TRANSFORMING OUTLIERS

OVERVIEW
- **outliers** = extreme values that tend to exert undue influence on result of analysis
- Know your options for dealing with outliers (so can get results that mean what you think they mean)

SUMMARY
- Log linear transformation brings it closer to the normal distribution - which is the assumption behind so many statistical procedures

PSEUDOCODE

INSTALL AND LOAD PACKAGES
LOAD AND PREPARE DATA
Check existence of outliers - with histogram:
Best way to check for outliers is with boxplot: (bc it marks outliers)
LOOK AT SOME OPTIONS FOR LEADING WITH OUTLIERS
*PART1: **REMOVE OUTLIERS***
See which ones are outliers:
Filter out continents
Create histogram & boxplot
*PART2: **BRING IN OUTLIERS***
Sort observations in descending order:
IF value is greater than 840 THEN change it to 840:
Graph results:
*PART3: **CREATE NEW GROUPS***
Sort in descending order:
Create new variable - called "landmass" > IF value is < 1000 THEN call it an island:
Boxplot of area of continents:
Boxplot of islands:
*PART4**: TRANSFORMING DATA***
Strong positive skew:
Log is natural logarithm with base e (base 10 is log10) > brings in extreme positive values:

INSTALL AND LOAD PACKAGES

```
pacman::p_load(datasets, pacman, tidyverse)
```

LOAD AND PREPARE DATA

from **datasets package** - **`islands`** dataset - has strong positive skew

```
?islands
```

- area of world's major land masses
- areas on 1000s of square mile - of landmasses that exceed 10,000 sq miles
- named vector - with 48 observations

```
islands
```

- results:

-alphabetical order - with area (from Africa with 11506, down to end of list with Victoria with 82 (thousands of square miles)

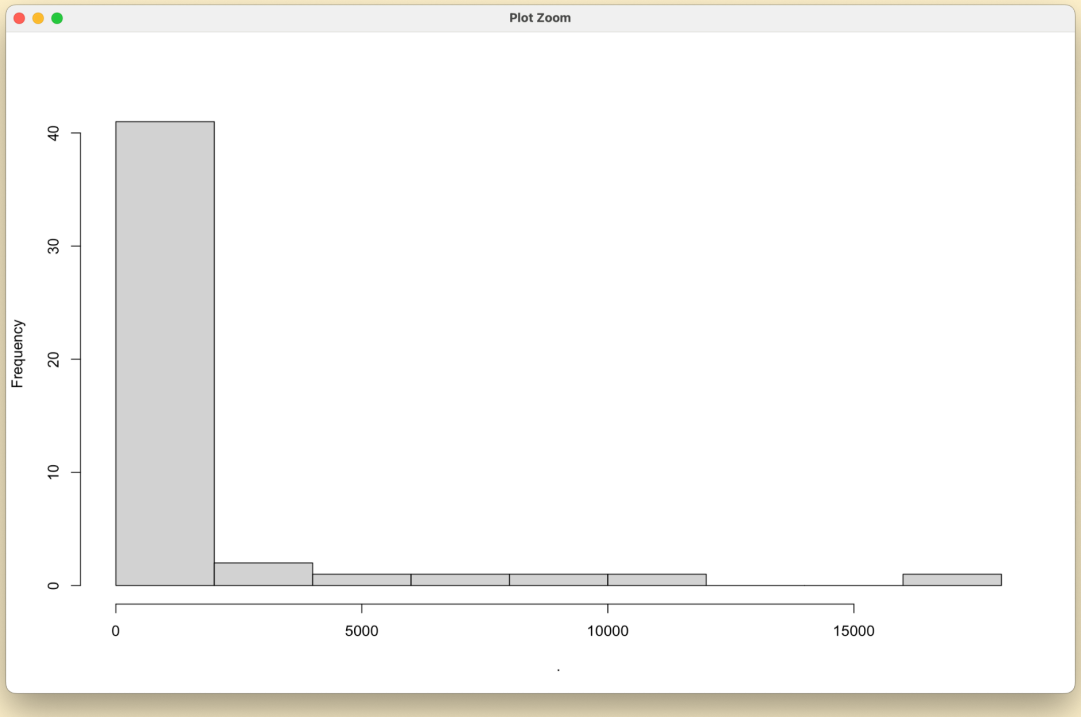
```
> islands
        Africa      Antarctica            Asia       Australia
         11506            5500           16988            2968
   Axel Heiberg          Baffin           Banks          Borneo
            16             184              23             280
       Britain         Celebes           Celon            Cuba
            84              73              25              43
         Devon       Ellesmere         Europe       Greenland
            21              82            3745             840
        Hainan      Hispaniola        Hokkaido          Honshu
            13              30              30              89
       Iceland         Ireland            Java          Kyushu
            40              33              49              14
        Luzon      Madagascar        Melville        Mindanao
            42             227              16              36
     Moluccas      New Britain      New Guinea  New Zealand (N)
            29              15             306              44
New Zealand (S)     Newfoundland   North America    Novaya Zemlya
            58              43            9390              32
Prince of Wales         Sakhalin   South America      Southampton
            13              29            6795              16
    Spitsbergen         Sumatra          Taiwan        Tasmania
            15             183              14              26
Tierra del Fuego           Timor       Vancouver        Victoria
            19              13              12              82
```

Check existence of outliers - with histogram:

```
islands %>% hist(main = NULL)
```

- result:

-most in smallest size bin - and few way out past rest of plot >> NOT normal distribution & big outliers (throws off analysis)

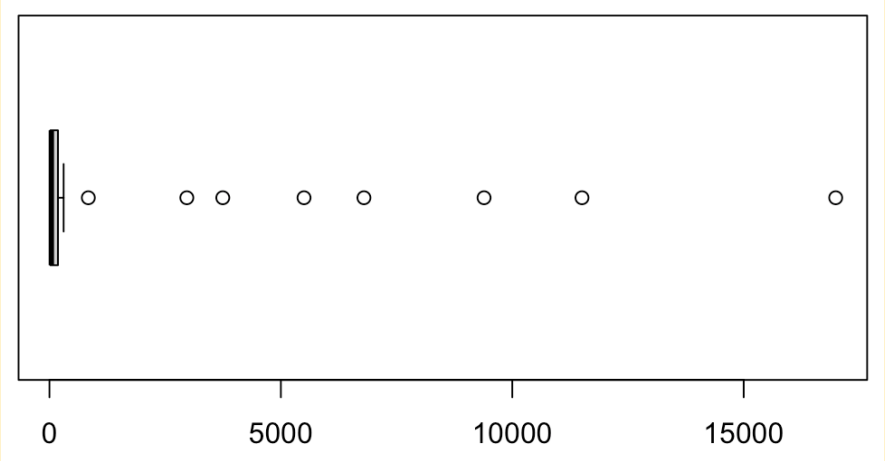


Best way to check for outliers is with boxplot: (bc it marks outliers)

```
islands %>% boxplot(horizontal = T)
```

- result:

-range of 50% scores is super compressed and at low end
-even the highest non-outlying data point is compressed
-8 outliers - one extreme (asia)



LOOK AT SOME OPTIONS FOR LEADING WITH OUTLIERS

there are sophisticated algorithms that deal with outliers:

- when using the non-parametric approach
  - decision trees dn get thrown off by outliers
  - neural networks - are more flexible wrt outliers

IF standard analysis (scatterplots, means) THEN want to deal with outliers:

PART1: ***REMOVE OUTLIERS***

- draconian - cut off outliers (throw then away)
- appropriate - as long as
  - 1)only care about non-outlier scores
  - 2)are specific , clear that you did that (and that focusing only on the major ones)

See which ones are outliers:

- *Sort observation in descending value:*

```
islands2 <- islands %>%
    enframe() %>%              # convert vector to tibble
  arrange(desc(value)) %>%     # sort by descending values
  print()
```

  - result:

-Asia 16K Africa 11K .. (top masses are continents - of course they are going to be huge)

```
> islands2 <- islands %>%
+   enframe() %>%             # Convert vector to tibble
+   arrange(desc(value)) %>%  # Sort by descending values
+   print()
# A tibble: 48 × 2
   name          value
   <chr>         <dbl>
 1 Asia          16988
 2 Africa        11506
 3 North America  9390
 4 South America  6795
 5 Antarctica     5500
 6 Europe         3745
 7 Australia      2968
 8 Greenland       840
 9 New Guinea      306
10 Borneo          280
# ℹ 38 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Data	
islands2	48 obs. of 2 variables

- → can say - we don't want to focus on continents > want to focus on islands (a way of defining your sample, that helps you deal with some of theses outliers)

Filter out continents

```
islands2 %>%
    filter(value < 1000) %>%
    print()
```

- result:

-filters out everything to Australia, and keep greenland and below

```
> islands2 %<>%
+   filter(value < 1000) %>%
+   print()
# A tibble: 41 × 2
   name       value
   <chr>      <dbl>
 1 Greenland    840
 2 New Guinea   306
 3 Borneo       280
 4 Madagascar   227
 5 Baffin       184
 6 Sumatra      183
 7 Honshu        89
 8 Britain       84
 9 Ellesmere     82
10 Victoria      82
# ℹ 31 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Data	
islands2	41 obs. of 2 variables

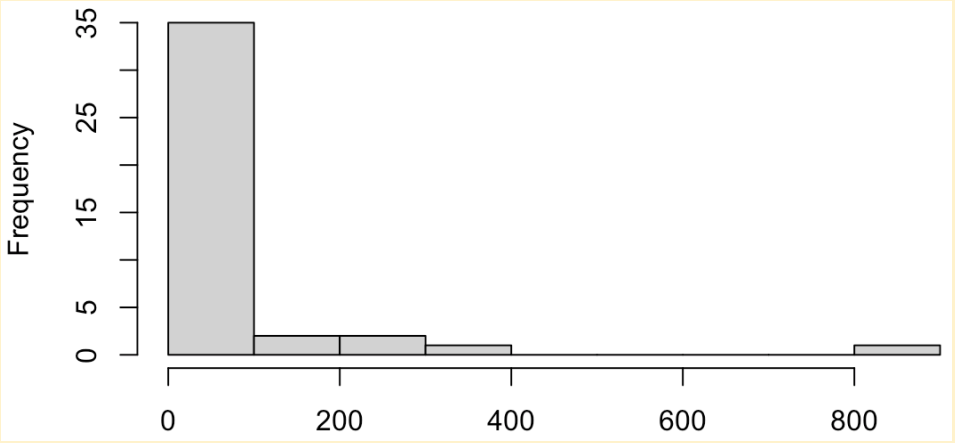
Create histogram & boxplot

- **`pull()`** creates a VECTOR (like `islands$value`)
- **`select()`** create a DATAFRAME
- (dift functions need dift data formats)

```
islands2 %>% pull(value) %>% histogram(main = NULL)
```

- result:

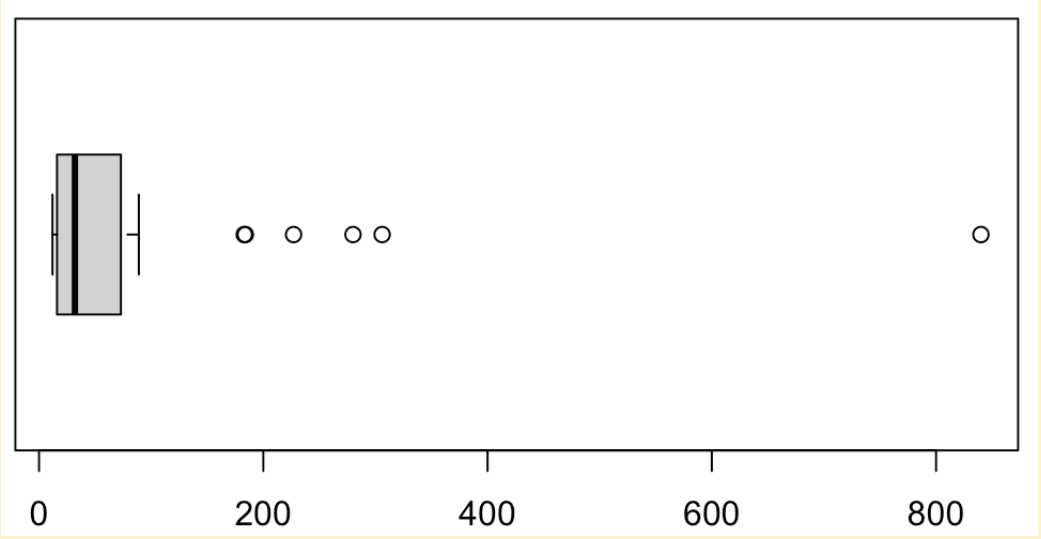
-still have outliers (greenland)




```
islands2 %>% select(value) %>% boxplot(horizontal = T)
```

- result:

-not quite as pathological as before (bd redefining group that interested in)



PART2: ***BRING IN OUTLIERS***

- **winsorizing** - option - that is not used very often
  - ex: times on races / time to graduation /financial data
- take extreme values -> change them to highest non-outlier value
  - ex: time to graduation - we are going up to 8 years, and anything after 8 we are going to code as 8

Sort observations in descending order:

```
islands3 <- islands %>%
    enframe() %>%                # convert vector to tibble
  arrange(desc(value)) %>%       # sort by descending values
   print()
```

- results:

The largest non-continent is Greenland, with an area of about 840,000 square miles (coded as 840 in the dataset)

```
> islands3 <- islands %>%
+   enframe() %>%             # Convert vector to tibble
+   arrange(desc(value)) %>%  # Sort by descending values
+   print()
# A tibble: 48 × 2
   name          value
   <chr>         <dbl>
 1 Asia          16988
 2 Africa        11506
 3 North America  9390
 4 South America  6795
 5 Antarctica     5500
 6 Europe         3745
 7 Australia      2968
 8 Greenland       840
 9 New Guinea      306
10 Borneo          280
# ℹ 38 more rows
# ℹ Use `print(n = ...)` to see more rows
```

islands3	48 obs. of 2 variables

IF value is greater than 840 THEN change it to 840:

```
islands3 <- islands3 %>%
    mutate(
        value = ifelse(
              value = 840,      # test if value is greater than 840
              840,              # if true, replace value with 840
              value             # if false, keep existing value
        )
    ) %>%
    print()
```

- results:

Now many 840s (for all the continents)

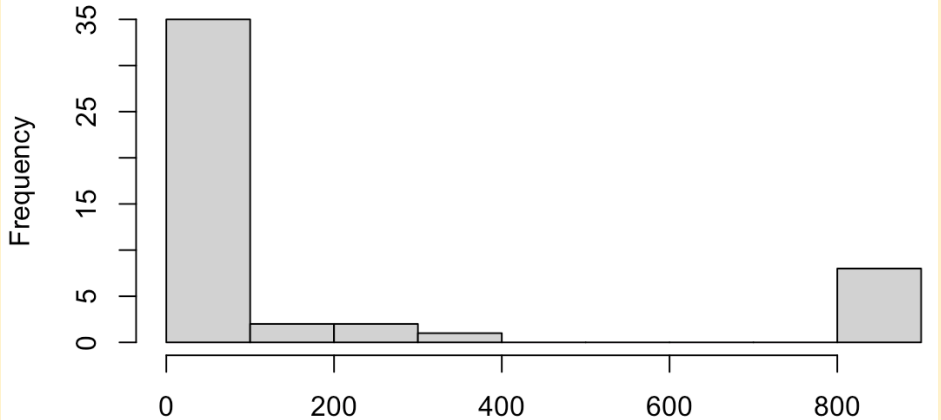
```
> islands3 <- islands3 %>%
+   mutate(
+     value = ifelse(
+       value > 840,  # Test if value is greater than 840
+       840,          # If true, replace value with 840
+       value         # If false, keep existing value
+     )
+   ) %>%
+   print()
# A tibble: 48 × 2
   name          value
   <chr>         <dbl>
 1 Asia            840
 2 Africa          840
 3 North America   840
 4 South America   840
 5 Antarctica      840
 6 Europe          840
 7 Australia       840
 8 Greenland       840
 9 New Guinea      306
10 Borneo          280
# ℹ 38 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Graph results:

```
islands3 %>% pull(value) %>% hist(main = NULL)
```

- results

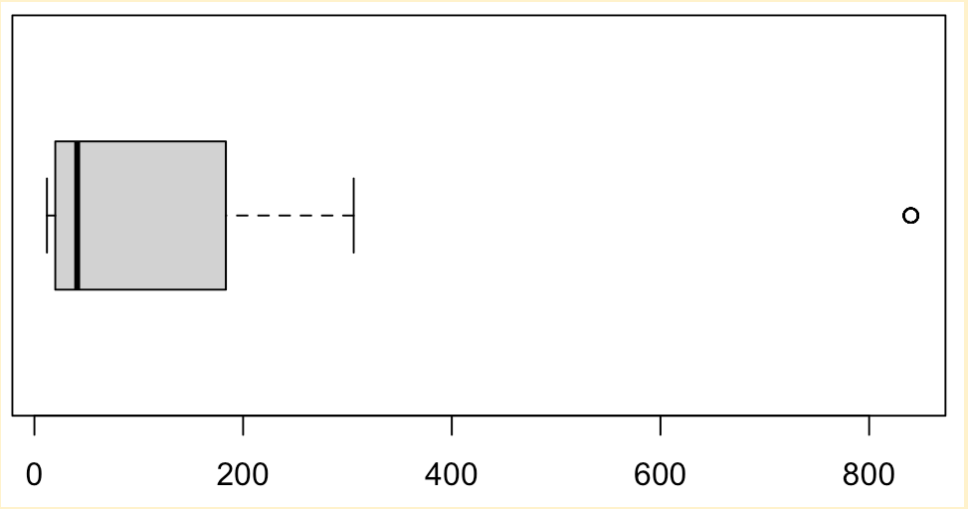
Big bump at high end - bc many observation s at 840




```
islands3 %>% select(value) %>% boxplot(horizontal = T)
```

- results

Several observation at high end stacked into of each other



PART3: ***CREATE NEW GROUPS***

Split into 2 groups: continents, islands > will treat then separately

Sort in descending order:

```
islands4 <- islands %>%
    enframe() %>%                # convert vector to tibble
   arrange(desc(value)) %>%      # sort in descending order
   print()
```

- results:

```
> islands4 <- islands %>%
+   enframe() %>%             # Convert vector to tibble
+   arrange(desc(value)) %>%  # Sort by descending values
+   print()
# A tibble: 48 × 2
   name          value
   <chr>         <dbl>
 1 Asia          16988
 2 Africa        11506
 3 North America  9390
 4 South America  6795
 5 Antarctica     5500
 6 Europe         3745
 7 Australia      2968
 8 Greenland       840
 9 New Guinea      306
10 Borneo          280
# ℹ 38 more rows
# ℹ Use `print(n = ...)` to see more rows
```

islands4	48 obs. of 2 variables

Create new variable - called "landmass" > IF value is < 1000 THEN call it an island:

```
island4 %>%
    mutate(
        landmass = ifelse
              value < 1000,      # test if value is less than 1k sq mi
              "island",          # if true, it's an island
              "continent"        # if false, it's a continent
        )
    ) %>%
    print()
```

- results:

console: name <chr> / value <dbl> / landmass <chr>

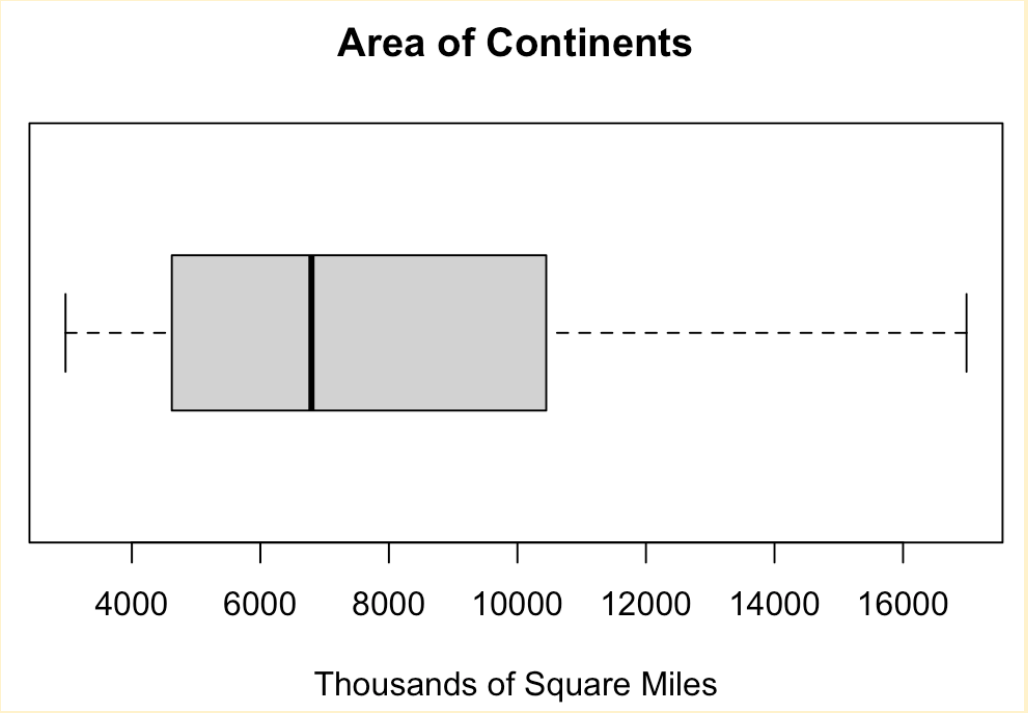
```
> islands4 %<>%
+   mutate(
+     landmass = ifelse(
+       value < 1000,  # Test if value is less than 1m sq mi
+       "island",      # If true, it's an island
+       "continent"    # If false, it's a continent
+     )
+   ) %>%
+   print()
# A tibble: 48 × 3
   name          value landmass
   <chr>         <dbl> <chr>
 1 Asia          16988 continent
 2 Africa        11506 continent
 3 North America  9390 continent
 4 South America  6795 continent
 5 Antarctica     5500 continent
 6 Europe         3745 continent
 7 Australia      2968 continent
 8 Greenland       840 island
 9 New Guinea      306 island
10 Borneo          280 island
# ℹ 38 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Boxplot of area of continents:

```
islands4 %>%
    filter(landmass == "continent") %>%
    select(value) %>%
    boxplot(
       horizontal = T,
       main = "Area of Continents",
       xlab = "Thousands of Square Miles"
    )
```

- results:

No outliers

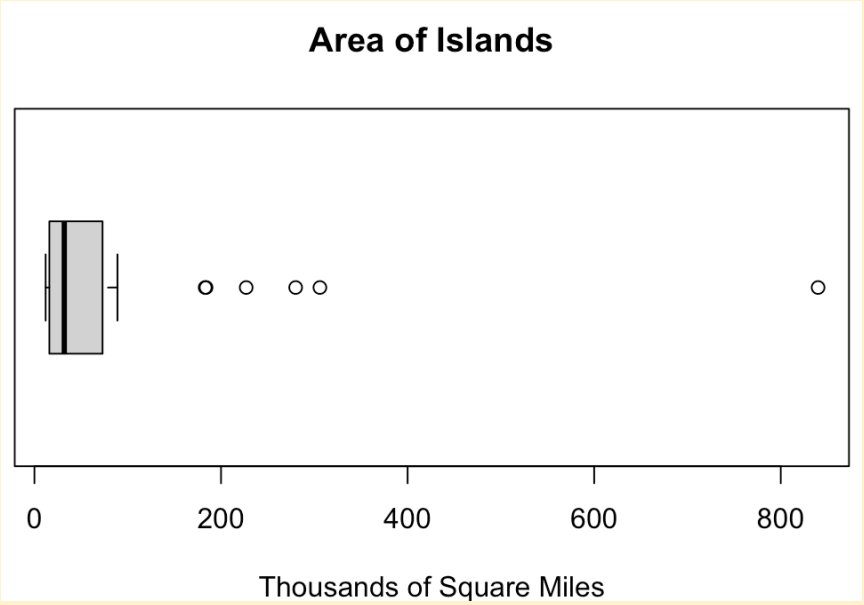


Boxplot of islands:

```
islands4 %>%
    filter(landmass == "island") %>%
    select(value) %>%
    boxplot(
       horizontal = T,
       main = "Area of Islands",
       xlab = "Thousands of Square Miles"
    )
```

- results:

Greenland is still outlier - but at least you can see what the box looks like now



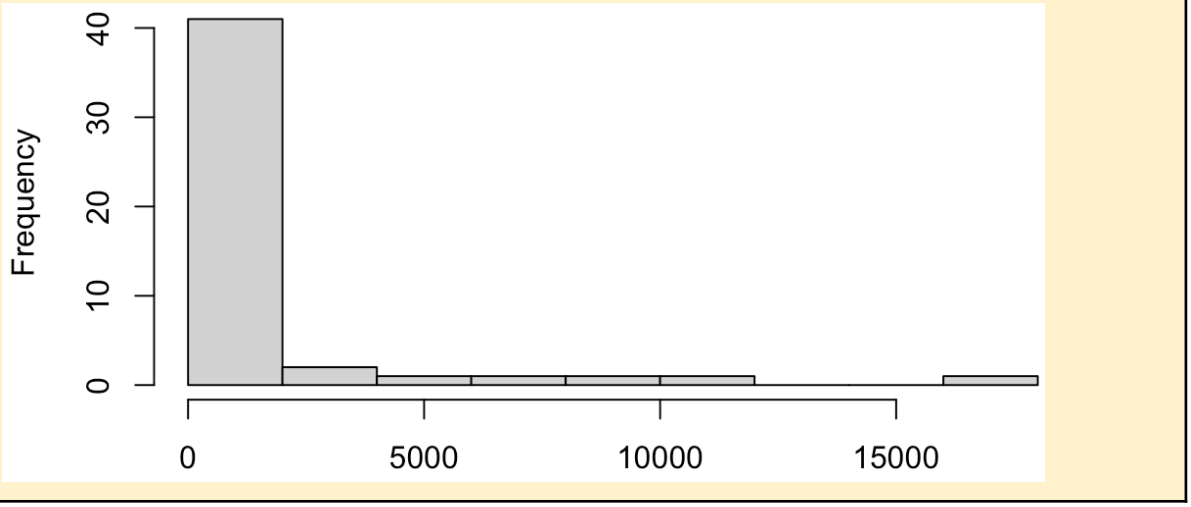
PART4: ***TRANSFORMING DATA***

- linear transformation - do transformation to all the data
- IF positively skewed data - and all values at least 1 THEN logarithm

Strong positive skew:

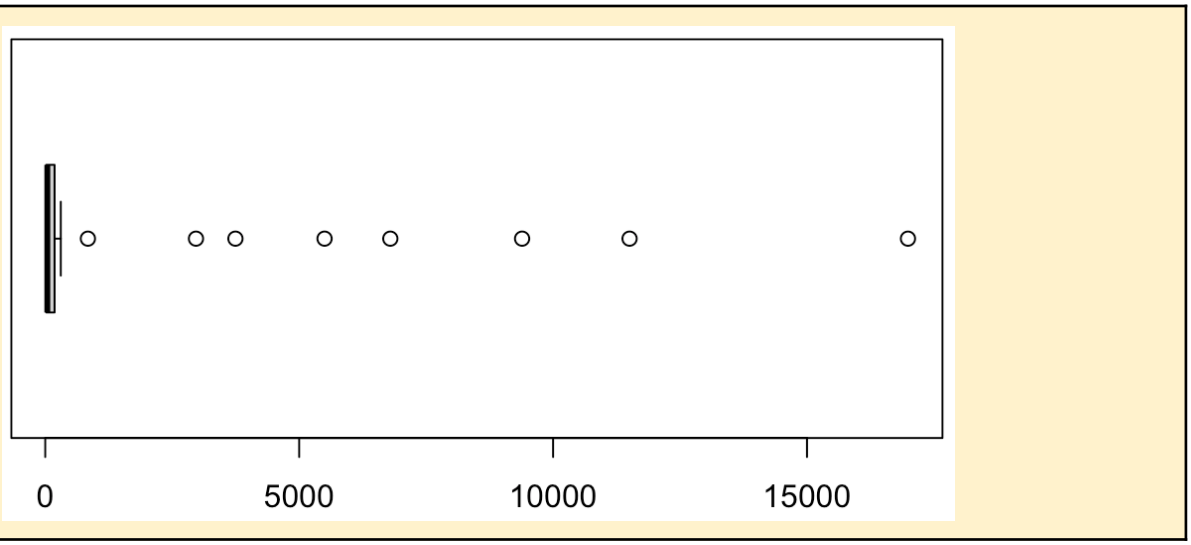
```
islands %>% histogram(main = NULL)
```

- result:




```
islands %>% boxplot(horizontal = T)
```

- result:

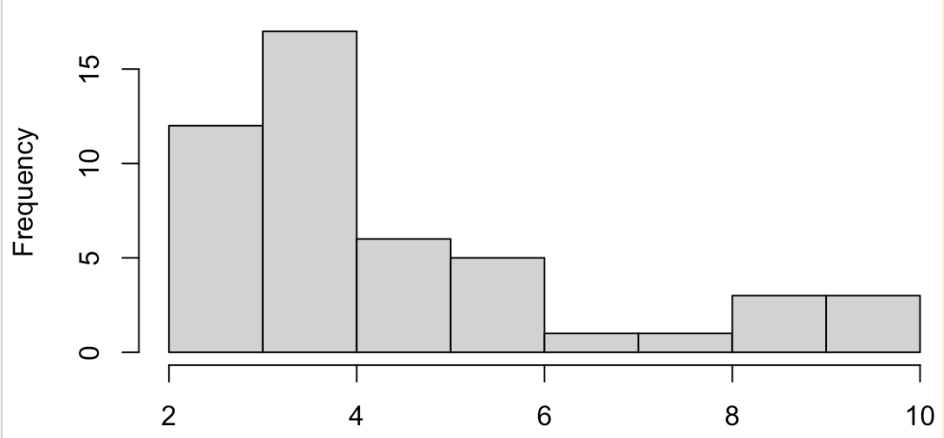


Log is natural logarithm with base e (base 10 is log10) > brings in extreme positive values:

```
islands %>% log() %>% histogram(main = NULL)
```

- result:

Still outliers , but not miles away from others




```
islands %>% log() %>% boxplot(horizontal = T)
```

- result:

Some outliers, but box is predominant