KPMG

Yini Lai

20/01/2021

Contents


```r
library(knitr)
library(tidyverse)
library(readxl)
library(visdat)
library(scales)
library(sjmisc)
library(rfm)
```

Introduction

```r
# Loading Data

Transactions <- read_excel("~/Documents/Projects/KPMG/KPMG_VI_New_raw_data_update_final.xlsx",
    sheet = "Transactions", skip = 1)

NewCustomerList <- read_excel("~/Documents/Projects/KPMG/KPMG_VI_New_raw_data_update_final.xlsx",
    sheet = "NewCustomerList", skip = 1)

CustomerDemographic <- read_excel("~/Documents/Projects/KPMG/KPMG_VI_New_raw_data_update_final.xlsx",
    sheet = "CustomerDemographic", skip = 1)

CustomerAdress <- read_excel("~/Documents/Projects/KPMG/KPMG_VI_New_raw_data_update_final.xlsx",
    sheet = "CustomerAddress", skip = 1)
```

, col_types = c("text", "text", "text", "numeric", "numeric", "text", "text", "text","text", "text", "numeric", "text", "text", "text", "text", "numeric","numeric", "numeric","numeric", "numeric", "numeric", "numeric", "numeric")

Data Wrangling

Transactions

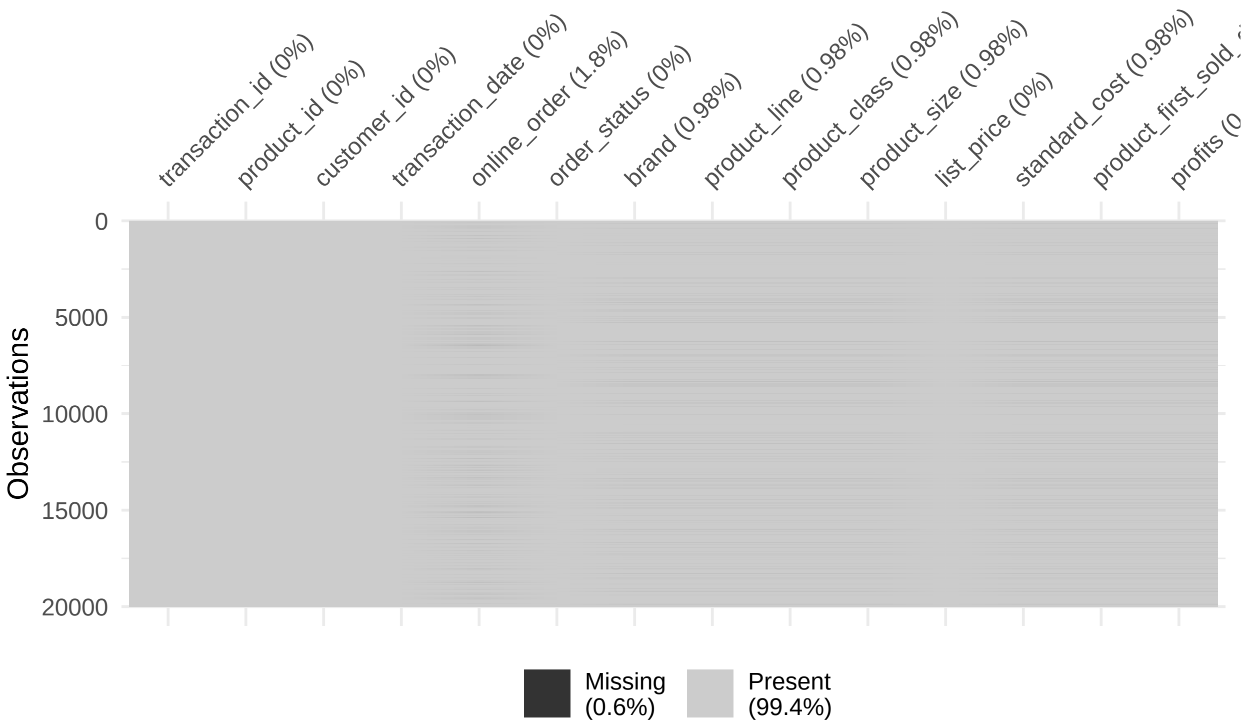
Accuracy

Create a profit column is helpful in checking the data accuracy issue with standard cost and list_price as we can figure out whether there is a negative profit or some of the profits are lower than what we expected.

```
Transactions <- Transactions %>% mutate(profits = list_price - standard_cost)
```

Completeness

```
vis_miss(Transactions)
```



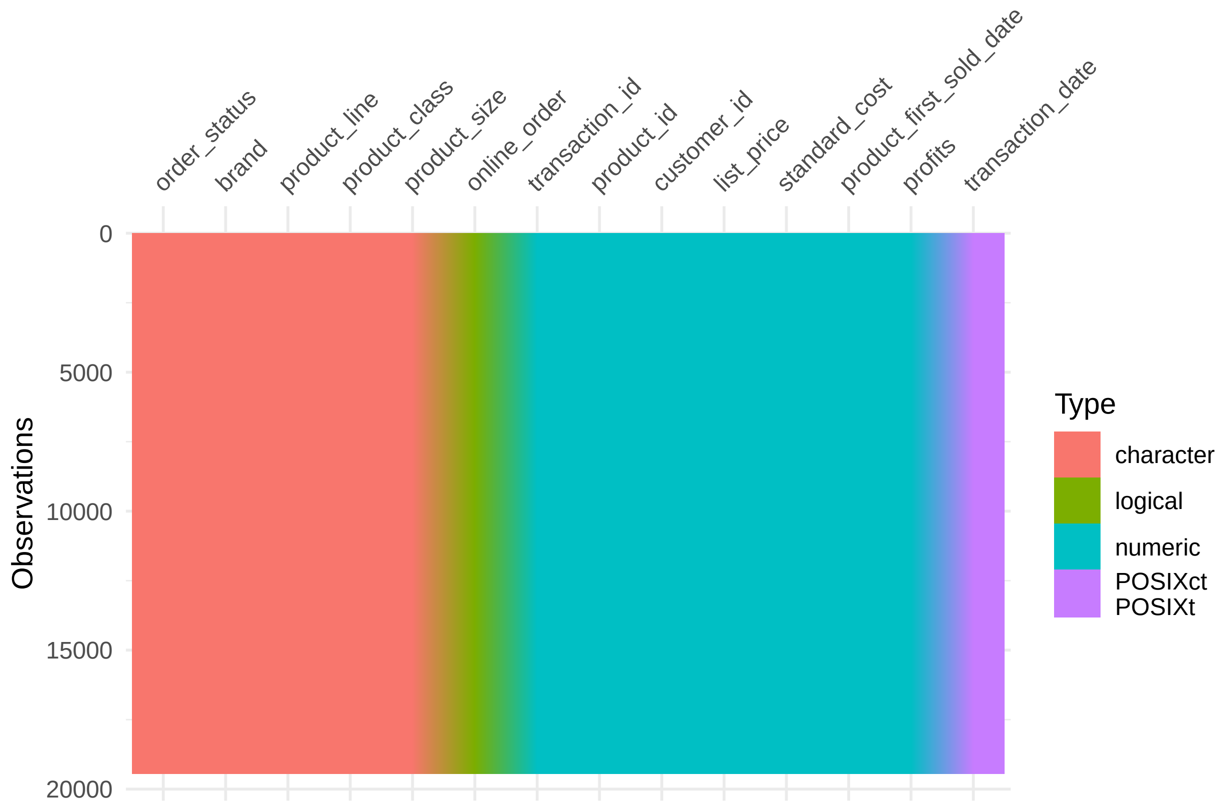
Since there is not too many missing values (0.6%), we can directly remove them from the data set.

```
Transactions <- na.omit(Transactions)
```

Consistency, Relevancy, Validity

Look at the data types within the transaction dataset.

```
vis_dat(Transactions)
```



Raw Data Types

Character: order_status, brand, product_line, product_class, product_size

These variables are in the correct data type, but better to convert them to factor for further analysis.

Logical: online_order

Correct

Numeric: transaction_id, product_id, customer_id, list_price, standard_cost, product_first_sold_date

- transaction_id & product_id & customer_id: better to be presented in character format
- list_price & standard_cost: should be presented in currency format **Validity**
- product_first_sold_date: should be in date format **Validity**

POSIXct: transaction_date

Correct, can convert to date to keep consistent

```
# Convert to factor for further analysis

Transactions$order_status <- as.factor(Transactions$order_status)
Transactions$brand <- as.factor(Transactions$brand)
Transactions$product_line <- as.factor(Transactions$product_line)
Transactions$product_class <- as.factor(Transactions$product_class)
Transactions$product_size <- as.factor(Transactions$product_size)
```

```
# Convert transaction_id & product_id & customer_id to character

Transactions$transaction_id <- as.character(Transactions$transaction_id)
Transactions$product_id <- as.character(Transactions$product_id)
Transactions$customer_id <- as.character(Transactions$customer_id)
```

```r
# Validity

# Convert list_price & standard_cost to currency format
Transactions$list_price <- dollar_format()(c(Transactions$list_price ))
Transactions$standard_cost <- dollar_format()(c(Transactions$standard_cost))
```

```r
# Validity

# Convert product_first_sold_date to Date format
Transactions$product_first_sold_date <- as.Date(Transactions$product_first_sold_date, origin = "1899-12-
```

```r
# Convert transaction_date to Date format
Transactions$transaction_date <- as.Date(Transactions$transaction_date)
```

Summary the data to have a overview of the data set

```r
summary(Transactions)
```

```
##   transaction_id      product_id        customer_id        transaction_date
##   Length:19445       Length:19445       Length:19445       Min.   :2017-01-01
##   Class :character   Class :character   Class :character   1st Qu.:2017-04-01
##   Mode  :character   Mode  :character   Mode  :character   Median :2017-07-03
##                                                            Mean   :2017-07-01
##                                                            3rd Qu.:2017-10-02
##                                                            Max.   :2017-12-30
##   online_order       order_status                 brand        product_line
##   Mode :logical   Approved :19273   Giant Bicycles:3244   Mountain:  418
##   FALSE:9706      Cancelled:  172   Norco Bicycles:2863   Road    : 3894
##   TRUE :9739                        OHM Cycles    :2993   Standard:13920
##                                     Solex         :4169   Touring : 1213
##                                     Trek Bicycles :2931
##                                     WeareA2B      :3245
##   product_class  product_size    list_price         standard_cost
##   high   : 2952  large : 3900   Length:19445       Length:19445
##   low    : 2906  medium:12767   Class :character   Class :character
##   medium:13587   small : 2778   Mode  :character   Mode  :character
##
##
##
##   product_first_sold_date    profits
##   Min.   :1991-01-21      Min.   :   4.8
##   1st Qu.:1997-08-25      1st Qu.: 133.8
##   Median :2004-08-17      Median : 445.2
##   Mean   :2004-08-02      Mean   : 551.8
##   3rd Qu.:2011-05-09      3rd Qu.: 830.2
##   Max.   :2016-12-06      Max.   :1702.5
```

From the summary result, we can see that there is no consistency issue since every element of each variable is recorded in the same way. However, there is a **relevancy** issue, since from the order_status, we can see that some of the orders had been canceled. Thus, we need to remove those canceled orders.

```r
Transactions <- Transactions %>% filter(order_status == "Approved")
```

Uniqueness

```
Transactions_duplicate <- Transactions %>% data.frame() %>% distinct()
```

```
dim(Transactions)[1]
```

```
## [1] 19273
```

```
dim(Transactions_duplicate)[1]
```

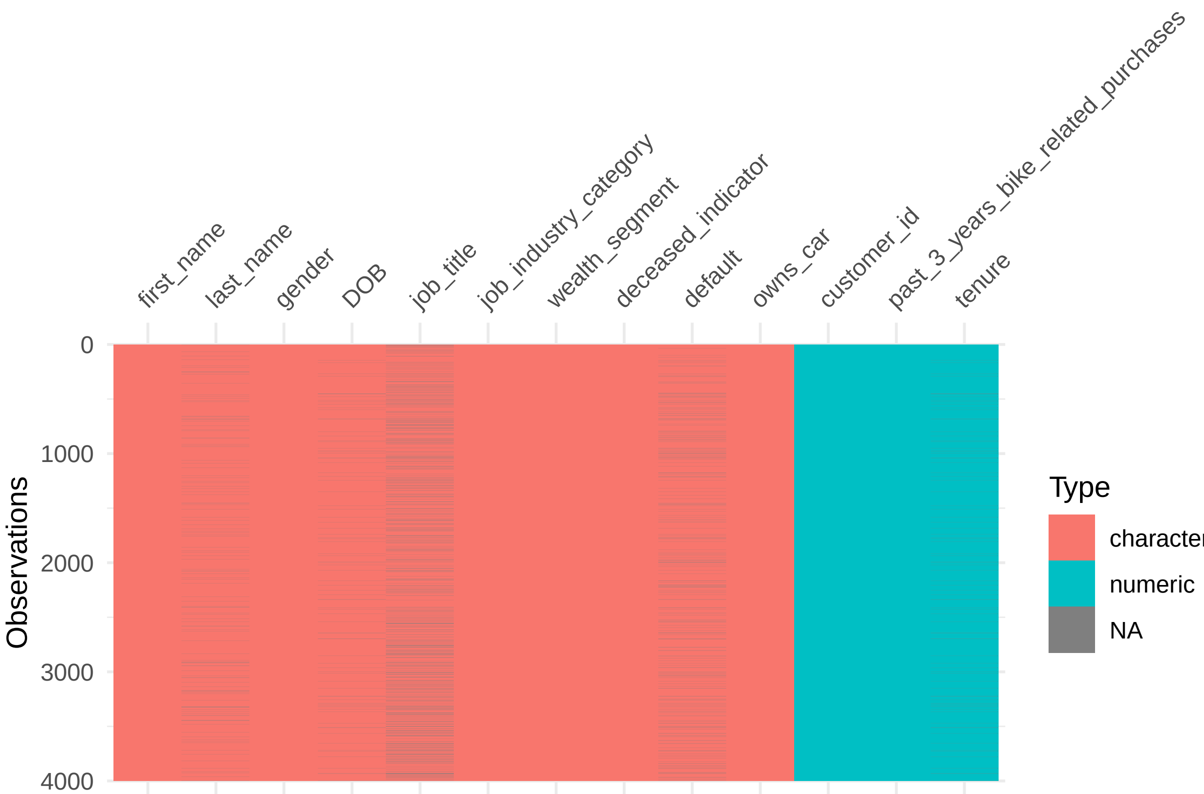
```
## [1] 19273
```

There is no duplicate rows in this dataset.

CustomerDemographic

Consistency, Relevancy, Validity

```
vis_dat(CustomerDemographic)
```



Most of the variables in this dataset are in correct types except DOB which should be in date format.

```
# Check whether there is any date like data in DOB column

index <- c()
for (i in 1:dim(CustomerDemographic)[1]) {
  if(str_contains(CustomerDemographic$DOB[i], c("-"))){
   index <- append(index, i)
  }else next
}
```

```
CustomerDemographic[index, "DOB"]
```

```
## # A tibble: 1 x 1
##   DOB
##   <chr>
## 1 1843-12-21
```

It's a bit strange that a customer was born in 1843 in this case. Thus we can remove this row as it may be considered as an outlier. This is an **accuracy issue**.

Except row 34, all the other values in column "DOB" are not in Date format. DOB should be in date format rather than character.

```
# Validity

CustomerDemographic$DOB <- as.Date(as.numeric(CustomerDemographic$DOB), origin = "1899-12-30")
```

```
# Convert some the character format variables to factor for further analysis

CustomerDemographic$gender <- as.factor(CustomerDemographic$gender)
CustomerDemographic$job_title <- as.factor(CustomerDemographic$job_title)
CustomerDemographic$job_industry_category <- as.factor(CustomerDemographic$job_industry_category)
CustomerDemographic$wealth_segment <- as.factor(CustomerDemographic$wealth_segment)
CustomerDemographic$deceased_indicator <- as.factor(CustomerDemographic$deceased_indicator)
CustomerDemographic$owns_car <- as.factor(CustomerDemographic$owns_car)
```

```
head(CustomerDemographic)
```

```
## # A tibble: 6 x 13
##   customer_id first_name     last_name gender past_3_years_bi~ DOB        job_title
##         <dbl> <chr>          <chr>     <fct>             <dbl> <date>     <fct>
## 1           1 Laraine        Medendorp F                    93 1953-10-12 Executiv~
## 2           2 Eli            Bockman   Male                 81 1980-12-16 Administ~
## 3           3 Arlin          Dearle    Male                 61 1954-01-20 Recruiti~
## 4           4 Talbot         <NA>      Male                 33 1961-10-03 <NA>
## 5           5 Sheila-kathryn Calton    Female               56 1977-05-13 Senior E~
## 6           6 Curr           Duckhouse Male                 35 1966-09-16 <NA>
## # ... with 6 more variables: job_industry_category <fct>, wealth_segment <fct>,
## #   deceased_indicator <fct>, default <chr>, owns_car <fct>, tenure <dbl>
```

```
summary(CustomerDemographic)
```

```
##   customer_id    first_name         last_name            gender
##  Min.   :   1   Length:4000        Length:4000        F     :   1
##  1st Qu.:1001   Class :character   Class :character   Femal :   1
##  Median :2000   Mode  :character   Mode  :character   Female:2037
##  Mean   :2000                                         M     :   1
##  3rd Qu.:3000                                         Male  :1872
##  Max.   :4000                                         U     :  88
##
##  past_3_years_bike_related_purchases      DOB
##  Min.   : 0.00                       Min.   :1931-10-23
##  1st Qu.:24.00                       1st Qu.:1968-01-25
##  Median :48.00                       Median :1977-07-25
##  Mean   :48.89                       Mean   :1977-07-25
##  3rd Qu.:73.00                       3rd Qu.:1987-02-28
##  Max.   :99.00                       Max.   :2002-03-11
```

```
##                                     NA's   :88
##                                job_title           job_industry_category
##  Business Systems Development Analyst:  45   Manufacturing     :799
##  Social Worker                       :  44   Financial Services:774
##  Tax Accountant                      :  44   n/a               :656
##  Internal Auditor                    :  42   Health            :602
##  Legal Assistant                     :  41   Retail            :358
##  (Other)                             :3278   Property          :267
##  NA's                                : 506   (Other)           :544
##           wealth_segment deceased_indicator   default           owns_car
##  Affluent Customer: 979   N:3998             Length:4000        No :1976
##  High Net Worth   :1021   Y:   2             Class :character   Yes:2024
##  Mass Customer    :2000                      Mode  :character
##
##
##
##
##      tenure
##  Min.   : 1.00
##  1st Qu.: 6.00
##  Median :11.00
##  Mean   :10.66
##  3rd Qu.:15.00
##  Max.   :22.00
##  NA's   :87
```

From the summary, we can see that for gender column, there are three ways in recording female. Thus we should make some adjustment on them to keep consistent.

```r
# Consistency issue in Gender column

CustomerDemographic <- CustomerDemographic %>%
  mutate(gender = case_when(
    gender == 'F' ~ 'Female',
    gender == 'Femal' ~ 'Female',
    gender == 'Female' ~ 'Female',
    gender == 'M' ~ 'Male',
    gender == 'Male' ~ 'Male',
    gender == 'U' ~ 'U'
  ))
CustomerDemographic$gender <- as.factor(CustomerDemographic$gender)
```

May also need to investigate variable job_title and job_industry_category

```r
unique(CustomerDemographic$job_title)
```

```
##   [1] Executive Secretary                  Administrative Officer
##   [3] Recruiting Manager                   <NA>
##   [5] Senior Editor                        Media Manager I
##   [7] Business Systems Development Analyst Senior Quality Engineer
##   [9] Nuclear Power Engineer               Developer I
##  [11] Account Executive                    Junior Executive
##  [13] Media Manager IV                     Sales Associate
##  [15] Professor                            Geological Engineer
##  [17] Project Manager                      Safety Technician I
##  [19] Research Assistant I                 Accounting Assistant III
```

```
##  [21] Editor                             Research Nurse
##  [23] Safety Technician III              Staff Accountant III
##  [25] Legal Assistant                    Product Engineer
##  [27] Information Systems Manager        VP Quality Control
##  [29] Social Worker                      Senior Cost Accountant
##  [31] Assistant Media Planner            Payment Adjustment Coordinator
##  [33] Food Chemist                       Accountant III
##  [35] Director of Sales                  Senior Financial Analyst
##  [37] Registered Nurse                   Biostatistician II
##  [39] Computer Systems Analyst II        Software Test Engineer II
##  [41] Paralegal                          VP Sales
##  [43] Chief Design Engineer              Office Assistant III
##  [45] Physical Therapy Assistant         Help Desk Operator
##  [47] Web Developer II                   Research Associate
##  [49] Teacher                            VP Product Management
##  [51] Statistician II                    Automation Specialist IV
##  [53] Data Coordiator                    Software Test Engineer III
##  [55] Internal Auditor                   Analyst Programmer
##  [57] Occupational Therapist             Speech Pathologist
##  [59] Quality Control Specialist         Civil Engineer
##  [61] Software Engineer III              Community Outreach Specialist
##  [63] Safety Technician IV               VP Accounting
##  [65] General Manager                    Nurse Practicioner
##  [67] Automation Specialist II           Marketing Assistant
##  [69] Marketing Manager                  Staff Scientist
##  [71] Assistant Professor                Budget/Accounting Analyst IV
##  [73] Associate Professor                Graphic Designer
##  [75] Administrative Assistant II        Compensation Analyst
##  [77] Systems Administrator III          Financial Advisor
##  [79] Chemical Engineer                  Web Designer I
##  [81] Senior Developer                   Office Assistant II
##  [83] Recruiter                          Operator
##  [85] Programmer Analyst III             Quality Engineer
##  [87] Environmental Tech                 Analog Circuit Design manager
##  [89] Cost Accountant                    Librarian
##  [91] Structural Analysis Engineer       Pharmacist
##  [93] Assistant Manager                  Accountant I
##  [95] Web Designer III                   Geologist III
##  [97] Software Test Engineer I           Structural Engineer
##  [99] Safety Technician II               Web Developer III
## [101] Programmer Analyst II              Design Engineer
## [103] Statistician I                     VP Marketing
## [105] Desktop Support Technician         Actuary
## [107] Database Administrator III         Electrical Engineer
## [109] Tax Accountant                     Clinical Specialist
## [111] Database Administrator IV          Systems Administrator II
## [113] Account Coordinator                Programmer III
## [115] Administrative Assistant III       Nurse
## [117] Technical Writer                   Staff Accountant II
## [119] Dental Hygienist                   Sales Representative
## [121] Budget/Accounting Analyst III      Computer Systems Analyst IV
## [123] Geologist I                        Financial Analyst
## [125] Accounting Assistant II            Senior Sales Associate
## [127] Database Administrator II          Engineer I
```

```
## [129] Budget/Accounting Analyst I          Developer IV
## [131] Database Administrator I             Environmental Specialist
## [133] Computer Systems Analyst I           Account Representative IV
## [135] Statistician IV                      Human Resources Manager
## [137] GIS Technical Architect              Programmer IV
## [139] Accounting Assistant IV              Software Engineer IV
## [141] Programmer II                        Engineer III
## [143] Software Consultant                  Biostatistician IV
## [145] Help Desk Technician                 Automation Specialist I
## [147] Developer III                        Human Resources Assistant I
## [149] Geologist IV                         Media Manager II
## [151] Statistician III                     Engineer II
## [153] Health Coach II                      Developer II
## [155] Systems Administrator I              Web Developer I
## [157] Software Engineer II                 Accounting Assistant I
## [159] Research Assistant II                Programmer Analyst IV
## [161] Health Coach I                       Accountant II
## [163] Automation Specialist III            Administrative Assistant I
## [165] Health Coach IV                      Media Manager III
## [167] Account Representative III           Web Designer IV
## [169] Budget/Accounting Analyst II         Web Developer IV
## [171] Programmer I                         Biostatistician III
## [173] Software Test Engineer IV            Research Assistant IV
## [175] Account Representative I             Accountant IV
## [177] Biostatistician I                    Human Resources Assistant IV
## [179] Administrative Assistant IV          Office Assistant I
## [181] Human Resources Assistant II         Mechanical Systems Engineer
## [183] Engineer IV                          Health Coach III
## [185] Office Assistant IV                  Software Engineer I
## [187] Human Resources Assistant III        Staff Accountant I
## [189] Computer Systems Analyst III         Geologist II
## [191] Web Designer II                      Staff Accountant IV
## [193] Account Representative II            Programmer Analyst I
## [195] Systems Administrator IV             Research Assistant III
## 195 Levels: Account Coordinator Account Executive ... Web Developer IV
```

```
unique(CustomerDemographic$job_industry_category)
```

```
##  [1] Health             Financial Services Property           IT
##  [5] n/a                Retail             Argiculture        Manufacturing
##  [9] Telecommunications Entertainment
## 10 Levels: Argiculture Entertainment Financial Services Health ... Telecommunications
```

For these two variables, better to make a list which contains the most common job titles and most common job industry categories. For those rare job titles and job industry categories, we can add another option named "other". In this way, we can better classify and also avoid writing the same category into different ways.

From the first table which includes the first 6 rows of the data set, we can see that the default column contains irrelevant information. Thus, we can get rid of this column.

```
# Relevancy

CustomerDemographic <- CustomerDemographic %>% select(c(-default))
```

Currency

From the summary table, we can see that there were two customers deceased. Thus their information should be removed from the data set.

```
CustomerDemographic <- CustomerDemographic %>% filter(CustomerDemographic$deceased_indicator == "N")
```

Accuracy

From the summary table, we can see that most of the categorical variables seems reasonable. However, since the dataset only records Date of Birth, it is hard to figure out whether there are any outliers in this column. Except the row with DOB "1843-12-21" which had been considered as an outlier and should be excluded from the table. We need to further investigate the other DOB value. Thus it is better to create a new variable 'age' which is helpful in detecting outliers.

```
CustomerDemographic <- CustomerDemographic %>% mutate(Age = round((Sys.Date() - DOB)/365,2))
```

```
summary(as.numeric(CustomerDemographic$Age))
```

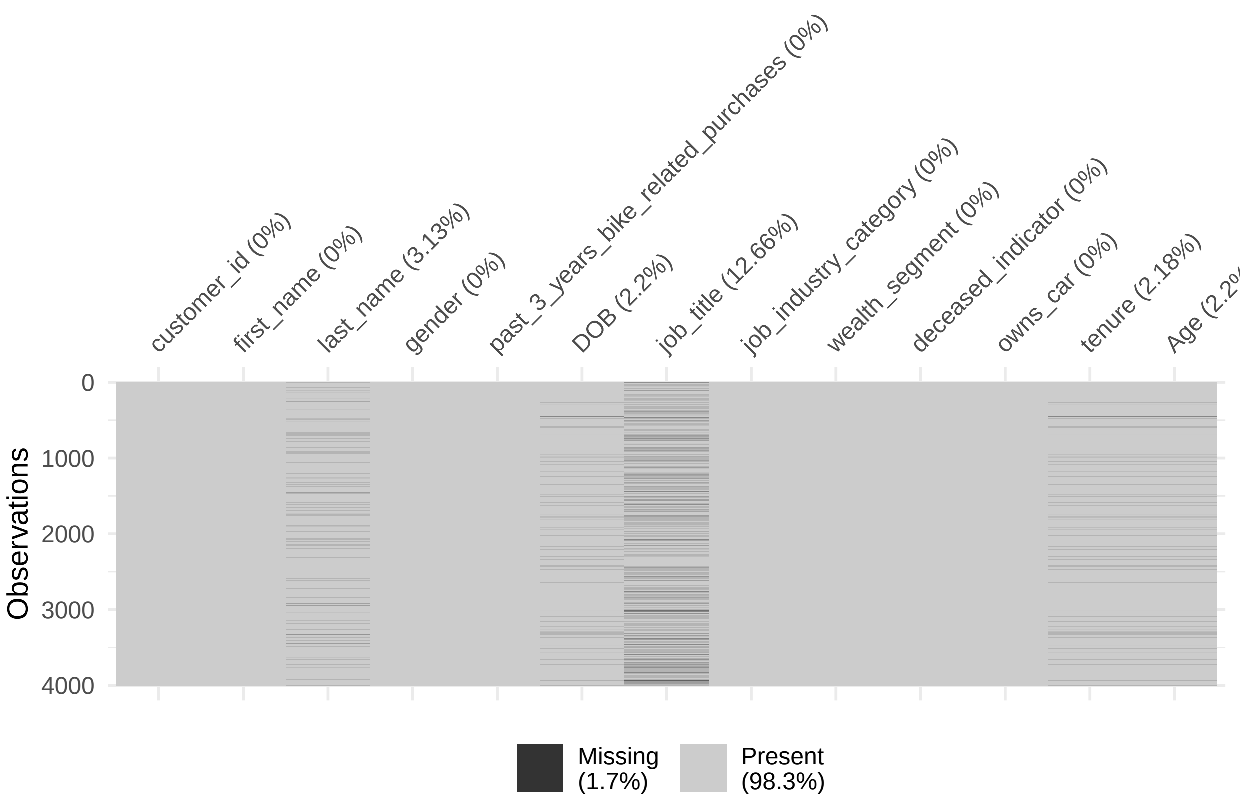
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   19.70   34.74   44.34   44.34   53.85   90.13      88
```

The range of age seems reasonable. The youngest one is nearly 19-year-old and the oldest one is nearly 90-year-old.

Recommendation: Create an Age column.

Completeness

```
vis_miss(CustomerDemographic)
```



From the summary table above, we can see that there are 656 "n/a" values in job_industry_category column. Since it takes a large proportion of the "job_industry_category" column, we may keep it for further analysis.

From the graph, we can see that there are 1.7 % missing value in this data set. The following columns contain missing values:

- last_name
- DOB
- job_title
- tenure
- Age

However, since we can distinguish the customer from their customer_id, we don't have to remove those observations with missing value in last_name column as this does not influence our analysis.

To mitigate this issue, we need to remove the observations that contain missing information.

```
CustomerDemographic <- CustomerDemographic[complete.cases(CustomerDemographic[, -c(3)]),]
```

Better to impute the missing value with some algorithms.

Uniqueness

```
CustomerDemographic_duplicate <- CustomerDemographic %>% data.frame() %>% distinct()
```

```
dim(CustomerDemographic)[1]
```

```
## [1] 3413
```

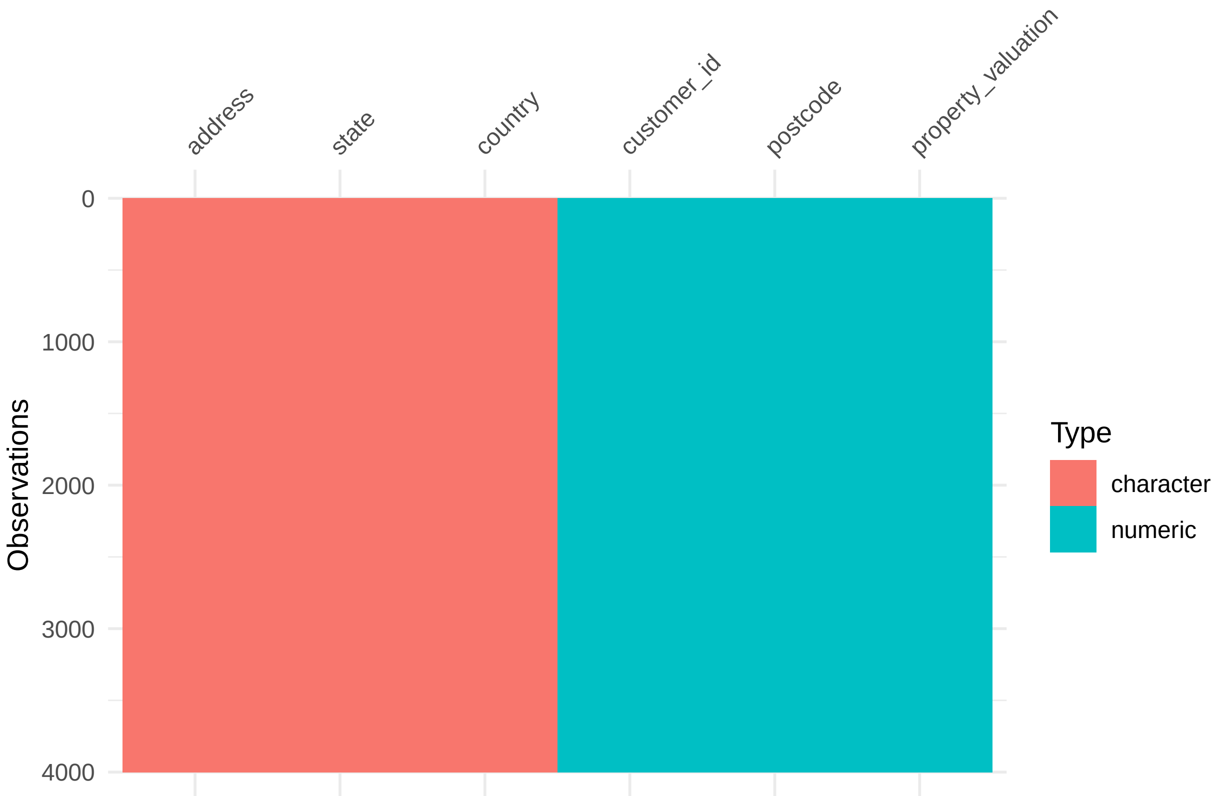
```
dim(CustomerDemographic_duplicate)[1]
```

```
## [1] 3413
```

CustomerAddress

Consistency, Relevancy, Validity

```
vis_dat(CustomerAdress)
```



There is no obvious data type issue (or validity issue) in this dataset.

```r
# Data type transformation for further investigation
CustomerAdress$state <- as.factor(CustomerAdress$state)
CustomerAdress$country <- as.factor(CustomerAdress$country)
CustomerAdress$postcode <- as.factor(CustomerAdress$postcode)
```

```r
head(CustomerAdress)
```

```
## # A tibble: 6 x 6
##   customer_id address            postcode state           country   property_valuat~
##         <dbl> <chr>              <fct>    <fct>           <fct>                <dbl>
## 1           1 060 Morning Avenue 2016     New South Wales Australia               10
## 2           2 6 Meadow Vale Court 2153     New South Wales Australia               10
## 3           4 0 Holy Cross Court 4211     QLD             Australia                9
## 4           5 17979 Del Mar Point 2448     New South Wales Australia                4
## 5           6 9 Oakridge Court    3216     VIC             Australia                9
## 6           7 4 Delaware Trail    2210     New South Wales Australia                9
```

```r
summary(CustomerAdress)
```

```
##   customer_id     address              postcode                   state
##  Min.   :   1   Length:3999        2170    :  31   New South Wales:  86
##  1st Qu.:1004   Class :character   2145    :  30   NSW            :2054
##  Median :2004   Mode  :character   2155    :  30   QLD            : 838
##  Mean   :2004                      2153    :  29   VIC            : 939
##  3rd Qu.:3004                      2560    :  26   Victoria       :  82
##  Max.   :4003                      2770    :  26
##                                    (Other):3827
##       country     property_valuation
```

```
##  Australia:3999   Min.   : 1.000
##                    1st Qu.: 6.000
##                    Median : 8.000
##                    Mean   : 7.514
##                    3rd Qu.:10.000
##                    Max.   :12.000
##
```

From the summary table, we can see that one of the states was recorded with abbreviation and some of the states were recorded with both full name and abbreviation. Thus the consistency issue exists.

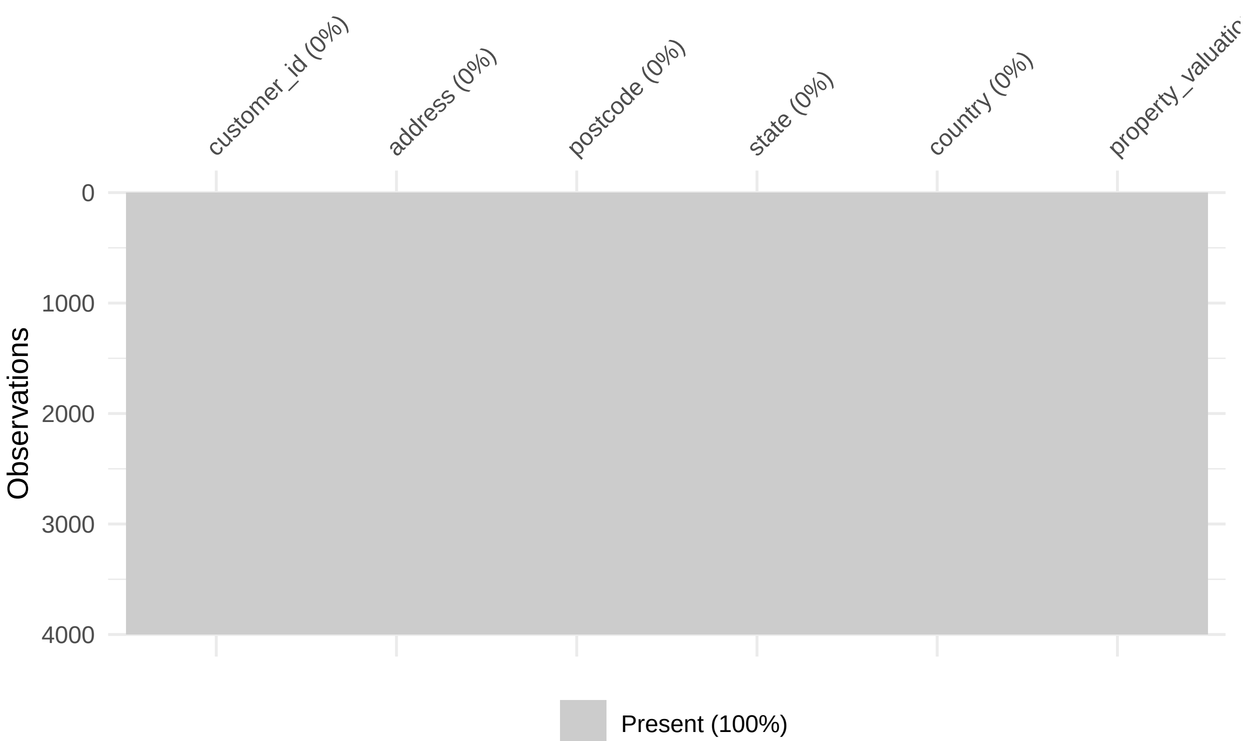
```
CustomerAdress <- CustomerAdress %>%
  mutate(state = case_when(
    state == 'New South Wales' ~ 'NSW',
    state == 'NSW' ~ 'NSW',
    state == 'QLD' ~ 'QLD',
    state == 'VIC' ~ 'VIC',
    state == 'Victoria' ~ 'VIC'
  ))
CustomerAdress$state <- as.factor(CustomerAdress$state)
```

Accuracy

From the summary table showing above, it seems there is no outlier in this dataset.

Completeness

```
vis_miss(CustomerAdress)
```



There is no missing value in this dataset

Uniqueness

```
CustomerAdress_duplicate <- CustomerAdress %>% data.frame() %>% distinct()
```

```
dim(CustomerAdress)[1]
```

```
## [1] 3999
```

```
dim(CustomerAdress_duplicate)[1]
```

```
## [1] 3999
```

Relation among three data sets.

```
Transactions$customer_id <- as.factor(Transactions$customer_id)
CustomerDemographic$customer_id <- as.factor(CustomerDemographic$customer_id)
CustomerAdress$customer_id <- as.factor(CustomerAdress$customer_id )
```

```
join_data <- Transactions %>% inner_join(CustomerDemographic, by = "customer_id") %>% inner_join(Custom
```

```
full_data <- Transactions %>% full_join(CustomerDemographic, by = "customer_id") %>% full_join(Customer
```

```
Transaction_customer <- list(unique(Transactions$customer_id))
CustomerDemographic_customer <- list(unique(CustomerDemographic$customer_id))
CustomerAdress_customer <- list(unique(CustomerAdress$customer_id))
full_customer <- list(unique(full_data$customer_id))
```

```
lengths(list(unique(join_data$customer_id)))
```

```
## [1] 2992
```

```
lengths(full_customer)
```

```
## [1] 4004
```

```
lengths(Transaction_customer)
```

```
## [1] 3490
```

```
lengths(CustomerDemographic_customer)
```

```
## [1] 3413
```

```
lengths(CustomerAdress_customer)
```

```
## [1] 3999
```

```
all(full_customer %in% Transaction_customer)
```

```
## [1] FALSE
```

```
all(full_customer %in% CustomerDemographic_customer)
```

```
## [1] FALSE
```

```
all(full_customer %in% CustomerAdress_customer)
```

```
## [1] FALSE
```

Only 2992 customers had completed information being recorded and none of the three datasets contains all the existed customer_id. However, this is not an issue for Transactions dataset as some of the customers may not have any transactions in the past 3 months. But this can be an issue for both “CustomerDemographic” and “CustomerAdress”. Thus customer_id is incompleted in these two data set.

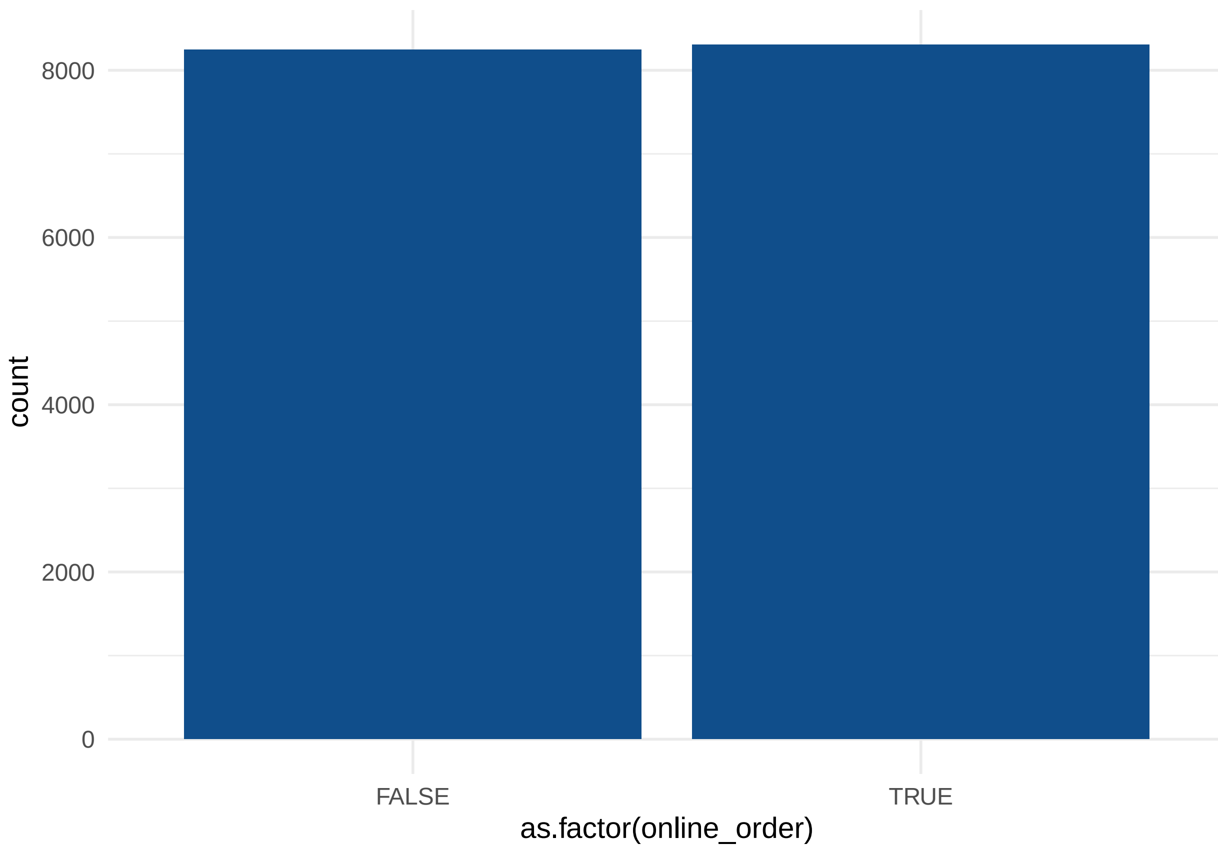
Data Exploration

```
join_data <- join_data %>% mutate(recency = (Sys.Date() - transaction_date))
join_data <- join_data %>% group_by(customer_id) %>% mutate(frequency = n())
join_data <- join_data %>% group_by(customer_id) %>% mutate(recency = min(recency))
join_data <- join_data %>% group_by(customer_id) %>% mutate(total_profit = sum(profits))
```

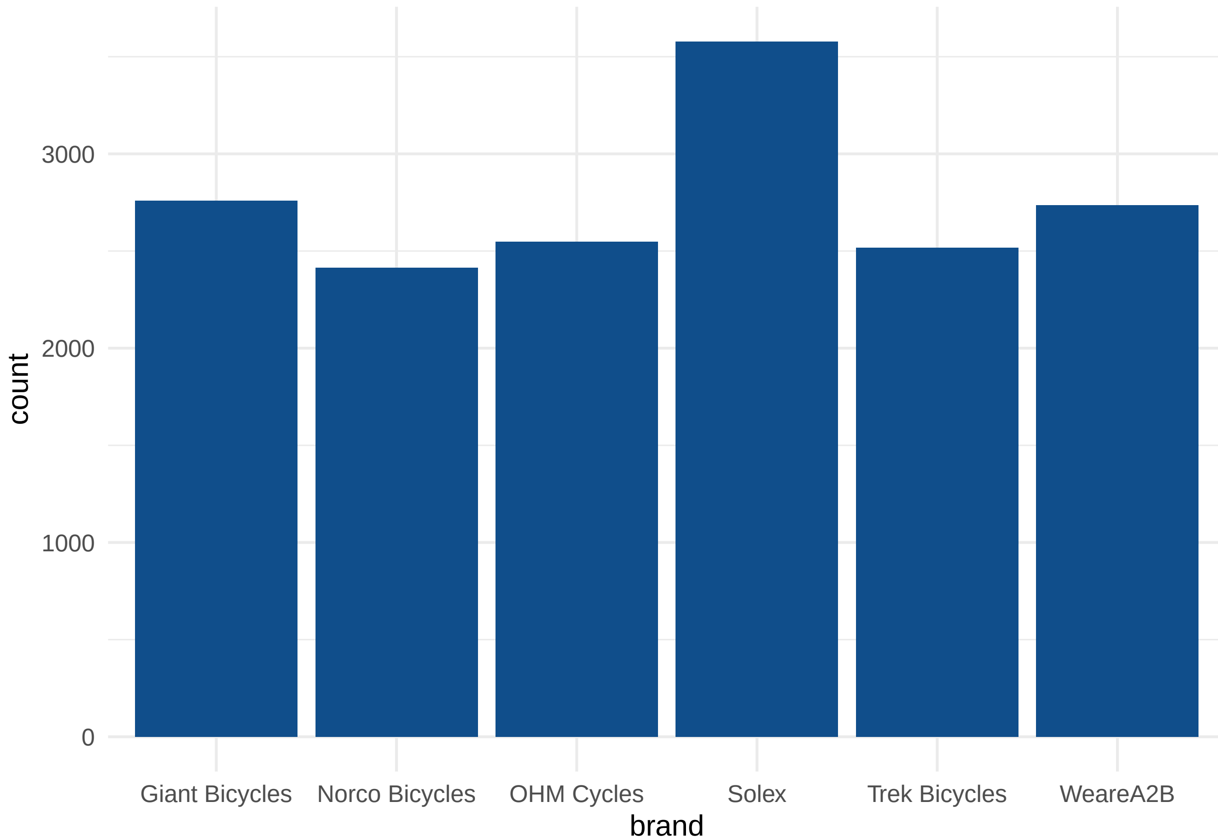
online_order brand gender past_3_years_bike_related_purchased job_title job_industry_category wealth_segment owns_car Age postcode state

Main effects

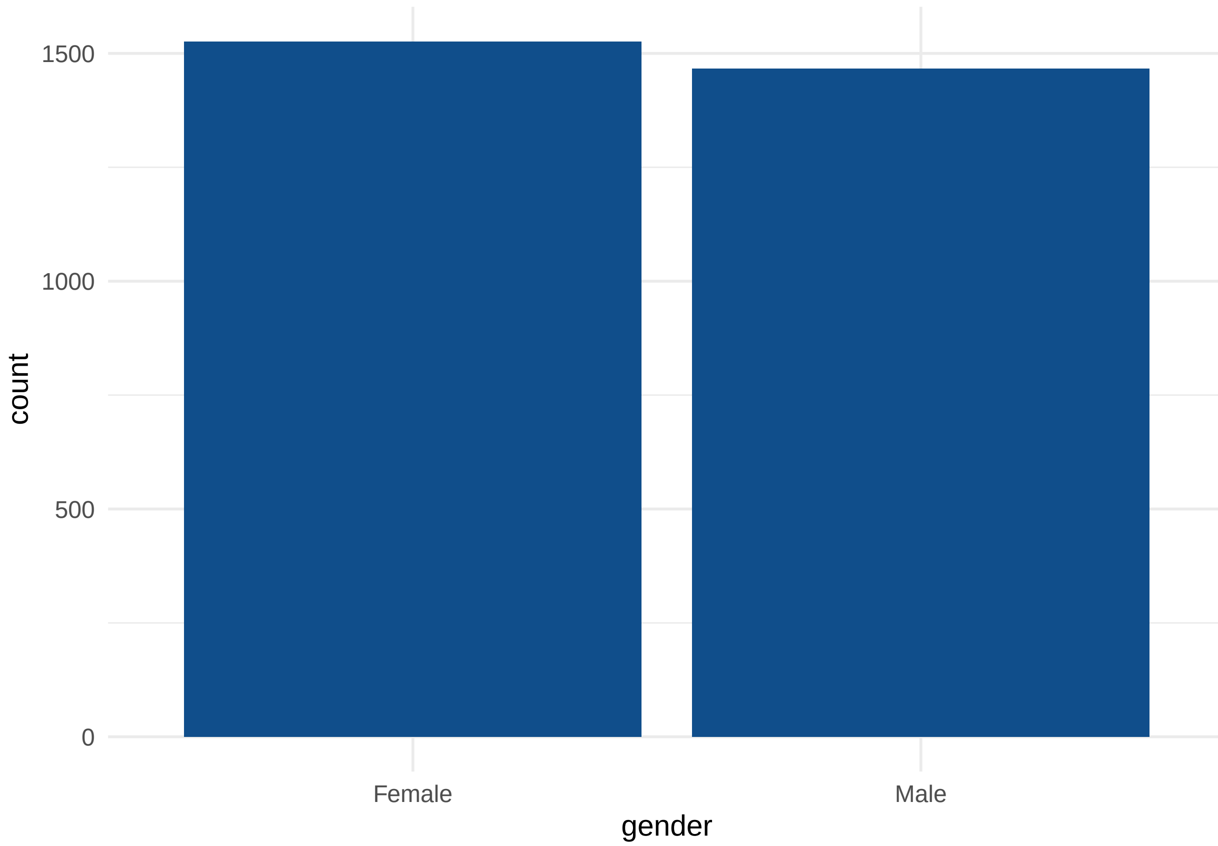
```
ggplot(data = join_data) +
  geom_bar(mapping = aes(x = as.factor(online_order)), fill = "dodgerblue4")+
  theme_minimal()
```



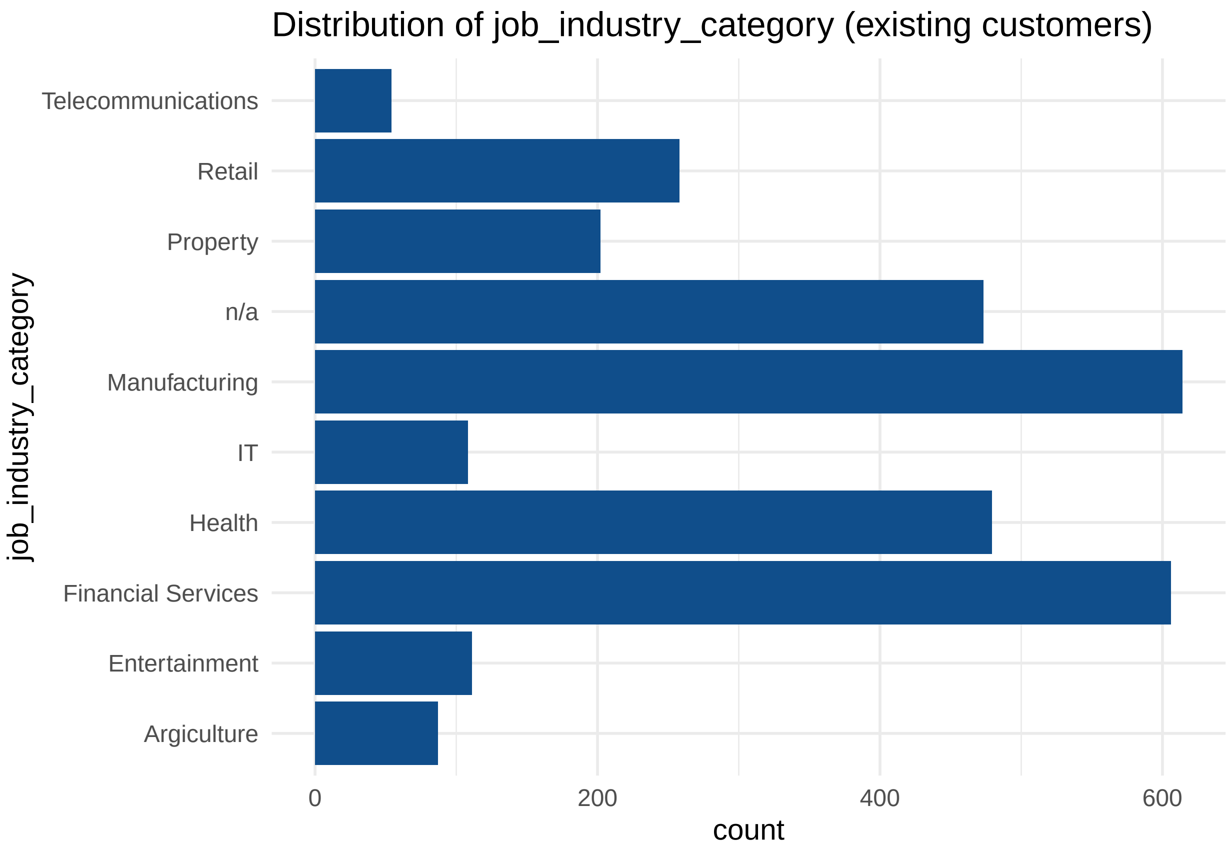

```
ggplot(data = join_data) +
  geom_bar(mapping = aes(x = brand), fill = "dodgerblue4")+
  theme_minimal()
```



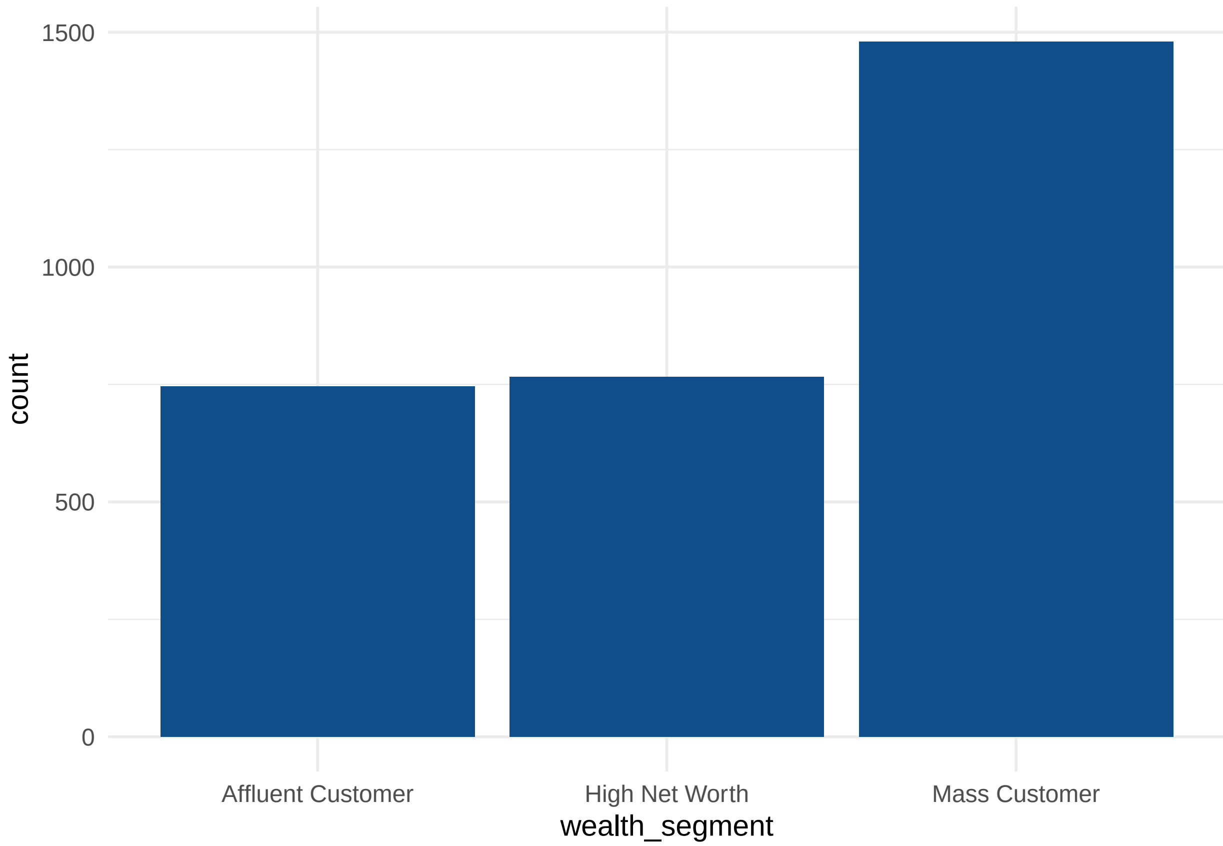

```
join_data %>% select(c(customer_id, gender)) %>% unique() %>% ggplot() +
  geom_bar(mapping = aes(x = gender), fill = "dodgerblue4")+
  theme_minimal()
```



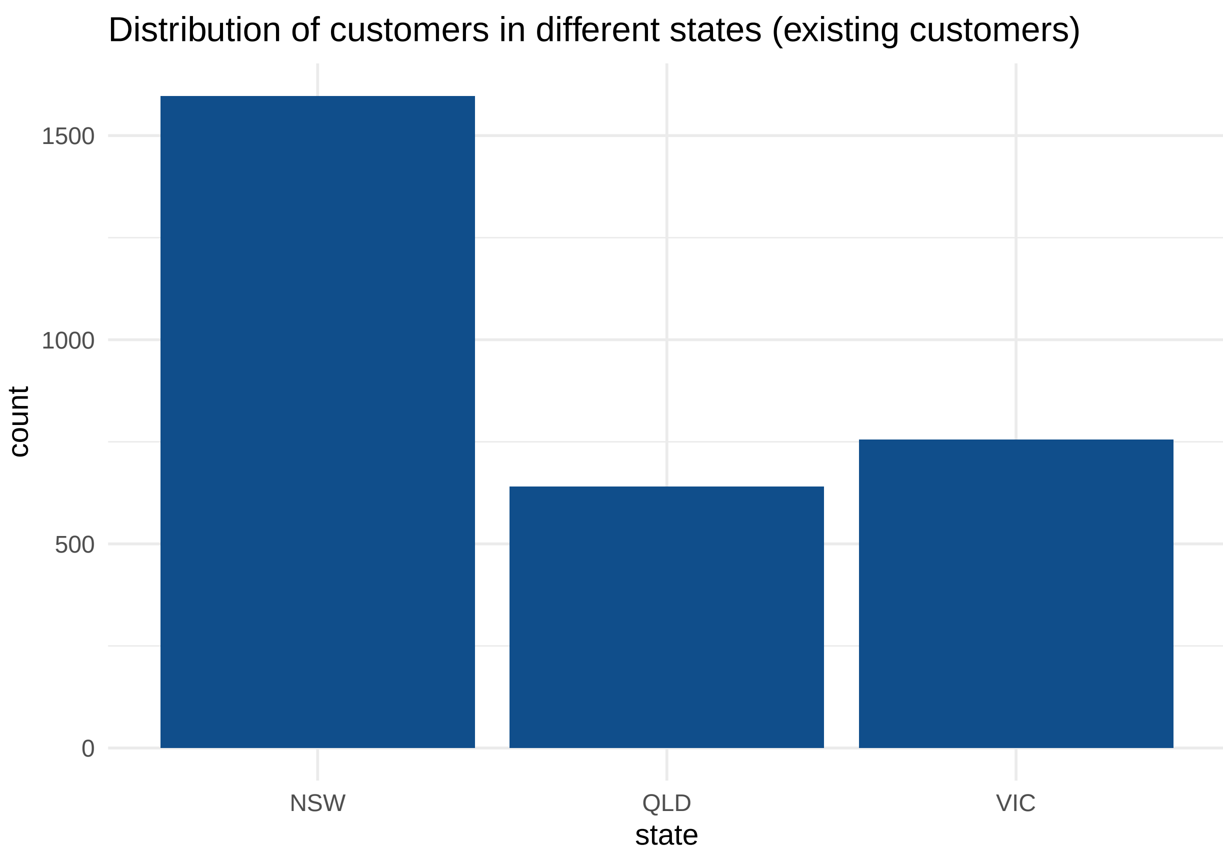

```r
bar <- join_data %>% select(c(customer_id, job_industry_category)) %>% unique() %>% ggplot() +
  geom_bar(mapping = aes(x = job_industry_category), fill = "dodgerblue4")+
  theme_minimal() +
  ggtitle("Distribution of job_industry_category (existing customers)")
bar + coord_flip()
```



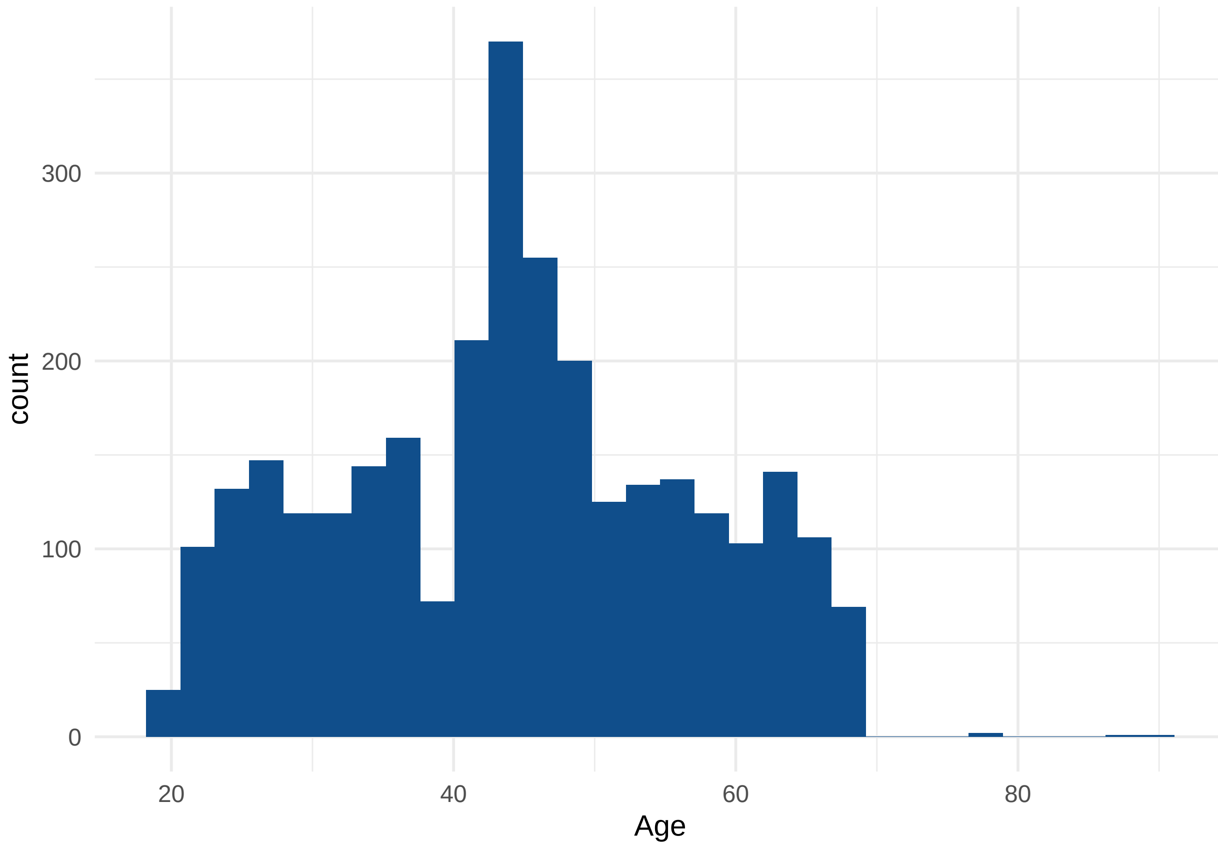

```
join_data %>% select(c(customer_id, wealth_segment)) %>% unique() %>% ggplot() +
  geom_bar(mapping = aes(x = wealth_segment), fill = "dodgerblue4")+
  theme_minimal()
```



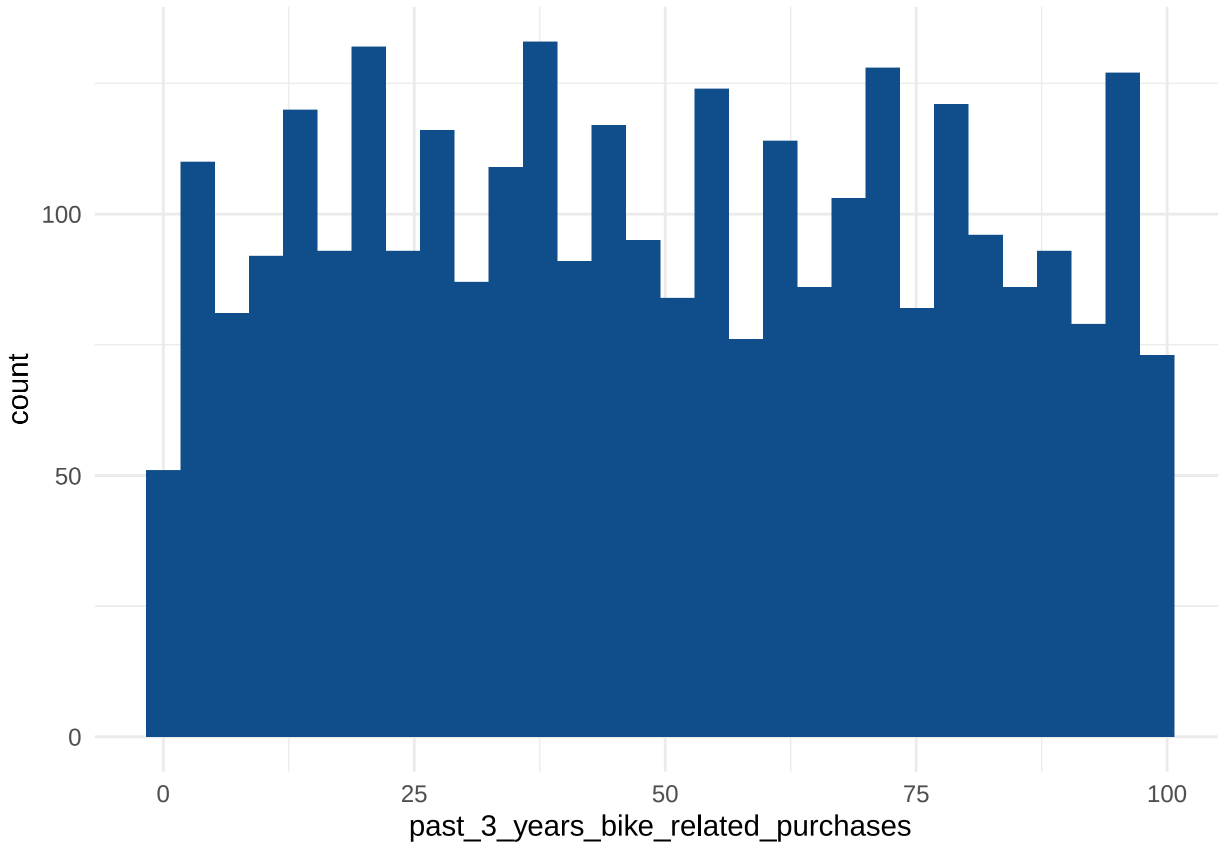

```
join_data %>% select(c(customer_id, state)) %>% unique() %>% ggplot() +
  geom_bar(mapping = aes(x = state), fill = "dodgerblue4")+
  ggtitle("Distribution of customers in different states (existing customers)") +
  theme_minimal()
```




```
join_data %>% select(c(customer_id, Age)) %>% unique() %>% ggplot() +
  geom_histogram(mapping = aes(x = Age), fill = "dodgerblue4")+
  theme_minimal()
```

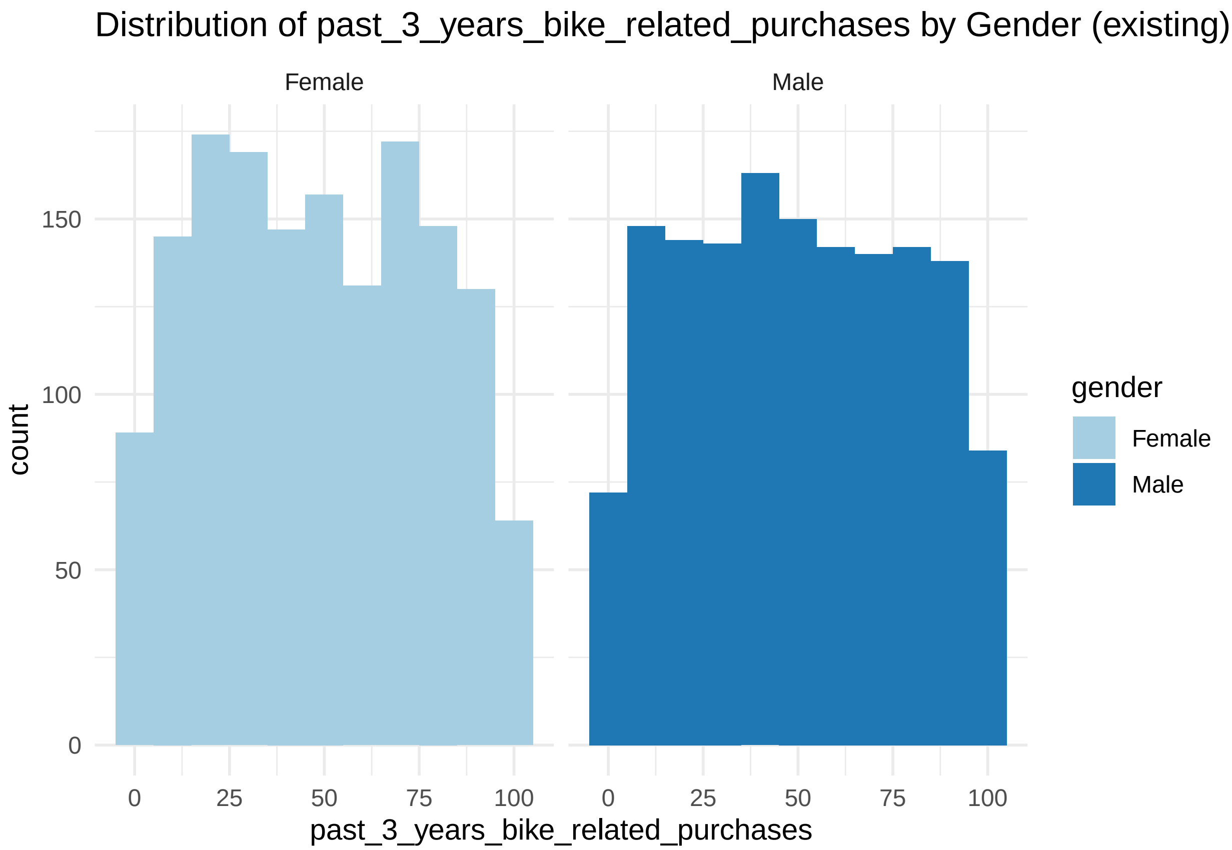



```
join_data %>% select(c(customer_id, past_3_years_bike_related_purchases)) %>% unique() %>% ggplot() +
  geom_histogram(mapping = aes(x = past_3_years_bike_related_purchases), fill = "dodgerblue4")+
  theme_minimal()
```



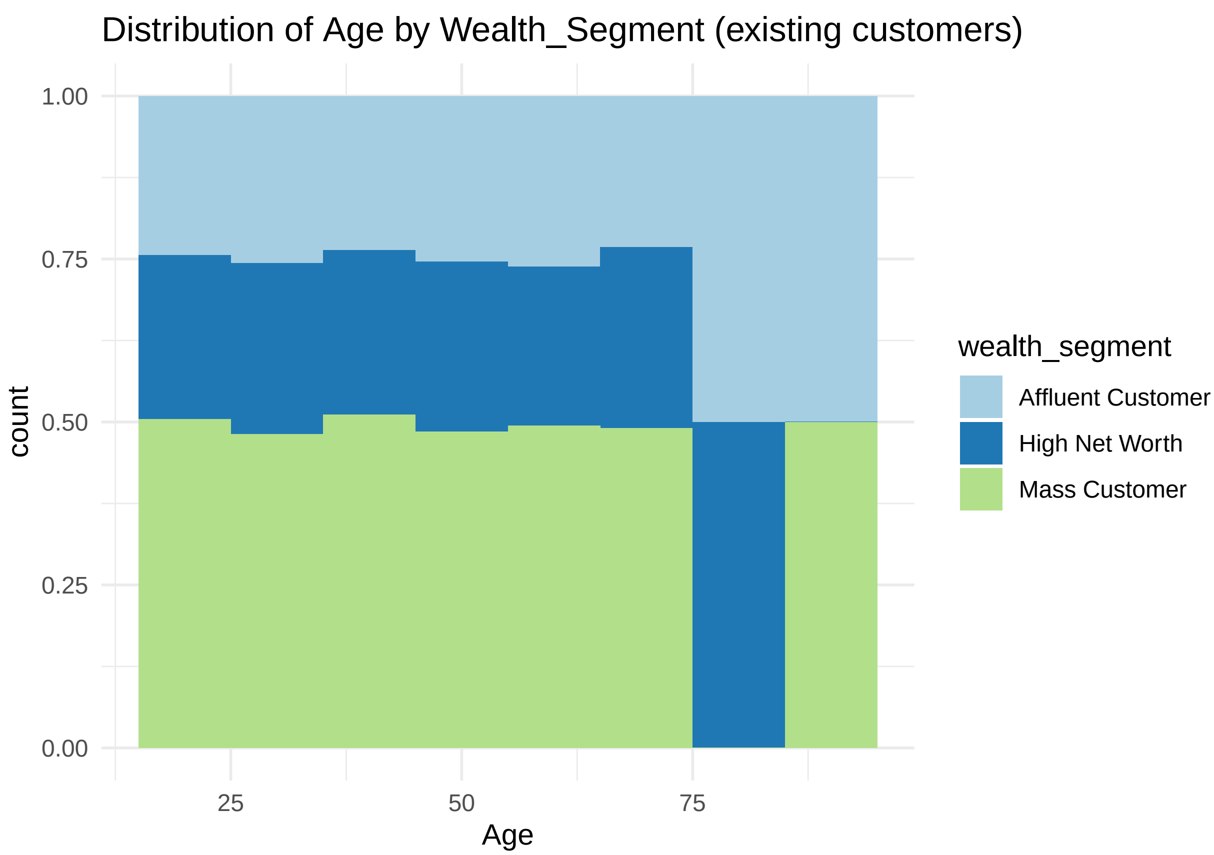
Relationship

```
join_data %>% select(c(customer_id, gender, past_3_years_bike_related_purchases)) %>% unique() %>%
  ggplot(aes(x = past_3_years_bike_related_purchases, fill = gender)) +
  geom_histogram(binwidth = 10) +
  facet_grid(~gender)+
  scale_fill_brewer(palette = "Paired") +
  ggtitle("Distribution of past_3_years_bike_related_purchases by Gender (existing)") +
  theme_minimal()
```




```r
join_data %>% select(c(customer_id, owns_car, state)) %>% unique() %>%
  ggplot(aes(x = state, fill = owns_car)) +
  geom_bar(position = "dodge")+
  scale_fill_brewer(palette = "Paired") +
  ggtitle("Distribution of customers in different states by car owning (existing customers)") +
  theme_minimal()
```


```r
join_data %>% select(c(customer_id, Age, wealth_segment)) %>% unique() %>% ggplot() +
  geom_histogram(mapping = aes(x = Age, fill = wealth_segment), position = "fill", binwidth = 10)+
  scale_fill_brewer(palette = "Paired") +
  ggtitle("Distribution of Age by Wealth_Segment (existing customers)") +
  theme_minimal()
```



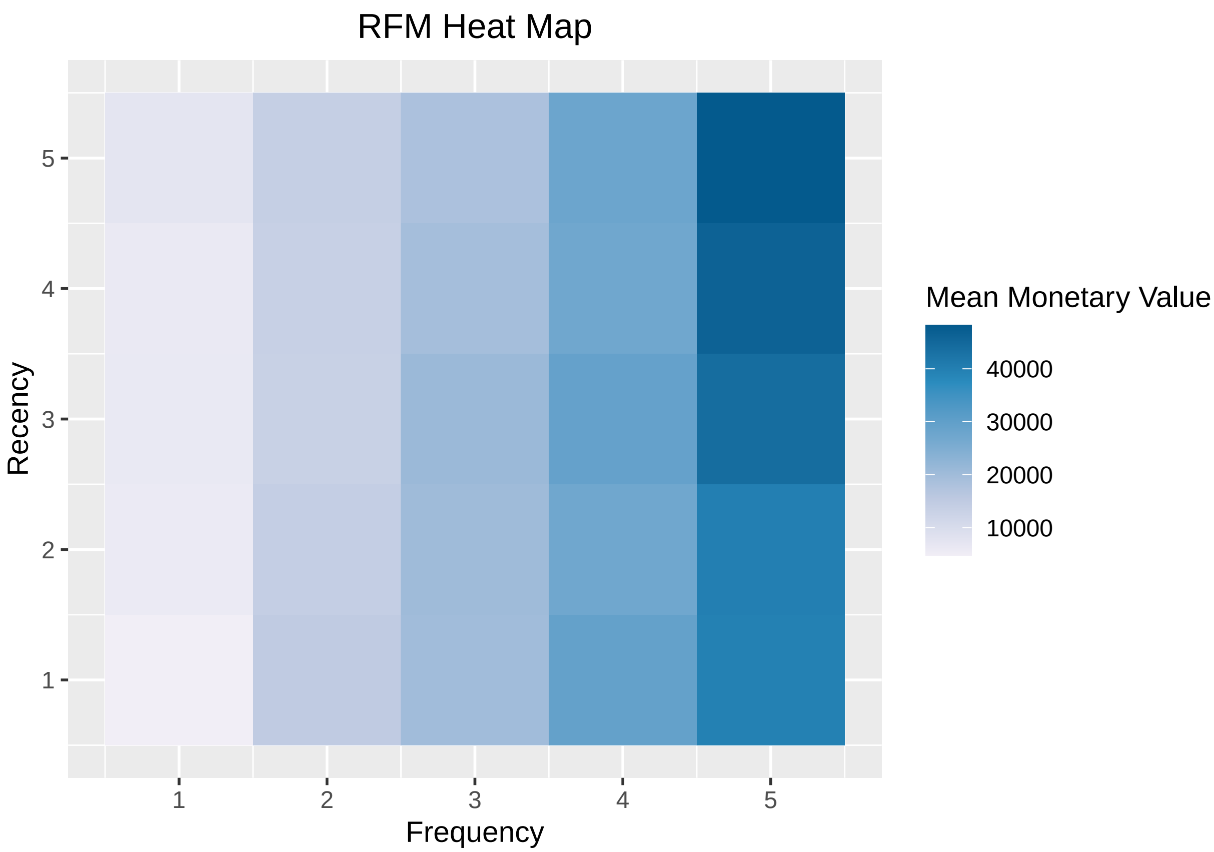
test

```
rfm_data <- data.frame(cbind(as.character(join_data$customer_id), join_data$transaction_date, join_data$

rfm_data <- rfm_data %>% rename(customer_id = "X1")

rfm_data <- rfm_data %>% rename(transactions_date = "X2")

rfm_data <- rfm_data %>% rename(total_profit = "X3")

rfm_data$transactions_date <- as.Date(as.numeric(rfm_data$transactions_date), origin = "1970-01-01")

rfm_data$total_profit <- as.numeric(rfm_data$total_profit)

rfm_data <- distinct(rfm_data)

rfm_result <- rfm_table_order(rfm_data, customer_id, transactions_date, total_profit, Sys.Date())

rfm_result
```

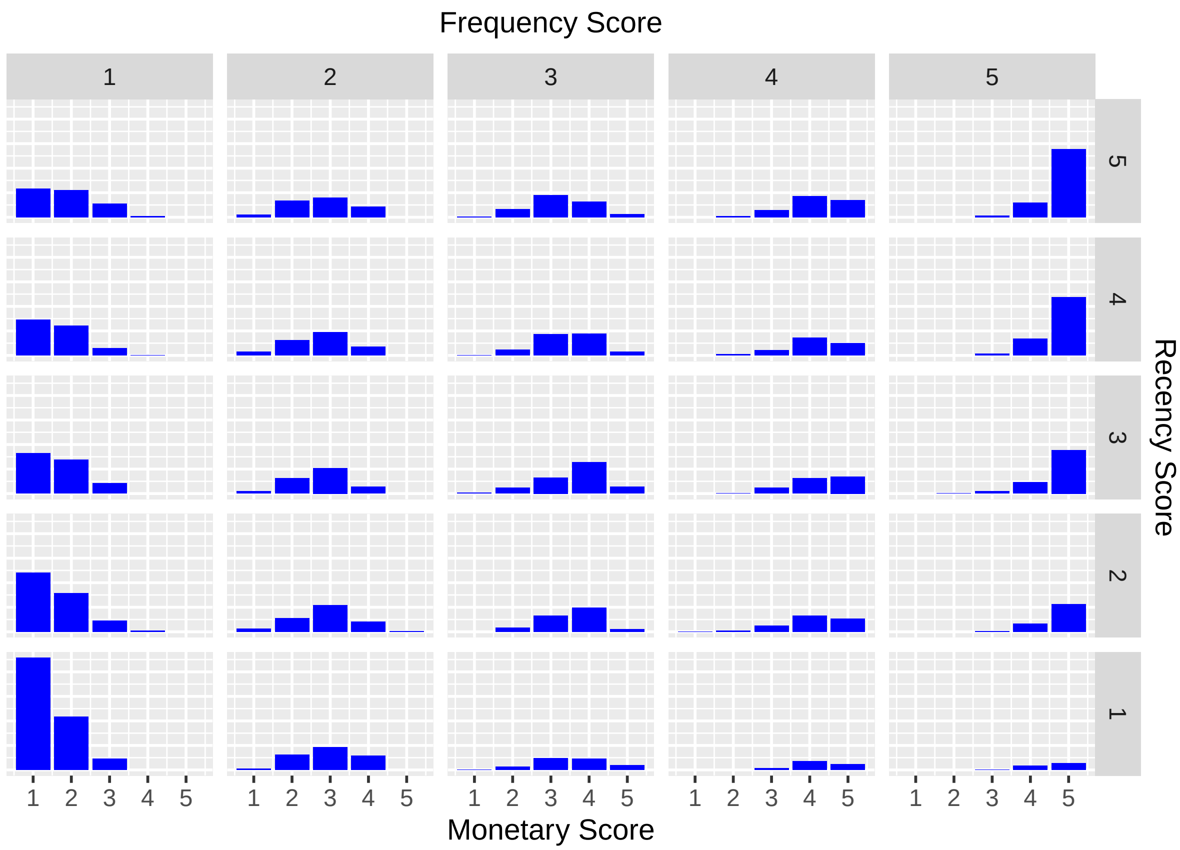
```
## # A tibble: 2,992 x 9
##    customer_id date_most_recent recency_days transaction_count amount
##    <chr>       <date>                  <dbl>             <dbl>  <dbl>
##  1 1           2017-12-23               1426                11 33199.
##  2 100         2017-12-19               1430                 2  1755.
##  3 1000        2017-12-30               1419                 9 48451.
```

```
##   4 1001        2017-11-18                 1461                 7 20189.
##   5 1002        2017-07-28                 1574                 3  6764.
##   6 1003        2017-11-13                 1466                 9 47703.
##   7 1004        2017-06-10                 1622                 6 21606.
##   8 1005        2017-07-24                 1578                 5 21826.
##   9 1006        2017-11-23                 1456                 8 37500.
## 10 1008        2017-12-13                 1436                 4 10094.
## # ... with 2,982 more rows, and 4 more variables: recency_score <int>,
## #   frequency_score <int>, monetary_score <int>, rfm_score <dbl>
```

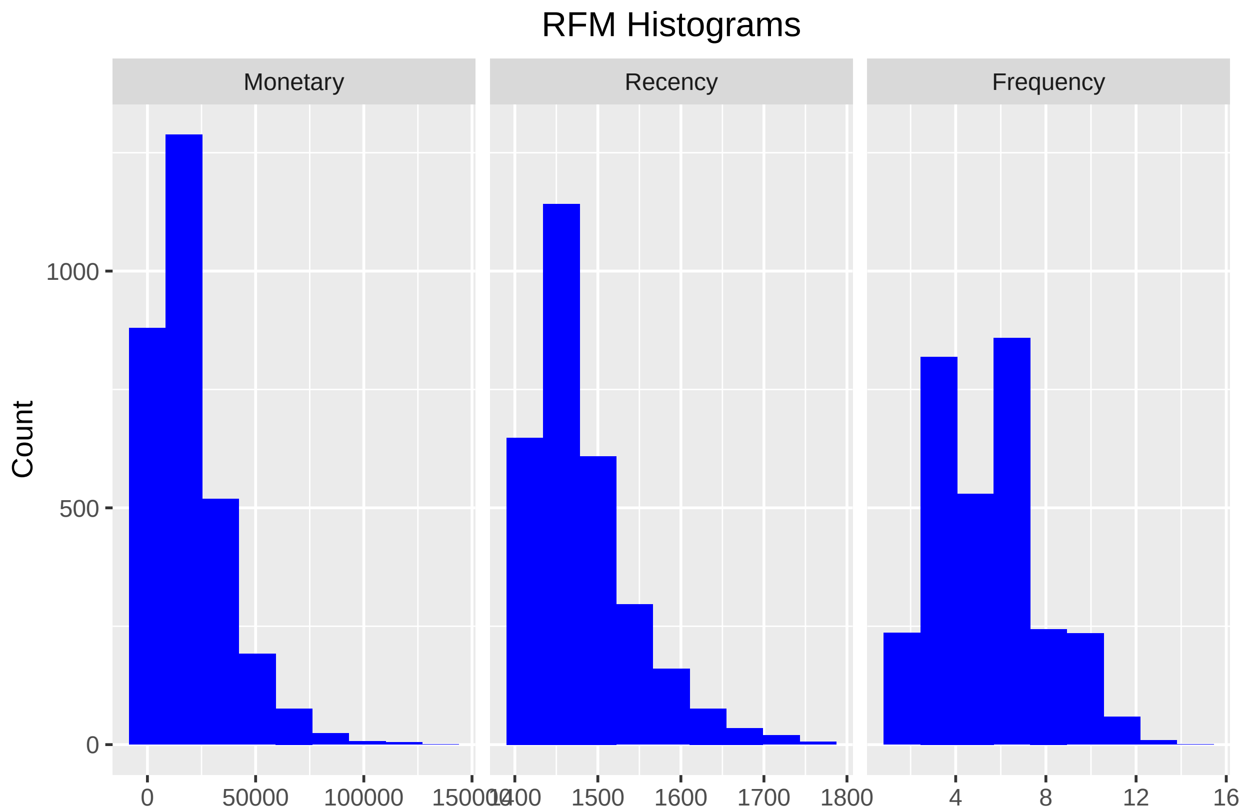
```
rfm_heatmap(rfm_result)
```




```
rfm_bar_chart(rfm_result)
```

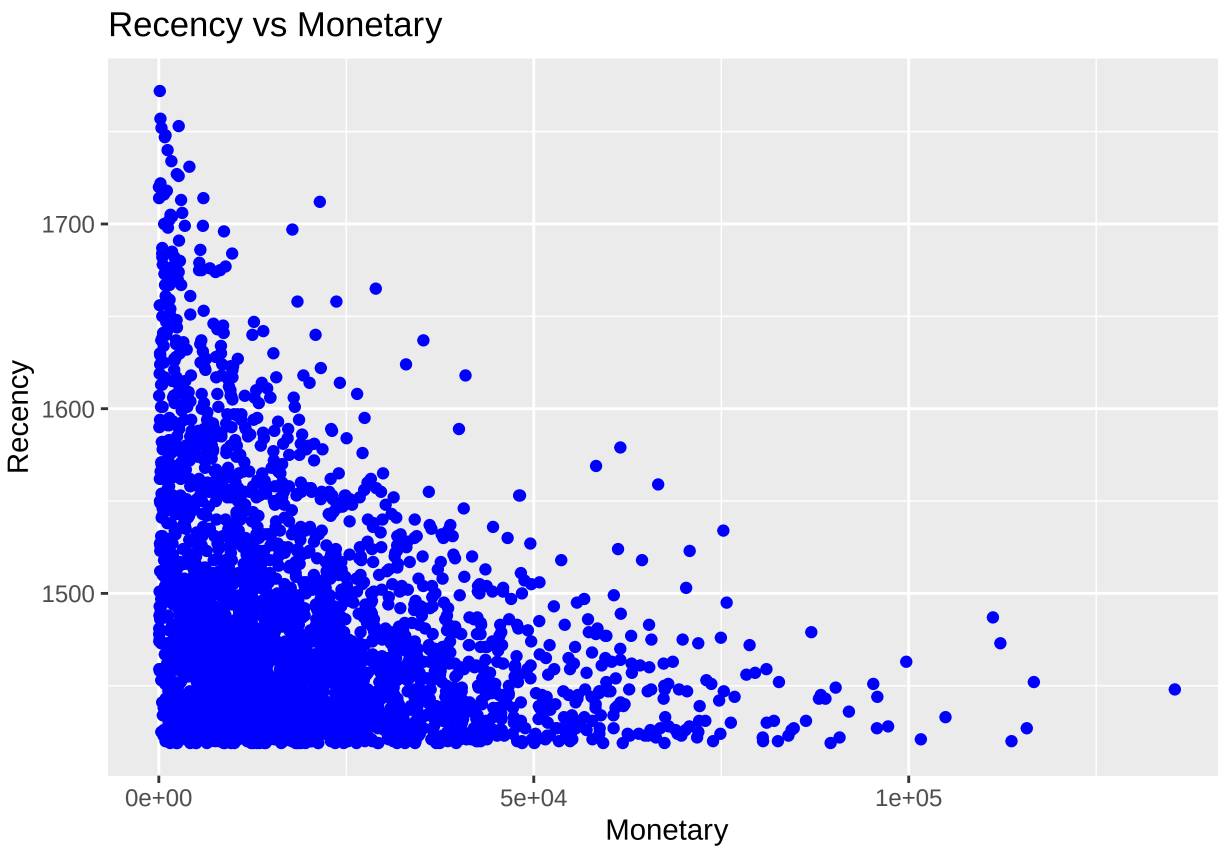



```r
rfm_histograms(rfm_result)
```

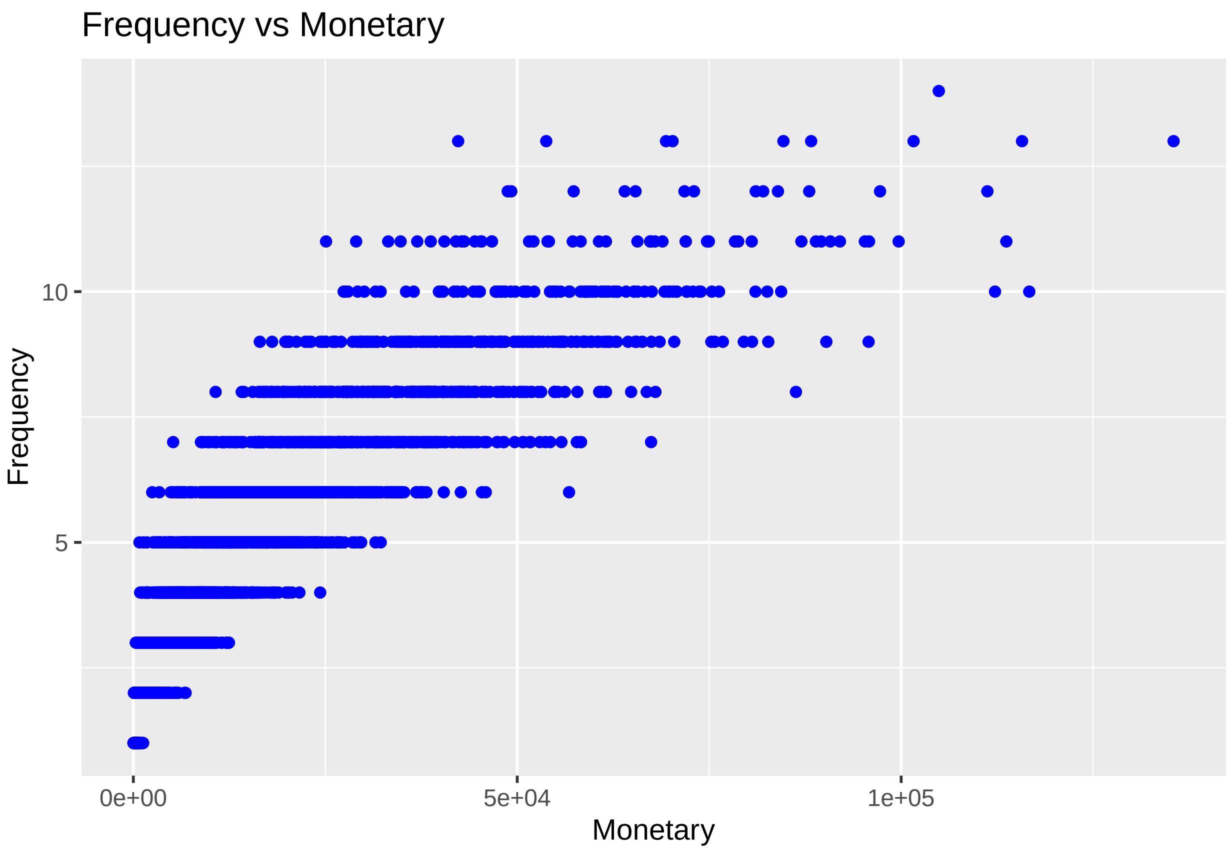



```
rfm_order_dist(rfm_result)
```

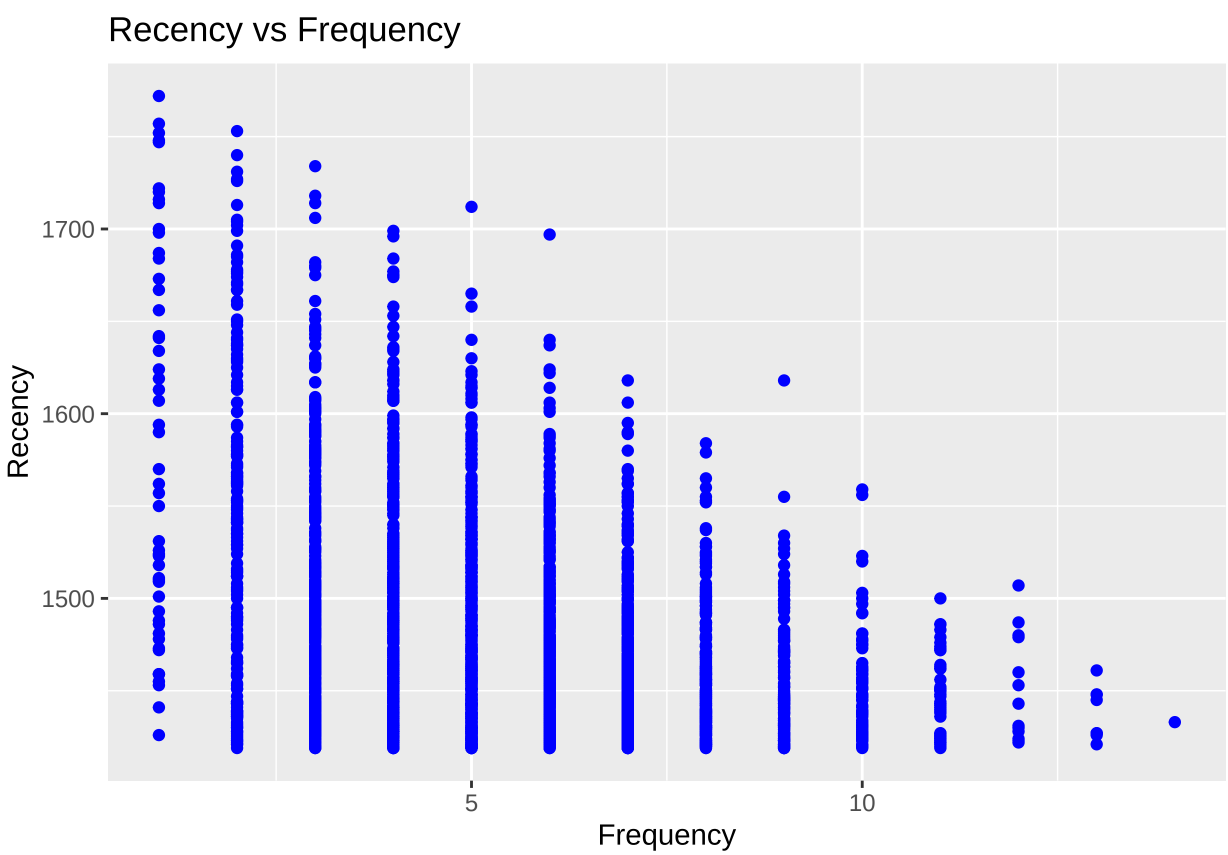

```
rfm_rm_plot(rfm_result)
```




```
rfm_fm_plot(rfm_result)
```




```r
rfm_rf_plot(rfm_result)
```



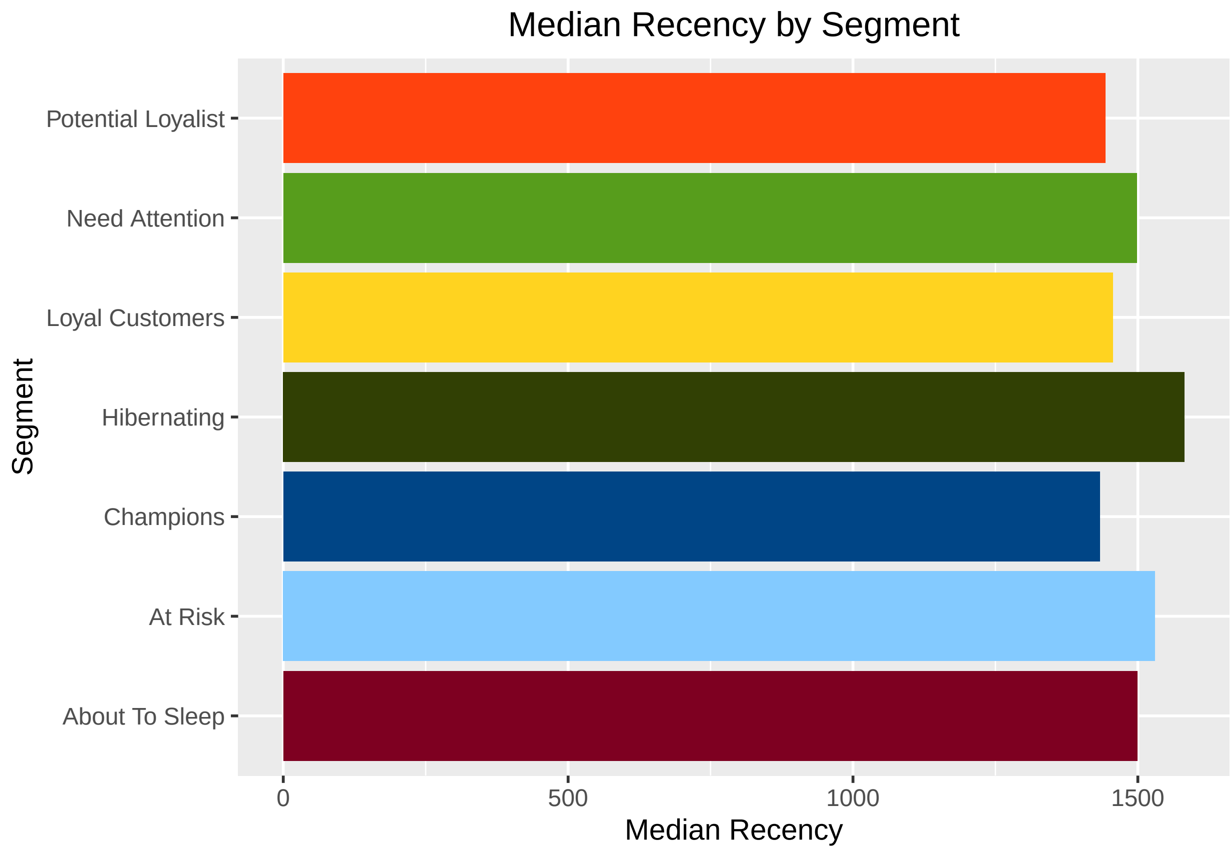

```
rfm_data <- rfm_data %>% select(-c('transactions_date'))
recency_score <- rfm_result %>% pull(rfm) %>% pull(recency_score)
frequency_score <- rfm_result %>% pull(rfm) %>% pull(frequency_score)
monetary_score <- rfm_result %>% pull(rfm) %>% pull(monetary_score)
rfm_score <- rfm_result %>% pull(rfm) %>% pull(rfm_score)
customer_id <- rfm_result %>% pull(rfm) %>% pull(customer_id)
rfm_info <- cbind.data.frame(customer_id, rfm_score, recency_score, frequency_score, monetary_score)
```

```
rfm_info <- rfm_info %>%
  mutate(segment = case_when(recency_score >= 4 & frequency_score >= 4 & monetary_score >= 4 ~ 'Champio
         recency_score >= 3 & frequency_score >= 3 & monetary_score >= 2 ~ 'Loyal Customers',
         recency_score >= 3 & 3 >= frequency_score & frequency_score >= 1 &
           4 >= monetary_score  & monetary_score >= 1 ~ 'Potential Loyalist',
         recency_score >= 4 & frequency_score <= 1 & monetary_score <= 1 ~ 'New Customers',
         4 >= recency_score & recency_score >= 3 & frequency_score <= 1 &
           monetary_score <= 1 ~ 'Promising',
         3 >= recency_score & recency_score >= 2 & 3 >= frequency_score &  frequency_score >= 1 &
           4 >= monetary_score & monetary_score >= 2 ~ 'Need Attention',
         3 >= recency_score & recency_score >= 2 & frequency_score <= 2 &
           monetary_score <= 2 ~ 'About To Sleep',
         recency_score <= 2 & 5 >= frequency_score & frequency_score >= 2 &
           5 >= monetary_score  & monetary_score >= 1 ~ 'At Risk',
         recency_score <= 1 & 5 >= frequency_score & frequency_score >= 4 &
           5 >= monetary_score & monetary_score >= 4 ~ 'Can't Lose Them',
         2 >= recency_score & recency_score >= 1 &
           3 >= frequency_score & frequency_score >= 1 &
           3 >= monetary_score & monetary_score >= 1 ~ 'Hibernating',
```

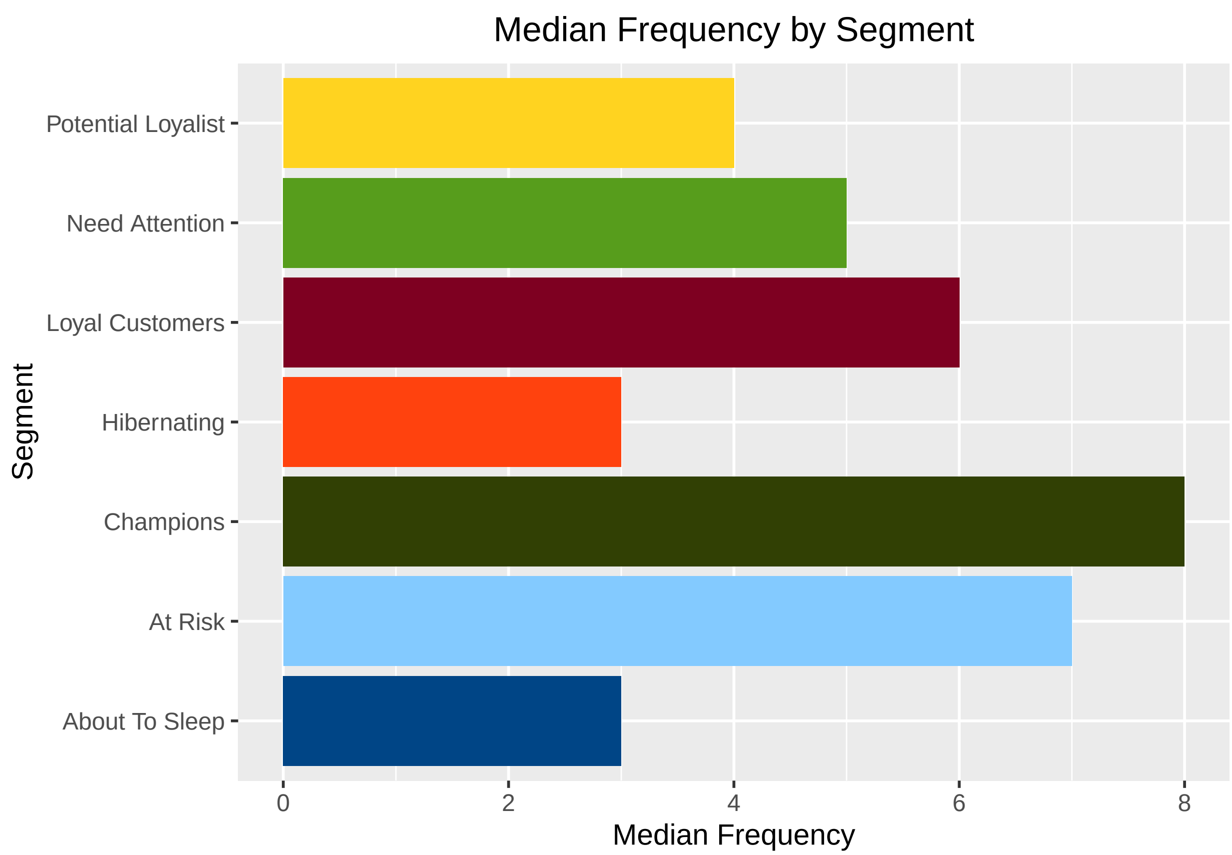
```
        recency_score <= 2 & frequency_score <= 2 & monetary_score <= 2 ~ 'Lost'))
```

```
rfm_data <- rfm_data %>% full_join(rfm_info)
customer_info <- cbind.data.frame(join_data$customer_id, join_data$gender,
                                  join_data$past_3_years_bike_related_purchases,
                                                join_data$job_industry_category,
                                  join_data$wealth_segment, join_data$owns_car, join_data$tenure,
                                                join_data$postcode, join_data$state,
                                  join_data$property_valuation, join_data$Age,
                                  join_data$recency, join_data$frequency, join_data$total_profit)

customer_info <- customer_info %>% rename(customer_id = 'join_data$customer_id')
customer_info <- customer_info %>% rename(gender = 'join_data$gender')
customer_info <- customer_info %>% rename(past_3_years_bike_related_purchases =
                                            'join_data$past_3_years_bike_related_purchases')
customer_info <- customer_info %>% rename(job_industry_category = 'join_data$job_industry_category')
customer_info <- customer_info %>% rename(wealth_segment = 'join_data$wealth_segment')
customer_info <- customer_info %>% rename(owns_car = 'join_data$owns_car')
customer_info <- customer_info %>% rename(tenurer = 'join_data$tenure')
customer_info <- customer_info %>% rename(postcode = 'join_data$postcode')
customer_info <- customer_info %>% rename(state = 'join_data$state')
customer_info <- customer_info %>% rename(property_valuation = 'join_data$property_valuation')
customer_info <- customer_info %>% rename(age = 'join_data$Age')
customer_info <- customer_info %>% rename(transaction_count = 'join_data$frequency')
customer_info <- customer_info %>% rename(recency_days = 'join_data$recency')
customer_info <- customer_info %>% rename(amount = 'join_data$total_profit')

rfm_data <- rfm_data %>% inner_join(customer_info)
rfm_data <- distinct(rfm_data)
```

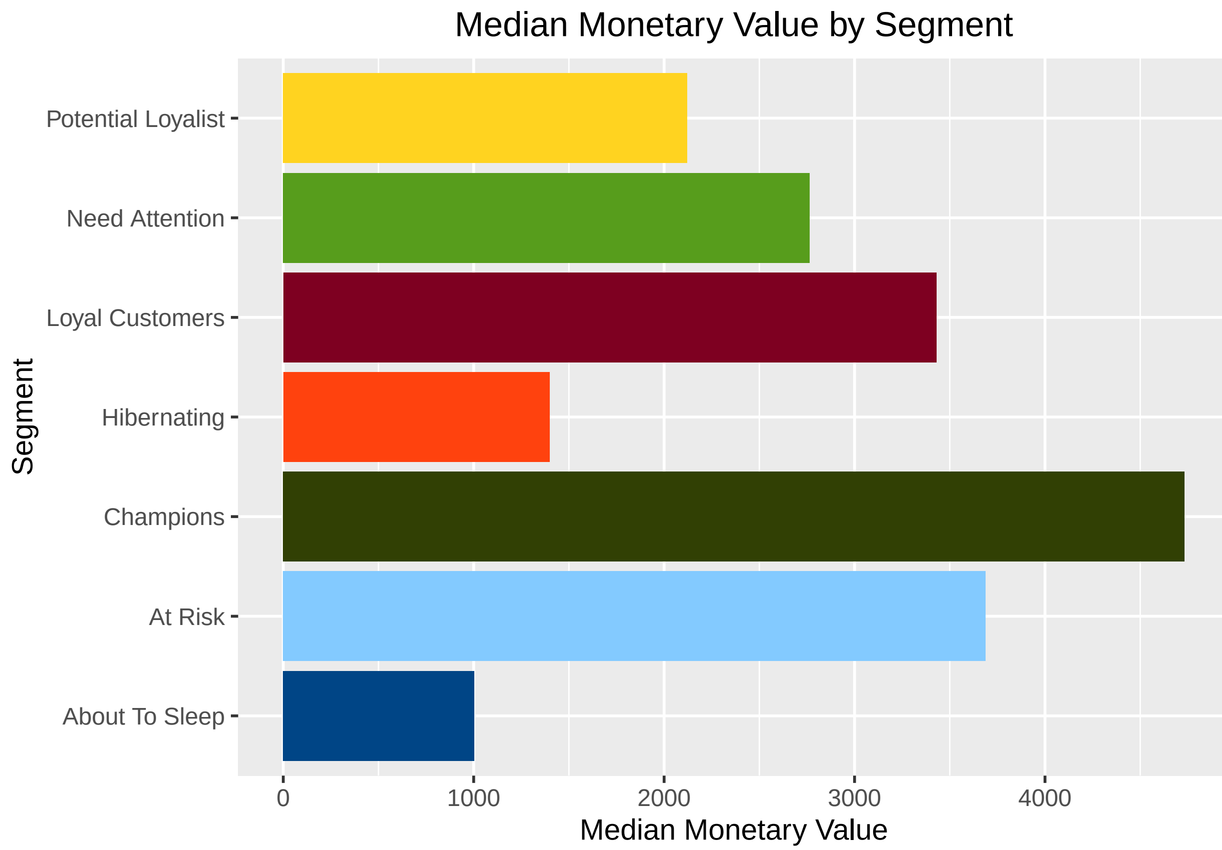
```
rfm_plot_median_recency(rfm_data)
```



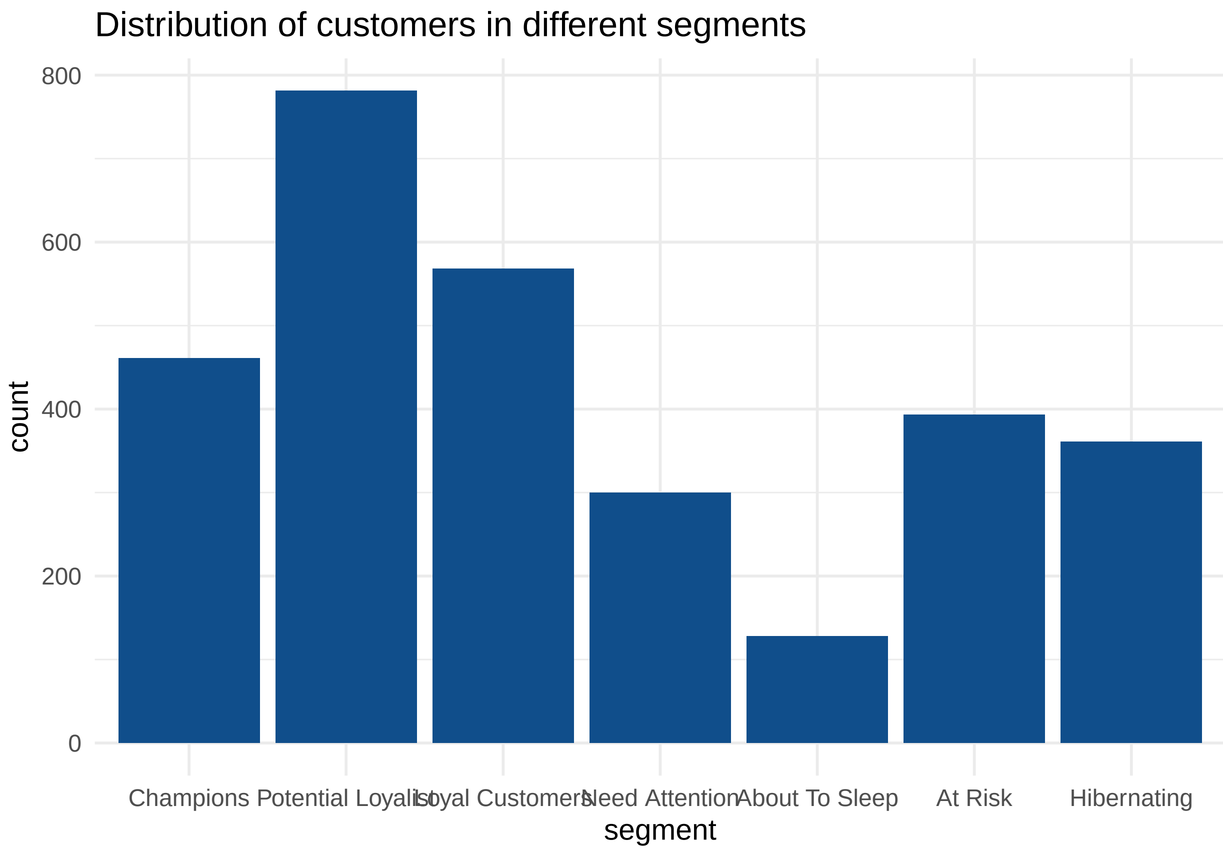

```r
rfm_plot_median_frequency(rfm_data)
```




```
rfm_plot_median_monetary(rfm_data)
```



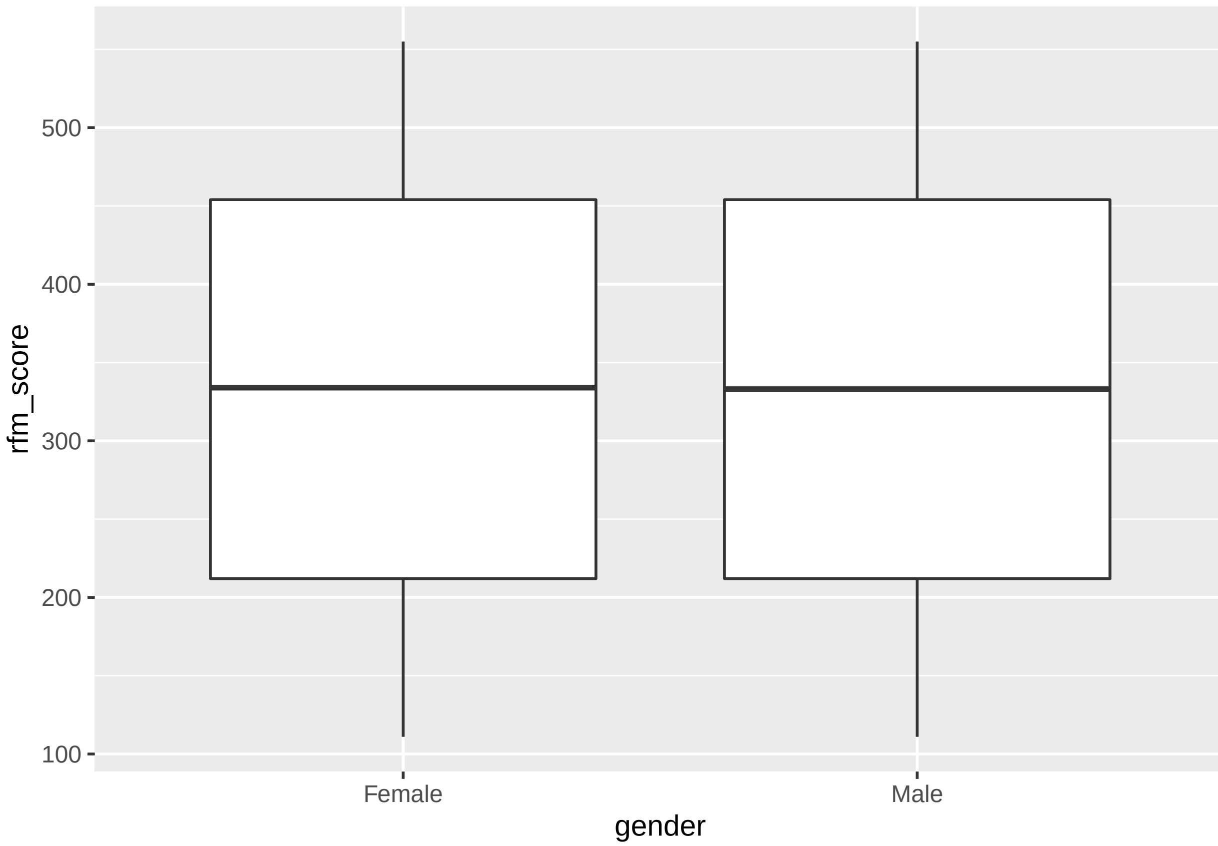

```r
rfm_data %>% mutate(segment = reorder(segment, desc(rfm_score))) %>% ggplot() +
  geom_bar(mapping = aes(x = segment), fill = "dodgerblue4")+
  ggtitle("Distribution of customers in different segments") +
  theme_minimal()
```



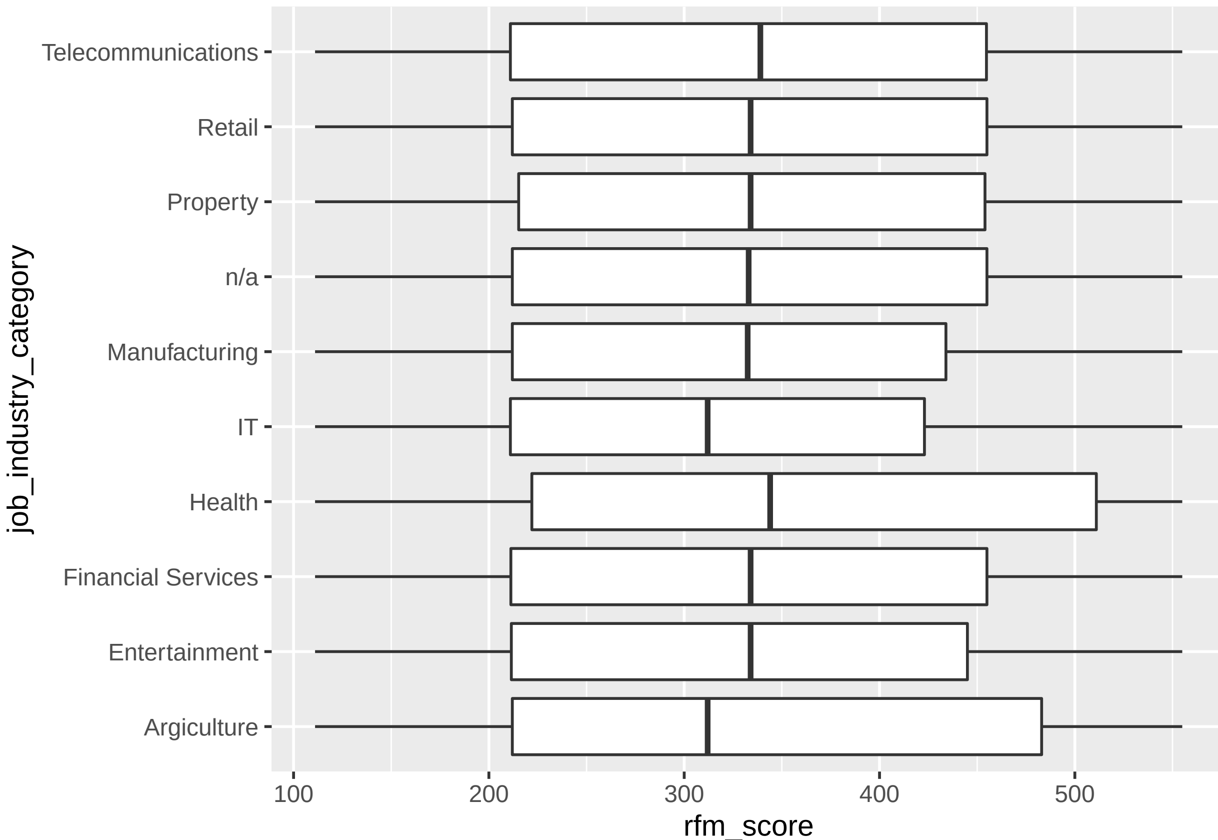

```
reorder <- rfm_data %>% select(segment, rfm_score)  %>% group_by(segment) %>% count()
reorder
```

```
## # A tibble: 7 x 2
## # Groups:   segment [7]
##   segment                n
##   <chr>              <int>
## 1 About To Sleep       128
## 2 At Risk              393
## 3 Champions            461
## 4 Hibernating          361
## 5 Loyal Customers      568
## 6 Need Attention       300
## 7 Potential Loyalist   781
```

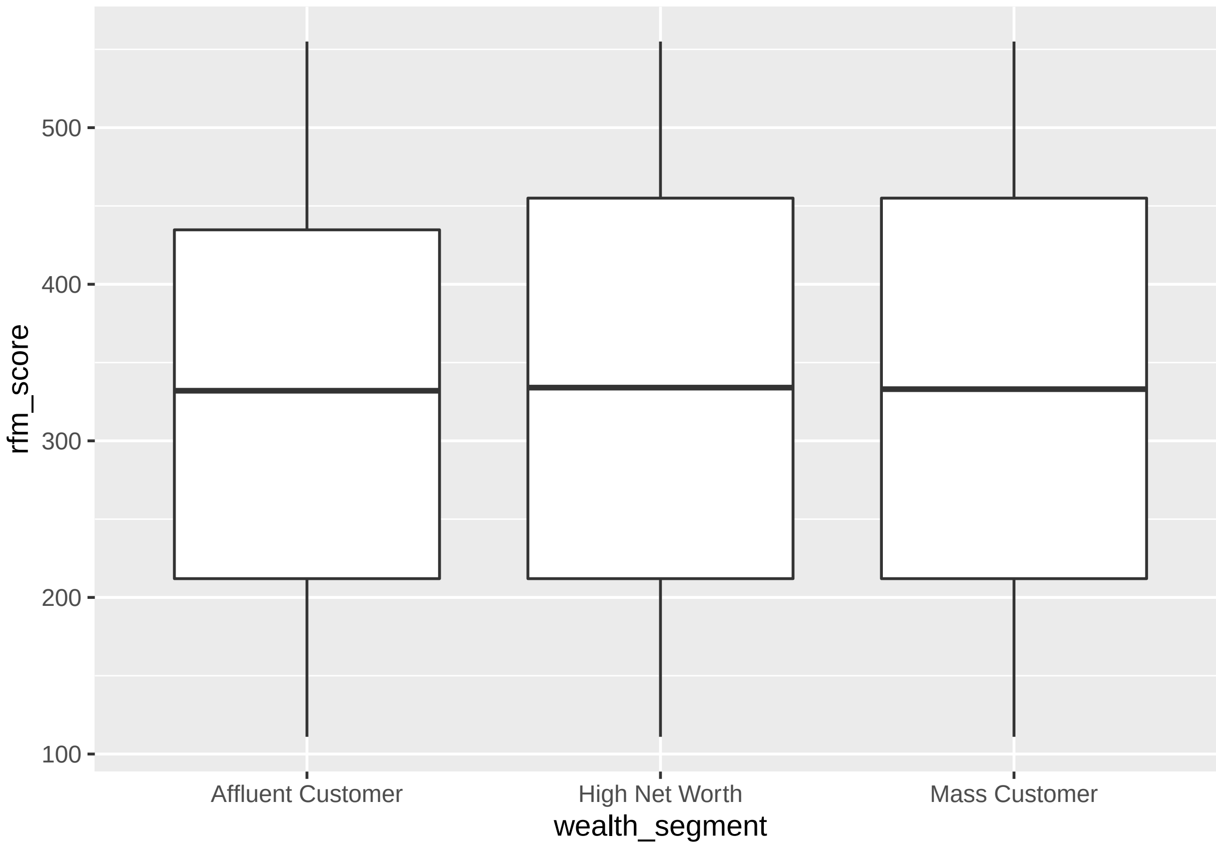
```
rfm_data %>% ggplot(aes(x = gender, y = rfm_score)) + geom_boxplot()
```



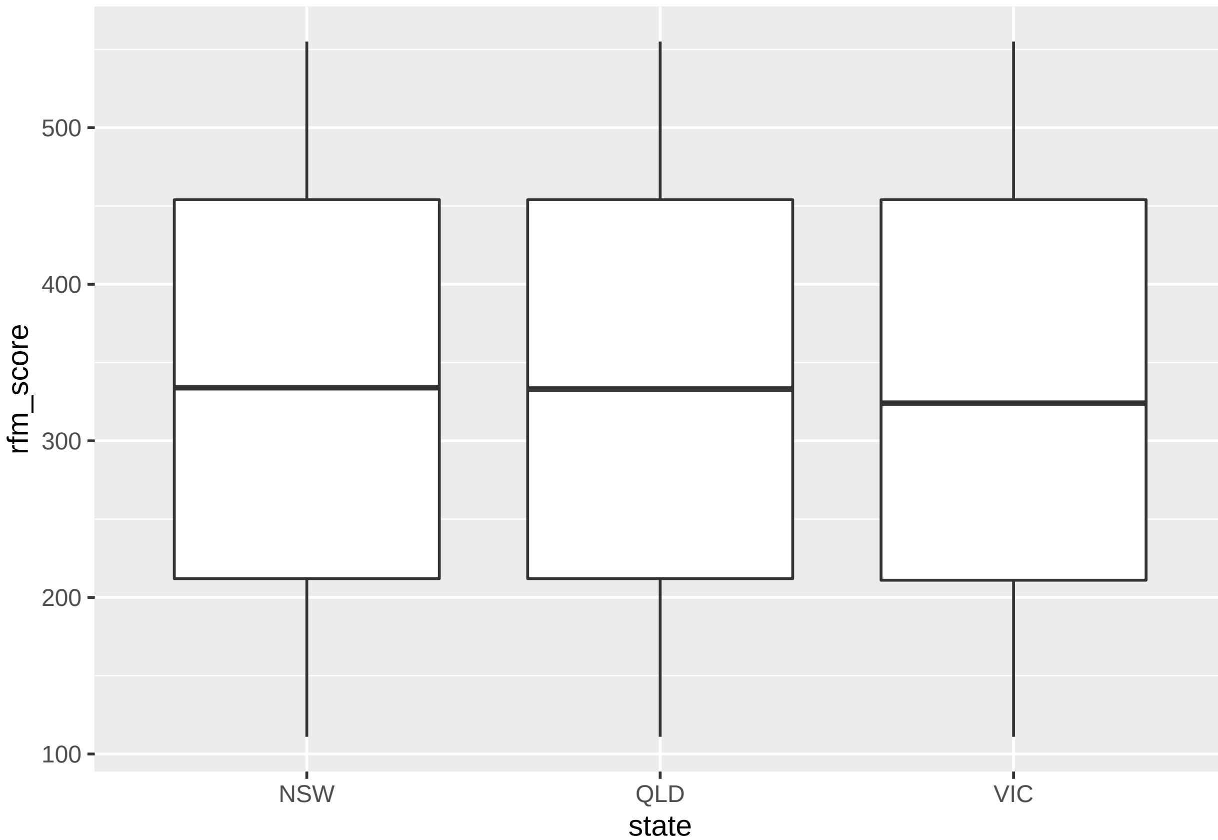

```
rfm_data %>% ggplot(aes(x = job_industry_category, y = rfm_score)) + geom_boxplot() + coord_flip()
```



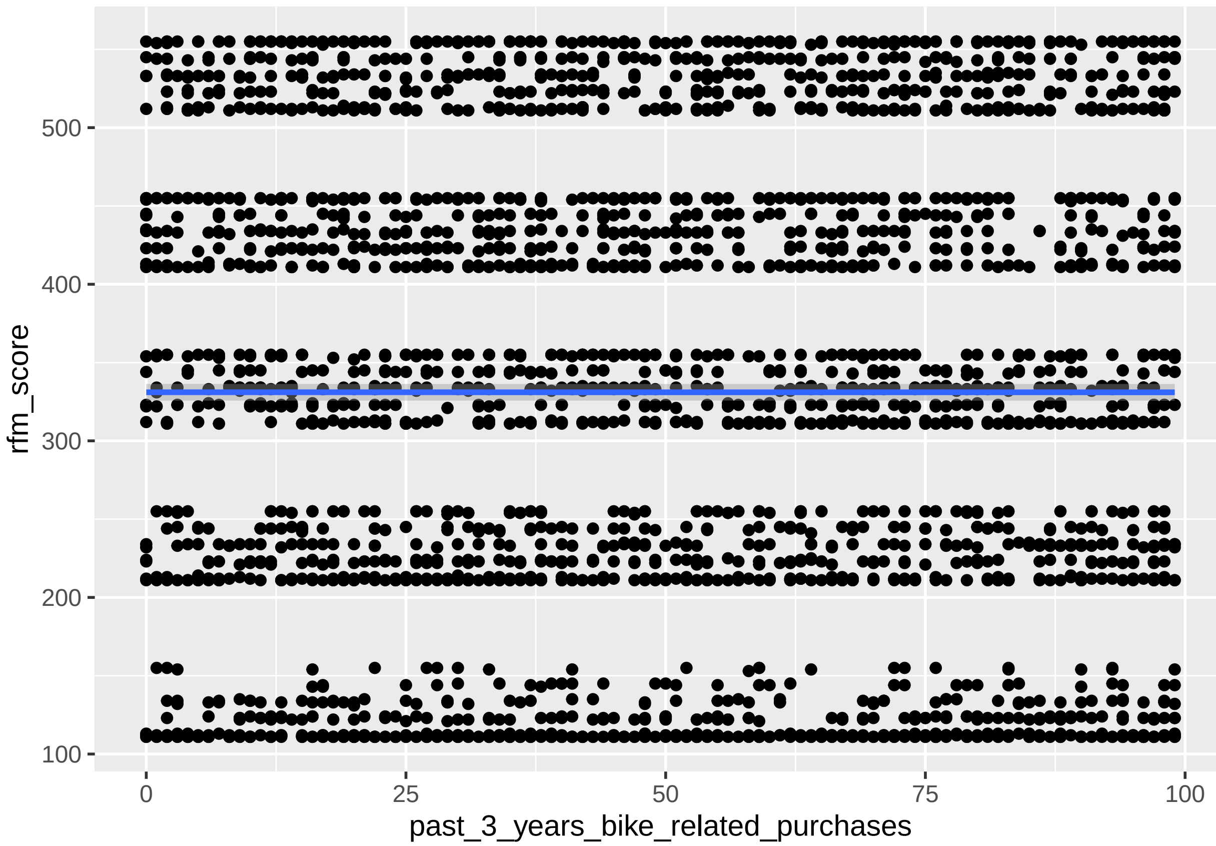

```
rfm_data %>% ggplot(aes(x = wealth_segment, y = rfm_score)) + geom_boxplot()
```



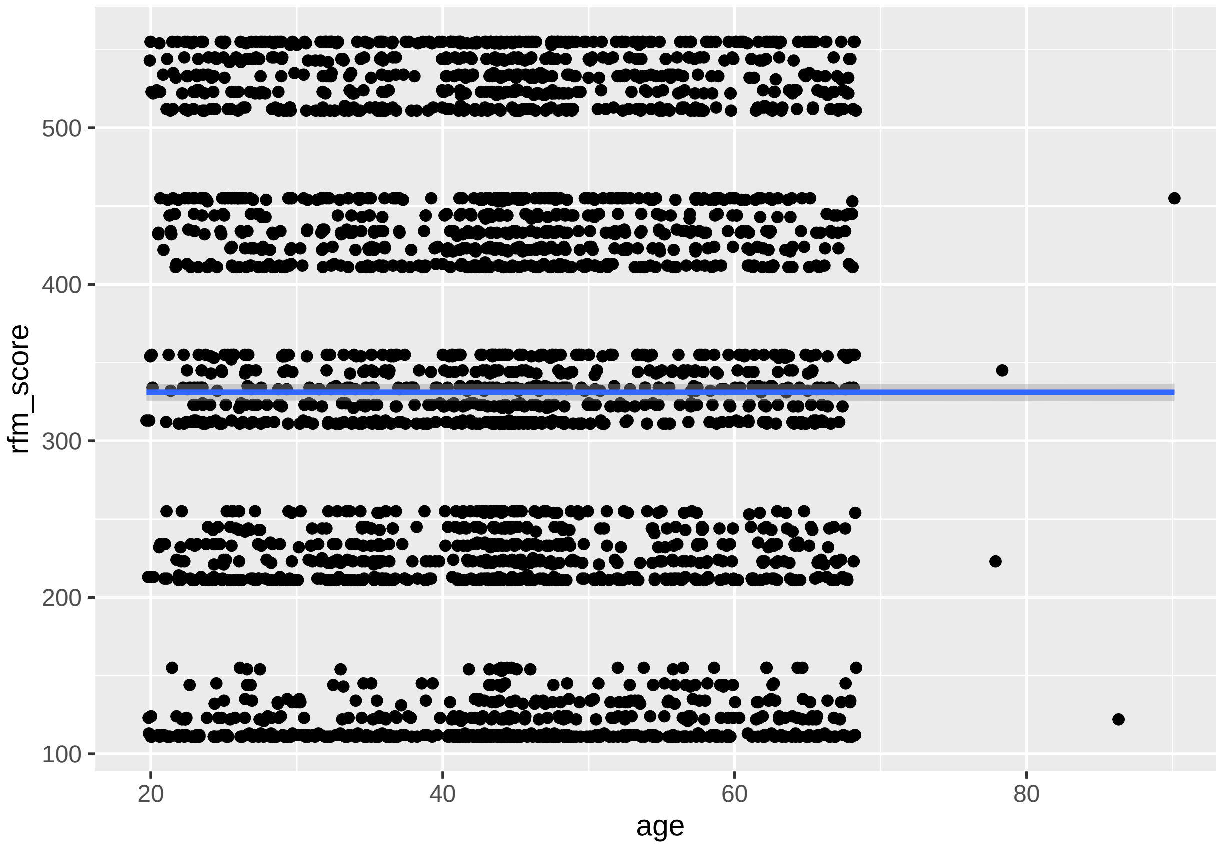

```
rfm_data %>% ggplot(aes(x = state, y = rfm_score)) + geom_boxplot()
```




```r
rfm_data %>% ggplot(aes(x = past_3_years_bike_related_purchases, y = rfm_score)) +
  geom_point() + geom_smooth()
```




```r
rfm_data %>% ggplot(aes(x = age, y = rfm_score)) + geom_point() + geom_smooth()
```



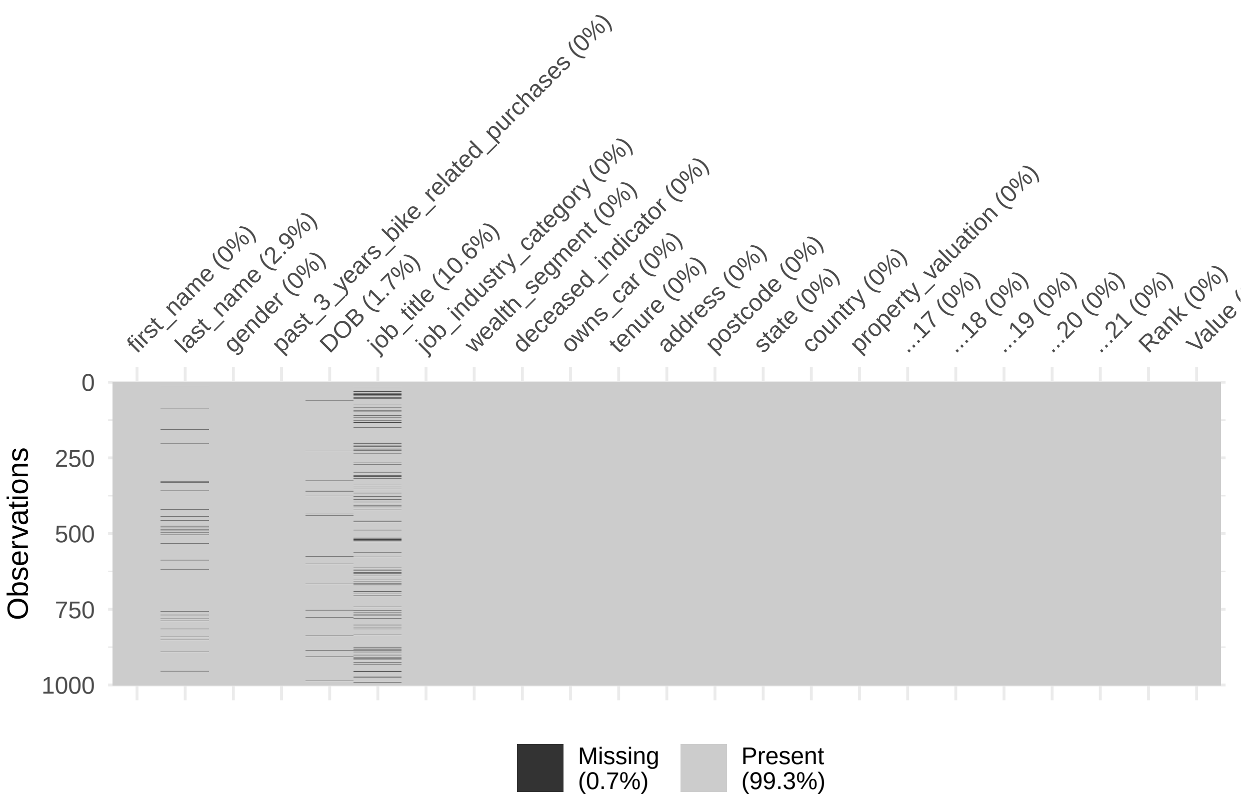
Tidy new data set

```
summary(NewCustomerList)
```

```
##   first_name          last_name            gender         
##  Length:1000        Length:1000        Length:1000       
##  Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character  
##                                                          
##                                                          
##                                                          
##  past_3_years_bike_related_purchases     DOB             job_title        
##  Length:1000                         Length:1000        Length:1000       
##  Class :character                    Class :character   Class :character  
##  Mode  :character                    Mode  :character   Mode  :character  
##                                                                           
##                                                                           
##                                                                           
##  job_industry_category wealth_segment     deceased_indicator   owns_car        
##  Length:1000           Length:1000        Length:1000        Length:1000       
##  Class :character      Class :character   Class :character   Class :character  
##  Mode  :character      Mode  :character   Mode  :character   Mode  :character  
##                                                                                
##                                                                                
##                                                                                
##     tenure          address              postcode             state          
```

```
##  Min.   : 0.00   Length:1000        Length:1000        Length:1000
##  1st Qu.: 7.00   Class :character   Class :character   Class :character
##  Median :11.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :11.39
##  3rd Qu.:15.00
##  Max.   :22.00
##    country          property_valuation     ...17           ...18
##  Length:1000        Length:1000        Min.   :0.400   Min.   :0.4000
##  Class :character   Class :character   1st Qu.:0.560   1st Qu.:0.6375
##  Mode  :character   Mode  :character   Median :0.740   Median :0.8125
##                                        Mean   :0.746   Mean   :0.8389
##                                        3rd Qu.:0.920   3rd Qu.:1.0250
##                                        Max.   :1.100   Max.   :1.3750
##      ...19            ...20            ...21             Rank
##  Min.   :0.4000   Min.   :0.3485   Min.   :   1.0   Min.   :   1.0
##  1st Qu.:0.7000   1st Qu.:0.6481   1st Qu.: 250.0   1st Qu.: 250.0
##  Median :0.9125   Median :0.8469   Median : 500.0   Median : 500.0
##  Mean   :0.9430   Mean   :0.8705   Mean   : 498.8   Mean   : 498.8
##  3rd Qu.:1.1625   3rd Qu.:1.0606   3rd Qu.: 750.2   3rd Qu.: 750.2
##  Max.   :1.7188   Max.   :1.7188   Max.   :1000.0   Max.   :1000.0
##      Value
##  Min.   :0.3400
##  1st Qu.:0.6495
##  Median :0.8600
##  Mean   :0.8817
##  3rd Qu.:1.0750
##  Max.   :1.7188
```

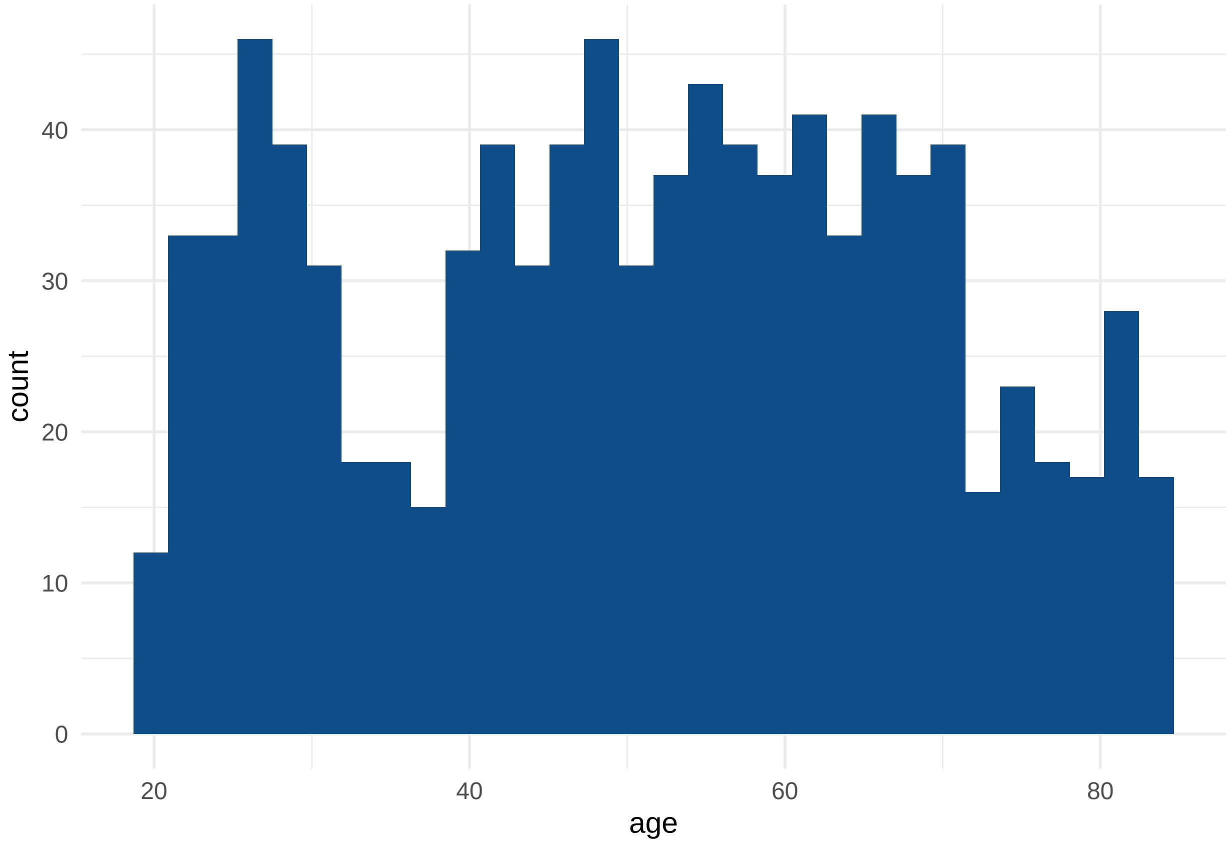
```r
vis_miss(NewCustomerList)
```



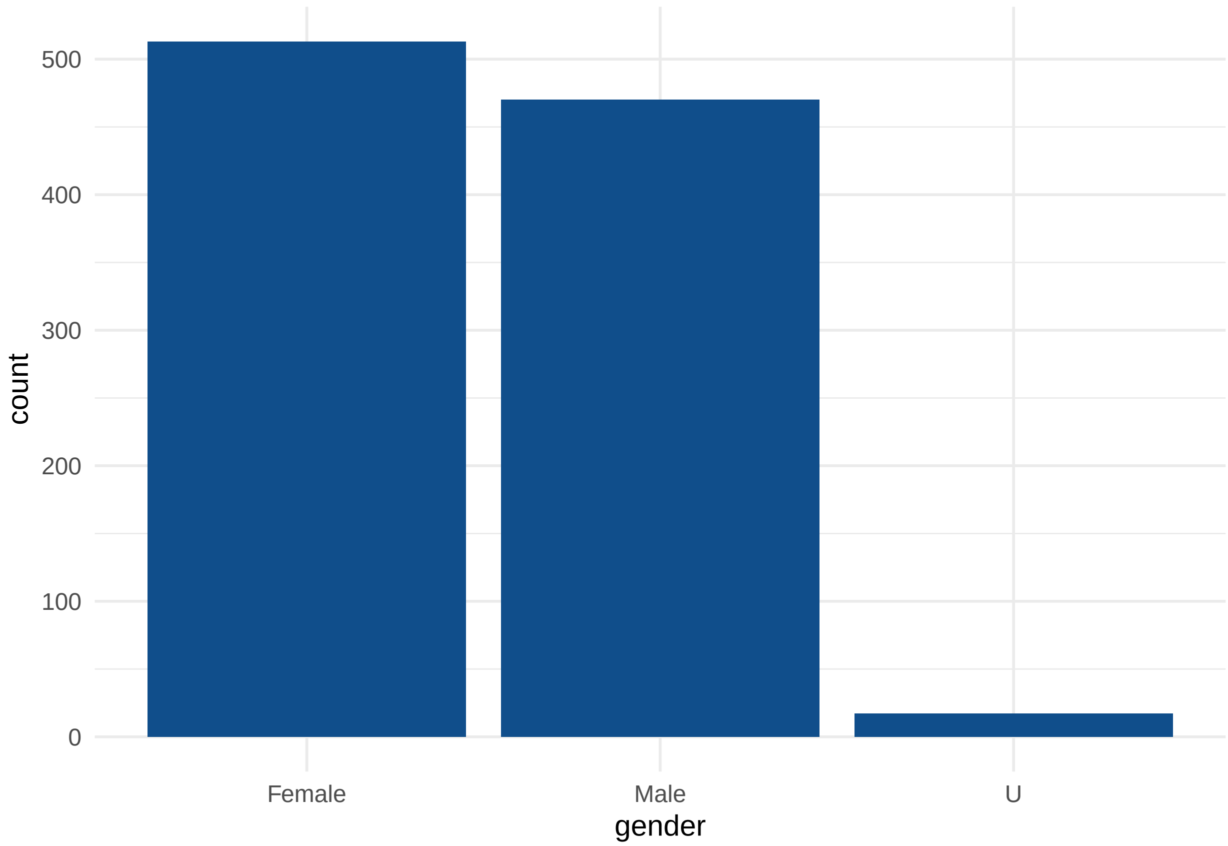
```r
NewCustomerList$gender <- as.factor(NewCustomerList$gender)
NewCustomerList$past_3_years_bike_related_purchases <- as.numeric(NewCustomerList$past_3_years_bike_rela

# *may lose some data
NewCustomerList$DOB <- as.Date(NewCustomerList$DOB, origin = "1970-01-01")
```

```r
NewCustomerList$age <- round((Sys.Date() - NewCustomerList$DOB)/365,2)
```

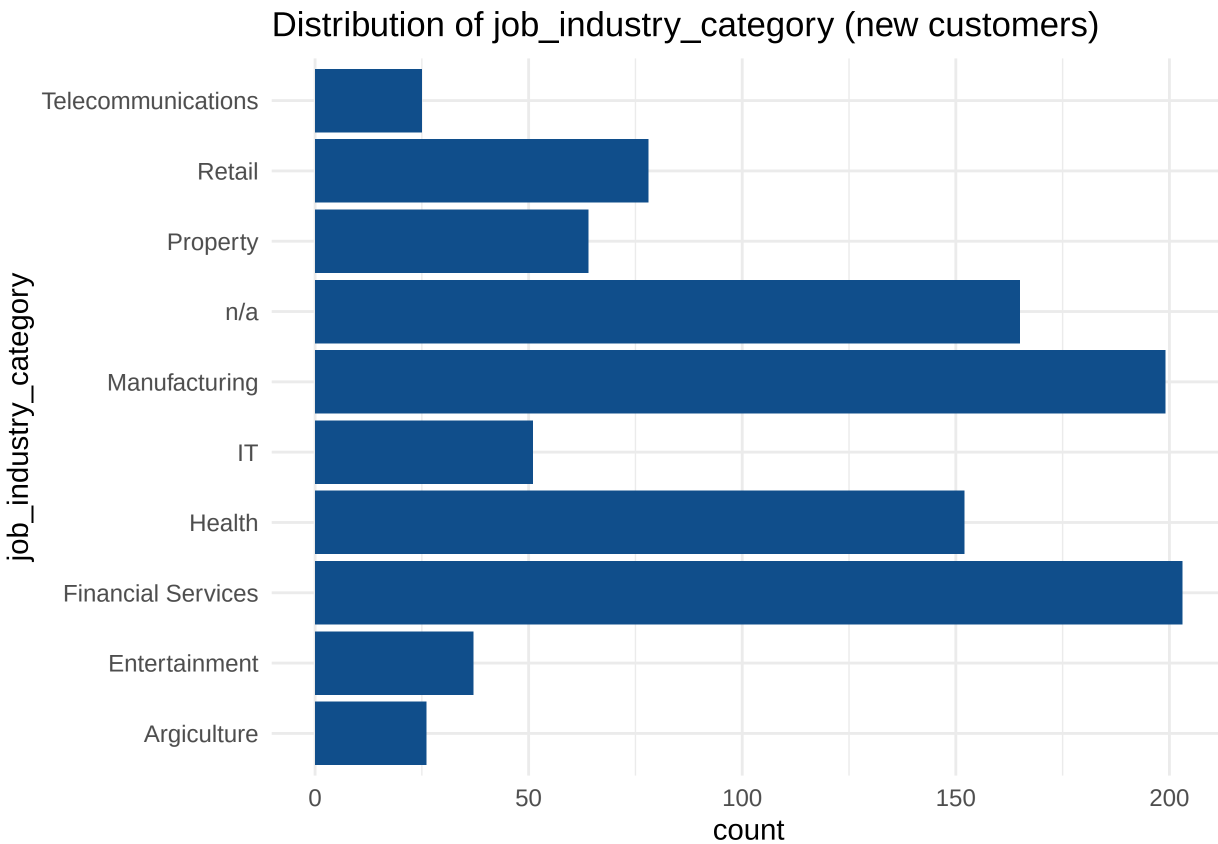
```r
NewCustomerList %>% select(c(first_name, last_name, age)) %>% unique() %>% ggplot() +
  geom_histogram(mapping = aes(x = age), fill = "dodgerblue4")+
  theme_minimal()
```



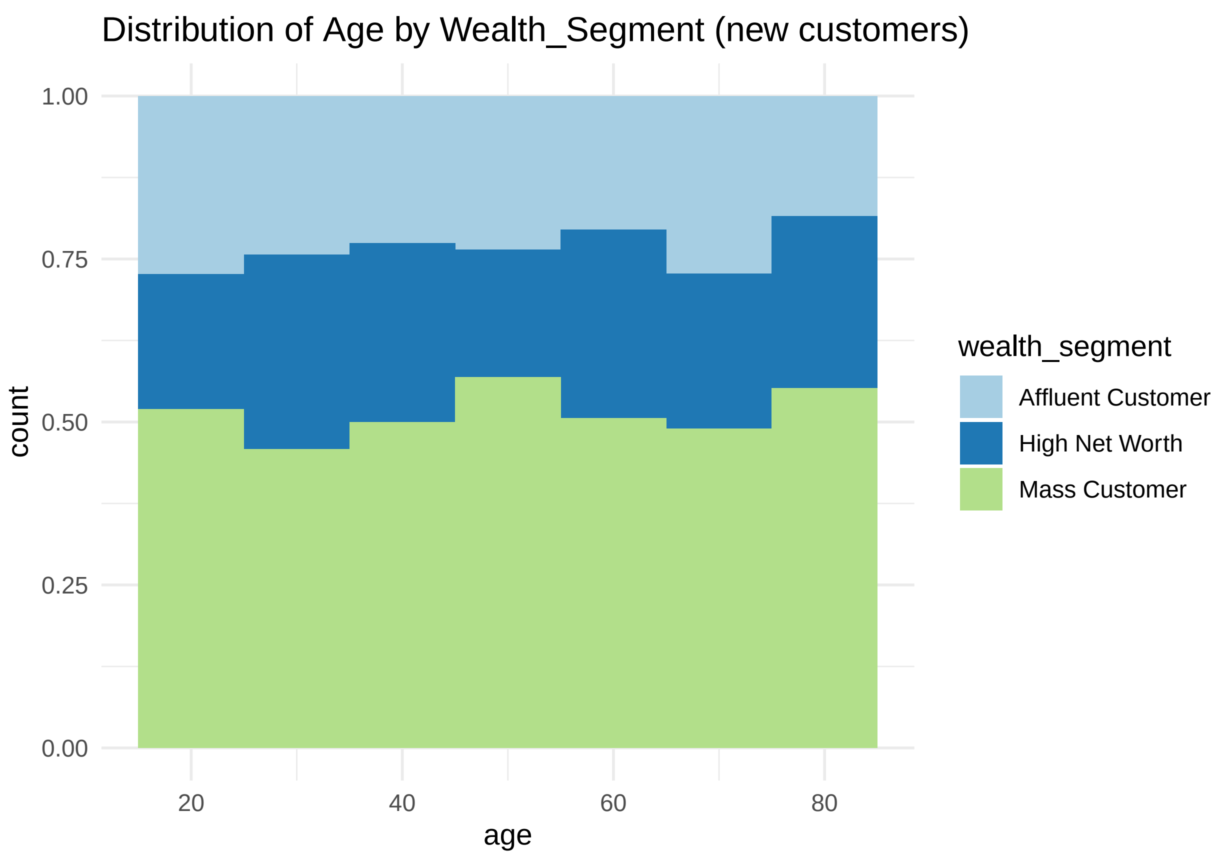

```r
NewCustomerList %>% select(c(first_name, last_name, gender)) %>% unique() %>% ggplot() +
  geom_bar(mapping = aes(x = gender), fill = "dodgerblue4")+
  theme_minimal()
```




```
bar1 <- NewCustomerList %>% select(c(first_name, last_name, job_industry_category)) %>% unique() %>% ggp
  geom_bar(mapping = aes(x = job_industry_category), fill = "dodgerblue4")+
  theme_minimal() +
  ggtitle("Distribution of job_industry_category (new customers)")
bar1 + coord_flip()
```

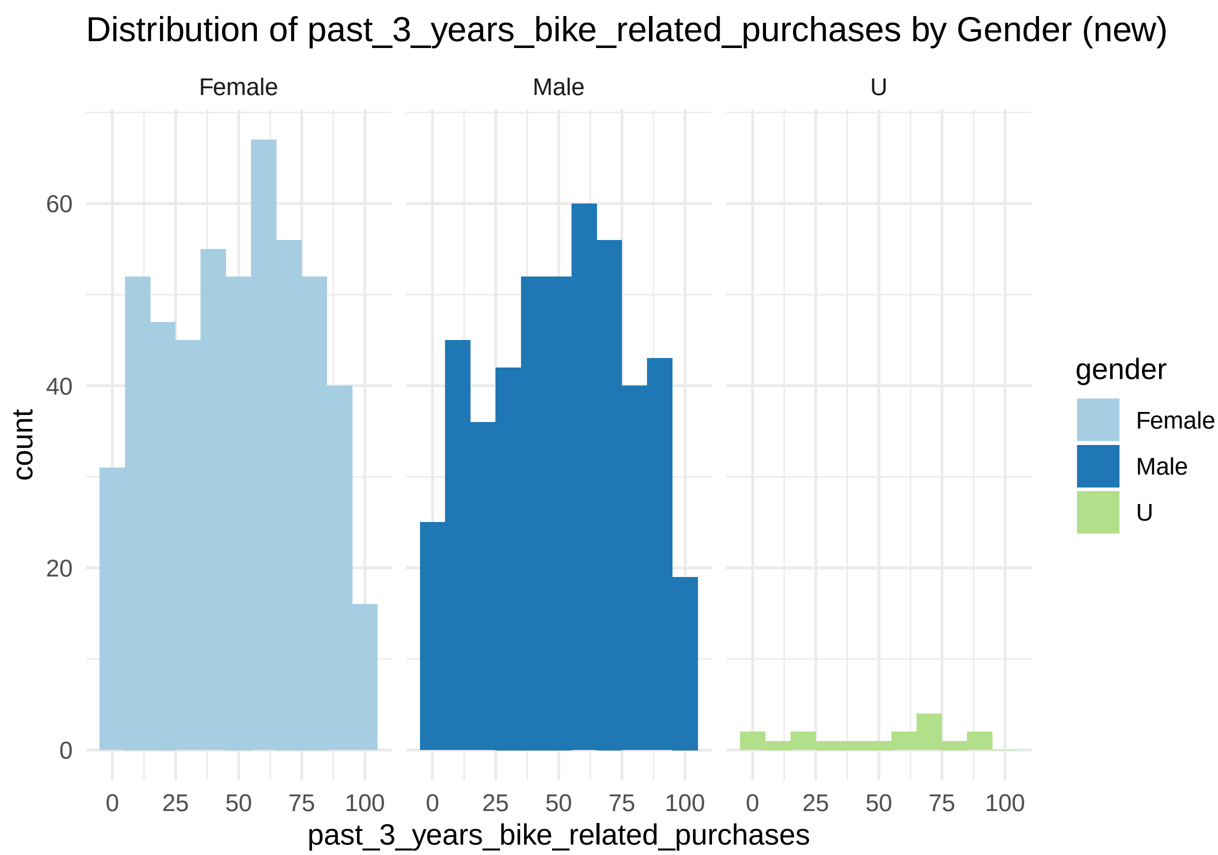



```
NewCustomerList %>% select(c(first_name, last_name, age, wealth_segment)) %>% unique() %>% ggplot() +
  geom_histogram(mapping = aes(x = age, fill = wealth_segment), position = "fill", binwidth = 10) +
  scale_fill_brewer(palette = "Paired") +
  ggtitle("Distribution of Age by Wealth_Segment (new customers)") +
  theme_minimal()
```




```r
NewCustomerList %>% select(c(first_name, last_name, owns_car, state)) %>% unique() %>%
  ggplot(aes(x = state, fill = owns_car)) +
  geom_bar(position = "dodge")+
  scale_fill_brewer(palette = "Paired") +
  ggtitle("Distribution of customers in different states by car owning (new customers)") +
  theme_minimal()
```


```
NewCustomerList %>% select(c(first_name, last_name, gender, past_3_years_bike_related_purchases)) %>% u
  ggplot(aes(x = past_3_years_bike_related_purchases, fill = gender)) +
  geom_histogram(binwidth = 10) +
  facet_grid(~gender)+
  scale_fill_brewer(palette = "Paired") +
  ggtitle("Distribution of past_3_years_bike_related_purchases by Gender (new)") +
  theme_minimal()
```




```
# write.xlsx(rfm_data, file = "rfm_data.xlsx")
```