SPROCKET CENTRAL PTY LTD

CUSTOMER ANALYIS

An analysis of Sprocket Central Pty Ltd customer data in order to understand new customers to target.

We will try to understand the different classifications of customers based on the recorded behaviours.

Then we will do customer segmentation.

```
library(tidyverse)
library(lubridate)
library(scales) ## for scales
library(VIM)  ## aggregate plotting of missing values
library(utf8)
library(corrplot) ## variables relationships
library(factoextra) ## k selection visualization
library(ggrepel) ## visualization
library(ggfortify)
```

```
transactions <- read_csv("transactions_data.csv")
newcustomerlist <- read_csv("NewCustomerList.csv")
```

```r
demographic <- read_csv("demographic_data.csv")
address <- read_csv("address_data.csv")
```

load the data and clean the names, dplyr::select useful columns

1 Transactions data

structure and dimensions of the data

```r
str(transactions)
```

```
## spc_tbl_ [20,000 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ transaction_id        : num [1:20000] 1 2 3 4 5 6 7 8 9 10 ...
##  $ product_id            : num [1:20000] 2 3 37 88 78 25 22 15 67 12 ...
##  $ customer_id           : num [1:20000] 2950 3120 402 3135 787 ...
##  $ transaction_date      : Date[1:20000], format: "2017-02-25" "2017-05-21" ...
##  $ online_order          : num [1:20000] 0 1 0 0 1 1 1 0 0 1 ...
##  $ order_status          : chr [1:20000] "Approved" "Approved" "Approved" "Approved" ...
##  $ brand                 : chr [1:20000] "Solex" "Trek Bicycles" "OHM Cycles" "Norco Bicycl
##  $ product_line          : chr [1:20000] "Standard" "Standard" "Standard" "Standard" ...
##  $ product_class         : chr [1:20000] "medium" "medium" "low" "medium" ...
##  $ product_size          : chr [1:20000] "medium" "large" "medium" "medium" ...
##  $ list_price            : num [1:20000] 71.5 2091.5 1793.4 1198.5 1765.3 ...
##  $ standard_cost         : num [1:20000] 53.6 388.9 248.8 381.1 709.5 ...
##  $ product_first_sold_date: Date[1:20000], format: "2012-12-04" "2014-03-05" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   transaction_id = col_double(),
##   ..   product_id = col_double(),
##   ..   customer_id = col_double(),
##   ..   transaction_date = col_date(format = ""),
##   ..   online_order = col_double(),
##   ..   order_status = col_character(),
##   ..   brand = col_character(),
##   ..   product_line = col_character(),
##   ..   product_class = col_character(),
##   ..   product_size = col_character(),
##   ..   list_price = col_double(),
##   ..   standard_cost = col_double(),
##   ..   product_first_sold_date = col_date(format = "")
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```r
dim(transactions) # data rows and columns
```

```
## [1] 20000    13
```

We have 20,000 recorded transactions.

Change some column names

```r
transactions <- transactions %>% rename(tran_id = transaction_id,
                                        tran_date = transaction_date,
                                        first_sold_date =
                                          product_first_sold_date)
names(transactions)
```

```
##  [1] "tran_id"          "product_id"       "customer_id"      "tran_date"
##  [5] "online_order"     "order_status"     "brand"            "product_line"
##  [9] "product_class"    "product_size"     "list_price"       "standard_cost"
## [13] "first_sold_date"
```

Each transaction should be unique, therefore the transaction id should be unique for all transactions

```r
n_distinct(transactions$tran_id)
```

```
## [1] 20000
```

All the 20,000 transactions are unique that is we don't have duplicate transactions.

We had order_status that gives information whether an order was cancelled or not

```r
transactions %>% count(order_status, sort = T)
```

```
## # A tibble: 2 x 2
##   order_status     n
##   <chr>        <int>
## 1 Approved     19821
## 2 Cancelled      179
```

We had 179 cancelled orders

Remove the cancelled orders

```
transactions_1 <- transactions %>% filter(order_status !="Cancelled")
```

We had 19,821 approved transactions

Product id

```
n_distinct(transactions_1$product_id) ### unique product_id
```

```
## [1] 101
```

```
class(transactions_1$product_id) ### How was it loaded
```

```
## [1] "numeric"
```

```
range(transactions_1$product_id) ### the representation i.e the coding
```

```
## [1]   0 100
```

```
setdiff(0:100, transactions_1$product_id) ### was any whole integer skipped
```

```
## integer(0)
```

We have 101 distinct products. the variable was loaded as numeric ranging from 0 to 100 with all the whole integers from 0 to 100

transactions date

```
range(transactions_1$tran_date)
```

```
## [1] "2017-01-01" "2017-12-30"
```

All the transactions were recorded in the year 2017 between January and December.

missing values

```r
sum(is.na(transactions_1))
```

```
## [1] 1530
```

Our data has missing values.

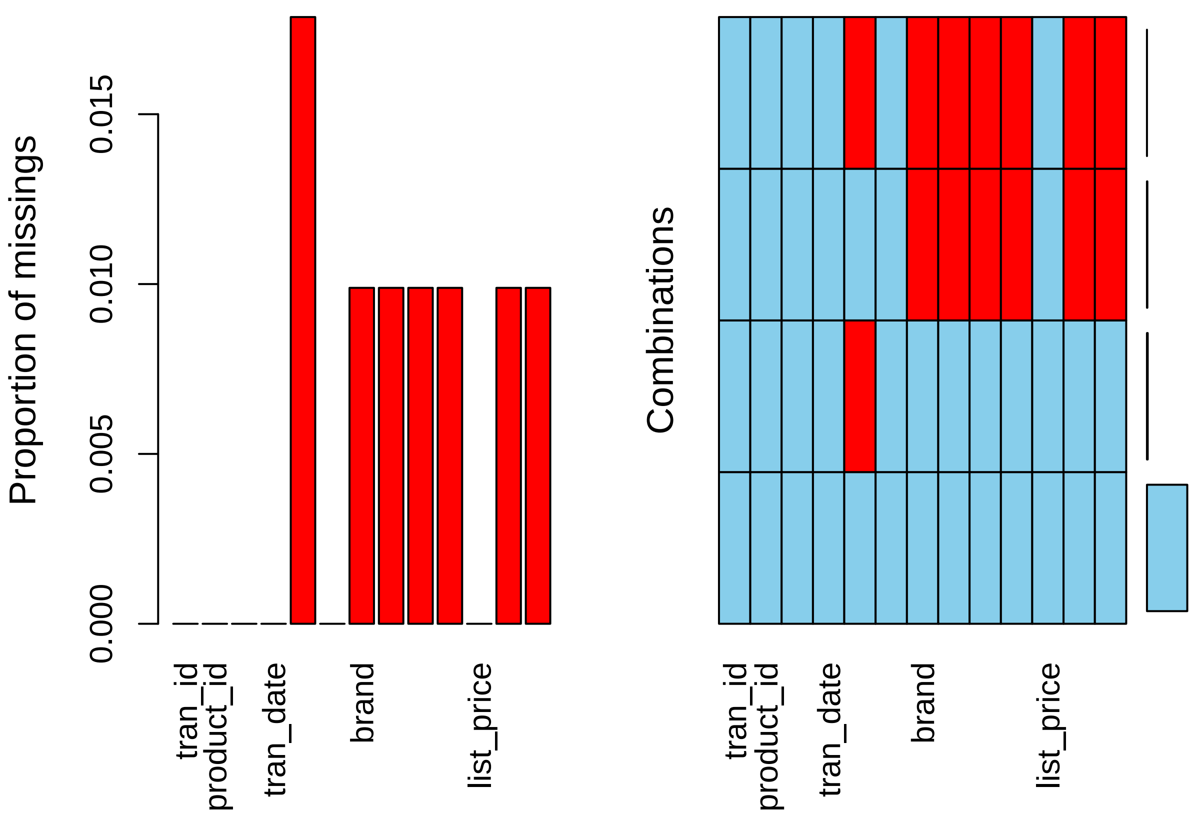
Columns with missing values

```r
names(which(colSums(is.na(transactions_1)) > 0))
```

```
## [1] "online_order"     "brand"            "product_line"     "product_class"
## [5] "product_size"     "standard_cost"    "first_sold_date"
```

combinations of variables with missing values

```r
aggr(transactions_1)
```



clearly brand, product_line, product_class, product_size, standard_cost and first_sold date have the same combination of missing values.

Understanding the missing values in the above variables

If we separate the data with the 196 same missing values with the customers that have values for the 6 variables, do we get that their is a customer with the missing values but there were instances that he had recorded values before?

```
tran_miss_1 <- transactions_1 %>% filter(is.na(brand) > 0)
tran_no_miss_1 <- transactions_1 %>% filter(!is.na(brand) > 0)
```

From the 196 records we have 191 distinct customers and 186 of them have visited only once.

```
n_distinct(tran_miss_1$customer_id)
```

```
## [1] 191
```

```
tran_miss_1_count <- tran_miss_1 %>% count(customer_id, sort = T)
single_purchase_tran_miss_1 <- tran_miss_1_count %>% filter( n < 2)
```

The 186 are they in other dataset that does not have the 196 with missing records

```
present_in_19625 <- tran_no_miss_1[tran_no_miss_1$customer_id %in% single_purchase_tran_miss_1$
dim(present_in_19625) ## those who visited once in 197 but visited in 19625
```

```
## [1] 1065   13
```

```
present_in_19625_count <- present_in_19625 %>% count(customer_id, sort = T)
single_purchase_present_in_19625 <- present_in_19625_count %>% filter( n < 2)
single_purchase_present_in_19625 ## number present in 19625 only once
```

```
## # A tibble: 6 x 2
##   customer_id     n
##         <dbl> <int>
## 1         431     1
## 2         922     1
## 3        1488     1
## 4        1920     1
## 5        2135     1
## 6        3464     1
```

Thus it is clear that the 196 records included customers who had visited before but had all the records taken therefore missing values can't be removed.

The missing values will be replaced as the analysis progresses.

create month and day from tran_date

```
transactions_1 <- transactions_1 %>%
  mutate(tran_month = month(tran_date, label = TRUE, abbr = TRUE),
         tran_day = wday(tran_date, label = TRUE, abbr = TRUE))
transactions_1 <- transactions_1 %>% select(1:4,14:15,5:13)
head(transactions_1)
```

```
## # A tibble: 6 x 15
##   tran_id product_id customer_id tran_date  tran_month tran_day online_order
##     <dbl>      <dbl>       <dbl> <date>     <ord>      <ord>           <dbl>
## 1       1          2        2950 2017-02-25 Feb        Sat                 0
## 2       2          3        3120 2017-05-21 May        Sun                 1
## 3       3         37         402 2017-10-16 Oct        Mon                 0
## 4       4         88        3135 2017-08-31 Aug        Thu                 0
## 5       5         78         787 2017-10-01 Oct        Sun                 1
## 6       6         25        2339 2017-03-08 Mar        Wed                 1
## # i 8 more variables: order_status <chr>, brand <chr>, product_line <chr>,
## #   product_class <chr>, product_size <chr>, list_price <dbl>,
## #   standard_cost <dbl>, first_sold_date <date>
```

1.1 Did the company have repeat customers

```
n_distinct(transactions_1$customer_id)
```

```
## [1] 3493
```

transactions data had 3493 distinct customers

```
range(transactions_1$customer_id) ## minimum and maximum assigned number
```

```
## [1]    1 5034
```

are the assigned numbers consistent from 1 to 5034

```r
trans_customer_id_notseen <- as_tibble(setdiff(1:5034, transactions_1$customer_id))
dim(trans_customer_id_notseen)
```

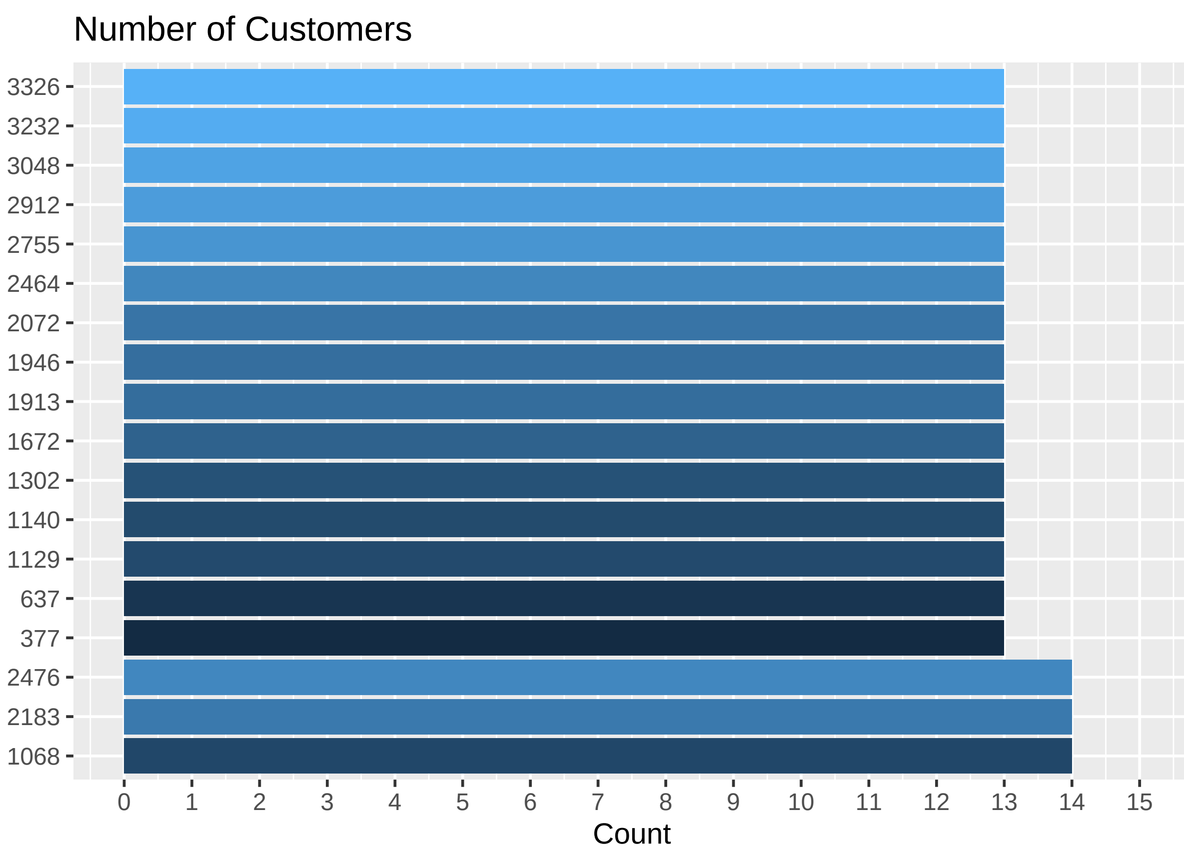
```
## [1] 1541    1
```

There were 1541 between 1 and 5034 not used.

Transactions per customer

```r
tran_count <- transactions_1 %>% count(customer_id, sort = T)
```

```r
transactions_1 %>% count(customer_id, sort = T) %>% filter(n > 12) %>%
  ggplot(aes(reorder(x = customer_id, -n), y = n, fill = customer_id)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(legend.position = "none") +
  scale_y_continuous("Count",
    breaks = seq(0, 15, by = 1),
    limits = c(0, 15)
  ) +
  labs(title = "Number of Customers", x = "") +
  coord_flip()
```



Customers recorded more than once

```
tran_count_more <- tran_count %>% filter(n > 1)
dim(tran_count_more) ## repeat customers
```

```
## [1] 3444    2
```

There were 3444 repeat customers. Of which the maximum number of times a customer has purchased was 14 times which had three customers.

Recorded only once

```
tran_count_once <- tran_count %>% filter(n < 2)
dim(tran_count_once) ## single transaction record
```

```
## [1] 49  2
```

49 customers were recorded only once

Customers recorded only once

```
one_time_customers <- transactions_1[transactions_1$customer_id %in% tran_count_once$customer_
```

These 49 customers only visited once in 2017

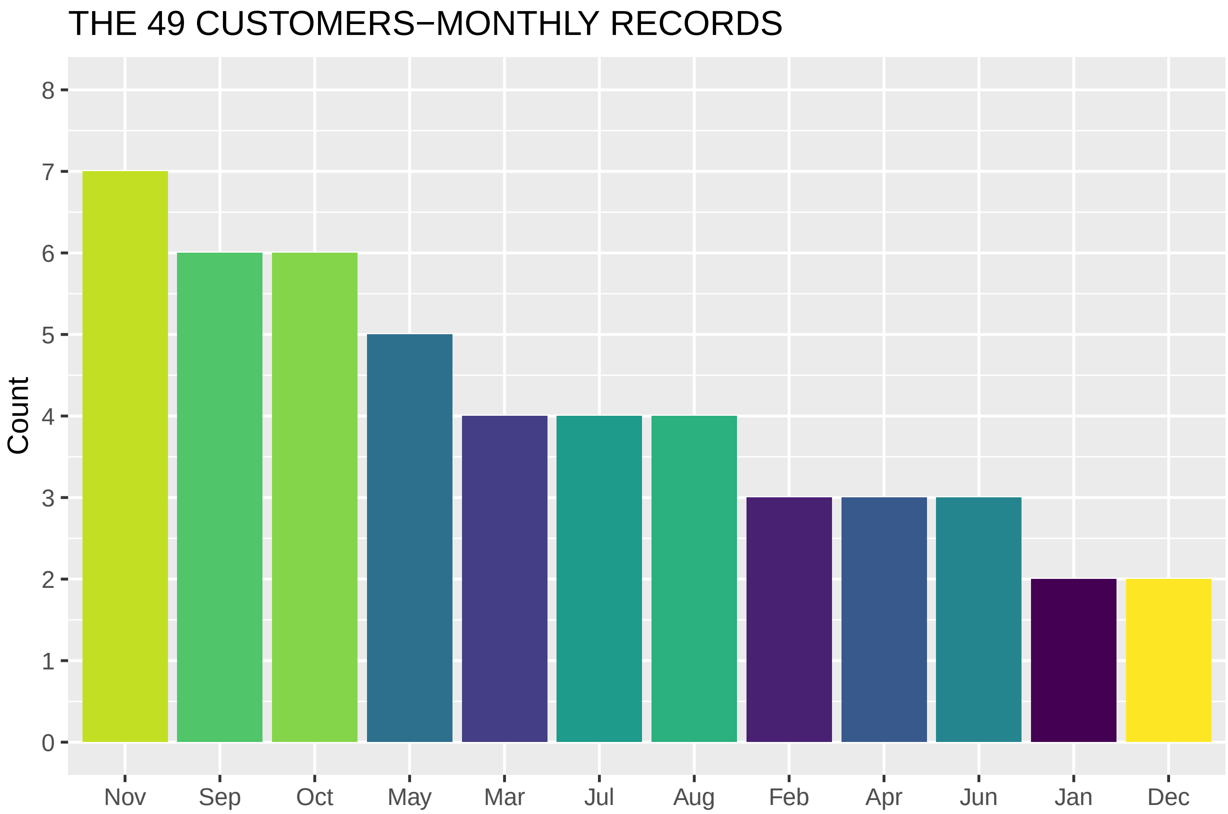
The 49-Record Month

```
one_time_customers %>% count(tran_month, sort = T)
```

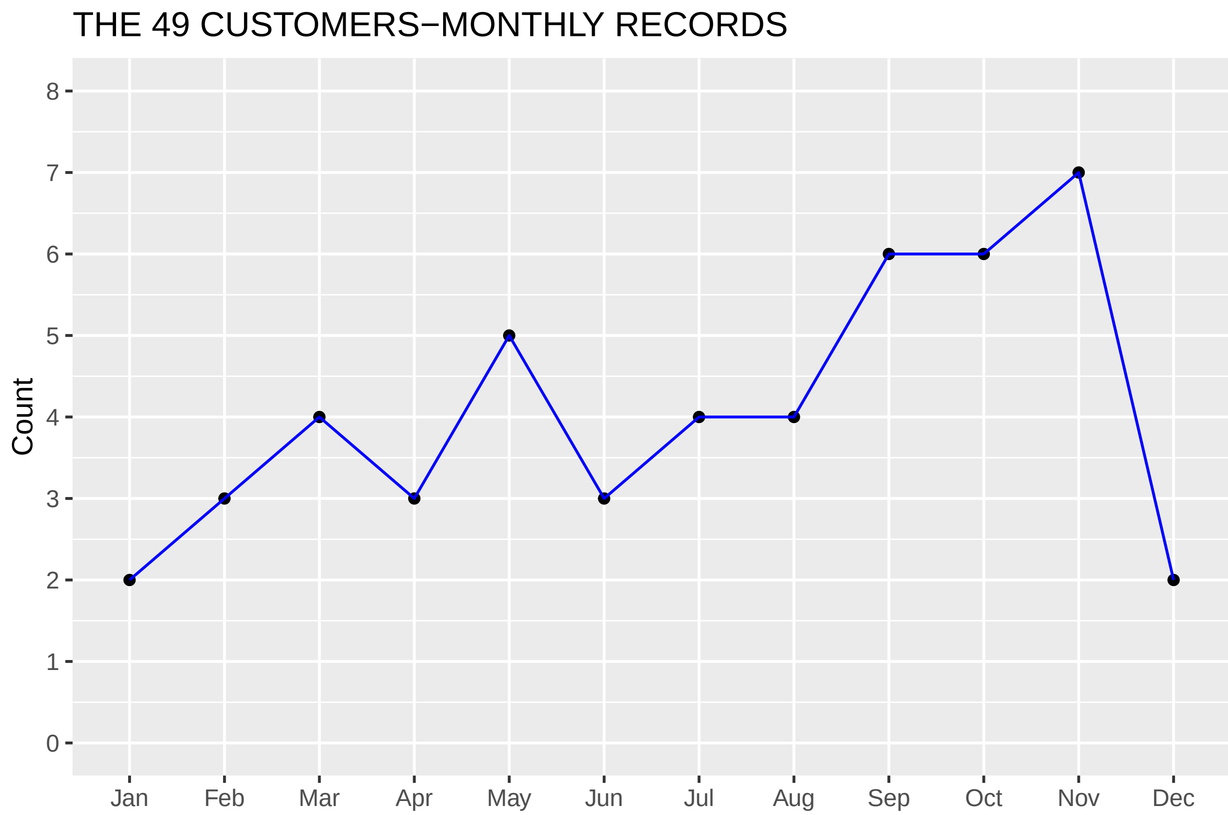
```
## # A tibble: 12 x 2
##    tran_month     n
##    <ord>      <int>
##  1 Nov            7
##  2 Sep            6
##  3 Oct            6
##  4 May            5
##  5 Mar            4
##  6 Jul            4
##  7 Aug            4
##  8 Feb            3
```

```
##   9 Apr            3
## 10 Jun            3
## 11 Jan            2
## 12 Dec            2
```

```r
one_time_customers %>% count(tran_month) %>%
  ggplot(aes(reorder(x = tran_month, -n), y = n, fill = tran_month)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(legend.position = "none") +
  scale_y_continuous("Count",
    breaks = seq(0, 8, by = 1),
    limits = c(0, 8)
  ) +
  labs(title = "THE 49 CUSTOMERS-MONTHLY RECORDS", x = "")
```




```r
one_time_customers %>% count(tran_month) %>%
  ggplot(aes(x = tran_month, y = n, group = 1)) +
  geom_point() +
  geom_line(colour = "blue") +
  theme(legend.position = "none") +
  scale_y_continuous("Count",
                     breaks = seq(0, 8, by = 1),
                     limits = c(0, 8)) +
  labs(title = "THE 49 CUSTOMERS-MONTHLY RECORDS", x = "")
```



It is seen that for the single records data the sales were mostly in November, September and October and they were lowest in January, June and December.

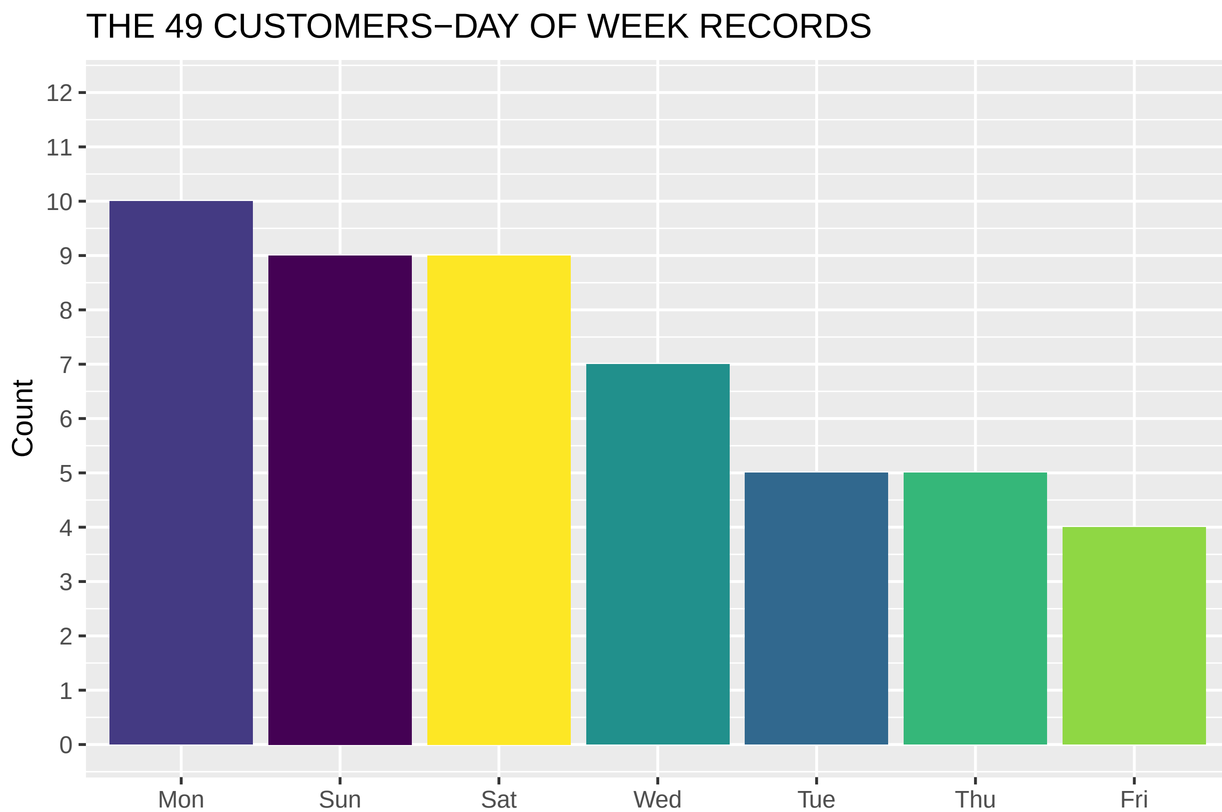
For the single sales for the year 2017, there was a steady rise from January to March, then a decline in April followed by a rise again in May then a sharp decline in June then a rise all through to November and a further sharp decline in the month of December.

The 49-Day of week that they purchased

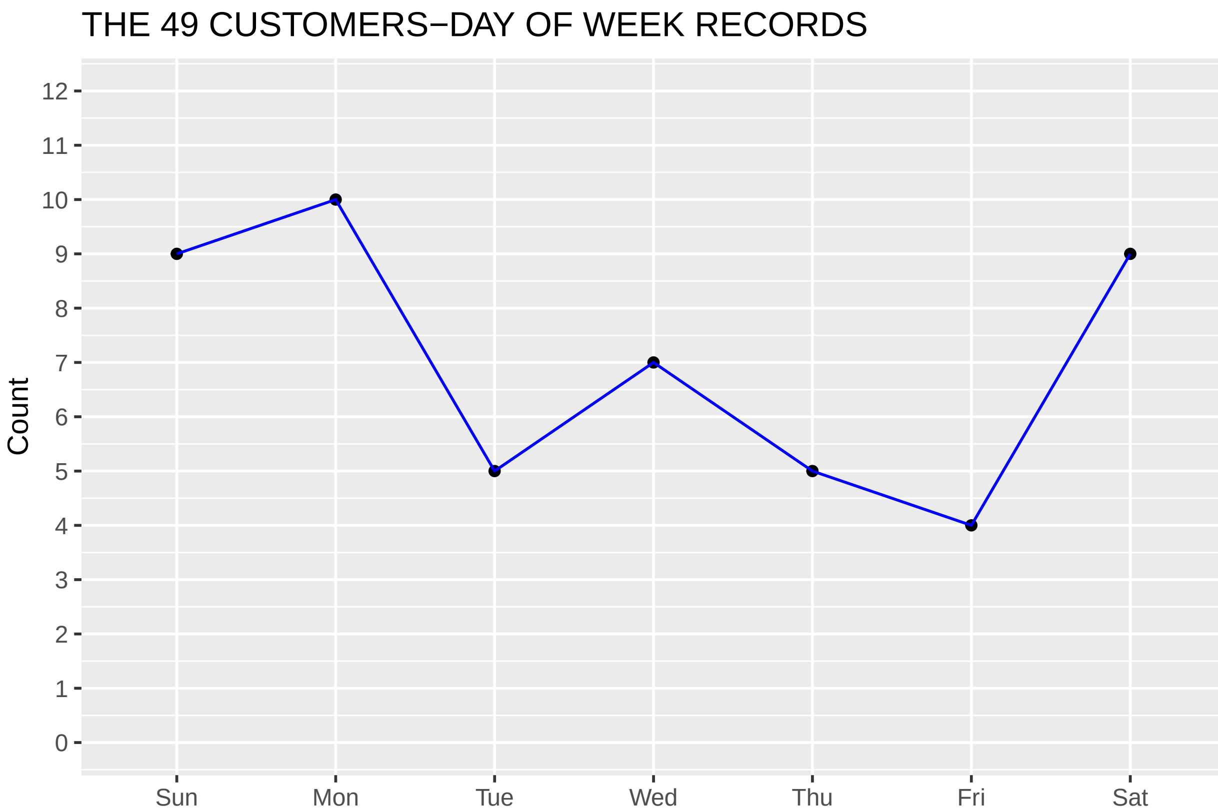
```
one_time_customers %>% count(tran_day, sort = T)
```

```
## # A tibble: 7 x 2
##   tran_day     n
##   <ord>    <int>
## 1 Mon         10
## 2 Sun          9
## 3 Sat          9
## 4 Wed          7
## 5 Tue          5
## 6 Thu          5
## 7 Fri          4
```

```
one_time_customers %>% count(tran_day) %>%
  ggplot(aes(reorder(x = tran_day, -n), y = n, fill = tran_day)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(legend.position = "none") +
  scale_y_continuous("Count",
    breaks = seq(0, 12, by = 1),
    limits = c(0, 12)
  ) +
  labs(title = "THE 49 CUSTOMERS-DAY OF WEEK RECORDS", x = "")
```




```
one_time_customers %>% count(tran_day) %>%
  ggplot(aes(x = tran_day, y = n, group = 1)) +
  geom_point() +
  geom_line(colour = "blue") +
  theme(legend.position = "none") +
  scale_y_continuous("Count",
                     breaks = seq(0, 12, by = 1),
                     limits = c(0, 12)) +
  labs(title = "THE 49 CUSTOMERS-DAY OF WEEK RECORDS", x = "")
```



Amongst the single record customers, they preferred to purchase on Monday and did not like to purchase on Friday. Saturday, Sunday and Monday were good purchase days.

The 49-ORDER TYPE

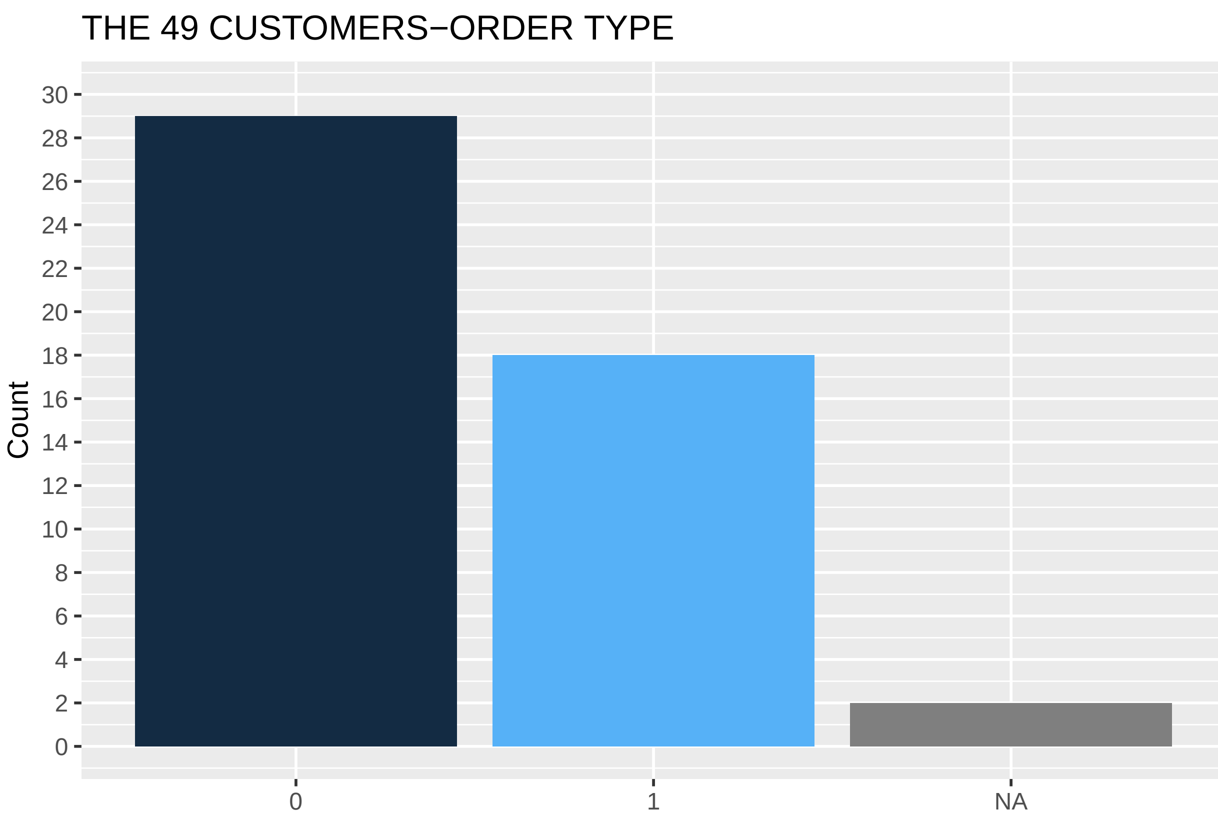
Taking 0 to mean physical order and 1 to mean online order

```
one_time_customers %>% count(online_order, sort = T)
```

```
## # A tibble: 3 x 2
##   online_order     n
##          <dbl> <int>
## 1            0    29
## 2            1    18
## 3           NA     2
```

```
one_time_customers %>% count(online_order) %>%
  ggplot(aes(reorder(x = online_order, -n), y = n, fill = online_order)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(legend.position = "none") +
  scale_y_continuous("Count",
```

```
    breaks = seq(0, 30, by = 2),
    limits = c(0, 30)
  ) +
  labs(title = "THE 49 CUSTOMERS-ORDER TYPE", x = "")
```



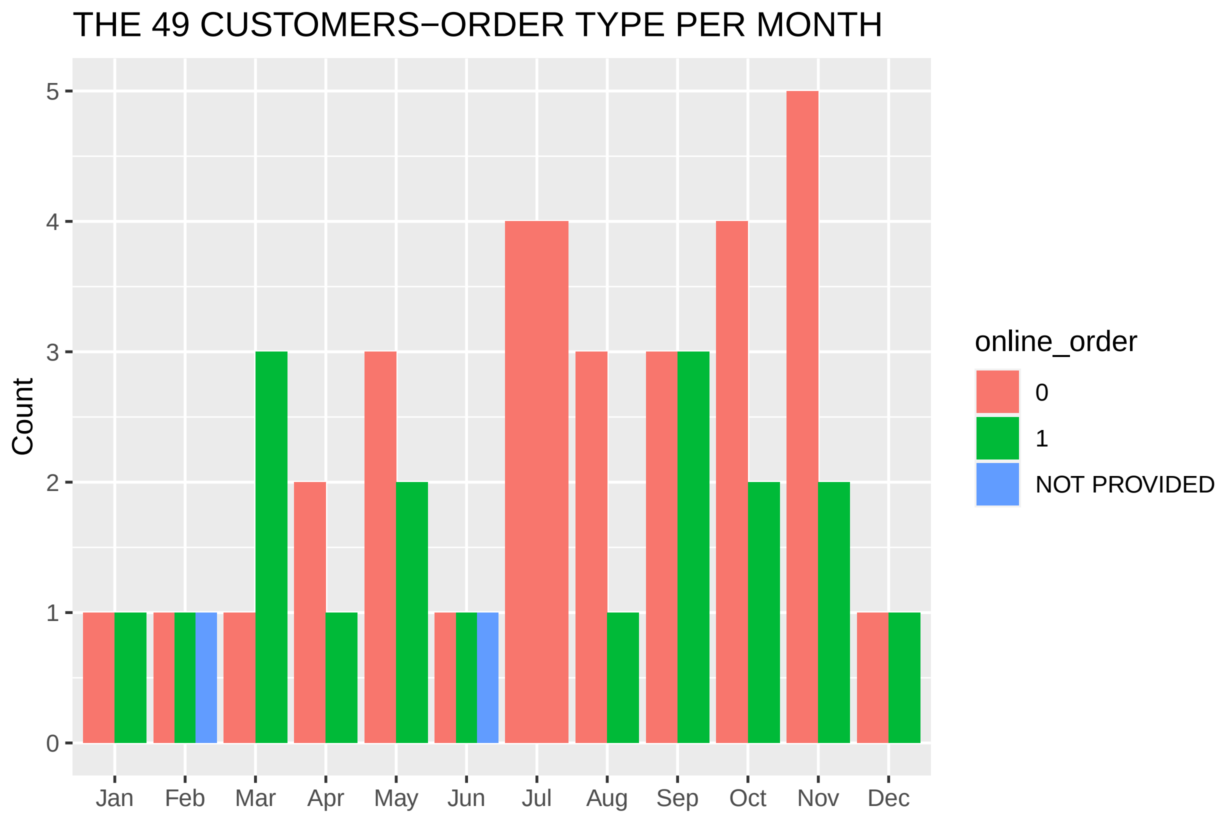
The orders were mostly of order type 0

Make online a factor with 3 levels.

```
one_time_customers$online_order[is.na(one_time_customers$online_order)] <- "NOT PROVIDED"
class(one_time_customers$online_order)
```

```
## [1] "character"
```

```
one_time_customers_month_order <- one_time_customers %>% group_by(tran_month) %>% count(online_
ggplot(one_time_customers_month_order, aes(tran_month, n,
                                           fill = online_order)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "THE 49 CUSTOMERS-ORDER TYPE PER MONTH", x = "",
       y = "Count")
```



Taking online_order 0 to mean order not made online we get that walk-ins were always higher than across months except in March.

The 49-Product ID

Convert product_id to factor with 101 levels

```
one_time_customers$product_id <- as.factor(
  as.numeric(one_time_customers$product_id))
class(one_time_customers$product_id)  ### class of product_id
```

```
## [1] "factor"
```

```
product_id_onetime_count <- one_time_customers %>% count(product_id, sort = T)
product_id_onetime_count_1 <- product_id_onetime_count %>% filter(n == 1)
product_id_onetime_count_2 <- product_id_onetime_count %>% filter(n > 1)
```

```
n_distinct(one_time_customers$product_id)
```

```
## [1] 40
```

Of the 49 one time customers, 8 products were bought more than once while 32 products were only purchased once.

Product id and online order

```
product_id_onetime_order_count <- one_time_customers %>% group_by(online_order) %>% count(produ
product_id_onetime_order_count
```

```
## # A tibble: 45 x 3
## # Groups:   online_order [3]
##    online_order product_id     n
##    <chr>        <fct>      <int>
##  1 0            2              1
##  2 0            4              1
##  3 0            5              1
##  4 0            7              1
##  5 0            13             1
##  6 0            16             1
##  7 0            21             1
##  8 0            25             1
##  9 0            30             1
## 10 0            45             2
## # i 35 more rows
```

```
product_id_onetime_order_count_1 <- product_id_onetime_order_count %>%
  filter(n == 1)
product_id_onetime_order_count_1
```

```
## # A tibble: 41 x 3
## # Groups:   online_order [3]
##    online_order product_id     n
##    <chr>        <fct>      <int>
##  1 0            2              1
##  2 0            4              1
##  3 0            5              1
##  4 0            7              1
##  5 0            13             1
##  6 0            16             1
##  7 0            21             1
##  8 0            25             1
##  9 0            30             1
## 10 0            51             1
## # i 31 more rows
```

```
product_id_onetime_order_count_2 <- product_id_onetime_order_count %>%
  filter(n > 1)
product_id_onetime_order_count_2
```

```
## # A tibble: 4 x 3
## # Groups:   online_order [2]
##   online_order product_id     n
##   <chr>        <fct>      <int>
## 1 0            45             2
## 2 0            86             2
## 3 1            0              2
## 4 1            21             2
```

4 products were ordered twice each

Product id and month

```
product_id_onetime_month_count <- one_time_customers %>% group_by(tran_month) %>% count(produc
product_id_onetime_month_count
```

```
## # A tibble: 48 x 3
## # Groups:   tran_month [12]
##    tran_month product_id     n
##    <ord>      <fct>      <int>
##  1 Jan        28             1
##  2 Jan        45             1
##  3 Feb        25             1
##  4 Feb        35             1
##  5 Feb        41             1
##  6 Mar        21             1
##  7 Mar        22             1
##  8 Mar        60             1
##  9 Mar        95             1
## 10 Apr        74             1
## # i 38 more rows
```

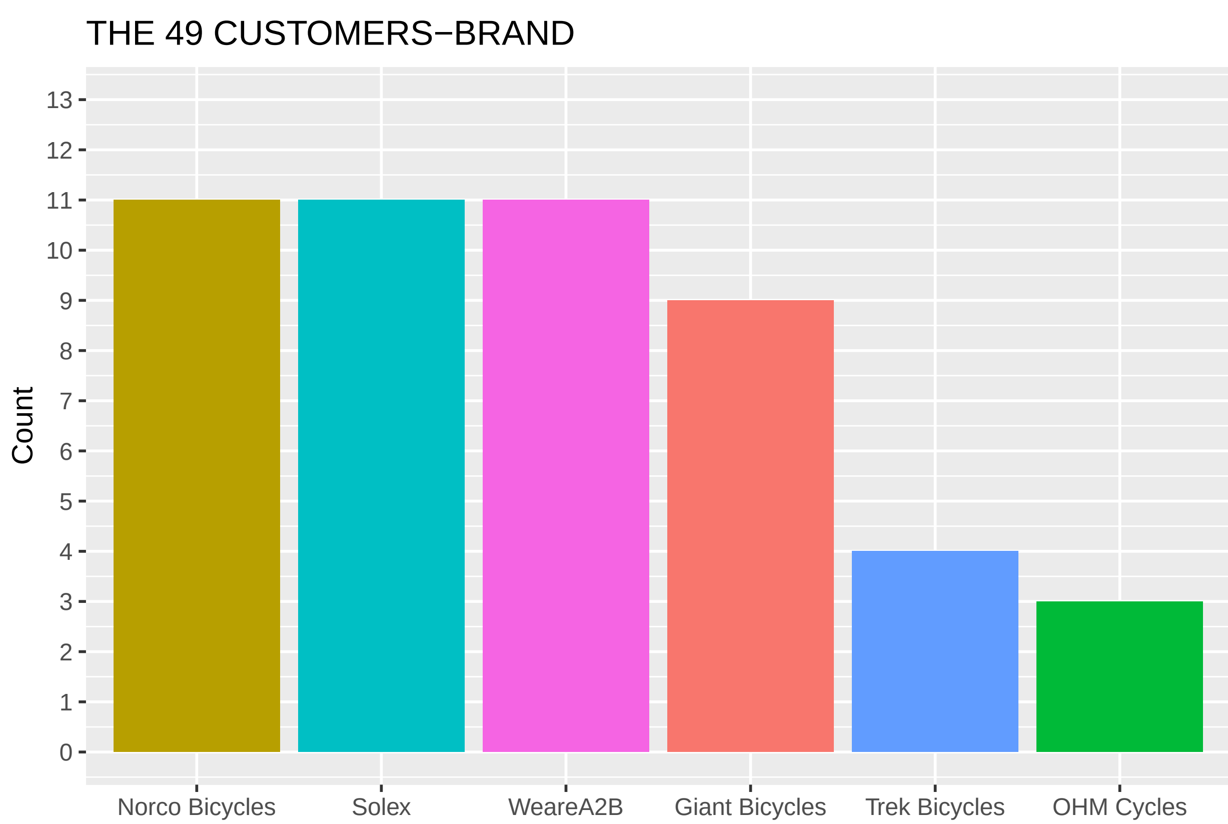
The orders picked in the usual month.

The 49-Brand

```
one_time_customers %>% count(brand, sort = T)
```

```
## # A tibble: 6 x 2
##   brand              n
##   <chr>          <int>
## 1 Norco Bicycles    11
## 2 Solex             11
## 3 WeareA2B          11
## 4 Giant Bicycles     9
## 5 Trek Bicycles      4
## 6 OHM Cycles         3
```

```r
one_time_customers %>% count(brand) %>%
  ggplot(aes(reorder(x = brand, -n), y = n, fill = brand)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(legend.position = "none") +
  scale_y_continuous("Count",
    breaks = seq(0, 13, by = 1),
    limits = c(0, 13)
  ) +
  labs(title = "THE 49 CUSTOMERS-BRAND", x = "")
```

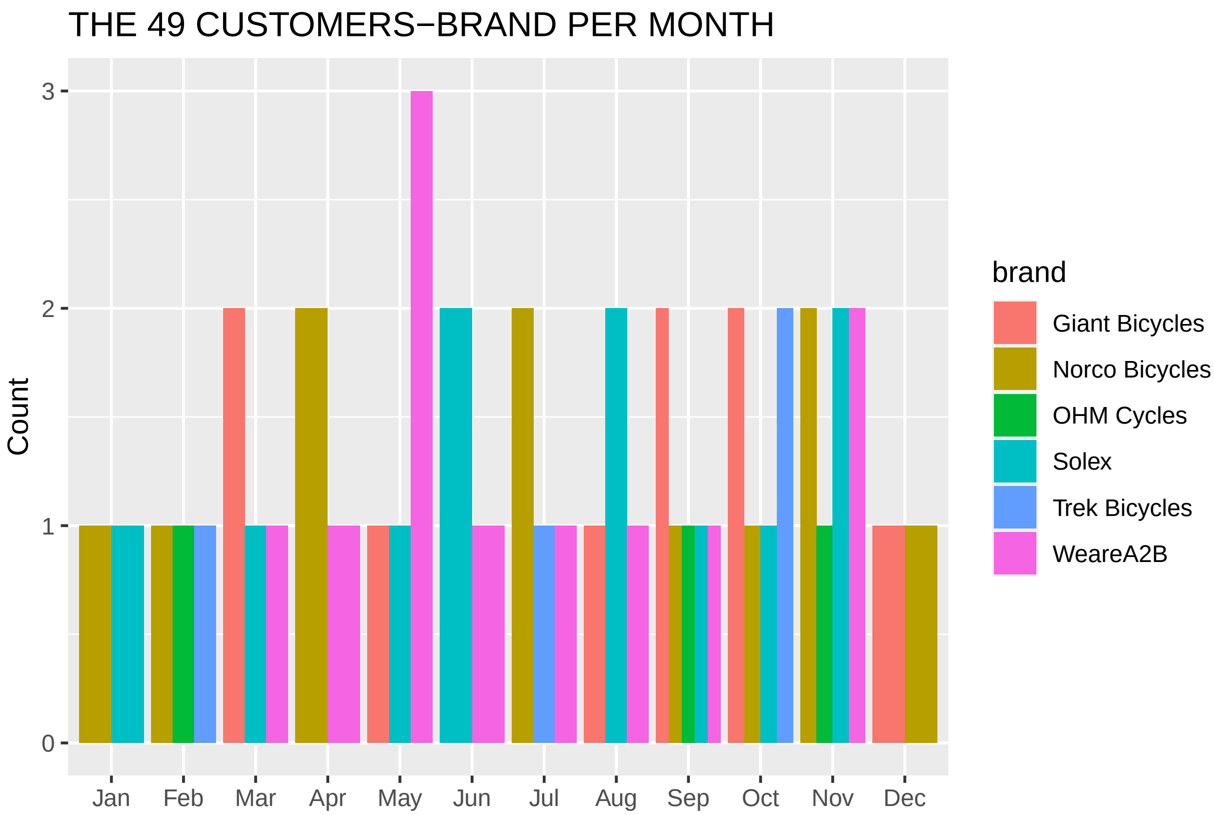


The 49-Brand Month

```r
one_time_brand_month <- one_time_customers %>% group_by(tran_month) %>% count(brand)
one_time_brand_month
```

```
## # A tibble: 36 x 3
## # Groups:   tran_month [12]
##    tran_month brand              n
##    <ord>      <chr>          <int>
##  1 Jan        Norco Bicycles     1
##  2 Jan        Solex              1
##  3 Feb        Norco Bicycles     1
##  4 Feb        OHM Cycles         1
##  5 Feb        Trek Bicycles      1
##  6 Mar        Giant Bicycles     2
##  7 Mar        Solex              1
##  8 Mar        WeareA2B           1
##  9 Apr        Norco Bicycles     2
## 10 Apr        WeareA2B           1
## # i 26 more rows
```

```r
ggplot(one_time_brand_month, aes(tran_month, n,
                                          fill = brand)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "THE 49 CUSTOMERS-BRAND PER MONTH", x = "",
       y = "Count")
```



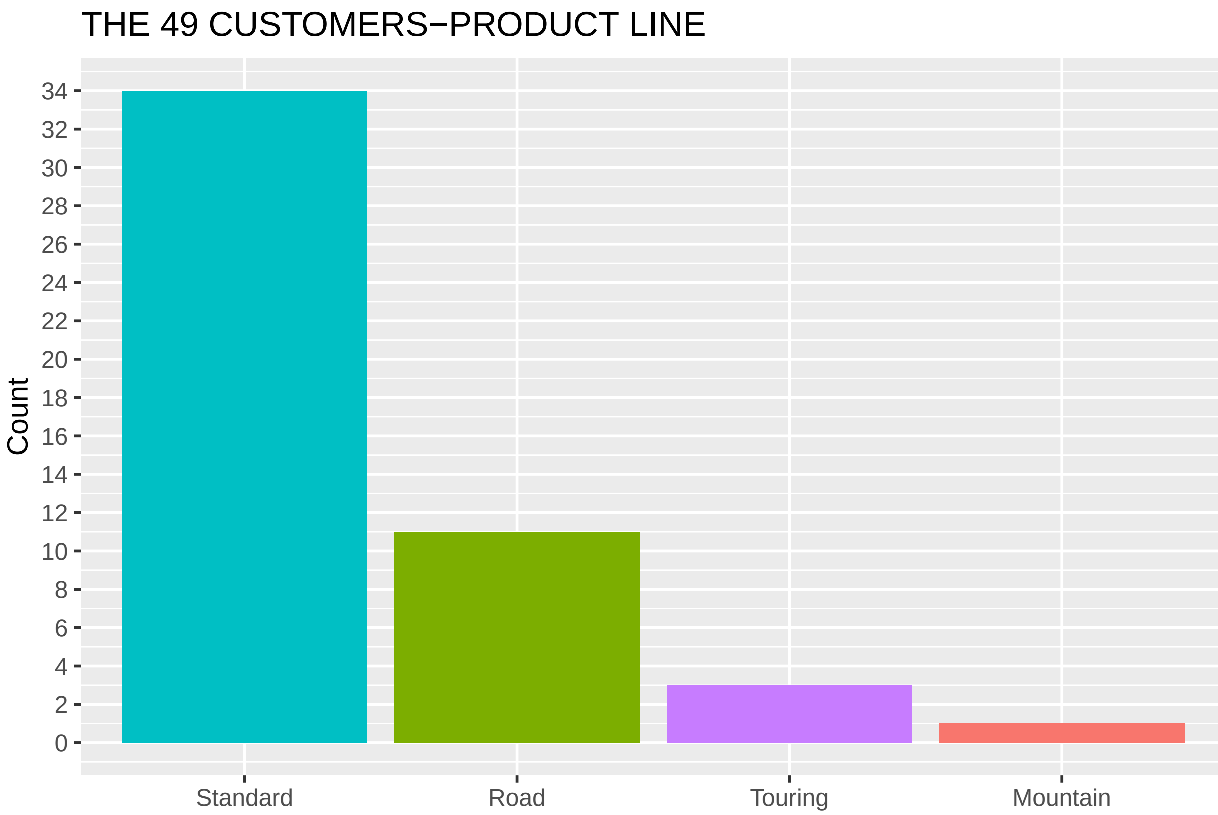
With the 49-the most purchased brand were Solex, Norco Bicycles and WeareA2B.

The 49-Product line

```
one_time_customers %>% count(product_line, sort = T)
```

```
## # A tibble: 4 x 2
##   product_line     n
##   <chr>        <int>
## 1 Standard        34
## 2 Road            11
## 3 Touring          3
## 4 Mountain         1
```

```
one_time_customers %>% count(product_line) %>%
  ggplot(aes(reorder(x = product_line, -n), y = n, fill = product_line)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(legend.position = "none") +
  scale_y_continuous("Count",
    breaks = seq(0, 34, by = 2),
    limits = c(0, 34)
  ) +
  labs(title = "THE 49 CUSTOMERS-PRODUCT LINE", x = "")
```

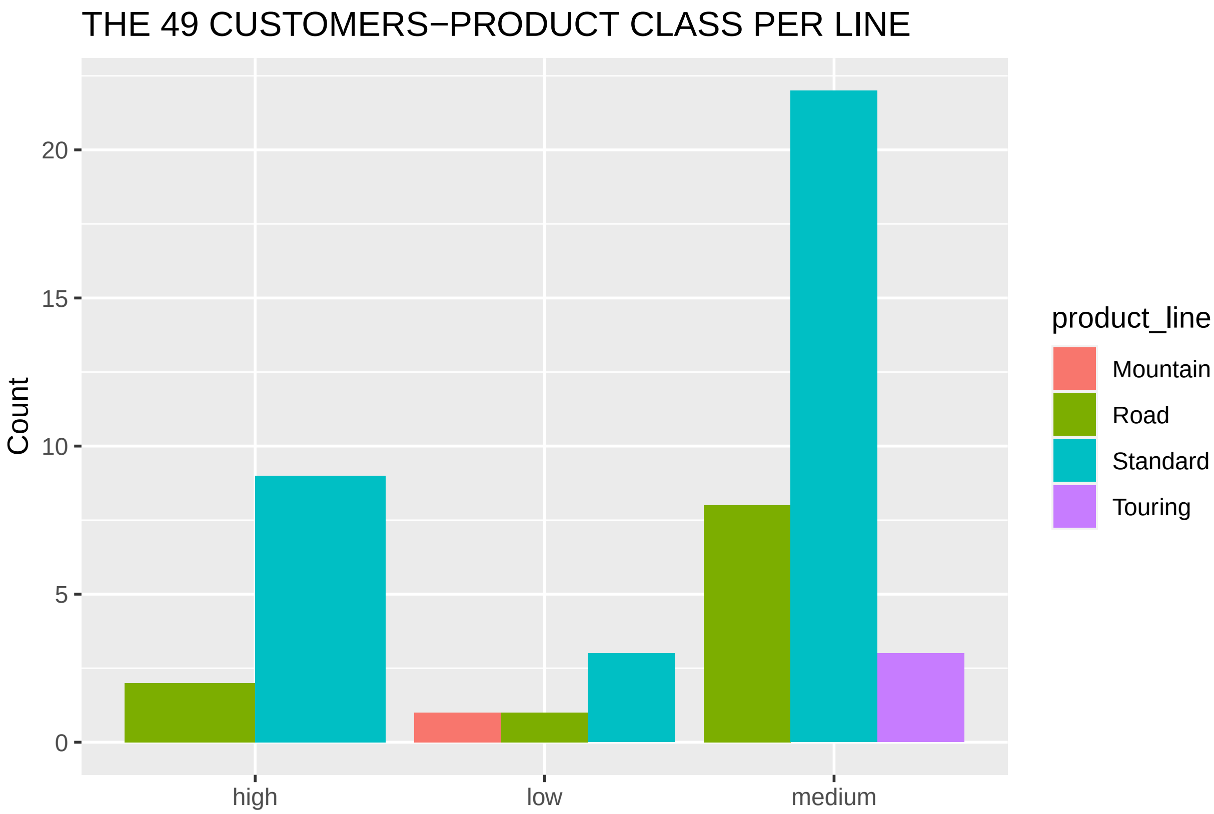


The 49-Product line and Product Class

```
one_time_lineclass <- one_time_customers %>% group_by(product_class) %>% count(product_line)
one_time_lineclass
```

```
## # A tibble: 8 x 3
## # Groups:   product_class [3]
##   product_class product_line     n
##   <chr>         <chr>        <int>
## 1 high          Road             2
## 2 high          Standard         9
## 3 low           Mountain         1
## 4 low           Road             1
## 5 low           Standard         3
## 6 medium        Road             8
## 7 medium        Standard        22
## 8 medium        Touring          3
```

```
ggplot(one_time_lineclass, aes(product_class, n,
                                          fill = product_line)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "THE 49 CUSTOMERS-PRODUCT CLASS PER LINE", x = "",
       y = "Count")
```



Standard product line was preferred across the 3 product class.

Medium class always had higher purchases in the different product lines

- The one time customers visits were clearly different for the different variables.

Regular Customers

```
regular_customers <- transactions_1[transactions_1$customer_id %in% tran_count_more$customer_ic
```

We thus have 19,772 approved transactions.

```
n_distinct(regular_customers$customer_id) ### How many were regular customers
```

```
## [1] 3444
```

We had 3444 regular customers with 19,772 distinct transactions.

Number of times recorded

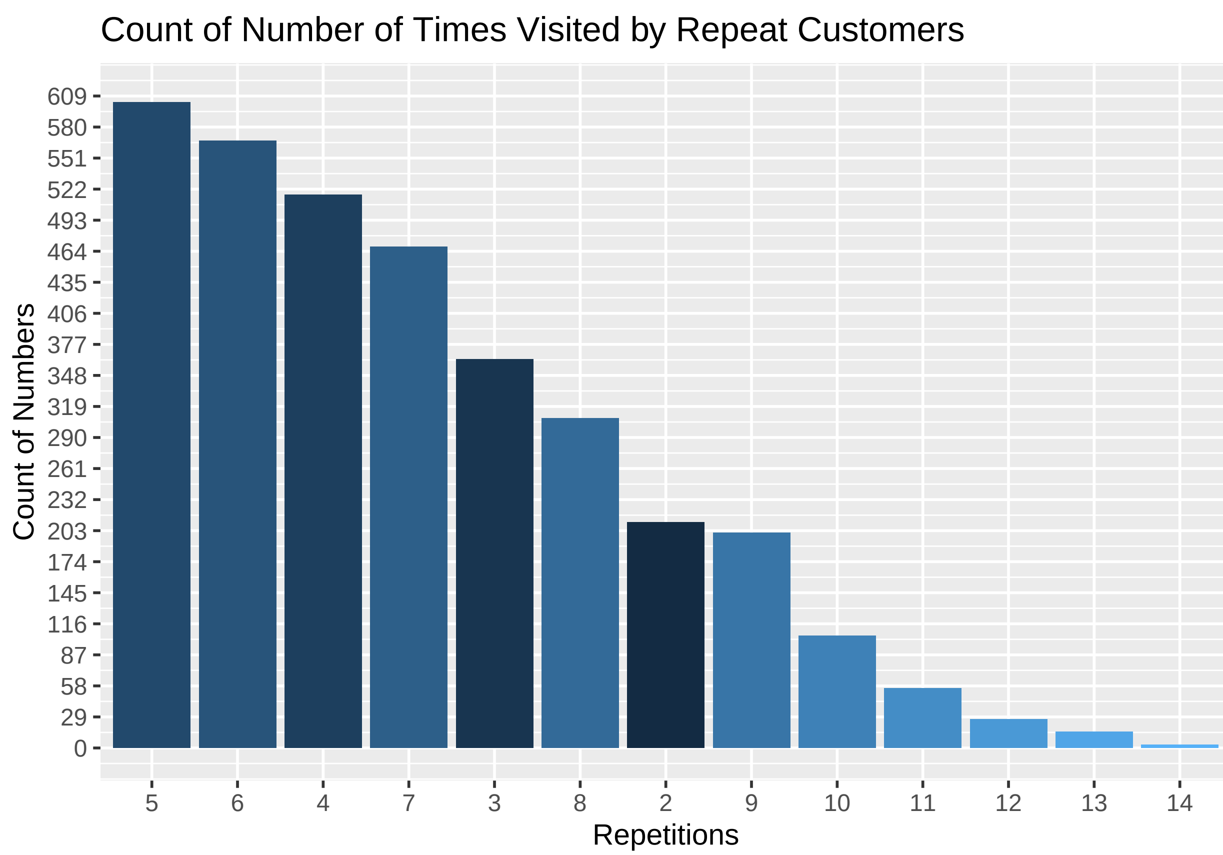
Convert customer_id to factors

```
regular_customers$customer_id <- as.factor(
  as.numeric(regular_customers$customer_id))
regular_customers_idcount <- regular_customers %>% count(customer_id, sort = T)
regular_customers_idcount <- regular_customers_idcount %>% rename(numbers = n)
regular_customers_idcount_1 <- regular_customers_idcount %>%
  count(numbers, sort = T)
regular_customers_idcount_1
```

```
## # A tibble: 13 x 2
##    numbers     n
##      <int> <int>
##  1       5   603
##  2       6   567
##  3       4   517
##  4       7   468
##  5       3   363
##  6       8   308
##  7       2   211
##  8       9   201
##  9      10   105
## 10      11    56
## 11      12    27
## 12      13    15
## 13      14     3
```

It is seen that customers who were recorded 5 times were the most with 603 customers having been recorded 5 times, 567 recorded 6 times, 4 recorded 517 times, 3 recorded 14 times, 15 recorded 13 times and 12 recorded 27 times.

```
regular_customers_idcount %>% count(numbers) %>%
  ggplot(aes(reorder(x = numbers, -n), y = n, fill = numbers)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(legend.position = "none") +
  scale_y_continuous("Count of Numbers",
    breaks = seq(0, 609, by = 29),
    limits = c(0, 609)
  ) +
  labs(title = "Count of Number of Times Visited by Repeat Customers",
       x = "Repetitions")
```



PRODUCT ID

```r
class(regular_customers$product_id)
```

```
## [1] "numeric"
```

```r
regular_customers$product_id <- as.factor(
  as.numeric(regular_customers$product_id))
regular_prodid_count <- regular_customers %>% count(product_id, sort = T)
regular_prodid_count
```

```
## # A tibble: 101 x 2
##    product_id     n
##    <fct>      <int>
##  1 0           1369
##  2 3            350
##  3 1            309
##  4 38           266
##  5 35           265
##  6 4            239
```

```
##   7 2              238
##   8 90             224
##   9 80             222
## 10 12             221
## # i 91 more rows
```

product of id 0 were the most sought.

Can a products have the same id and be different.

Filter product_id 0

```
regular_prod0 <- regular_customers %>% filter(product_id == 0)
regular_solex <- regular_customers %>% filter(brand == "Solex" &
                                               product_line == "Standard" &
                                               product_class == "medium" &
                                               product_size == "medium")
head(regular_prod0)
```

```
## # A tibble: 6 x 15
##   tran_id product_id customer_id tran_date  tran_month tran_day online_order
##     <dbl> <fct>      <fct>       <date>     <ord>      <ord>           <dbl>
## 1      35 0          2171        2017-08-20 Aug        Sun                 0
## 2      40 0          2448        2017-11-28 Nov        Tue                 1
## 3      55 0          3140        2017-09-18 Sep        Mon                 0
## 4      61 0          1839        2017-02-24 Feb        Fri                 0
## 5      64 0          2000        2017-07-08 Jul        Sat                 0
## 6      83 0          3398        2017-04-01 Apr        Sat                 1
## # i 8 more variables: order_status <chr>, brand <chr>, product_line <chr>,
## #   product_class <chr>, product_size <chr>, list_price <dbl>,
## #   standard_cost <dbl>, first_sold_date <date>
```

Each brand can have different product_id 0 and we have 101 distinct product ids, therefore it can be said that brand, product_line, prodduct_class and product_size describe a product.

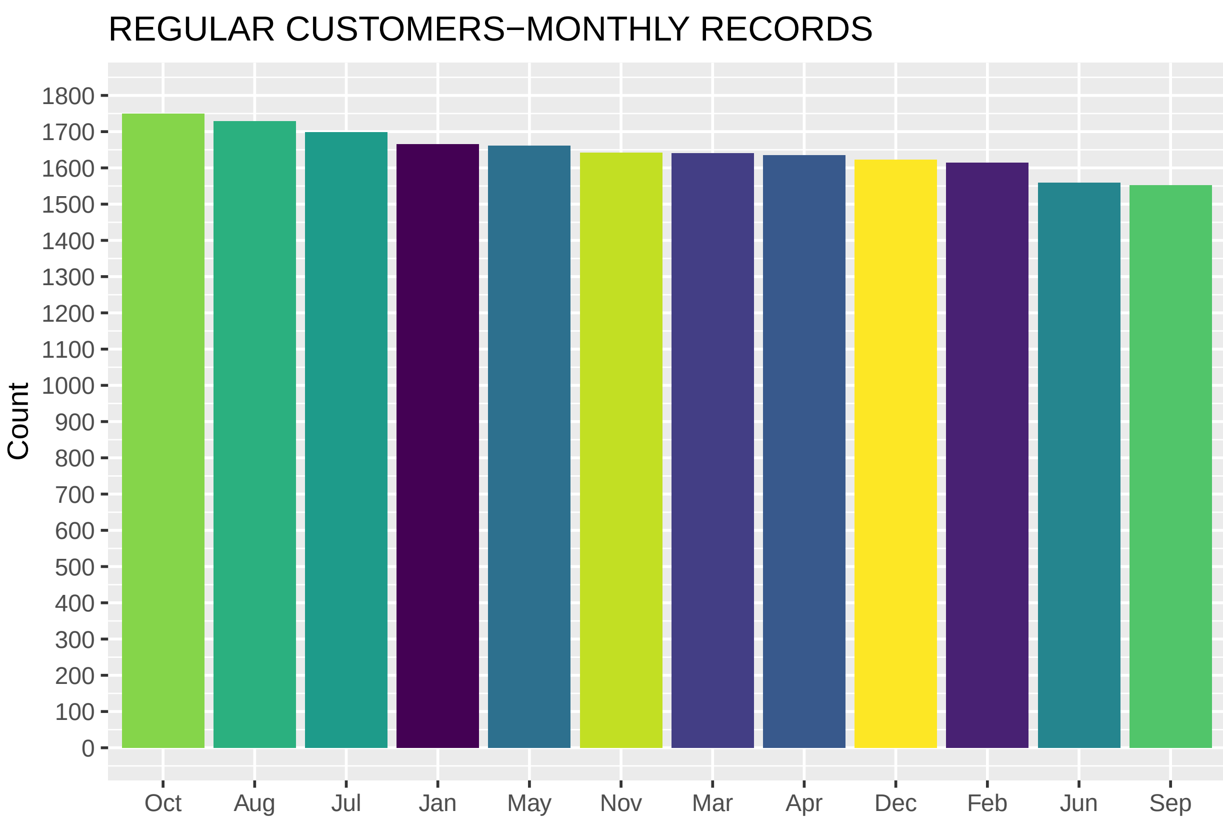
Regular Customers Month

```
regular_customers %>% count(tran_month, sort = T)
```

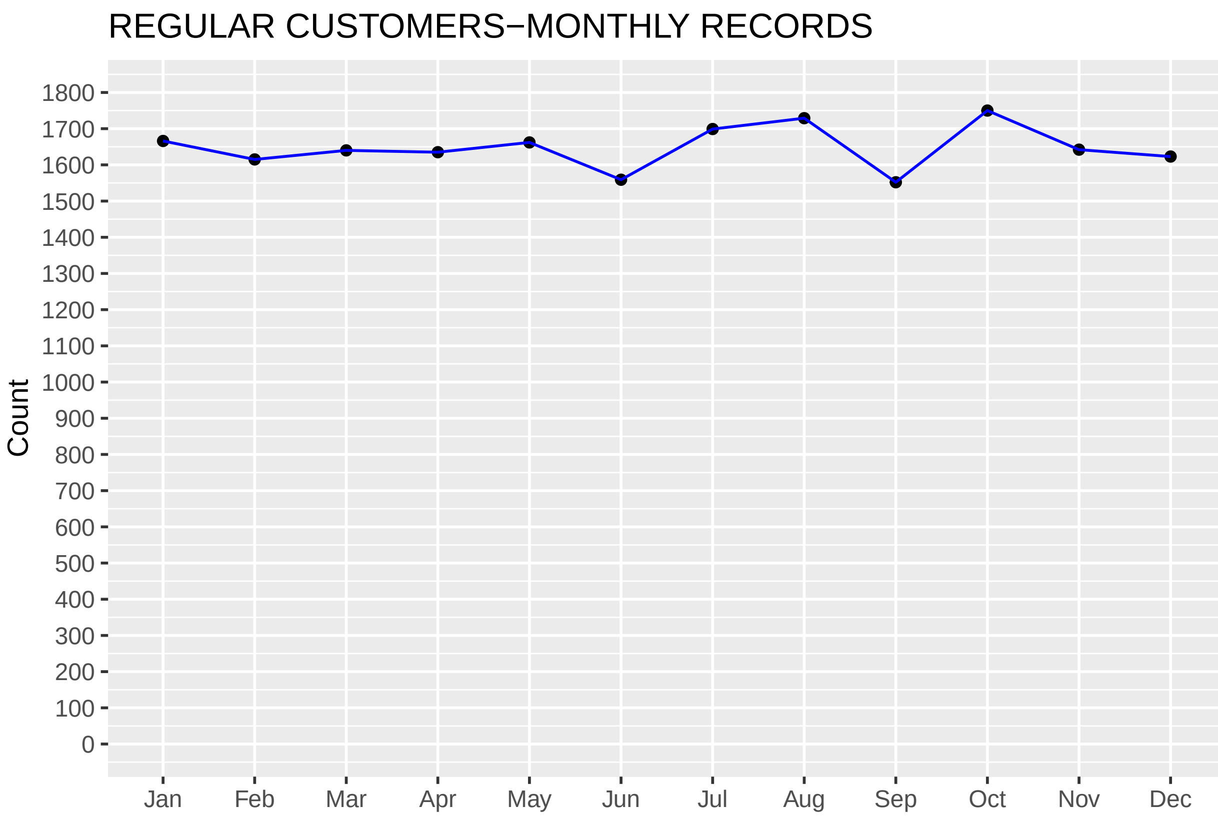
```
## # A tibble: 12 x 2
##    tran_month     n
##    <ord>      <int>
##  1 Oct         1750
```

```
##   2 Aug          1729
##   3 Jul          1699
##   4 Jan          1666
##   5 May          1662
##   6 Nov          1642
##   7 Mar          1640
##   8 Apr          1635
##   9 Dec          1623
##  10 Feb          1615
##  11 Jun          1559
##  12 Sep          1552
```

```r
regular_customers %>% count(tran_month) %>%
  ggplot(aes(reorder(x = tran_month, -n), y = n, fill = tran_month)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(legend.position = "none") +
  scale_y_continuous("Count",
    breaks = seq(0, 1800, by = 100),
    limits = c(0, 1800)
  ) +
  labs(title = "REGULAR CUSTOMERS-MONTHLY RECORDS", x = "")
```



```
regular_customers %>% count(tran_month) %>%
  ggplot(aes(x = tran_month, y = n, group = 1)) +
  geom_point() +
  geom_line(colour = "blue") +
  theme(legend.position = "none") +
  scale_y_continuous("Count",
                     breaks = seq(0, 1800, by = 100),
                     limits = c(0, 1800)) +
  labs(title = "REGULAR CUSTOMERS-MONTHLY RECORDS", x = "")
```



The difference between the highest Month sale and lowest monthly was 198 sales. Monthly differences are not so much but the sales picked in October and lowest in September. Sales were high in October, August and July and lowest in February, June and September.

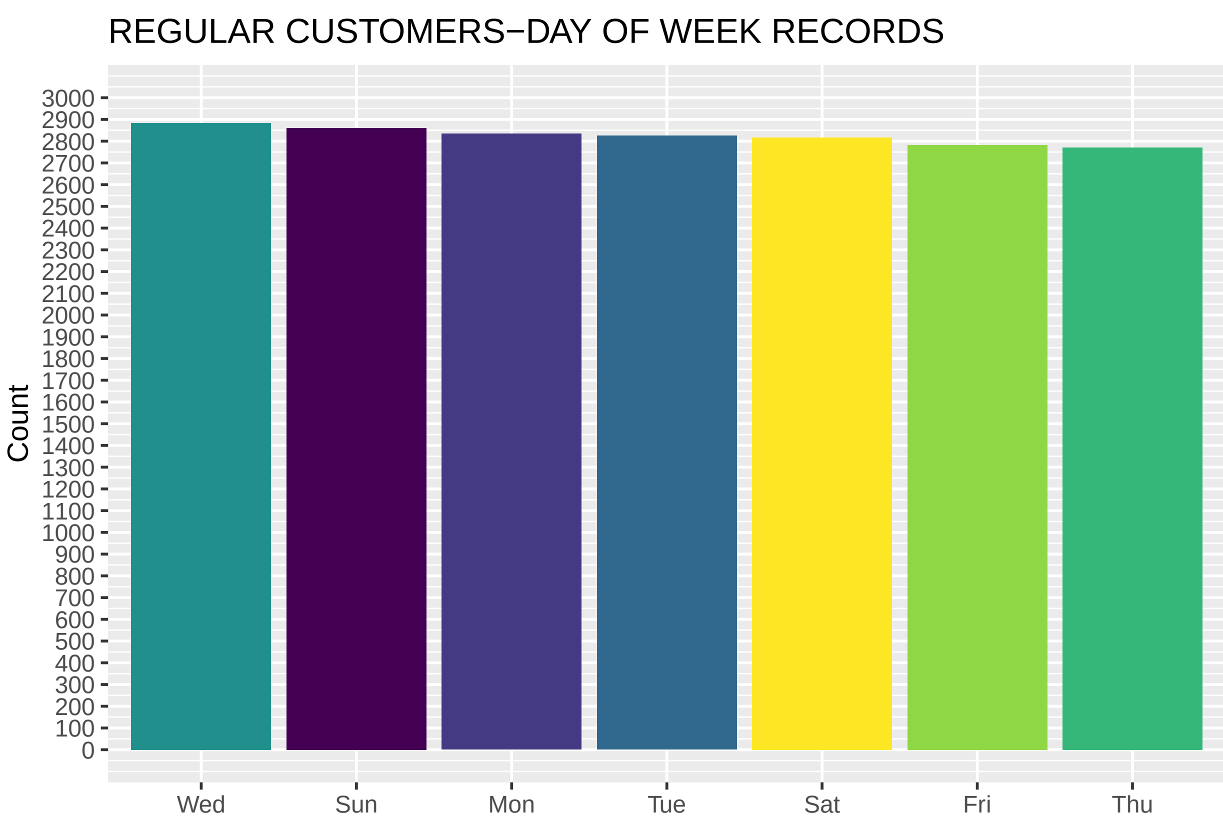
August to September had the sharpest decline while September to October had the highest ascend.

Regular Customers day of week that they purchased

```
regular_customers %>% count(tran_day, sort = T)
```

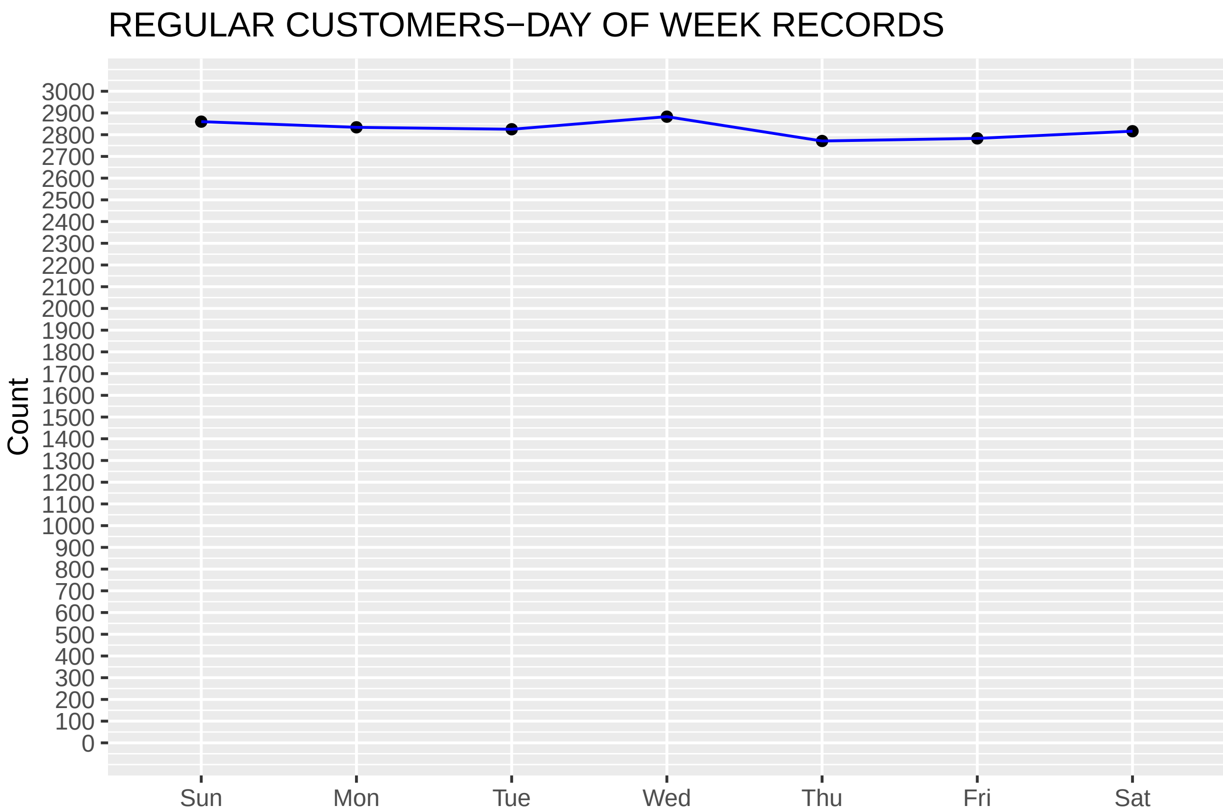
```
## # A tibble: 7 x 2
##   tran_day     n
##   <ord>    <int>
## 1 Wed       2883
## 2 Sun       2860
## 3 Mon       2834
## 4 Tue       2825
## 5 Sat       2816
## 6 Fri       2783
## 7 Thu       2771
```

```
regular_customers %>% count(tran_day) %>%
  ggplot(aes(reorder(x = tran_day, -n), y = n, fill = tran_day)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(legend.position = "none") +
  scale_y_continuous("Count",
    breaks = seq(0, 3000, by = 100),
    limits = c(0, 3000)
  ) +
  labs(title = "REGULAR CUSTOMERS-DAY OF WEEK RECORDS", x = "")
```




```
regular_customers %>% count(tran_day) %>%
  ggplot(aes(x = tran_day, y = n, group = 1)) +
  geom_point() +
```

```r
  geom_line(colour = "blue") +
  theme(legend.position = "none") +
  scale_y_continuous("Count",
                     breaks = seq(0, 3000, by = 100),
                     limits = c(0, 3000)) +
  labs(title = "REGULAR CUSTOMERS-DAY OF WEEK RECORDS", x = "")
```



The difference across the day of week with highest sales on Wednesday and the lowest sales on Thursday is 112.

From the data on Monthly and Daily Sales numbers we get say that there wasn't a signifcant difference in the customer visits.

Online Order

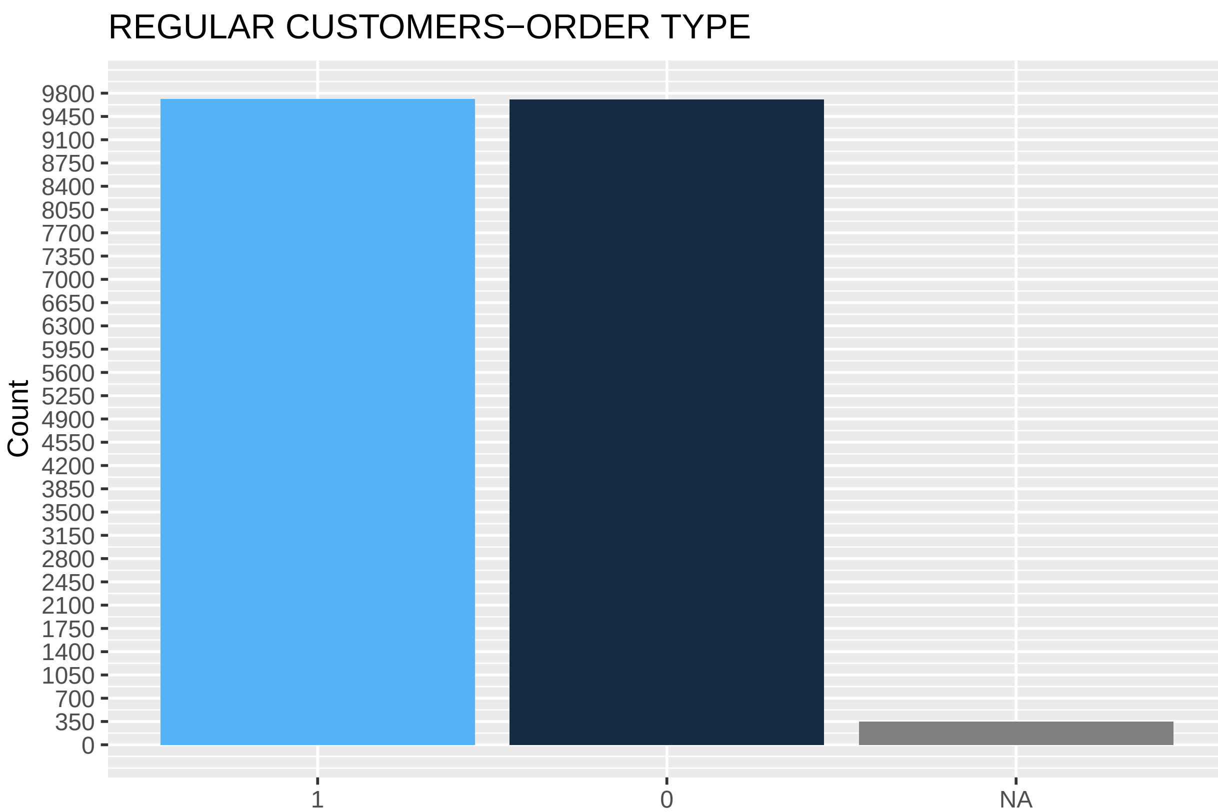
Taking 0 to mean physical order and 1 to mean online order

```r
regular_customers %>% count(online_order, sort = T)
```

```
## # A tibble: 3 x 2
##   online_order     n
##          <dbl> <int>
```

```
## 1             1  9714
## 2             0  9706
## 3            NA   352
```

```
regular_customers %>% count(online_order) %>%
  ggplot(aes(reorder(x = online_order, -n), y = n, fill = online_order)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(legend.position = "none") +
  scale_y_continuous("Count",
    breaks = seq(0, 9800, by = 350),
    limits = c(0, 9800)
  ) +
  labs(title = "REGULAR CUSTOMERS-ORDER TYPE", x = "")
```



Make online a factor with 3 levels.

```
regular_customers$online_order[is.na(regular_customers$online_order)] <- "NOT PROVIDED"
regular_customers$online_order <- as.factor(
  as.character(regular_customers$online_order))
class(regular_customers$online_order)
```

```
## [1] "factor"
```

The difference between order type 0 and 1 was only 8 transactions.

The Regular Customers Brand

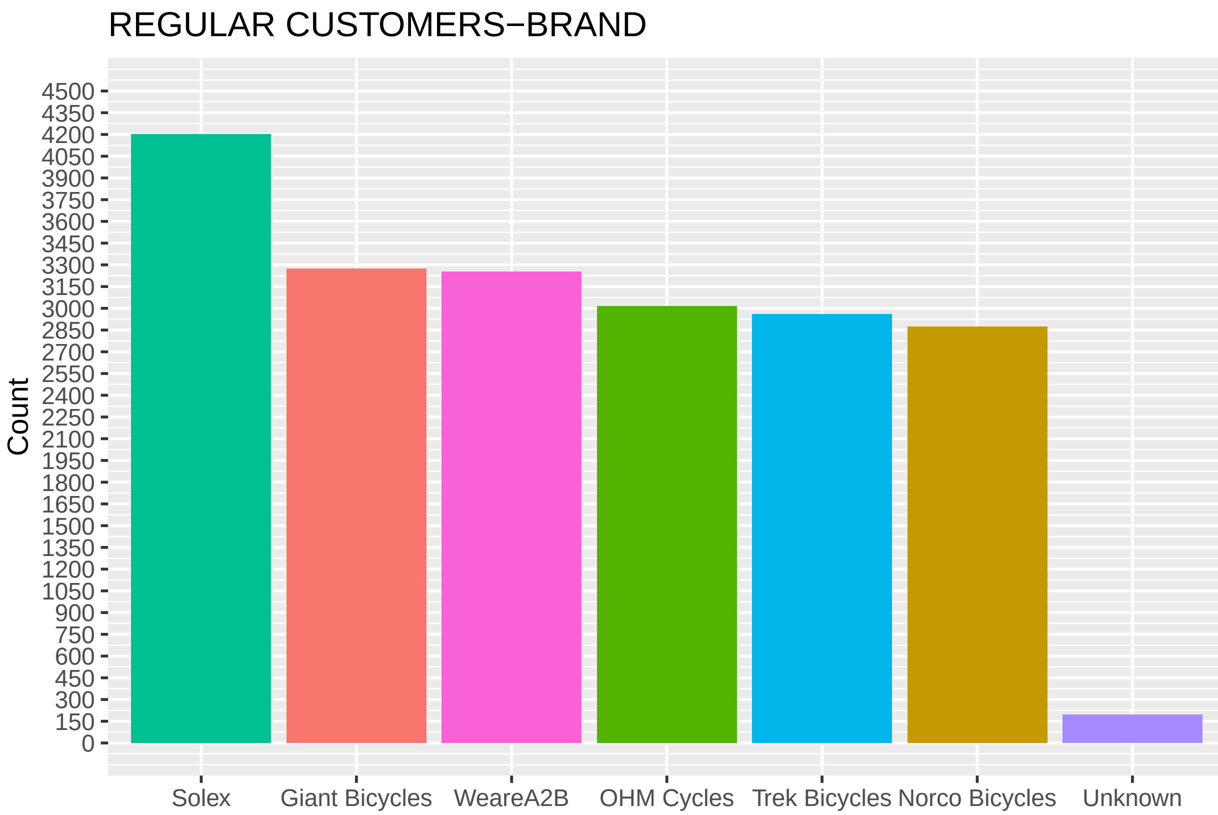
```r
regular_customers %>% count(brand, sort = T)
```

```
## # A tibble: 7 x 2
##   brand              n
##   <chr>          <int>
## 1 Solex           4200
## 2 Giant Bicycles  3274
## 3 WeareA2B        3254
## 4 OHM Cycles      3013
## 5 Trek Bicycles   2961
## 6 Norco Bicycles  2874
## 7 <NA>             196
```

Make NAs to Unknown

```r
regular_customers$brand[is.na(regular_customers$brand)] <- "Unknown"
```

```r
regular_customers %>% count(brand) %>%
  ggplot(aes(reorder(x = brand, -n), y = n, fill = brand)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(legend.position = "none") +
  scale_y_continuous("Count",
    breaks = seq(0, 4500, by = 150),
    limits = c(0, 4500)
  ) +
  labs(title = "REGULAR CUSTOMERS-BRAND", x = "")
```

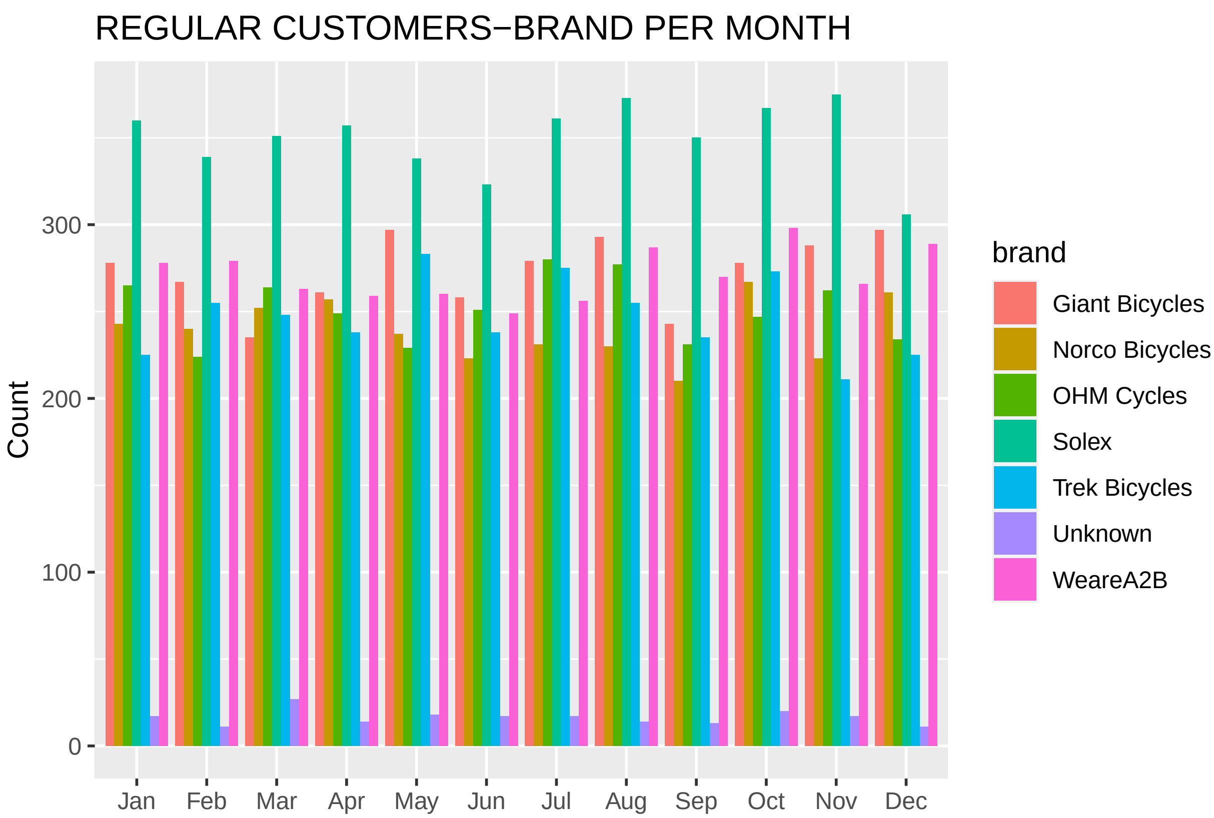


Brand Month

```r
regular_brand_month <- regular_customers %>% group_by(tran_month) %>% count(brand)
regular_brand_month
```

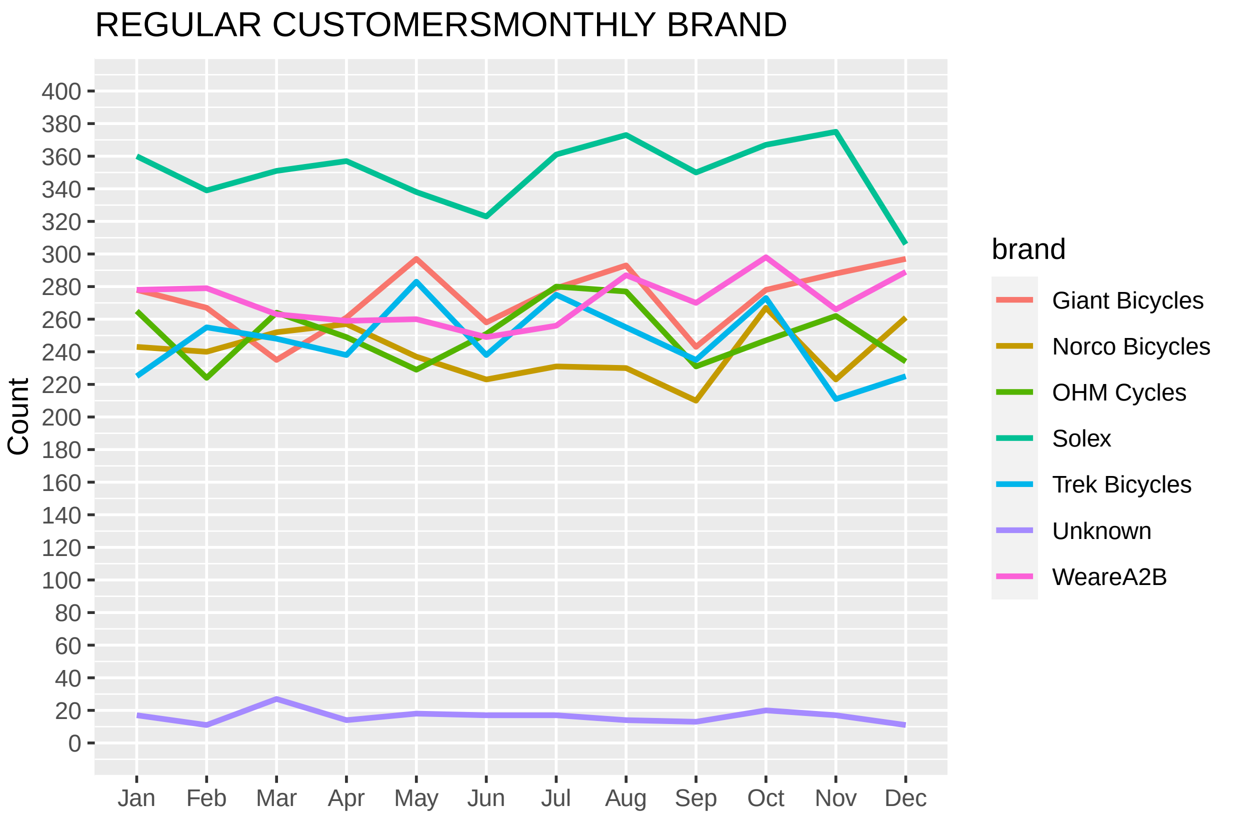
```
## # A tibble: 84 x 3
## # Groups:   tran_month [12]
##    tran_month brand              n
##    <ord>      <chr>          <int>
##  1 Jan        Giant Bicycles   278
##  2 Jan        Norco Bicycles   243
##  3 Jan        OHM Cycles       265
##  4 Jan        Solex            360
##  5 Jan        Trek Bicycles    225
##  6 Jan        Unknown           17
##  7 Jan        WeareA2B         278
##  8 Feb        Giant Bicycles   267
##  9 Feb        Norco Bicycles   240
## 10 Feb        OHM Cycles       224
## # i 74 more rows
```

```
ggplot(regular_brand_month, aes(tran_month, n,
                                            fill = brand)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "REGULAR CUSTOMERS-BRAND PER MONTH", x = "",
       y = "Count")
```



Solex were the preffered brand. The difference between the most purchased brand and the least purchased was 1326. A significant difference.

```
ggplot(regular_brand_month,
       aes(tran_month, n, colour = brand, group = brand)) +
  geom_line(linewidth = 1) +
  scale_y_continuous("Count",
    breaks = seq(0, 400, by = 20),
    limits = c(0, 400)
  ) +
  labs(title = "REGULAR CUSTOMERSMONTHLY BRAND", x = "")
```



Clearly brands sales differed across months. June and September was low for all brands. Solex brand was distinctively apart.

The Regular Customers Product line

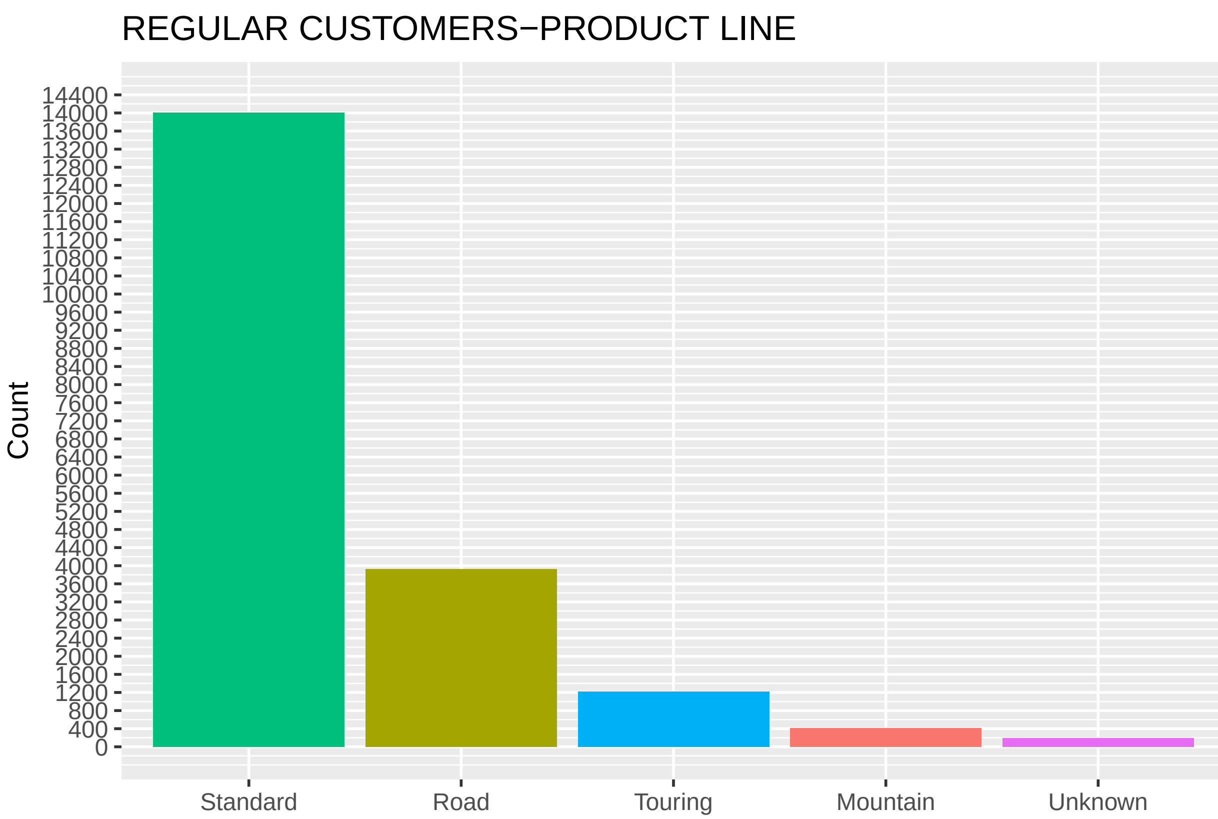
```
regular_customers %>% count(product_line, sort = T)
```

```
## # A tibble: 5 x 2
##   product_line     n
##   <chr>        <int>
## 1 Standard     14014
## 2 Road          3921
## 3 Touring       1222
## 4 Mountain       419
## 5 <NA>           196
```

Make NAs to Unknown

```
regular_customers$product_line[is.na(regular_customers$product_line)] <- "Unknown"
```

```r
regular_customers %>% count(product_line) %>%
  ggplot(aes(reorder(x = product_line, -n), y = n, fill = product_line)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(legend.position = "none") +
  scale_y_continuous("Count",
    breaks = seq(0, 14400, by = 400),
    limits = c(0, 14400)
  ) +
  labs(title = "REGULAR CUSTOMERS-PRODUCT LINE", x = "")
```



By far the Standard Product line was preffered. The least preffered was the Mountain product line.

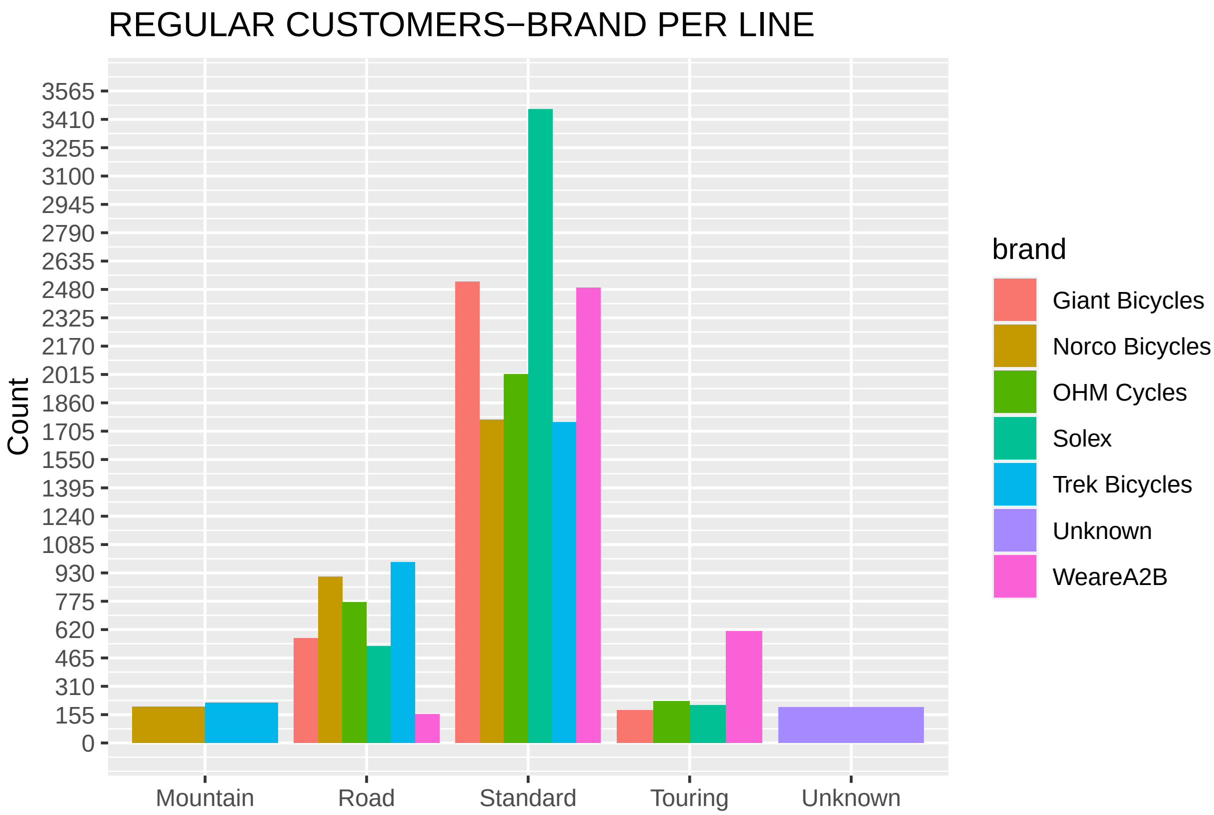
Product Line and Brand

```r
regular_line_brand <- regular_customers %>% group_by(product_line) %>% count(brand)
regular_line_brand
```

```
## # A tibble: 19 x 3
## # Groups:   product_line [5]
##    product_line brand              n
##    <chr>        <chr>          <int>
```

```
##   1 Mountain     Norco Bicycles   198
##   2 Mountain     Trek Bicycles    221
##   3 Road         Giant Bicycles   573
##   4 Road         Norco Bicycles   908
##   5 Road         OHM Cycles       769
##   6 Road         Solex            528
##   7 Road         Trek Bicycles    987
##   8 Road         WeareA2B         156
##   9 Standard     Giant Bicycles  2523
##  10 Standard     Norco Bicycles  1768
##  11 Standard     OHM Cycles      2016
##  12 Standard     Solex           3466
##  13 Standard     Trek Bicycles   1753
##  14 Standard     WeareA2B        2488
##  15 Touring      Giant Bicycles   178
##  16 Touring      OHM Cycles       228
##  17 Touring      Solex            206
##  18 Touring      WeareA2B         610
##  19 Unknown      Unknown          196
```

```
ggplot(regular_line_brand, aes(product_line, n, fill = brand)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 3565, by = 155),
    limits = c(0, 3565)
  ) +
  labs(title = "REGULAR CUSTOMERS-BRAND PER LINE", x = "")
```



Standard line is preffered by all brands. For the Mountain line we have only Norco Bicycles and Trek Bicycles.

The Regular Customers Product Class

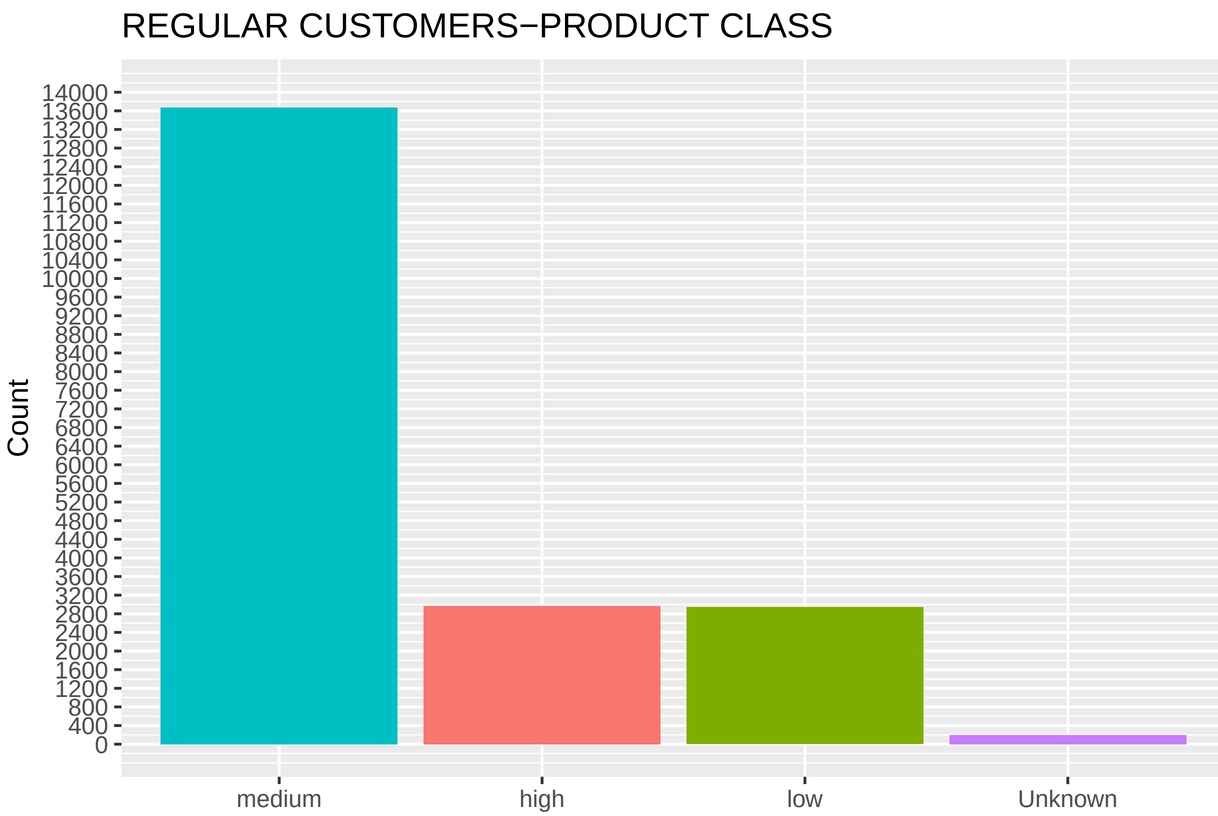
```
regular_customers %>% count(product_class, sort = T)
```

```
## # A tibble: 4 x 2
##   product_class     n
##   <chr>         <int>
## 1 medium        13668
## 2 high           2967
## 3 low            2941
## 4 <NA>            196
```

Make NAs to Unknown

```
regular_customers$product_class[is.na(regular_customers$product_class)] <- "Unknown"
```

```
regular_customers %>% count(product_class) %>%
  ggplot(aes(reorder(x = product_class, -n), y = n, fill = product_class)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(legend.position = "none") +
  scale_y_continuous("Count",
    breaks = seq(0, 14000, by = 400),
    limits = c(0, 14000)) +
  labs(title = "REGULAR CUSTOMERS-PRODUCT CLASS", x = "")
```



Medium Class was the most transacted. High and Low class were almost equal.

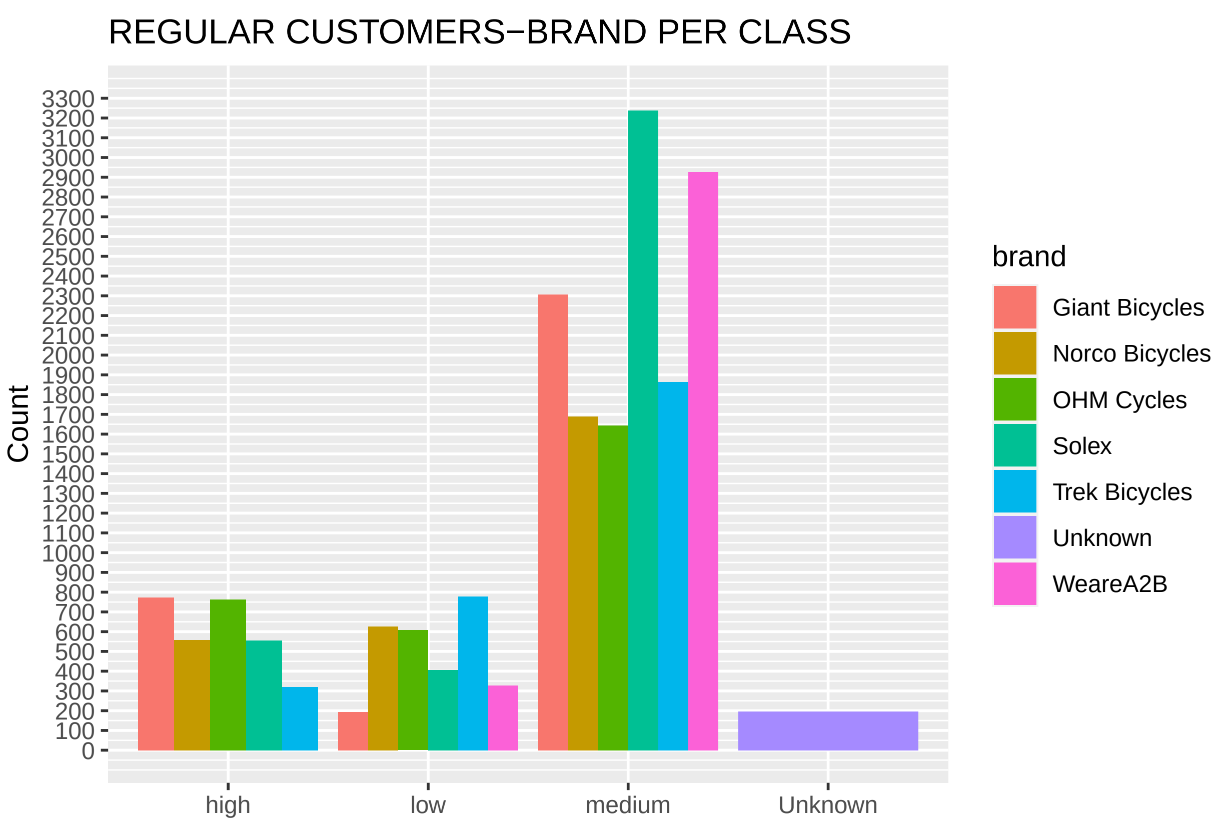
Product Class and Brand

```
regular_line_brand_class <- regular_customers %>% group_by(product_class) %>% count(brand)
regular_line_brand_class
```

```
## # A tibble: 18 x 3
## # Groups:   product_class [4]
##    product_class brand              n
##    <chr>         <chr>          <int>
##  1 Unknown       Unknown          196
##  2 high          Giant Bicycles   773
```

```
##   3 high          Norco Bicycles   557
##   4 high          OHM Cycles       762
##   5 high          Solex            556
##   6 high          Trek Bicycles    319
##   7 low           Giant Bicycles   194
##   8 low           Norco Bicycles   627
##   9 low           OHM Cycles       607
##  10 low           Solex            407
##  11 low           Trek Bicycles    779
##  12 low           WeareA2B         327
##  13 medium        Giant Bicycles  2307
##  14 medium        Norco Bicycles  1690
##  15 medium        OHM Cycles      1644
##  16 medium        Solex           3237
##  17 medium        Trek Bicycles   1863
##  18 medium        WeareA2B        2927
```

```
ggplot(regular_line_brand_class, aes(product_class, n, fill = brand)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 3300, by = 100),
    limits = c(0, 3300)
  ) +
  labs(title = "REGULAR CUSTOMERS-BRAND PER CLASS", x = "")
```



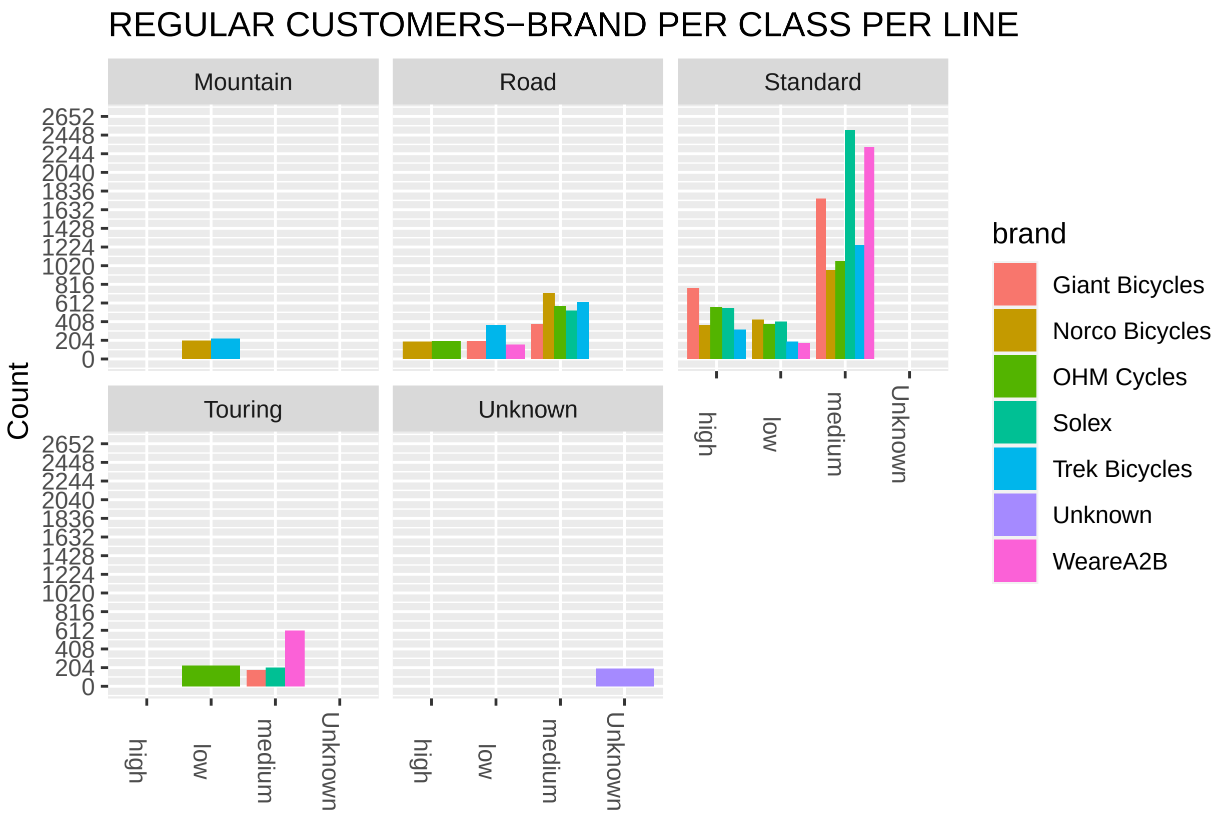
Brands in medium class were the most sold.

Product Class and Brand

```
regular_line_brand_class_line <- regular_customers %>%
  group_by(product_class, product_line) %>% count(brand)
regular_line_brand_class_line
```

```
## # A tibble: 33 x 4
## # Groups:   product_class, product_line [10]
##    product_class product_line brand              n
##    <chr>         <chr>        <chr>          <int>
##  1 Unknown       Unknown      Unknown          196
##  2 high          Road         Norco Bicycles   187
##  3 high          Road         OHM Cycles       194
##  4 high          Standard     Giant Bicycles   773
##  5 high          Standard     Norco Bicycles   370
##  6 high          Standard     OHM Cycles       568
##  7 high          Standard     Solex            556
##  8 high          Standard     Trek Bicycles    319
##  9 low           Mountain     Norco Bicycles   198
## 10 low           Mountain     Trek Bicycles    221
## # i 23 more rows
```

```
ggplot(regular_line_brand_class_line,
       aes(product_class, n, fill = brand)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 2652, by = 204),
    limits = c(0, 2652)) +
  theme(axis.text.x = element_text(angle = -90)) +
  facet_wrap(~product_line) +
  labs(title = "REGULAR CUSTOMERS-BRAND PER CLASS PER LINE", x = "")
```



Mountain line had only low class with two brands.

Touring line did not have high class.

Road and Standard lines had all the classes.

Standard line had all brands in all classes.

Standard line medium class are the most preffered.

The Regular Customers Product Size

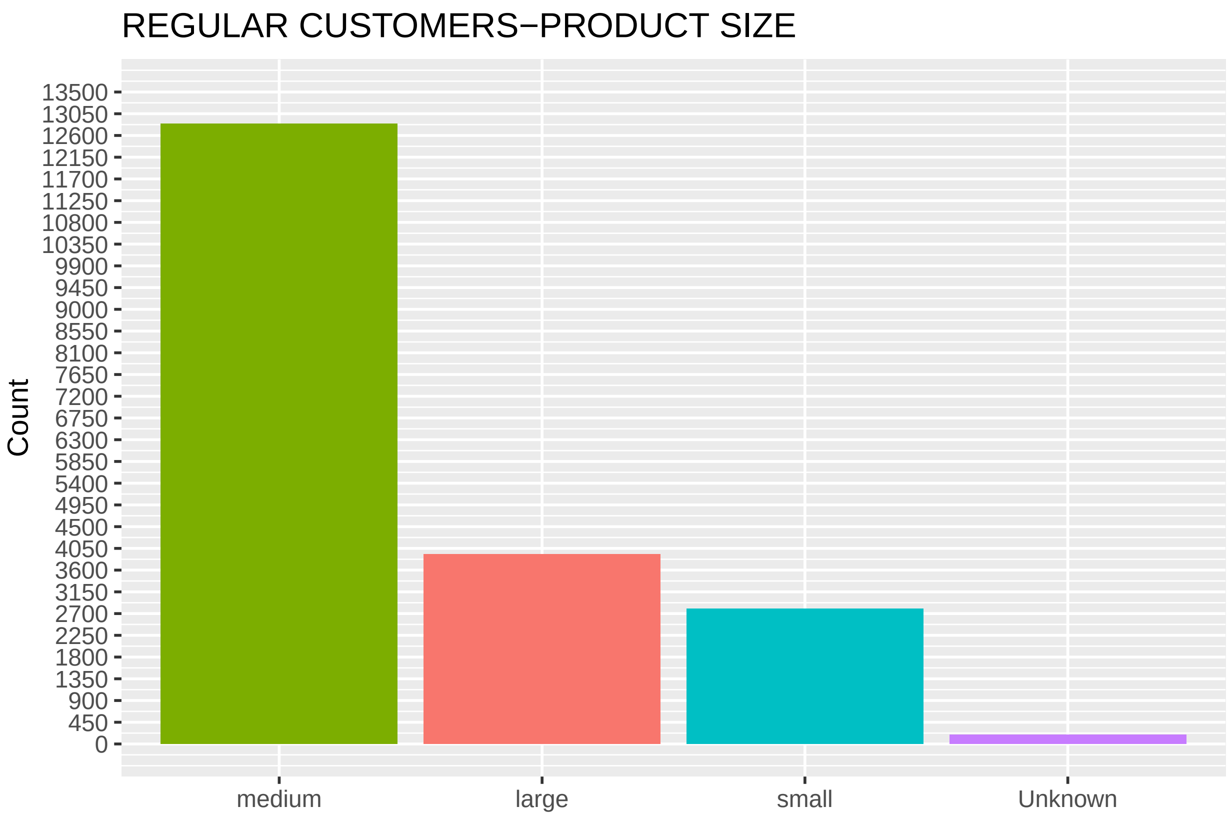
```
regular_customers %>% count(product_size, sort = T)
```

```
## # A tibble: 4 x 2
##   product_size     n
##   <chr>        <int>
## 1 medium       12844
## 2 large         3928
## 3 small         2804
## 4 <NA>           196
```

Make NAs to Unknown

```
regular_customers$product_size[is.na(regular_customers$product_size)] <- "Unknown"
```

```
regular_customers %>% count(product_size) %>%
  ggplot(aes(reorder(x = product_size, -n), y = n,
             fill = product_size)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(legend.position = "none") +
  scale_y_continuous("Count",
    breaks = seq(0, 13500, by = 450),
    limits = c(0, 13500)) +
  labs(title = "REGULAR CUSTOMERS-PRODUCT SIZE", x = "")
```



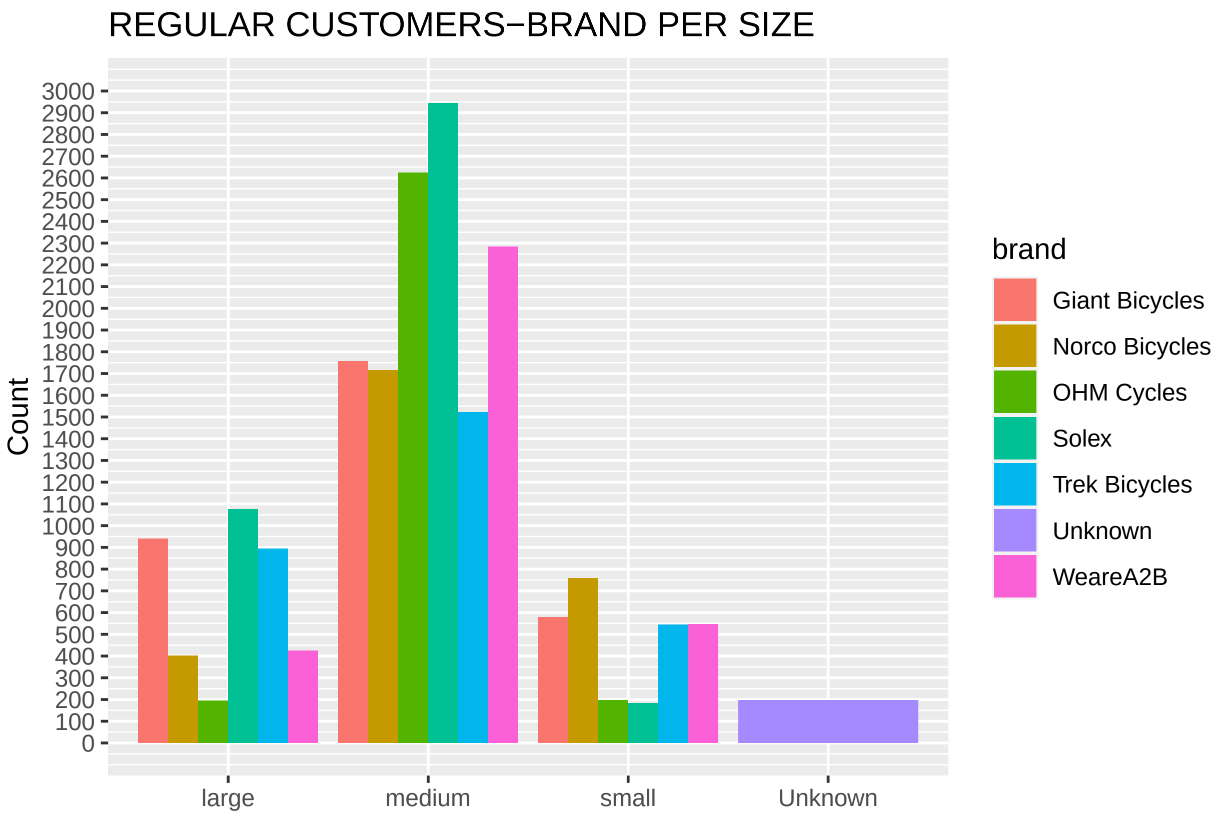
Medium Size was the most transacted.

Product Size and Brand

```
regular_size_brand <- regular_customers %>% group_by(product_size) %>% count(brand)
regular_size_brand
```

```
## # A tibble: 19 x 3
## # Groups:   product_size [4]
##    product_size brand               n
##    <chr>        <chr>           <int>
##  1 Unknown      Unknown           196
##  2 large        Giant Bicycles    939
##  3 large        Norco Bicycles    402
##  4 large        OHM Cycles        194
##  5 large        Solex            1075
##  6 large        Trek Bicycles     894
##  7 large        WeareA2B          424
##  8 medium       Giant Bicycles   1757
##  9 medium       Norco Bicycles   1715
## 10 medium       OHM Cycles       2623
## 11 medium       Solex            2943
## 12 medium       Trek Bicycles    1523
## 13 medium       WeareA2B         2283
## 14 small        Giant Bicycles    578
## 15 small        Norco Bicycles    757
## 16 small        OHM Cycles        196
## 17 small        Solex             182
## 18 small        Trek Bicycles     544
## 19 small        WeareA2B          547
```

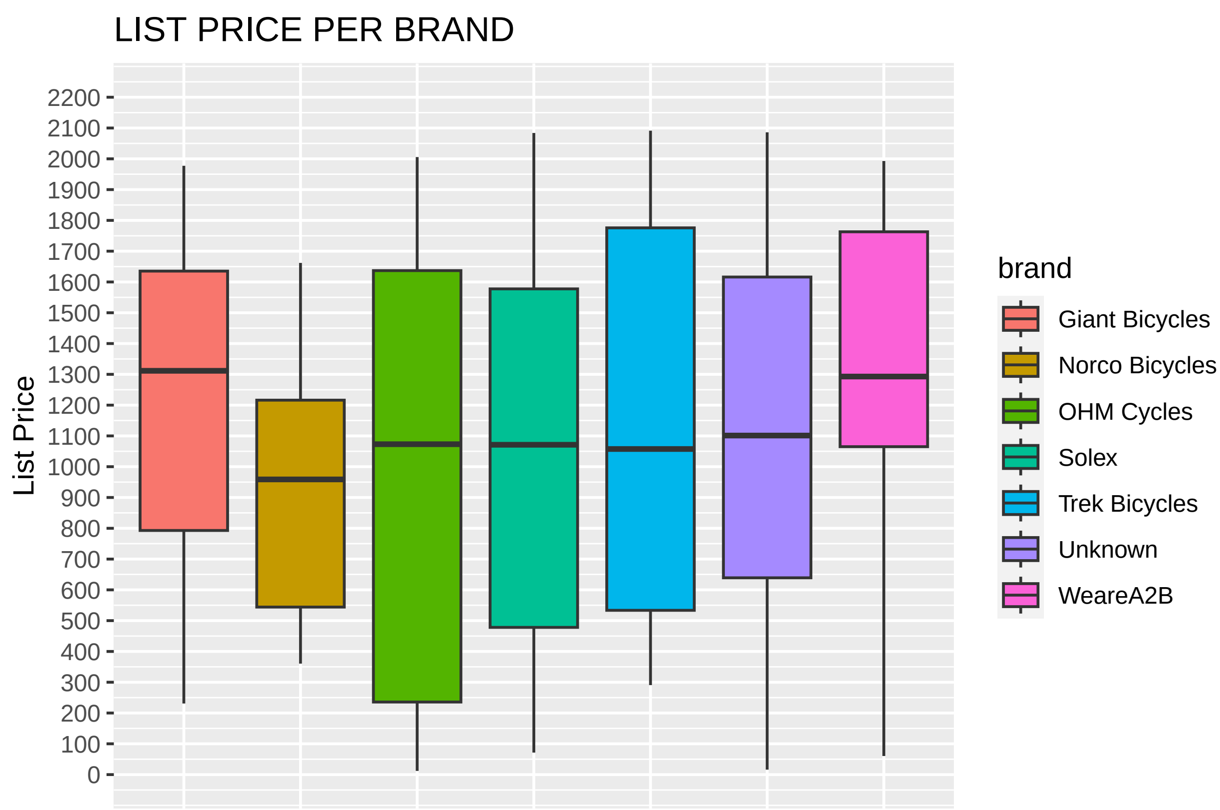
```r
ggplot(regular_size_brand, aes(product_size, n, fill = brand)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 3000, by = 100),
    limits = c(0, 3000)
  ) +
  labs(title = "REGULAR CUSTOMERS-BRAND PER SIZE", x = "")
```



All brands prefferred medium size.

Comparing Large size and small size, we get that for Giant bycycles large was preferred, for Norco Bicycles small was preferred, for OHM Cycles the difference was small but small was preffered, for Solex and Trek large was preferred and for WeareA2B Small are preferred.

```
ggplot(regular_customers, aes(brand, list_price, fill = brand)) +
  geom_boxplot() +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) +
  scale_y_continuous("List Price",
    breaks = seq(0, 2200, by = 100),
    limits = c(0, 2200)) +
  labs(title = "LIST PRICE PER BRAND", x = "")
```

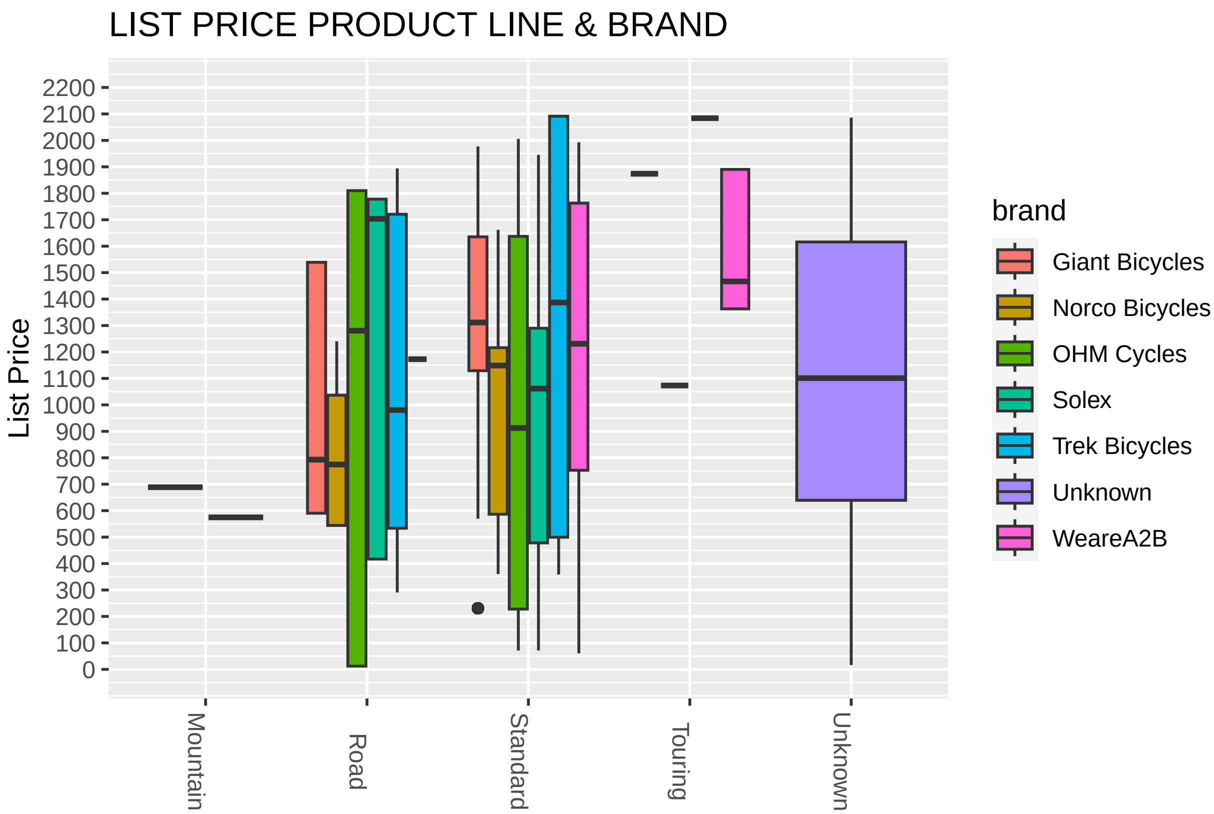


We can see that Narco Bicycles List Price are not so different for the different customers.

List Price for OHM Cycles are so different for the different categories.

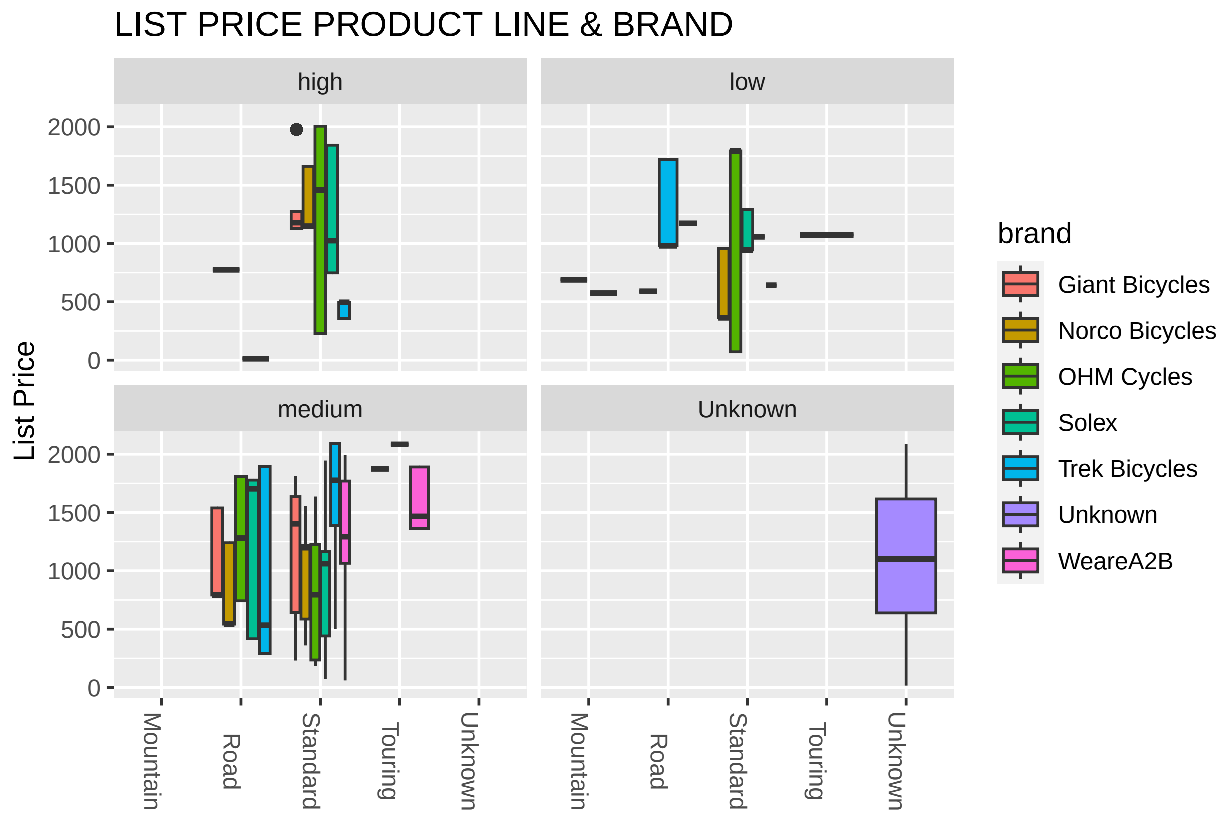
The median values for OHM Cycles, Solex and Trek Bicycles were on the same level but the distributions were all different.

```r
ggplot(regular_customers, aes(product_line, list_price, fill = brand)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = -90)) +
  scale_y_continuous("List Price",
    breaks = seq(0, 2200, by = 100),
    limits = c(0, 2200)) +
  labs(title = "LIST PRICE PRODUCT LINE & BRAND", x = "")
```



List Price differed per brand per product line.

```
ggplot(regular_customers, aes(product_line, list_price, fill = brand)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = -90)) +
  facet_wrap(~product_class) +
  labs(title = "LIST PRICE PRODUCT LINE & BRAND",
       x = "", y = "List Price")
```



Create a column that has the difference between list Price and Standard Cost and replace NAs with 0

```
regular_customers <- regular_customers %>%
  mutate(price_diff = list_price - standard_cost)
regular_customers <- regular_customers %>%
  select(1:13, 16, 14:15)
regular_customers$price_diff[is.na(regular_customers$price_diff)] <- 0
```

There were no instances where the standard_cost was greater than list_price

```
regular_customers %>% filter(standard_cost > list_price)
```

```
## # A tibble: 0 x 16
## # i 16 variables: tran_id <dbl>, product_id <fct>, customer_id <fct>,
## #   tran_date <date>, tran_month <ord>, tran_day <ord>, online_order <fct>,
## #   order_status <chr>, brand <chr>, product_line <chr>, product_class <chr>,
## #   product_size <chr>, list_price <dbl>, price_diff <dbl>,
## #   standard_cost <dbl>, first_sold_date <date>
```

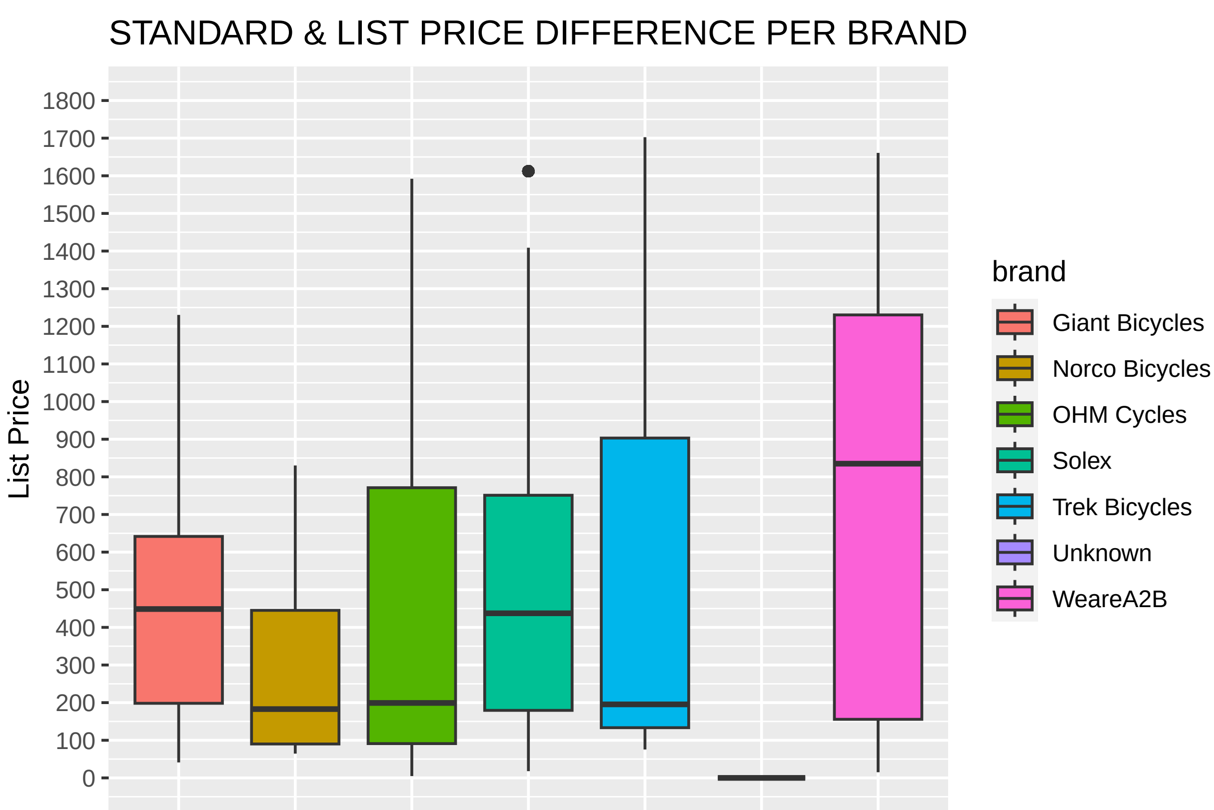
There were no instances where the list_price and standard_cost were equal

```
regular_customers %>% filter(standard_cost == list_price)
```

```
## # A tibble: 0 x 16
## # i 16 variables: tran_id <dbl>, product_id <fct>, customer_id <fct>,
## #   tran_date <date>, tran_month <ord>, tran_day <ord>, online_order <fct>,
## #   order_status <chr>, brand <chr>, product_line <chr>, product_class <chr>,
## #   product_size <chr>, list_price <dbl>, price_diff <dbl>,
## #   standard_cost <dbl>, first_sold_date <date>
```

Instances where the difference between list was 0 were instances where the Standard cost was NAs, which were the same 196 transactions that had similar missing values across columns.

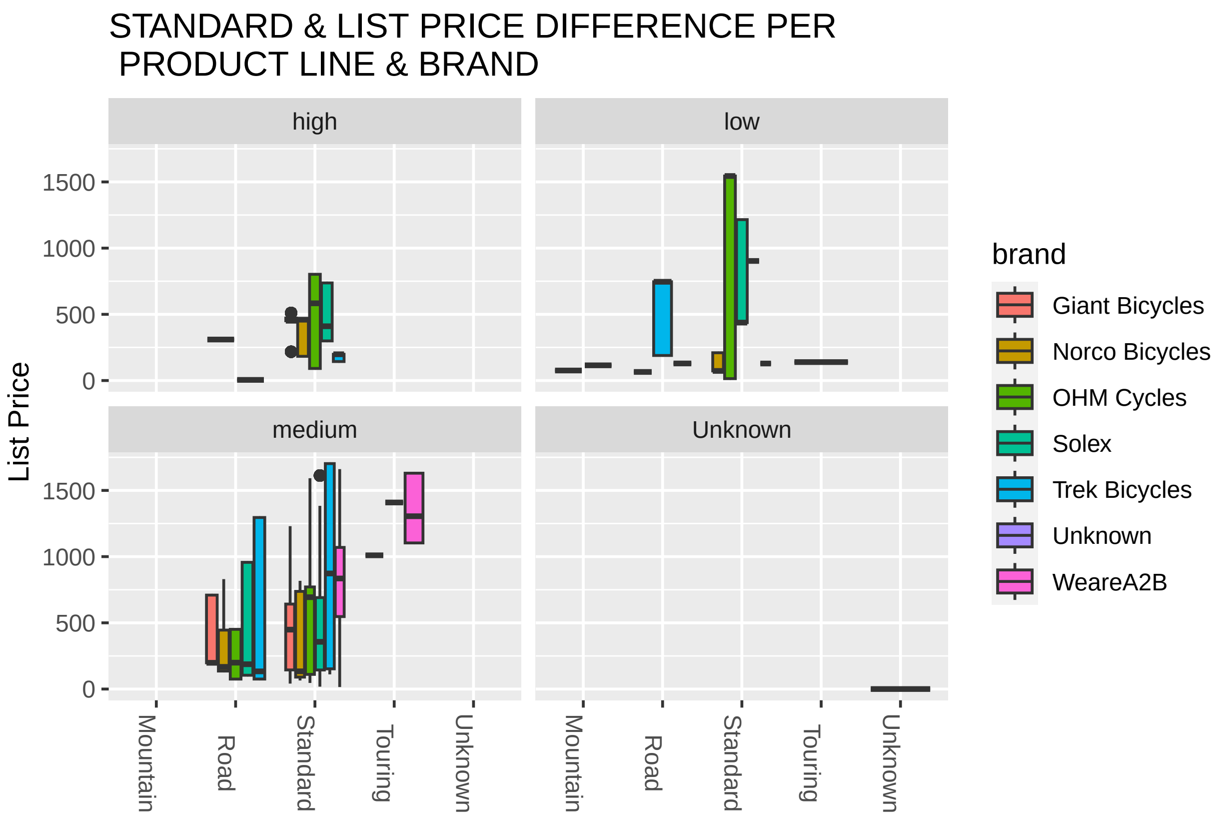
```
ggplot(regular_customers, aes(brand, price_diff, fill = brand)) +
  geom_boxplot() +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) +
  scale_y_continuous("List Price",
    breaks = seq(0, 1800, by = 100),
    limits = c(0, 1800)) +
  labs(title = "STANDARD & LIST PRICE DIFFERENCE PER BRAND", x = "")
```



We can see that for the Solex brand we had a difference between the list price and standard cost that was an outlier, that is the difference was way above.

Generally it can be said that the differences between the list price and standard cost across the different brands was substantial.

```
ggplot(regular_customers, aes(product_line, price_diff, fill = brand)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = -90)) +
  facet_wrap(~product_class) +
  labs(title = "STANDARD & LIST PRICE DIFFERENCE PER \n PRODUCT LINE & BRAND",
       x = "", y = "List Price")
```



The differences were substantial in Medium line for the standard and Road classes.

Create a column for the number of days between the transactions of a customer.

```
regular_customers_1 <- regular_customers
regular_customers_1 <- regular_customers_1 %>%
  arrange(customer_id, tran_date) %>% group_by(customer_id) %>%
  mutate(d = tran_date - lag(tran_date),
                   days_diff = as.numeric(d, units = 'days'))
```

```r
regular_customers_1 <- regular_customers_1 %>% select(-d)
regular_customers_1 <- regular_customers_1 %>% select(1:4, 17, 5:16)
head(regular_customers_1, 10)
```

```
## # A tibble: 10 x 17
## # Groups:   customer_id [1]
##    tran_id product_id customer_id tran_date  days_diff tran_month tran_day
##      <dbl> <fct>      <fct>       <date>         <dbl> <ord>      <ord>
##  1    9785 72         1           2017-01-05        NA Jan        Thu
##  2   13424 2          1           2017-02-21        47 Feb        Tue
##  3   14486 23         1           2017-03-27        34 Mar        Mon
##  4   18970 11         1           2017-03-29         2 Mar        Wed
##  5    3765 38         1           2017-04-06         8 Apr        Thu
##  6    5157 47         1           2017-05-11        35 May        Thu
##  7   13644 25         1           2017-05-19         8 May        Fri
##  8   15663 32         1           2017-06-04        16 Jun        Sun
##  9   16423 9          1           2017-12-09       188 Dec        Sat
## 10   14931 31         1           2017-12-14         5 Dec        Thu
## # i 10 more variables: online_order <fct>, order_status <chr>, brand <chr>,
## #   product_line <chr>, product_class <chr>, product_size <chr>,
## #   list_price <dbl>, price_diff <dbl>, standard_cost <dbl>,
## #   first_sold_date <date>
```

2 Customer Demographics data

```r
names(demographic)
```

```
##  [1] "customer_id"                         "first_name"
##  [3] "last_name"                           "gender"
##  [5] "past_3_years_bike_related_purchases" "DOB"
##  [7] "age"                                 "job_industry_category"
##  [9] "wealth_segment"                      "deceased_indicator"
## [11] "owns_car"                            "tenure"
```

Rename some columns names

```r
demographic_1 <- demographic ## create duplicate
demographic_1 <- demographic_1 %>%
  rename(past_purchases = past_3_years_bike_related_purchases,
         job_industry = job_industry_category,
         deceased = deceased_indicator,
         dob = DOB)
```

Since these purchases are all bikes related and the column that has the data is for the past 3 years, we remove those spefications from the column name.

Unite first name and last name

First we replace the NAs with empty in the columns

```
demographic_1$first_name[is.na(demographic_1$first_name)] <- ""
demographic_1$last_name[is.na(demographic_1$last_name)] <- ""
demographic_1 <- unite(demographic_1, customer_name, first_name:last_name, sep = " ")
demographic_1$customer_name <- str_squish(demographic_1$customer_name)
```

column_id

```
class(demographic_1$customer_id)
```

```
## [1] "numeric"
```

```
demographic_1$customer_id <- as.factor(as.numeric(demographic_1$customer_id))
```

We join the regular transactions data with the demographics data and keep only the values that are in both.

```
trans_demographic <- as.data.frame(regular_customers_1 %>%
  inner_join(demographic_1, by = "customer_id"))
dim(trans_demographic)
```

```
## [1] 19769    27
```

```
class(trans_demographic)
```

```
## [1] "data.frame"
```

Only demographic data of 3 customers present in the regular customers data was not present in the demographic data.

What was the gender of the regular customers

```r
class(trans_demographic$gender)
```

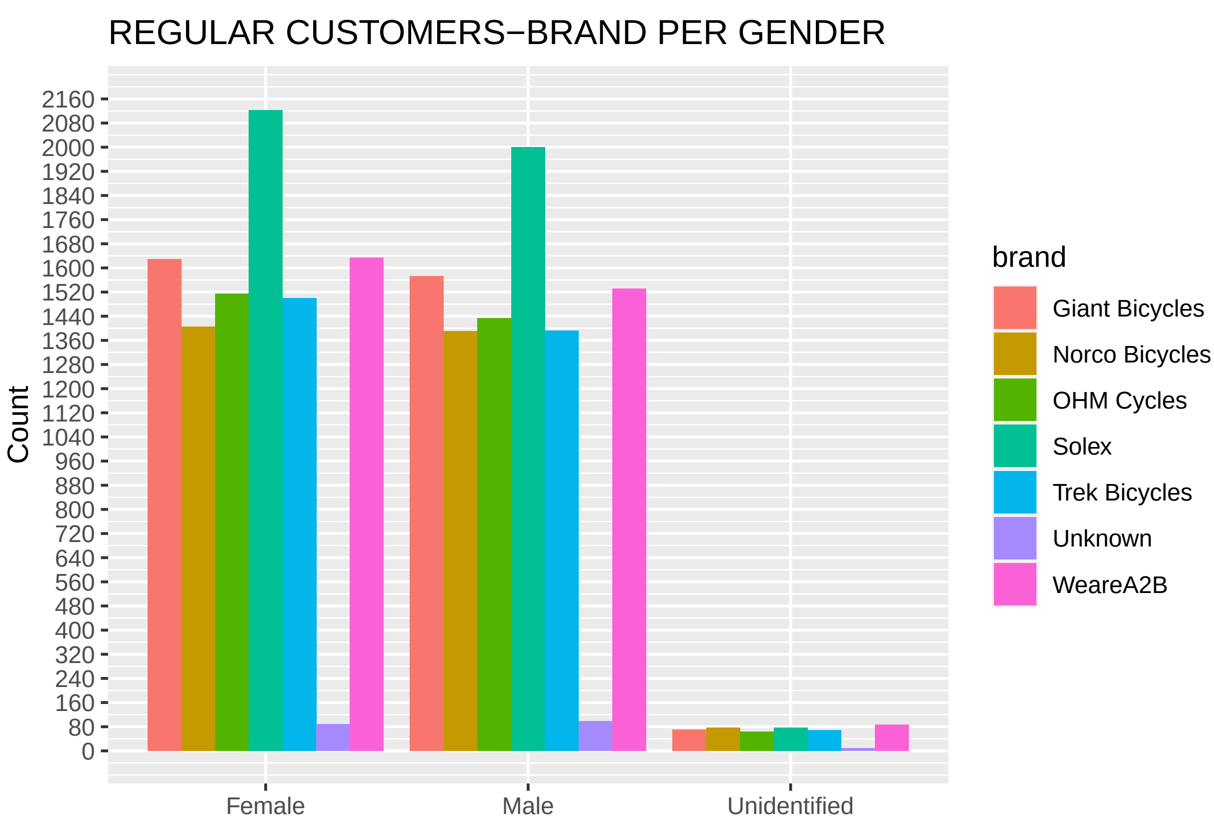
```
## [1] "character"
```

```r
trans_demographic %>% count(gender, sort = T)
```

```
##          gender    n
## 1        Female 9897
## 2          Male 9419
## 3 Unidentified  453
```

Brand and Gender

```r
brand_gender <- trans_demographic %>% group_by(gender) %>% count(brand)
ggplot(brand_gender, aes(gender, n, fill = brand)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 2160, by = 80),
    limits = c(0, 2160)) +
  labs(title = "REGULAR CUSTOMERS-BRAND PER GENDER", x = "")
```



Males and Females preffered Solex brands.

Brand prefference was the same within genders.

Age

```
summary(trans_demographic$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   21.00   36.00   46.00   45.61   55.00   91.00     444
```

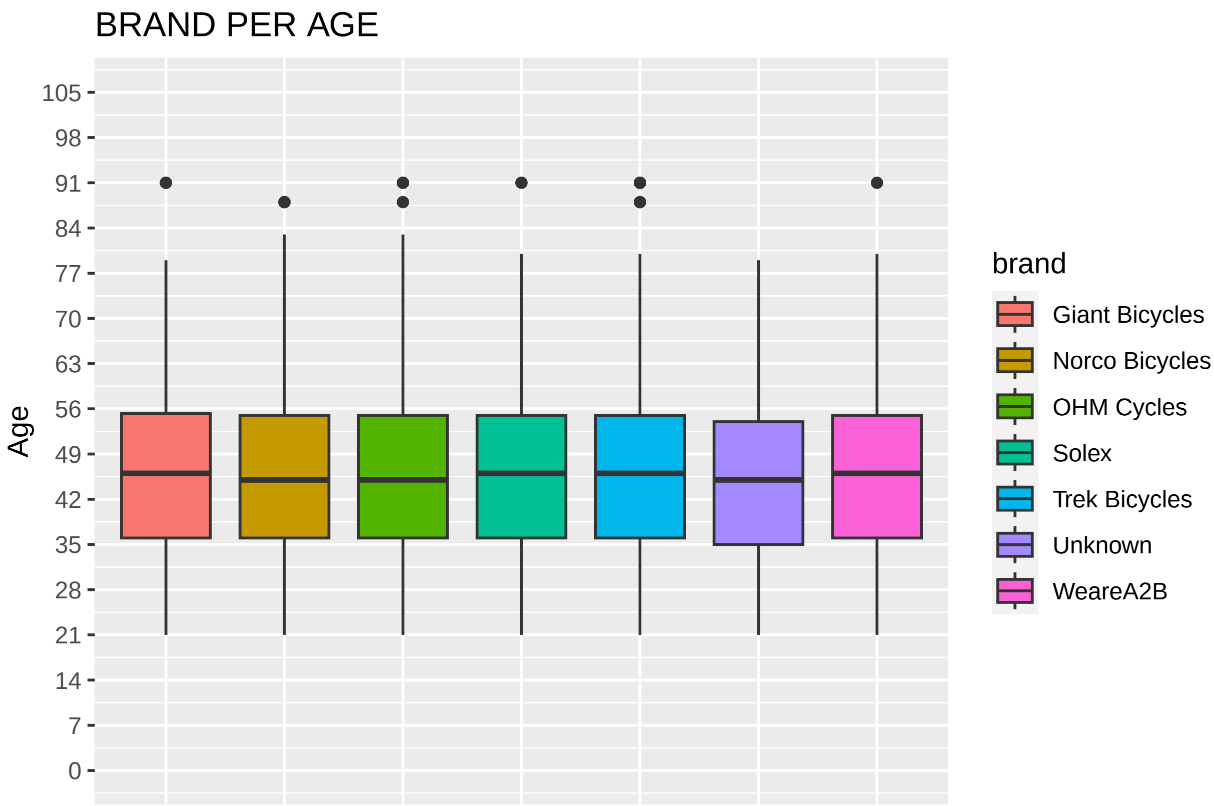
We have NAs for the age.

```
trans_demographic %>% group_by(brand) %>%
  summarise(average = mean(age, na.rm = TRUE)) %>% arrange(desc(average))
```

```
## # A tibble: 7 x 2
##   brand          average
##   <chr>            <dbl>
## 1 Giant Bicycles    46.0
## 2 WeareA2B          45.7
## 3 Trek Bicycles     45.6
## 4 OHM Cycles        45.5
## 5 Norco Bicycles    45.5
## 6 Solex             45.4
## 7 Unknown           45.1
```

After removing NAs we get that we have an almost same average of customers.

```
ggplot(trans_demographic, aes(brand, age, fill = brand)) +
  geom_boxplot() +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) +
  scale_y_continuous("Age",
    breaks = seq(0, 105, by = 7),
    limits = c(0, 105)) +
  labs(title = "BRAND PER AGE", x = "")
```



The minimum age was 21 for all brands.

Other than the outliers, the distribution of age across the brands was almost the same.

We can say that age did not create a significant difference.

Past Related Purchases

```
class(trans_demographic$past_purchases)
```

```
## [1] "numeric"
```

```
summary(trans_demographic$past_purchases)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   24.00   48.00   48.79   73.00   99.00
```

```
trans_demographic %>% group_by(brand) %>%
  summarise(average = mean(past_purchases)) %>% arrange(desc(average))
```

```
## # A tibble: 7 x 2
##   brand           average
##   <chr>             <dbl>
## 1 Solex              49.3
## 2 WeareA2B           49.3
## 3 Giant Bicycles     48.8
## 4 Trek Bicycles      48.6
## 5 OHM Cycles         48.5
## 6 Norco Bicycles     48.2
## 7 Unknown            46.1
```

```r
ggplot(trans_demographic, aes(brand, past_purchases, fill = brand)) +
  geom_boxplot() +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) +
  scale_y_continuous("PAST PURCHASES",
    breaks = seq(0, 105, by = 7),
    limits = c(0, 105)) +
  labs(title = "BRAND PER PAST PURCHASES", x = "")
```

It seems like customers past 3 years bike related purchases did not affect the brand prefference.

Job Industry

Shorten some job industry names

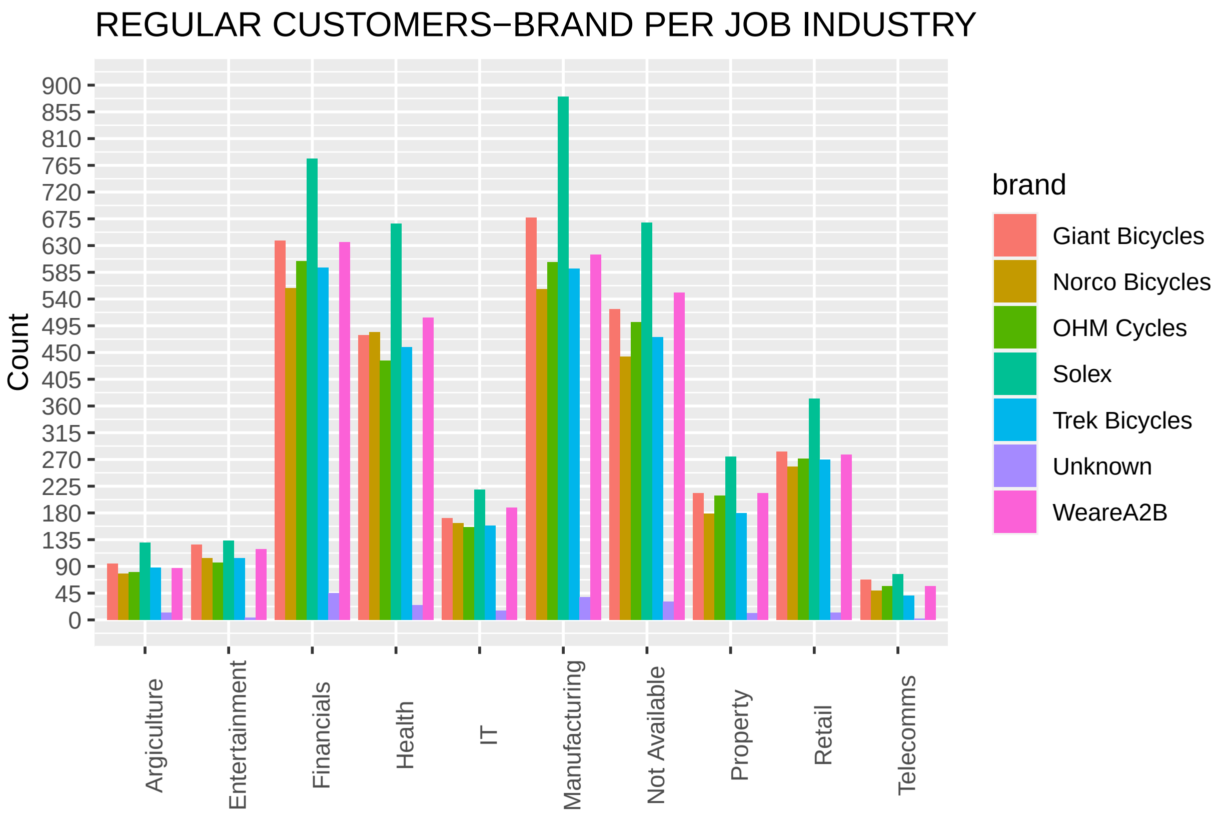
```
trans_demographic <- trans_demographic %>%
    mutate(job_industry = case_when(
        str_detect(job_industry, "Financial") ~ "Financials",
        str_detect(job_industry, "Telecomm") ~ "Telecomms",
        TRUE ~ job_industry
    ))
```

```
trans_demographic %>% count(job_industry, sort = T)
```

```
##     job_industry    n
## 1  Manufacturing 3961
## 2     Financials 3850
## 3  Not Available 3194
## 4         Health 3059
## 5         Retail 1744
## 6       Property 1280
## 7             IT 1073
## 8  Entertainment  687
## 9    Argiculture  570
## 10     Telecomms  351
```

Job Industry and Brand

```
brand_job <- trans_demographic %>% group_by(job_industry) %>% count(brand)
ggplot(brand_job, aes(job_industry, n, fill = brand)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 900, by = 45),
    limits = c(0, 900)) +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(title = "REGULAR CUSTOMERS-BRAND PER JOB INDUSTRY", x = "")
```



The visits clearly differed per job industry a customer was in.

Wealth Segment

Shorten some classifications

The 3 classes are Mass Customer, High Net Worth and Affluent Customer

```
trans_demographic <- trans_demographic %>%
    mutate(wealth_segment = case_when(
        str_detect(wealth_segment, "Affluent") ~ "Affluent",
        str_detect(wealth_segment, "High") ~ "High Net",
        str_detect(wealth_segment, "Mass") ~ "Mass",
        TRUE ~ wealth_segment
    ))
```

```
trans_demographic %>% count(wealth_segment, sort = T)
```

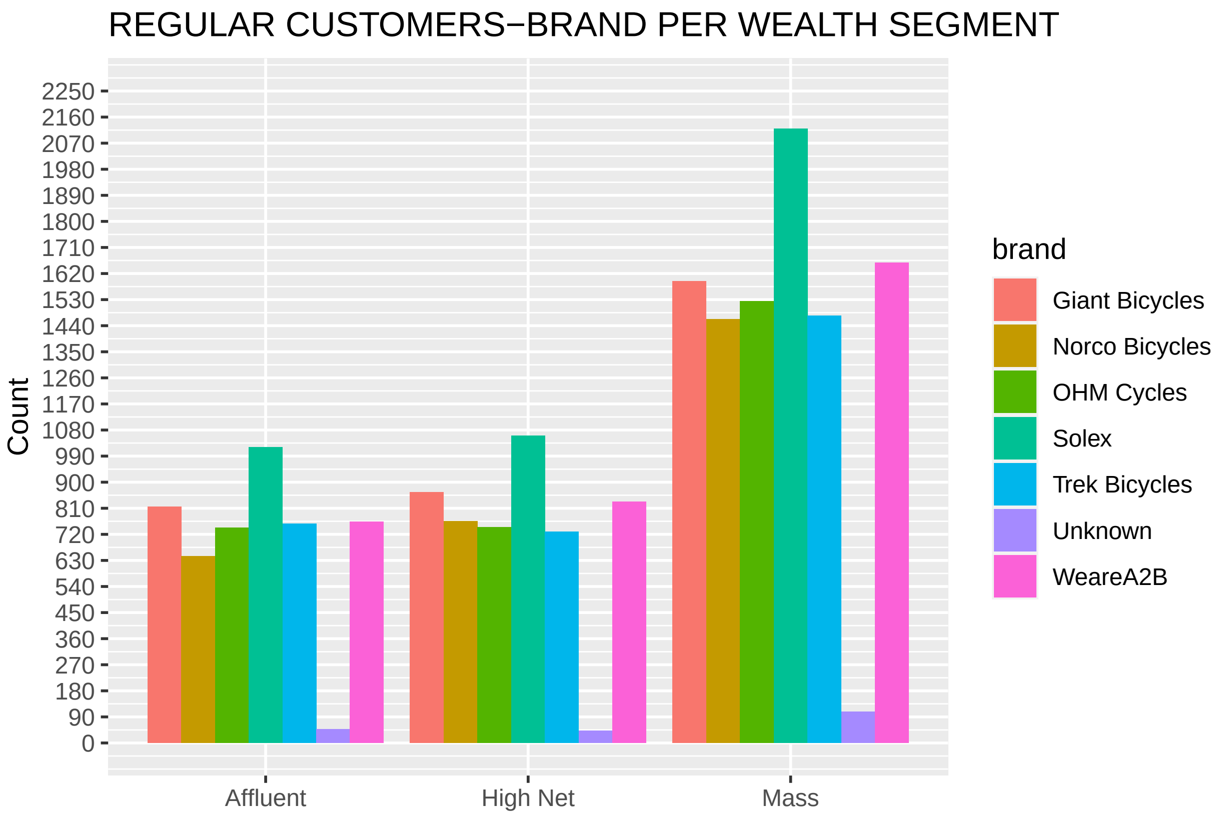
```
##   wealth_segment    n
## 1           Mass 9940
## 2       High Net 5039
## 3       Affluent 4790
```

Wealth Segment and Brand

```r
brand_wealth <- trans_demographic %>% group_by(wealth_segment) %>%
  count(brand)
brand_wealth
```

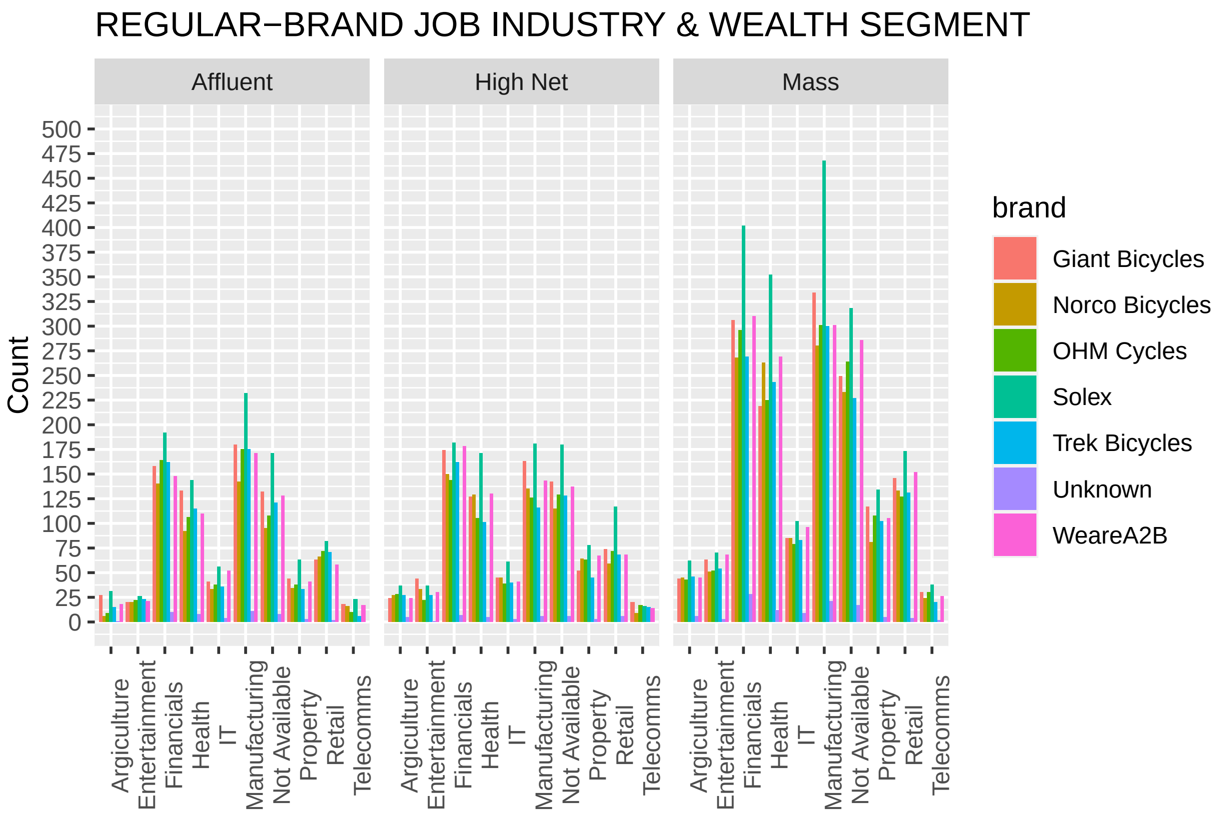
```
## # A tibble: 21 x 3
## # Groups:   wealth_segment [3]
##    wealth_segment brand               n
##    <chr>          <chr>           <int>
##  1 Affluent       Giant Bicycles    816
##  2 Affluent       Norco Bicycles    644
##  3 Affluent       OHM Cycles        742
##  4 Affluent       Solex            1020
##  5 Affluent       Trek Bicycles     757
##  6 Affluent       Unknown            47
##  7 Affluent       WeareA2B          764
##  8 High Net       Giant Bicycles    865
##  9 High Net       Norco Bicycles    766
## 10 High Net       OHM Cycles        745
## # i 11 more rows
```

```r
ggplot(brand_wealth, aes(wealth_segment, n, fill = brand)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 2250, by = 90),
    limits = c(0, 2250)) +
  labs(title = "REGULAR CUSTOMERS-BRAND PER WEALTH SEGMENT", x = "")
```



Solex was the most preffered brand per wealth segment. Affluent and High Net Worth customers behaved almost the same way.

```
brand_wealth_industry <- trans_demographic %>% group_by(wealth_segment, job_industry) %>% coun
ggplot(brand_wealth_industry, aes(job_industry, n, fill = brand)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 500, by = 25),
    limits = c(0, 500)) +
  theme(axis.text.x = element_text(angle = 90)) +
  facet_wrap(~wealth_segment) +
  labs(title = "REGULAR-BRAND JOB INDUSTRY & WEALTH SEGMENT", x = "")
```



Deceased Indicator

```
trans_demographic %>% count(deceased, sort = T)
```

```
##   deceased     n
## 1        N 19761
## 2        Y     8
```

The deceased customers

```
deceased_customers <- trans_demographic %>% filter(deceased == "Y")
deceased_customers %>% count(customer_id)
```

```
##   customer_id n
## 1         753 8
```

It is customer 753 who is deceased and she has purchased 8 distinct transactions.

The column deceased can be removed but the customer behavior can be analyzed to inform on the behavior of other customers

```
trans_demographic_1 <- trans_demographic %>% select(-deceased)
```

Owns Car

```
trans_demographic_1 %>% count(owns_car, sort = T) %>%
  mutate(percent = n / sum(n) * 100)
```

```
##   owns_car    n  percent
## 1      Yes 9960 50.38191
## 2       No 9809 49.61809
```

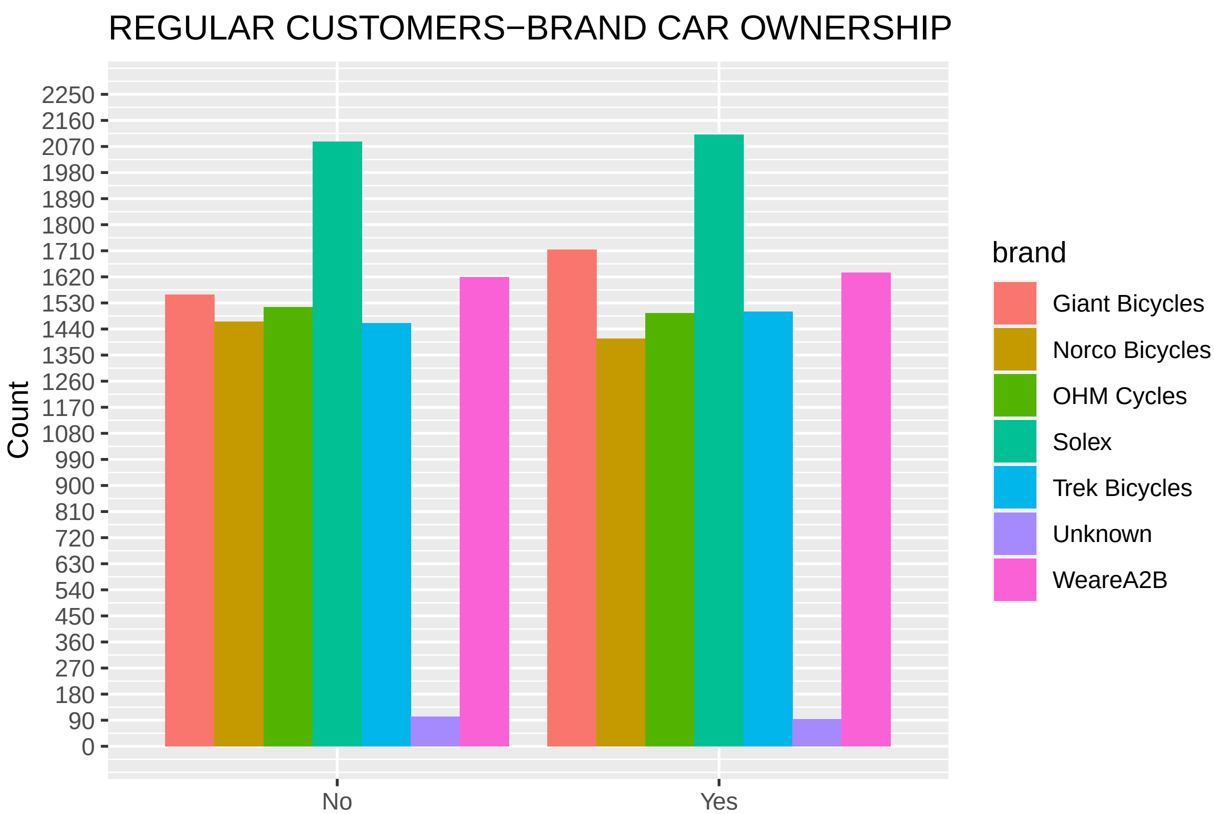
On Occassions that customers visited it can be said that when they did own a car and when they owned a car was almost split in the middle.

Owns car and brand

```
brand_car <- trans_demographic_1 %>% group_by(owns_car) %>% count(brand)
brand_car
```

```
## # A tibble: 14 x 3
## # Groups:   owns_car [2]
##    owns_car brand              n
##    <chr>    <chr>          <int>
##  1 No       Giant Bicycles  1559
##  2 No       Norco Bicycles  1465
##  3 No       OHM Cycles      1516
##  4 No       Solex           2087
##  5 No       Trek Bicycles   1461
##  6 No       Unknown          102
##  7 No       WeareA2B        1619
##  8 Yes      Giant Bicycles  1715
##  9 Yes      Norco Bicycles  1408
## 10 Yes      OHM Cycles      1496
## 11 Yes      Solex           2112
## 12 Yes      Trek Bicycles   1500
## 13 Yes      Unknown           94
## 14 Yes      WeareA2B        1635
```

```r
ggplot(brand_car, aes(owns_car, n, fill = brand)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 2250, by = 90),
    limits = c(0, 2250)) +
  labs(title = "REGULAR CUSTOMERS-BRAND CAR OWNERSHIP", x = "")
```



The difference was not much.

Tenure

```r
summary(trans_demographic_1$tenure)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1.00    6.00   11.00   10.67   15.00   22.00     444
```

We have NAs for the Tenure.

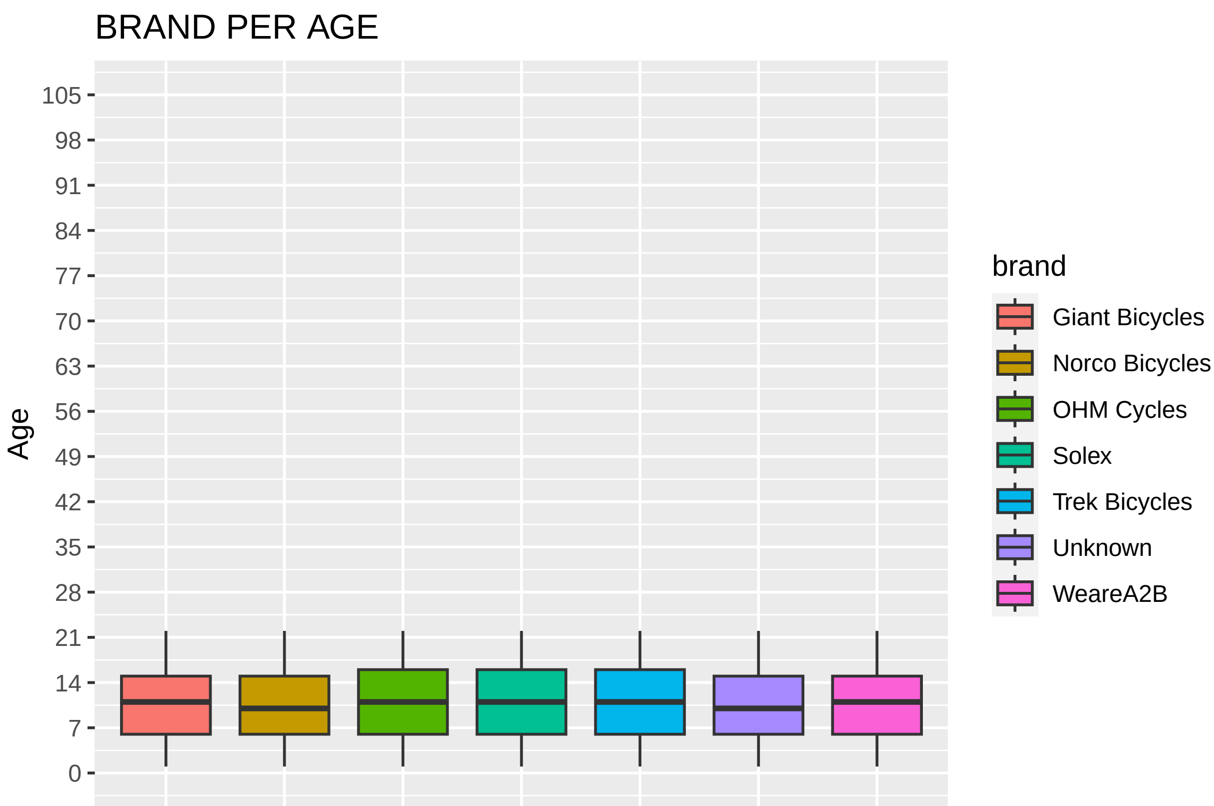
```r
trans_demographic %>% group_by(brand) %>%
  summarise(average = mean(tenure, na.rm = TRUE)) %>% arrange(desc(average))
```

```
## # A tibble: 7 x 2
##   brand           average
##   <chr>             <dbl>
## 1 OHM Cycles         10.8
## 2 WeareA2B           10.7
## 3 Giant Bicycles     10.7
## 4 Solex              10.7
## 5 Trek Bicycles      10.7
## 6 Norco Bicycles     10.5
## 7 Unknown            10.4
```

After removing NAs we get that we have an almost same average of customers.

Tenure and dob have the same observations of missing values.

```
ggplot(trans_demographic_1, aes(brand, tenure, fill = brand)) +
  geom_boxplot() +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) +
  scale_y_continuous("Age",
    breaks = seq(0, 105, by = 7),
    limits = c(0, 105)) +
  labs(title = "BRAND PER AGE", x = "")
```



CUSTOMER ADDRESS

Join with the customer address data by customer_id

```
class(trans_demographic_1$customer_id)
```

```
## [1] "factor"
```

```
class(address$customer_id)
```

```
## [1] "numeric"
```

```
address$customer_id <- as.factor(as.numeric(address$customer_id))
trans_data <- as.data.frame(trans_demographic_1 %>%
  inner_join(address, by = "customer_id"))
dim(trans_data)
```

```
## [1] 19741    31
```

Country

```
trans_data %>% count(country, sort = T)
```

```
##     country     n
## 1 Australia 19741
```

All customers are from Australia, thus we drop the country column

```
trans_data <- trans_data %>% select(-country)
dim(trans_data)
```

```
## [1] 19741    30
```

Address of the customers

```
n_distinct(trans_data$address)
```

```
## [1] 2917
```

```r
n_distinct(trans_data$customer_id)
```

```
## [1] 3439
```

Postcode

```r
n_distinct(trans_data$postcode)
```

```
## [1] 830
```

state

```r
n_distinct(trans_data$state)
```

```
## [1] 3
```

There are 2917 distinct addresses, 3438 distinct customer_id, 830 distinct postcodes and 3 distinct states.

State

```r
trans_data %>% count(state, sort = T) %>%
  mutate(percent = n / sum(n) * 100)
```

```
##   state     n  percent
## 1   NSW 10560 53.49273
## 2   VIC  4960 25.12537
## 3   QLD  4221 21.38190
```

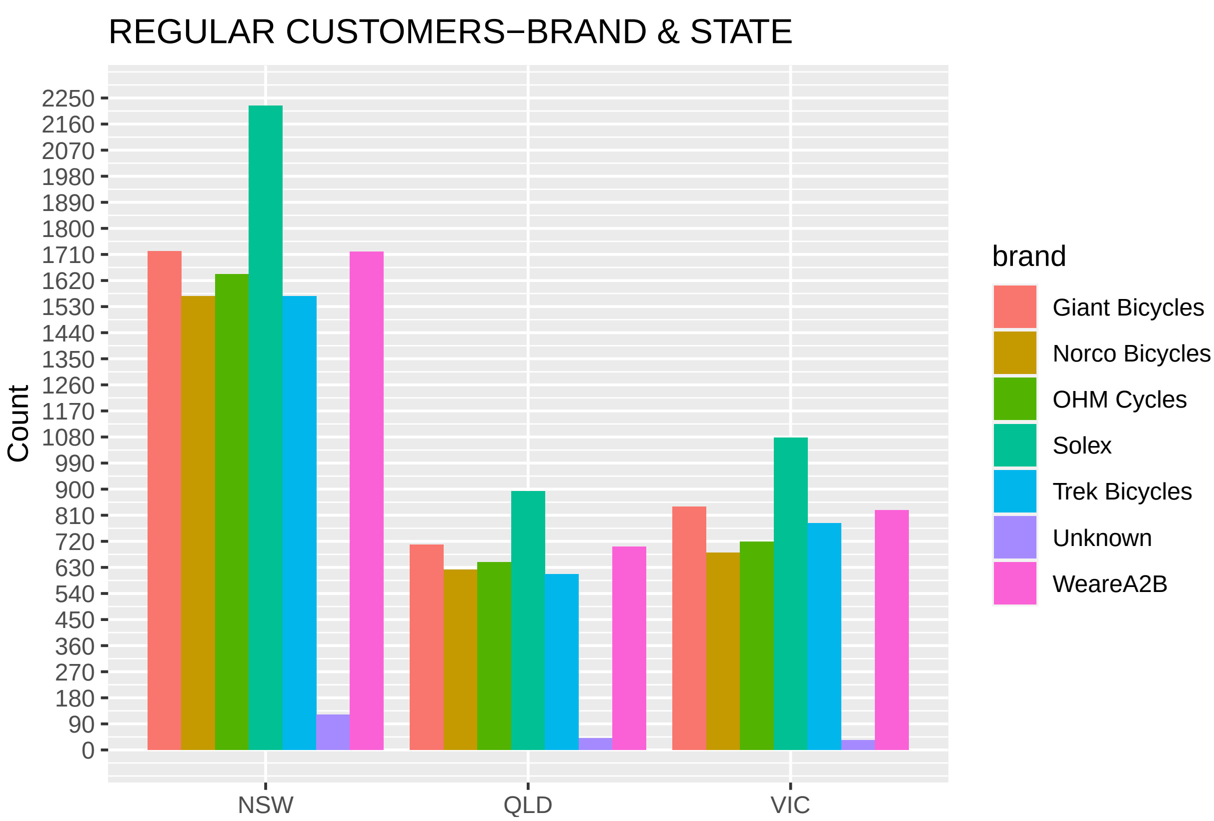
A significant majority are from New South Wales.

Brand and State

```r
brand_state <- trans_data %>% group_by(state) %>% count(brand) %>%
  mutate(percent = n / sum(n) * 100)
brand_state
```

```
## # A tibble: 21 x 4
## # Groups:   state [3]
##    state brand               n percent
##    <chr> <chr>           <int>   <dbl>
##  1 NSW   Giant Bicycles   1721   16.3
##  2 NSW   Norco Bicycles   1566   14.8
##  3 NSW   OHM Cycles       1642   15.5
##  4 NSW   Solex            2224   21.1
##  5 NSW   Trek Bicycles    1566   14.8
##  6 NSW   Unknown           121    1.15
##  7 NSW   WeareA2B         1720   16.3
##  8 QLD   Giant Bicycles    709   16.8
##  9 QLD   Norco Bicycles    622   14.7
## 10 QLD   OHM Cycles        648   15.4
## # i 11 more rows
```

```r
ggplot(brand_state, aes(state, n, fill = brand)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 2250, by = 90),
    limits = c(0, 2250)) +
  labs(title = "REGULAR CUSTOMERS-BRAND & STATE", x = "")
```



Brand prefference across states was the same.

Property Value

```
class(trans_data$property_valuation)
```

```
## [1] "numeric"
```

```
summary(trans_data$property_valuation)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   6.000   8.000   7.514  10.000  12.000
```

```
setdiff(1:12, trans_data$property_valuation)
```

```
## integer(0)
```

Property valuation take integers between 1 and 12. The column is numeric. We can convert it to factor

```
trans_data$property_valuation <- as.factor(as.numeric(trans_data$property_valuation))
class(trans_data$property_valuation)
```

```
## [1] "factor"
```

```
trans_data %>% count(property_valuation, sort = T)
```

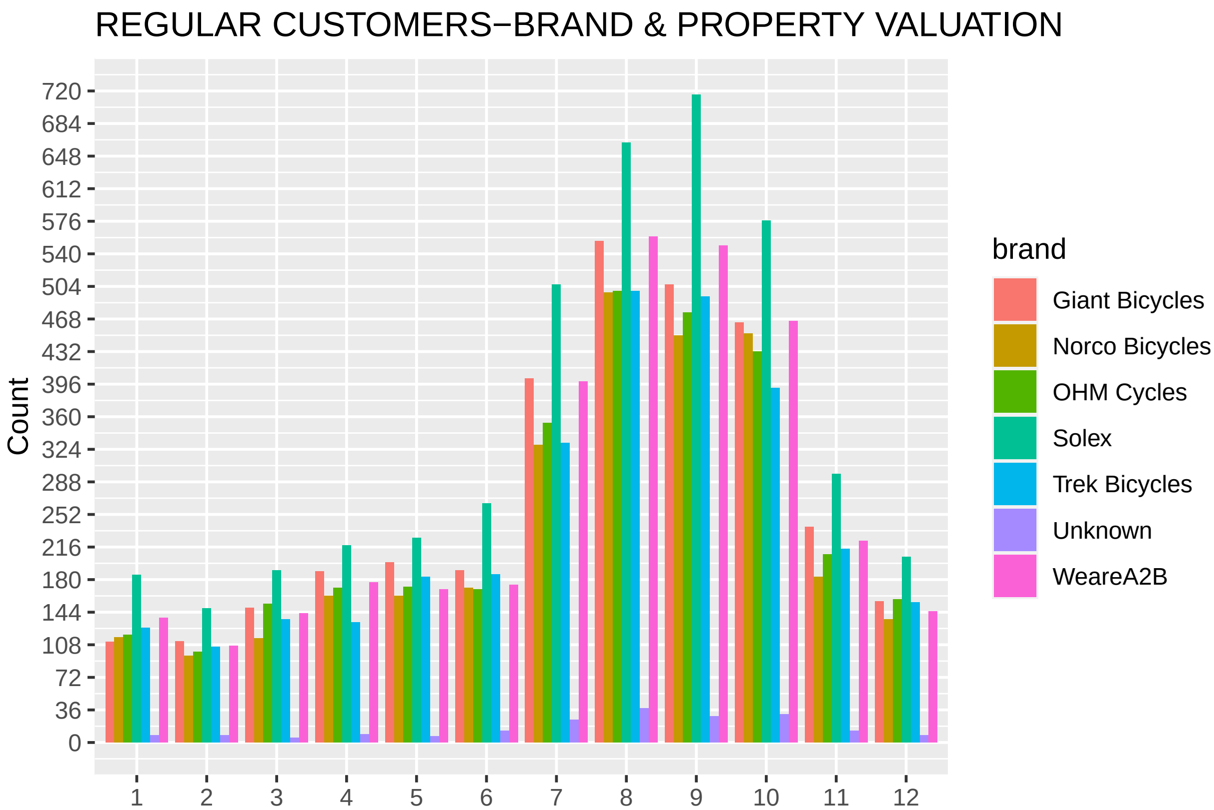
```
##    property_valuation    n
## 1                   8 3309
## 2                   9 3218
## 3                  10 2814
## 4                   7 2345
## 5                  11 1376
## 6                   6 1167
## 7                   5 1118
## 8                   4 1059
## 9                  12  963
## 10                  3  891
## 11                  1  804
## 12                  2  677
```

Brand and Property Valuation

```r
brand_property <- trans_data %>% group_by(property_valuation) %>%
  count(brand) %>% mutate(percent = n / sum(n) * 100)
brand_property
```

```
## # A tibble: 84 x 4
## # Groups:   property_valuation [12]
##    property_valuation brand               n percent
##    <fct>              <chr>           <int>   <dbl>
##  1 1                  Giant Bicycles    111  13.8
##  2 1                  Norco Bicycles    116  14.4
##  3 1                  OHM Cycles        119  14.8
##  4 1                  Solex             185  23.0
##  5 1                  Trek Bicycles     127  15.8
##  6 1                  Unknown             8   0.995
##  7 1                  WeareA2B          138  17.2
##  8 2                  Giant Bicycles    112  16.5
##  9 2                  Norco Bicycles     96  14.2
## 10 2                  OHM Cycles        100  14.8
## # i 74 more rows
```

```r
ggplot(brand_property, aes(property_valuation, n, fill = brand)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 720, by = 36),
    limits = c(0, 720)) +
  labs(title = "REGULAR CUSTOMERS-BRAND & PROPERTY VALUATION", x = "")
```



Within same property valuation group brand choice was almost similar.

Create a column that count visits by each customer

There are days where a single customer had different transaction id implying that a customer can visit on the same day but purchase different products. Therefore visits are counted by distinct transaction dates.

```
customer_same_trandate <- trans_data %>% filter(days_diff == 0)
head(customer_same_trandate %>% select(1:5))
```

```
##   tran_id product_id customer_id  tran_date days_diff
## 1   13818         44          12 2017-08-21         0
## 2   17302         24          90 2017-03-09         0
## 3    9663          7          91 2017-07-28         0
## 4    7199         45          94 2017-11-27         0
## 5   11916         86         104 2017-06-17         0
## 6   15722         32         115 2017-09-29         0
```

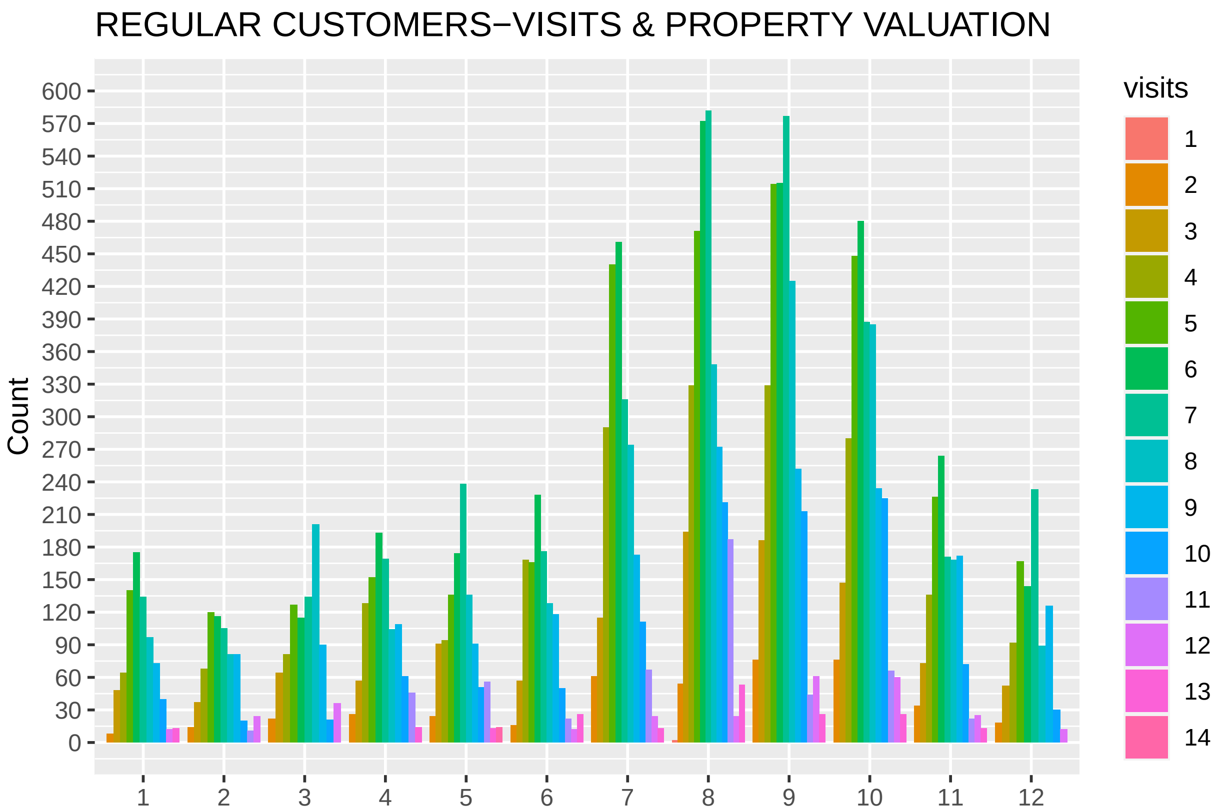
```
trans_data_1 <- trans_data
trans_data_1 <- trans_data_1 %>% group_by(customer_id) %>%
  mutate(visits = n_distinct(tran_date))
```

```r
trans_data_1$visits <- as.factor(as.numeric(trans_data_1$visits))
head(trans_data_1 %>% select(1:5, 31), 13)
```

```
## # A tibble: 13 x 6
## # Groups:   customer_id [2]
##    tran_id product_id customer_id tran_date  days_diff visits
##      <dbl> <fct>      <fct>       <date>         <dbl> <fct>
##  1    9785 72         1           2017-01-05        NA 11
##  2   13424 2          1           2017-02-21        47 11
##  3   14486 23         1           2017-03-27        34 11
##  4   18970 11         1           2017-03-29         2 11
##  5    3765 38         1           2017-04-06         8 11
##  6    5157 47         1           2017-05-11        35 11
##  7   13644 25         1           2017-05-19         8 11
##  8   15663 32         1           2017-06-04        16 11
##  9   16423 9          1           2017-12-09       188 11
## 10   14931 31         1           2017-12-14         5 11
## 11      94 86         1           2017-12-23         9 11
## 12    2261 1          2           2017-05-04        NA 3
## 13    6743 85         2           2017-06-11        38 3
```

```r
property_visits <- trans_data_1 %>% group_by(property_valuation) %>%
  count(visits) %>% mutate(percent = n / sum(n) * 100)
```

```r
ggplot(property_visits, aes(property_valuation, n, fill = visits)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 600, by = 30),
    limits = c(0, 600)) +
  labs(title = "REGULAR CUSTOMERS-VISITS & PROPERTY VALUATION", x = "")
```



Clearly property valuation of a customer affected their visits, where property valuation of 7,8,9 and 10 had high visits but 1,2,3 and 4 had lower visits. The visits also dropped once a customer hit property valuation above 9.

LRFMP Model

The model is prepared by looking at;

- Length-Number of days between a customer's first and last visit.

```
trans_data_2 <- trans_data_1
trans_data_2 <- trans_data_2 %>% group_by(customer_id) %>%
  mutate(length = (min(tran_date) %--% max(tran_date)) %/% days(1))
head(trans_data_2 %>% select(1:5, 31:32), 13)
```

```
## # A tibble: 13 x 7
## # Groups:   customer_id [2]
##    tran_id product_id customer_id tran_date  days_diff visits length
##      <dbl> <fct>      <fct>       <date>         <dbl> <fct>   <dbl>
## 1     9785 72         1           2017-01-05        NA 11        352
## 2    13424 2          1           2017-02-21        47 11        352
## 3    14486 23         1           2017-03-27        34 11        352
## 4    18970 11         1           2017-03-29         2 11        352
```

```
## 5     3765 38          1            2017-04-06         8 11          352
## 6     5157 47          1            2017-05-11        35 11          352
## 7    13644 25          1            2017-05-19         8 11          352
## 8    15663 32          1            2017-06-04        16 11          352
## 9    16423 9           1            2017-12-09       188 11          352
## 10   14931 31          1            2017-12-14         5 11          352
## 11      94 86          1            2017-12-23         9 11          352
## 12    2261 1           2            2017-05-04        NA 3           112
## 13    6743 85          2            2017-06-11        38 3           112
```

- Recency-Number of days between a customer's last visit date and the data last observation date.

```
trans_data_3 <- trans_data_2
trans_data_3 <- trans_data_3 %>% group_by(customer_id) %>%
  mutate(recency =
           (max(tran_date) %--% max(trans_data_3$tran_date)) %/% days(1))
head(trans_data_3 %>% select(1:5, 31:33), 13)
```

```
## # A tibble: 13 x 8
## # Groups:   customer_id [2]
##    tran_id product_id customer_id tran_date  days_diff visits length recency
##      <dbl> <fct>      <fct>       <date>         <dbl> <fct>   <dbl>   <dbl>
## 1     9785 72         1           2017-01-05        NA 11        352       7
## 2    13424 2          1           2017-02-21        47 11        352       7
## 3    14486 23         1           2017-03-27        34 11        352       7
## 4    18970 11         1           2017-03-29         2 11        352       7
## 5     3765 38         1           2017-04-06         8 11        352       7
## 6     5157 47         1           2017-05-11        35 11        352       7
## 7    13644 25         1           2017-05-19         8 11        352       7
## 8    15663 32         1           2017-06-04        16 11        352       7
## 9    16423 9          1           2017-12-09       188 11        352       7
## 10   14931 31         1           2017-12-14         5 11        352       7
## 11      94 86         1           2017-12-23         9 11        352       7
## 12    2261 1          2           2017-05-04        NA 3         112     128
## 13    6743 85         2           2017-06-11        38 3         112     128
```

- Frequency-Number of visits per customer.

We have visits thus we rename the column to frequency

```
trans_data_4 <- trans_data_3
trans_data_4 <- trans_data_4 %>% mutate(frequency = visits)
head(trans_data_4 %>% select(3:5, 31:34), 13)
```

```
## # A tibble: 13 x 7
## # Groups:   customer_id [2]
##    customer_id tran_date  days_diff visits length recency frequency
##    <fct>       <date>         <dbl> <fct>   <dbl>   <dbl> <fct>
##  1 1           2017-01-05        NA 11        352       7 11
##  2 1           2017-02-21        47 11        352       7 11
##  3 1           2017-03-27        34 11        352       7 11
##  4 1           2017-03-29         2 11        352       7 11
##  5 1           2017-04-06         8 11        352       7 11
##  6 1           2017-05-11        35 11        352       7 11
##  7 1           2017-05-19         8 11        352       7 11
##  8 1           2017-06-04        16 11        352       7 11
##  9 1           2017-12-09       188 11        352       7 11
## 10 1           2017-12-14         5 11        352       7 11
## 11 1           2017-12-23         9 11        352       7 11
## 12 2           2017-05-04        NA 3         112     128 3
## 13 2           2017-06-11        38 3         112     128 3
```

- Monetary-Average Amount of money spent by a customer per visit

```
trans_data_5 <- trans_data_4
class(trans_data_5$frequency)
```

```
## [1] "factor"
```

```
trans_data_5$frequency <- as.numeric(as.factor(trans_data_5$frequency))
trans_data_5 <- trans_data_5 %>% group_by(customer_id) %>%
    mutate(monetary = sum(list_price)/frequency)
head(trans_data_5 %>% select(3:5, 31:35), 13)
```

```
## # A tibble: 13 x 8
## # Groups:   customer_id [2]
##    customer_id tran_date  days_diff visits length recency frequency monetary
##    <fct>       <date>         <dbl> <fct>   <dbl>   <dbl>     <dbl>    <dbl>
##  1 1           2017-01-05        NA 11        352       7        11     826.
##  2 1           2017-02-21        47 11        352       7        11     826.
##  3 1           2017-03-27        34 11        352       7        11     826.
##  4 1           2017-03-29         2 11        352       7        11     826.
##  5 1           2017-04-06         8 11        352       7        11     826.
##  6 1           2017-05-11        35 11        352       7        11     826.
##  7 1           2017-05-19         8 11        352       7        11     826.
##  8 1           2017-06-04        16 11        352       7        11     826.
##  9 1           2017-12-09       188 11        352       7        11     826.
## 10 1           2017-12-14         5 11        352       7        11     826.
## 11 1           2017-12-23         9 11        352       7        11     826.
## 12 2           2017-05-04        NA 3         112     128         3    1383.
## 13 2           2017-06-11        38 3         112     128         3    1383.
```

- Periodicity-Median inter visit days of a customer. Median of days_diff.

```
trans_data_6 <- trans_data_5
trans_data_6 <- trans_data_6 %>% group_by(customer_id) %>%
  mutate(periodicity = median(days_diff, na.rm = TRUE))
head(trans_data_6 %>% select(3,5, 31:36), 13)
```

```
## # A tibble: 13 x 8
## # Groups:   customer_id [2]
##    customer_id days_diff visits length recency frequency monetary periodicity
##    <fct>           <dbl> <fct>   <dbl>   <dbl>     <dbl>    <dbl>       <dbl>
##  1 1                  NA 11        352       7        11     826.        12.5
##  2 1                  47 11        352       7        11     826.        12.5
##  3 1                  34 11        352       7        11     826.        12.5
##  4 1                   2 11        352       7        11     826.        12.5
##  5 1                   8 11        352       7        11     826.        12.5
##  6 1                  35 11        352       7        11     826.        12.5
##  7 1                   8 11        352       7        11     826.        12.5
##  8 1                  16 11        352       7        11     826.        12.5
##  9 1                 188 11        352       7        11     826.        12.5
## 10 1                   5 11        352       7        11     826.        12.5
## 11 1                   9 11        352       7        11     826.        12.5
## 12 2                  NA 3         112     128         3    1383.        56
## 13 2                  38 3         112     128         3    1383.        56
```

distinct customers

```
trans_data_7 <- trans_data_6
trans_data_8 <- trans_data_7 %>%
  select(customer_id, frequency, length:periodicity)
trans_data_8 <- trans_data_8 %>% distinct(customer_id, .keep_all = TRUE)
head(trans_data_8, 14)
```

```
## # A tibble: 14 x 6
## # Groups:   customer_id [14]
##    customer_id frequency length recency monetary periodicity
##    <fct>           <dbl>  <dbl>   <dbl>    <dbl>       <dbl>
##  1 1                  11    352       7     826.        12.5
##  2 2                   3    112     128    1383.        56
##  3 4                   2     76     195     524.        76
##  4 5                   6    286      16     984.        56
##  5 6                   5    272      64    1186.        72.5
##  6 7                   3     62     253     332.        31
##  7 8                  10    338      22    1202.        29
##  8 9                   6    251      78     893.        41
```

```
##   9 11                  6     226       46     1130.         37
## 10 12                  6     254       67     1066.         21.5
## 11 13                  7     331       27     1105.         38
## 12 14                  4     186       47      898.         72
## 13 15                  6     309       35      821.         53
## 14 16                  5     222       99     1553.         59
```

```
dim(trans_data_8)
```

```
## [1] 3439    6
```

Missing values

```
sum(is.na(trans_data_8))
```

```
## [1] 0
```

```
summary(trans_data_8)
```

```
##   customer_id     frequency          length          recency      
##  1      :   1   Min.   : 1.000   Min.   :  0.0   Min.   :  0.00  
##  2      :   1   1st Qu.: 4.000   1st Qu.:199.0   1st Qu.: 17.00  
##  4      :   1   Median : 6.000   Median :258.0   Median : 43.00  
##  5      :   1   Mean   : 5.703   Mean   :243.5   Mean   : 59.76  
##  6      :   1   3rd Qu.: 7.000   3rd Qu.:303.0   3rd Qu.: 84.00  
##  7      :   1   Max.   :14.000   Max.   :362.0   Max.   :321.00  
##  (Other):3433                                                    
##     monetary        periodicity    
##  Min.   :  71.49   Min.   :  0.00  
##  1st Qu.: 935.15   1st Qu.: 28.00  
##  Median :1113.25   Median : 41.00  
##  Mean   :1115.02   Mean   : 52.83  
##  3rd Qu.:1291.84   3rd Qu.: 64.00  
##  Max.   :2182.98   Max.   :357.00  
## 
```

There are customers who visited only once even though we had removed customers who only visited once by looking at single counts of customer_id. It is possible for a customer to have more customer_id counts but they visited once since every product_id has different records.

```
trans_data_8 %>% filter(frequency == 1)
```

```
## # A tibble: 1 x 6
## # Groups:   customer_id [1]
##   customer_id frequency length recency monetary periodicity
##   <fct>           <dbl>  <dbl>   <dbl>    <dbl>       <dbl>
## 1 922                 1      0     188    1769.           0
```

We can remove

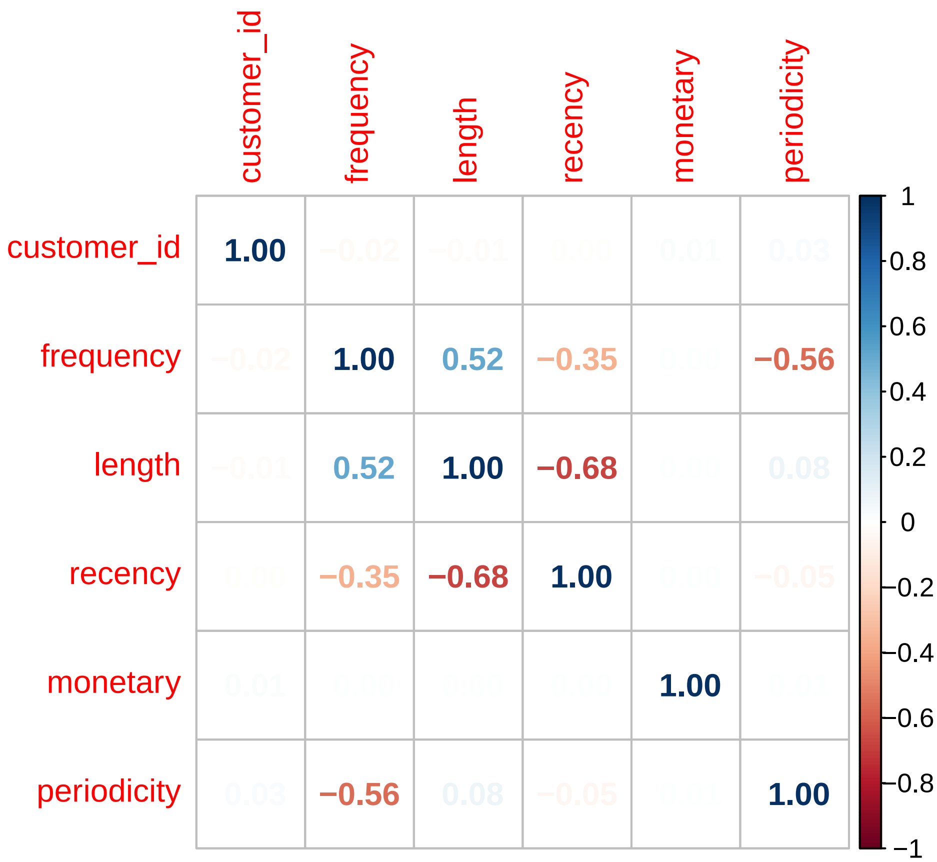
```
trans_data_8 <- trans_data_8 %>% filter(frequency !=1)
dim(trans_data_8)
```

```
## [1] 3438    6
```

```
summary(trans_data_8)
```

```
##   customer_id     frequency          length         recency      
##  1      :   1   Min.   : 2.000   Min.   :  1.0   Min.   :  0.00  
##  2      :   1   1st Qu.: 4.000   1st Qu.:199.0   1st Qu.: 17.00  
##  4      :   1   Median : 6.000   Median :258.0   Median : 43.00  
##  5      :   1   Mean   : 5.705   Mean   :243.5   Mean   : 59.72  
##  6      :   1   3rd Qu.: 7.000   3rd Qu.:303.0   3rd Qu.: 84.00  
##  7      :   1   Max.   :14.000   Max.   :362.0   Max.   :321.00  
##  (Other):3432                                                    
##     monetary        periodicity    
##  Min.   :  71.49   Min.   :  1.00  
##  1st Qu.: 935.15   1st Qu.: 28.00  
##  Median :1113.17   Median : 41.00  
##  Mean   :1114.83   Mean   : 52.84  
##  3rd Qu.:1291.82   3rd Qu.: 64.00  
##  Max.   :2182.98   Max.   :357.00  
## 
```

```
trans_data_8$customer_id <- as.numeric(as.factor(trans_data_8$customer_id))
cormatrix <- cor(trans_data_8)
corrplot(cormatrix, method = "number")
```



Not all vriables are of the same scale

Scaling

```
trans_data_scaled <- trans_data_8 %>%
  column_to_rownames("customer_id") %>% scale()
head(trans_data_scaled)
```

```
##    frequency     length      recency   monetary periodicity
## 1  2.3700067  1.4082320 -0.94483015 -1.0319704 -0.98300989
## 2 -1.2105846 -1.7075698  1.22362377  0.9577649  0.07690711
## 4 -1.6581585 -2.1749400  2.42433793 -2.1104643  0.56422527
## 5  0.1321371  0.5513865 -0.78354019 -0.4676965  0.07690711
## 6 -0.3154368  0.3696314  0.07667294  0.2553656  0.47894459
## 7 -1.2105846 -2.3566951  3.46376213 -2.7963695 -0.53224059
```

Comparing the means-Scaled and Non Scaled

Scaled-means

```
attr(trans_data_scaled, "scaled:center")
```

```
##   frequency      length     recency    monetary periodicity
##     5.70477   243.52850    59.72164  1114.83070    52.84366
```

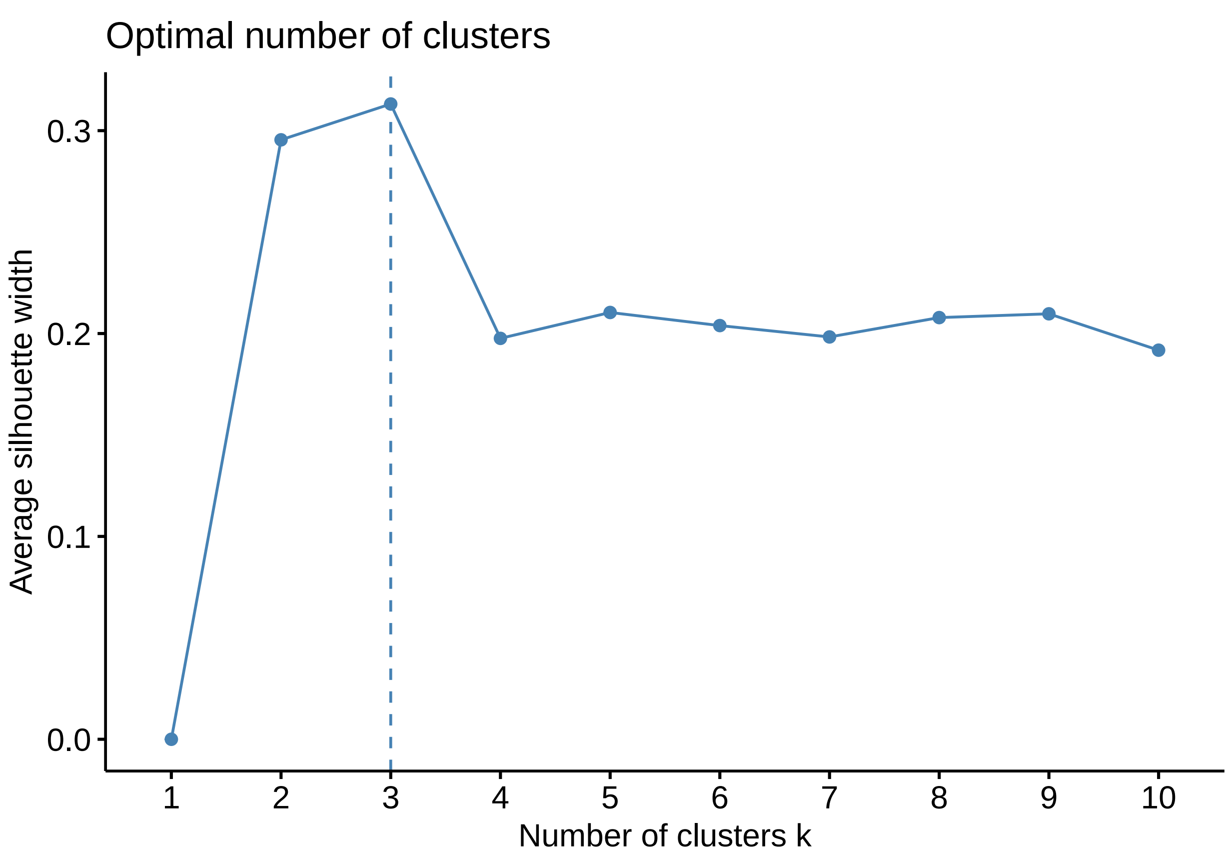
Data-means

```
colMeans(trans_data_8[, c(2:6)])
```

```
##   frequency      length     recency    monetary periodicity
##     5.70477   243.52850    59.72164  1114.83070    52.84366
```

Determining number of clusters

```
fviz_nbclust(trans_data_scaled, kmeans, method = "silhouette")
```

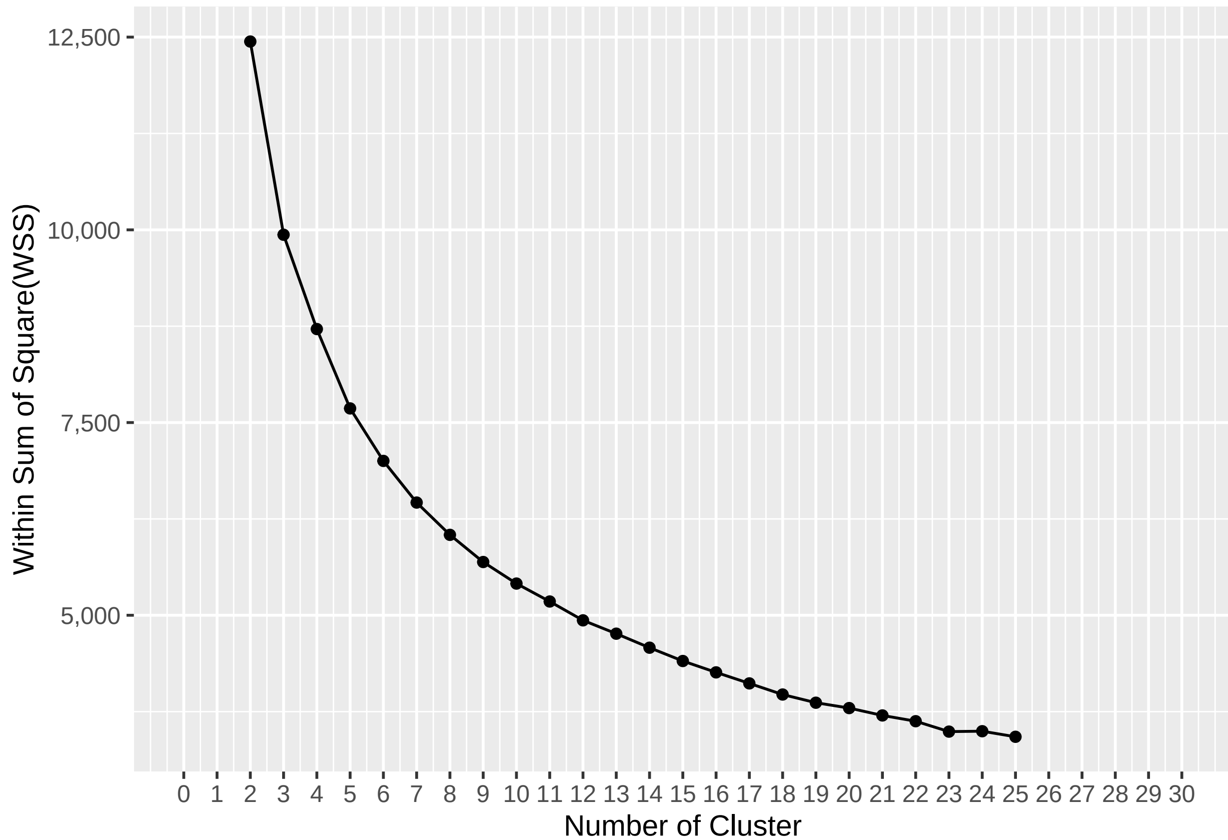


The optimal number of clusters is 3 as per silhouette score

```
clust_test <- map_df(2:25,
                      function(x){
                          set.seed(123)
                          cluster_k <- kmeans(trans_data_scaled, centers = x,
                                              iter.max = 1000)
                          data.frame(n_clust = x,
                                     wss = cluster_k$tot.withinss)
                        })
```

within sum squares plot

```
clust_test %>%
  ggplot(aes(n_clust, wss)) +
  geom_line() +
  geom_point() +
  scale_x_continuous("Number of Cluster",
                     breaks = seq(0, 30, 1),
                     limits = c(0, 30)) +
  scale_y_continuous("Within Sum of Square(WSS)",
                     labels = comma_format())
```



Optimal cluster can be 3 but we can chose 4 because we can segment our data into 4 customer loyalty categories.

Clustering

```
set.seed(123)
trans_clust <- kmeans(trans_data_scaled, centers = 4, iter.max = 1000)
list_clust <- data.frame(customer_id = names(trans_clust$cluster),
                         cluster = trans_clust$cluster)
class(trans_data_8$customer_id)
```

```
## [1] "numeric"
```

```
class(list_clust$customer_id)
```

```
## [1] "character"
```

```
trans_data_8$customer_id <- as.character(as.numeric(trans_data_8$customer_id))
class(trans_data_8$customer_id)
```

```
## [1] "character"
```

```r
trans_data_clustered <- trans_data_8 %>%
  inner_join(list_clust, by = "customer_id")
trans_data_clustered$cluster <- as.character(trans_data_clustered$cluster)
head(trans_data_clustered, 13)
```

```
## # A tibble: 13 x 7
## # Groups:   customer_id [13]
##    customer_id frequency length recency monetary periodicity cluster
##    <chr>           <dbl>  <dbl>   <dbl>    <dbl>       <dbl> <chr>
##  1 1                  11    352       7     826.        12.5 1
##  2 2                   3    112     128    1383.        56   4
##  3 4                   2     76     195     524.        76   4
##  4 5                   6    286      16     984.        56   3
##  5 6                   5    272      64    1186.        72.5 3
##  6 7                   3     62     253     332.        31   4
##  7 8                  10    338      22    1202.        29   1
##  8 9                   6    251      78     893.        41   3
##  9 11                  6    226      46    1130.        37   3
## 10 12                  6    254      67    1066.        21.5 3
## 11 13                  7    331      27    1105.        38   1
## 12 14                  4    186      47     898.        72   3
## 13 15                  6    309      35     821.        53   3
```

Profiling customers

Customers per cluster

```r
trans_data_clustered <- as.data.frame(trans_data_clustered)
trans_data_clustered %>% count(cluster, sort = T) %>%
  mutate(percent = n / sum(n) * 100)
```

```
##   cluster    n  percent
## 1       1 1188 34.55497
## 2       3 1116 32.46073
## 3       4  787 22.89122
## 4       2  347 10.09308
```

35% of the customers were in cluster 1, 32% in cluster 3, 23% in cluster 4 and 10% in cluster 2.

We will use the centroid of the mean of each variable from each cluster

```
cluster_summary <- trans_data_clustered %>% group_by(cluster) %>%
  summarise(customers_no = n_distinct(customer_id),
            across(frequency:periodicity, mean)) %>%
  mutate(count_percent = customers_no / sum(customers_no)) %>%
  arrange(desc(customers_no))
cluster_summary <- cluster_summary %>%select(1:2,8,3:7)
cluster_summary$count_percent <- percent(cluster_summary$count_percent, accuracy = 1)
cluster_summary
```

```
## # A tibble: 4 x 8
##   cluster customers_no count_percent frequency length recency monetary
##   <chr>          <int> <chr>             <dbl>  <dbl>   <dbl>    <dbl>
## 1 1               1188 35%                7.90   289.    34.0    1216.
## 2 3               1116 32%                5.36   265.    39.8     954.
## 3 4                787 23%                4.10   140.   132.     1167.
## 4 2                347 10%                2.95   254.    48.5    1167.
## # i 1 more variable: periodicity <dbl>
```

Cluster 1 has 35% of the customers, with the highest frequency of visits and they also have the most recent member to visit the store. They are the most loyal as they have the largest length score.

Cluster 3 has 32% of the customers, with the 2nd highest frequency, length and with the 2nd recent visit but the least net spenders. They can be classified as regular customers.

Cluster 4 has 23% of the customers, they visit the store more times a month than cluster 2 customers but they are the least loyal as seen by the length. They visit the store at least in every 2 months. They are hibernating customers as their last visit was almost 4 months ago as shown by the recency score.

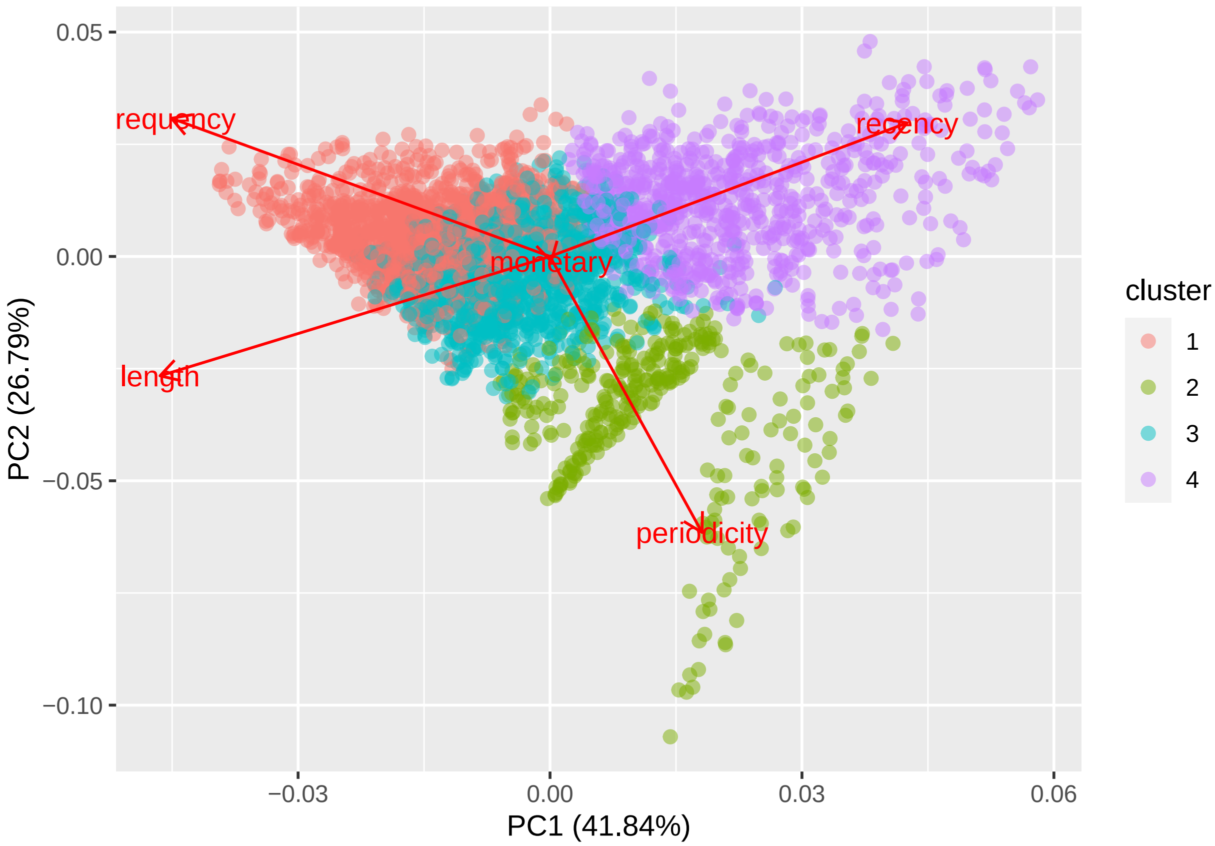
Cluster 2 has the least count of customers. They have the lowest frequency of visit, only 3 times, with an average visit of every 4 months based on the periodicity score. They are the least loyal.

Thus we have;

- Cluster 1:Most loyal
- Cluster 3:Regular
- Cluster 4:Hibernating
- Cluster 2:Seasonal

Visualize Cluster

```
autoplot(trans_clust, data = trans_data_scaled,
         colour = "cluster", size = 2, alpha = 0.5, loadings = T,
         loadings.label = T, loadings.label.size = 4)
```



PC1 gives us 41.84% of information while PC2 gives us 26.79% of information, thus we get 69% of information from the plot while the other 31% is not presented.

As seen cluster 1 is located towards the high frequency direction as it has the highest frequency while it also has the lowest recency as seen with the recency arrow.

Cluster 4 has the highest recency value from the arrow and also from the cluster_summary table cluster had the highest recency thus it was the least recent.

Cluster 2 had the highest periodicity and least frequency as also seen in the plot.

From the boxplot of age and brand prefferences we did not see a significant difference between the ages. We can thus do age groups and see different customer counts.

```
trans_data_extra <- trans_data_1 %>%
  mutate(age_group = case_when(
    age <= 35 ~ "Youth",
    age > 35 & age <= 55 ~ "Middle",
    age > 55 ~ "Older"
  ))
trans_data_extra <- trans_data_extra %>% select(1:22,32,23:31)
```

Age Groups count

replace NAs with unknown

```
class(trans_data_extra$age_group)
```

```
## [1] "character"
```

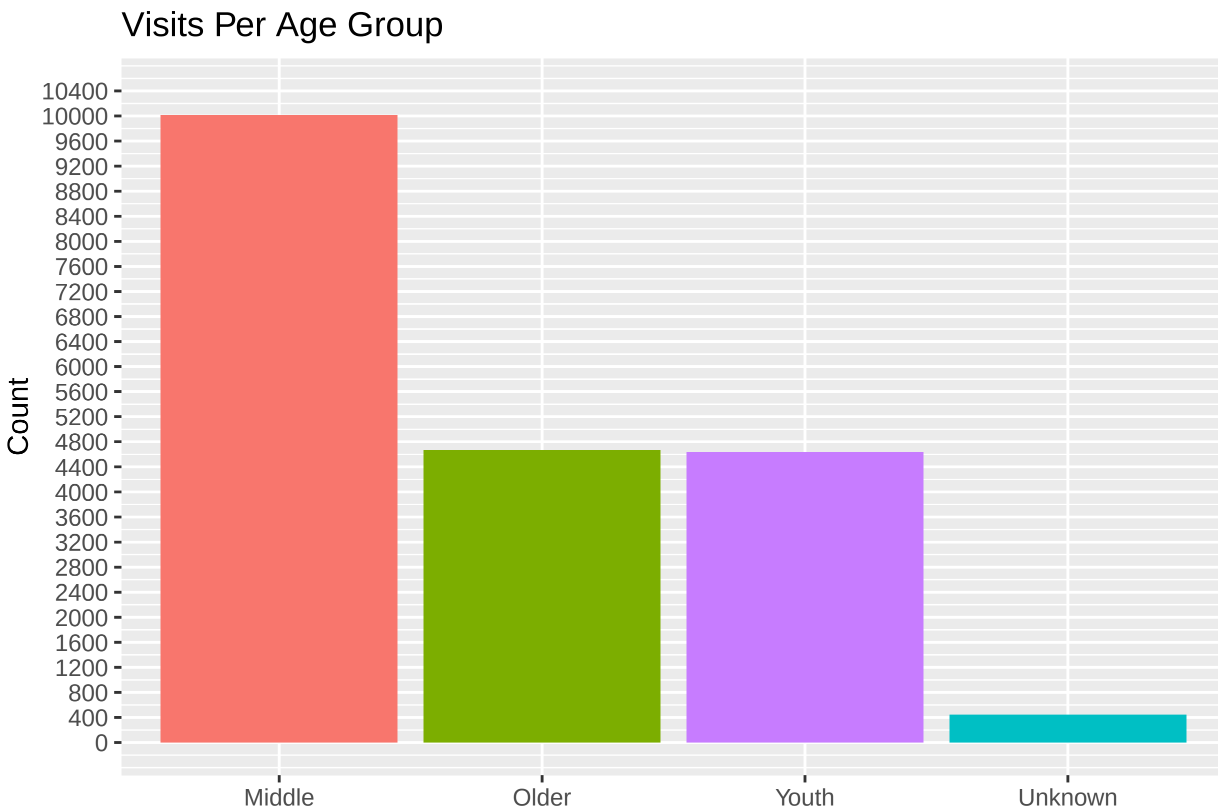
```
trans_data_extra <- as.data.frame(trans_data_extra)
trans_data_extra$age_group[is.na(trans_data_extra$age_group)] <- "Unknown"
trans_data_extra %>% count(age_group, sort = T) %>%
  mutate(percent = round(n / sum(n) * 100))
```

```
##   age_group     n percent
## 1    Middle 10012      51
## 2     Older  4659      24
## 3     Youth  4626      23
## 4   Unknown   444       2
```

51% of the transactions were done by Middle aged individual of between 36-55 years of age.

24% were of Older persons and 23% were youth.

```
trans_data_extra %>% count(age_group, sort = T) %>%
  ggplot(aes(reorder(x = age_group, -n), y = n, fill = age_group)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(legend.position = "none") +
  scale_y_continuous("Count",
    breaks = seq(0, 10400, by = 400),
    limits = c(0, 10400)) +
  labs(title = "Visits Per Age Group", x = "")
```



Clustering with LFRMP and Age Groups

```
class(trans_data_extra$visits)
```

```
## [1] "factor"
```

```
trans_data_extra$visits <- as.numeric(trans_data_extra$visits)
trans_data_extra_1 <- trans_data_extra %>%
  group_by(customer_id, age_group) %>%
  mutate(frequency = visits,
         monetary = sum(list_price)/visits)
trans_data_extra_2 <- trans_data_extra_1 %>%
  select(customer_id, age_group, monetary)
trans_data_extra_2 <- trans_data_extra_2 %>%
  distinct(customer_id, .keep_all = TRUE)
head(trans_data_extra_2)
```

```
## # A tibble: 6 x 3
## # Groups:   customer_id, age_group [6]
##   customer_id age_group monetary
##   <fct>       <chr>        <dbl>
## 1 1           Older         826.
```

```
## 2 2           Middle       1383.
## 3 4           Older         524.
## 4 5           Middle        984.
## 5 6           Older        1186.
## 6 7           Middle        332.
```

```
trans_data_age <- trans_data_extra_2 %>%
  pivot_wider(names_from = age_group,
              values_from = monetary)
head(trans_data_age, 13)
```

```
## # A tibble: 13 x 5
## # Groups:   customer_id [13]
##    customer_id Older Middle Youth Unknown
##    <fct>       <dbl>  <dbl> <dbl>   <dbl>
##  1 1            826.     NA    NA      NA
##  2 2             NA   1383.    NA      NA
##  3 4            524.     NA    NA      NA
##  4 5             NA    984.    NA      NA
##  5 6           1186.     NA    NA      NA
##  6 7             NA    332.    NA      NA
##  7 8           1202.     NA    NA      NA
##  8 9             NA    893.    NA      NA
##  9 11          1130.     NA    NA      NA
## 10 12            NA     NA  1066.      NA
## 11 13          1105.     NA    NA      NA
## 12 14            NA    898.    NA      NA
## 13 15            NA     NA   821.      NA
```

Replace missing values with 0 since a customer can not fall in all age groups.

```
trans_data_age[is.na(trans_data_age)] <- 0
head(trans_data_age, 13)
```

```
## # A tibble: 13 x 5
## # Groups:   customer_id [13]
##    customer_id Older Middle Youth Unknown
##    <fct>       <dbl>  <dbl> <dbl>   <dbl>
##  1 1            826.      0     0       0
##  2 2              0   1383.     0       0
##  3 4            524.      0     0       0
##  4 5              0    984.     0       0
##  5 6           1186.      0     0       0
##  6 7              0    332.     0       0
##  7 8           1202.      0     0       0
```

```
## 8 9                0     893.     0          0
## 9 11           1130.      0       0          0
## 10 12             0       0    1066.         0
## 11 13          1105.      0       0          0
## 12 14             0     898.     0          0
## 13 15             0       0     821.        0
```

Combining with previous LRFMP model

```
trans_data_age_2 <- trans_data_8 %>%
  inner_join(trans_data_age, by = "customer_id")
head(trans_data_age_2, 13)
```

```
## # A tibble: 13 x 10
## # Groups:   customer_id [13]
##    customer_id frequency length recency monetary periodicity Older Middle Youth
##    <chr>           <dbl>  <dbl>   <dbl>    <dbl>       <dbl> <dbl>  <dbl> <dbl>
##  1 1                  11    352       7     826.        12.5  826.      0     0
##  2 2                   3    112     128    1383.        56       0  1383.     0
##  3 4                   2     76     195     524.        76     524.      0     0
##  4 5                   6    286      16     984.        56       0   984.     0
##  5 6                   5    272      64    1186.        72.5 1186.      0     0
##  6 7                   3     62     253     332.        31       0   332.     0
##  7 8                  10    338      22    1202.        29    1202.      0     0
##  8 9                   6    251      78     893.        41       0   893.     0
##  9 11                  6    226      46    1130.        37    1130.      0     0
## 10 12                  6    254      67    1066.        21.5     0      0  1066.
## 11 13                  7    331      27    1105.        38    1105.      0     0
## 12 14                  4    186      47     898.        72       0   898.     0
## 13 15                  6    309      35     821.        53       0      0   821.
## # i 1 more variable: Unknown <dbl>
```

Scaling

```
trans_data_age_scaled <- trans_data_age_2 %>%
  column_to_rownames("customer_id") %>%
  scale()
head(trans_data_age_scaled, 13)
```

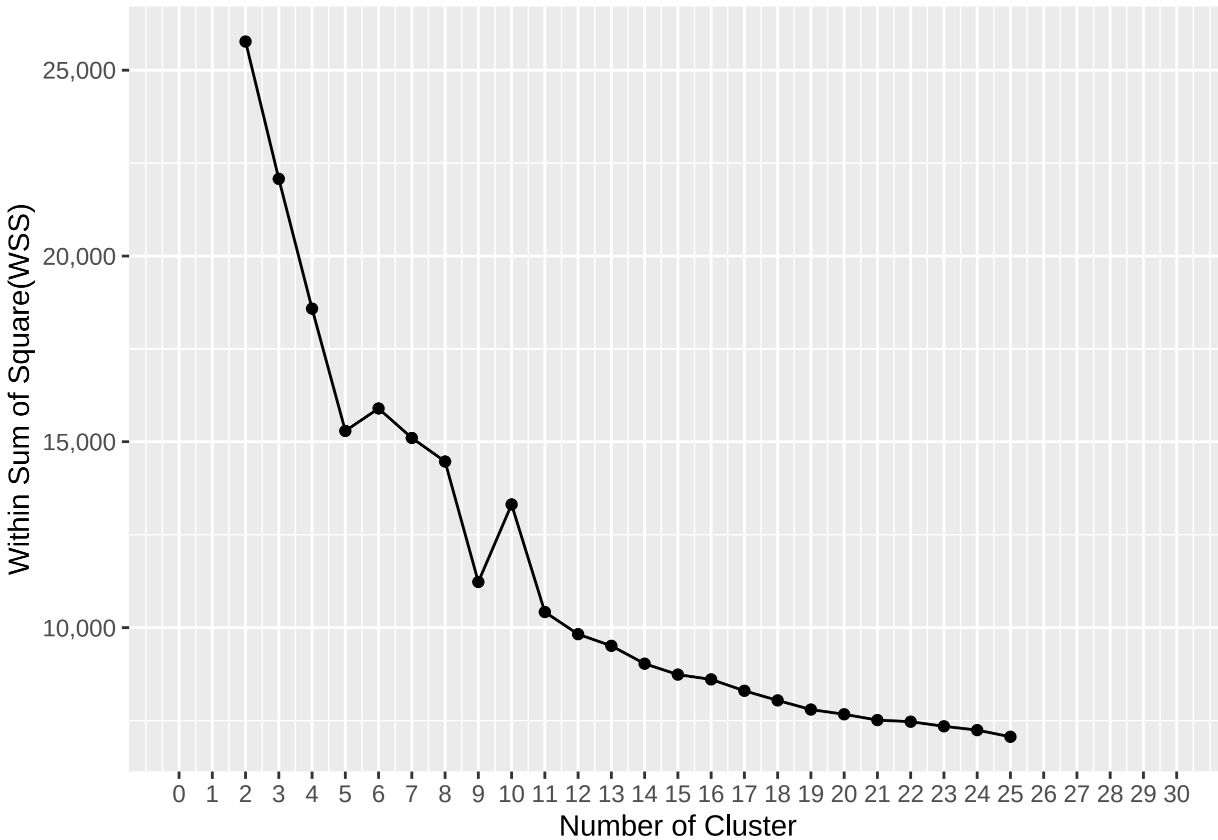
```
##     frequency      length     recency    monetary  periodicity      Older
## 1   2.3740999  1.40901157 -0.94353536 -1.03149456 -0.984172809  1.1424200
## 2  -1.2131105 -1.71018316  1.22068261  0.95929279  0.077927898 -0.5334066
## 4  -1.6615118 -2.17806237  2.41905124 -2.11055873  0.566250062  0.5296059
## 5   0.1320934  0.55123302 -0.78256047 -0.46692229  0.077927898 -0.5334066
```

```
## 6  -0.3163079  0.36928000  0.07597228  0.25652215  0.480793684  1.8739010
## 7  -1.2131105 -2.36001539  3.45644498 -2.79682655 -0.532474807 -0.5334066
## 8   1.9256986  1.22705855 -0.67524388  0.31418436 -0.581307024  1.9066481
## 9   0.1320934  0.09635046  0.32637766 -0.79186331 -0.288313725 -0.5334066
## 11  0.1320934 -0.22856566 -0.24597750  0.05488899 -0.385978158  1.7593909
## 12  0.1320934  0.13534039  0.12963057 -0.17452003 -0.764427835 -0.5334066
## 13  0.5804947  1.13608204 -0.58581338 -0.03423701 -0.361562050  1.7087752
## 14 -0.7647092 -0.74843145 -0.22809141 -0.77232754  0.468585630 -0.5334066
## 15  0.1320934  0.85015585 -0.44272459 -1.05044050  0.004679574 -0.5334066
##        Middle      Youth    Unknown
## 1  -0.9531294 -0.5341058 -0.1463688
## 2   1.3917434 -0.5341058 -0.1463688
## 4  -0.9531294 -0.5341058 -0.1463688
## 5   0.7149857 -0.5341058 -0.1463688
## 6  -0.9531294 -0.5341058 -0.1463688
## 7  -0.3905842 -0.5341058 -0.1463688
## 8  -0.9531294 -0.5341058 -0.1463688
## 9   0.5607970 -0.5341058 -0.1463688
## 11 -0.9531294 -0.5341058 -0.1463688
## 12 -0.9531294  1.6305840 -0.1463688
## 13 -0.9531294 -0.5341058 -0.1463688
## 14  0.5700670 -0.5341058 -0.1463688
## 15 -0.9531294  1.1326371 -0.1463688
```

Number of clusters

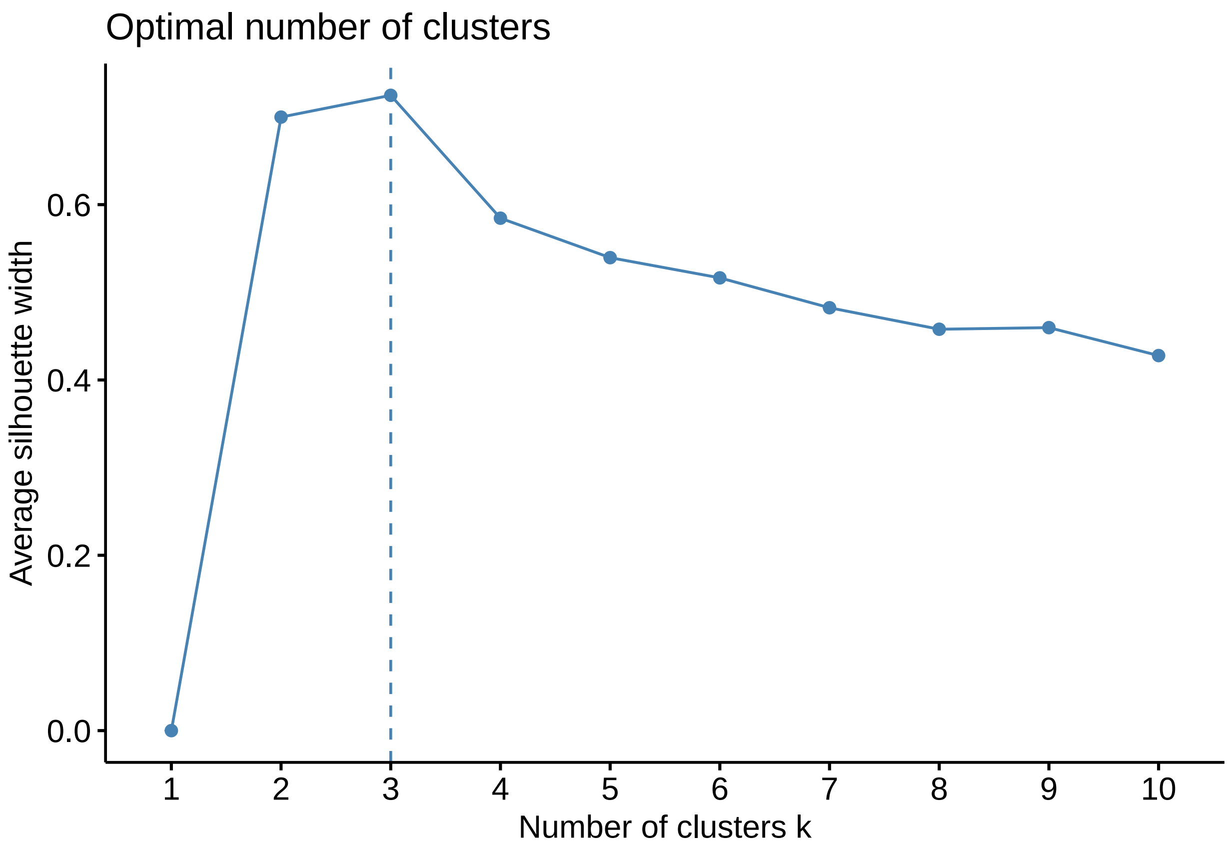
```r
clust_age_test <- map_df(2:25,
                      function(x){
                          set.seed(123)
                          cluster_k <- kmeans(trans_data_age_scaled,
                                              centers = x, iter.max = 1000)
                          data.frame(n_clust = x,
                                     wss = cluster_k$tot.withinss)
                        })
```

```r
clust_age_test %>%
  ggplot(aes(n_clust, wss)) +
  geom_line() +
  geom_point() +
  scale_x_continuous("Number of Cluster",
                     breaks = seq(0, 30, 1),
                     limits = c(0, 30)) +
  scale_y_continuous("Within Sum of Square(WSS)",
                     labels = comma_format())
```

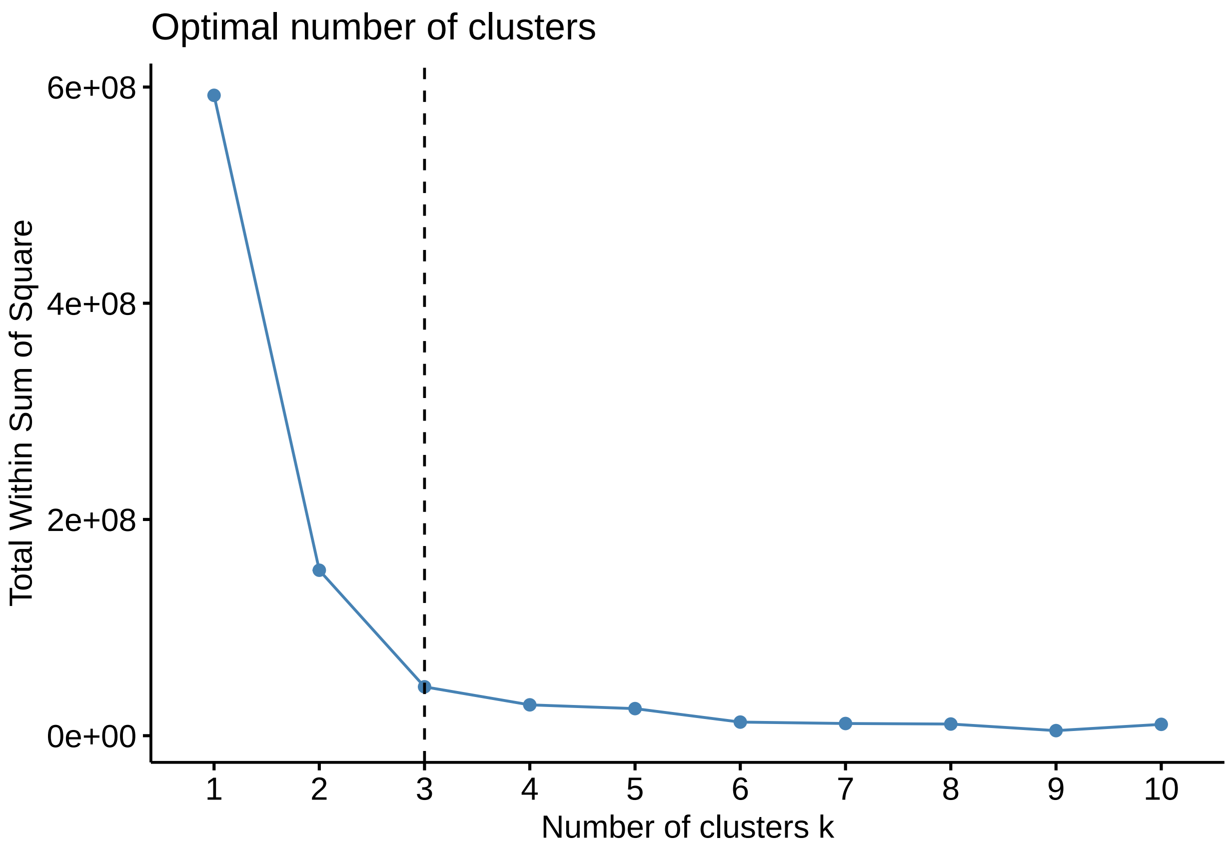


WSS

```
set.seed(123)
fviz_nbclust(clust_age_test, kmeans, method = "silhouette")
```




```r
set.seed(123)
fviz_nbclust(clust_age_test, kmeans, method = "wss") +
  geom_vline(xintercept = 3, linetype = 2)
```



Optimal number of clusters is 3

Clustering

```
set.seed(123)
trans_age_clust <- kmeans(trans_data_age_scaled, centers = 3, iter.max = 1000)
list_age_clust <- data.frame(customer_id = names(trans_age_clust$cluster),
                             cluster = trans_age_clust$cluster)
class(trans_data_age_2$customer_id)
```

```
## [1] "character"
```

```
trans_age_clustered <- trans_data_age_2 %>%
  inner_join(list_age_clust, by = "customer_id")
trans_age_clustered$cluster <- as.character(trans_age_clustered$cluster)
head(trans_age_clustered, 13)
```

```
## # A tibble: 13 x 11
## # Groups:   customer_id [13]
##    customer_id frequency length recency monetary periodicity Older Middle Youth
##    <chr>           <dbl>  <dbl>   <dbl>    <dbl>       <dbl> <dbl>  <dbl> <dbl>
```

```
##   1 1                    11     352        7      826.         12.5  826.      0      0
##   2 2                     3     112      128     1383.         56       0   1383.     0
##   3 4                     2      76      195      524.         76     524.      0      0
##   4 5                     6     286       16      984.         56       0    984.     0
##   5 6                     5     272       64     1186.         72.5 1186.      0      0
##   6 7                     3      62      253      332.         31       0    332.     0
##   7 8                    10     338       22     1202.         29    1202.      0      0
##   8 9                     6     251       78      893.         41       0    893.     0
##   9 11                    6     226       46     1130.         37    1130.      0      0
##  10 12                    6     254       67     1066.         21.5     0       0  1066.
##  11 13                    7     331       27     1105.         38    1105.      0      0
##  12 14                    4     186       47      898.         72       0    898.     0
##  13 15                    6     309       35      821.         53       0       0   821.
## # i 2 more variables: Unknown <dbl>, cluster <chr>
```

Profiling customers

Customers per cluster

```
trans_age_clustered <- as.data.frame(trans_age_clustered)
trans_age_clustered %>% count(cluster, sort = T) %>%
  mutate(percent = n / sum(n) * 100)
```

```
##   cluster    n  percent
## 1       1 1353 40.01775
## 2       3 1295 38.30228
## 3       2  733 21.67998
```

40% of the customers were in cluster 1, 38% in cluster 3 and 21% in cluster 2.

We will use the centroid of the mean of each variable from each cluster

```
cluster_age_summary <- trans_age_clustered %>% group_by(cluster) %>%
  summarise(customers_no = n_distinct(customer_id),
            across(frequency:periodicity, mean),
            across(Older:Unknown, median)) %>%
  mutate(count_percent = customers_no / sum(customers_no)) %>%
  arrange(desc(customers_no))
cluster_age_summary <- cluster_age_summary %>%select(1:2,12,3:11)
cluster_age_summary$count_percent <- percent(cluster_age_summary$count_percent, accuracy = 1)
```

summary

```
cluster_age_summary %>% select(1:7)
```

```
## # A tibble: 3 x 7
##   cluster customers_no count_percent frequency length recency monetary
##   <chr>          <int> <chr>             <dbl>  <dbl>   <dbl>    <dbl>
## 1 1               1353 40%                6.37   274.    39.5    1105.
## 2 3               1295 38%                6.26   272.    40.6    1124.
## 3 2                733 22%                3.49   138.   131.     1116.
```

```
cluster_age_summary %>% select(1:3, 8:12)
```

```
## # A tibble: 3 x 8
##   cluster customers_no count_percent periodicity Older Middle Youth Unknown
##   <chr>          <int> <chr>               <dbl> <dbl>  <dbl> <dbl>   <dbl>
## 1 1               1353 40%                  49.5     0  1117.     0       0
## 2 3               1295 38%                  48.8     0     0      0       0
## 3 2                733 22%                  66.0     0     0      0       0
```

- Cluster 1 and 3 have almost the same LRFMP scores but only middle aged individuals of cluster 1 have more than 50% of the customers having monetary value that is more than the median monetary value
- 22% are in cluster 2, where they have the highest periodicity and recency-they are the least loyal.

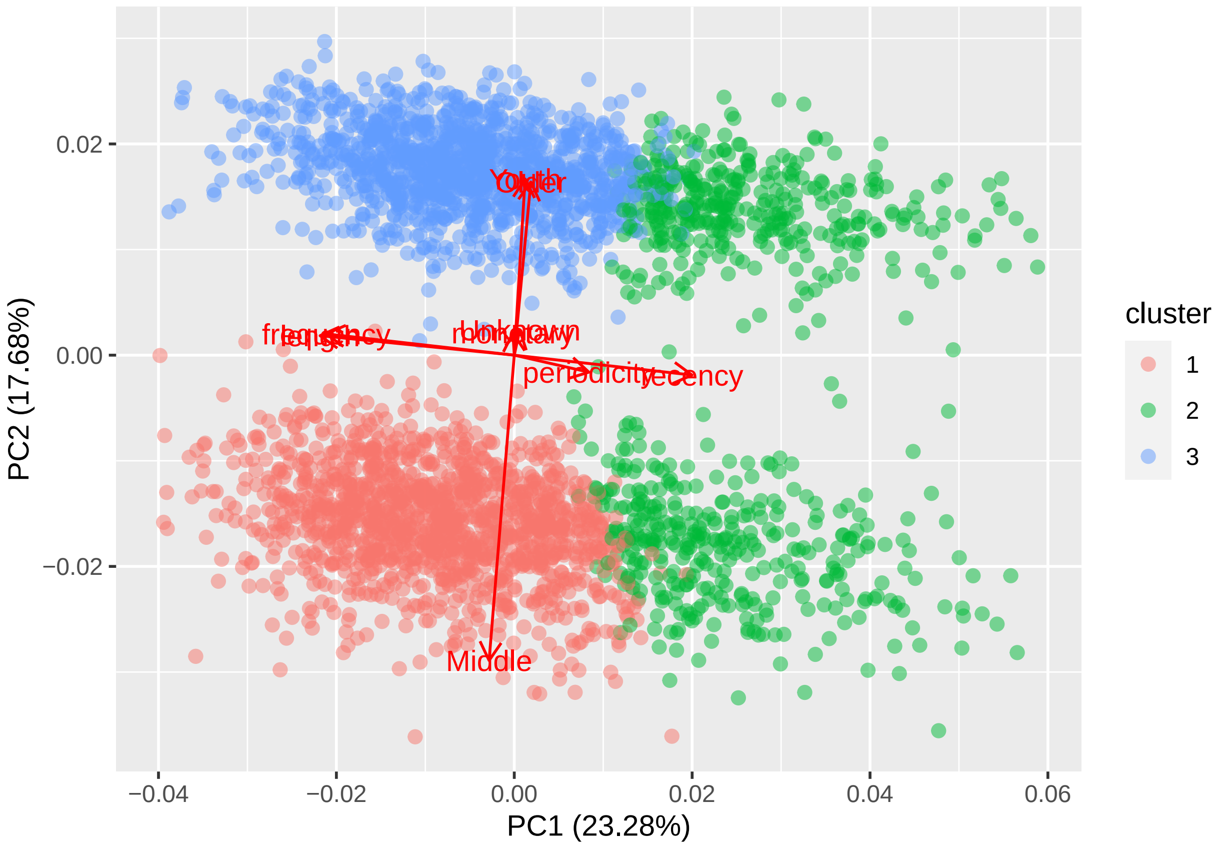
```
unknown_age <- trans_age_clustered %>% filter(Unknown != 0)
```

Clusters;

- Cluster 1:Most loyal Middle aged individuals
- Cluster 3:Regular Older and Youth
- Cluster 2:Seasonal Unknown

Visualize Cluster

```
autoplot(trans_age_clust, data = trans_data_age_scaled,
         colour = "cluster", size = 2, alpha = 0.5, loadings = T,
         loadings.label = T, loadings.label.size = 4)
```



Plot has 41% of information thus the plot is not very informative

Join the customer data with the first LRFMP to try and get the different behaviour of customers per cluster

```
lrfmp_customers <- as.data.frame(trans_data_extra %>%
  inner_join(trans_data_clustered, by = "customer_id"))
```

Past 3 years bike related purchases was coded with digits from 0 to 99 with no missing digit

```
range(lrfmp_customers$past_purchases)
```

```
## [1]  0 99
```

```
setdiff(0:99, lrfmp_customers$past_purchases)
```

```
## integer(0)
```

```
summary(lrfmp_customers$past_purchases)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   24.00   48.00   48.77   73.00   99.00
```

We can code the different past purchases

```
lrfmp_customers <- lrfmp_customers %>%
  mutate(past_purchase_group = case_when(
    past_purchases <= 24 ~ "Bad",
    past_purchases > 24 & past_purchases <= 59 ~ "Good",
    past_purchases > 59 & past_purchases <= 84 ~ "Better",
    past_purchases >= 85 ~ "Excellent"
  ))
```

1 CLUSTERS:GROUPS

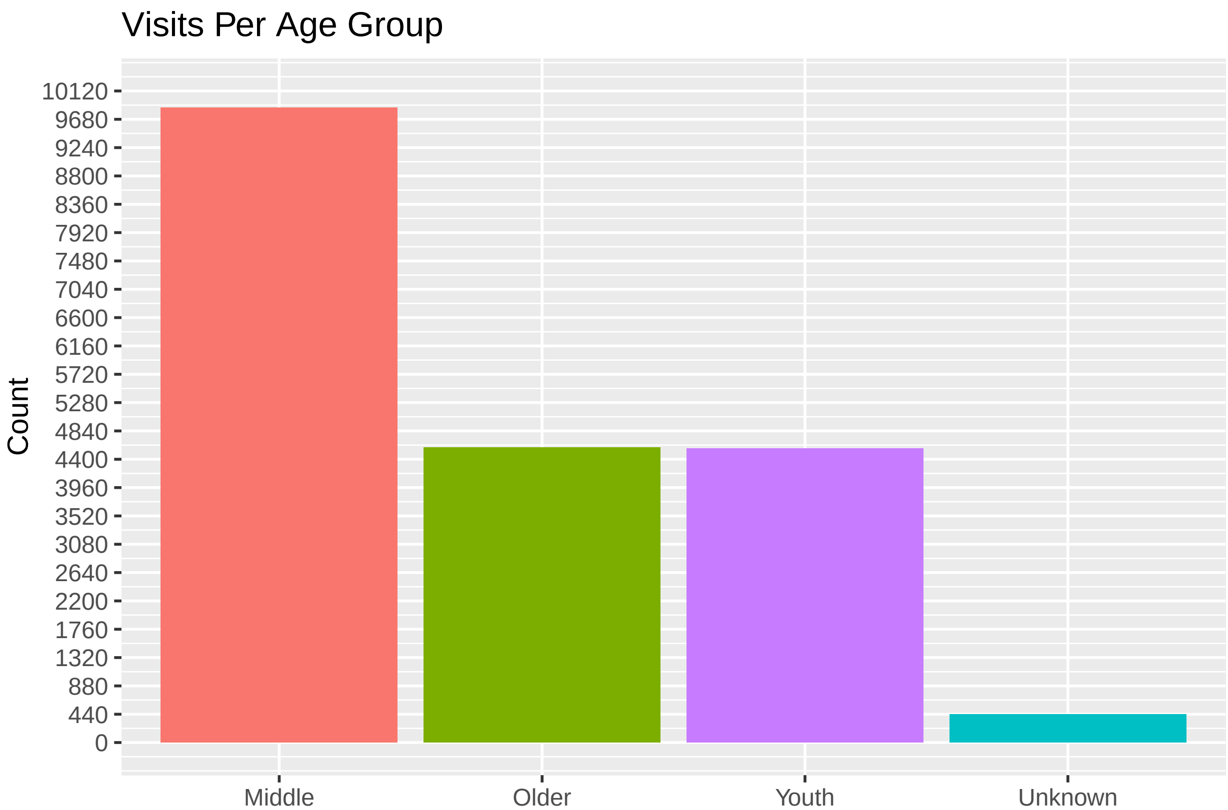
1.1 Age Group

```
lrfmp_customers %>% count(age_group, sort = T) %>%
  mutate(percent = round(n / sum(n) * 100))
```

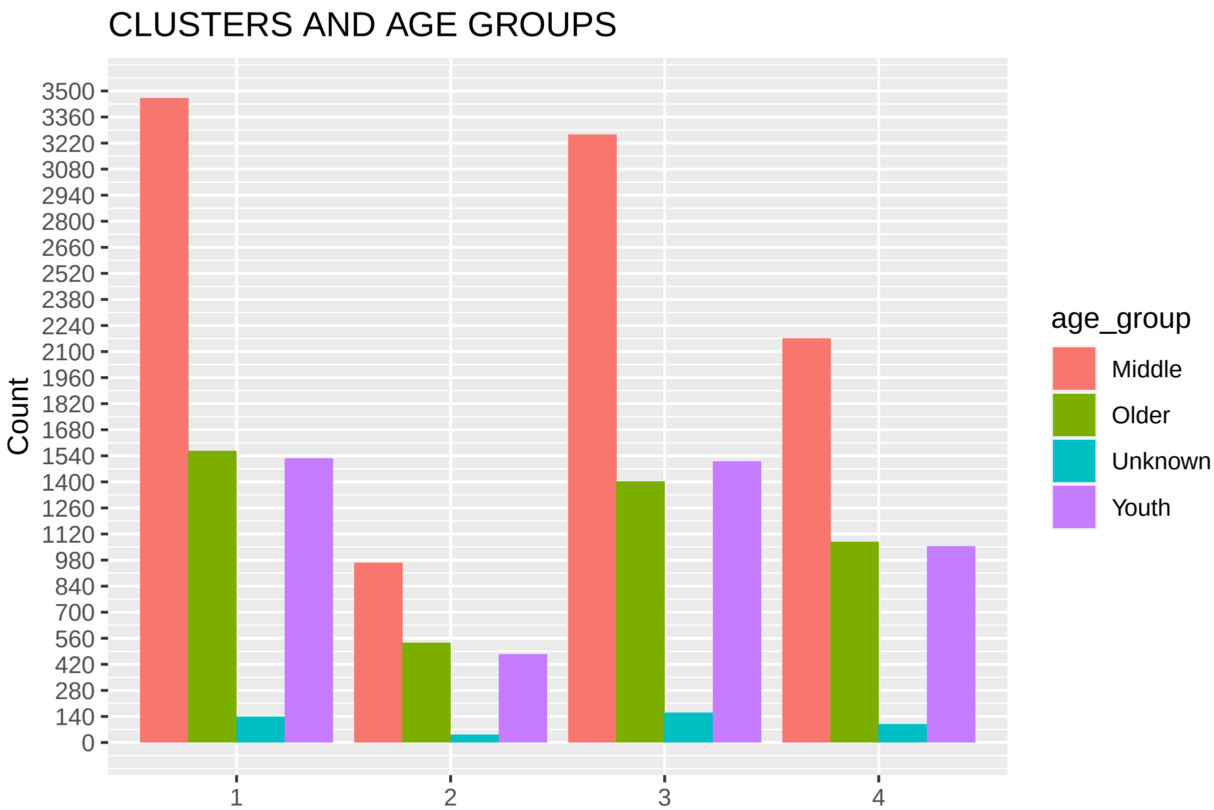
```
##   age_group    n percent
## 1    Middle 9858      51
## 2     Older 4581      24
## 3     Youth 4565      23
## 4   Unknown  440       2
```

51% of customers were Middle aged individuals of between 36-55 years of age. 24% were Older citizens of over 55years and 23% were Youth under 35years.

```
lrfmp_customers %>% count(age_group, sort = T) %>%
  ggplot(aes(reorder(x = age_group, -n), y = n, fill = age_group)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(legend.position = "none") +
  scale_y_continuous("Count",
    breaks = seq(0, 10120, by = 440),
    limits = c(0, 10120)) +
  labs(title = "Visits Per Age Group", x = "")
```




```r
clust_age_group <- lrfmp_customers %>% group_by(cluster) %>% count(age_group)
ggplot(clust_age_group, aes(cluster, n, fill = age_group)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 3500, by = 140),
    limits = c(0, 3500)) +
  labs(title = "CLUSTERS AND AGE GROUPS", x = "")
```



Middle aged individuals were leading across all the clusters

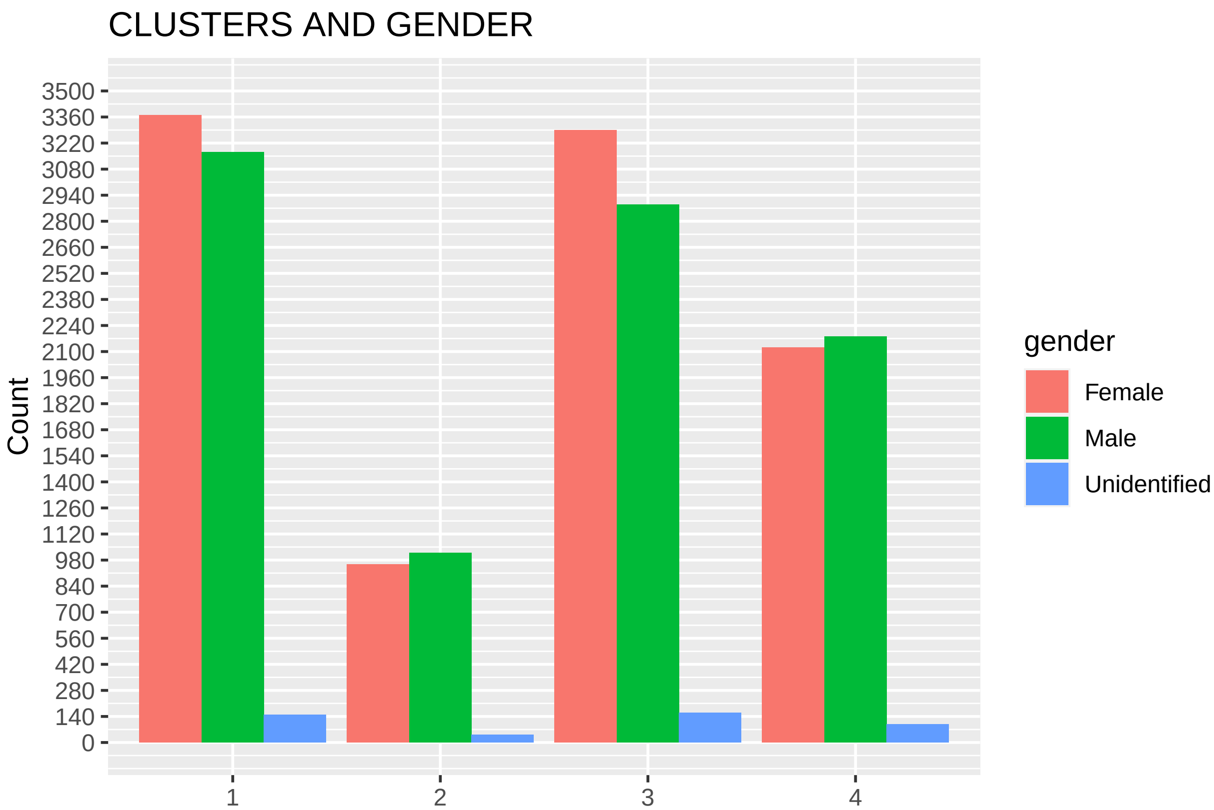
1.2 Gender

```
lrfmp_customers %>% count(gender, sort = T) %>%
  mutate(percent = round(n / sum(n) * 100))
```

```
##         gender    n percent
## 1       Female 9737      50
## 2         Male 9258      48
## 3 Unidentified  449       2
```

50% of customers were Female while 48 were Males. 2% could not identify their gender.

```
clust_gender <- lrfmp_customers %>% group_by(cluster) %>% count(gender)
ggplot(clust_gender, aes(cluster, n, fill = gender)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 3500, by = 140),
    limits = c(0, 3500)) +
  labs(title = "CLUSTERS AND GENDER", x = "")
```

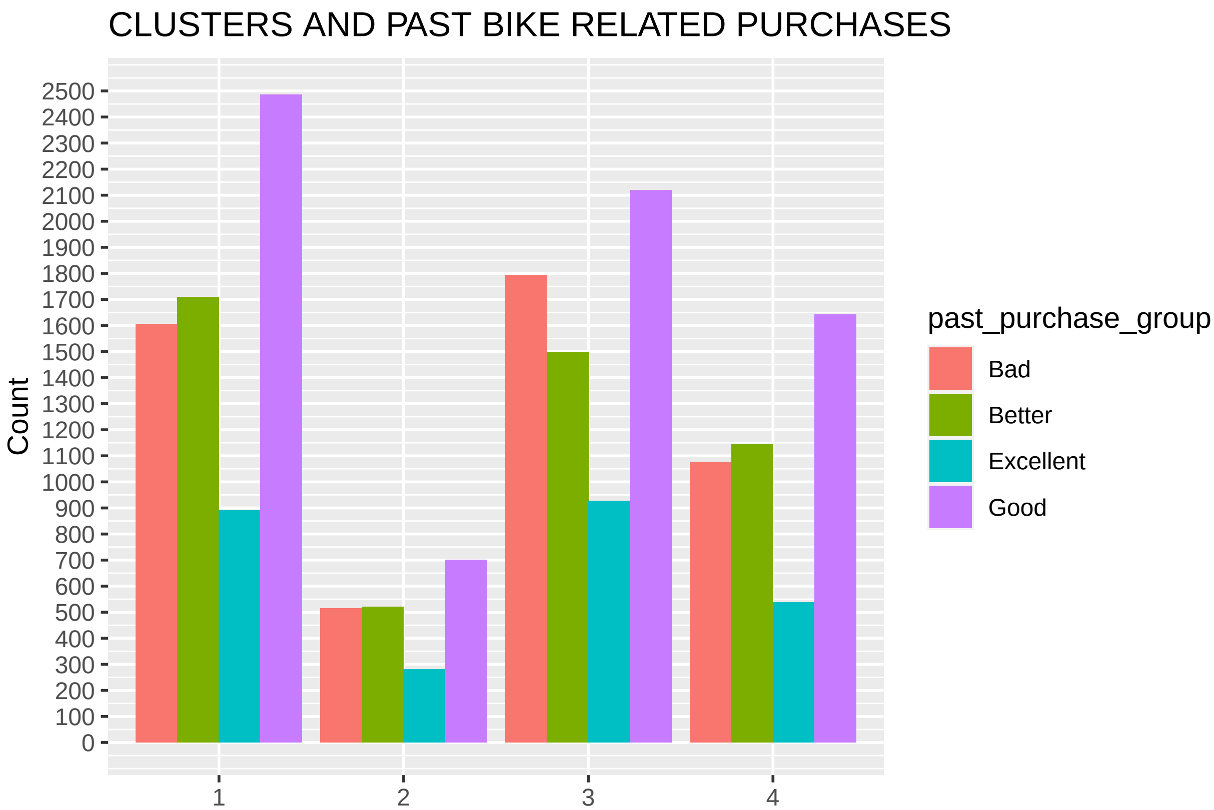


1.3 Past Bike Related Purchases

```
lrfmp_customers %>% count(past_purchase_group, sort = T) %>%
  mutate(percent = round(n / sum(n) * 100))
```

```
##   past_purchase_group    n percent
## 1                Good 6946      36
## 2                 Bad 4991      26
## 3              Better 4870      25
## 4           Excellent 2637      14
```

```
clust_past <- lrfmp_customers %>% group_by(cluster) %>%
  count(past_purchase_group)
ggplot(clust_past, aes(cluster, n, fill = past_purchase_group)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 2500, by = 100),
    limits = c(0, 2500)) +
  labs(title = "CLUSTERS AND PAST BIKE RELATED PURCHASES", x = "")
```



Visits by those in the Good category were always higher across the clusters

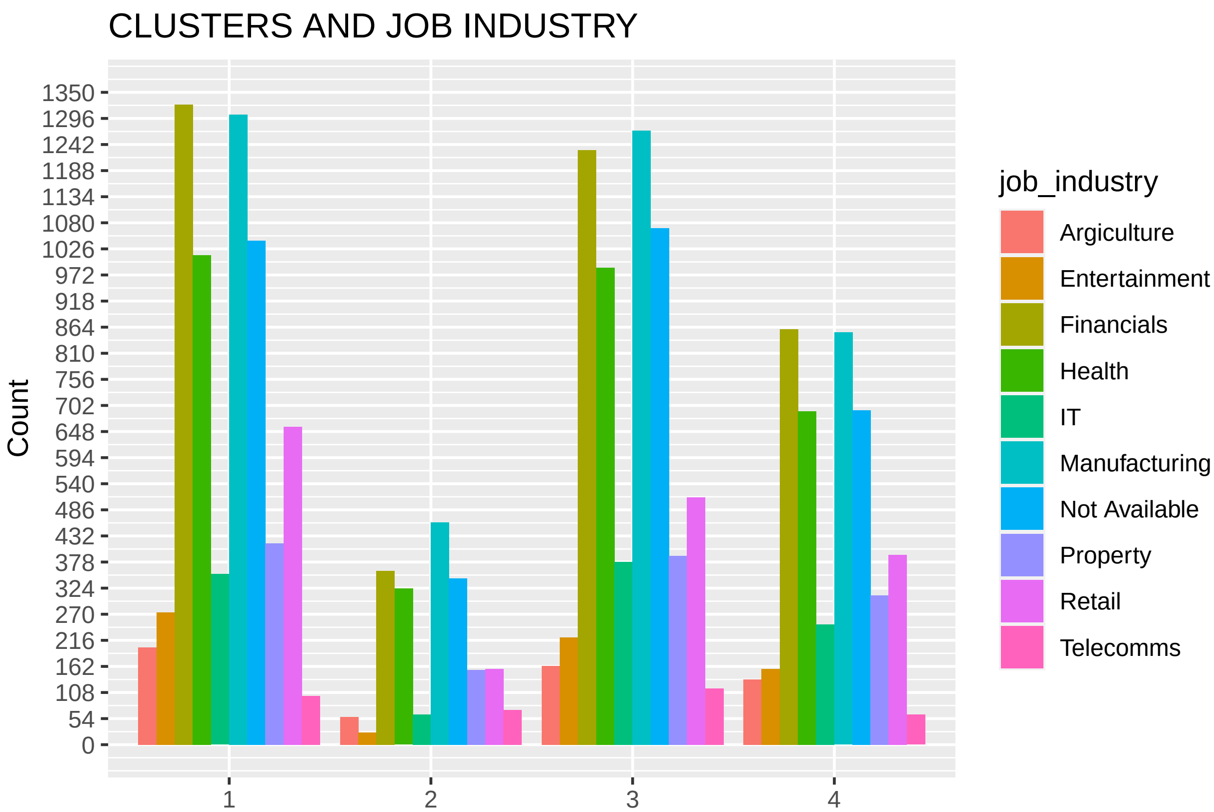
Across all genders customers seemed to belong in the same cluster.

1.4 Job Industry

```
lrfmp_customers %>% count(job_industry, sort = T) %>%
  mutate(percent = round(n / sum(n) * 100))
```

```
##      job_industry    n percent
## 1  Manufacturing 3888      20
## 2     Financials 3775      19
## 3  Not Available 3148      16
## 4         Health 3013      15
## 5         Retail 1720       9
## 6       Property 1272       7
## 7             IT 1043       5
## 8  Entertainment  678       3
## 9    Argiculture  556       3
## 10     Telecomms  351       2
```

```
clust_job <- lrfmp_customers %>% group_by(cluster) %>% count(job_industry)
ggplot(clust_job, aes(cluster, n, fill = job_industry)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 1350, by = 54),
    limits = c(0, 1350)) +
  labs(title = "CLUSTERS AND JOB INDUSTRY", x = "")
```



Customers from Financial Services, Health industry, Manufacturing and those that did not identify their industry were always the largest visitors across the 4 clusters.

1.5 Wealth Segment

```
lrfmp_customers %>% count(wealth_segment, sort = T) %>%
  mutate(percent = round(n / sum(n) * 100))
```

```
##   wealth_segment    n percent
## 1           Mass 9802      50
## 2       High Net 4929      25
## 3       Affluent 4713      24
```

50% of the visits were by mass cutomers while high net worth and affluent were at 25% and 24% respectively

```
clust_wealth <- lrfmp_customers %>% group_by(cluster) %>% count(wealth_segment)
ggplot(clust_wealth, aes(cluster, n, fill = wealth_segment)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 3550, by = 142),
    limits = c(0, 3550)) +
  labs(title = "CLUSTERS AND WEALTH SEGMENT", x = "")
```

Mass customers always had more visits across the clusters while high net worth and affluent had almost the same number of visits across the clusters.

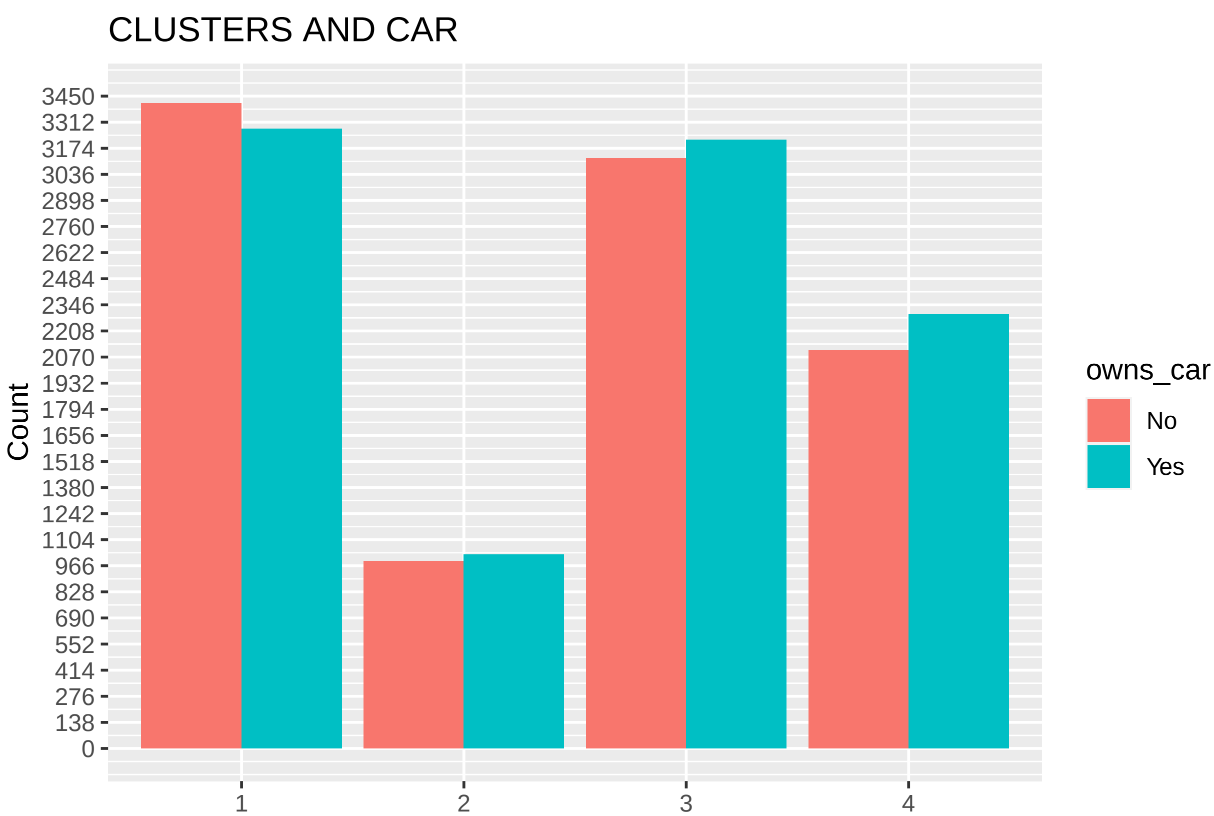
1.6 Owns Car

```
lrfmp_customers %>% count(owns_car, sort = T) %>%
  mutate(percent = round(n / sum(n) * 100))
```

```
##   owns_car    n percent
## 1      Yes 9815      50
## 2       No 9629      50
```

50% owned while 50% did not own.

```
clust_car <- lrfmp_customers %>% group_by(cluster) %>% count(owns_car)
ggplot(clust_car, aes(cluster, n, fill = owns_car)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 3450, by = 138),
    limits = c(0, 3450)) +
  labs(title = "CLUSTERS AND CAR", x = "")
```



Car ownership was always almost a 50-50 afffair across the clusters.

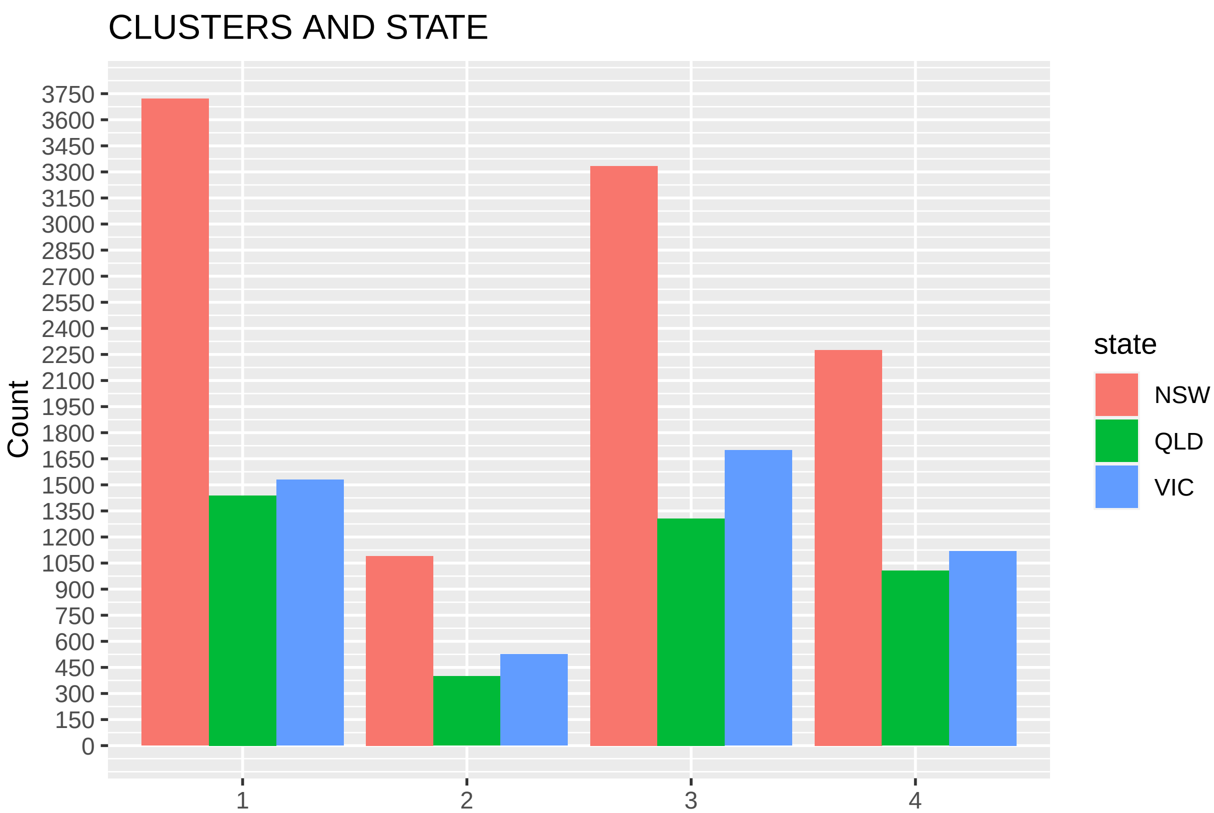
1.7 State

```
lrfmp_customers %>% count(state, sort = T) %>%
  mutate(percent = round(n / sum(n) * 100))
```

```
##   state     n percent
## 1   NSW 10421      54
## 2   VIC  4873      25
## 3   QLD  4150      21
```

54% of the visits were by customers from NSW state while VIC and QLD were at 25% and 21% respectively.

```
clust_state <- lrfmp_customers %>% group_by(cluster) %>% count(state)
ggplot(clust_state, aes(cluster, n, fill = state)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 3750, by = 150),
    limits = c(0, 3750)) +
  labs(title = "CLUSTERS AND STATE", x = "")
```



Visits by NWS stators were always the largest across the clusters.

Property Valuation is coded with digits from 1 to 12 with no missing digit

```
class(lrfmp_customers$property_valuation)
```

```
## [1] "factor"
```

```r
lrfmp_customers$property_valuation <- as.numeric(lrfmp_customers$property_valuation)
range(lrfmp_customers$property_valuation)
```

```
## [1]  1 12
```

```r
setdiff(1:12, lrfmp_customers$property_valuation)
```

```
## integer(0)
```

```r
summary(lrfmp_customers$property_valuation)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    6.00    8.00    7.51   10.00   12.00
```

We can code the different property valuation to 3 categories

```r
lrfmp_customers <- lrfmp_customers %>%
  mutate(pvaluation_group = case_when(
    property_valuation <= 3 ~ "Minimum",
    property_valuation > 3 & property_valuation <= 7 ~ "Average",
    property_valuation > 7 ~ "Wealthy"
  ))
```

1.8 Property Valuation

```r
lrfmp_customers %>% count(pvaluation_group, sort = T) %>%
  mutate(percent = round(n / sum(n) * 100))
```

```
##   pvaluation_group     n percent
## 1          Wealthy 11494      59
## 2          Average  5610      29
## 3          Minimum  2340      12
```

59% of the visits were from the Wealthy while Average and Minimum were at 29% and 12% respectively.

```r
clust_valuation <- lrfmp_customers %>% group_by(cluster) %>%
  count(pvaluation_group)
ggplot(clust_valuation, aes(cluster, n, fill = pvaluation_group)) +
```

```r
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous("Count",
    breaks = seq(0, 4000, by = 160),
    limits = c(0, 4000)) +
  labs(title = "CLUSTERS AND PROPERTY VALUATION", x = "")
```

The Wealthy always had more visits per cluster

We can therefore say that the following categories could make regular customers or loyal customers;

- Middle aged individuals-aged 36-55
- Those working in the Financial services, Health, Manufacturing industry and unknown
- Those categorized as Mass Customers in the Wealth Segment
- Those from NWS State
- Those with a past bike related purchases of Good
- And those with a property valuation of Wealthy.

4 NEW CUSTOMER LIST

WE will filter the list with the conditions above.

```
newcustomerlist_1 <- newcustomerlist %>% select(1:16)
```

Missing values

```
sum(is.na(newcustomerlist_1))
```

```
## [1] 152
```

Columns with missing values

```
names(which(colSums(is.na(newcustomerlist)) > 0))
```

```
## [1] "last_name" "DOB"       "job_title"
```

We can work with the missing last_name, DOB and job_title but will fill the age and age_group

We create age groups, past 3 years related purchases group and property valuation groups.

Create a new data frame that age will not change as of today

```
newcustomerlist_2 <- newcustomerlist_1 %>% mutate(
  age = trunc((DOB %--% today())/ years(1))
)
```

```
write_csv(newcustomerlist_2, "newcustomerlist_2.csv")
```

age and age groups

```
newcustomerlist_3 <- read_csv("newcustomerlist_2.csv")
newcustomerlist_4 <- newcustomerlist_3 %>%
  mutate(age_group = case_when(
    age <= 35 ~ "Youth",
    age > 35 & age <= 55 ~ "Middle",
```

```
    age > 55 ~ "Older"
  ))
newcustomerlist_4$age_group[is.na(newcustomerlist_4$age_group)] <- "unidentified"
```

past bike related purchases

```
newcustomerlist_4 <- newcustomerlist_4 %>%
  rename(past_purchases = past_3_years_bike_related_purchases,
         job_industry = job_industry_category,
         deceased = deceased_indicator,
         dob = DOB)
```

```
newcustomerlist_4 <- newcustomerlist_4 %>%
  mutate(past_purchase_group = case_when(
    past_purchases <= 24 ~ "Bad",
    past_purchases > 24 & past_purchases <= 59 ~ "Good",
    past_purchases > 59 & past_purchases <= 84 ~ "Better",
    past_purchases >= 85 ~ "Excellent"
  ))
```

We can code the different property valuation to 3 categories

```
newcustomerlist_4 <- newcustomerlist_4 %>%
  mutate(pvaluation_group = case_when(
    property_valuation <= 3 ~ "Minimum",
    property_valuation > 3 & property_valuation <= 7 ~ "Average",
    property_valuation > 7 ~ "Wealthy"
  ))
```

missing values

```
sum(is.na(newcustomerlist_4))
```

```
## [1] 169
```

```
names(which(colSums(is.na(newcustomerlist_4)) > 0))
```

```
## [1] "last_name" "dob"       "job_title" "age"
```

age_group

```r
newcustomerlist_4 %>% count(age_group, sort = T)
```

```
## # A tibble: 4 x 2
##   age_group        n
##   <chr>        <int>
## 1 Older          427
## 2 Middle         344
## 3 Youth          212
## 4 unidentified    17
```

job industry

```r
newcustomerlist_4 %>% count(job_industry, sort = T)
```

```
## # A tibble: 10 x 2
##    job_industry           n
##    <chr>              <int>
##  1 Financial Services   203
##  2 Manufacturing        199
##  3 n/a                  165
##  4 Health               152
##  5 Retail                78
##  6 Property              64
##  7 IT                    51
##  8 Entertainment         37
##  9 Argiculture           26
## 10 Telecommunications    25
```

We hane n/a in job_industry, replace with unknown

```r
newcustomerlist_4$job_industry[newcustomerlist_4$job_industry == "n/a"] <- "unknown"
newcustomerlist_4 %>% count(job_industry, sort = T)
```

```
## # A tibble: 10 x 2
##    job_industry           n
##    <chr>              <int>
##  1 Financial Services   203
##  2 Manufacturing        199
##  3 unknown              165
##  4 Health               152
```

```
##   5 Retail                 78
##   6 Property               64
##   7 IT                     51
##   8 Entertainment          37
##   9 Argiculture            26
##  10 Telecommunications     25
```

wealth segment

```r
newcustomerlist_4 %>% count(wealth_segment, sort = T)
```

```
## # A tibble: 3 x 2
##   wealth_segment        n
##   <chr>             <int>
## 1 Mass Customer       508
## 2 High Net Worth      251
## 3 Affluent Customer   241
```

State

```r
newcustomerlist_4 %>% count(state, sort = T)
```

```
## # A tibble: 3 x 2
##   state     n
##   <chr> <int>
## 1 NSW     506
## 2 VIC     266
## 3 QLD     228
```

Past 3 years bike related purchase

```r
newcustomerlist_4 %>% count(past_purchase_group, sort = T)
```

```
## # A tibble: 4 x 2
##   past_purchase_group     n
##   <chr>               <int>
## 1 Good                  366
## 2 Better                281
## 3 Bad                   229
## 4 Excellent             124
```

Property Valuation

```
newcustomerlist_4 %>% count(pvaluation_group, sort = T)
```

```
## # A tibble: 3 x 2
##   pvaluation_group     n
##   <chr>            <int>
## 1 Wealthy            559
## 2 Average            318
## 3 Minimum            123
```

4.1 The List

```
newcustomerlist_5 <- newcustomerlist_4
newcustomerlist_5 <- newcustomerlist_5 %>%
  filter(age_group == "Middle"|job_industry =="Manufacturing"|job_industry =="Financial Servic
```

```
dim(newcustomerlist_5)
```

```
## [1] 986  20
```

```
write_csv(newcustomerlist_5, "customerfocuslist.csv")
```

```
write_csv(lrfmp_customers, "lrfmp_customers.csv")
```