BIOS 611: Project 1

Wonkyung Jang

Background: What is "The Urban Ministries of Durham (UMD) Project"?

The Urban Ministries of Durham (UMD) Project aims for connecting with the community to diminish homelessness and fighting poverty through providing food, shelter and a future to neighbors with special needs (UMD, 2019). Through exploring some hidden patterns of the data using data science analytics, this paper will suggest what evidence social workers can glean from this data and help them better understand their clients and services.

Data Cited

The dataset is offered by the Urban Ministries of Durham (UMD) Project Team (http://www.umdurham.org/), which includes the dataset with 79838 observations from 1990's to 2019. It has 9 variables like below:

Variables

1. Client File Number (Identifier)
2. Bus Tickets: Service discontinued
3. Food: # of people in the family for which food was provided
4. Food Pounds: # of pounds of food that each individual or family received when shopping the food pantry
5. Clothing Items: # of clothing items that each individual or family received in the clothing closet
6. Diapers: # of packs of diapers received (individuals/families are given 2 packs of diapers per child, and packs contain 22 diapers on average)
7. School Kits
8. Hygiene Kits: # of kits received per individual or family. Kits contain soap, shampoo, conditioner, lotion, deodorant, a toothbrush, toothpaste, a washcloth, a disposable razor, and a bottle of shaving cream.
9. Financial Support: Service discontinued

```r
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------ tidyverse 1.2.1 -
```

```
## v ggplot2 3.1.1       v purrr   0.3.2
## v tibble  2.0.1       v dplyr   0.8.0.1
## v tidyr   0.8.2       v stringr 1.4.0
## v readr   1.3.1       v forcats 0.3.0
```

```
## -- Conflicts --------------------------------------------------------------- tidyverse_conflicts() -
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(dplyr)
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(ggplot2)
library(grDevices)
library(EnvStats)
```

```
##
## Attaching package: 'EnvStats'
```

```
## The following objects are masked from 'package:stats':
##
##     predict, predict.lm
```

```
## The following object is masked from 'package:base':
##
##     print.default
```

```r
library(forecast)
library(tidyquant)
```

```
## Loading required package: lubridate
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
## Loading required package: PerformanceAnalytics
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
##
## Attaching package: 'xts'
```

```
## The following objects are masked from 'package:dplyr':
##
##     first, last
```

```
##
## Attaching package: 'PerformanceAnalytics'
```

```
## The following objects are masked from 'package:EnvStats':
##
##     kurtosis, skewness
```

```
## The following object is masked from 'package:graphics':
##
##     legend
```

```
## Loading required package: quantmod
```

```
## Loading required package: TTR
```

```
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```
## == Need to Learn tidyquant? ========================================================================
## Business Science offers a 1-hour course - Learning Lab #9: Performance Analysis & Portfolio Optimiza
## </> Learn more at: https://university.business-science.io/p/learning-labs-pro </>
```

```r
library(timetk)
library(sweep)
```

Analyses

Data Cleaning

1. Import the data

```r
rawdata <- read.table(file = 'UMD_Services_Provided_20190719.txt', sep = '\t', fill = TRUE, header = TRU
```

2. Rename the variables

```r
data <- rawdata %>%
  rename(ClientID = Client.File.Number, Bus = Bus.Tickets..Number.of., Note = Notes.of.Service,
         Food = Food.Provided.for, Clothing = Clothing.Items) %>%
  select(Date, ClientID, Bus, Food, Food.Pounds, Clothing, Diapers, School.Kits, Hygiene.Kits,
         Referrals, Note, Financial.Support)
```

3. Filtering

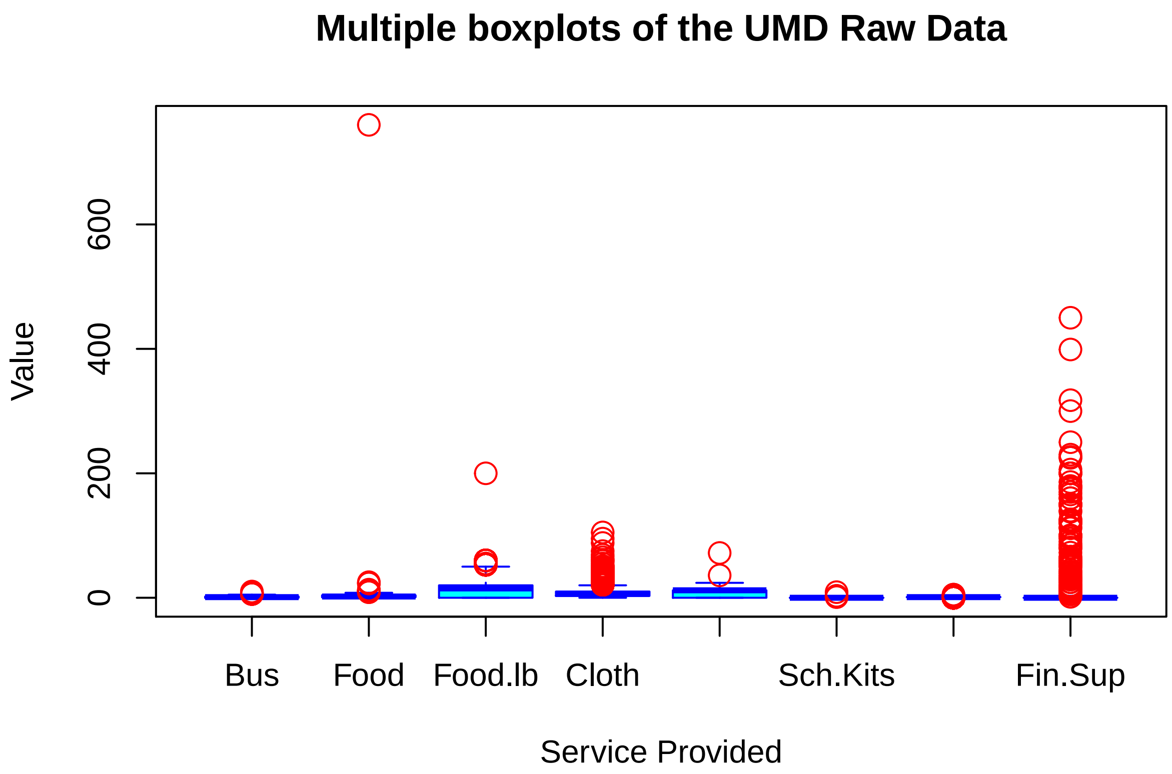
Given that the UMD was estabblished in 1983, we should remove some rows with dates before 1983 or after 2019. Here, the Date data was converted into DATE format.

```
data$Date <- as.Date(data$Date, format = "%m/%d/%Y")
data = data %>%
  filter(Date >= "1983-01-01" & Date <= "2019-10-01")
```

4. Trimming: detect, visualize and test for outliers

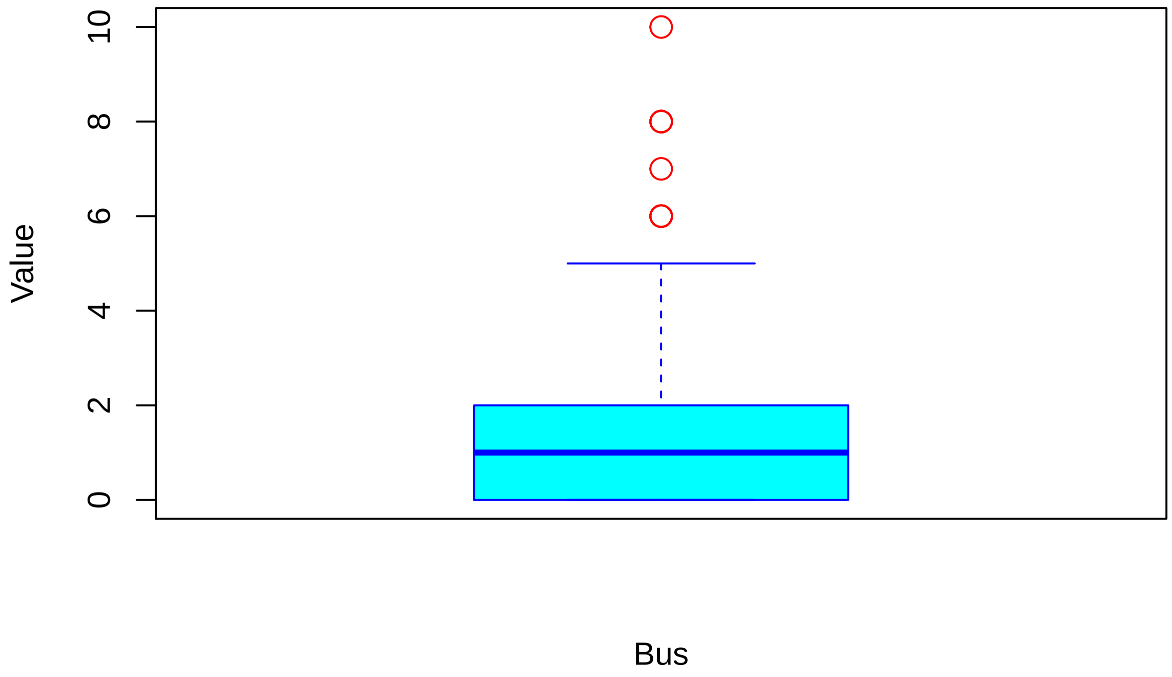
First, I have a look at columns of the UMD dataset with boxplot.

```
boxplot(data$Bus, data$Food, data$Food.Pounds, data$Cloth, data$Diapers, data$School.Kits,
        data$Hygiene.Kits, data$Financial.Support,
        main = "Multiple boxplots of the UMD Raw Data",
        xlab = "Service Provided", ylab = "Value",
        names = c("Bus", "Food", "Food.lb", "Cloth", "Diapers", "Sch.Kits", "Hyg.Kits",
                  "Fin.Sup"),
        col = "cyan", border = "Blue", outcol = "red", outcex = 1.5)
```



- Bus According to the box-plot and rosnertest, I got 4 outliers greater than or equal to 7. Particularly, through the rosnertest, I could get rows in which the outliers are and the actual values of the outliers.

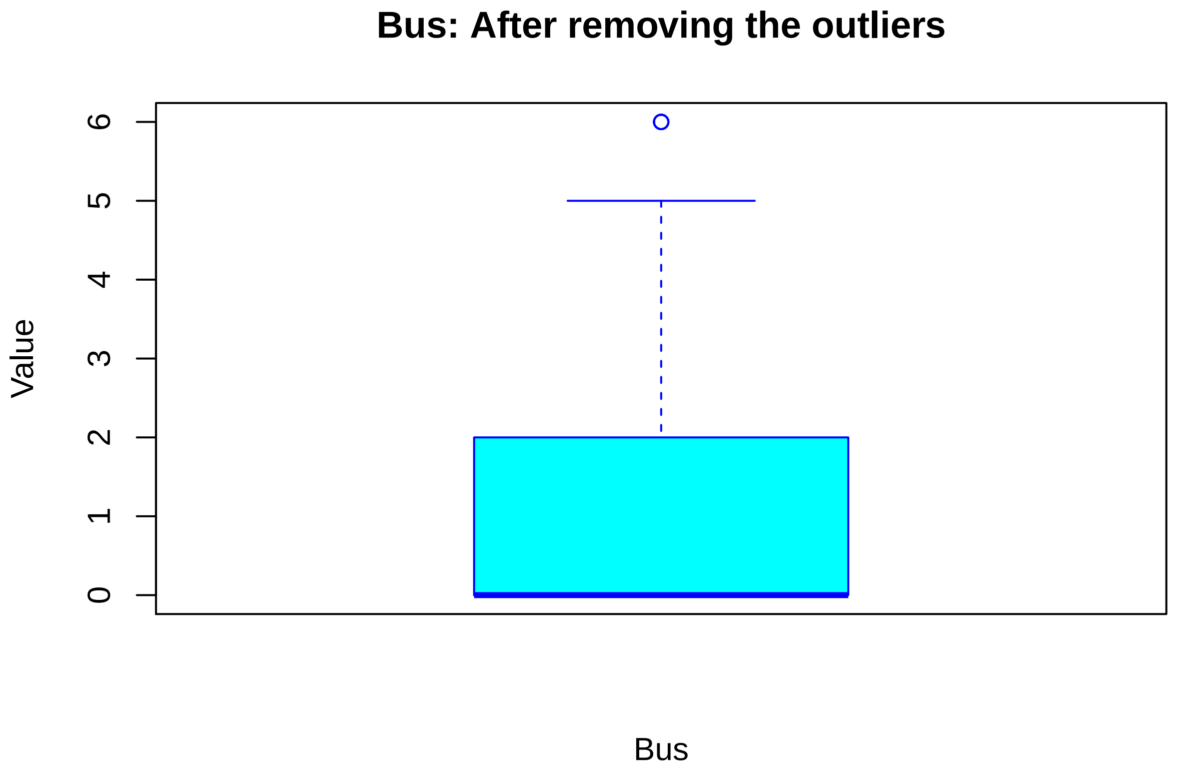
```
summary(data$Bus)
boxplot(data$Bus, outcol = "red", outcex = 1.5, xlab = "Bus", ylab = "Value", col = "cyan", border = "B
```



```
rosnerTest(data$Bus, k = 10, warn = F)
```

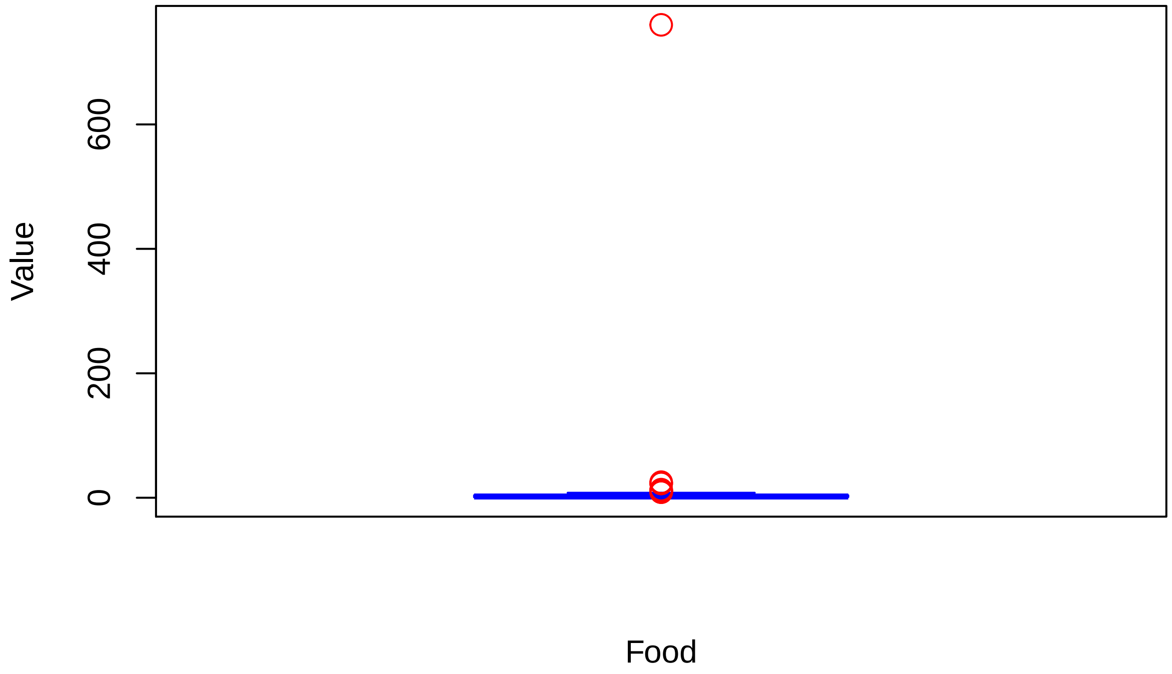
The rows containing the outliers were removed so that I could notice that those pesky outliers are gone.

```
data <- data[-which(data$Bus >= 7),]
boxplot(data$Bus, main = "Bus: After removing the outliers", xlab = "Bus", ylab = "Value", col = "cyan"
```



- Food According to the box-plot and rosnertest, I got 9 outliers greater than or equal to 11.

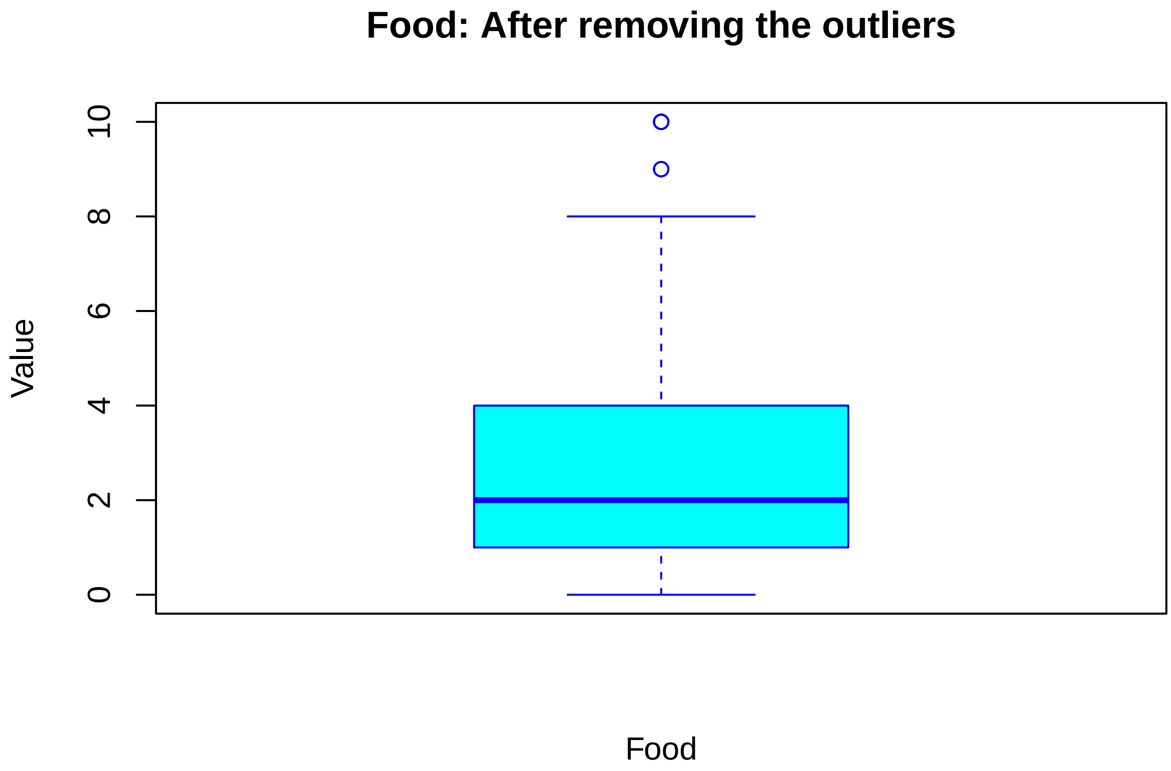
```
summary(data$Food)
boxplot(data$Food, outcol = "red", outcex = 1.5, xlab = "Food", ylab = "Value", col = "cyan", border =
```



```
rosnerTest(data$Food, k = 10, warn = F)
```

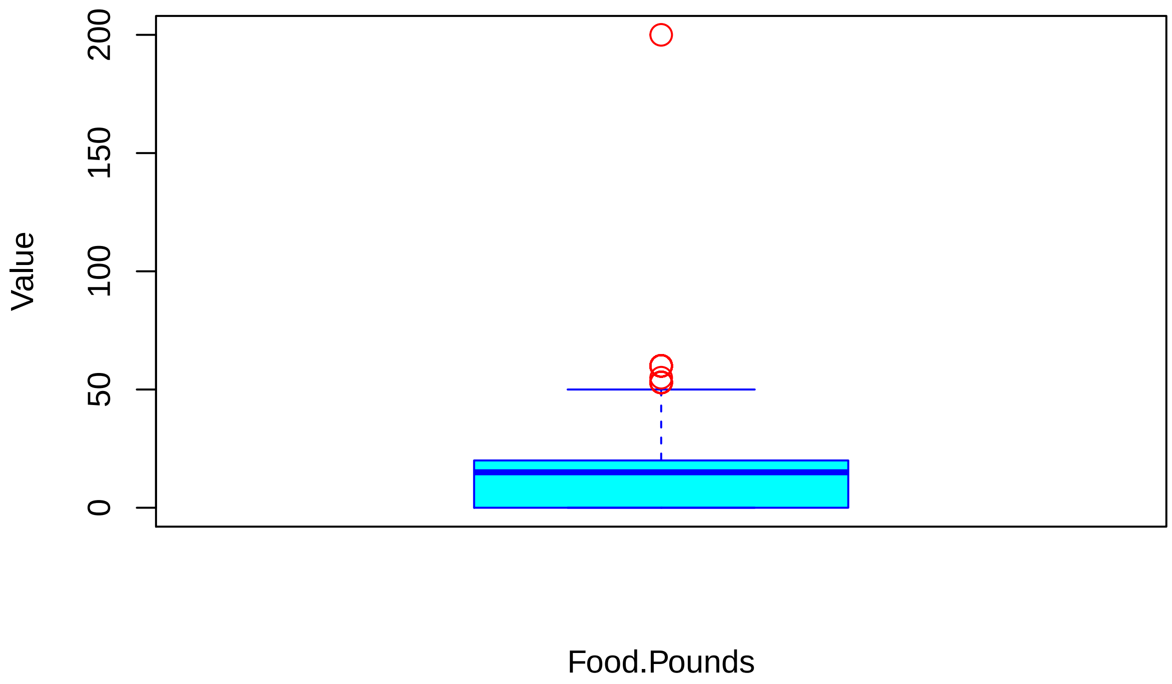
The rows containing the outliers were removed so that I could notice that those pesky outliers are gone.

```
data <- data[-which(data$Food >= 11),]
boxplot(data$Food, main = "Food: After removing the outliers", xlab = "Food", ylab = "Value", col = "cya
```



- Food.Pounds According to the box-plot and rosnertest, I got 1 outlier greater than or equal to 200.

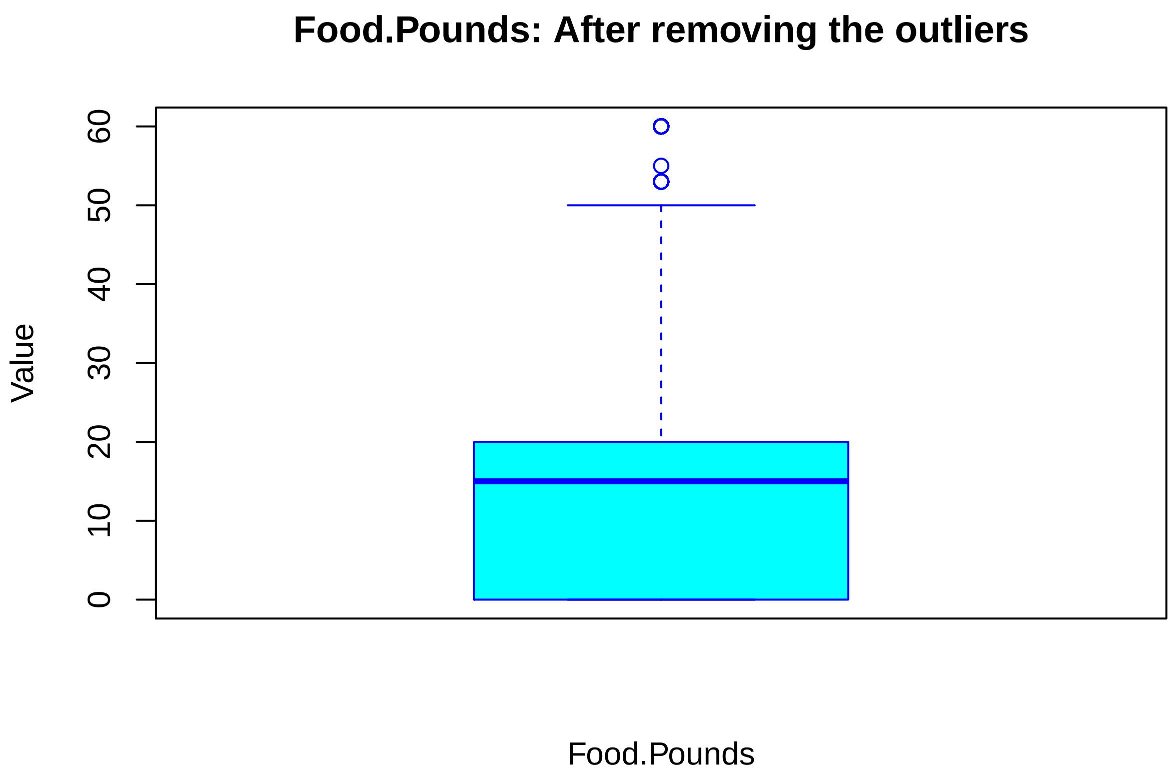
```
summary(data$Food.Pounds)
boxplot(data$Food.Pounds, outcol = "red", outcex = 1.5, xlab = "Food.Pounds", ylab = "Value", col = "cy
```



```
rosnerTest(data$Food.Pounds, k = 5, warn = F)
```

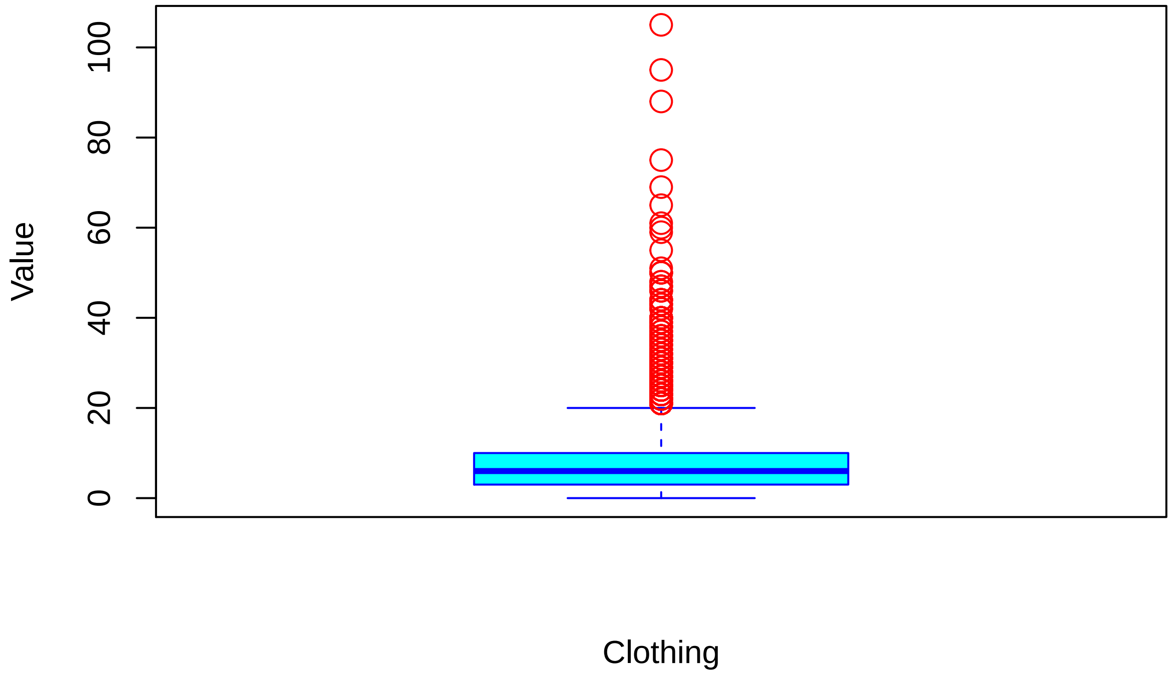
The outliers were excluded.

```
data <- data[-which(data$Food.Pounds >= 200),]
boxplot(data$Food.Pounds, main = "Food.Pounds: After removing the outliers", xlab = "Food.Pounds", ylab
```



- Clothing According to the box-plot and rosnertest, I got outliers greater than or equal to 36.

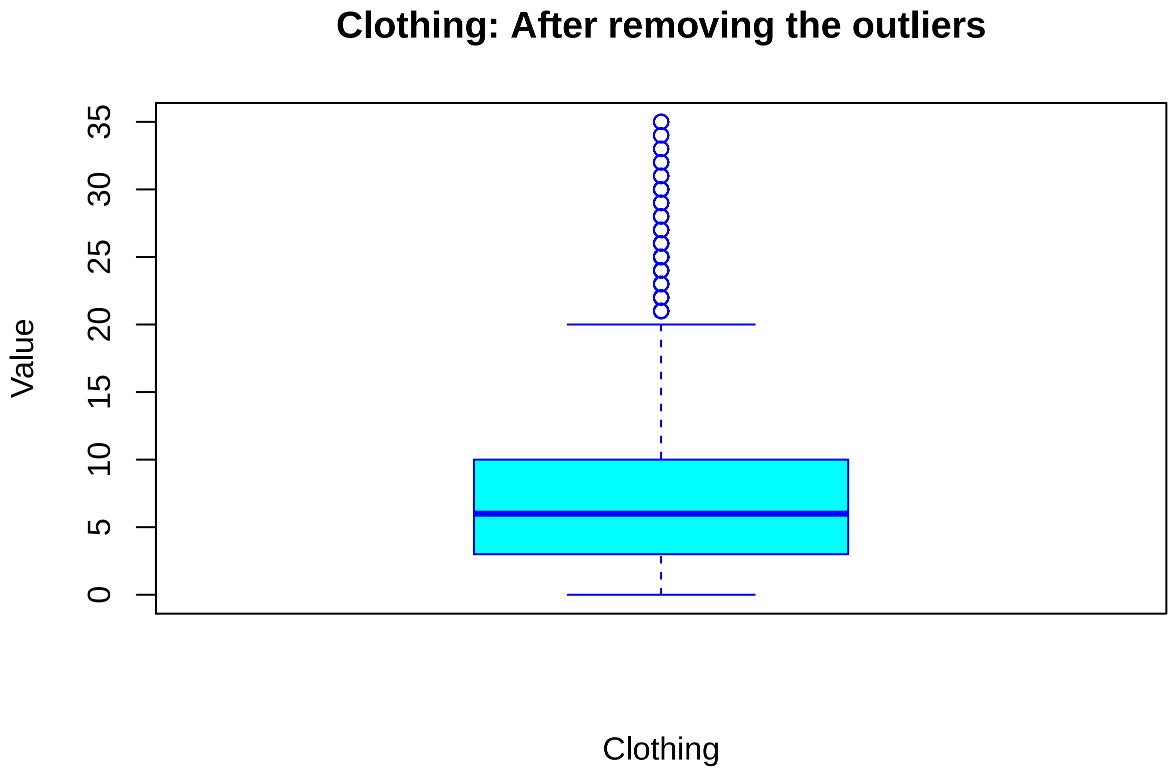
```
summary(data$Clothing)
boxplot(data$Clothing, outcol = "red", outcex = 1.5, xlab = "Clothing", ylab = "Value", col = "cyan", b
```



```
rosnerTest(data$Clothing, k = 100, warn = F)
```

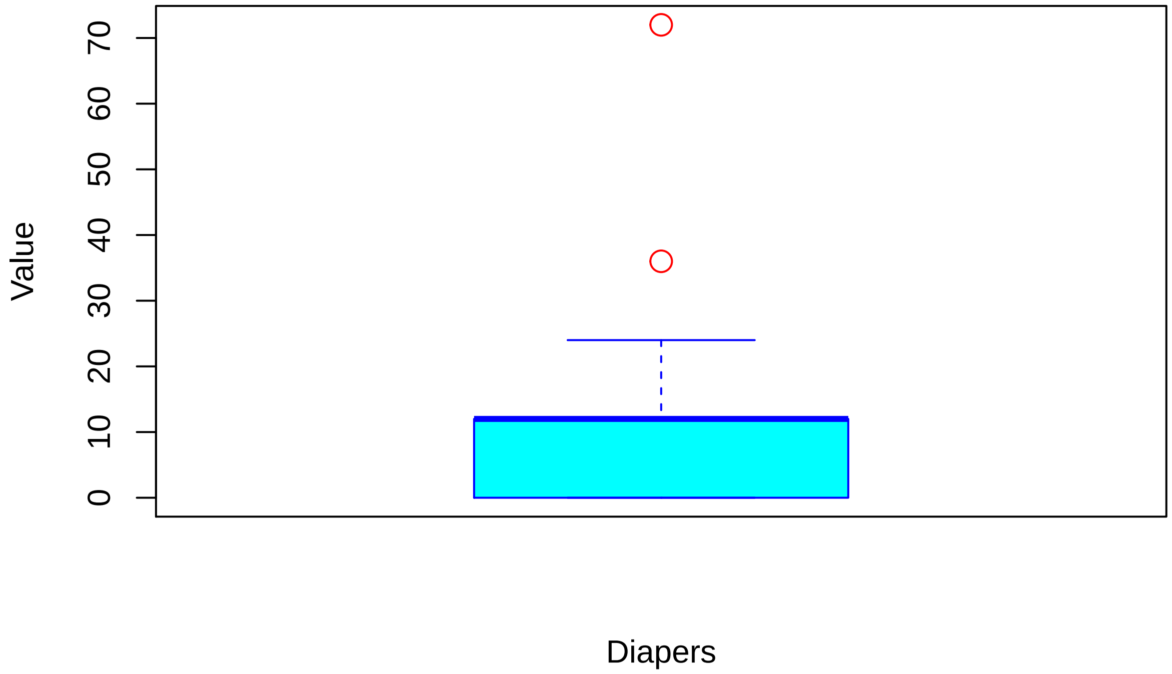
The outliers were excluded.

```
data <- data[-which(data$Clothing >= 36),]
boxplot(data$Clothing, main = "Clothing: After removing the outliers", xlab = "Clothing", ylab = "Value
```



- Diapers According to the box-plot and rosnertest, I got outliers greater than or equal to 36.

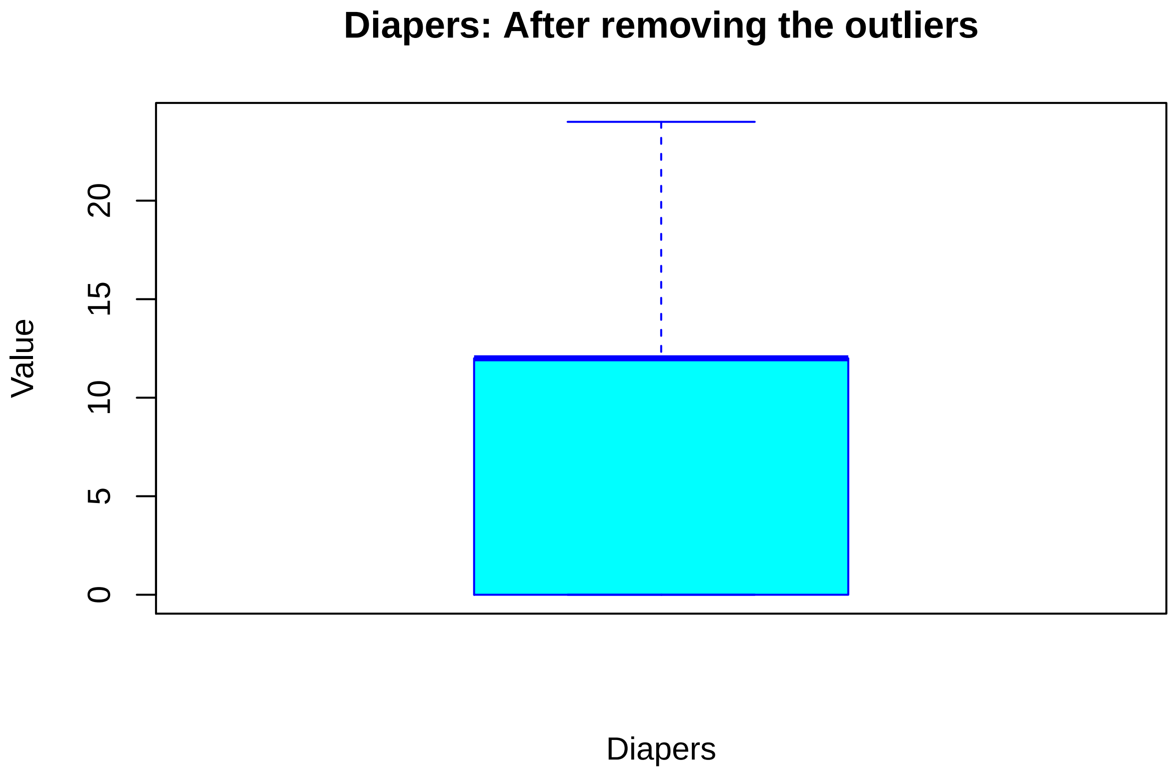
```
summary(data$Diapers)
boxplot(data$Diapers, outcol = "red", outcex = 1.5, xlab = "Diapers", ylab = "Value", col = "cyan", bord
```



```
rosnerTest(data$Diapers, k = 5, warn = F)
```

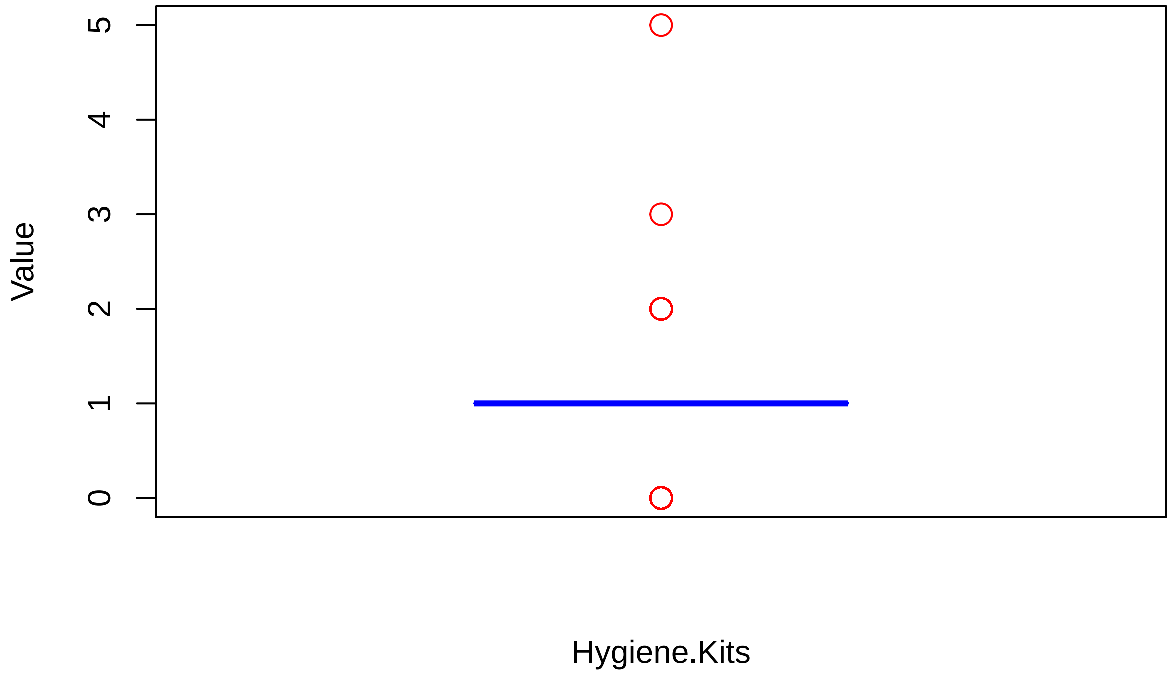
The outliers were excluded.

```
data <- data[-which(data$Diapers >= 36),]
boxplot(data$Diapers, main = "Diapers: After removing the outliers", xlab = "Diapers", ylab = "Value", 
```



- Hygiene.Kits According to the box-plot and rosnertest, I got outliers greater than or equal to 3.

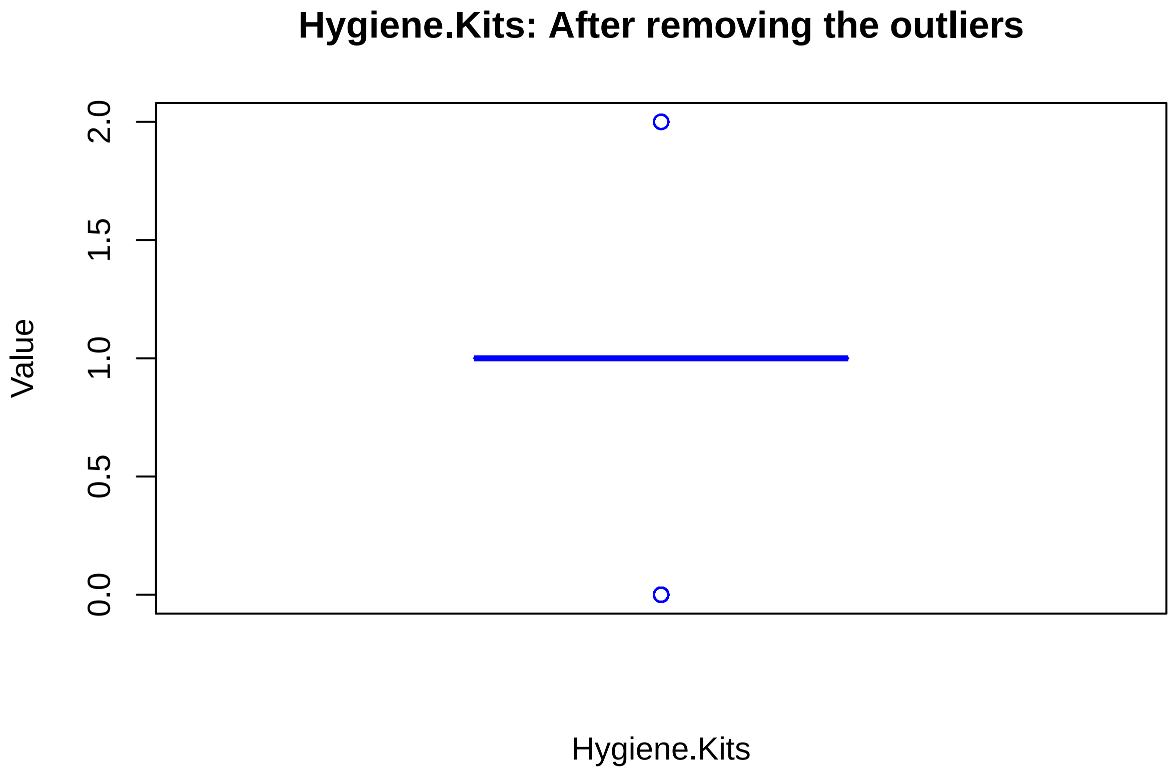
```
boxplot(data$Hygiene.Kits, outcol = "red", outcex = 1.5, xlab = "Hygiene.Kits", ylab = "Value", col = "
```



```
summary(data$Hygiene.Kits)
rosnerTest(data$Hygiene.Kits, k = 10, warn = F)
```

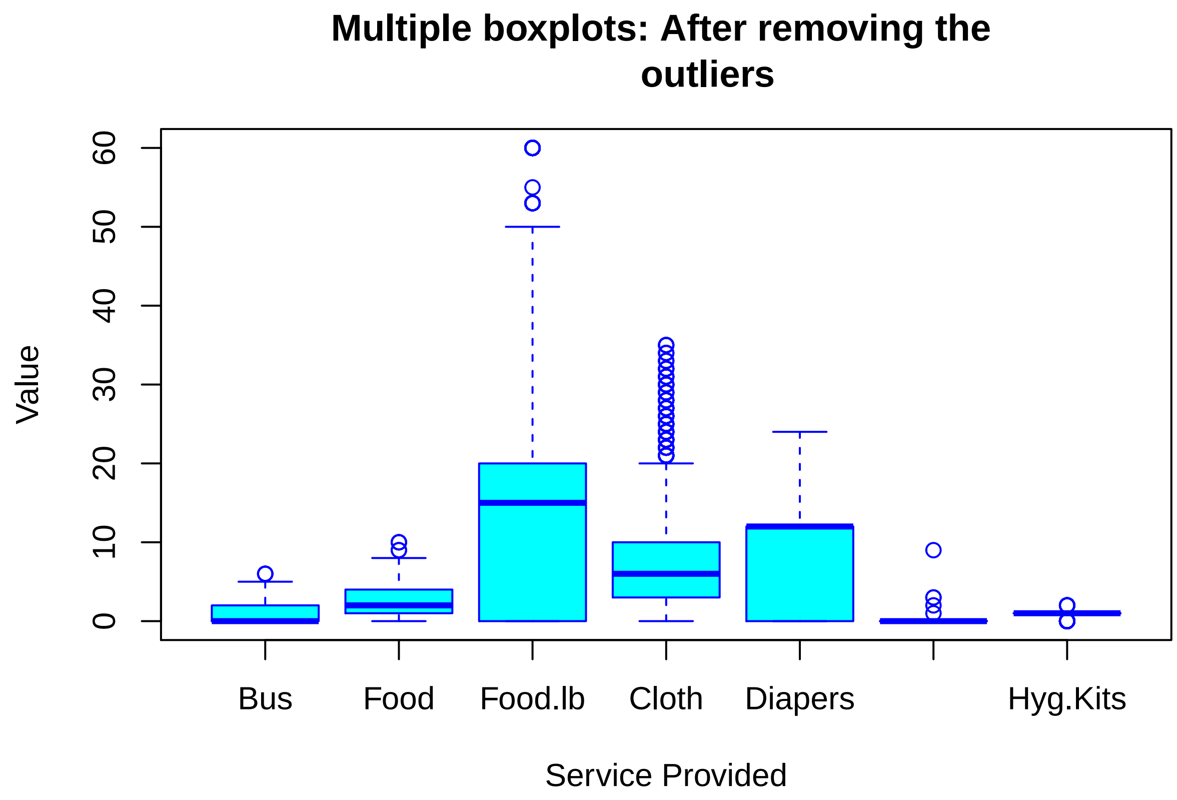
The outliers were excluded.

```
data <- data[-which(data$Hygiene.Kits >= 3),]
boxplot(data$Hygiene.Kits, main = "Hygiene.Kits: After removing the outliers", xlab = "Hygiene.Kits", yl
```



- Multiple boxplots

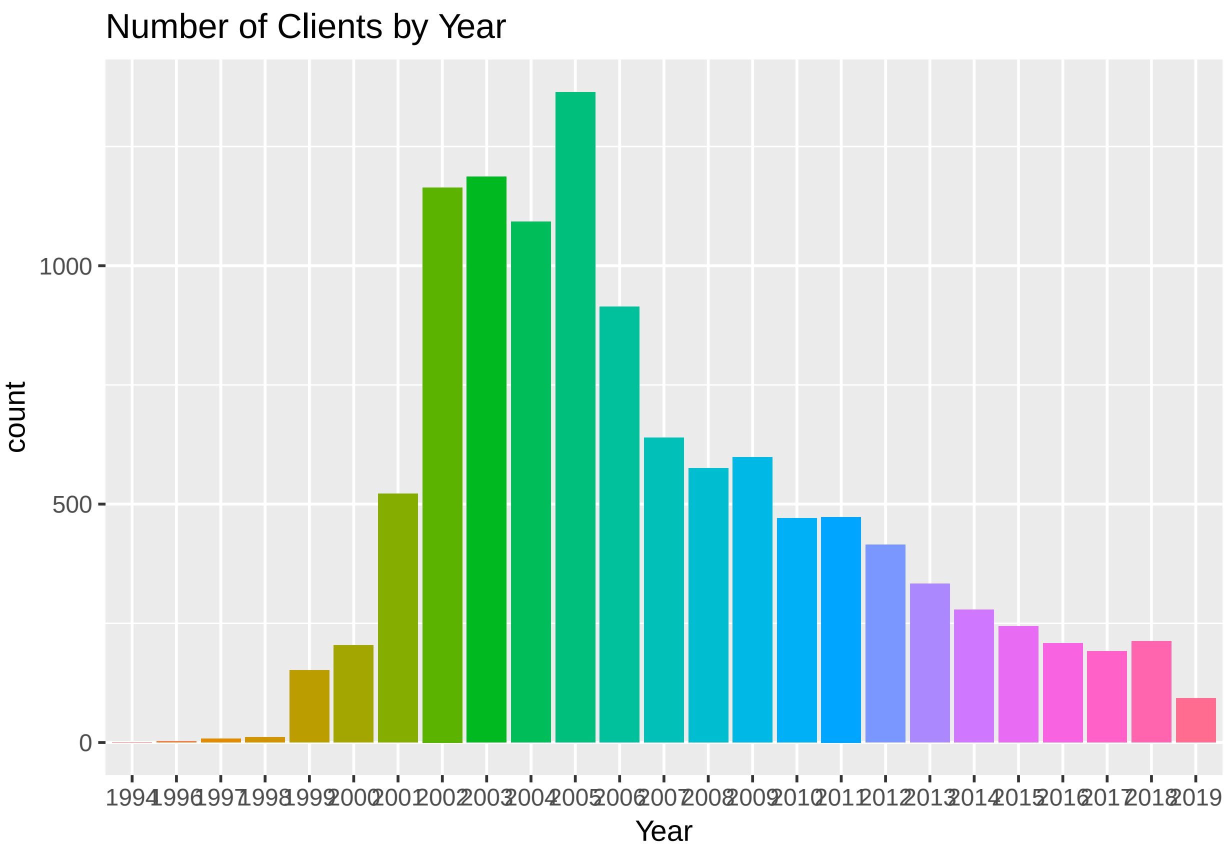
```
boxplot(data$Bus, data$Food, data$Food.Pounds, data$Cloth, data$Diapers, data$School.Kits,
        data$Hygiene.Kits, main = "Multiple boxplots: After removing the
        outliers", xlab = "Service Provided", ylab = "Value",
        names = c("Bus", "Food", "Food.lb", "Cloth", "Diapers", "Sch.Kits", "Hyg.Kits"),
        col = "cyan", border = "Blue")
```



The Number of (New) Clients Served by UMD and Duration of Assistance

1. The number of clients served by UMD between 1994 and 2019

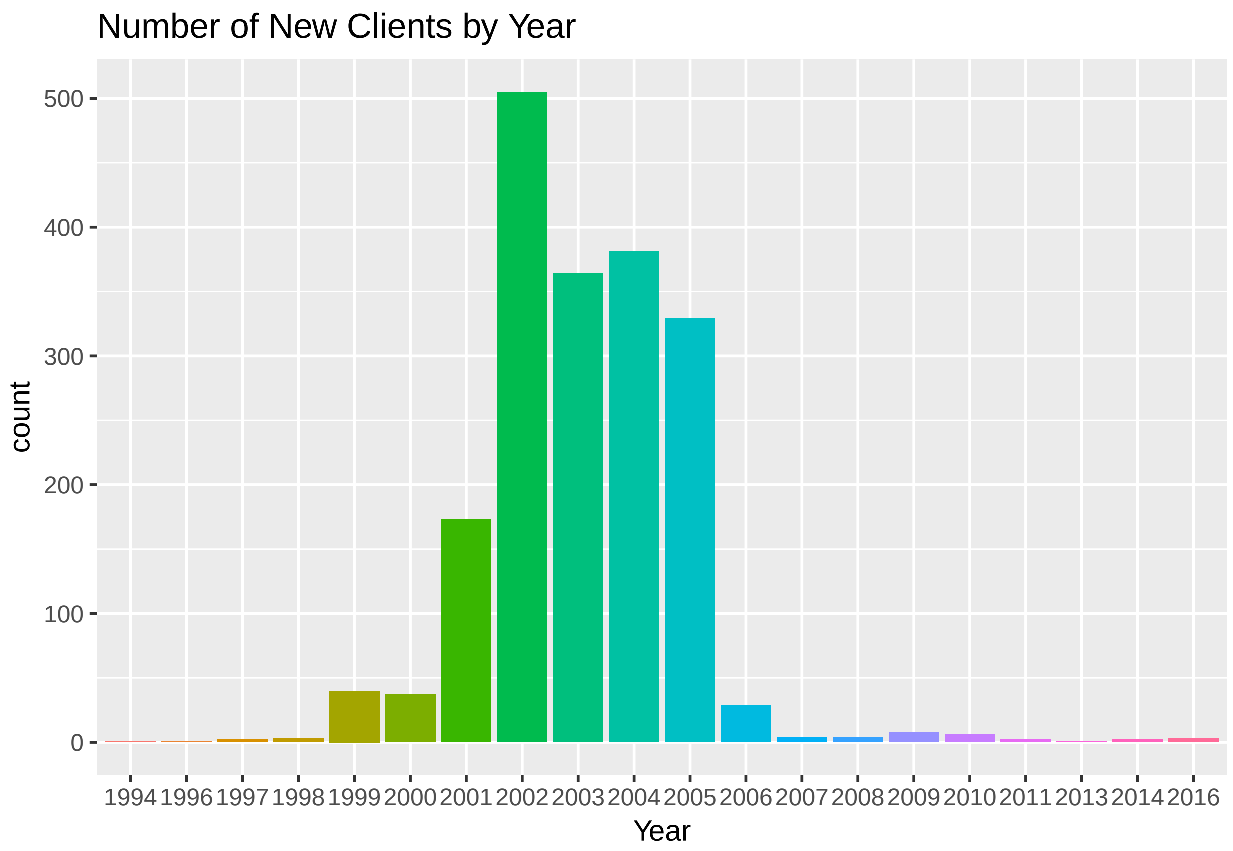
```
client.data = data %>%
  select(Date, ClientID) %>%
  drop_na() %>%
  separate(Date, sep = "-", into = c("Year", "Month", "Day"))
ggplot(data = client.data) +
  geom_bar(mapping = aes(x = Year, fill = Year)) +
  labs(title = "Number of Clients by Year") +
  theme(legend.position = "none")
```



2.The number of new clients served by UMD between 1994 and 2019

The number of unique client file numbers were counted with the function distinct().

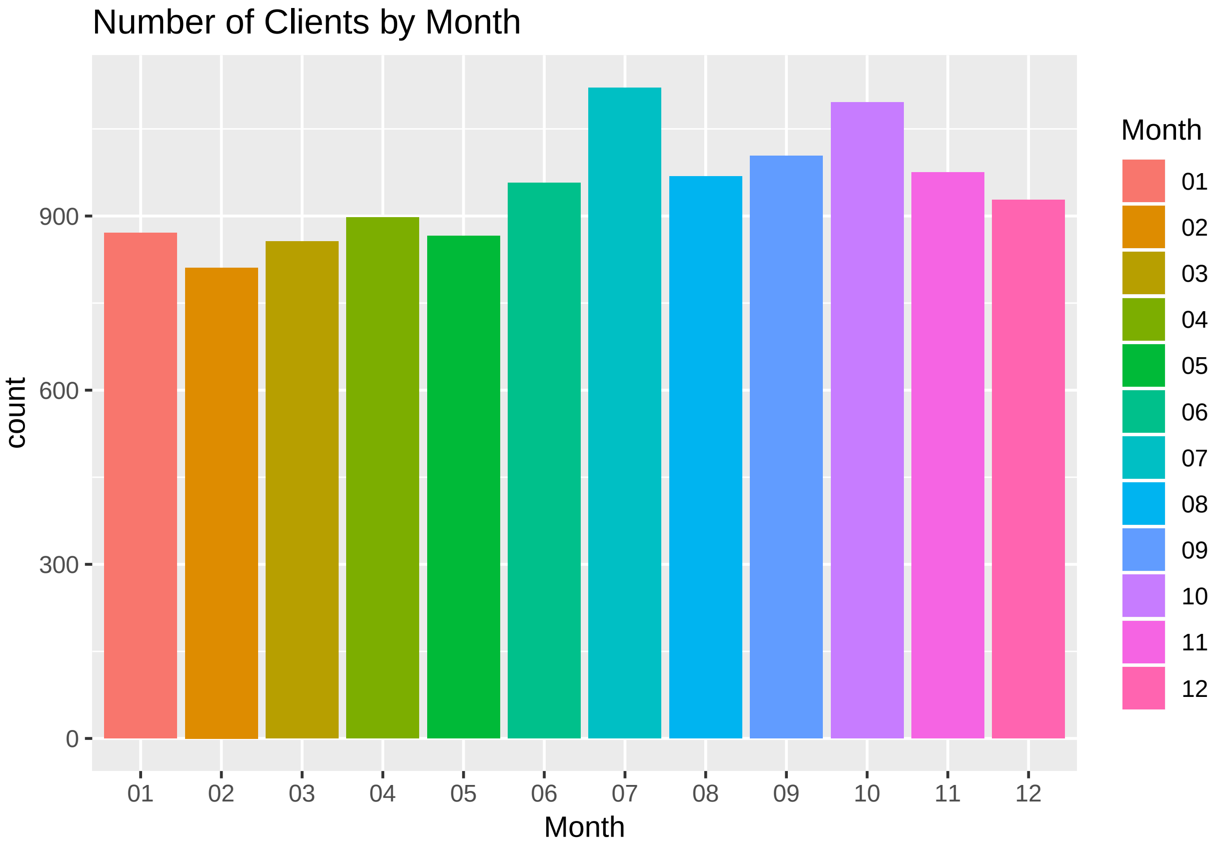
```
client.data = data %>%
  select(Date, ClientID) %>%
  drop_na() %>%
  separate(Date, sep = "-", into = c("Year", "Month", "Day")) %>%
  distinct(ClientID, .keep_all = TRUE)
ggplot(data = client.data) +
  geom_bar(mapping = aes(x = Year, fill = Year)) +
  labs(title = "Number of New Clients by Year") +
  theme(legend.position = "none")
```



The number of new clients served by UMD had been increasing until 2002 but has been decreasing up to now.

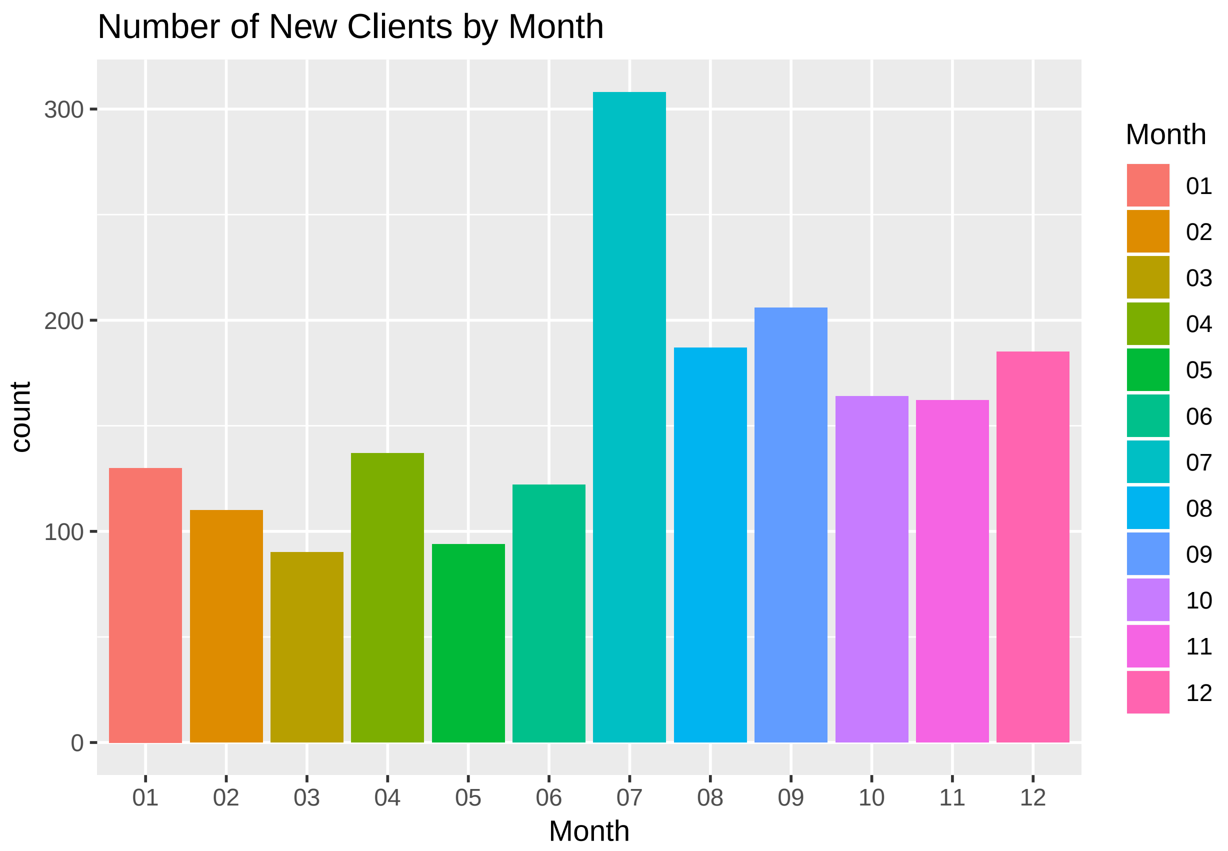
3. The Number of clients by month

```
client.data = data %>%
  select(Date, ClientID) %>%
  drop_na() %>%
  separate(Date, sep = "-", into = c("Year", "Month", "Day"))
ggplot(data = client.data) +
  geom_bar(mapping = aes(x = Month, fill = Month)) +
  labs(title = "Number of Clients by Month")
```



4. The Number of new clients by month

```
client.data = data %>%
  select(Date, ClientID) %>%
  drop_na() %>%
  separate(Date, sep = "-", into = c("Year", "Month", "Day")) %>%
  distinct(ClientID, .keep_all = TRUE)
ggplot(data = client.data) +
  geom_bar(mapping = aes(x = Month, fill = Month)) +
  labs(title = "Number of New Clients by Month")
```

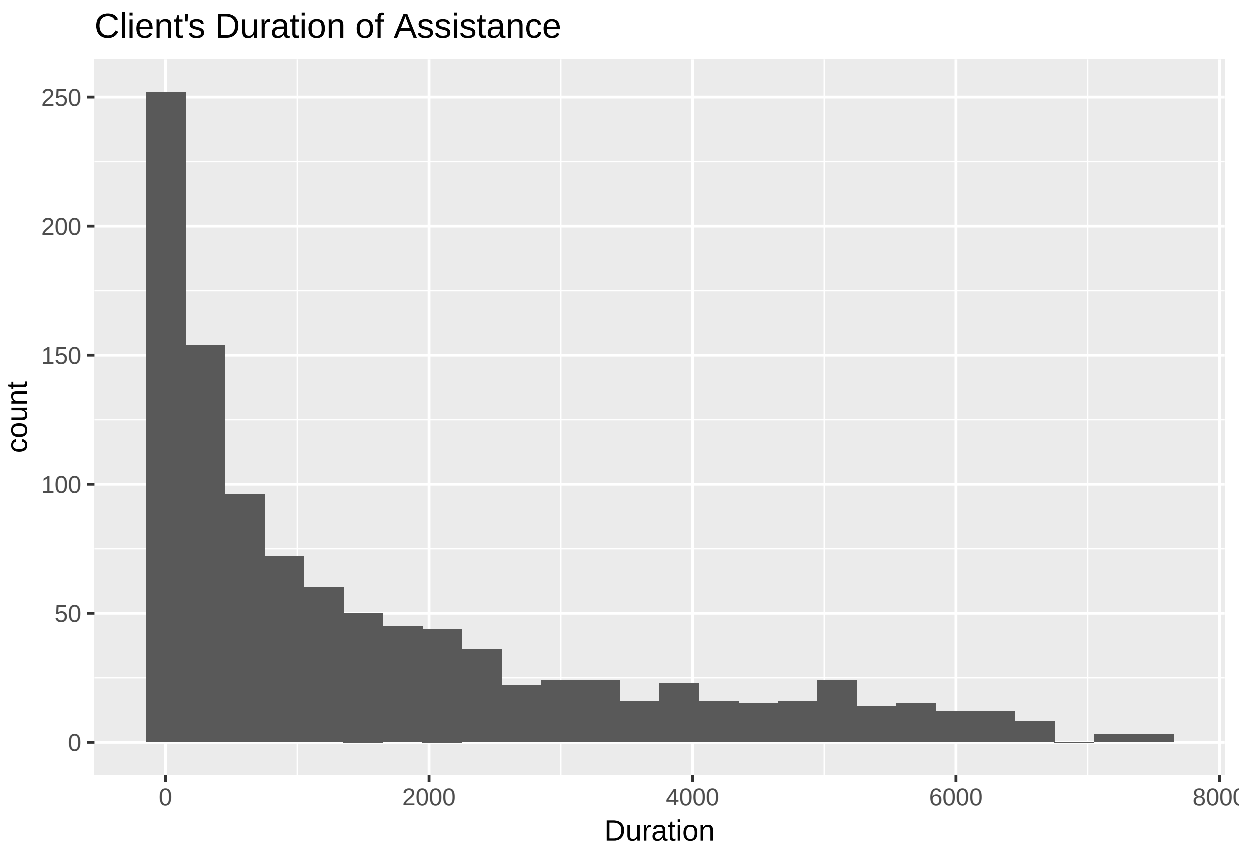


The number of clients served by UMD each month do differ. Generally, March is the month when the UMD serves the least amount of clients, and July is the month when the UMD serves the most amount of clients between 1994 and 2019.

5. Client's duration of assistance

```
client.data.duration <- data %>%
  group_by(ClientID) %>%
  summarize(Duration = difftime(max(Date),min(Date))) %>%
  filter(Duration > 0 & Duration < 10000)
ggplot(data = client.data.duration) +
  geom_histogram(mapping = aes(x = Duration, fill = Duration), binwidth = 300) +
  labs(title = "Client's Duration of Assistance")
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

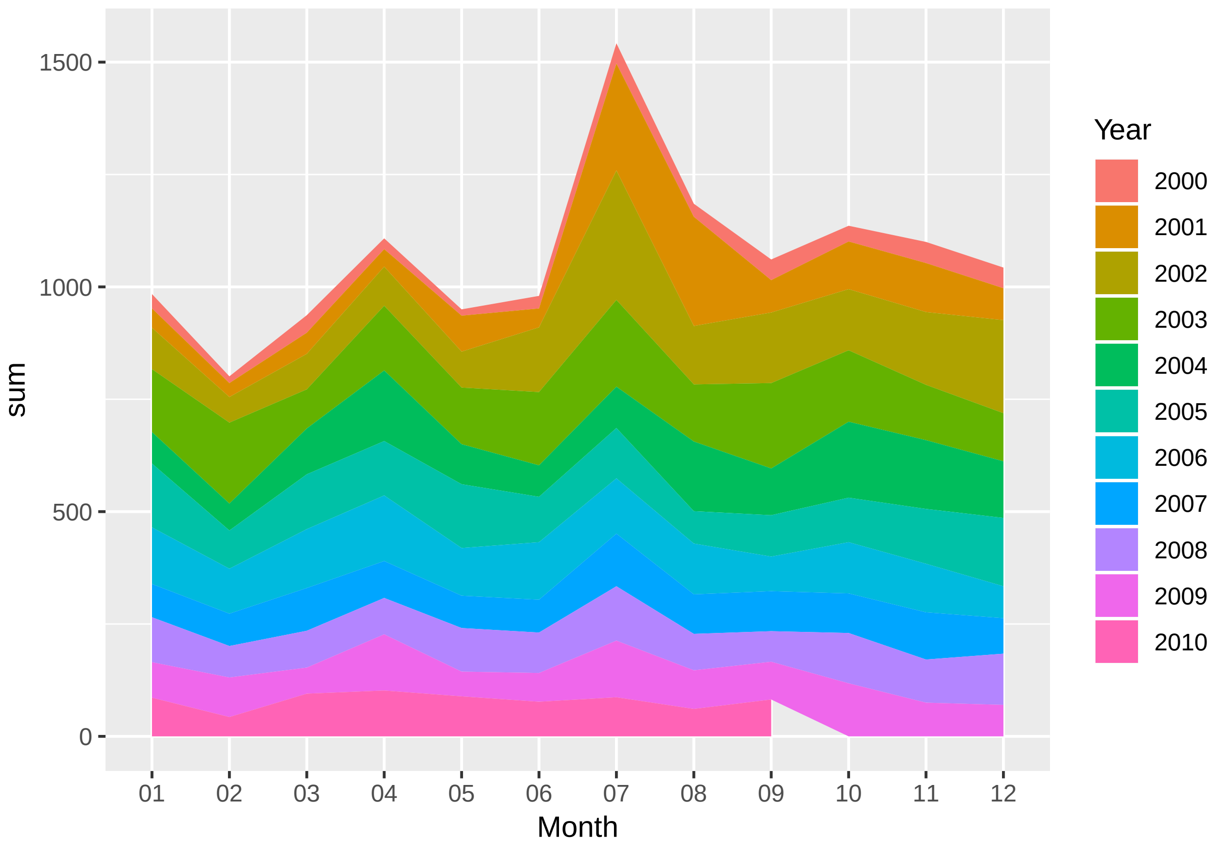


The histogram of the duration of assistance was skewed righ with a tail going off to the right. The most amount of clients was served by the UMD for 0-500 days and there are people who have been served over 10-20 years.

Seasonality

1. The amount of foods between 2000 and 2010

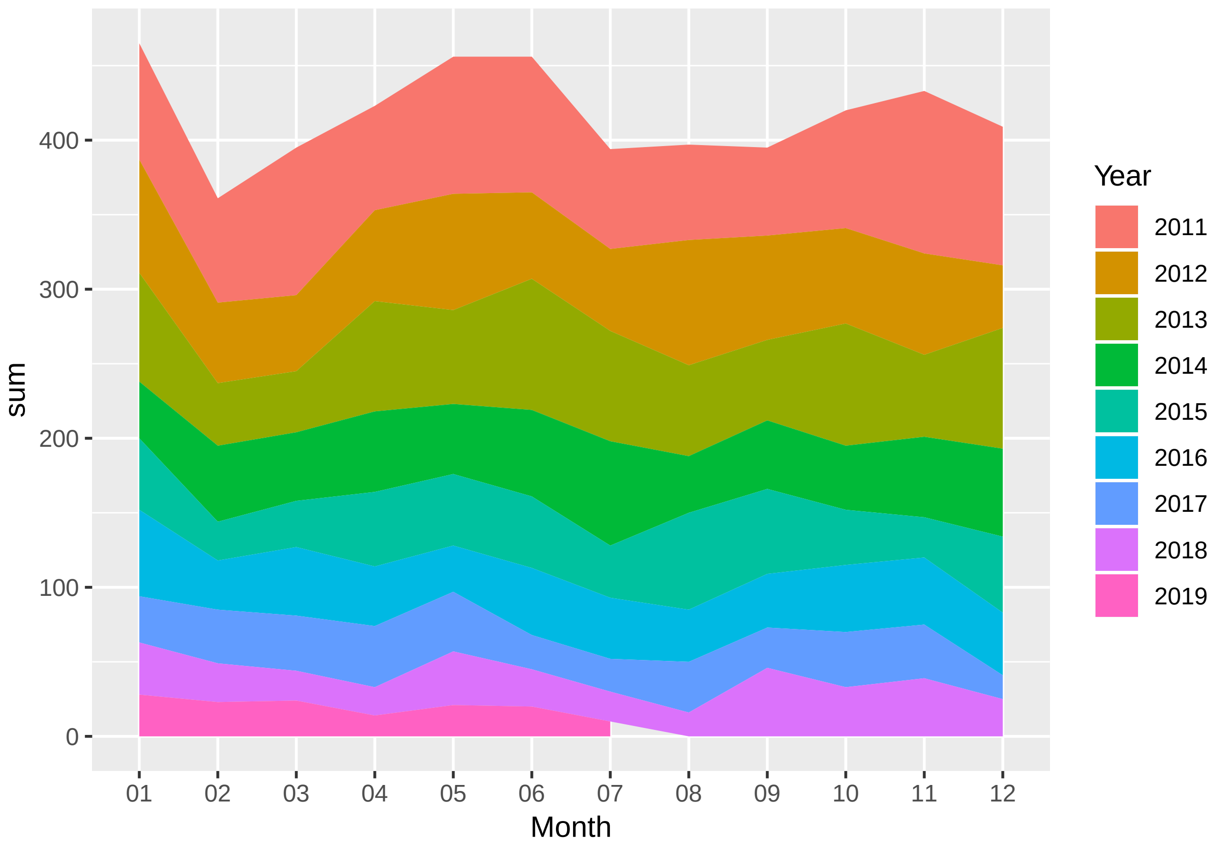
```
food.data = data %>%
  select(Date, Food) %>%
  filter(Date >= "2000-01-01" & Date <= "2010-10-01") %>%
  drop_na() %>%
  separate(Date, sep = "-", into = c("Year", "Month", "Day"))%>%
  group_by(Year, Month) %>%
  summarise(sum = sum(Food))
ggplot(food.data, aes(x = Month, y = sum, group = Year)) +
  geom_area(aes(fill = Year), position = "stack")
```



By seasonality, I mean periodic fluctuations. Between 2000 to 2010, sum of the amount of foods tended to peak for July and then decline after summer. So time series of the food amount typically showed increasing pattern from January through July and declining pattern from July to December.

2. The amount of foods between 2011 and 2019

```
food.data = data %>%
  select(Date, Food) %>%
  filter(Date >= "2011-01-01" & Date <= "2019-10-01") %>%
  drop_na() %>%
  separate(Date, sep = "-", into = c("Year", "Month", "Day"))%>%
  group_by(Year, Month) %>%
  summarise(sum = sum(Food))
ggplot(food.data, aes(x = Month, y = sum, group = Year)) +
  geom_area(aes(fill = Year), position = "stack")
```



On the other hand, between 2011 to 2019, Sum of the amount of foods tend to peak for May and June, then decline after summer and rise again during winter. So time series of the food amount typically show increasing pattern from February through May , declining pattern from July to september and increasing pattern again from September to January.

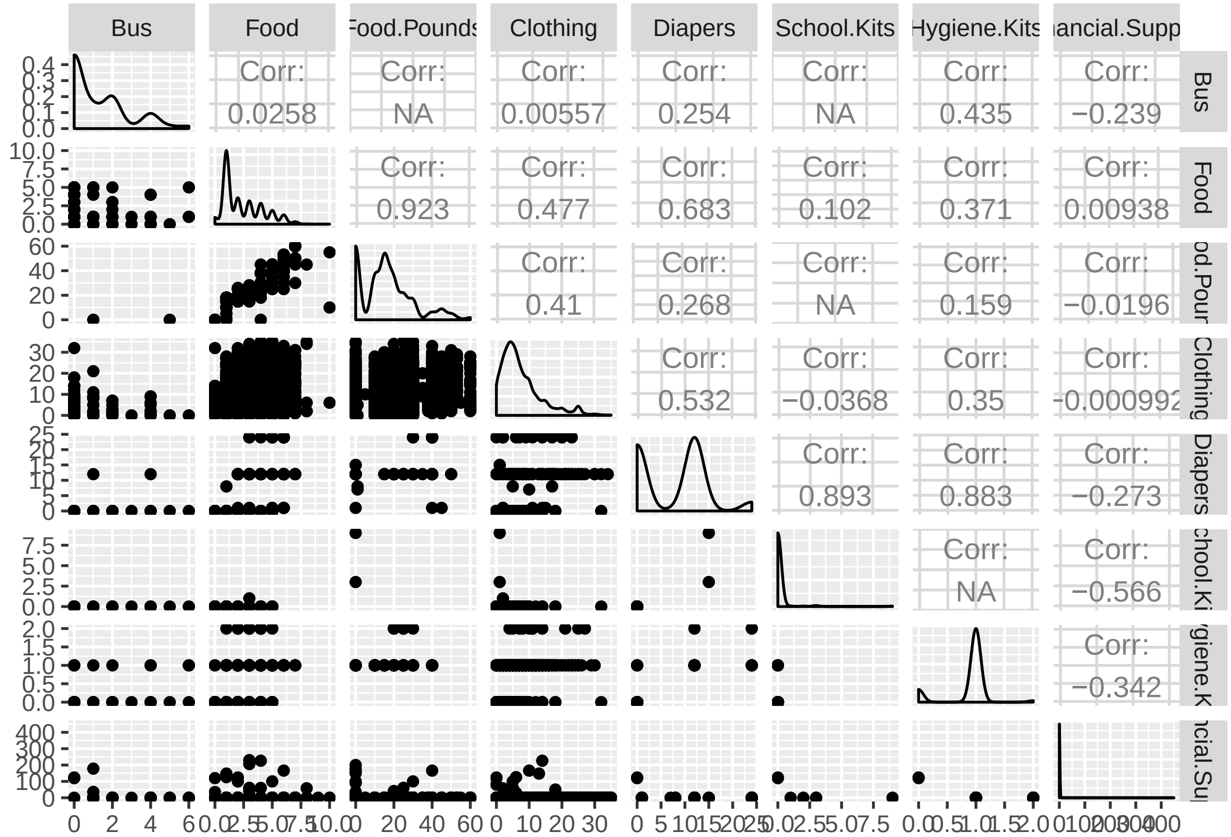
Correlation

```
library(GGally)
```

```
##
## Attaching package: 'GGally'

## The following object is masked from 'package:dplyr':
##
##     nasa
```

```
vars = c("Bus", "Food", "Food.Pounds", "Clothing", "Diapers", "School.Kits", "Hygiene.Kits", "Financial
data2 <- data[,c(vars)]
ggpairs(data2, progress = FALSE)
```



There isstatistically significant relationship between 1) Clothing and Food, 2) Diapers and Food, 3) Hygiene.Kits and Bus, 4) Hygiene.Kits and Food, 5) Hygiene.Kits and Diapers, and 6) Clothing and Food.pounds.

Conclusion

After importing the data, I renamed the variables and remove some rows with dates before 1983 or after 2019, given that the UMD was estabblished in 1983.

Before analyzing the data, I detected, visualized and tested for outliers. When I had a look at columns of the UMD dataset with boxplot, I got a number of outliers in each variable. Particularly, through the rosnertest, I could get rows in which the outliers are and the actual values of the outliers. Accordingly, the rows containing the outliers were removed and I could notice that those pesky outliers are gone.

1. To begin with, through the graphs of the number of (new) clients served by UMD between 1994 and 2019, I could demonstrate that the number of new clients served by UMD had been increasing until 2002 but has been decreasing up to now.
2. Second, the number of clients served by UMD each month do differ. Generally, March is the month when the UMD serves the least amount of clients, and July is the month when the UMD serves the most amount of clients between 1994 and 2019.
3. Third, the histogram of the duration of assistance was skewed righ with a tail going off to the right. The most amount of clients was served by the UMD for 0-500 days and there are people who have been served over 10-20 years.
4. Fourth, between 2000 to 2010, sum of the amount of foods tended to peak for July and then decline after summer. So time series of the food amount typically showed increasing pattern from January through July and declining pattern from July to December.

5. On the other hand, between 2011 to 2019, Sum of the amount of foods tend to peak for May and June, then decline after summer and rise again during winter. So time series of the food amount typically show increasing pattern from February through May , declining pattern from July to september and increasing pattern again from September to January.

6) Lastly, there isstatistically significant relationship between 1) Clothing and Food, 2) Diapers and Food, 3) Hygiene.Kits and Bus, 4) Hygiene.Kits and Food, 5) Hygiene.Kits and Diapers, and 6) Clothing and Food.pounds.

In the presentation, the social workers said they are particularly interested in "which clients they can help make a permanent transition to self-sufficiency, and which clients continue to need help." In this context, I suggest they perform analyses to find some hidden patterns and differences between different types of clients with fine-grained dataset (e.g., after shrinking the size of the variables through PCA, we can perform a unsupervised cluster analysis to find various types of clients and compare them in terms of mean, variance, kurtosis, and skewness (e.g., parametric methods) and general shape of distribution (e.g., nonparametric methods), based off of the principle components).