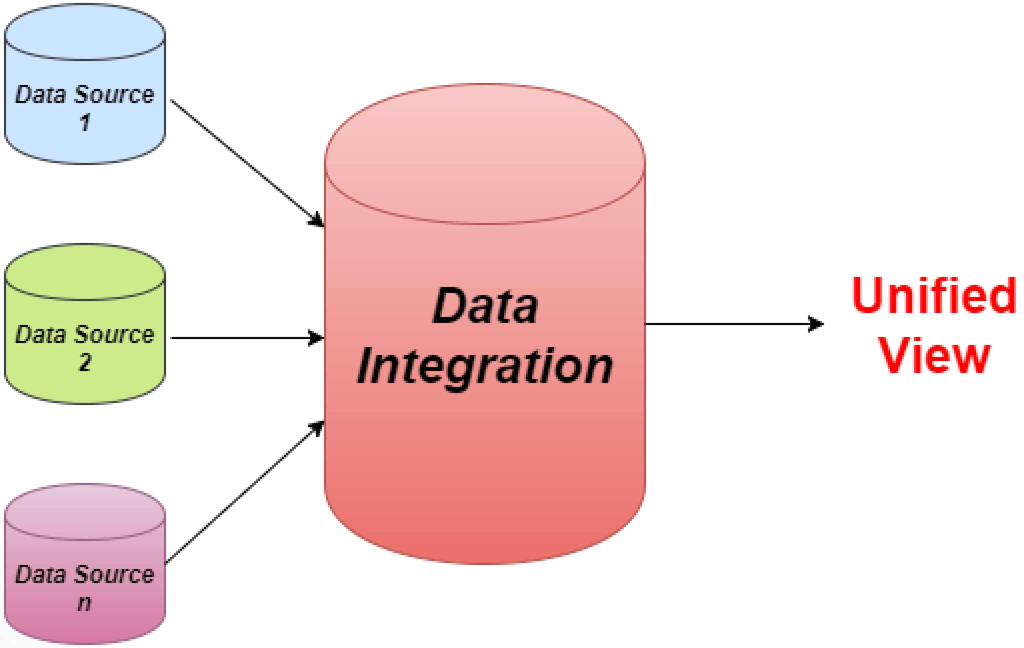
Data Preprocessing II

2. Data Integration:

Data integration is the process of combining data from different sources into a single, unified format.

- Data integration is becoming more common, as numerous apps and companies race to meet consumer demand to have all of their data collected in one place and in a useful format.



2.1 Applications of Data Integration:

- Marketing.
- Healthcare.
- Telecommunications.
- Insurance.
- Government.
- Science.
- Other applications.

Example (Combine and Merge Data sets in R):

Let's create two datasets about some statistical books. The first dataset contains the surname, nationality, and retired. The second dataset contains authors' names, the book title, and the other authors.

1. Enter the first dataset and label the file as authors.

Filename Attributename Observations

```
> authors = data.frame(surname = c("Tukey", "Venables", "Tierney", "Ripley", "McNeil"),
+                      nationality = c("US", "Australia", "US", "UK", "Australia"),
+                      retired = c("yes", rep("no", 4)))
> authors
   surname nationality retired
1    Tukey          US     yes
2 Venables   Australia      no
3  Tierney          US      no
4   Ripley          UK      no
5   McNeil   Australia      no
```

Example (Combine and Merge Data sets in R):

2. Enter the second dataset and label the file as books.

```
> books <- data.frame(name = c("Tukey", "Venables", "Tierney", "Ripley", "Ripley", "McNeil"),
+                     title = c("Exploratory Data Analysis",
+                               "Modern Applied Statistics ...",
+                               "LISP-STAT",
+                               "Spatial Statistics", "Stochastic Simulation",
+                               "Interactive Data Analysis"),
+                     other.author = c(NA, "Ripley", NA, NA, NA, NA))
> books
      name                         title other.author
1    Tukey     Exploratory Data Analysis         <NA>
2 Venables Modern Applied Statistics ...       Ripley
3  Tierney                     LISP-STAT         <NA>
4   Ripley            Spatial Statistics         <NA>
5   Ripley         Stochastic Simulation         <NA>
6   McNeil     Interactive Data Analysis         <NA>
```

Note: the symbols (= and <-) are the same.

Example (Combine and Merge Data sets in R):

3. Find the files dimension.

```
> dim(authors)
[1] 5 3
> dim(books)
[1] 6 3
```

4. Combine both files together by the author’s name. Show the first file attributes first.

```
> authors_and_books1 = merge(authors, books, by.x="surname", by.y="name")
> authors_and_books1
   surname nationality retired                          title other.author
1   McNeil   Australia      no      Interactive Data Analysis         <NA>
2   Ripley          UK      no             Spatial Statistics         <NA>
3   Ripley          UK      no          Stochastic Simulation         <NA>
4  Tierney          US      no                      LISP-STAT         <NA>
5    Tukey          US     yes      Exploratory Data Analysis         <NA>
6 Venables   Australia      no Modern Applied Statistics ...       Ripley
```

Example (Combine and Merge Data sets in R):

5. Combine both files together by the author's name.

Show the second file attributes first.

```
> authors_and_books2 = merge(books, authors, by.x="name", by.y="surname")
> authors_and_books2
      name                          title other.author nationality retired
1   McNeil      Interactive Data Analysis         <NA>   Australia      no
2   Ripley             Spatial Statistics         <NA>          UK      no
3   Ripley          Stochastic Simulation         <NA>          UK      no
4  Tierney                      LISP-STAT         <NA>          US      no
5    Tukey      Exploratory Data Analysis         <NA>          US     yes
6 Venables Modern Applied Statistics ...        Ripley   Australia      no
```

Example 2:

The dataset *airquality* have been divided into 5 files according to the months.

1. Import the datasets to R-Studio, then combine them together in one dataset. Check the dimensions of the original dataset and the new one.

```
> airquality_New = rbind(airquality_May,airquality_Jun,
+                         airquality_Jul,airquality_Aug,airquality_Sep)
> head(airquality_New)
  Ozone Solar.R Wind Temp Month Day
1    41     190  7.4   67     5   1
2    36     118  8.0   72     5   2
3    12     149 12.6   74     5   3
4    18     313 11.5   62     5   4
5    NA      NA 14.3   56     5   5
6    28      NA 14.9   66     5   6
```

```
> dim(airquality)
[1] 153   6
> dim(airquality_New)
[1] 153   6
```

Note:

- **cbind():** is horizontal combination of data (combining vectors or lists with equal number of rows).
- **rbind():** combining lists with equal number of columns. All columns must be of the same data type.
- **merge():** merge two data frames by common columns or row names, or do other versions of database join operations.

3. Data Organization:

Data organization refers to the method of classifying and organizing data sets to make them more useful.

dplyr is a powerful R-package to manipulate data with rows and columns.

https://dplyr.tidyverse.org/reference/index.html

```
install.packages("dplyr")
library(dplyr)
```

Example using dplyr:

Using *airquality* dataset,

1. Create a new dataset by selecting the attributes Ozone, Temp, and Month. Then use head function to print the first rows of the new dataset.

```
> airquality1 = select(airquality, Ozone, Temp, Month)
> head(airquality1)
  Ozone Temp Month
1    41   67     5
2    36   72     5
3    12   74     5
4    18   62     5
5    NA   56     5
6    28   66     5
```

Note: we may use another format to select the attributes,

```
> airquality11 = airquality %>% select(Ozone, Temp, Month)
> head(airquality11)
  Ozone Temp Month
1    41   67     5
2    36   72     5
3    12   74     5
4    18   62     5
5    NA   56     5
6    28   66     5
```

Example using dplyr:

Note: Select function can be used in different format. In the following command, we selected all attributes from Ozone to Month, but since we don't need to select the attributes Solar.R and Wind, so we can subtract them from the selection subset.

```
> airquality111 = select(airquality, Ozone:Month, -Solar.R, -Wind)
> head(airquality111)
  Ozone Temp Month
1    41   67     5
2    36   72     5
3    12   74     5
4    18   62     5
5    NA   56     5
6    28   66     5
```

Example using dplyr:

2. Rename the attribute **Temp** as **Temp.F**, then create a new attribute which the temperature in Celsius "**Temp.C**" by using the formula $^{o}C = (\ ^{o}F - 32) \times 5/9$.

```
> airquality2 = rename(airquality1, Temp.F = Temp)
> head(airquality2)
  Ozone Temp.F Month
1    41     67     5
2    36     72     5
3    12     74     5
4    18     62     5
5    NA     56     5
6    28     66     5
```

```
> airquality3 = mutate(airquality2, Temp.C = (Temp.F-32)*5/9)
> head(airquality3)
  Ozone Temp.F Month   Temp.C
1    41     67     5 19.44444
2    36     72     5 22.22222
3    12     74     5 23.33333
4    18     62     5 16.66667
5    NA     56     5 13.33333
6    28     66     5 18.88889
```

Example using dplyr:

Note: we can round the Temp.C attribute by using round function.

```
> airquality3 = mutate(airquality2, Temp.C = round((Temp.F-32)*5/9))
> head(airquality3)
  Ozone Temp.F Month Temp.C
1    41     67     5     19
2    36     72     5     22
3    12     74     5     23
4    18     62     5     17
5    NA     56     5     13
6    28     66     5     19
```

Example using dplyr:

3. Sort the new dataset by the temperature (min→max).

```
> airquality4 = arrange(airquality3, Temp.F)
> head(airquality4)
  Ozone Temp.F Month Temp.C
1    NA     56     5     13
2     6     57     5     14
3    NA     57     5     14
4    NA     57     5     14
5    18     58     5     14
6    NA     58     5     14
```

4. Sort the new dataset by the temperature in descending order (max→min).

```
> airquality41 = arrange(airquality3, desc(Temp.F))
> head(airquality41)
  Ozone Temp.F Month Temp.C
1    76     97     8     36
2    84     96     8     36
3   118     94     8     34
4    85     94     8     34
5    NA     93     6     34
6    73     93     9     34
```

Example using dplyr:

5. Select the days with temperature below $70^{o}F$.

```
> airquality5 = filter(airquality3, Temp.F < 70)
> head(airquality5)
  Ozone Temp.F Month Temp.C
1    41     67     5     19
2    18     62     5     17
3    NA     56     5     13
4    28     66     5     19
5    23     65     5     18
6    19     59     5     15
```

6. Select the days with ozone above 100.

```
> airquality51 = filter(airquality3, Ozone > 100)
> airquality51
  Ozone Temp.F Month Temp.C
1   115     79     5     26
2   135     84     7     29
3   108     85     7     29
4   122     89     8     32
5   110     90     8     32
6   168     81     8     27
7   118     94     8     34
```

Example using dplyr:

7. Select a random sample of 5 values from the dataset.

```
> airquality6 = sample_n(airquality3, 5)
> airquality6
  Ozone Temp.F Month Temp.C
1    20     65     6     18
2    16     82     8     28
3   122     89     8     32
4    35     82     7     28
5    NA     76     6     24
```

8. Select a random sample of 5% from the dataset.

```
> airquality7 = sample_frac(airquality3, 0.05)
> airquality7
  Ozone Temp.F Month Temp.C
1    32     61     5     16
2    48     81     7     27
3    NA     75     8     24
4    14     75     9     24
5    45     81     5     27
6    35     85     8     29
7     4     61     5     16
8    11     62     5     17
```

Example using dplyr:

9. Group the dataset by month.

```
> airquality8 = group_by(airquality3, Month)
```

10. How many data values do we have by month?

```
> summarize(airquality8, n = n())
# A tibble: 5 x 2
  Month     n
  <int> <int>
1     5    31
2     6    30
3     7    31
4     8    31
5     9    30
```

Note: count function does both grouping and counting.

```
> count(airquality3, Month)
# A tibble: 5 x 2
  Month     n
  <int> <int>
1     5    31
2     6    30
3     7    31
4     8    31
5     9    30
```

Example using dplyr:

11. Find the summary statistics for the dataset.

```
> summary(airquality3)
     Ozone             Temp.F          Month          Temp.C     
 Min.   :  1.00   Min.   :56.00   Min.   :5.000   Min.   :13.00  
 1st Qu.: 18.00   1st Qu.:72.00   1st Qu.:6.000   1st Qu.:22.00  
 Median : 31.50   Median :79.00   Median :7.000   Median :26.00  
 Mean   : 42.13   Mean   :77.88   Mean   :6.993   Mean   :25.46  
 3rd Qu.: 63.25   3rd Qu.:85.00   3rd Qu.:8.000   3rd Qu.:29.00  
 Max.   :168.00   Max.   :97.00   Max.   :9.000   Max.   :36.00  
 NA's   :37                                                      
```

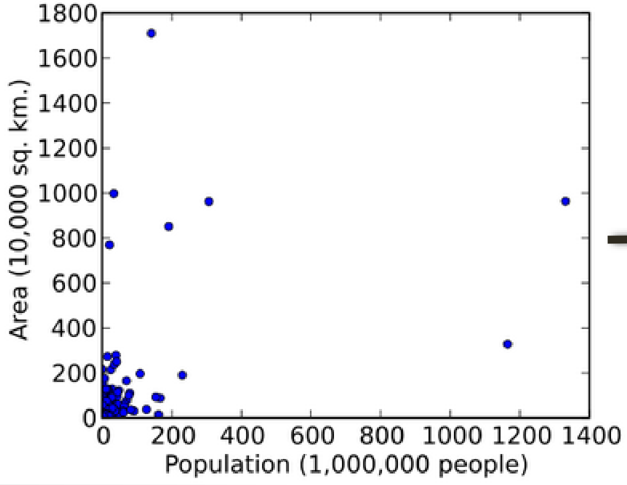
12. Find the temperature mean by month.

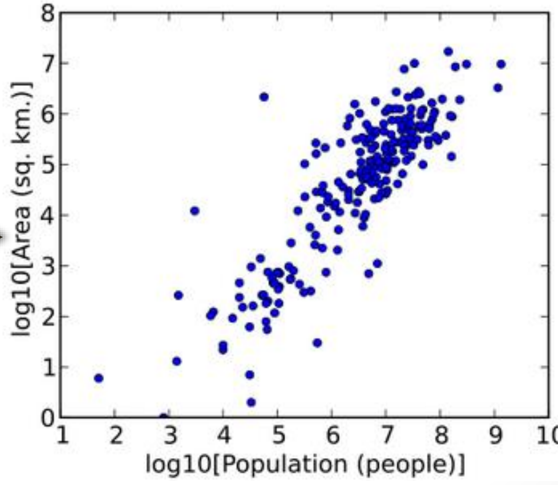
```
> summarize(airquality8, mean_Temp.F = mean(Temp.F, na.rm = TRUE))
# A tibble: 5 x 2
  Month mean_Temp.F
  <int>       <dbl>
1     5        65.5
2     6        79.1
3     7        83.9
4     8        84.0
5     9        76.9
```

4. Data Transformation:

Data transformation is the process of converting the values of observations (attribute) through some transforming operation.

- Data transformation allows users to derive new attribute from existing ones.
- The transformation process can change the scale of the attributes, the grouping of the values, and the type of the attributes.





4.1 Reasons for using Transformations:

1. Convenience:

    more convenient for a specific purpose.

2. Reducing skewness:

    reduce data skewness.

3. Equal spreads:

    reduce the variation in data.

4. Linear relationships:

    to make the relationship more linear.

4.2 Choosing the Right Transformation:

There are many transformations we could use, but it is better to use a transformation that other researchers commonly use in your field.

- It is important that we decide which transformation to use before we analyze the data.
- To make data more convenient, we can use normalization, standardization, or scaling.
- To reduce data skewness, we may use the log, square root, reciprocal transformation.

4.2.1 Normalization, Standardization:

Normalization, rescales an attribute to have values in the range [0,1].

$$x_{new} = \frac{x_{original} - x_{min}}{x_{max} - x_{min}} = \frac{x_{original} - x_{min}}{Range}$$

- useful for sparse attribute features and algorithms using distance to learn such as KNN (K Nearest Neighbor).

Standardization, transforms an attribute to have a mean 0 and standard deviation 1,

$$x_{new} = \frac{x - \mu}{\sigma}$$

- works better with linear regression, logistic regression and linear discriminate analysis.

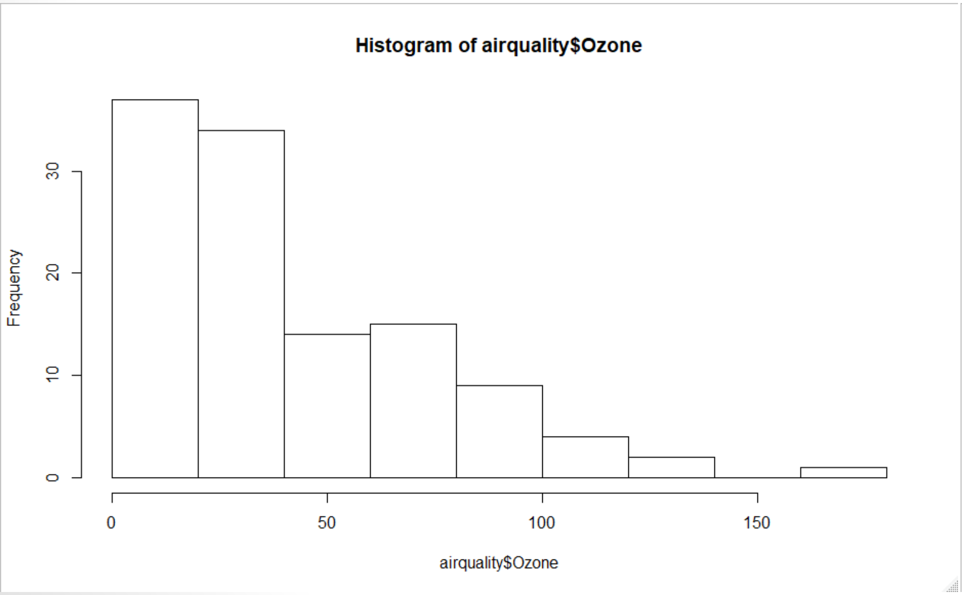
Example:

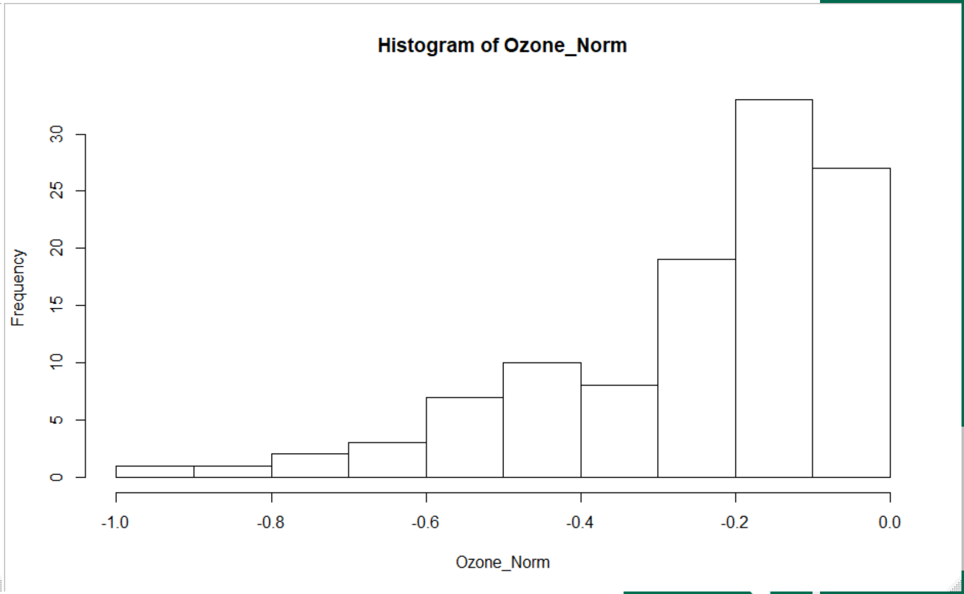
Using *airquality* dataset,

1. Graph the histogram of Ozone, then transform it using normalization and graph it.

```
> hist(airquality$Ozone)
```

```
> Air_1 = na.omit(airquality)
> Ozone_Norm=(Air_1$Ozone-min(Air_1$Ozone))/(min(Air_1$Ozone)-max(Air_1$Ozone))
> hist(Ozone_Norm)
```



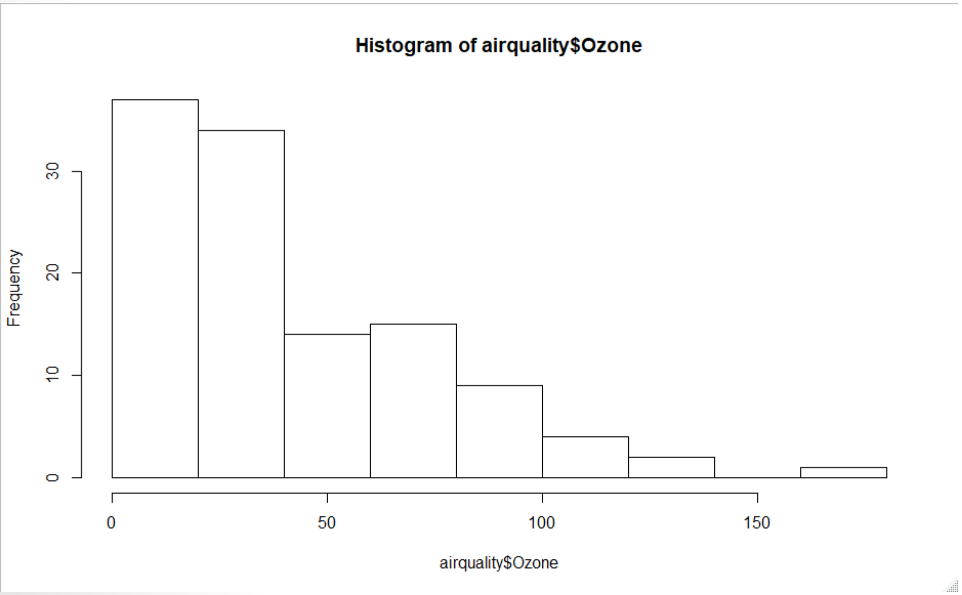


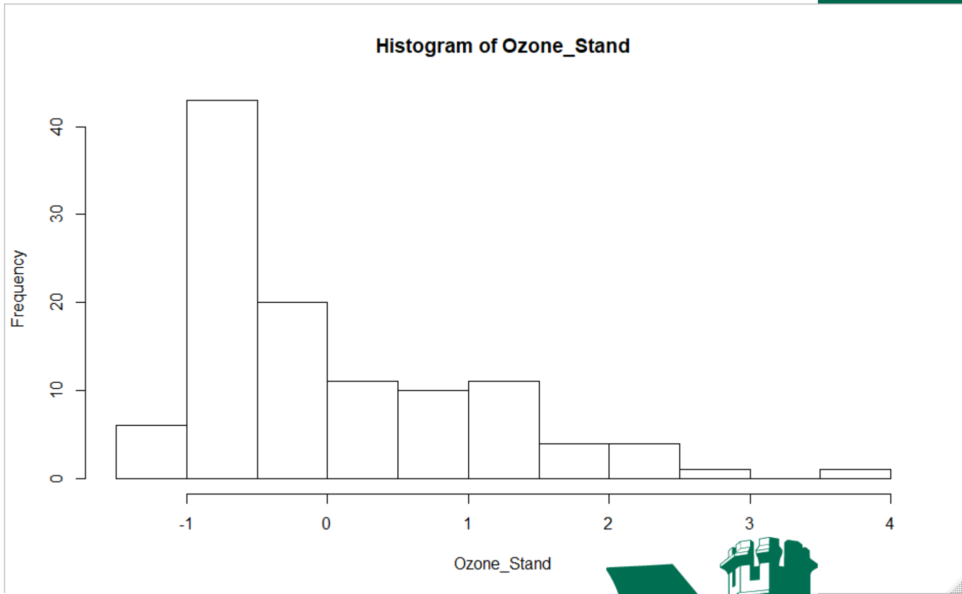
Example:

2. Graph the histogram of Ozone, then transform it using standardization and graph it.

```
> hist(airquality$Ozone)
```

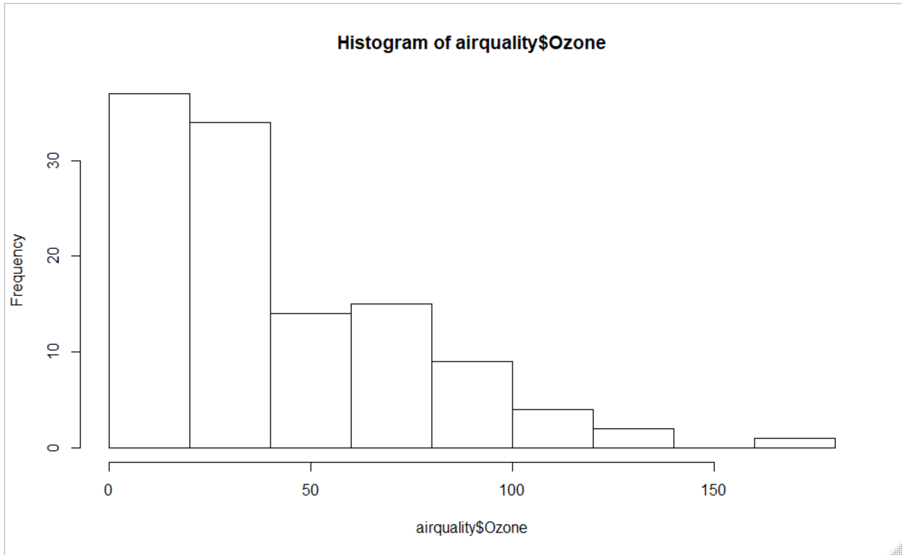
```
> Ozone_Stand=(Air_1$Ozone-mean(Air_1$Ozone))/sd(Air_1$Ozone)
> hist(Ozone_Stand)
```



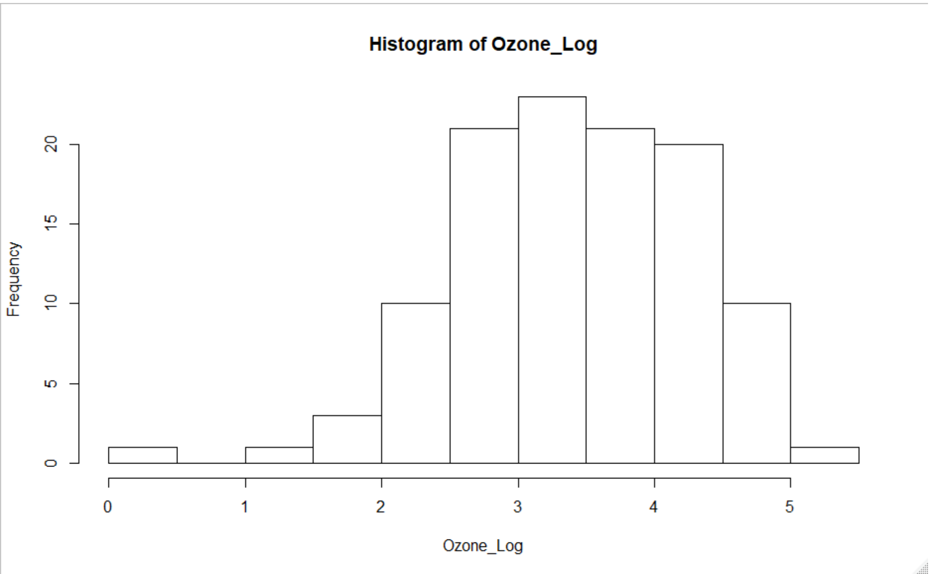


Example:

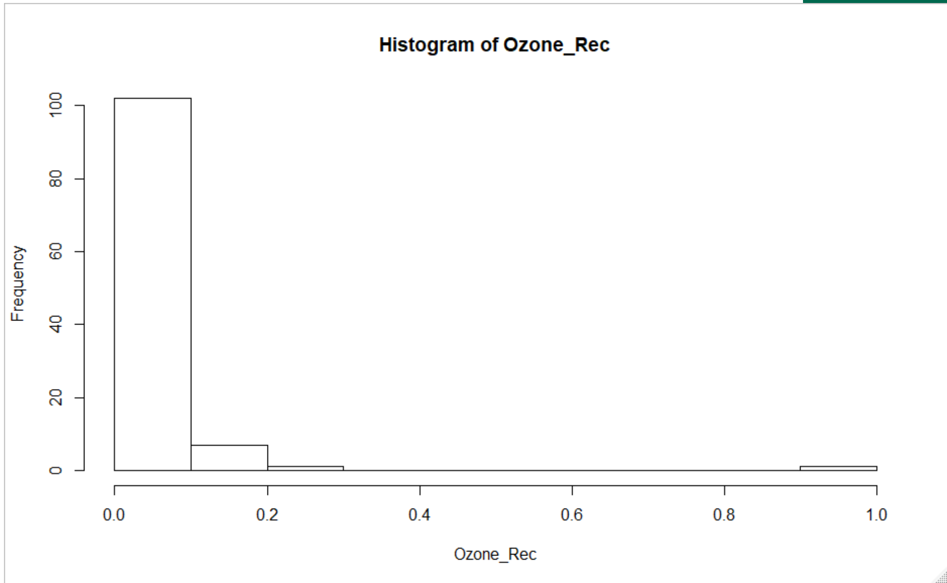
3. Use some other transformation to reduce skewness.

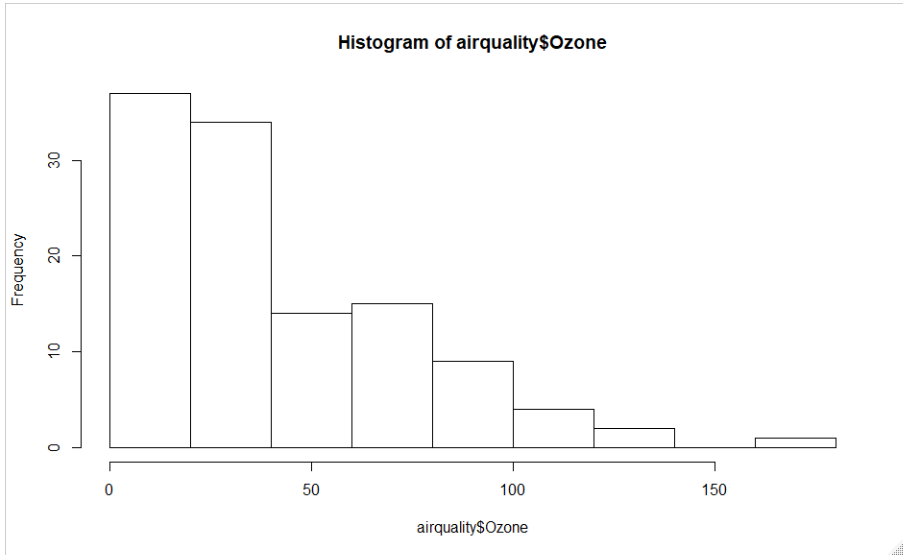



```
> Ozone_Log = log(Air_1$Ozone)
> hist(Ozone_Log)
```



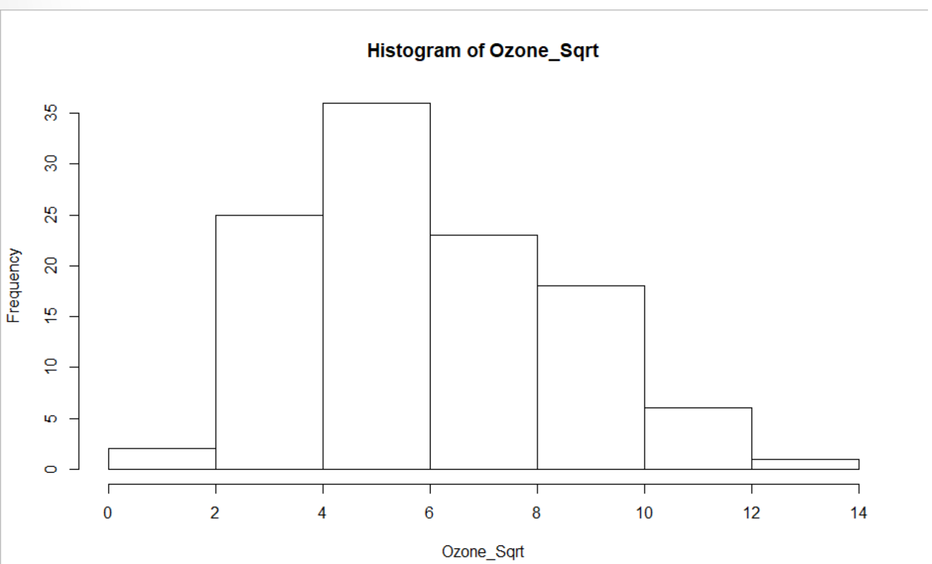

```
> Ozone_Rec = 1 / (Air_1$Ozone)
> hist(Ozone_Rec)
```

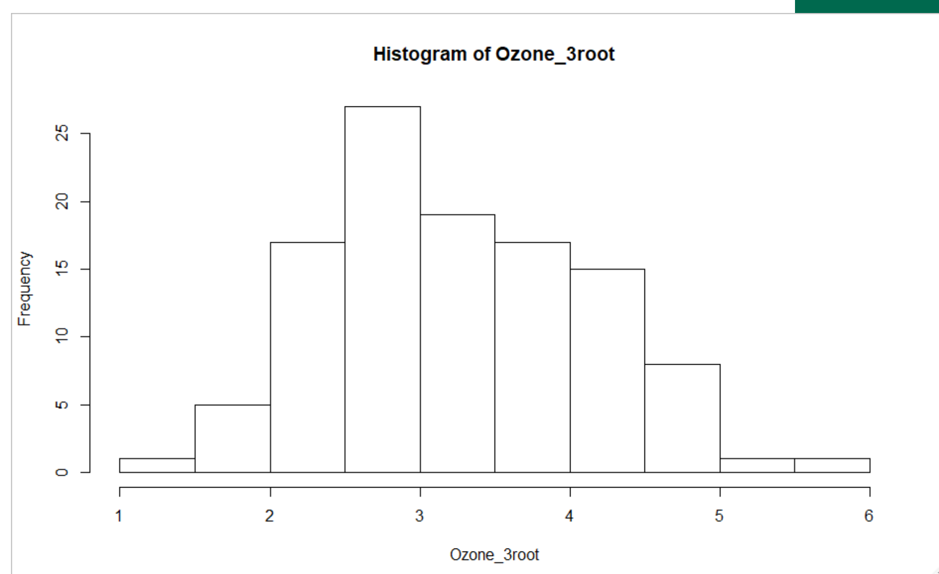





```
> Ozone_Sqrt = sqrt(Air_1$Ozone)
> hist(Ozone_Sqrt)
```

```
> Ozone_3root = (Air_1$Ozone)^(1/3)
> hist(Ozone_3root)
```





Box Cox Transformation:

propose a *family* of transformations that are indexed by a parameter (λ):

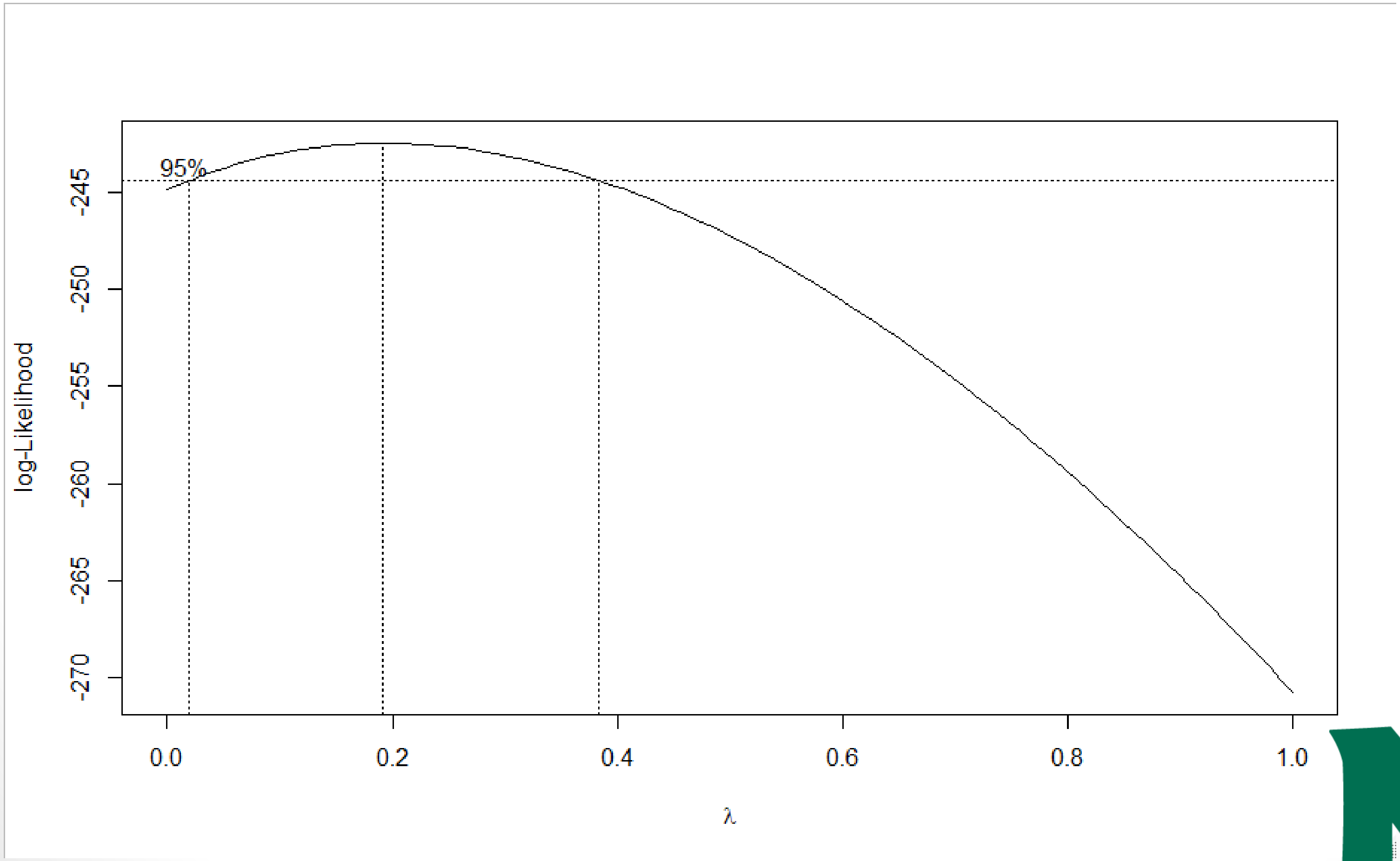
$$x^* = \begin{cases} \dfrac{x^\lambda - 1}{\lambda} & if \quad \lambda \neq 0 \\ ln(x) & if \quad \lambda = 0 \end{cases}$$

- First, we find the value of the parameter (λ), then we use it in the formula.

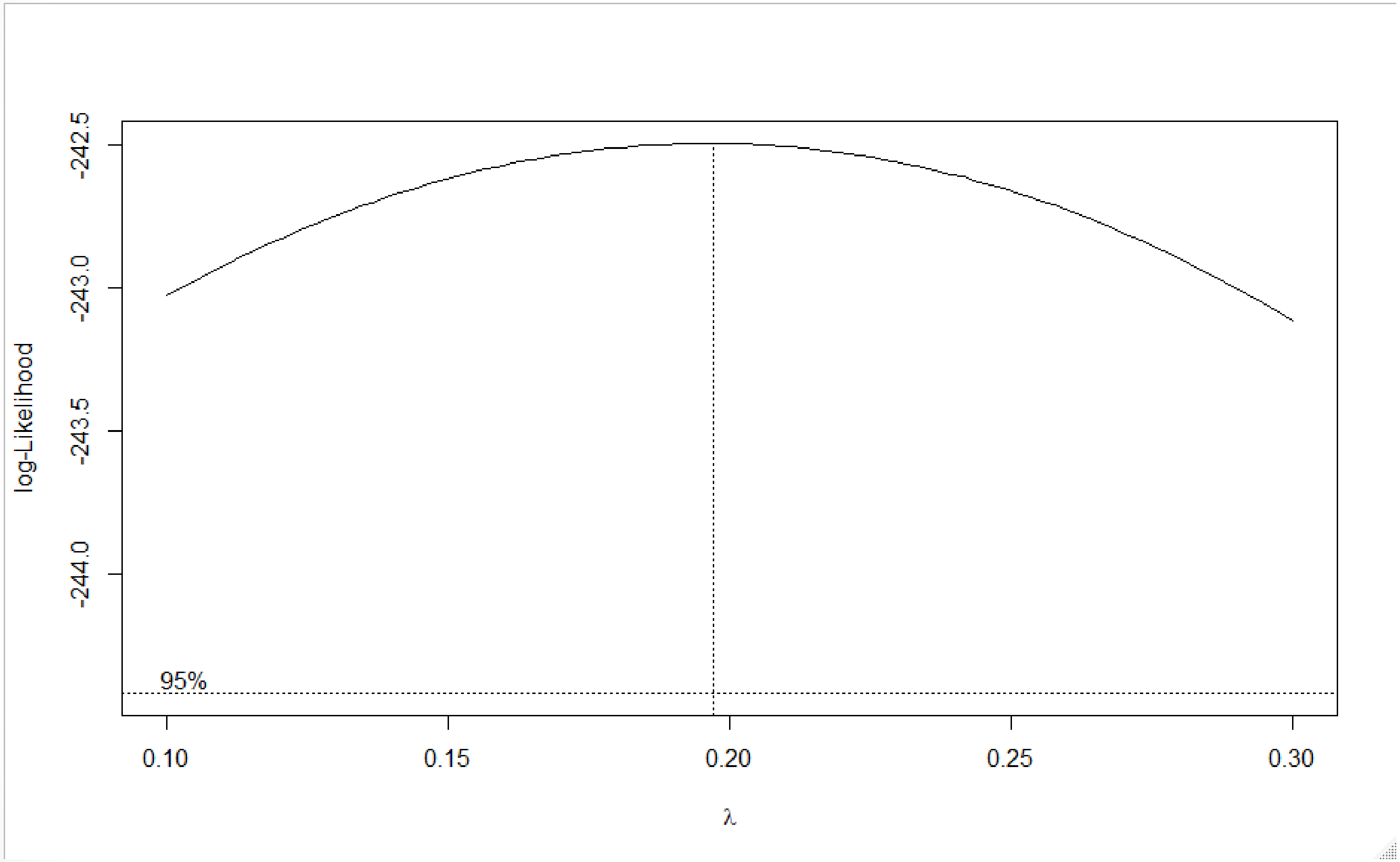
Example:

4. Find the optimal (λ) for Box-Cox transformation.

```
> lambda_opt = boxcox(Air_1$Ozone~1, lambda = seq(0.0, 1, by = 0.1))
```

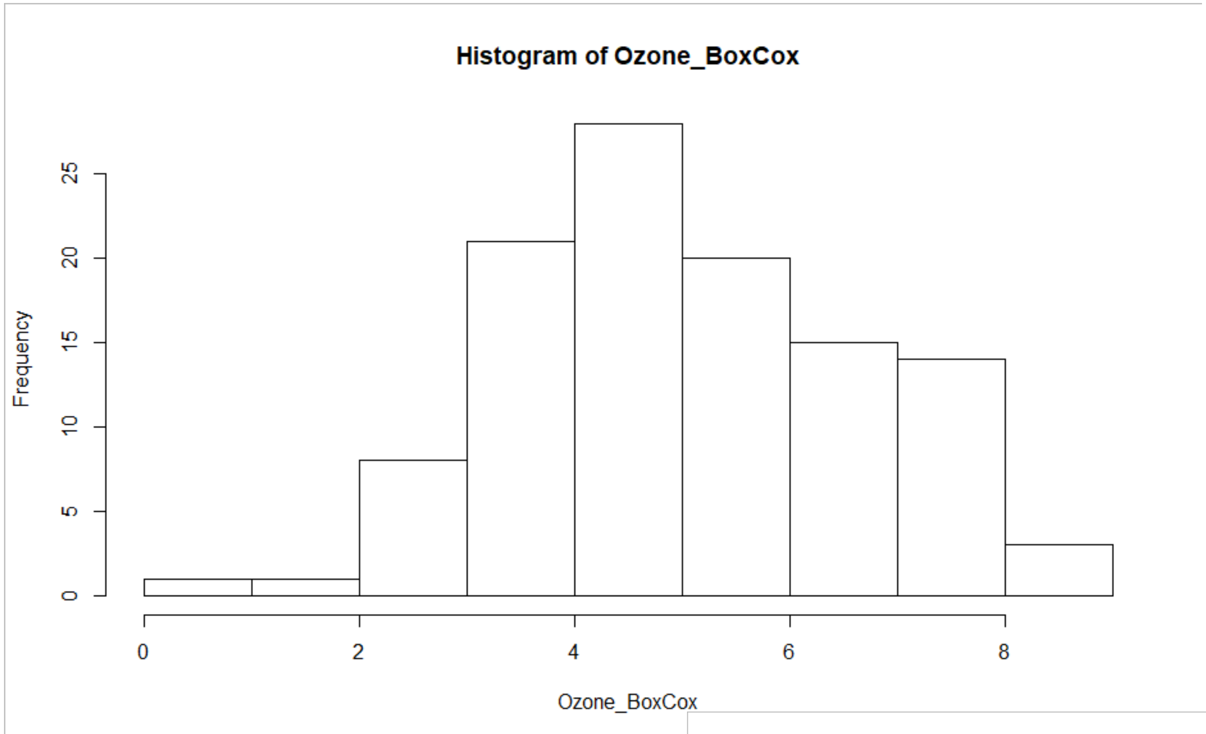


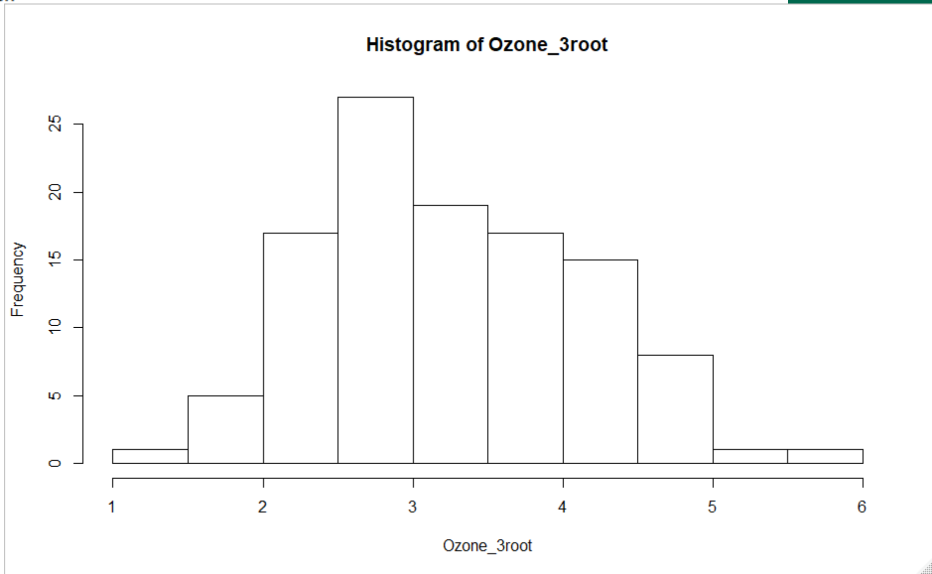
```
> lambda_opt = boxcox(Air_1$Ozone~1, lambda = seq(0.1, 0.3, by = 0.1))
```



$\lambda \approx 0.2$, let's use it in the formula

```
> Ozone_BoxCox = (Air_1$Ozone^0.2 - 1) / 0.2
> hist(Ozone_BoxCox)
```





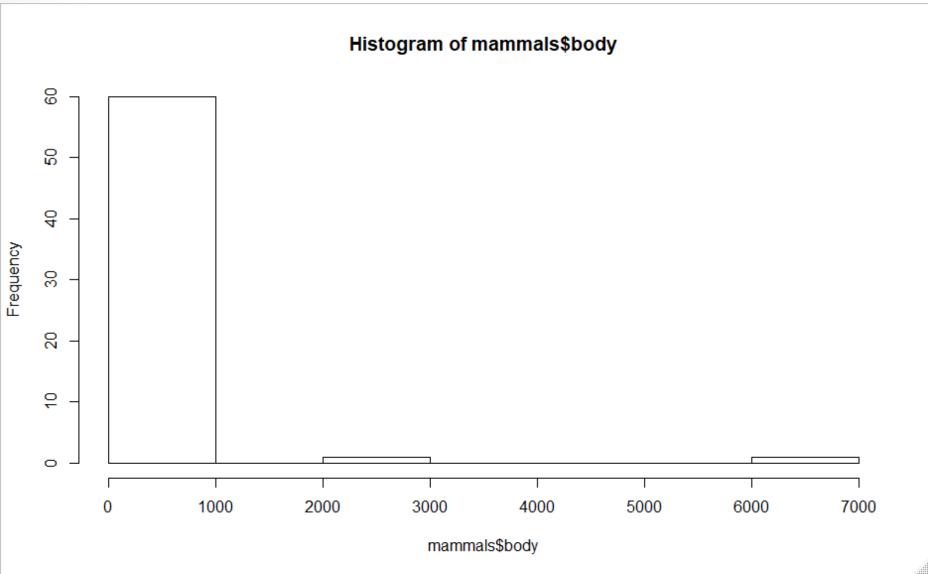
Example:

mammals (data set in r) includes the average brain and body weights for 62 species of land mammals.

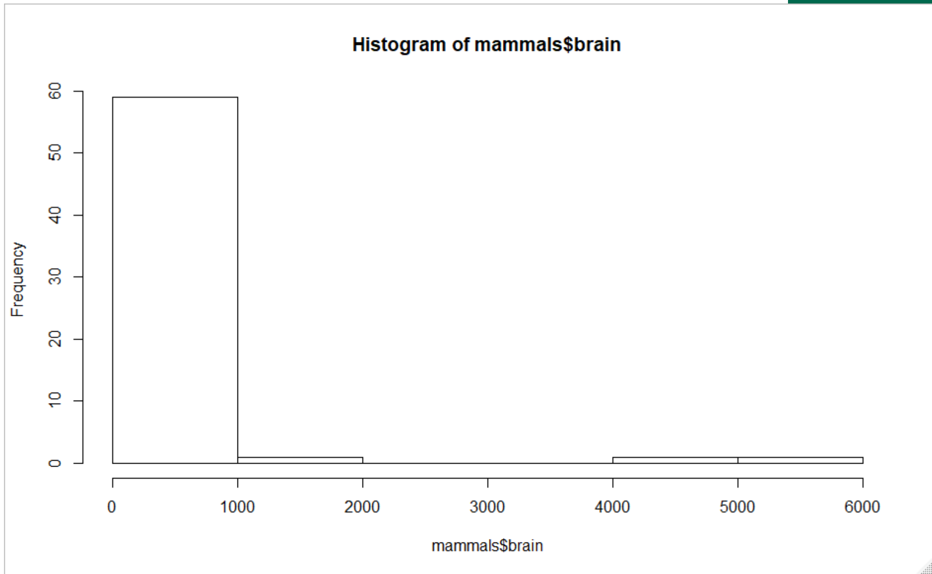
1. Graph a histogram for each variable.

```
> head(mammals)
                   body brain
Arctic fox        3.385  44.5
Owl monkey        0.480  15.5
Mountain beaver   1.350   8.1
Cow             465.000 423.0
Grey wolf        36.330 119.5
Goat             27.660 115.0
```

```
> hist(mammals$body)
```



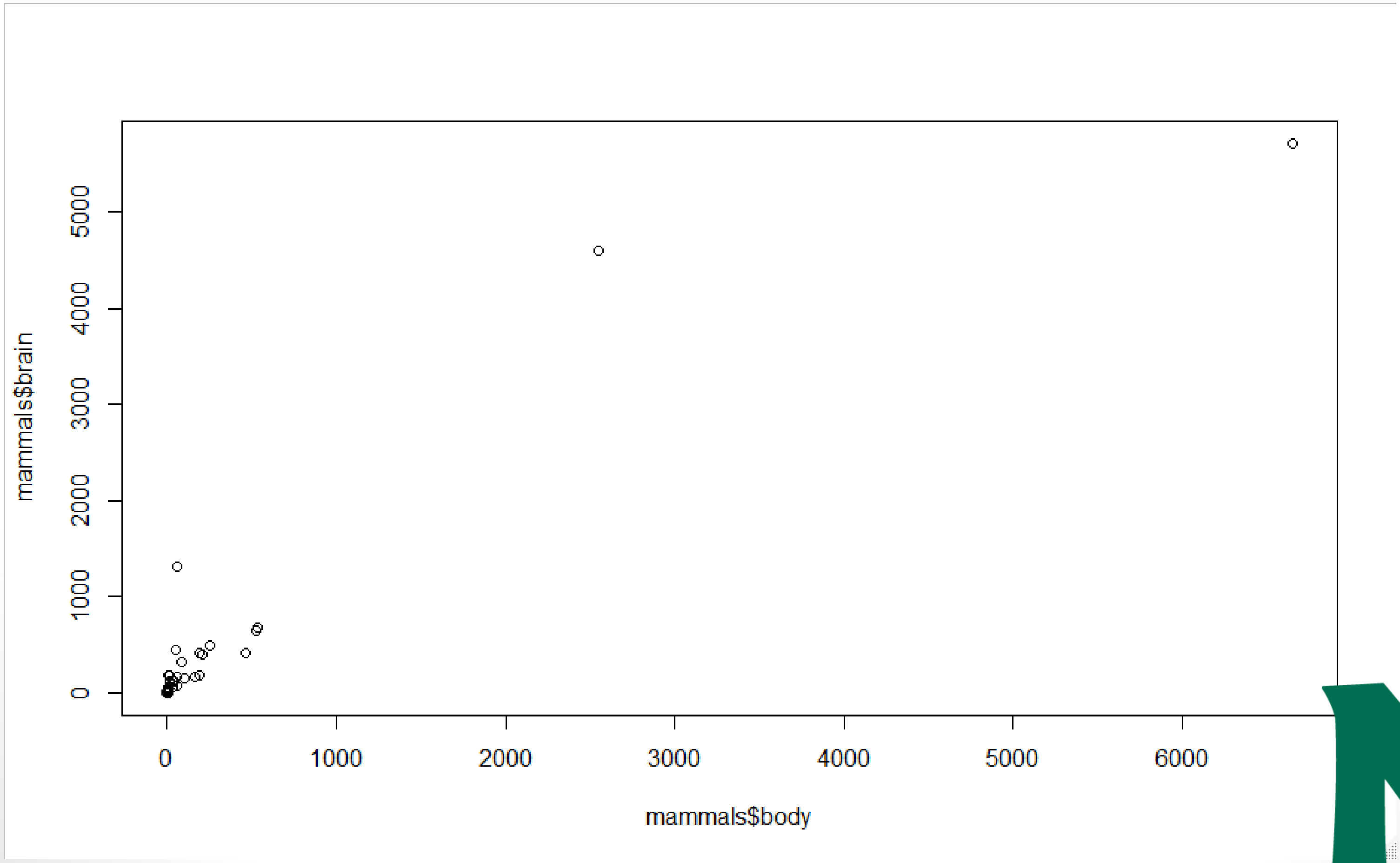

```
> hist(mammals$brain)
```



Example:

2. Graph a scatterplot to describe the association between the average brain and body weights.

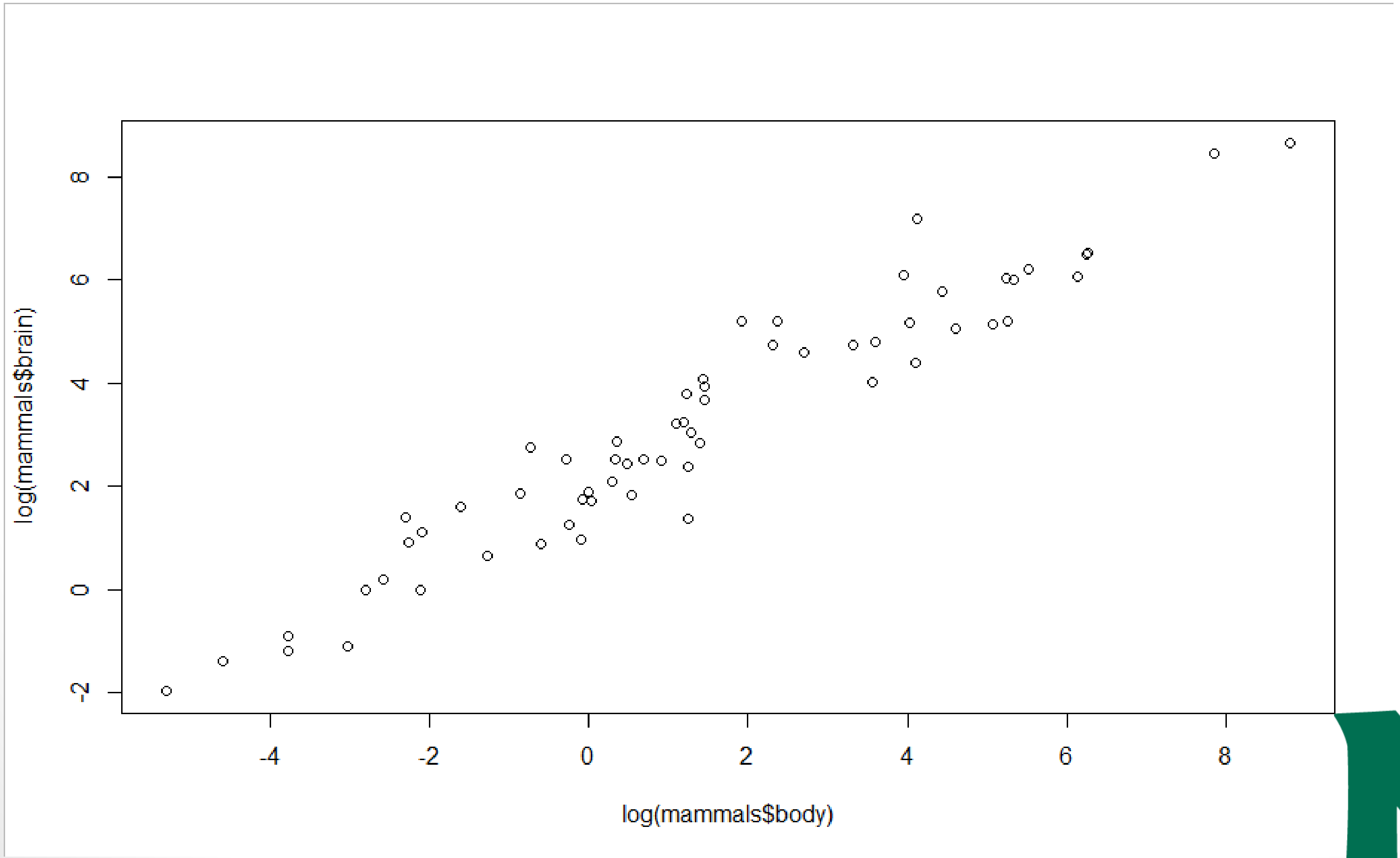
```
> plot(mammals$body, mammals$brain)
```



Example:

2. Transform the two attribute by using log transformation, and graph the scatterplot.

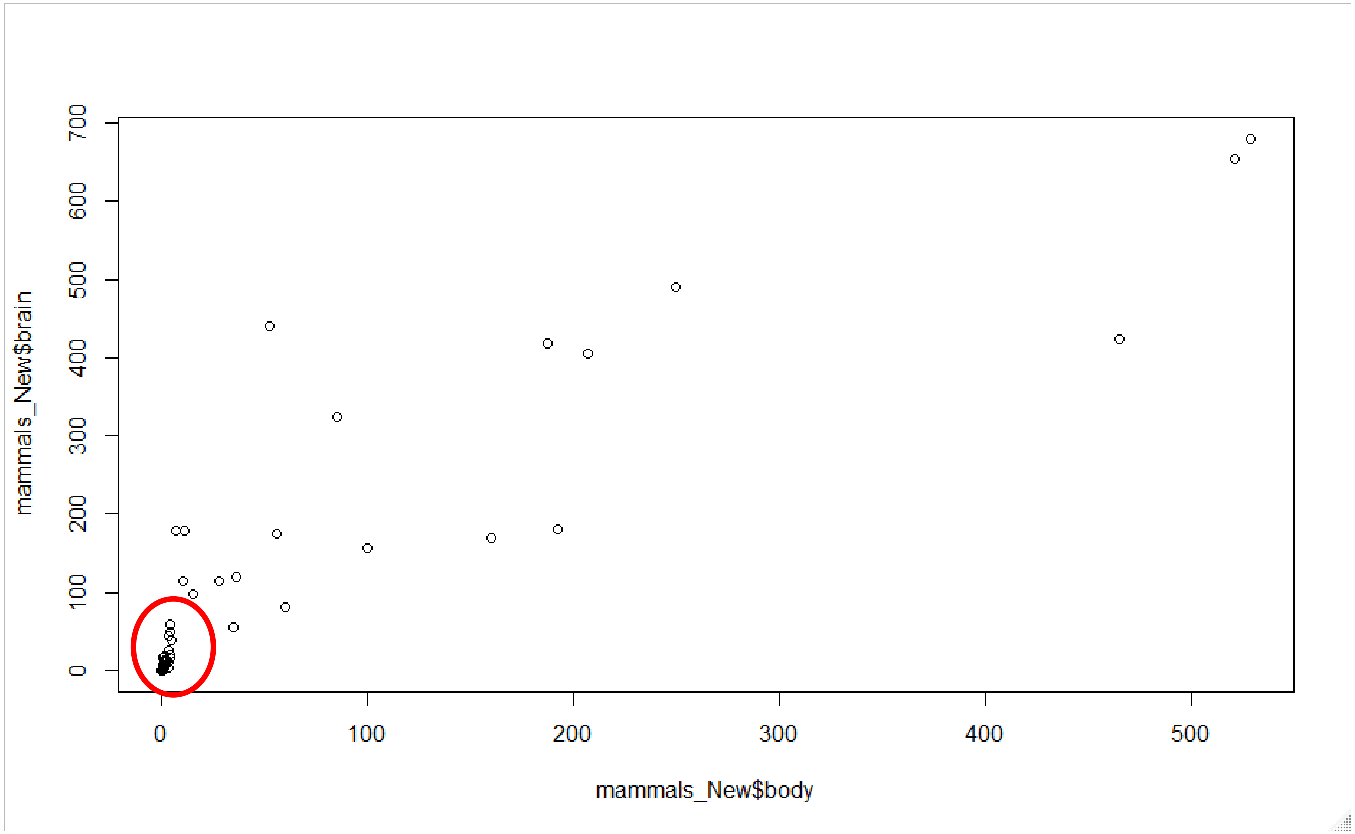
```
> plot(log(mammals$body), log(mammals$brain))
```



Example:

3. Let’s redo part(1) after removing the outliers.

```
> mammals_New = filter(mammals, body < 1000, brain < 1000)
> plot(mammals_New$body, mammals_New$brain)
```



We can observe that this scatterplot wasn’t good as the graph in part (2).