# Tidyverse with Groceries Data

DH Kim

This document shows some data work with the `readr`, `dplyr`, `stringr`, and `ggplot2` libraries in tidyverse, focusing on how to use functions. The dataset used comes from Kaggle Groceries dataset.

```
# libraries in Tidyverse
library(readr)
library(dplyr)
library(stringr)
library(ggplot2)
```

### Importing data with readr::read_*()

It shows how to use the `read_csv()` function in the readr library and its, which is different from the built-in `read.csv()` function. The R documentation for read_csv() is here and the one for read.csv() is here.

```
groceries <-
  read_csv(url("https://raw.githubusercontent.com/HwanKim2/data_repo/main/Groceries_dataset.csv"))
```

```
## Parsed with column specification:
## cols(
##   Member_number = col_double(),
##   Date = col_character(),
##   itemDescription = col_character()
## )
```

```
glimpse(groceries)
```

```
## Rows: 38,765
## Columns: 3
## $ Member_number   <dbl> 1808, 2552, 2300, 1187, 3037, 4941, 4501, 3803, 276...
## $ Date            <chr> "21-07-2015", "05-01-2015", "19-09-2015", "12-12-20...
## $ itemDescription <chr> "tropical fruit", "whole milk", "pip fruit", "other...
```

The resulting data frame is *groceries*. Some variants are as follows. The first one explicitly writes down the default options in the function.

```
gro_varOne <- readr::read_csv(
  url("https://raw.githubusercontent.com/HwanKim2/data_repo/main/Groceries_dataset.csv"),
  col_names = TRUE, col_types = NULL)
```

```
## Parsed with column specification:
## cols(
##   Member_number = col_double(),
##   Date = col_character(),
##   itemDescription = col_character()
## )
```

```
glimpse(gro_varOne)
```

```
## Rows: 38,765
## Columns: 3
## $ Member_number   <dbl> 1808, 2552, 2300, 1187, 3037, 4941, 4501, 3803, 276...
## $ Date            <chr> "21-07-2015", "05-01-2015", "19-09-2015", "12-12-20...
## $ itemDescription <chr> "tropical fruit", "whole milk", "pip fruit", "other...
```

```
identical(groceries, gro_varOne)
```

```
## [1] TRUE
```

It shows how to specify the `col_types` option.

```
gro_wayTwo <-
  read_csv(
    url("https://raw.githubusercontent.com/HwanKim2/data_repo/main/Groceries_dataset.csv"),
       col_types = cols(
                   Member_number = col_double(),
                   Date = col_character(),
                   itemDescription = col_character()
                   )
          )
identical(groceries, gro_wayTwo)
```

```
## [1] TRUE
```

### Counting observations by group with count()

```
item_count <- groceries %>%
  dplyr::count(itemDescription) %>%
  arrange(desc(n))
item_count[1:10,]
```

```
## # A tibble: 10 x 2
##    itemDescription      n
##    <chr>            <int>
##  1 whole milk        2502
##  2 other vegetables  1898
##  3 rolls/buns        1716
##  4 soda              1514
##  5 yogurt            1334
##  6 root vegetables   1071
##  7 tropical fruit    1032
##  8 bottled water      933
##  9 sausage            924
## 10 citrus fruit       812
```

The above data work is simplified with the sort option.

```
item_count_varOne <- groceries %>%
  dplyr::count(itemDescription, sort = TRUE)
item_count_varOne[1:10,]
```

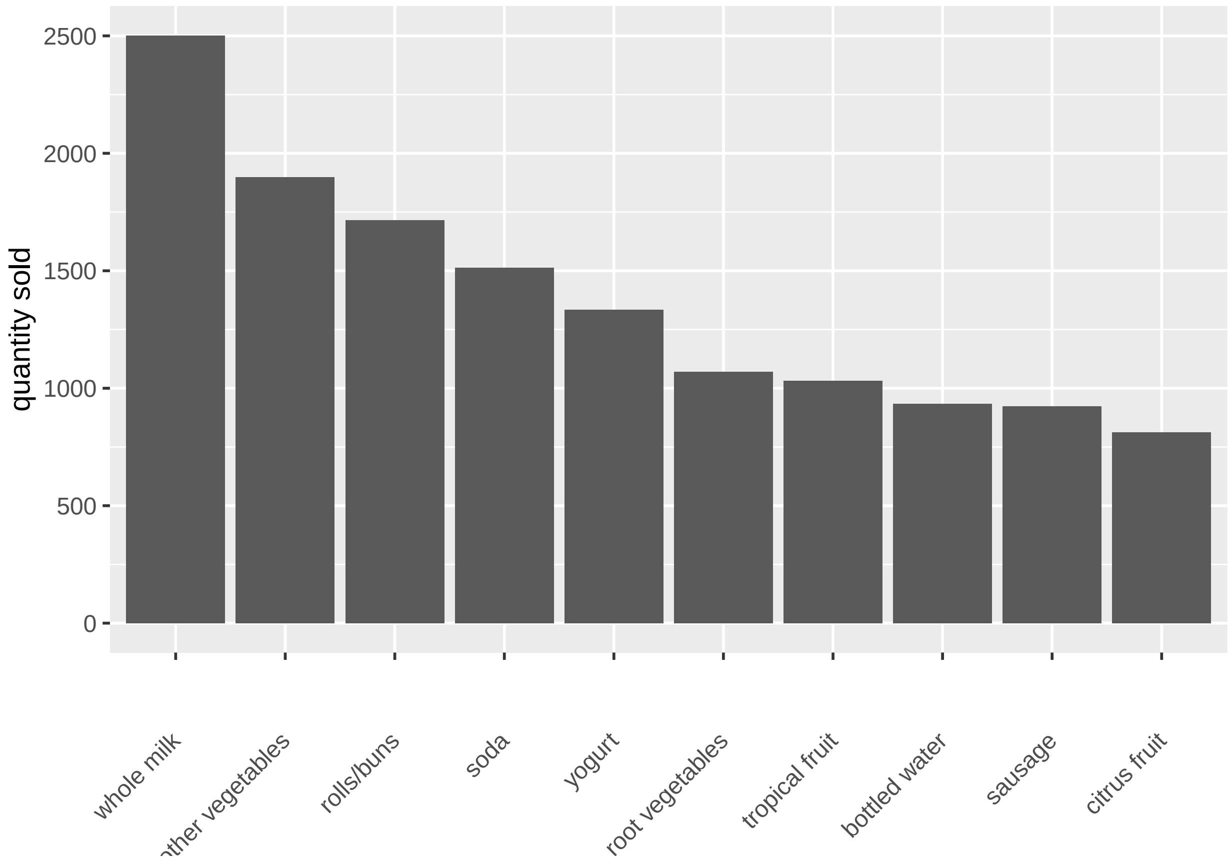
```
## # A tibble: 10 x 2
##    itemDescription      n
##    <chr>            <int>
##  1 whole milk        2502
##  2 other vegetables  1898
##   3 rolls/buns        1716
##   4 soda              1514
##   5 yogurt            1334
##   6 root vegetables   1071
##   7 tropical fruit    1032
##   8 bottled water      933
##   9 sausage            924
## 10 citrus fruit        812
```

```
identical(item_count, item_count_varOne)
```

```
## [1] TRUE
```

### Plot with ggplot()

```
ggplot(item_count[1:10,],
       aes(x=reorder(itemDescription, -n), y = n)) +
geom_bar(stat="identity") +
labs(x = "", y = "quantity sold") +
theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust=1))
```


quantity sold
whole milk
other vegetables
rolls/buns
soda
yogurt
root vegetables
tropical fruit
bottled water
sausage
citrus fruit