Instacart Market Basket Analysis dataset obtained from https://www.kaggle.com/c/instacart-market-basket-analysis/data (https://www.kaggle.com/c/instacart-market-basket-analysis/data)

```
setwd("C:/Users/qt09n/Desktop/market")

library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.0.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.3
```

```
##
## 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
```

```
aisles <- read.csv("aisles.csv")
department <- read.csv("departments.csv")
orders <- read.csv("orders.csv")
order_products_prior <- read.csv("order_products__prior.csv")
order_products_train <- read.csv("order_products__train.csv")
products <- read.csv("products.csv")
sample_submission <- read.csv("sample_submission.csv")
```

datasets have different number of rows, may be hard to combine

select which dataset to include??

```
str(order_products_prior)
```

```
## 'data.frame':    32434489 obs. of  4 variables:
##  $ order_id         : int  2 2 2 2 2 2 2 2 2 3 ...
##  $ product_id       : int  33120 28985 9327 45918 30035 17794 40141 1819 43668 33754 ...
##  $ add_to_cart_order: int  1 2 3 4 5 6 7 8 9 1 ...
##  $ reordered        : int  1 1 0 1 0 1 1 1 0 1 ...
```

32434489 obs. of 4 variable:

```
str(orders)
```

```
## 'data.frame':    3421083 obs. of  7 variables:
##  $ order_id              : int  2539329 2398795 473747 2254736 431534 3367565 550135 3108588 2295261
##  $ user_id               : int  1 1 1 1 1 1 1 1 1 ...
##  $ eval_set              : chr  "prior" "prior" "prior" "prior" ...
##  $ order_number          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ order_dow             : int  2 3 3 4 4 2 1 1 1 4 ...
##  $ order_hour_of_day     : int  8 7 12 7 15 7 9 14 16 8 ...
##  $ days_since_prior_order: num  NA 15 21 29 28 19 20 14 0 30 ...
```

```
str(products)
```

```
## 'data.frame':    49688 obs. of  4 variables:
##  $ product_id   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ product_name : chr  "Chocolate Sandwich Cookies" "All-Seasons Salt" "Robust Golden Unsweetened Oo
##  $ aisle_id     : int  61 104 94 38 5 11 98 116 120 115 ...
##  $ department_id: int  19 13 7 1 13 11 7 1 16 7 ...
```

```
str(order_products_train)
```

```
## 'data.frame':    1384617 obs. of  4 variables:
##  $ order_id         : int  1 1 1 1 1 1 1 1 36 36 ...
##  $ product_id       : int  49302 11109 10246 49683 43633 13176 47209 22035 39612 19660 ...
##  $ add_to_cart_order: int  1 2 3 4 5 6 7 8 1 2 ...
##  $ reordered        : int  1 1 0 0 1 0 0 1 0 1 ...
```

```
str(order_products_prior)
```

```
## 'data.frame':    32434489 obs. of  4 variables:
##  $ order_id         : int  2 2 2 2 2 2 2 2 2 3 ...
##  $ product_id       : int  33120 28985 9327 45918 30035 17794 40141 1819 43668 33754 ...
##  $ add_to_cart_order: int  1 2 3 4 5 6 7 8 9 1 ...
##  $ reordered        : int  1 1 0 1 0 1 1 1 0 1 ...
```

```
str(aisles)
```

```
## 'data.frame':    134 obs. of  2 variables:
##  $ aisle_id: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ aisle   : chr  "prepared soups salads" "specialty cheeses" "energy granola bars" "instant foods"
```

```
str(department)
```

```
## 'data.frame':    21 obs. of  2 variables:
##  $ department_id: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ department   : chr  "frozen" "other" "bakery" "produce" ...
```

```
attributes <-c("aisle_id", "aisle", "department_id", "department", "order_id", "product_id","add_to_car
```

```
length(attributes)
```

```
## [1] 15
```

```r
length(unique(aisles$aisle_id))
```

```
## [1] 134
```

```r
length(unique(aisles$aisle))
```

```
## [1] 134
```

```r
#merge aisles and products data frame by aisle_id
combined <- merge(products, aisles, by="aisle_id")

#join department with combined(aisles + products dataset) by common department_id column
combined2 <- merge(combined, department, by="department_id")
```

#Info about Order_products___*.csv from Kaggle website:

order_products___*.csv

These files specify which products were purchased in each order. order_products___prior.csv contains previous order contents for all customers. 'reordered' indicates that the customer has a previous order that contains the product. Note that some orders will have no reordered items. You may predict an explicit 'None' value for orders with no reordered items. See the evaluation page for full details.

orders.csv

This file tells to which set (prior, train, test) an order belongs. You are predicting reordered items only for the test set orders. 'order_dow' is the day of week.

```r
glimpse(orders)
```

```
## Rows: 3,421,083
## Columns: 7
## $ order_id               <int> 2539329, 2398795, 473747, 2254736, 431534, 3...
## $ user_id                <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2,...
## $ eval_set               <chr> "prior", "prior", "prior", "prior", "prior",...
## $ order_number           <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, ...
## $ order_dow              <int> 2, 3, 3, 4, 4, 2, 1, 1, 1, 4, 4, 2, 5, 1, 2,...
## $ order_hour_of_day      <int> 8, 7, 12, 7, 15, 7, 9, 14, 16, 8, 8, 11, 10,...
## $ days_since_prior_order <dbl> NA, 15, 21, 29, 28, 19, 20, 14, 0, 30, 14, N...
```

```r
glimpse(order_products_prior)
```

```
## Rows: 32,434,489
## Columns: 4
## $ order_id          <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3...
## $ product_id        <int> 33120, 28985, 9327, 45918, 30035, 17794, 40141, 1...
## $ add_to_cart_order <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8...
## $ reordered         <int> 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1...
```

```r
glimpse(order_products_train)
```

```
## Rows: 1,384,617
## Columns: 4
## $ order_id          <int> 1, 1, 1, 1, 1, 1, 1, 1, 36, 36, 36, 36, 36, 36, 3...
## $ product_id        <int> 49302, 11109, 10246, 49683, 43633, 13176, 47209, ...
## $ add_to_cart_order <int> 1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8, 1...
## $ reordered         <int> 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0...
```

```
unique(orders$eval_set)
```

```
## [1] "prior" "train" "test"
```

we are predicting reordered items only for the test set orders. 'order_dow' is the day of week.

To make a separate dataframe for test data:

```
test <- orders[orders$eval_set=="test",]
```

To make a separate dataframe just for the train and prior rows from the eval_set column of orders:

```
orders_train_prior <- dplyr::filter(orders, eval_set %in% c("prior", "train"))
```