06681   8.408  < 2e-16 ***
## hdd1024      0.18120    0.05099   3.553 0.000401 ***
## hdd2048      0.46122    0.37131   1.242 0.214510
## hdd512       0.54710    0.07609   7.190 1.38e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3655 on 882 degrees of freedom
## Multiple R-squared:  0.5021, Adjusted R-squared:  0.4948
## F-statistic: 68.43 on 13 and 882 DF,  p-value: < 2.2e-16
```

Table 2.2 - Posterior Distribution Results (Brand Factor NOT included)

```
##
## Iterations = 8001:23000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 15000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                    Mean      SD  Naive SE Time-series SE
## beta.hdd[1]      0.0000 0.00000 0.0000000       0.000000
## beta.hdd[2]      0.1763 0.05101 0.0004165       0.002435
```

```
## beta.hdd[3]      0.4542 0.37116 0.0030305        0.004868
## beta.hdd[4]      0.5399 0.07658 0.0006252        0.003393
## beta.ram_gb[1]   0.0000 0.00000 0.0000000        0.000000
## beta.ram_gb[2]   0.4005 0.26197 0.0021390        0.003064
## beta.ram_gb[3]  -0.4936 0.03990 0.0003258        0.001127
## beta.ram_gb[4]  -0.2545 0.03659 0.0002987        0.001008
## beta.ssd[1]      0.0000 0.00000 0.0000000        0.000000
## beta.ssd[2]      1.1012 0.07650 0.0006246        0.004544
## beta.ssd[3]      0.5419 0.12066 0.0009852        0.003575
## beta.ssd[4]      1.3347 0.27123 0.0022146        0.004696
## beta.ssd[5]      0.2393 0.05644 0.0004608        0.002822
## beta.ssd[6]      2.1591 0.37624 0.0030720        0.005062
## beta.ssd[7]     -1.0392 0.46147 0.0037679        0.006439
## beta.ssd[8]      0.5535 0.06688 0.0005460        0.003943
## beta0           10.8411 0.07489 0.0006114        0.005149
##
## 2. Quantiles for each variable:
##
##                    2.5%     25%     50%     75%   97.5%
## beta.hdd[1]     0.00000  0.0000  0.0000  0.0000  0.0000
## beta.hdd[2]     0.07837  0.1416  0.1755  0.2104  0.2791
## beta.hdd[3]    -0.27379  0.2040  0.4525  0.7011  1.1790
## beta.hdd[4]     0.39068  0.4886  0.5390  0.5908  0.6919
## beta.ram_gb[1]  0.00000  0.0000  0.0000  0.0000  0.0000
## beta.ram_gb[2] -0.11450  0.2266  0.4013  0.5754  0.9145
## beta.ram_gb[3] -0.57174 -0.5207 -0.4934 -0.4668 -0.4159
## beta.ram_gb[4] -0.32664 -0.2793 -0.2541 -0.2296 -0.1838
## beta.ssd[1]     0.00000  0.0000  0.0000  0.0000  0.0000
## beta.ssd[2]     0.95384  1.0495  1.0990  1.1514  1.2550
## beta.ssd[3]     0.30438  0.4611  0.5414  0.6235  0.7792
## beta.ssd[4]     0.79760  1.1519  1.3376  1.5177  1.8611
## beta.ssd[5]     0.12965  0.2015  0.2388  0.2767  0.3532
## beta.ssd[6]     1.42220  1.9064  2.1599  2.4120  2.8949
## beta.ssd[7]    -1.95187 -1.3511 -1.0366 -0.7286 -0.1324
## beta.ssd[8]     0.42501  0.5082  0.5519  0.5972  0.6903
## beta0          10.68367 10.7921 10.8443 10.8900 10.9863
```

Figure 3.1 – Normality Diagnostic Plot (Brand Factor Included)

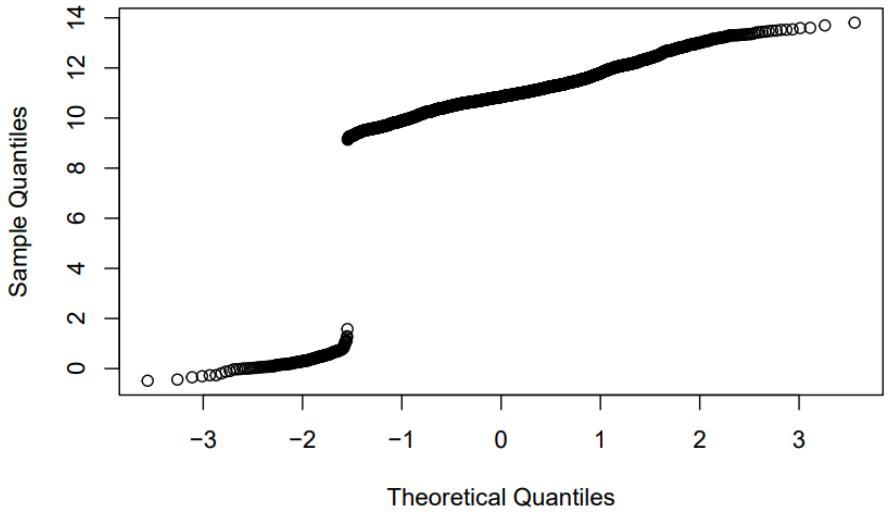


Figure 3.2 – Independence & Homoscedasticity Diagnostic Plot (Brand Factor Included)

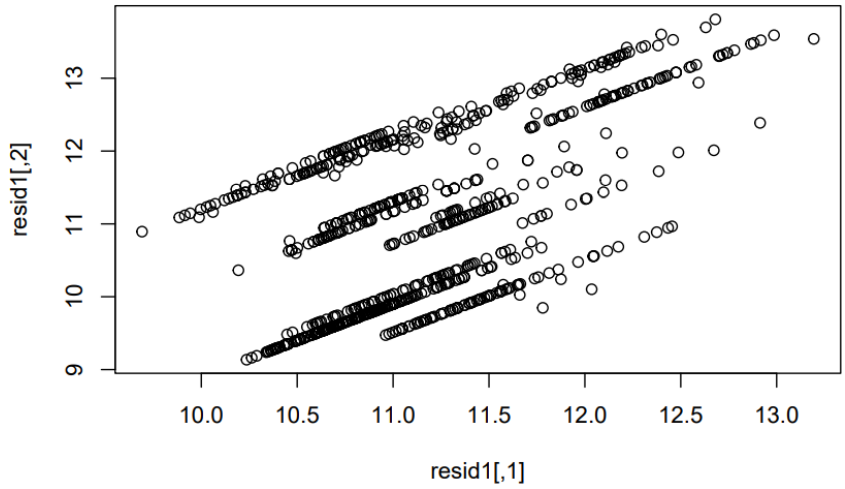


Figure 3.3 – Linearity Diagnostic Plot (Brand Factor Included)

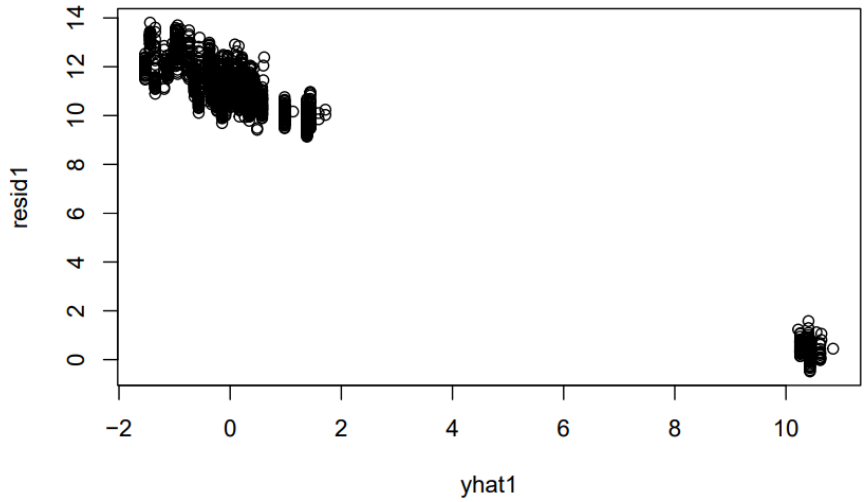


Figure 4.1 – Normality Diagnostic Plot (Brand Factor NOT Included)

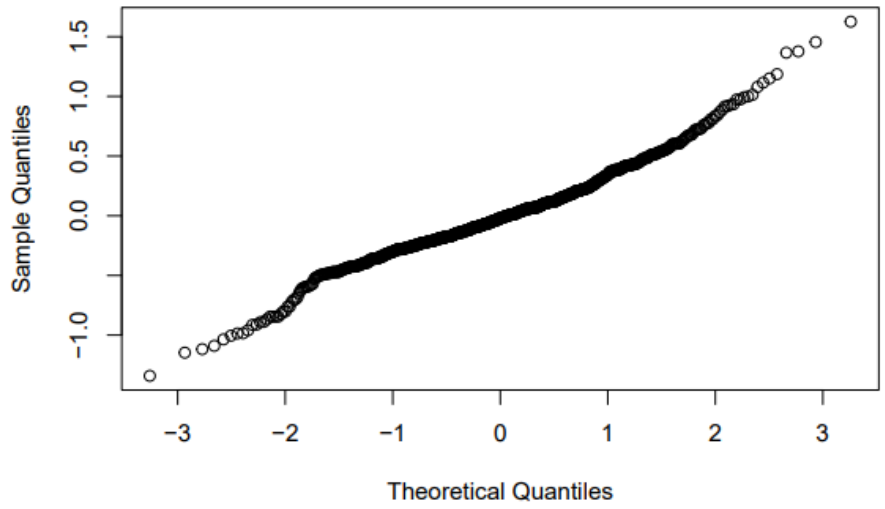


Figure 4.2 – Independence & Homoscedasticity Diagnostic Plot (Brand Factor NOT Included)

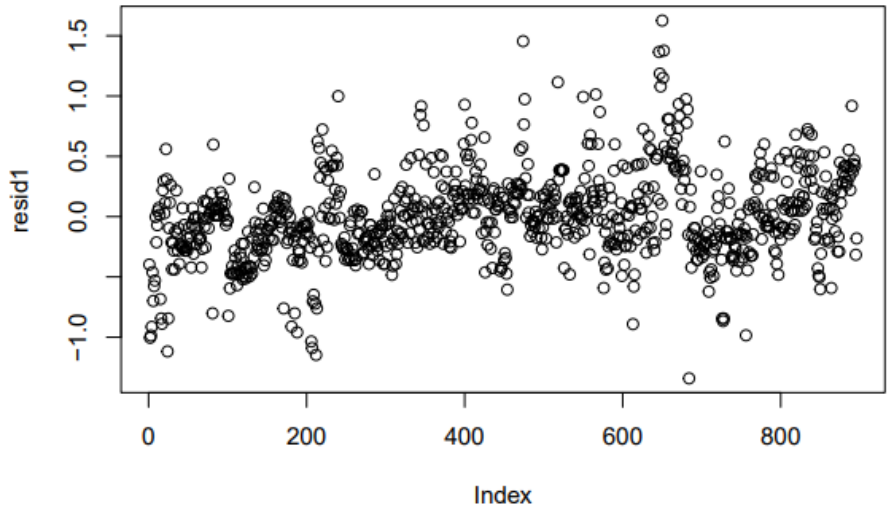
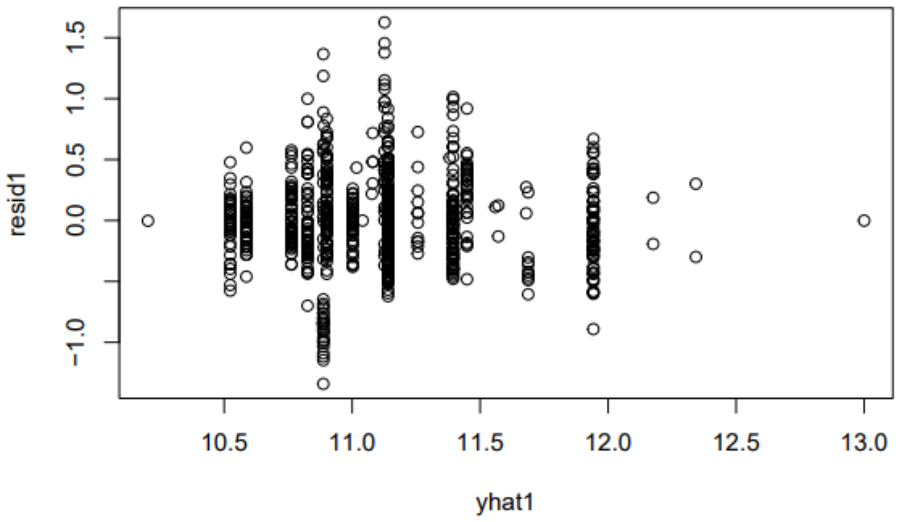


Figure 4.3 – Linearity Diagnostic Plot (Brand NOT Factor Included)



REFERENCES

Videos:

1. *Bayesian Modeling with R and Stan (Reupload)*. (2018, November 15). [Video]. YouTube.
2. *Bayesian Statistics - Regression with JAGS Part 3*. (2021, May 24). [Video]. YouTube.

Literature/Academic:

1. A. (2022, May 16). *Bayesian Statistics Overview and your first Bayesian Linear Regression Model*. Towardsdatascience. Retrieved June 6, 2022.
2. Durso, C. (2022), COMP4442-2: Advanced Probability and Statistics 2.
3. Falster, D. F. R. (2013, June 10). *Markov Chain Monte Carlo - Nice R Code*. Github. Retrieved June 1, 2022.
4. Karajannis, N. (2017, November 20). *RPubs - Bayes Regression using JAGS*. Rpubs.Com. Retrieved May 28, 2022.
5. Htoon, K. S. (2021, December 13). *Log Transformation: Purpose and Interpretation - Kyaw Saw Htoon*. Medium. Retrieved June 6, 2022.
6. Sbnfk. (2019). *mcmc_diagnostics.utf8.md*. Github. Retrieved June 2, 2022.

Data:

1. *Laptop Specs and latest price*. (2022, April 3). Kaggle.