Predicting the Weights of Bananas

Ethan Scott

19/01/2023

Abstract

The goal of this report is to predict the weight of a banana using it's radius and length. 90 bananas were collected and weighed to the nearest gram, their radius and length were recorded to the nearest millimeter. Using 63 of the bananas data, 3 models were created to predict the weight of the bananas. The remaining 27 bananas were used to test these models to find the prediction accuracy, a 95% confidence interval was also found for the density of the bananas. After testing and comparing the results, the best model created to predict banana accuracy is a model that takes both the radius and length as predictors.

Introduction

With bananas being grown in more than 150 countries and 105 million tonnes of bananas being produced each year, bananas are a very popular fruit in many markets around the world (Banana Link, 2023). Since many bananas are sold based off of their weight, the weight of a banana is a very important characteristic. This study was done to try and predict the weights of bananas using linear models, given two physical characteristics of the bananas. The goal of this study is to see how accurately a bananas weight can be predicted using measurements that are easy enough for anybody to collect.

The remainder of this report will show how this was accomplished. Firstly, it will cover the materials and methods used to collect and clean the data. Next, the models created and the results of those models will be discussed. Finally, the results of the models, the confidence interval for density and the shortcomings of project will be discussed.

Materials and Methods

The materials needed to conduct this experiment are:

- 90 bananas of various sizes but similar composition,
- A scale that can calculate weight to the nearest gram,
- A ruler that can measure to the nearest millimeter.

Once these materials have been acquired we can begin to collect our data by:

1. Weighing a single banana on a scale and recording the weight to the nearest gram
2. Measure; to the nearest millimeter, the circumference of the banana near the middle of the banana
3. Divide the circumference by 2π to find the radius, round this to the nearest millimeter and record the value

4. Measure the curved length of the banana and record the value to the nearest millimeter

For this study each person collected and measured 6 of our own bananas and then the data was consolidated into a single data-set. Once this data was collected, each banana was given a banana ID. With this, the data was organized with the banana's ID in the first column, weight in the second column, the corresponding radius in the third column and the length of the banana in the fourth and final column. After consolidating the data, it needed to be cleaned. To clean the data I had to make sure that each column contained the appropriate information since some measurements for radius were in the length column and vice versa. There were also some data in the radius column that seemed to be much to larger so they needed to be divided by 2 since they were only divided by π when collected and not 2π. Data that was recorded with decimal points was also rounded to the nearest integer.

Once all of the data was collected and cleaned three models were created to predict the weight of the bananas. Before creating these models we split the data into a training and testing set, 63 of the bananas were used in the training set and the remaining 27 bananas were put into the testing set. Then we created the models, the three models were $log(W) = \beta_0 + \beta_1 log(r) + \epsilon$, $log(W) = \beta_0 + \beta_2 log(l) + \epsilon$ and $log(W) = \beta_0 + \beta_1 log(r) + \beta_2 log(l) + \epsilon$. After all of the models were created the Mean Absolute Error (MAE) and Mean Percentage Absolute Error (MPAE) were calculated in order to evaluate the accuracy of the models. Finally, a 95% confidence interval was created in order to estimate the density of the kind of bananas used in our models.

Results

To predict the weight of the bananas we must build our models to predict the weight of the bananas. To do this we will build 3 linear models where weight is our response variable. Our first model will be $log(W) = \beta_0 + \beta_1 log(r) + \epsilon$, where W is the weight of the bananas and r is the radius of the bananas. The second model will be $log(W) = \beta_0 + \beta_2 log(l) + \epsilon$, where l is the length of the bananas. Our third and final model will be $log(W) = \beta_0 + \beta_1 log(r) + \beta_2 log(l) + \epsilon$. Using 63 of our bananas as training data for our models, we could create all 3 models and then use the other 27 bananas to test our models and pick which model would be best. Using the 27 bananas as testing data, the Mean Absolute Error (MAE) and Mean Percentage Absolute Error (MPAE) were calculated for each of the models to compare the prediction accuracy. Below is a table showing each model with the respective MAE and MPAE.

	Model #1	Model #2	Model #3
MAE	18.63718	17.45108	14.46446
MPAE	0.1037255	0.09779171	0.08202072

From this table we can see that the MAE and MPAE are the lowest in Model #3, because of this the model that is best for predicting the weight of the bananas is Model #3.

For our 95% confidence interval on the density of the bananas, the lower bound is 0.0006272348 g/mm^3 and the upper bound is 0.0007010319 g/mm^3.

Discussion

Given these results, we can see that using the third model we achieve the best results for predicting the weights of the bananas. This means that using the radius and length of the bananas as factors we can achieve the most accurate prediction for the weight of the banana. One of the shortcomings of the project is the data collection process. Since there were multiple people recording the data for 6 bananas each, the measurements could have been taken in different ways by each person. The devices used to measure and weight each banana could also have inconsistent measures of accuracy, since each person used different scales and rulers, each scale and ruler could give different measurements than the others. Another limitation of the project is data collection process, each value that was recorded was rounded to the nearest integer which could cause errors in the model building. There were also assumptions about data being entered into the wrong columns of the dataset and those numbers were changed, these assumptions could have been misguided and therefore this could be a shortcoming of this project.

Conclusion

In conclusion, for this project the weight, radius and length of 90 bananas were recorded with weight being recorded to the nearest gram and radius and length being recorded to the nearest millimeter. Given this data, the best model used to predict the weight of the bananas is the model that takes the length and radius as factors. Finally, the density of the bananas is estimated to be between 0.0006272348 g/mm^3 and 0.0007010319 g/mm^3 with 95% confidence.

Acknowledgements

Dr. Gemai Chen

References

R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing,

Vienna, Austria. URL https://www.R-project.org/.

Wickham H, Bryan J (2022). _readxl: Read Excel Files_. R package version 1.4.1,

https://CRAN.R-project.org/package=readxl.

Banana Link. (2023, February 2). *All About Bananas | Producers, Where They're Grown & Why They Matter*. https://www.bananalink.org.uk/all-about-bananas/

Appendix

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.2.2
```

```
data = read_excel("C:\\Users\\ethan\\Downloads\\Stat 517\\Project #2\\Cleaned Data.xlsx")
data
```

```
## # A tibble: 90 x 4
##       ID ‘Weight (g)‘ ‘Radius (mm)‘ ‘Length (mm)‘
##    <dbl>        <dbl>         <dbl>         <dbl>
##  1     1          189            20           236
##  2     2          200            20           241
##  3     3          199            20           240
##  4     4          190            19           236
##  5     5          194            19           240
##  6     6          175            19           223
##  7     7          153            19           247
##  8     8          148            18           238
##  9     9          154            19           251
## 10    10          152            18           239
## # ... with 80 more rows
```

```
sub.vec =c(11,12,16,20,22,24,27,30,33,34,36,40,44,48,52,53,56,58,60,67,69,71,72,80,87,88,89)
test = data[data$ID %in% sub.vec,]
train = data[-sub.vec,]
weight = train$`Weight (g)`
radius = train$`Radius (mm)`
length = train$`Length (mm)`
```

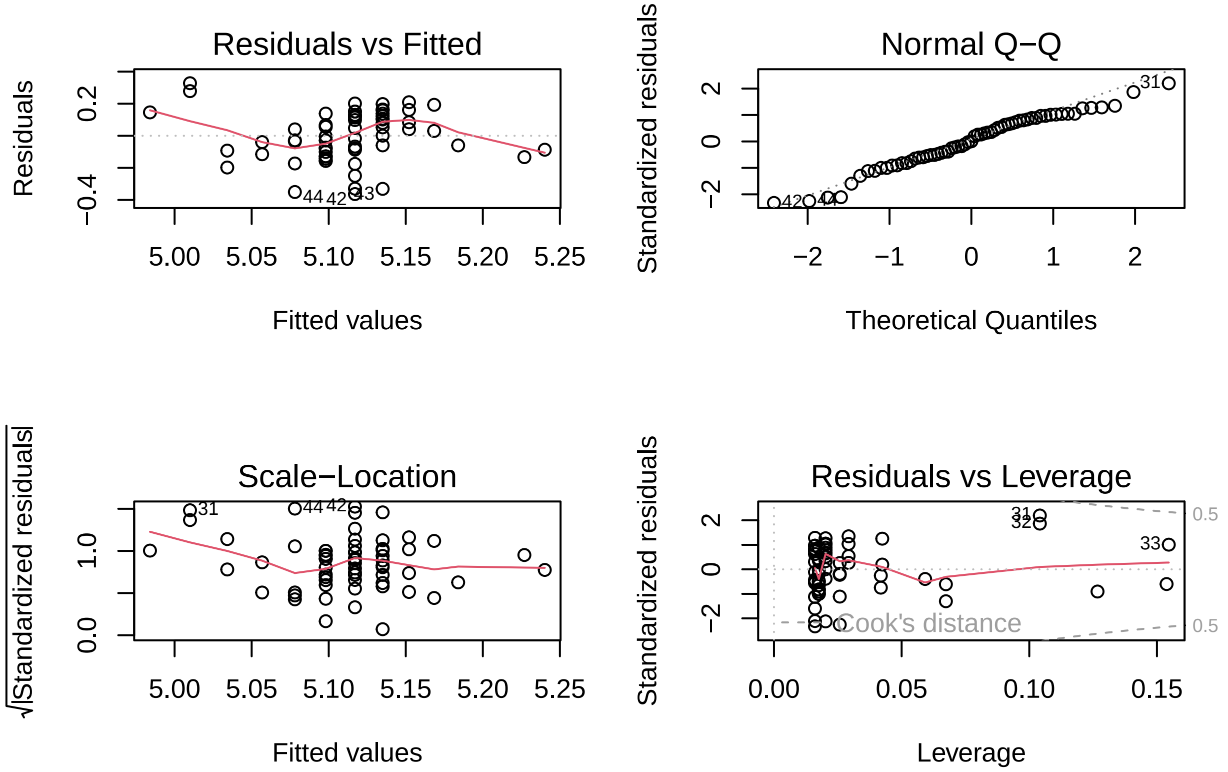
```
model_1 = lm(log(weight)~log(radius))
summary(model_1)
```

```
##
## Call:
## lm(formula = log(weight) ~ log(radius))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3634 -0.0967  0.0008  0.1274  0.3276
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.0850     0.4703   8.686 2.93e-12 ***
## log(radius)   0.3505     0.1603   2.186   0.0326 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.1575 on 61 degrees of freedom
## Multiple R-squared:  0.07267,    Adjusted R-squared:  0.05746
## F-statistic:  4.78 on 1 and 61 DF,  p-value: 0.03264
```

```
coefficients(model_1)
```

```
## (Intercept) log(radius)
##   4.0850240   0.3504917
```

```
par(mfrow = c(2,2))
plot(model_1)
```



```
fitted = rep(0,27)
for(i in 1:27){
  fitted[i] = 4.0850240+0.3504917*log(test$`Radius (mm)`[i])
}

n=27
sum=0
#loop for MAE
for (i in 1:27){
  sum = abs(test$`Weight (g)`[i] - exp(fitted[i])) + sum
}
MAE_1 = sum/n
MAE_1
```

```
## [1] 18.63718
```

```
sum=0
#loop for MPAE
for (i in 1:27){
  sum = abs(test$`Weight (g)`[i] - exp(fitted[i]))/abs(test$`Weight (g)`[i]) + sum
}
MPAE_1 = sum/n
MPAE_1
```

```
## [1] 0.1037255
```

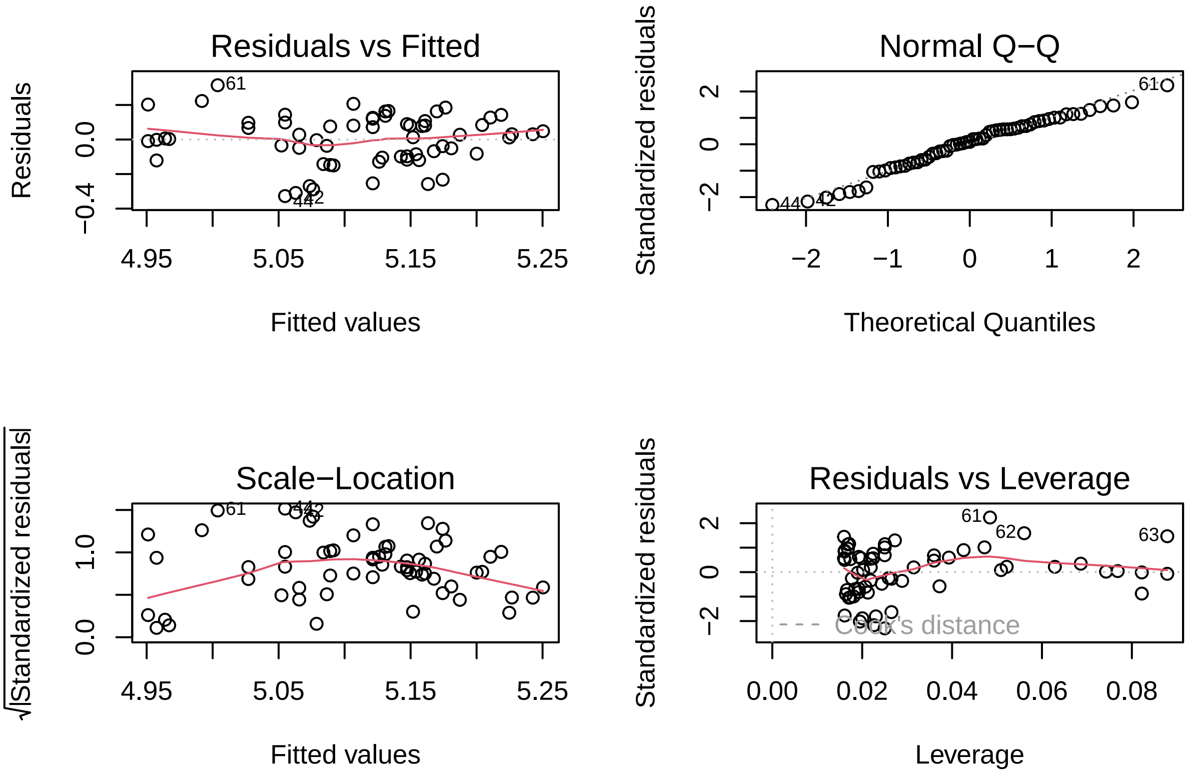
```
model_2 = lm(log(weight)~log(length))
summary(model_2)
```

```
##
## Call:
## lm(formula = log(weight) ~ log(length))
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.32748 -0.09832  0.01292  0.09766  0.31429 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   2.0099     0.7453   2.697  0.00904 ** 
## log(length)   0.5695     0.1368   4.164  0.00010 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1443 on 61 degrees of freedom
## Multiple R-squared:  0.2213, Adjusted R-squared:  0.2085 
## F-statistic: 17.34 on 1 and 61 DF,  p-value: 1e-04
```

```
coefficients(model_2)
```

```
## (Intercept) log(length) 
##   2.0098833   0.5694631
```

```
par(mfrow = c(2,2))
plot(model_2)
```



```
fitted = rep(0,27)
for(i in 1:27){
  fitted[i] = 2.0098833+0.5694631*log(test$`Length (mm)`[i])
}

n=27
sum=0
#loop for MAE
for (i in 1:27){
  sum = abs(test$`Weight (g)`[i] - exp(fitted[i])) + sum
}
MAE_2 = sum/n
MAE_2
```

```
## [1] 17.45108
```

```
sum=0
#loop for MPAE
for (i in 1:27){
  sum = abs(test$`Weight (g)`[i] - exp(fitted[i]))/abs(test$`Weight (g)`[i]) + sum
}
MPAE_2 = sum/n
MPAE_2
```

```
## [1] 0.09779171
```

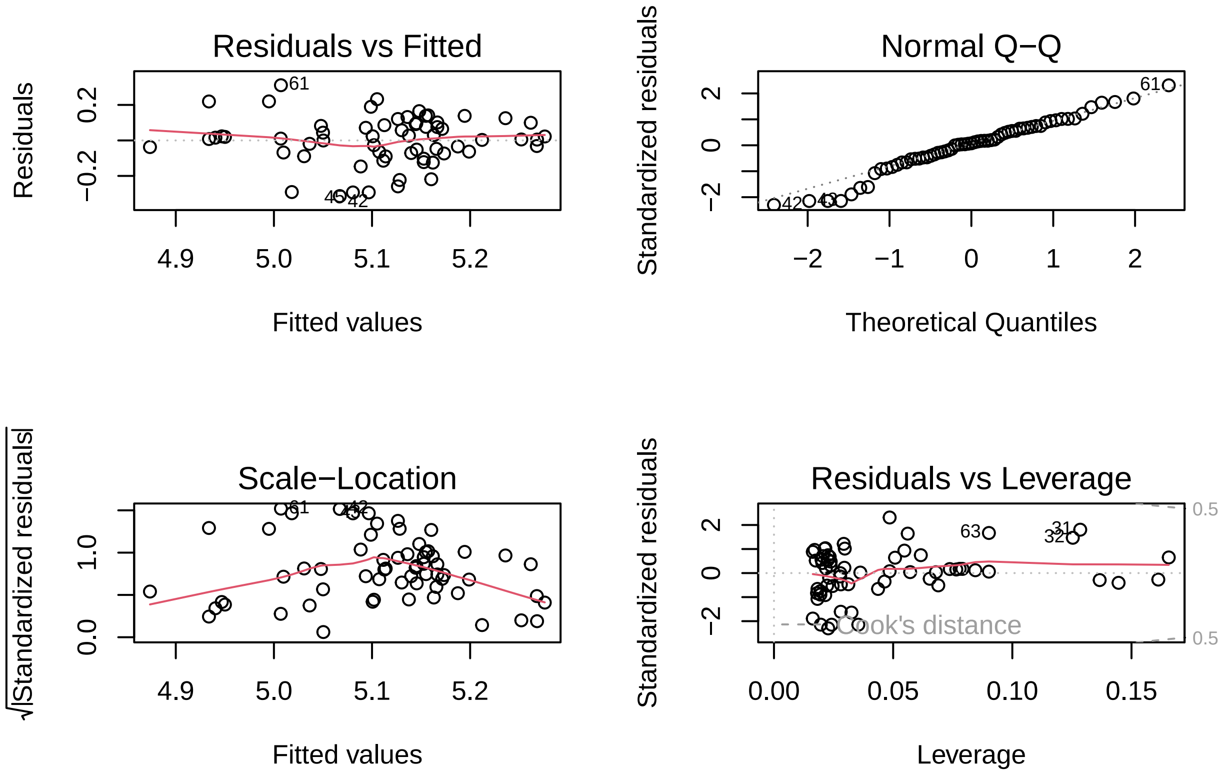
```
model_3 = lm(log(weight)~log(radius) + log(length))
summary(model_3)
```

```
##
## Call:
## lm(formula = log(weight) ~ log(radius) + log(length))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.31355 -0.06957  0.01030  0.08891  0.31096
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.8916     0.8316   1.072   0.2880
## log(radius)   0.3659     0.1404   2.605   0.0116 *
## log(length)   0.5779     0.1307   4.420 4.21e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1379 on 60 degrees of freedom
## Multiple R-squared:  0.3004, Adjusted R-squared:  0.2771
## F-statistic: 12.88 on 2 and 60 DF,  p-value: 2.212e-05
```

```
coefficients(model_3)
```

```
## (Intercept) log(radius) log(length)
##   0.8916333   0.3658574   0.5778951
```

```
par(mfrow = c(2,2))
plot(model_3)
```



```r
fitted = rep(0,27)
for(i in 1:27){
  fitted[i] = 0.8916333+0.3658574*log(test$`Radius (mm)`[i])+0.5778951*log(test$`Length (mm)`[i])
}

n=27
sum=0
#loop for MAE
for (i in 1:27){
  sum = abs(test$`Weight (g)`[i] - exp(fitted[i])) + sum
}
MAE_3 = sum/n
MAE_3
```

```
## [1] 14.46446
```

```r
sum=0
#loop for MPAE
for (i in 1:27){
  sum = abs(test$`Weight (g)`[i] - exp(fitted[i]))/abs(test$`Weight (g)`[i]) + sum
}
MPAE_3 = sum/n
MPAE_3
```

```
## [1] 0.08202072
```

```
#Confidence interval of density of bananas
#log(Density) = log(Weight) - log(pi) - 2log(r) - log(l)
density = list()

for(i in 1:90){
  density[i] = log(data$`Weight (g)`[i])-log(pi)-2*log(data$`Radius (mm)`[i])-log(data$`Length (mm)`[i]
}
density = unlist(density)
density = exp(density)

density_mean = mean(density)
density_sd = sd(density)

q1=qt(0.975,89)*density_sd/sqrt(89)
lower = density_mean-q1
upper = density_mean+q1
cat("lower: ", lower,", upper: ", upper)
```

```
## lower:  0.0006272348 , upper:  0.0007010319
```

```
citation()
```

```
##
## To cite R in publications use:
##
##   R Core Team (2022). R: A language and environment for statistical
##   computing. R Foundation for Statistical Computing, Vienna, Austria.
##   URL https://www.R-project.org/.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {R: A Language and Environment for Statistical Computing},
##     author = {{R Core Team}},
##     organization = {R Foundation for Statistical Computing},
##     address = {Vienna, Austria},
##     year = {2022},
##     url = {https://www.R-project.org/},
##   }
##
## We have invested a lot of time and effort in creating R, please cite it
## when using it for data analysis. See also 'citation("pkgname")' for
## citing R packages.
```

```
citation("readxl")
```

```
##
## To cite package 'readxl' in publications use:
##
##   Wickham H, Bryan J (2022). _readxl: Read Excel Files_. R package
##   version 1.4.1, <https://CRAN.R-project.org/package=readxl>.
##
```

```
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {readxl: Read Excel Files},
##     author = {Hadley Wickham and Jennifer Bryan},
##     year = {2022},
##     note = {R package version 1.4.1},
##     url = {https://CRAN.R-project.org/package=readxl},
##   }
```