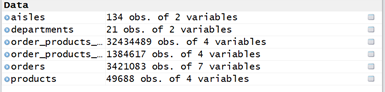
**Structures of the datasets:**



**Libraries used:**

* ouliers
* readr
* ggplot2
* dplyr
* scales

# Display structure of datasets

str(order\_products\_prior)

str(order\_products\_train)

str(products)

str(orders)

str(departments)

str(aisles)

***Output:***

*str(order\_products\_prior)*

*Classes ‘tbl\_df’, ‘tbl’ and '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(order\_products\_train)*

*Classes ‘tbl\_df’, ‘tbl’ and '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*

*$ 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(products)*

*Classes ‘tbl\_df’, ‘tbl’ and '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 Oolong Tea" "Smart Ones Classic Favorites Mini*

*$ 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(orders)*

*Classes ‘tbl\_df’, ‘tbl’ and 'data.frame': 3421083 obs. of 7 variables:*

*$ order\_id : int 2539329 2398795 473747 2254736 431534*

*$ user\_id : int 1 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 : chr "08" "07" "12" "07" ...*

*$ days\_since\_prior\_order: num NA 15 21 29 28 19 20 14 0 30 ...*

*> str(departments)*

*Classes ‘tbl\_df’, ‘tbl’ and '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" ...*

*> str(aisles)*

*Classes ‘tbl\_df’, ‘tbl’ and '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*

# Data treatment

# Missing value analysis

apply(orders,2, function(x)sum(is.na(x)))

***Output:***

*order\_id user\_id eval\_set order\_number*

*0 0 0 0*

*order\_dow order\_hour\_of\_day days\_since\_prior\_order*

*0 0 206209*

# orders$days\_since\_prior\_order has 206209 missing values

# Outlier Analysis

library(outliers)

outlier(orders$days\_since\_prior\_order, opposite=FALSE)

#Grubbs test for outliers

grubbs.test(orders$days\_since\_prior\_order, type = 10)

# outliers: day 30 - No extreme outliers found

***Output:***

*Grubbs test for one outlier*

*data: orders$days\_since\_prior\_order*

*G = 2.0512, U = 1.0000, p-value = 1*

*alternative hypothesis: highest value 30 is an outlier*

*# Imputing missing values not required since missing values mean that some users did not reorder some products*

#---------------------------------------------------------------------------------------------------------------------

# Predictive data analysis

# Visualisation

library(ggplot2)

library(dplyr)

library(scales)

# Hour of day

ggplot(data=orders,aes(x=order\_hour\_of\_day)) +

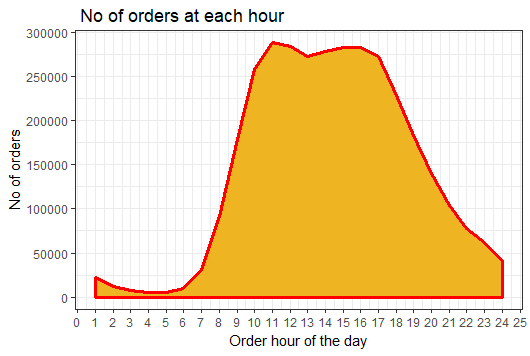
geom\_area(stat="count",fill="goldenrod2",colour = "red",size=1.2)+

ggtitle(' No of orders at each hour')+ theme\_bw()+

xlab('Order hour of the day') + ylab('No of orders') +

scale\_x\_continuous(breaks=pretty\_breaks(n=24))+

scale\_y\_continuous(breaks=pretty\_breaks(n=7))



# Day of Week

ggplot(data=orders,aes(x=order\_dow))+

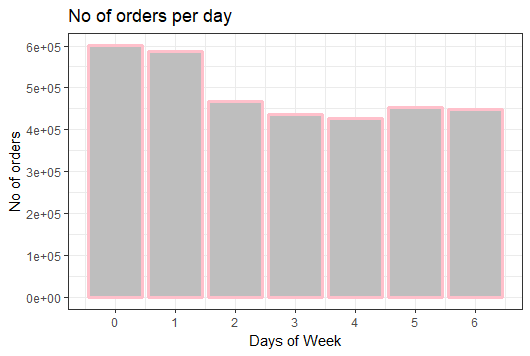
geom\_histogram(stat="count",fill="grey",size=1.2, colour = "pink") +

ggtitle('No of orders per day')+ theme\_bw()+

xlab('Days of Week') + ylab('No of orders')+

scale\_x\_continuous(breaks=pretty\_breaks(n=7))+

scale\_y\_continuous(breaks=pretty\_breaks(n=7))



# Days taken to reorder

ggplot(data=orders,aes(x=days\_since\_prior\_order))+

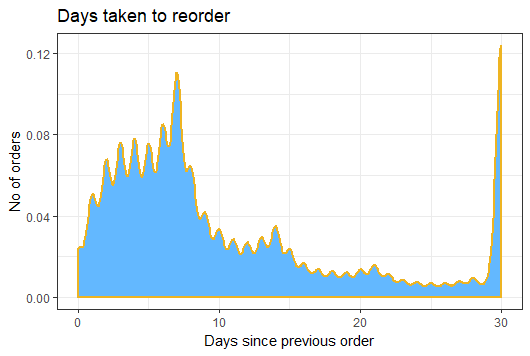
geom\_density(stat="density",fill="steelblue1",colour = "goldenrod2",size=1.05) +

ggtitle('Days taken to reorder') + theme\_bw() +

xlab('Days since previous order') + ylab('No of orders')

scale\_x\_continuous(breaks=pretty\_breaks(n=15))+

scale\_y\_continuous(breaks=pretty\_breaks(n=7))



# Retrieve test data from orders

test = orders[orders$eval\_set=="test",]

***Output***

*head(test)*

*# A tibble: 6 x 7*

*order\_id user\_id eval\_set order\_number order\_dow order\_hour\_of\_day days\_since\_prior\_order*

*<int> <int> <chr> <int> <int> <chr> <dbl>*

*1 2774568 3 test 13 5 15 11*

*2 329954 4 test 6 3 12 30*

*3 1528013 6 test 4 3 16 22*

*4 1376945 11 test 8 6 11 8*

*5 1356845 12 test 6 1 20 30*

*6 2161313 15 test 23 1 09 7*

# Retrieve prior users with prior and test by their user\_id

# Create temporary dataframes by left join with products

products\_aisles = left\_join(products, aisles, by = "aisle\_id")

products\_dept = left\_join(products, departments, by = "department\_id")

products\_aisles\_dept = left\_join(products\_aisles,products\_dept, by = "product\_id",copy=FALSE)

# Remove redundant columns

products\_aisles\_dept = products\_aisles\_dept[,c(1,2,3,4,9,5)]

# Rename the columns

colnames(products\_aisles\_dept) = c("product\_id","product\_name","aisle\_id", "department\_id","department","aisle")

# Remove the temporary dataframes

rm(products\_aisles, products\_dept)

# Rearrange and group by order\_id

orders\_train = left\_join(order\_products\_train, orders, by = "order\_id")

orders\_prior = left\_join(order\_products\_prior, orders, by = "order\_id")

# Rearrange and group by product\_id, remove redundant columns

orders\_train = left\_join(orders\_train, products\_aisles\_dept,by = 'product\_id')

orders\_train = orders\_train[,c(-11, -14, -15)]

orders\_prior = left\_join(orders\_prior, products\_aisles\_dept,by = 'product\_id')

orders\_prior = orders\_prior[,c(-11, -14, -15)]

#row bind train and prior data with all order columns

orders\_train\_prior = rbind(orders\_train,orders\_prior)

head(orders\_train\_prior,10)

#Pick the users for train data

test\_users = orders\_train\_prior[orders\_train\_prior$user\_id %in% test$user\_id,]

# Merging products with order\_products\_prior

order\_products\_prior\_pid = left\_join(order\_products\_prior, products\_aisles\_dept, by = "product\_id",copy=FALSE)

# Top 15 products

# Count and sort the products by product\_id

Top15products <- order\_products\_prior\_pid %>%

group\_by(product\_id) %>%

summarize(count = n()) %>%

top\_n(15, wt = count) %>%

left\_join(select(products,product\_id,product\_name),by="product\_id") %>%

arrange(desc(count))

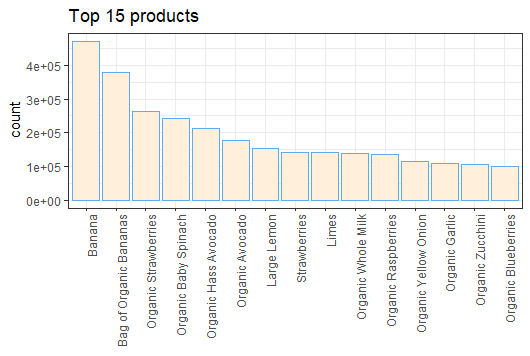
Top15products

Top15products %>%

ggplot(aes(x=reorder(product\_name,-count), y=count))+ theme\_bw()+

geom\_bar(stat="identity",fill="antiquewhite1",colour = "steelblue2")+

theme(axis.text.x=element\_text(angle=90, hjust=1),axis.title.x = element\_blank())+ggtitle("Top 15 products")



# Top 5 departments

# Count and sort the department by department names

Top5departs <- order\_products\_prior\_pid %>%

group\_by(department) %>%

summarize(count = n()) %>%

top\_n(5, wt = count) %>%

left\_join(select(departments,department\_id,department),by="department") %>%

arrange(desc(count))

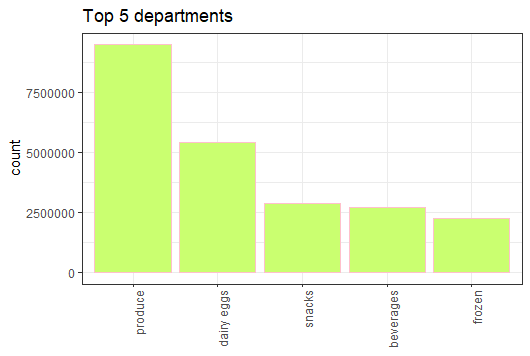
Top5departs

Top5departs %>%

ggplot(aes(x=reorder(department,-count), y=count))+ theme\_bw()+

geom\_bar(stat="identity",fill="darkolivegreen1",colour = "pink")+

theme(axis.text.x=element\_text(angle=90, hjust=1),axis.title.x = element\_blank())+ggtitle("Top 5 departments")



# Top 5 aisles

# Count and sort the aisles by aisle names

Top5aisles <- order\_products\_prior\_pid %>%

group\_by(aisle) %>%

summarize(count = n()) %>%

top\_n(5, wt = count) %>%

left\_join(select(aisles,aisle\_id,aisle),by="aisle") %>%

arrange(desc(count))

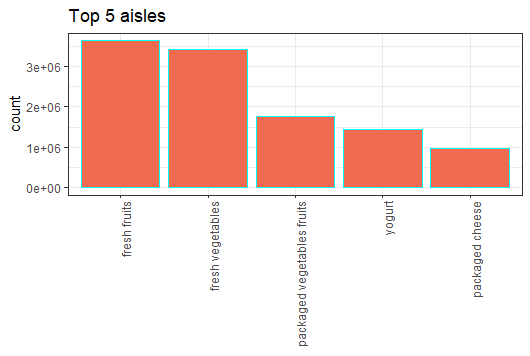
Top5aisles

Top5aisles %>%

ggplot(aes(x=reorder(aisle,-count), y=count))+ theme\_bw()+

geom\_bar(stat="identity",fill="coral2",colour = "cyan")+

theme(axis.text.x=element\_text(angle=90, hjust=1),axis.title.x = element\_blank())+ggtitle("Top 5 aisles")



#--------------------------------------------------------------------------------------------------------

# Recommendation

# calcuating maximum order number

order\_items\_number <- order\_products\_prior %>%

group\_by(order\_id) %>%

summarise(number\_of\_items = n())

user\_max\_order\_number <- orders %>%

group\_by(user\_id) %>%

summarise(max\_order\_number = max(order\_number))

order\_x <- orders %>%

inner\_join(order\_items\_number) %>%

inner\_join(user\_max\_order\_number)

# calcuating average number of items in basket

order\_latest\_4 <- order\_x %>%

filter(order\_number + 4 >= max\_order\_number)

user\_average\_items <- order\_latest\_4 %>%

group\_by(user\_id) %>%

summarise(avg\_n\_of\_items = mean(number\_of\_items))

order\_product\_latest\_4 <- order\_latest\_4 %>%

inner\_join(order\_products\_prior)

# Finding out favoured product that is reordered most

user\_favor\_product <- order\_product\_latest\_4 %>%

group\_by(user\_id, product\_id) %>%

summarise(favor = sum((reordered\*9+1)\*(order\_number+5-max\_order\_number)))

user\_product\_rank <- user\_favor\_product %>%

arrange(user\_id,favor) %>%

group\_by(user\_id) %>%

mutate(rank = rank(desc(favor)))

user\_next\_basket <- user\_product\_rank %>%

inner\_join(user\_average\_items) %>%

filter (rank <= avg\_n\_of\_items)

# Retrive users who reordered the favoured products

# And retrive users who would reorder the favoured products

sample\_submission\_user <- sample\_submission %>%

inner\_join(orders) %>%

select(user\_id,order\_id)

submission\_user\_next\_basket <- user\_next\_basket %>%

inner\_join(sample\_submission\_user) %>%

select(user\_id, order\_id, product\_id)

submission\_order\_buy <- submission\_user\_next\_basket %>%

group\_by(order\_id) %>%

summarise(products = paste(product\_id, collapse=' '))

submission\_order <- sample\_submission\_user %>%

left\_join(submission\_order\_buy) %>%

select(order\_id,products)

# Store results in the submission file

write.table(submission\_order, file="./submission.csv", sep=",",

row.names = FALSE, quote = FALSE, na="NA")

submission = read\_csv("C:/MyFiles/Kshiti/Edwisor/PROJECT/submission.csv")

View(submission)

#--------------------------------------------------------------------------------------------