Introduce Cleansing Techniques

Hanh Nguyen

We often wish to tidy and reshape a dataset so that we can create certain plots. Here I introduce the two packages **tidyr** and **reshape2** to help the need and also to see how functions in **tidyr** and **reshape2** overlap and differ.
We first compare the functions gather(), separate() and spread(), from tidyr, with the functions melt(), colsplit() and dcast(), from reshape2.

The original dataset

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

tidyr package

gather {tidyr}: takes multiple columns and collapses into key-value pairs, duplicating all other columns as needed. You use gather() when you notice that you have columns that are not variables.
Simply put, gather() takes wide-format data and turns it into long-format data

```
iris.tidyr <- iris %>%
  gather(key,value,-Species)
```

```
##   Species          key value
## 1  setosa Sepal.Length   5.1
## 2  setosa Sepal.Length   4.9
## 3  setosa Sepal.Length   4.7
## 4  setosa Sepal.Length   4.6
## 5  setosa Sepal.Length   5.0
## 6  setosa Sepal.Length   5.4
```

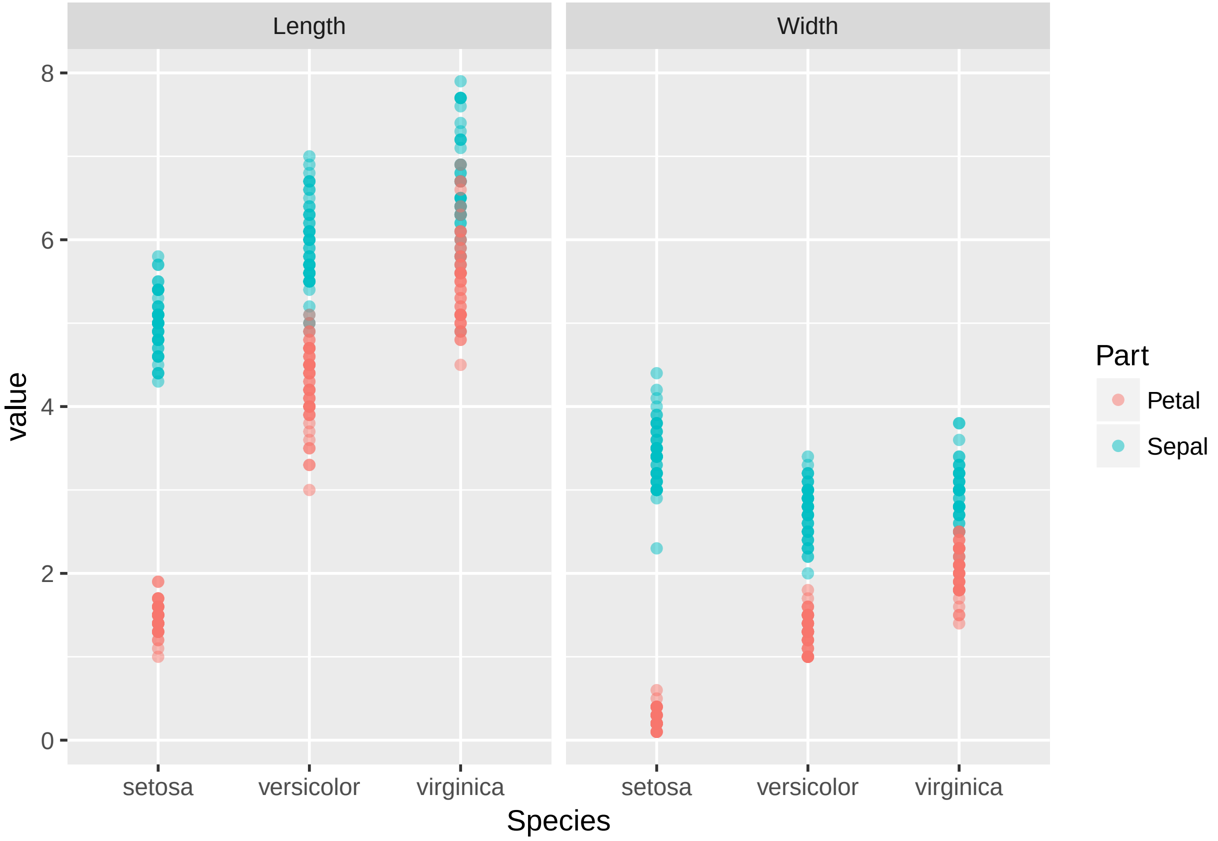
Our next step is to split the column key into two different columns: Part of a flower (Sepal or Petal) and Measure of that part (Length or Width), hence we use separate() function.
separate {tidyr}: turns a single character column into multiple columns.

```
iris.tidyr <- iris %>%
  gather(key,value,-Species) %>%
  separate(key,into=c("Part","Measure"),sep="\\.")
```

```
##   Species  Part Measure value
## 1  setosa Sepal  Length   5.1
## 2  setosa Sepal  Length   4.9
## 3  setosa Sepal  Length   4.7
## 4  setosa Sepal  Length   4.6
## 5  setosa Sepal  Length   5.0
## 6  setosa Sepal  Length   5.4
```

With this dataset structure, we now can create a plot as shown below.

```
iris.tidyr %>%
  ggplot(aes(x = Species, y = value, col = Part)) +
  geom_point(alpha =0.5) +
  facet_grid(. ~ Measure)
```



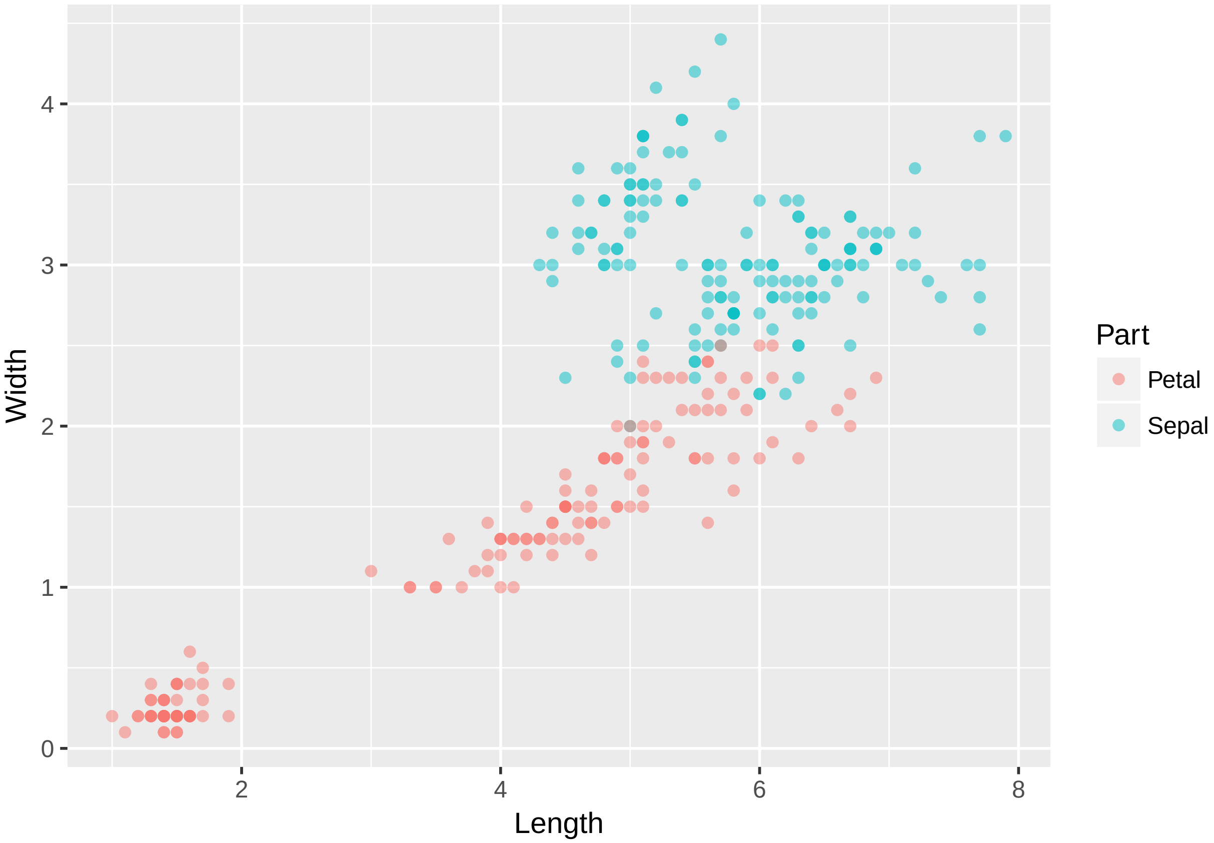
spread {tidyr}: spreads a key-value pair across multiple columns. In contrast to gather(), spread() takes long-format data and turns it into wide-format data.

```
iris$Flower <- 1:nrow(iris)
iris.tidyr <- iris %>%
  gather(key, value, - Species, - Flower) %>%
  separate(key, c("Part", "Measure"), "\\.") %>%
  spread(Measure, value)
```

```
##   Species Flower  Part Length Width
## 1  setosa      1 Petal    1.4   0.2
## 2  setosa      1 Sepal    5.1   3.5
## 3  setosa      2 Petal    1.4   0.2
## 4  setosa      2 Sepal    4.9   3.0
## 5  setosa      3 Petal    1.3   0.2
## 6  setosa      3 Sepal    4.7   3.2
```

With this dataset structure, we now can create a plot as shown below.

```
iris.tidyr %>%
  ggplot(aes(x=Length,y=Width,col=Part)) +
  geom_point(alpha=0.5)
```



reshape2 package

melt {reshape2}: converts an object into a molten data frame, giving same result with the gather() function from tidyr.
However, gather() cannot handle matrices or arrays, while melt() can!

```
iris.re <- iris %>%
  melt(id.vars="Species")
```

```
##   Species     variable value
## 1  setosa Sepal.Length   5.1
## 2  setosa Sepal.Length   4.9
## 3  setosa Sepal.Length   4.7
## 4  setosa Sepal.Length   4.6
## 5  setosa Sepal.Length   5.0
## 6  setosa Sepal.Length   5.4
```

colsplit {reshape2}: splits variable names that is a combination of multiple variables.
Again, we can achieve the same result with separate() function from tidyr, however, colsplit() operates only on a single column so we use cbind() to insert the new two columns in the data frame. While separate() performs all the operation at once.

```
iris$Flower <- 1:nrow(iris)
iris.re <- iris %>%
  melt(id.vars=c("Species","Flower"))
iris.re = cbind(Species=iris.re[,1],
              Flower=iris.re[,2],
                colsplit(iris.re[,3],"\\.",c("Part","Measure")),
                value=iris.re[,4])
```

```
##   Species Flower  Part Measure value
## 1  setosa      1 Sepal  Length   5.1
## 2  setosa      2 Sepal  Length   4.9
## 3  setosa      3 Sepal  Length   4.7
## 4  setosa      4 Sepal  Length   4.6
## 5  setosa      5 Sepal  Length   5.0
## 6  setosa      6 Sepal  Length   5.4
```

Again, the same result produced by spread() from tidyr can be obtained using dcast() from reshape2 by specifying the correct formula.
cast {reshape2}: casts a molten data frame into an array or data frame.

```
iris.re = dcast(iris.re, formula=Flower+Species+Part ~Measure)
```

```
##   Flower Species  Part Length Width
## 1      1  setosa Petal    1.4   0.2
## 2      1  setosa Sepal    5.1   3.5
## 3      2  setosa Petal    1.4   0.2
## 4      2  setosa Sepal    4.9   3.0
## 5      3  setosa Petal    1.3   0.2
## 6      3  setosa Sepal    4.7   3.2
```

Example

Next, we explore an MBTA ridership dataset. The Massachusetts Bay Transportation Authority ("MBTA" or just "the T" for short) manages America's oldest subway, as well as Greater Boston's commuter rail, ferry, and bus systems.
The dataset is stored as an Excel spreadsheet called mbta.xlsx. The first row is a title, so it needs to be skipped.

```
library(readxl)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
setwd("/Users/user/GitHub/data-vis")
mbta = read_excel("mbta.xlsx",skip=1)
```

First of all, we start with basic commands to explore the dataset.

```
head(mbta)
```

```
## # A tibble: 6 × 60
##    X__1             mode `2007-01` `2007-02` `2007-03` `2007-04` `2007-05`
##   <dbl>            <chr>     <chr>     <chr>     <dbl>     <chr>     <chr>
## 1     1 All Modes by Qtr        NA        NA  1187.653        NA        NA
## 2     2             Boat         4       3.6    40.000       4.3       4.9
## 3     3              Bus   335.819   338.675   339.867   352.162   354.367
## 4     4    Commuter Rail     142.2     138.5   137.700     139.5       139
## 5     5       Heavy Rail   435.294   448.271   458.583   472.201   474.579
## 6     6       Light Rail   227.231   240.262   241.444   255.557   248.262
## # ... with 53 more variables: `2007-06` <dbl>, `2007-07` <chr>,
## #   `2007-08` <chr>, `2007-09` <dbl>, `2007-10` <chr>, `2007-11` <chr>,
## #   `2007-12` <dbl>, `2008-01` <chr>, `2008-02` <chr>, `2008-03` <dbl>,
## #   `2008-04` <chr>, `2008-05` <chr>, `2008-06` <dbl>, `2008-07` <chr>,
## #   `2008-08` <chr>, `2008-09` <dbl>, `2008-10` <chr>, `2008-11` <chr>,
## #   `2008-12` <dbl>, `2009-01` <chr>, `2009-02` <chr>, `2009-03` <dbl>,
## #   `2009-04` <chr>, `2009-05` <chr>, `2009-06` <dbl>, `2009-07` <chr>,
## #   `2009-08` <chr>, `2009-09` <dbl>, `2009-10` <chr>, `2009-11` <chr>,
## #   `2009-12` <dbl>, `2010-01` <chr>, `2010-02` <chr>, `2010-03` <dbl>,
## #   `2010-04` <chr>, `2010-05` <chr>, `2010-06` <dbl>, `2010-07` <chr>,
## #   `2010-08` <chr>, `2010-09` <dbl>, `2010-10` <chr>, `2010-11` <chr>,
## #   `2010-12` <dbl>, `2011-01` <chr>, `2011-02` <chr>, `2011-03` <dbl>,
## #   `2011-04` <chr>, `2011-05` <chr>, `2011-06` <dbl>, `2011-07` <chr>,
## #   `2011-08` <chr>, `2011-09` <dbl>, `2011-10` <chr>
```

```r
str(mbta)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    11 obs. of  60 variables:
##  $ X__1   : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ mode   : chr  "All Modes by Qtr" "Boat" "Bus" "Commuter Rail" ...
##  $ 2007-01: chr  "NA" "4" "335.819" "142.2" ...
##  $ 2007-02: chr  "NA" "3.6" "338.675" "138.5" ...
##  $ 2007-03: num  1188 40 340 138 459 ...
##  $ 2007-04: chr  "NA" "4.3" "352.162" "139.5" ...
##  $ 2007-05: chr  "NA" "4.9" "354.367" "139" ...
##  $ 2007-06: num  1246 5.8 350.5 143 477 ...
##  $ 2007-07: chr  "NA" "6.521" "357.519" "142.391" ...
##  $ 2007-08: chr  "NA" "6.572" "355.479" "142.364" ...
##  $ 2007-09: num  1256.57 5.47 372.6 143.05 499.57 ...
##  $ 2007-10: chr  "NA" "5.145" "368.847" "146.542" ...
##  $ 2007-11: chr  "NA" "3.763" "330.826" "145.089" ...
##  $ 2007-12: num  1216.89 2.98 312.92 141.59 448.27 ...
##  $ 2008-01: chr  "NA" "3.175" "340.324" "142.145" ...
##  $ 2008-02: chr  "NA" "3.111" "352.905" "142.607" ...
##  $ 2008-03: num  1253.52 3.51 361.15 137.45 494.05 ...
##  $ 2008-04: chr  "NA" "4.164" "368.189" "140.389" ...
##  $ 2008-05: chr  "NA" "4.015" "363.903" "142.585" ...
##  $ 2008-06: num  1314.82 5.19 362.96 142.06 518.35 ...
##  $ 2008-07: chr  "NA" "6.016" "370.921" "145.731" ...
##  $ 2008-08: chr  "NA" "5.8" "361.057" "144.565" ...
##  $ 2008-09: num  1307.04 4.59 389.54 141.91 517.32 ...
##  $ 2008-10: chr  "NA" "4.285" "357.974" "151.957" ...
##  $ 2008-11: chr  "NA" "3.488" "345.423" "152.952" ...
##  $ 2008-12: num  1232.65 3.01 325.77 140.81 446.74 ...
##  $ 2009-01: chr  "NA" "3.014" "338.532" "141.448" ...
##  $ 2009-02: chr  "NA" "3.196" "360.412" "143.529" ...
##  $ 2009-03: num  1209.79 3.33 353.69 142.89 467.22 ...
##  $ 2009-04: chr  "NA" "4.049" "359.38" "142.34" ...
##  $ 2009-05: chr  "NA" "4.119" "354.75" "144.225" ...
##  $ 2009-06: num  1233.1 4.9 347.9 142 473.1 ...
##  $ 2009-07: chr  "NA" "6.444" "339.477" "137.691" ...
##  $ 2009-08: chr  "NA" "5.903" "332.661" "139.158" ...
##  $ 2009-09: num  1230.5 4.7 374.3 139.1 500.4 ...
##  $ 2009-10: chr  "NA" "4.212" "385.868" "137.104" ...
##  $ 2009-11: chr  "NA" "3.576" "366.98" "129.343" ...
##  $ 2009-12: num  1207.85 3.11 332.39 126.07 440.93 ...
##  $ 2010-01: chr  "NA" "3.207" "362.226" "130.91" ...
##  $ 2010-02: chr  "NA" "3.195" "361.138" "131.918" ...
##  $ 2010-03: num  1208.86 3.48 373.44 131.25 483.4 ...
##  $ 2010-04: chr  "NA" "4.452" "378.611" "131.722" ...
##  $ 2010-05: chr  "NA" "4.415" "380.171" "128.8" ...
##  $ 2010-06: num  1244.41 5.41 363.27 129.14 490.26 ...
##  $ 2010-07: chr  "NA" "6.513" "353.04" "122.935" ...
##  $ 2010-08: chr  "NA" "6.269" "343.688" "129.732" ...
##  $ 2010-09: num  1225.5 4.7 381.6 132.9 521.1 ...
##  $ 2010-10: chr  "NA" "4.402" "384.987" "131.033" ...
##  $ 2010-11: chr  "NA" "3.731" "367.955" "130.889" ...
##  $ 2010-12: num  1216.26 3.16 326.34 121.42 450.43 ...
##  $ 2011-01: chr  "NA" "3.14" "334.958" "128.396" ...
##  $ 2011-02: chr  "NA" "3.284" "346.234" "125.463" ...
##  $ 2011-03: num  1223.45 3.67 380.4 134.37 516.73 ...
##  $ 2011-04: chr  "NA" "4.251" "380.446" "134.169" ...
##  $ 2011-05: chr  "NA" "4.431" "385.289" "136.14" ...
##  $ 2011-06: num  1302.41 5.47 376.32 135.58 529.53 ...
##  $ 2011-07: chr  "NA" "6.581" "361.585" "132.41" ...
##  $ 2011-08: chr  "NA" "6.733" "353.793" "130.616" ...
##  $ 2011-09: num  1291 5 388 137 550 ...
##  $ 2011-10: chr  "NA" "4.484" "398.456" "128.72" ...
```

```r
summary(mbta)
```

```
##       X__1          mode             2007-01            2007-02
##  Min.   : 1.0   Length:11          Length:11          Length:11
##  1st Qu.: 3.5   Class :character   Class :character   Class :character
##  Median : 6.0   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 6.0
##  3rd Qu.: 8.5
##  Max.   :11.0
##     2007-03           2007-04            2007-05
##  Min.   :   0.114   Length:11          Length:11
##  1st Qu.:   9.278   Class :character   Class :character
##  Median : 137.700   Mode  :character   Mode  :character
##  Mean   : 330.293
##  3rd Qu.: 399.225
##  Max.   :1204.725
##     2007-06           2007-07            2007-08
##  Min.   :   0.096   Length:11          Length:11
##  1st Qu.:   5.700   Class :character   Class :character
##  Median : 143.000   Mode  :character   Mode  :character
##  Mean   : 339.846
##  3rd Qu.: 413.788
##  Max.   :1246.129
##     2007-09           2007-10            2007-11
##  Min.   :  -0.007   Length:11          Length:11
##  1st Qu.:   5.539   Class :character   Class :character
##  Median : 143.051   Mode  :character   Mode  :character  
##  Mean   : 352.554                                        
##  3rd Qu.: 436.082                                        
##  Max.   :1310.764                                        
##     2007-12          2008-01           2008-02         
##  Min.   :  -0.060   Length:11          Length:11         
##  1st Qu.:   4.385   Class :character   Class :character  
##  Median : 141.585   Mode  :character   Mode  :character  
##  Mean   : 321.588                                        
##  3rd Qu.: 380.594                                        
##  Max.   :1216.890                                        
##     2008-03          2008-04           2008-05         
##  Min.   :   0.058   Length:11          Length:11         
##  1st Qu.:   5.170   Class :character   Class :character  
##  Median : 137.453   Mode  :character   Mode  :character  
##  Mean   : 345.604                                        
##  3rd Qu.: 427.601                                        
##  Max.   :1274.031                                        
##     2008-06          2008-07           2008-08         
##  Min.   :   0.060   Length:11          Length:11         
##  1st Qu.:   5.742   Class :character   Class :character  
##  Median : 142.057   Mode  :character   Mode  :character  
##  Mean   : 359.667                                        
##  3rd Qu.: 440.656                                        
##  Max.   :1320.728                                        
##     2008-09          2008-10           2008-11         
##  Min.   :   0.021   Length:11          Length:11         
##  1st Qu.:   5.691   Class :character   Class :character  
##  Median : 141.907   Mode  :character   Mode  :character  
##  Mean   : 362.099                                        
##  3rd Qu.: 453.430                                        
##  Max.   :1338.015                                        
##     2008-12          2009-01           2009-02         
##  Min.   :  -0.015   Length:11          Length:11         
##  1st Qu.:   4.689   Class :character   Class :character  
##  Median : 140.810   Mode  :character   Mode  :character  
##  Mean   : 319.882                                        
##  3rd Qu.: 386.255                                        
##  Max.   :1232.655                                        
##     2009-03          2009-04           2009-05         
##  Min.   :  -0.050   Length:11          Length:11         
##  1st Qu.:   5.003   Class :character   Class :character  
##  Median : 142.893   Mode  :character   Mode  :character  
##  Mean   : 330.142                                        
##  3rd Qu.: 410.455                                        
##  Max.   :1210.912                                        
##     2009-06          2009-07           2009-08         
##  Min.   :  -0.079   Length:11          Length:11         
##  1st Qu.:   5.845   Class :character   Class :character  
##  Median : 142.006   Mode  :character   Mode  :character  
##  Mean   : 333.194                                        
##  3rd Qu.: 410.482                                        
##  Max.   :1233.085                                        
##     2009-09          2009-10           2009-11         
##  Min.   :  -0.035   Length:11          Length:11
##  1st Qu.:   5.693   Class :character   Class :character
##  Median : 139.087   Mode  :character   Mode  :character
##  Mean   : 346.687
##  3rd Qu.: 437.332
##  Max.   :1291.564
##     2009-12            2010-01            2010-02
##  Min.   :  -0.022   Length:11          Length:11
##  1st Qu.:   4.784   Class :character   Class :character
##  Median : 126.066   Mode  :character   Mode  :character
##  Mean   : 312.962
##  3rd Qu.: 386.659
##  Max.   :1207.845
##     2010-03            2010-04            2010-05
##  Min.   :   0.012   Length:11          Length:11
##  1st Qu.:   5.274   Class :character   Class :character
##  Median : 131.252   Mode  :character   Mode  :character
##  Mean   : 332.726
##  3rd Qu.: 428.420
##  Max.   :1225.556
##     2010-06            2010-07            2010-08
##  Min.   :   0.008   Length:11          Length:11
##  1st Qu.:   6.436   Class :character   Class :character
##  Median : 129.144   Mode  :character   Mode  :character
##  Mean   : 335.964
##  3rd Qu.: 426.769
##  Max.   :1244.409
##     2010-09            2010-10            2010-11
##  Min.   :   0.001   Length:11          Length:11
##  1st Qu.:   5.567   Class :character   Class :character
##  Median : 132.892   Mode  :character   Mode  :character
##  Mean   : 346.524
##  3rd Qu.: 451.361
##  Max.   :1293.117
##     2010-12            2011-01            2011-02
##  Min.   :  -0.004   Length:11          Length:11
##  1st Qu.:   4.466   Class :character   Class :character
##  Median : 121.422   Mode  :character   Mode  :character
##  Mean   : 312.917
##  3rd Qu.: 388.385
##  Max.   :1216.262
##     2011-03           2011-04            2011-05
##  Min.   :   0.05   Length:11          Length:11
##  1st Qu.:   6.03   Class :character   Class :character
##  Median : 134.37   Mode  :character   Mode  :character
##  Mean   : 345.17
##  3rd Qu.: 448.56
##  Max.   :1286.66
##     2011-06            2011-07            2011-08
##  Min.   :   0.054   Length:11          Length:11
##  1st Qu.:   6.926   Class :character   Class :character
##  Median : 135.581   Mode  :character   Mode  :character
##  Mean   : 353.331
##  3rd Qu.: 452.923
##  Max.   :1302.414
##     2011-09           2011-10
##  Min.   :   0.043   Length:11
##  1st Qu.:   6.660   Class :character
##  Median : 136.901   Mode  :character
##  Mean   : 362.555
##  3rd Qu.: 469.204
##  Max.   :1348.754
```

There're some unnecessary rows and columns. All of the NA values are stored in the All Modes by Qtr row. This row is a quarterly average of weekday MBTA ridership and since this dataset tracks monthly average ridership, it can be removed. Similarly, the 7th row (Pct Chg / Yr) and the 11th row (TOTAL) are not really observations and will be removed. The first column also needs to be removed because it's just listing the row numbers.

```
mbta = mbta[-c(1, 7, 11), ]
mbta = mbta[, -1]
```

The different modes of transportation (commuter rail, bus, subway, ferry, etc.) are variables, providing information about each month's average ridership. The months themselves are observations. The variables are stored in rows instead of columns and since we actually want to represent variables in columns rather than rows, we use the **gather()** and **separate()** functions from the tidyr package.

Also, we change the average weekday ridership column, thou_riders, into numeric values rather than character strings.

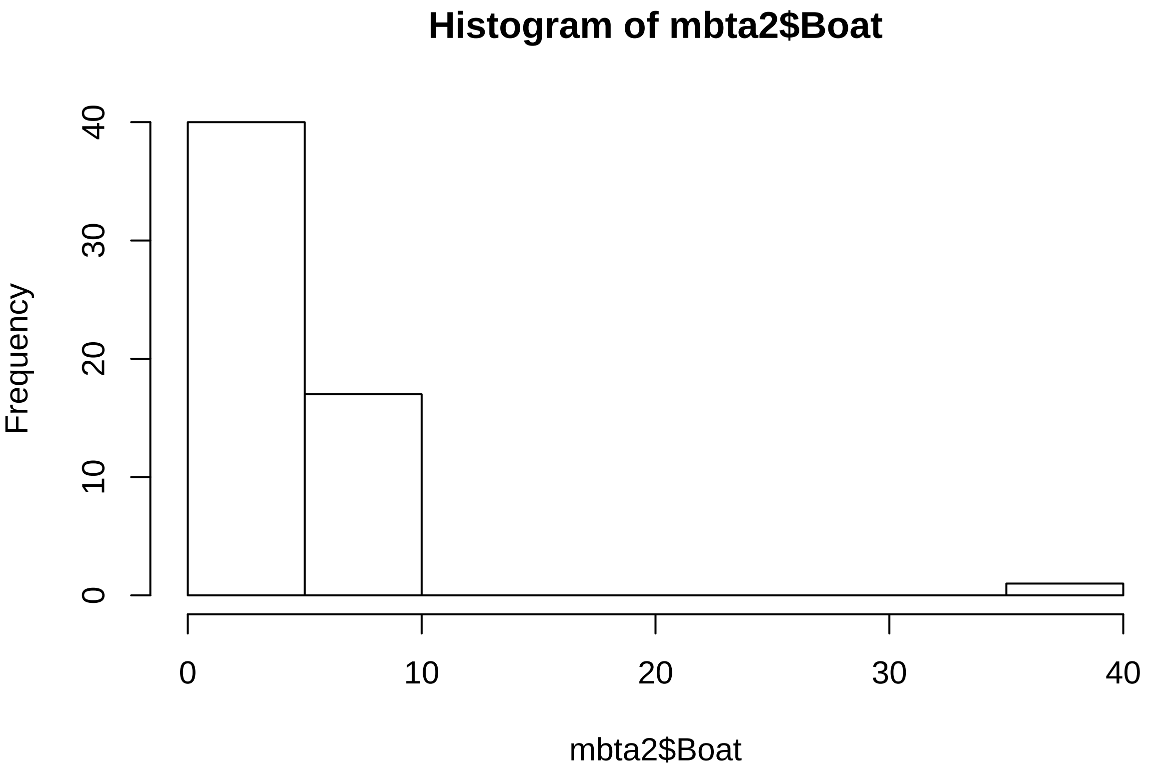
```
mbta2 = mbta %>%
  gather(month, thou_riders, -mode)
mbta2$thou_riders = as.numeric(mbta2$thou_riders)
mbta2 = mbta2 %>%
  spread(mode,thou_riders) %>%
  separate(month, into=c("year","month"),sep="-")
```

By running *summary(mbta2), hist(mbta2$Boat)*, we see that every value of the Boat column clustered around 4 and one loner out around 40.

```
summary(mbta2)
```

```
##      year              month                 Boat             Bus
##  Length:58          Length:58          Min.   : 2.985   Min.   :312.9
##  Class :character   Class :character   1st Qu.: 3.494   1st Qu.:345.6
##  Mode  :character   Mode  :character   Median : 4.293   Median :359.9
##                                        Mean   : 5.068   Mean   :358.6
##                                        3rd Qu.: 5.356   3rd Qu.:372.2
##                                        Max.   :40.000   Max.   :398.5
##  Commuter Rail     Heavy Rail      Light Rail      Private Bus
##  Min.   :121.4   Min.   :435.3   Min.   :194.4   Min.   :2.213
##  1st Qu.:131.4   1st Qu.:471.1   1st Qu.:220.6   1st Qu.:2.641
##  Median :138.8   Median :487.3   Median :231.9   Median :2.820
##  Mean   :137.4   Mean   :489.3   Mean   :233.0   Mean   :3.352
##  3rd Qu.:142.4   3rd Qu.:511.3   3rd Qu.:244.5   3rd Qu.:4.167
##  Max.   :153.0   Max.   :554.9   Max.   :271.1   Max.   :4.878
##       RIDE       Trackless Trolley
##  Min.   :4.900   Min.   : 5.777
##  1st Qu.:5.965   1st Qu.:11.679
##  Median :6.615   Median :12.598
##  Mean   :6.604   Mean   :12.125
##  3rd Qu.:7.149   3rd Qu.:13.320
##  Max.   :8.598   Max.   :15.109
```

```
hist(mbta2$Boat)
```



Every month, average weekday commuter boat ridership was on either side of four thousand. Then, one month it jumped to 40 thousand without warning? This value is likely an error as being accidentally typed 40 instead of 4. Therefore, we'll locate the incorrect value and change it to 4.

```
i = which(mbta2$Boat > 30)
mbta2$Boat[i] = 4
```

A quick look at the new dataset

```
summary(mbta2)
```

```
##      year              month                Boat             Bus
##  Length:58          Length:58          Min.   :2.985   Min.   :312.9
##  Class :character   Class :character   1st Qu.:3.494   1st Qu.:345.6
##  Mode  :character   Mode  :character   Median :4.268   Median :359.9
##                                        Mean   :4.447   Mean   :358.6
##                                        3rd Qu.:5.178   3rd Qu.:372.2
##                                        Max.   :6.733   Max.   :398.5
##  Commuter Rail     Heavy Rail      Light Rail     Private Bus
##  Min.   :121.4   Min.   :435.3   Min.   :194.4   Min.   :2.213
##  1st Qu.:131.4   1st Qu.:471.1   1st Qu.:220.6   1st Qu.:2.641
##  Median :138.8   Median :487.3   Median :231.9   Median :2.820
##  Mean   :137.4   Mean   :489.3   Mean   :233.0   Mean   :3.352
##  3rd Qu.:142.4   3rd Qu.:511.3   3rd Qu.:244.5   3rd Qu.:4.167
##  Max.   :153.0   Max.   :554.9   Max.   :271.1   Max.   :4.878
##       RIDE       Trackless Trolley
##  Min.   :4.900   Min.   : 5.777
##  1st Qu.:5.965   1st Qu.:11.679
##  Median :6.615   Median :12.598
##  Mean   :6.604   Mean   :12.125
##  3rd Qu.:7.149   3rd Qu.:13.320
##  Max.   :8.598   Max.   :15.109
```

```r
head(mbta2)
```

```
## # A tibble: 6 × 10
##    year month  Boat     Bus `Commuter Rail` `Heavy Rail` `Light Rail`
##   <chr> <chr> <dbl>   <dbl>           <dbl>        <dbl>        <dbl>
## 1  2007    01   4.0 335.819           142.2      435.294      227.231
## 2  2007    02   3.6 338.675           138.5      448.271      240.262
## 3  2007    03   4.0 339.867           137.7      458.583      241.444
## 4  2007    04   4.3 352.162           139.5      472.201      255.557
## 5  2007    05   4.9 354.367           139.0      474.579      248.262
## 6  2007    06   5.8 350.543           143.0      477.032      246.108
## # ... with 3 more variables: `Private Bus` <dbl>, RIDE <dbl>, `Trackless
## #   Trolley` <dbl>
```

Sources:
https://blog.rstudio.org/2014/07/22/introducing-tidyr/
http://www.milanor.net/blog/reshape-data-r-tidyr-vs-reshape2/
https://www.datacamp.com/courses/importing-cleaning-data-in-r-case-studies