ANZ_Module_1_Exploratory Data Analysis

Chanyanart KiattipornOpas

10/27/2021

About Data

This task is based on a synthesised transaction dataset containing 3 months worth of transactions for **100 hypothetical customers**. It contains purchases, recurring transactions, and salary transactions.

Data Preparation

1. Load Library

```
library(tidyverse)
library(readxl)
library(naniar)
library(lubridate)
```

2. Load dataset in Rstudio

```
transac <- read_excel("ANZ synthesised transaction dataset.xlsx")
```

3. Preview Dataset

```
glimpse(transac)
```

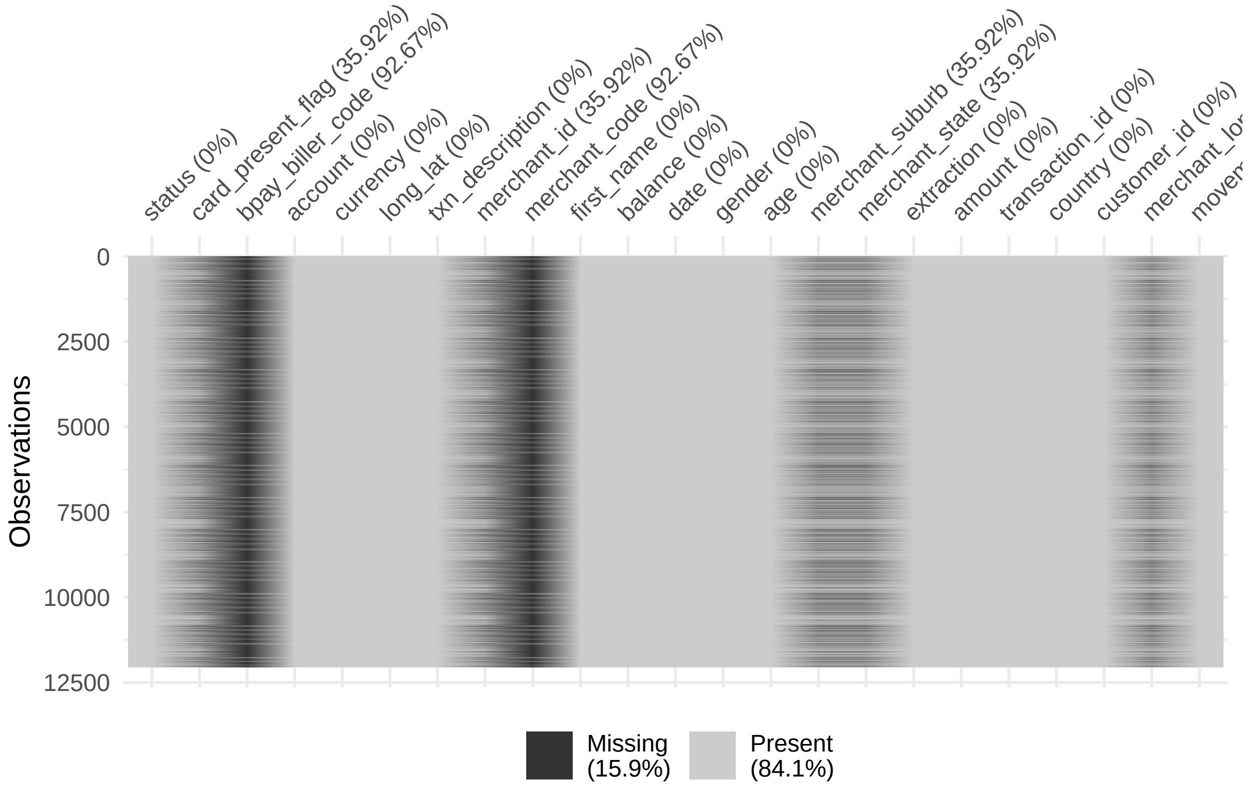
```
## Rows: 12,043
## Columns: 23
## $ status            <chr> "authorized", "authorized", "authorized", "authorize~
## $ card_present_flag <dbl> 1, 0, 1, 1, 1, NA, 1, 1, 1, NA, NA, NA, 1, NA, NA, 1~
## $ bpay_biller_code  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ account           <chr> "ACC-1598451071", "ACC-1598451071", "ACC-1222300524"~
## $ currency          <chr> "AUD", "AUD", "AUD", "AUD", "AUD", "AUD", "AUD", "AU~
## $ long_lat          <chr> "153.41 -27.95", "153.41 -27.95", "151.23 -33.94", "~
## $ txn_description   <chr> "POS", "SALES-POS", "POS", "SALES-POS", "SALES-POS",~
## $ merchant_id       <chr> "81c48296-73be-44a7-befa-d053f48ce7cd", "830a451c-31~
## $ merchant_code     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ first_name        <chr> "Diana", "Diana", "Michael", "Rhonda", "Diana", "Rob~
## $ balance           <dbl> 35.39, 21.20, 5.71, 2117.22, 17.95, 1705.43, 1248.36~
## $ date              <dttm> 2018-08-01, 2018-08-01, 2018-08-01, 2018-08-01, 201~
## $ gender            <chr> "F", "F", "M", "F", "F", "M", "F", "F", "F", "M", "M~
## $ age               <dbl> 26, 26, 38, 40, 26, 20, 43, 43, 27, 40, 19, 43, 27, ~
## $ merchant_suburb   <chr> "Ashmore", "Sydney", "Sydney", "Buderim", "Mermaid B~
## $ merchant_state    <chr> "QLD", "NSW", "NSW", "QLD", "QLD", NA, "VIC", "VIC",~
## $ extraction        <chr> "2018-08-01T01:01:15.000+0000", "2018-08-01T01:13:45~
## $ amount            <dbl> 16.25, 14.19, 6.42, 40.90, 3.25, 163.00, 61.06, 15.6~
## $ transaction_id    <chr> "a623070bfead4541a6b0fff8a09e706c", "13270a2a902145d~
## $ country           <chr> "Australia", "Australia", "Australia", "Australia", ~
## $ customer_id       <chr> "CUS-2487424745", "CUS-2487424745", "CUS-2142601169"~
## $ merchant_long_lat <chr> "153.38 -27.99", "151.21 -33.87", "151.21 -33.87", "~
## $ movement          <chr> "debit", "debit", "debit", "debit", "debit", "debit"~
```

There are 23 Columns and 12,043 Observation.

From the preview of data set above, there are many columns. However, we clearly seen some column shown has many Null value (NA). So, let's check those column first for consideration of dropping NA columns if there are too many NA.

4. Check NA all Columns

```
# Let find NA position and Percentage by Visualization
vis_miss(transac)
```



```
# Create check NA function
check_na <- function(x) {
    sum(is.na(x))
}

# Apply function to all columns
transac %>% summarise_all(.funs = check_na)
```

```
## # A tibble: 1 x 23
##   status card_present_flag bpay_biller_code account currency long_lat
##    <int>             <int>            <int>   <int>    <int>    <int>
## 1      0              4326            11160       0        0        0
## # ... with 17 more variables: txn_description <int>, merchant_id <int>,
## #   merchant_code <int>, first_name <int>, balance <int>, date <int>,
## #   gender <int>, age <int>, merchant_suburb <int>, merchant_state <int>,
## #   extraction <int>, amount <int>, transaction_id <int>, country <int>,
## #   customer_id <int>, merchant_long_lat <int>, movement <int>
```

We can see the same number of missing data **4,326 Null value** in 5 columns which are

```
- card_present_flag
- merchant_id
- merchant_suburb
- merchant_state
- merchant_long_lat
```

We also found the large number of null **(11,160 Null value)** in 2 columns which are

```
- bpay_biller_code
- merchant_code
```

So, Do we need to drop those observations who has NULL value in their information? , Or should we Drop columns instead?

We've seen from Visualization above, there are 11,160 Null value, however we have total observations are 12,043

```
# Look at 'bpay_biller_code' column for an example.

sum(complete.cases(transac$bpay_biller_code))
```

```
## [1] 883
```

```
# only columns of 883 customers are complete

mean(complete.cases(transac$bpay_biller_code))
```

```
## [1] 0.0733206
```

```
# only 7% completed columns.
```

As we seen the ‘bpay_biller_code’ column above, it look like **7% column completed**. So, if we drop these 2 columns, it will not affect that much.

```
# to drop column from dataset
transac <- transac %>% select(-bpay_biller_code,
                              -merchant_code)
```

However, Do we also need to drop NA value that appeared on 5 columns? There are the same number of NULL value among 5 columns (4,326 observations of 5 columns), we will check again later.

So, now we still have 12,043 observations, but 21 columns.

Data Exploration

1. Discrete Data

1.1 Account and Customer_id

Account and Customer-id are the number of each customer which use for identifying customers who is owning the bank account.

```
# Change Data Type from chr. to factor
transac$account <- as.factor(transac$account)
transac$customer_id <- as.factor(transac$customer_id)
```

```
# Number of account and customer_id should be equal.
length(unique(transac$account))
```

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

```
length(unique(transac$customer_id))
```

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

Both data has 100 unique account and customer id. So, we can assume that the data are sync correctly. We also know that we have 100 customers in this data set.

1.2 Country and Currency

Identifying where the collected data are from, and what currency they used.

```
# Check distinct value of each columns with unique()
unique(transac$country)
```

```
## [1] "Australia"
```

```
unique(transac$currency)
```

```
## [1] "AUD"
```

It look like all observations in these columns are the same value. So, we can drop these columns out because there are not providing additional information for analyzing.

```
transac <- transac %>%
            select(-country, -currency)
```

Even though we drop them out, we can keep in mind that this data set are collected from customers that are only in "Australia" and using "AUD currency".

Move on to other columns !

1.3 Card_present_flag

Card Present means the traditional transaction with the debit/credit cards on the card reader machines.

Card Not Present means other method of payment, such as Online shopping, buy on website, Recurring or subscription billing, Electronic invoicing, Orders taken over the phone, Payment apps on smart phones.

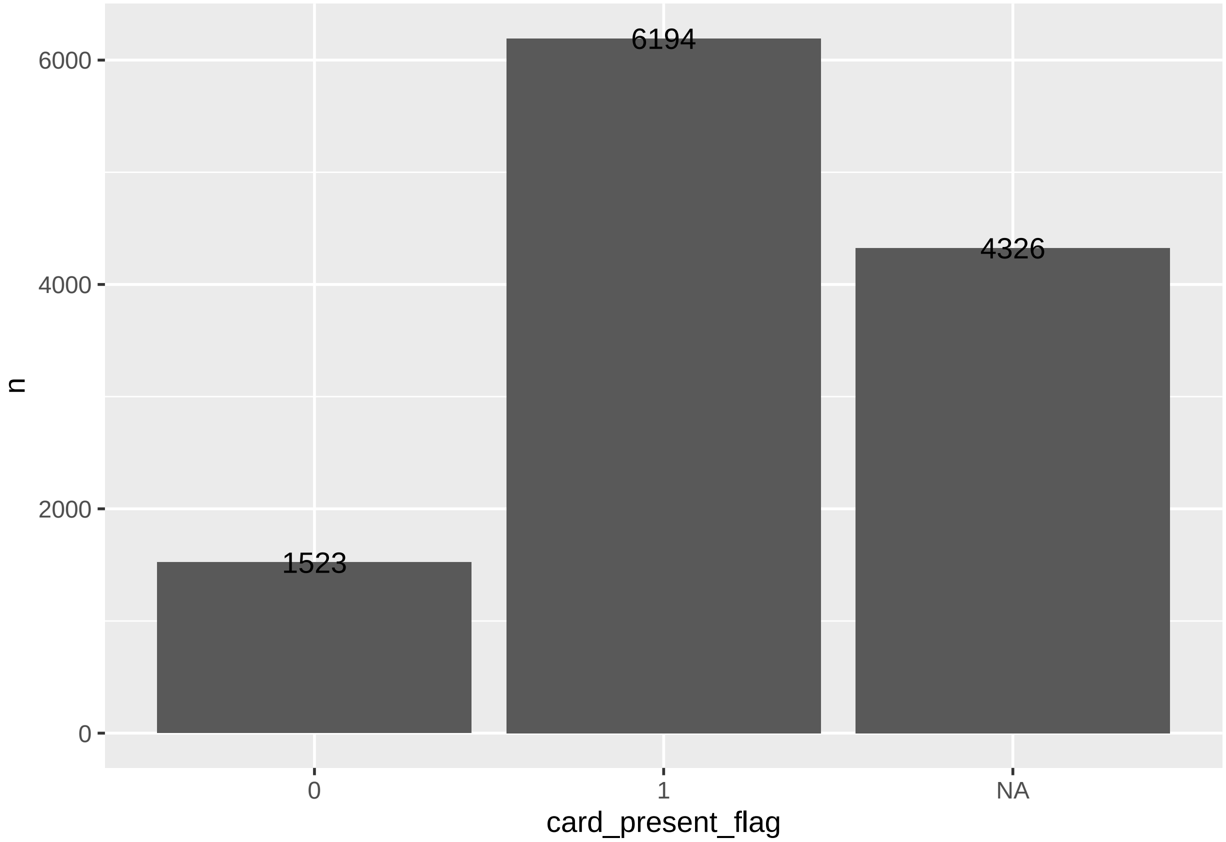
```
unique(transac$card_present_flag)
```

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

There are 2 types which shown (0 / 1) and NA value. So, we should convert Numeric type to Factor type

```
# Change Data Type from Numerical to Factor
transac$card_present_flag <-  as.factor(transac$card_present_flag)
```

```
# To look at the proportion of 0  and  1
card <- transac %>% count(card_present_flag)

ggplot(card,aes(card_present_flag, n)) +
    geom_col(na.rm = FALSE) +
    geom_text(aes(label = n),
              size = 4)
```



We knew that main customers was identified as 1, a triple than the customer is defined as 0. But there are a big size of NA which comes to the second.

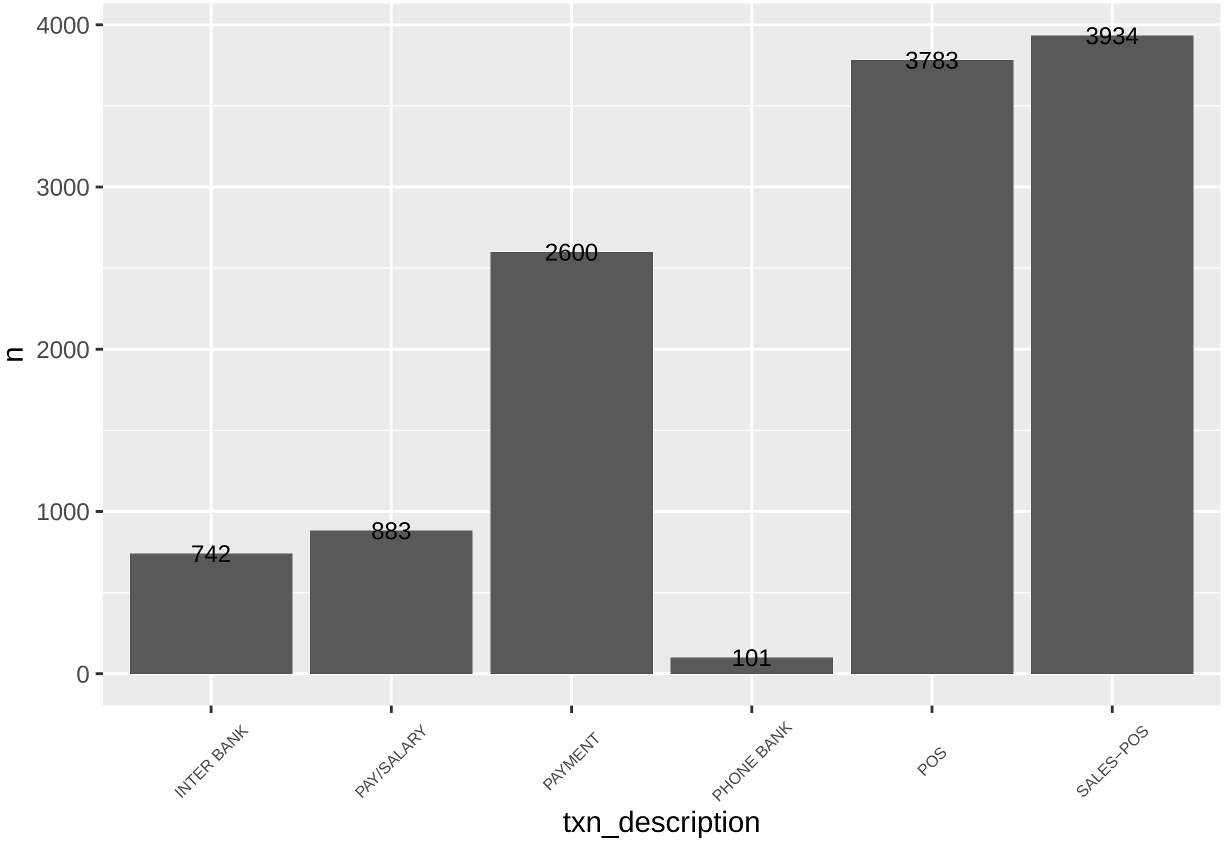
1.4 txn_description

```
# check distinct value of txn_description
unique(transac$txn_description)
```

```
## [1] "POS"        "SALES-POS"  "PAYMENT"    "INTER BANK" "PAY/SALARY"
## [6] "PHONE BANK"
```

```
# Change Data Type from chr. to factor
transac$txn_description <-
    as.factor(transac$txn_description)
```

```
transac %>% count(txn_description) %>%
  ggplot(aes(txn_description,n)) +
      geom_col(na.rm = FALSE) +
      geom_text(aes(label = n),
                    size = 3) +
            theme(axis.text.x = element_text(angle = 45,
                                             vjust = 0.5,
                                               size = 6))
```



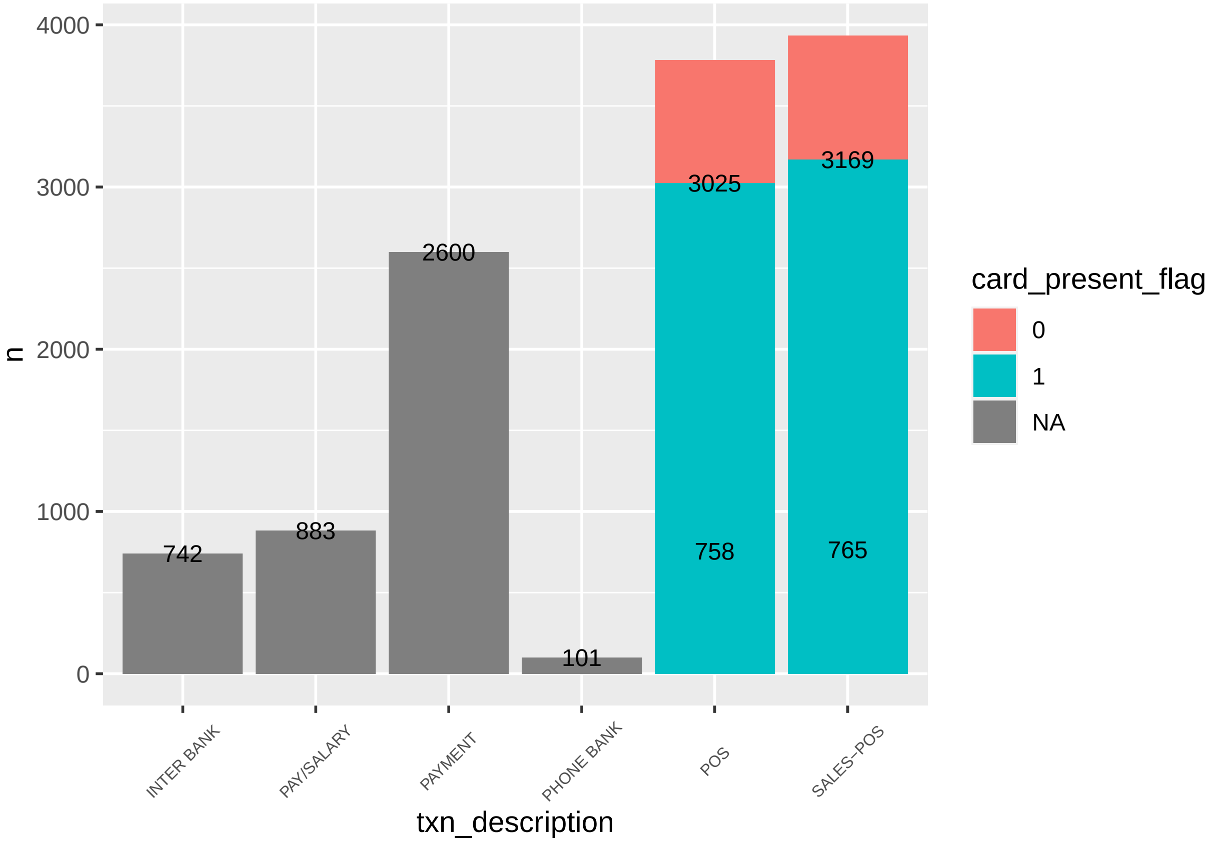
There are 6 usage types of transaction(txn) on Bank Account.

- **POS and SALES-POS** The quantity of this 2 types quite similar. SALSE-POSE is a bit higher than POS

A point of sale (POS) is a place where a customer executes the payment for goods or services and where sales taxes may become payable. A POS transaction may occur in person or online, with receipts generated either in print or electronically.

- **Payment method** is the third of transaction method.
- **Inter Bank** and **Pay/Salary** are relatively similar.
- The least quantity of transaction method is **Phone Bank**

```
# Add on 2 columns into plot

txn_card <- transac %>%
            count(txn_description, card_present_flag)

ggplot(txn_card, aes(txn_description,n,
                    fill = card_present_flag)) +
      geom_col(na.rm = FALSE) +
      geom_text(aes(label = n),
                    size = 3) +
            theme(axis.text.x = element_text(angle = 45,
                                             vjust = 0.5,
                                               size = 6))
```



POS and **SALES-POS** are only two columns that show the amount of customer use the card or paid without card while doing their transactions on the stores.

Null Value(NA) are from 4 transaction types that are **Inter Bank, Pay/Salary, Payment and Phone Bank**. And We understand that those type no need to have information of card_present-flag columns (Null) because they do not use the credit card to buy the products.

So, Back to the question "Should we drop NA value on any rows that has NA in the row?"

The answer should be "No" - because it may give some insight from data that come from these transactions.

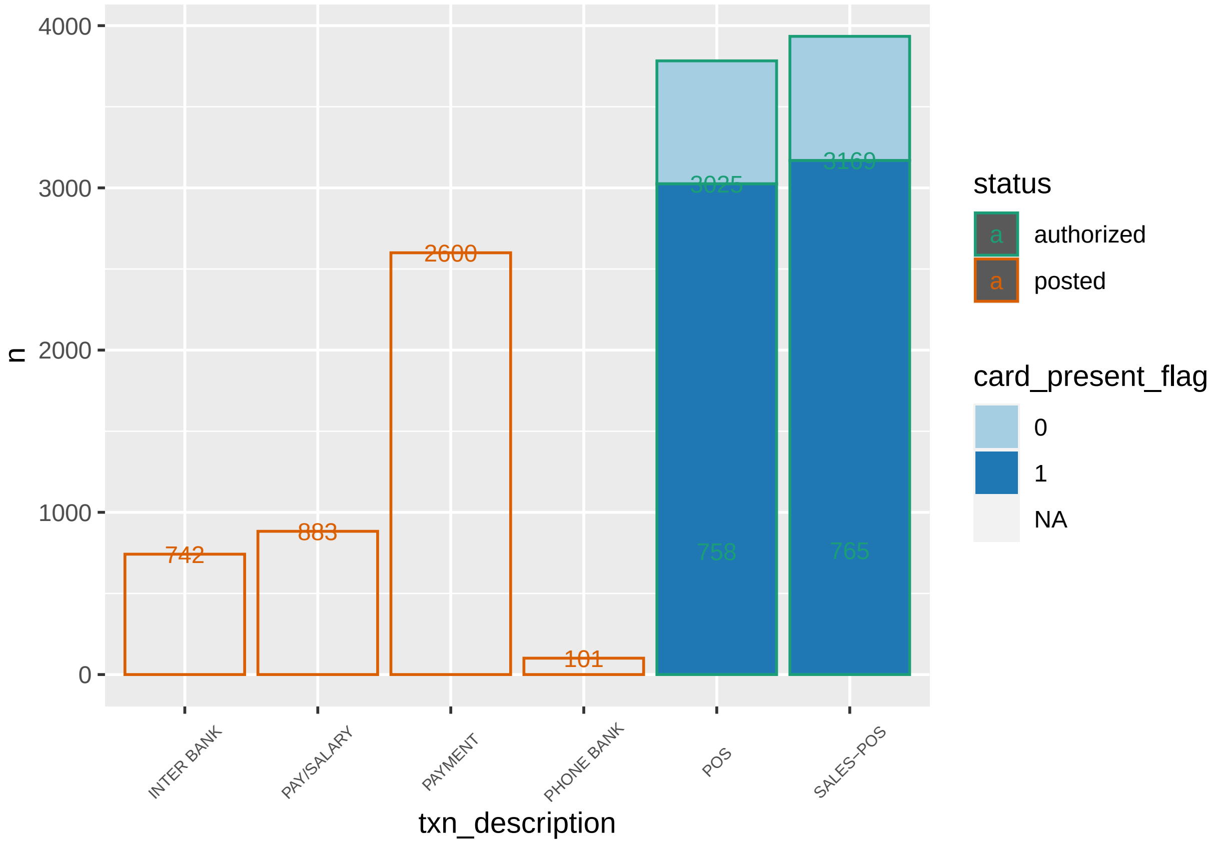
We will keep them as NA for Null value

1.5 Status

```
transac %>% count(status)
```

```
## # A tibble: 2 x 2
##   status         n
##   <chr>      <int>
## 1 authorized  7717
## 2 posted      4326
```

```
status_txn_card <- transac %>%
            count(status,
                  txn_description,
                  card_present_flag)

ggplot(status_txn_card, aes(txn_description,n,
                    fill = card_present_flag,
                    color = status)) +
      geom_col(na.rm = FALSE) +
    scale_color_brewer(palette = "Dark2") +
    scale_fill_brewer(palette = "Paired") +
      geom_text(aes(label = n),
                    size = 3) +
            theme(axis.text.x = element_text(angle = 45,
                                            vjust = 0.5,
                                              size = 6))
```



Status “Authorized” only show in **POS and SALES-POS“** payment method. Apart from those two are”Posted" status.

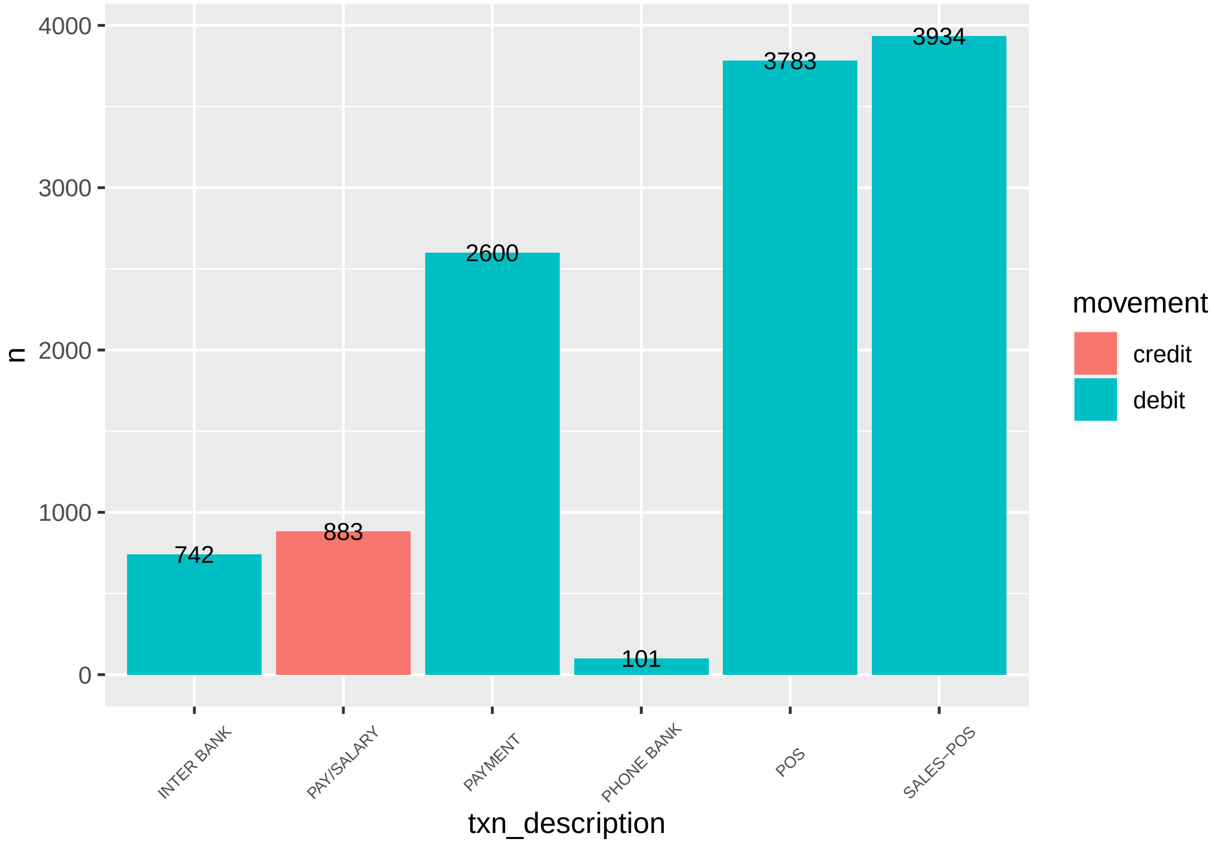
1.6 Movement

```
transac$movement <- as.factor(transac$movement)
transac %>% count(movement)
```

```
## # A tibble: 2 x 2
##   movement     n
##   <fct>     <int>
## 1 credit      883
## 2 debit     11160
```

"Debit" showed far out number of usage than "Credit"

```
movement_txn <- transac %>%
            count(movement,
                  txn_description)

ggplot(movement_txn, aes(txn_description,n,
                    fill = movement)) +
      geom_col(na.rm = FALSE) +
      geom_text(aes(label = n),
                    size = 3) +
            theme(axis.text.x = element_text(angle = 45,
                                             vjust = 0.5,
                                               size = 6))
```



Only Pay/Salary Method showed the movement of "Credit", Others are "Debit"

1.7 Merchant_States

```
transac$merchant_state <- as.factor(transac$merchant_state)

# Check State
unique(transac$merchant_state)
```

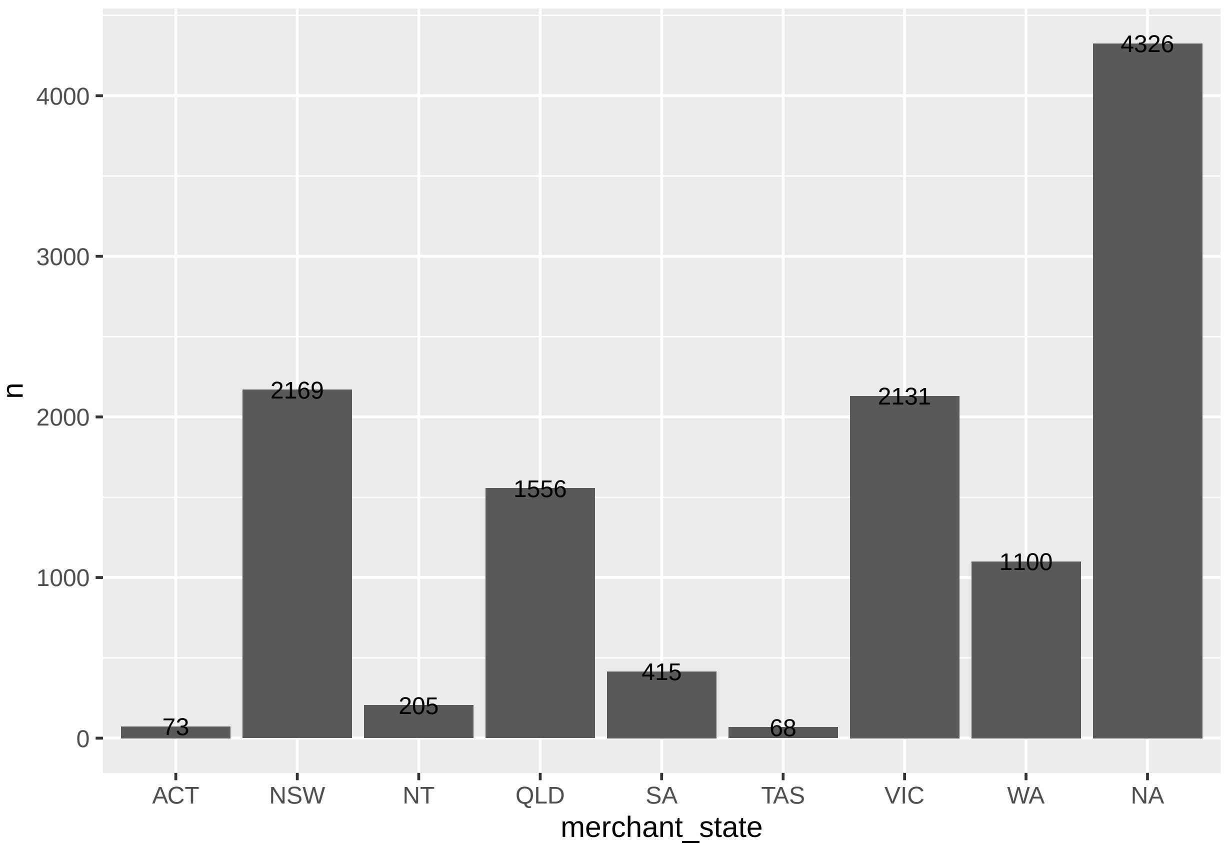
```
## [1] QLD  NSW  <NA> VIC  WA   SA   NT   TAS  ACT
## Levels: ACT NSW NT QLD SA TAS VIC WA
```

There are 8 states, without duplicated or misspelling

```
states <- transac %>% count(merchant_state) %>%
    arrange(desc(n))

states
```

```
## # A tibble: 9 x 2
##   merchant_state     n
##   <fct>          <int>
## 1 <NA>            4326
## 2 NSW             2169
## 3 VIC             2131
## 4 QLD             1556
## 5 WA              1100
## 6 SA               415
## 7 NT               205
## 8 ACT               73
## 9 TAS               68
```

```
ggplot(states, aes(merchant_state, n)) +
    geom_col() +
    geom_text(aes(label = n),
              size = 3)
```



1.8 Merchant_Suburb

```
# Check Merchant_Suburb (Not include NA)
length(unique(na.omit(transac$merchant_suburb)))
```

```
## [1] 1609
```

There are 1,609 suburbs.

```
transac %>% count(merchant_suburb) %>%
    arrange(desc(n)) %>%
    head(10)
```

```
## # A tibble: 10 x 2
##    merchant_suburb     n
##    <chr>           <int>
##  1 <NA>             4326
##  2 Melbourne         255
##  3 Sydney            233
##  4 Southport          82
##  5 Brisbane City      79
##  6 Chatswood          55
##  7 Perth              45
##  8 Adelaide           44
##  9 Broadbeach         41
## 10 Mount Gambier      41
```

Melbourne (255) and Sydney(233) are top two of Merchant Suburbs where occurred transaction which far out number from others.

1.9 Gender

```
transac$gender <- as.factor(transac$gender)

# Extract unique customer_id
u_id <- transac %>% group_by(customer_id, gender) %>%
            summarize(n_count = n_distinct(customer_id))

# count Gender
u_id %>% as.data.frame() %>% count(gender)
```

```
##   gender  n
## 1      F 44
## 2      M 56
```

Male customers are higher than Female customers.

2. Date Manipulation

2.1 Start Date and End Date

```
min(transac$date)
```

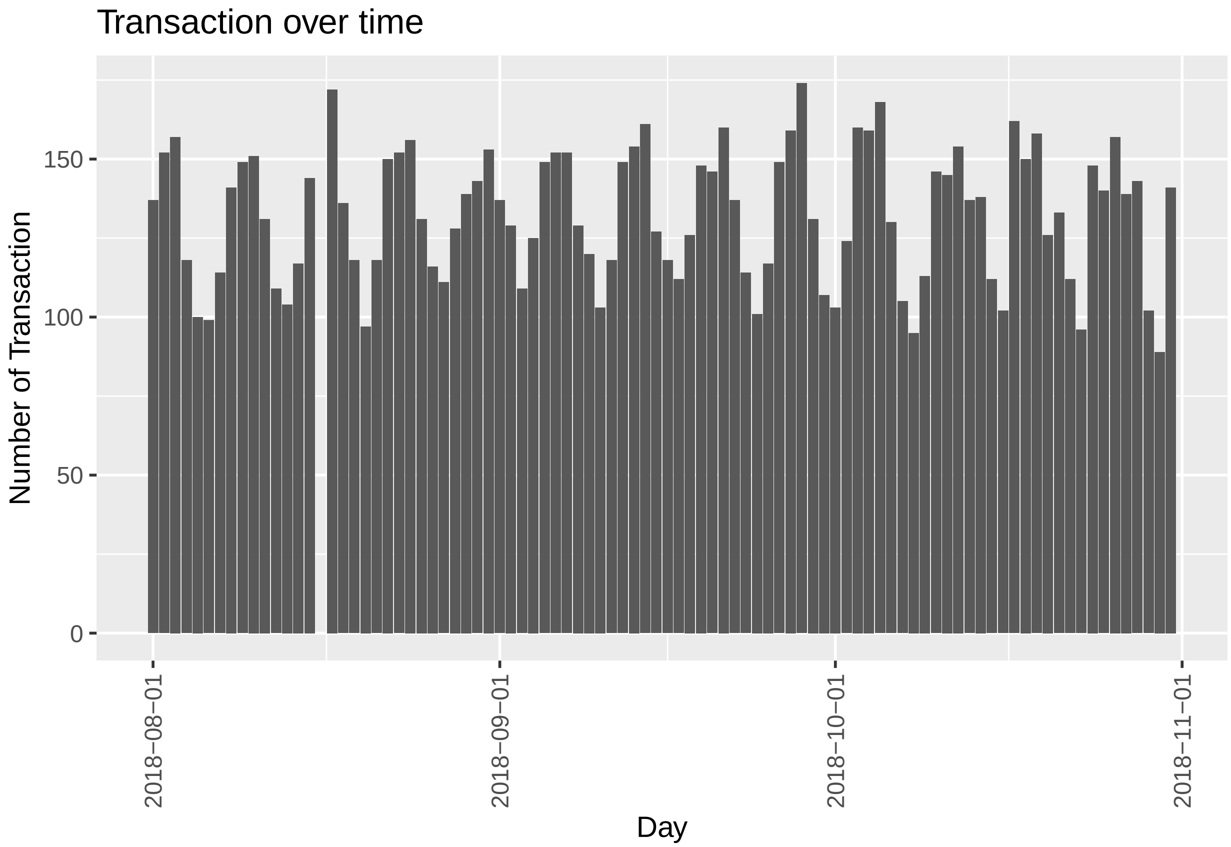
```
## [1] "2018-08-01 UTC"
```

```
max(transac$date)
```

```
## [1] "2018-10-31 UTC"
```

Data is collected from 01-08-2018 to 31-10-2018

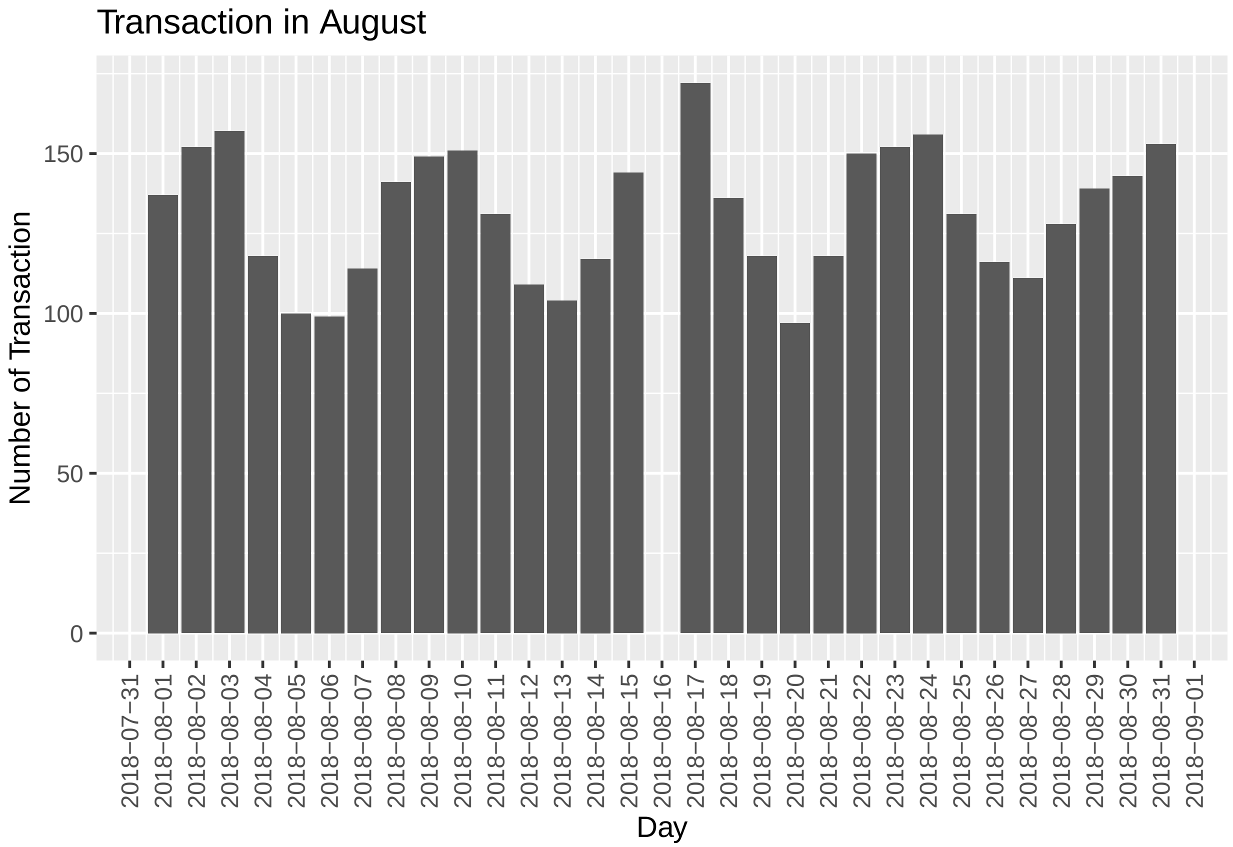
```
transac$date <- as.Date(transac$date)
date_freq <- transac %>% count(date) %>% arrange(desc(n))

p_date_freq <- ggplot(date_freq, aes(date, n)) +
    geom_col() +
    labs(x = "Day",
         y = "Number of Transaction",
         title = "Transaction over time") +
    scale_x_date(breaks = "1 month") +
    theme(axis.text.x = element_text(angle = 90,
                                     vjust = 0.5))
p_date_freq
```



It look like there are a pattern of a peak days. But on the graph showed there is missing data in August, so let look on which day?

2.2 Missing Date

```
august <- transac %>% filter(date >= "2018-08-01" &
                             date <= "2018-08-31") %>%
           count(date)

p_august <- ggplot(august, aes(date, n)) +
    geom_col() +
    labs(x = "Day",
         y = "Number of Transaction",
         title = "Transaction in August") +
    scale_x_date(breaks = "1 day") +
    theme(axis.text.x = element_text(angle = 90,
                                     vjust = 0.5))
p_august
```



There are no transaction data of "2018-08-16"

Then, let's check Peak days from top 10 transaction dates.

2.3 Peak Date

```
date_freq %>% head(10)
```

```
## # A tibble: 10 x 2
##    date           n
##    <date>     <int>
##  1 2018-09-28   174
##  2 2018-08-17   172
##  3 2018-10-05   168
##  4 2018-10-17   162
##  5 2018-09-14   161
##  6 2018-09-21   160
##  7 2018-10-03   160
##  8 2018-09-27   159
##  9 2018-10-04   159
## 10 2018-10-19   158
```

Top 10 of highest transaction date.

```
wday("2018-09-29", label = TRUE)  # Sat
```

```
## [1] Sat
## Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
```

```
wday("2018-09-01", label = TRUE)  # Sat
```

```
## [1] Sat
## Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
```

```
wday("2018-09-28", label = TRUE)  # Fri
```

```
## [1] Fri
## Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
```

```
wday("2018-10-05", label = TRUE)  # Fri
```

```
## [1] Fri
## Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
```

```
wday("2018-08-17", label = TRUE)  # Fri
```

```
## [1] Fri
## Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
```

```
wday("2018-09-21", label = TRUE)  # Fri
```

```
## [1] Fri
## Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
```

```
wday("2018-09-22", label = TRUE)  # Sat
```

```
## [1] Sat
## Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
```

```
wday("2018-10-20", label = TRUE)  # Sat
```

```
## [1] Sat
## Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
```

```
wday("2018-08-18", label = TRUE)  # Sat
```

```
## [1] Sat
## Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
```

```
wday("2018-08-31", label = TRUE)  # Fri
```

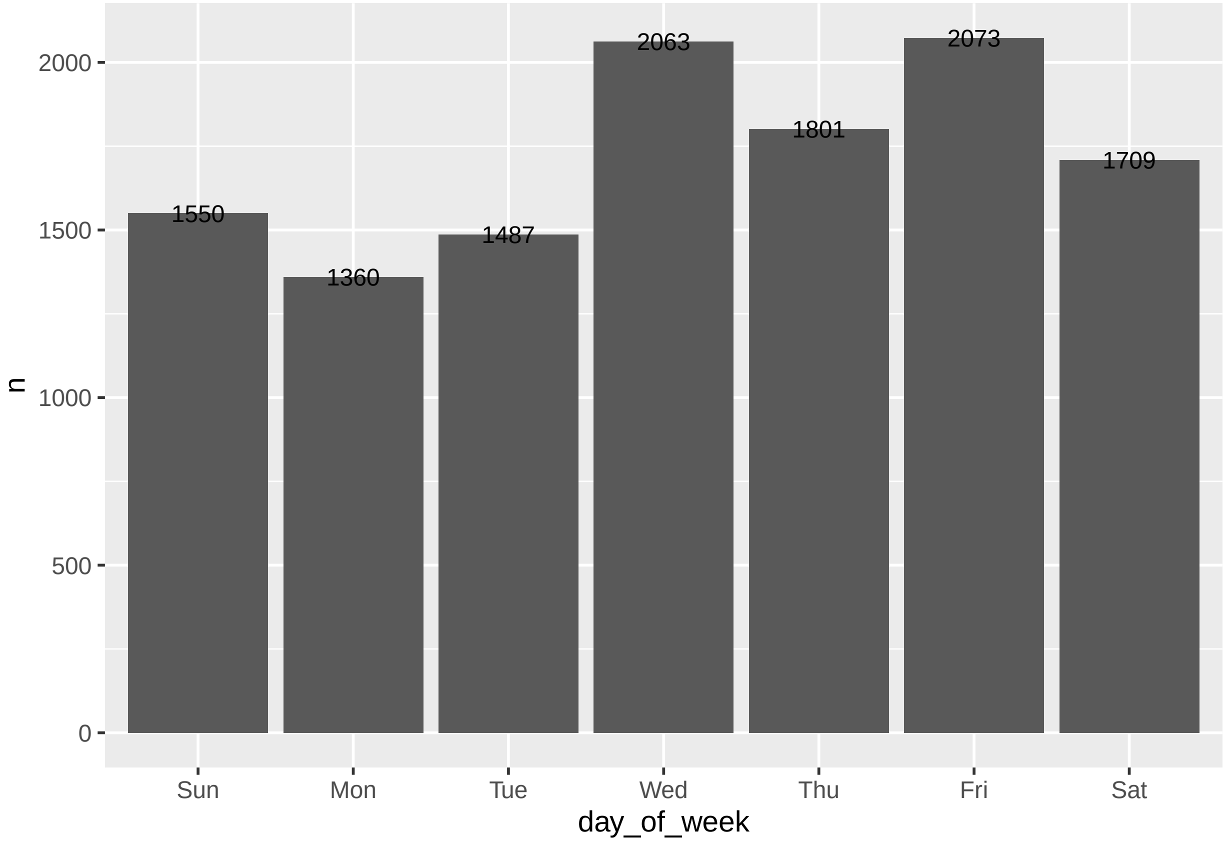
```
## [1] Fri
## Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
```

Peak day of transaction are Friday and Saturday.

2.4 Create Day_of_the_week Columns

```
transac$date <- as.POSIXct(transac$date ,
                      format = "%Y-%m-%d")

transac$day_of_week <- wday(transac$date, label = TRUE)
```

```
transac %>%
  count(day_of_week) %>%
  arrange(desc(n)) %>%
  ggplot(aes(day_of_week, n)) +
  geom_col() +
  geom_text(aes(label = n),
            size = 3)
```



Most transaction are on Friday and Wednesday

3. Time Extraction

Let's extract the time from "extraction" column

```
# Replace the T with blank space to separate date&time

transac$extraction <- str_replace_all(transac$extraction,
                                      "T", " ")
```

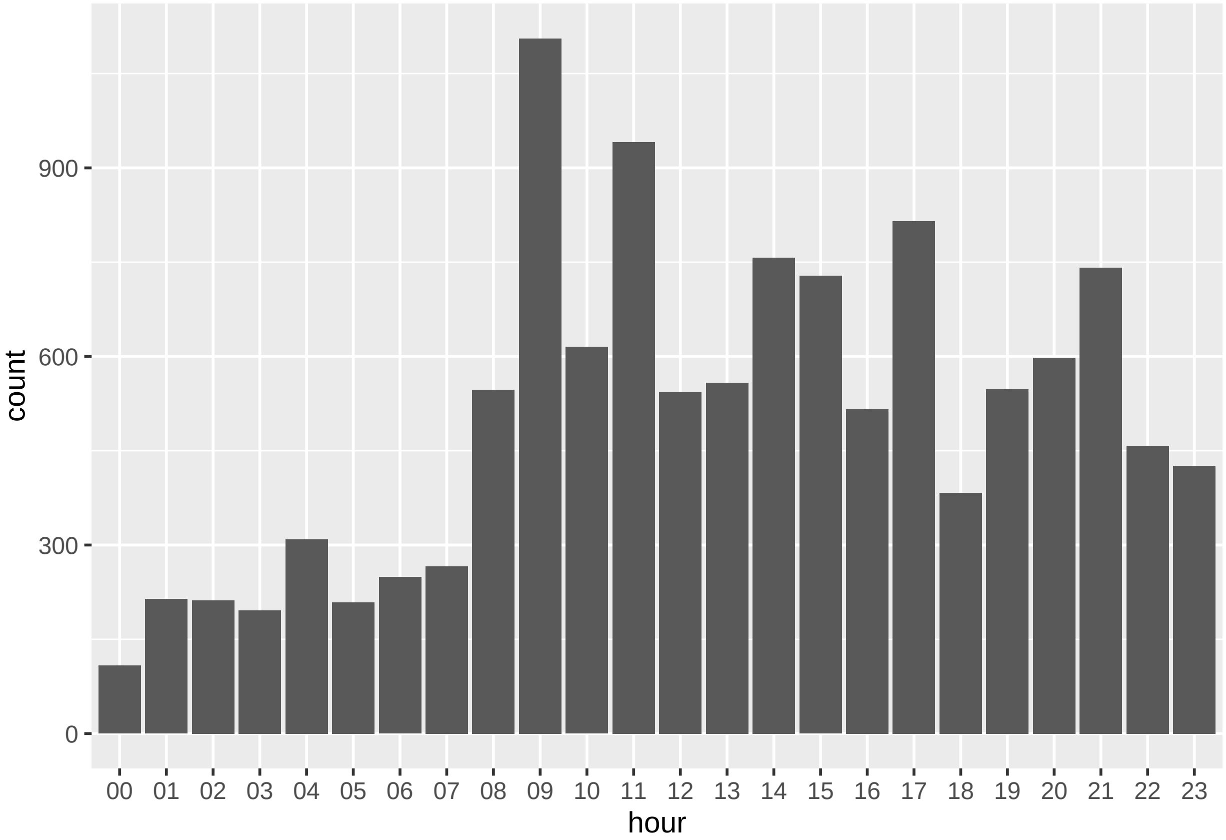
```
# Remove the .000+0000 from date&time
transac$extraction <- str_replace_all(transac$extraction,
                                "[.][0]++[+][0]++", " ")
```

To Extract only time

```
# Change Char to Date type with identify date format
transac$extraction <- as.POSIXct(transac$extraction,
                      format = "%Y-%m-%d %H:%M:%S")

# Create new column and extract only time
transac$time <- format(transac$extraction,
                       format = "%H:%M:%S")

# Extract only hours
transac$hour <- format(transac$extraction, format = "%H")

# Extract only month
transac$month <- format(transac$extraction, format = "%m")

# Delete extraction column
transac <- transac %>% select(-extraction)
```

3.1 Peak Time of transaction

```
# Plot graph to find the Peak Time of Transaction.
transac$hour <-  as.factor(transac$hour)

ggplot(transac,aes(hour)) +
    geom_bar()
```



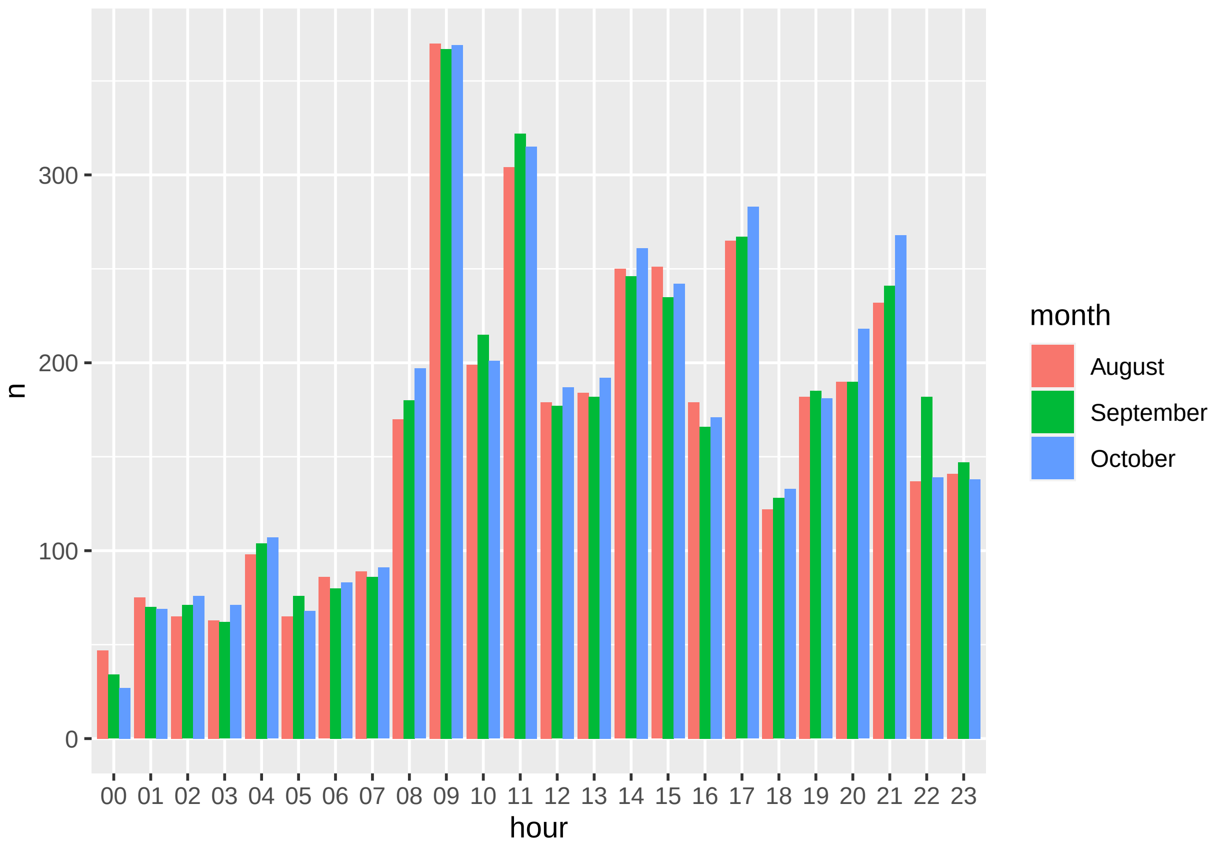
We found that during 09:00 - 10:00 is the peak time of Transaction across the states.

3.2 Peak Time across 3 months

```
# Are there any different of Peak Time in each months?

transac$month <-  factor(transac$month,
                         levels = c("08","09","10"),
                         labels = c("August",
                                    "September",
                                    "October"))
```

```
byMonth <- transac %>% select(hour, month) %>%
                        group_by(month) %>%
                        count(hour)
```

```
ggplot(byMonth, aes(hour,n, fill=month)) +
    geom_col(position = "dodge")
```



There is a same pattern of Peak Time across 3 months. Example: A peak Time 9:00 am. of transaction amount are more than 300+ in those three month.

4. Location data

Separate Lat and Long

```
# To separate Lat - Long of customers
transac$long_lat <- as.character(transac$long_lat)

transac <- transac %>% separate(long_lat,
                     into = c("long_cust", "lat_cust"),
                     sep = " ", fill = "right")
```

```
# To separate Lat - Long of merchant

transac$merchant_long_lat <-
    as.character(transac$merchant_long_lat)

transac <- transac %>% separate(merchant_long_lat,
                     into = c("long_merch", "lat_merch"),
                     sep = " ", fill = "right")
```

5. Continuous Data

There are 3 columns of continuous data which are

```
- Balance
- Amount
- Age
```

This data are from ANZ Bank of Australia, which Bank account also keep individual money. So, there are a large range of **Balance and Amount columns**, which depends on each individual customers.

5.1 Balance

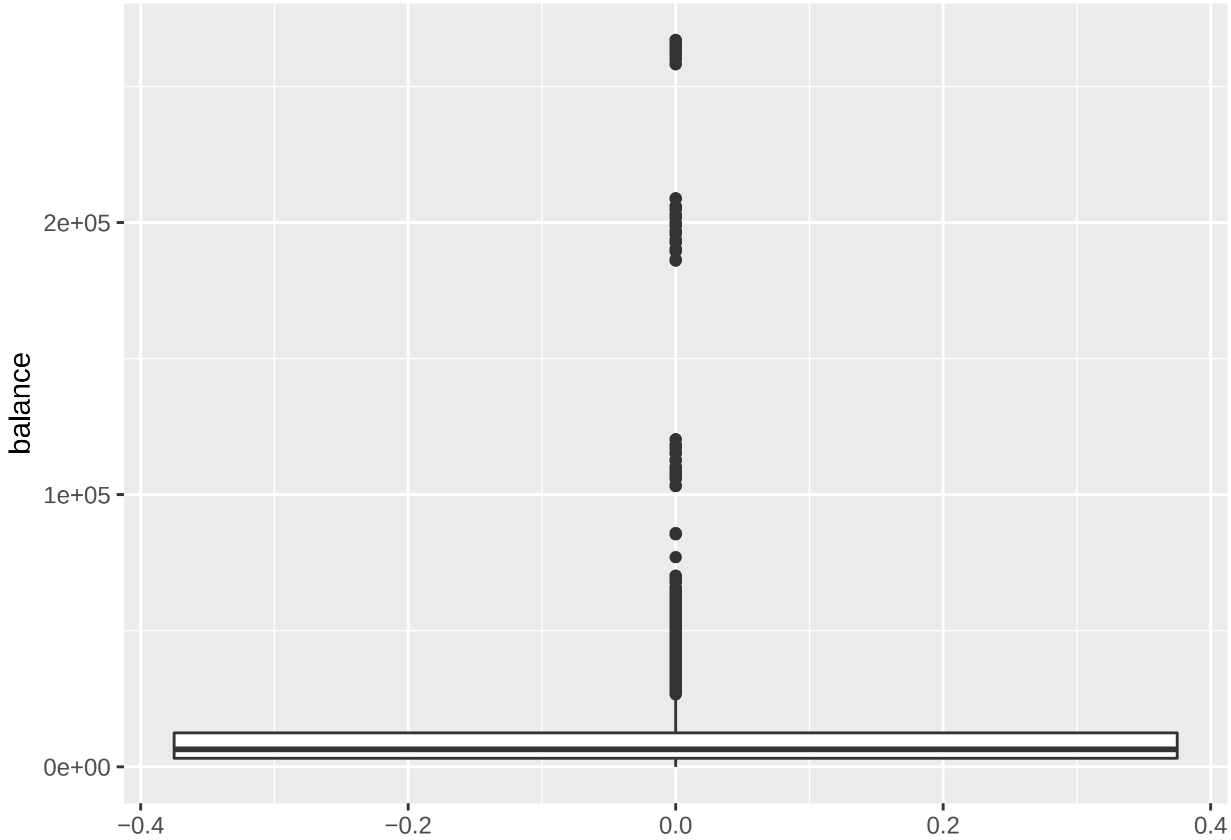
```
summary(transac$balance)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##      0.24   3158.58   6432.01  14704.20  12465.94 267128.52
```

It is clear that there are a big range of data, And Mean is far higher than Median. But it can understandable because Balance will show the remaining after transactions, so Balance data always recount.

Just plot to look into what position of data look like.

```
# boxplot to visualize outlier
ggplot(transac, aes(balance)) +
    geom_boxplot() +
    coord_flip()
```



We can see the people who has high of Balance column after their transaction, by checking Outlier.

```
# Which observation are outliers?
out_balance <- boxplot.stats(transac$balance)$out
out_balance_iden <- which(transac$balance
                          %in% c(out_balance))
transac[out_balance_iden, ]
```

```
## # A tibble: 1,293 x 24
##    status     card_present_flag account        long_cust lat_cust txn_description
##    <chr>      <fct>             <fct>          <chr>     <chr>    <fct>
##  1 posted     <NA>              ACC-2014856841 144.99    -37.90   INTER BANK
##  2 authorized 0                 ACC-2615038700 145.35    -38.03   POS
##  3 authorized 1                 ACC-721712940  145.09    -37.82   POS
##  4 authorized 1                 ACC-2615038700 145.35    -38.03   POS
##  5 authorized 1                 ACC-2615038700 145.35    -38.03   SALES-POS
##  6 authorized 1                 ACC-1334819143 145.13    -37.70   POS
##  7 posted     <NA>              ACC-3879258709 143.83    -37.66   PAYMENT
##  8 authorized 1                 ACC-3827517394 151.12    -33.89   POS
##  9 authorized 0                 ACC-2615038700 145.35    -38.03   POS
## 10 posted     <NA>              ACC-38923874   151.27    -33.90   PAYMENT
## # ... with 1,283 more rows, and 18 more variables: merchant_id <chr>,
## #   first_name <chr>, balance <dbl>, date <dttm>, gender <fct>, age <dbl>,
## #   merchant_suburb <chr>, merchant_state <fct>, amount <dbl>,
## #   transaction_id <chr>, customer_id <fct>, long_merch <chr>, lat_merch <chr>,
## #   movement <fct>, day_of_week <ord>, time <chr>, hour <fct>, month <fct>
```

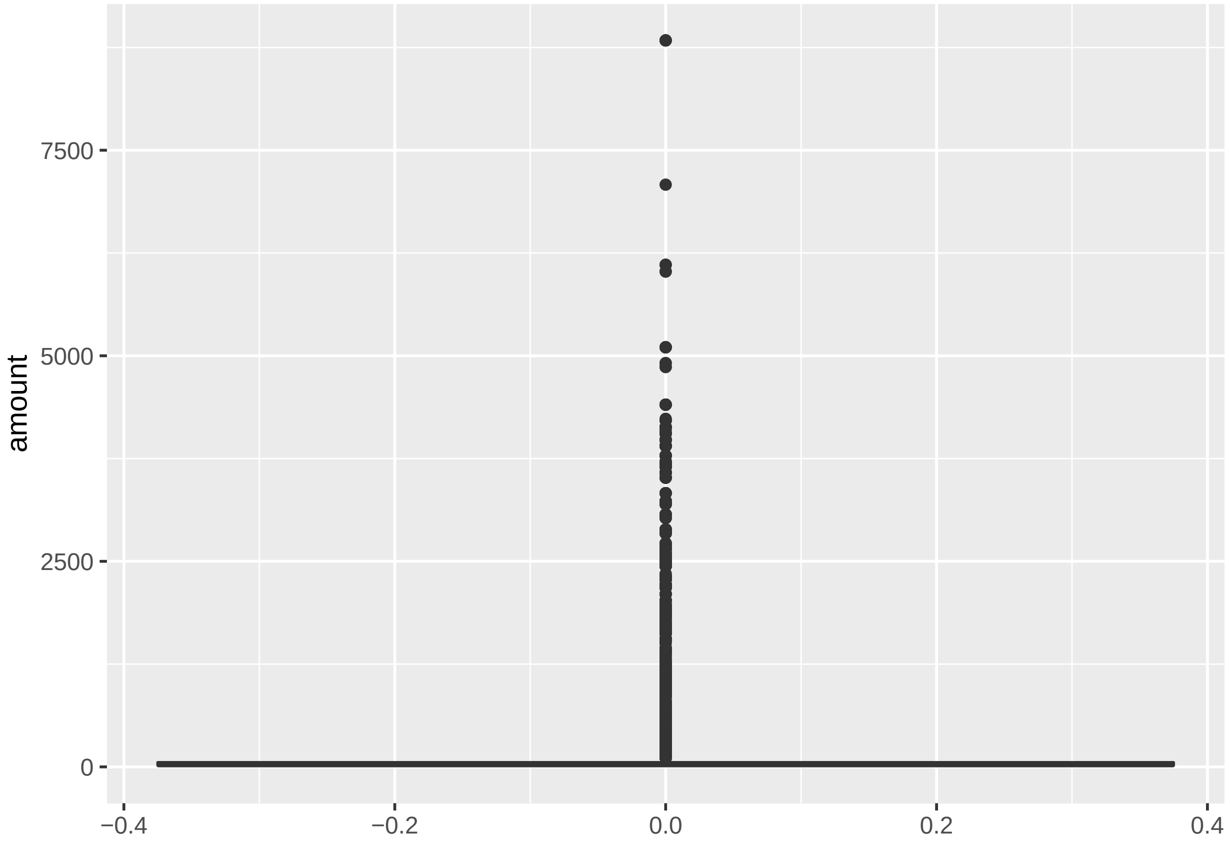
5.2 Amount

```
summary(transac$amount)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.10   16.00   29.00  187.93   53.66 8835.98
```

Mean(187.93) is really far from Median(29) because the MAx number is really high (8,835.98).

```
# boxplot to visualize outlier
ggplot(transac, aes(amount)) +
    geom_boxplot() +
    coord_flip()
```



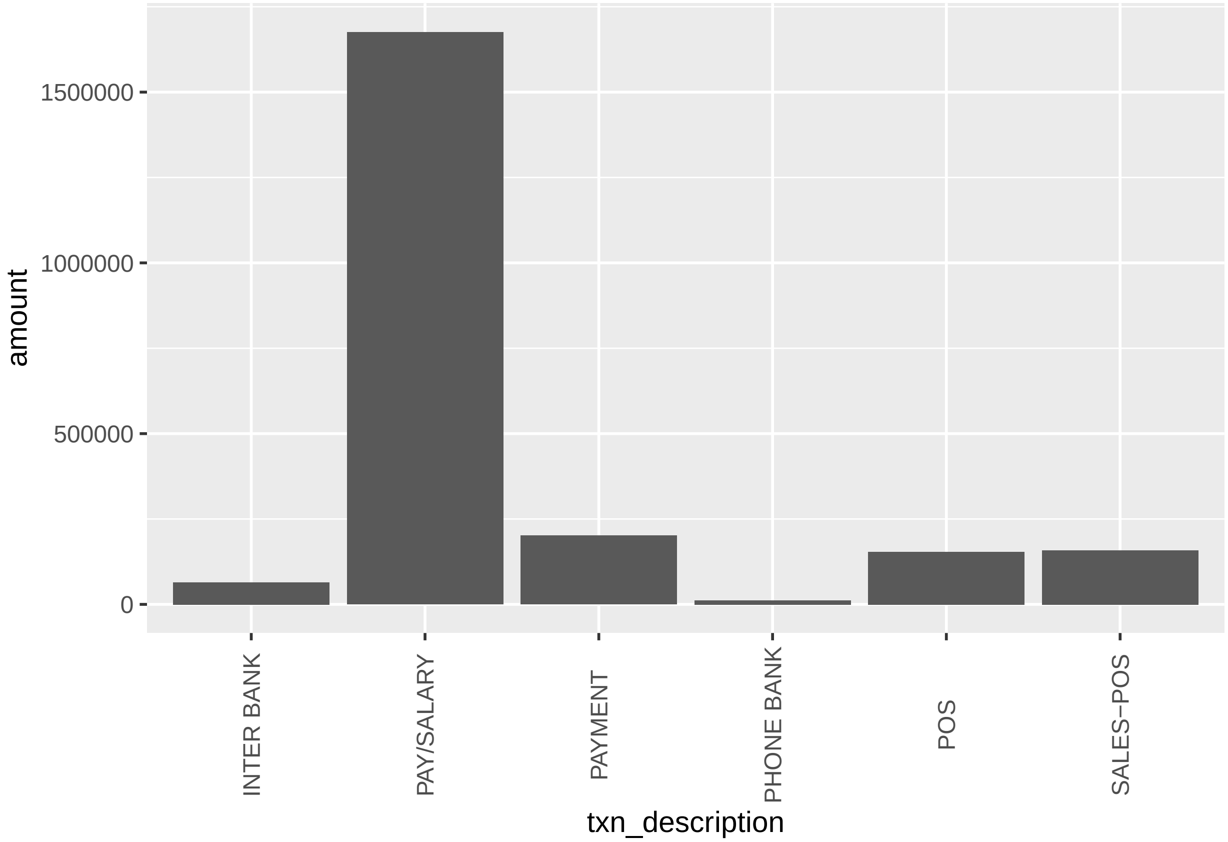
```
out_amount <- boxplot.stats(transac$amount)$out
out_amount_iden <- which(transac$amount
                         %in% c(out_amount))
transac[out_amount_iden, ] %>% arrange(desc(amount))
```

```
## # A tibble: 1,844 x 24
##    status     card_present_flag account         long_cust lat_cust txn_description
##    <chr>      <fct>             <fct>           <chr>     <chr>    <fct>
##  1 posted     <NA>              ACC-1523339231 115.83    -31.90   PAY/SALARY
##  2 posted     <NA>             ACC-1523339231 115.83    -31.90   PAY/SALARY
##  3 posted     <NA>             ACC-1523339231 115.83    -31.90   PAY/SALARY
##  4 posted     <NA>             ACC-1523339231 115.83    -31.90   PAY/SALARY
##  5 authorized 1                ACC-819621312  145.04    -37.85   POS
##  6 posted     <NA>             ACC-3100725361 145.73    -17.03   PAY/SALARY
##  7 posted     <NA>             ACC-3100725361 145.73    -17.03   PAY/SALARY
##  8 posted     <NA>             ACC-2673069055 152.99    -27.49   PAY/SALARY
##  9 posted     <NA>             ACC-2673069055 152.99    -27.49   PAY/SALARY
## 10 posted     <NA>             ACC-354106658  151.04    -33.80   PAY/SALARY
## # ... with 1,834 more rows, and 18 more variables: merchant_id <chr>,
## #   first_name <chr>, balance <dbl>, date <dttm>, gender <fct>, age <dbl>,
## #   merchant_suburb <chr>, merchant_state <fct>, amount <dbl>,
## #   transaction_id <chr>, customer_id <fct>, long_merch <chr>, lat_merch <chr>,
## #   movement <fct>, day_of_week <ord>, time <chr>, hour <fct>, month <fct>
```

We found that the high amount transactions are from PAY/SALARY which affect the range of data in this amount column.

They got the Salary through this Bank account, so this is the reason why there are a big range of number.

```
ggplot(transac, aes(txn_description, amount)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 90,
                                   vjust = 0.5))
```



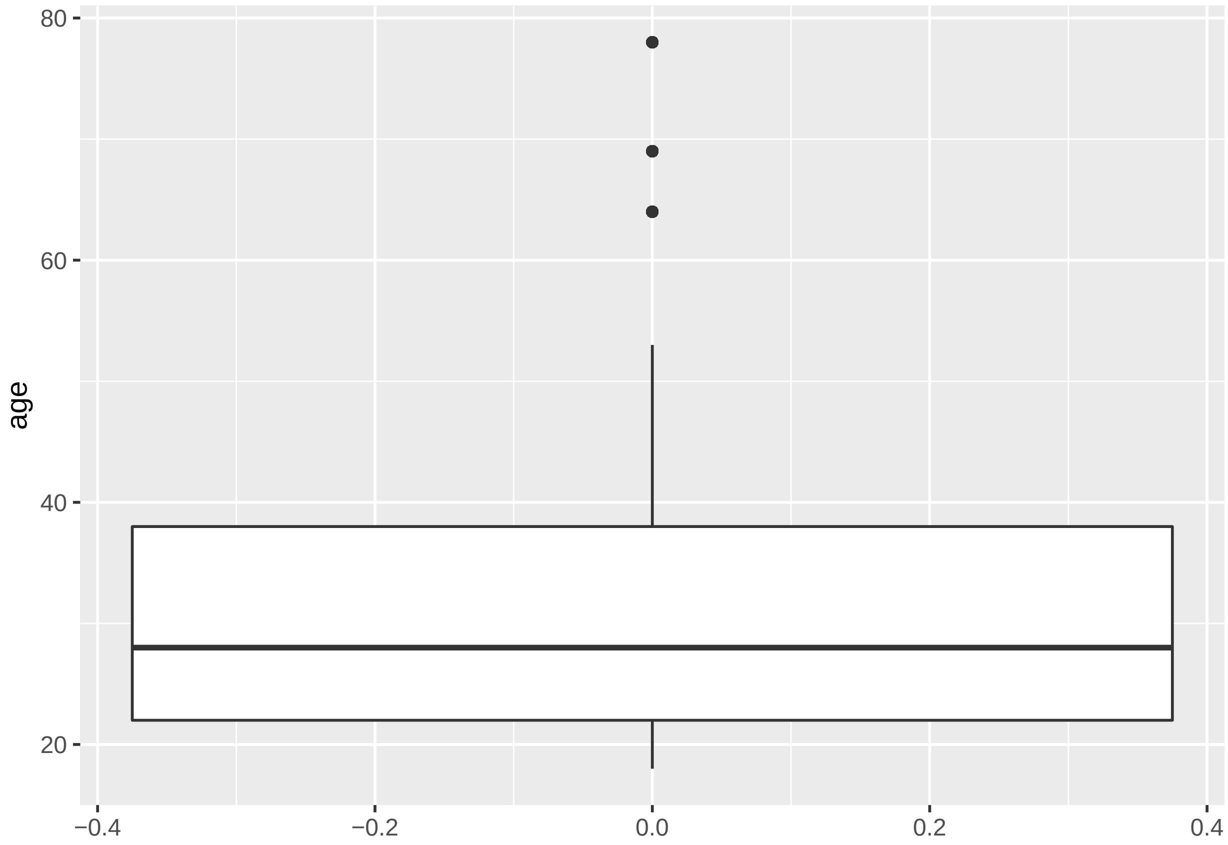
5.3 Age

```
summary(transac$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   22.00   28.00   30.58   38.00   78.00
```

An average age of customers who use ANZ Bank is 30.58.

```
ggplot(transac, aes(age)) +
    geom_boxplot() +
    coord_flip()
```

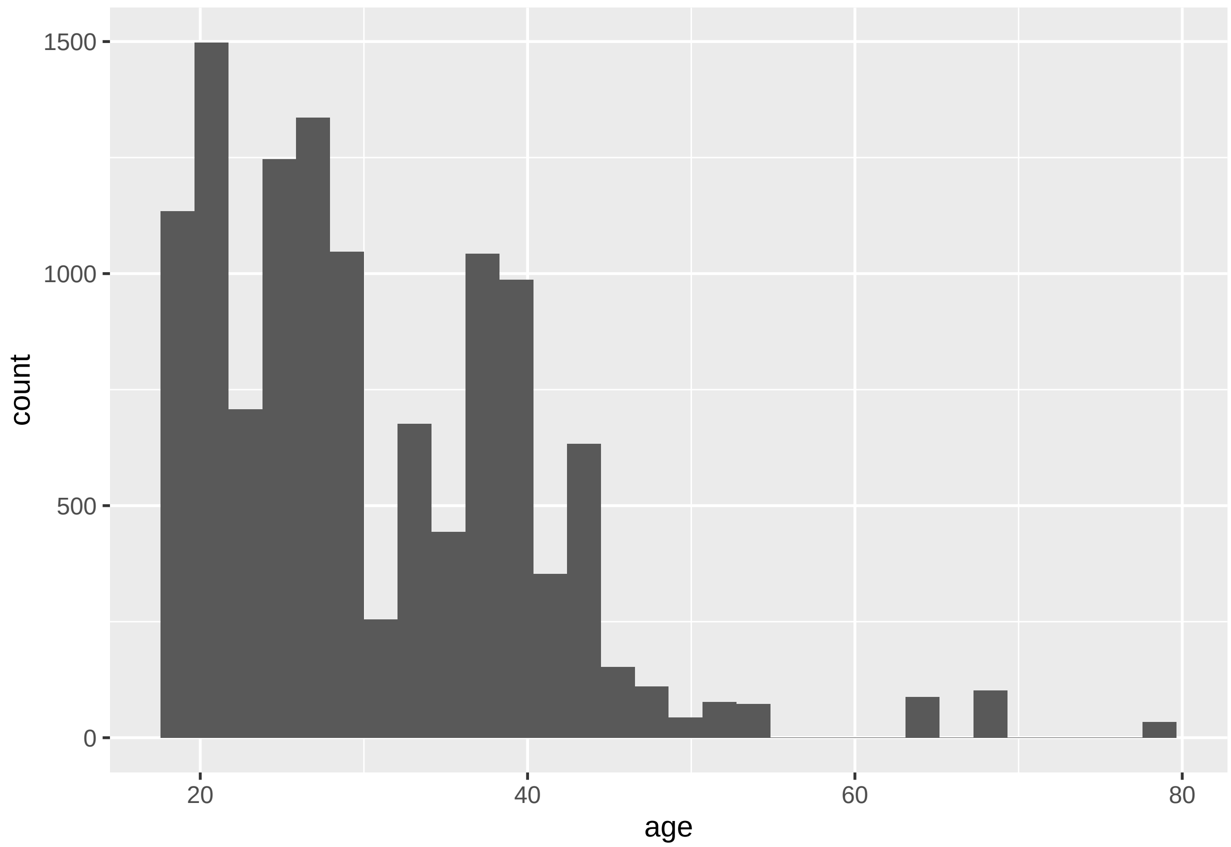


```
out_age <- boxplot.stats(transac$age)$out
out_age_iden <- which(transac$age %in% c(out_age))
transac[out_age_iden, ] %>%
  arrange(desc(age))
```

```
## # A tibble: 224 x 24
##    status     card_present_flag account        long_cust lat_cust txn_description
##    <chr>      <fct>             <fct>          <chr>     <chr>    <fct>
##  1 authorized 1                 ACC-721712940 145.09    -37.82   POS
##  2 authorized 1                 ACC-721712940 145.09    -37.82   POS
##  3 posted     <NA>              ACC-721712940 145.09    -37.82   PAY/SALARY
##  4 posted     <NA>              ACC-721712940 145.09    -37.82   PAYMENT
##  5 authorized 1                 ACC-721712940 145.09    -37.82   POS
##  6 posted     <NA>              ACC-721712940 145.09    -37.82   PAY/SALARY
##  7 posted     <NA>              ACC-721712940 145.09    -37.82   PAYMENT
##  8 posted     <NA>              ACC-721712940 145.09    -37.82   PAYMENT
##  9 posted     <NA>              ACC-721712940 145.09    -37.82   PHONE BANK
## 10 posted     <NA>              ACC-721712940 145.09    -37.82   PAYMENT
## # ... with 214 more rows, and 18 more variables: merchant_id <chr>,
## #   first_name <chr>, balance <dbl>, date <dttm>, gender <fct>, age <dbl>,
## #   merchant_suburb <chr>, merchant_state <fct>, amount <dbl>,
## #   transaction_id <chr>, customer_id <fct>, long_merch <chr>, lat_merch <chr>,
## #   movement <fct>, day_of_week <ord>, time <chr>, hour <fct>, month <fct>
```

From the box plot and table above show 3 customers who is the outliers Andrew[78y.], Tyler[69y.] and Mary[64 y.]

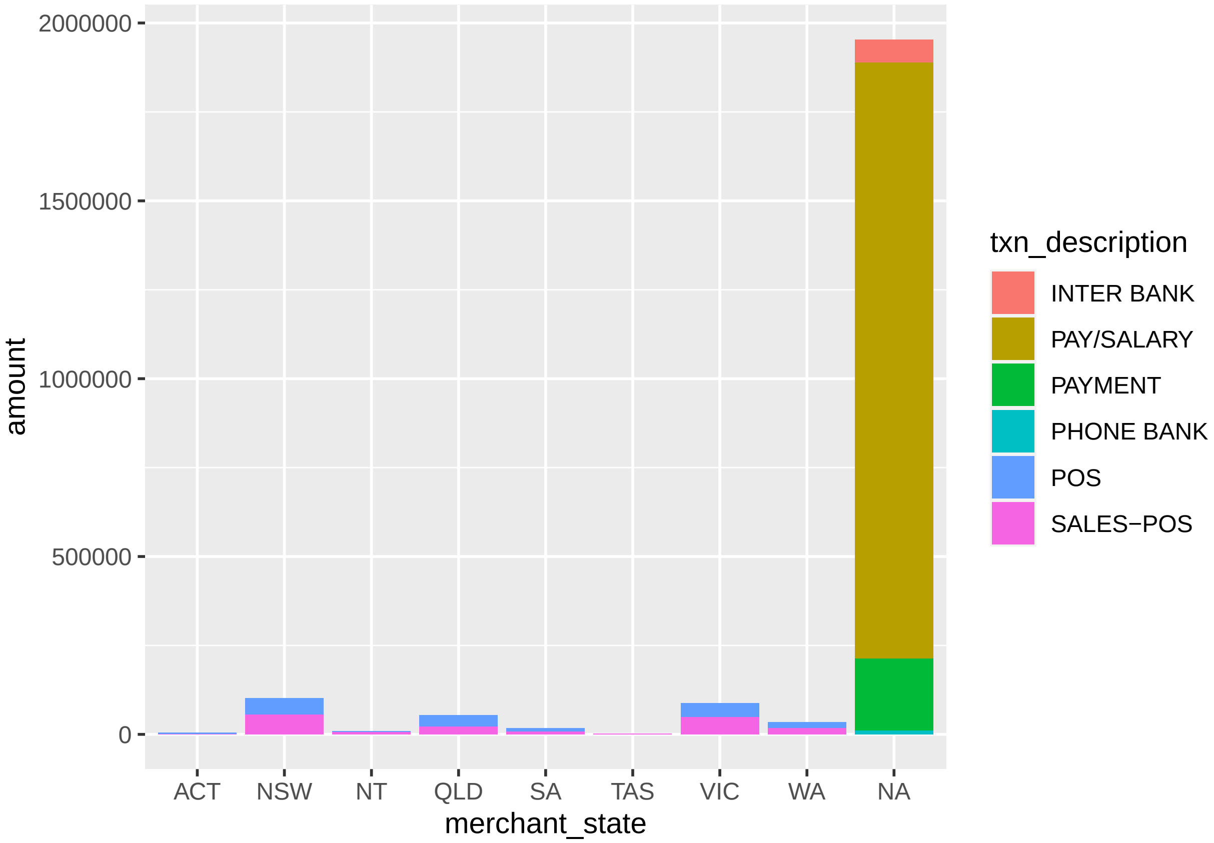
```r
ggplot(transac, aes(age)) +
    geom_histogram(bins = 30)
```



Data Analysis

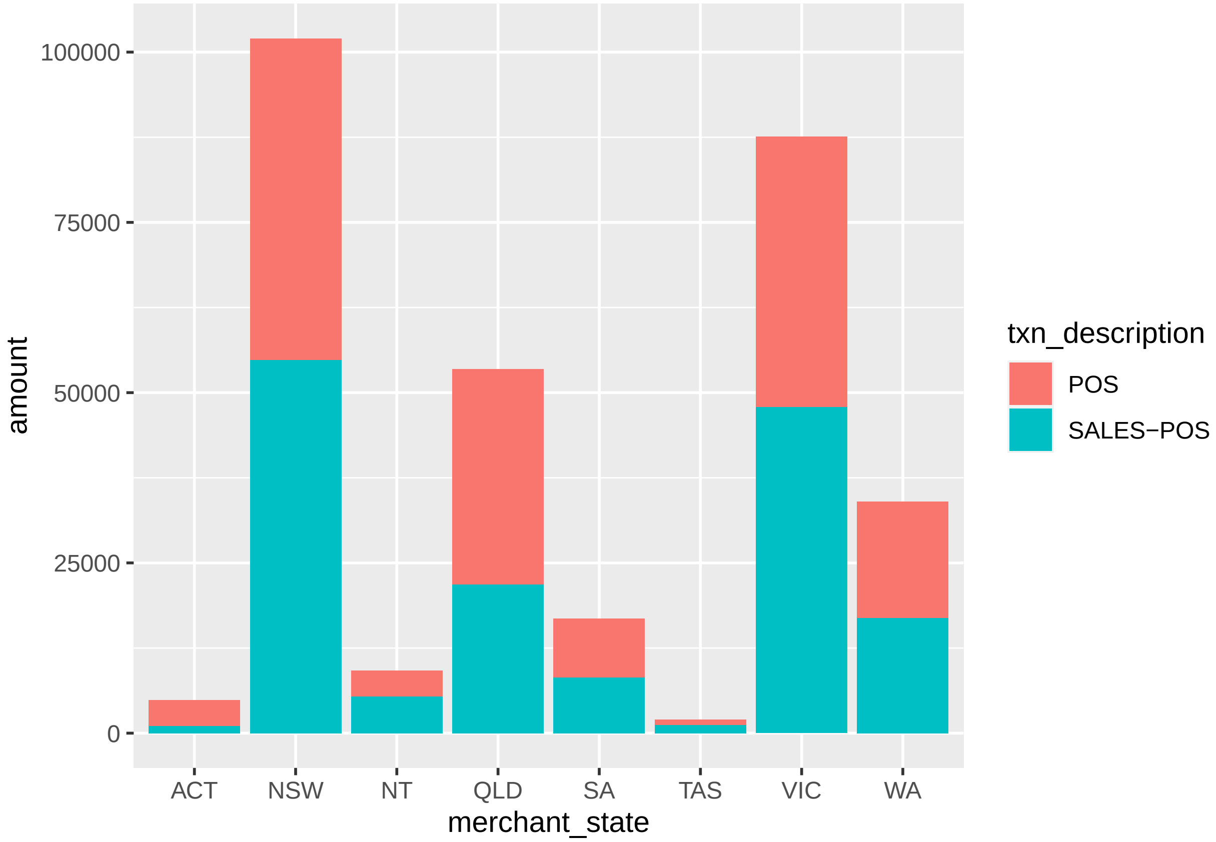
1. Amount by Merchant_States and Transaction Type

```r
ggplot(transac, aes(merchant_state, amount,
                    fill = txn_description)) +
  geom_col()
```



NA value includes many type of transaction. So, we will exclude them first and focus on states.

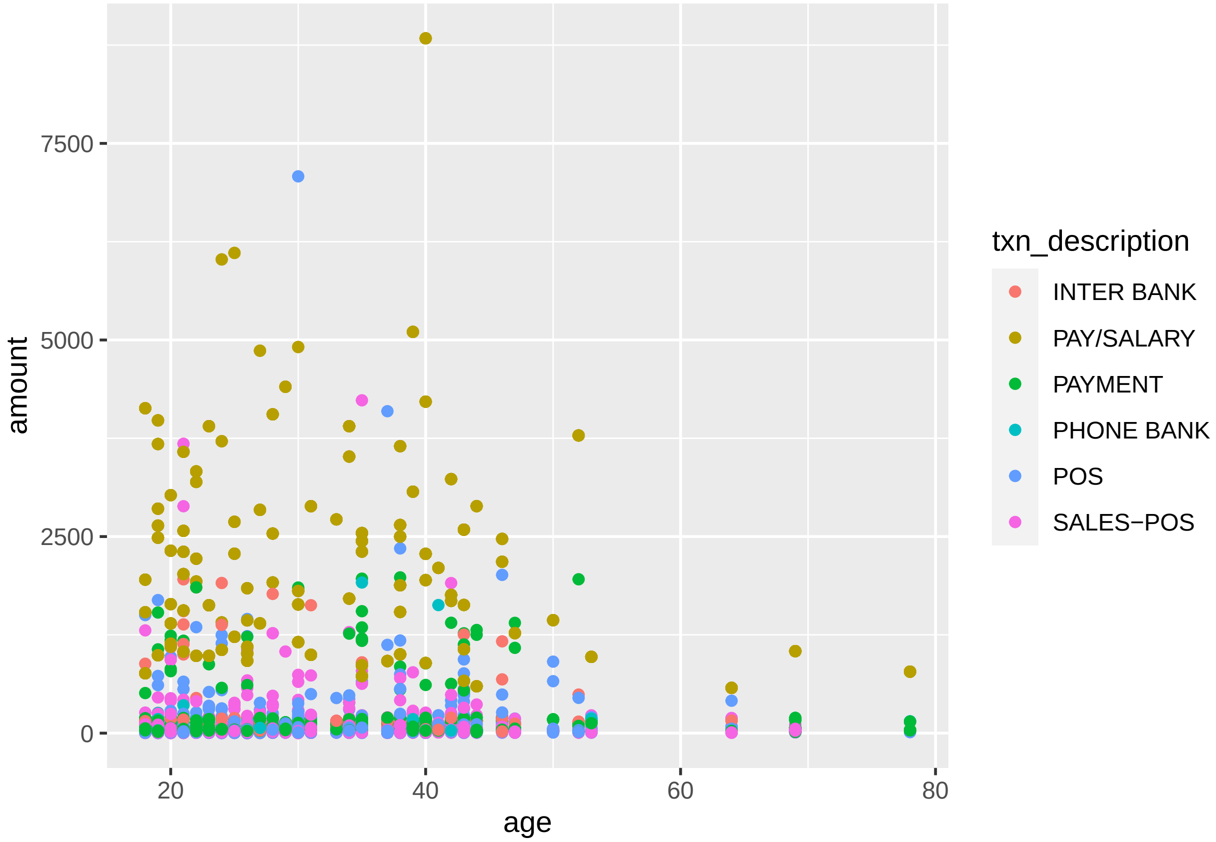
```r
# filter out NA
filter_na_state <- transac %>%
                   drop_na(merchant_state)

# Plotting Chart
ggplot(filter_na_state, aes(merchant_state, amount,
                    fill = txn_description)) +
  geom_col()
```



Top 3 Sates are NSW, VIC and QLD of POS and SALE-POS

2. Amount by Age and Transaction Type

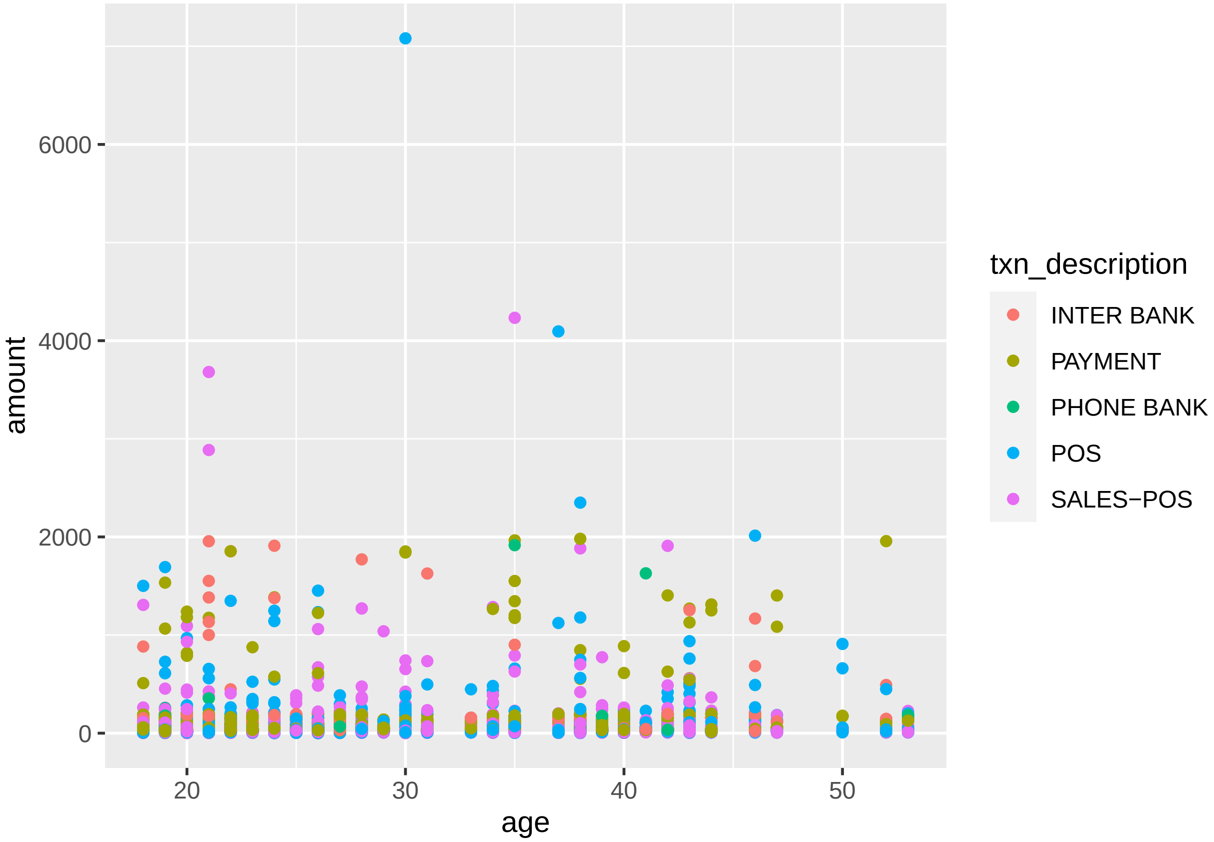
```
# Plotting Chart
ggplot(transac, aes(age, amount,
                    color = txn_description)) +
  geom_point()
```



PAY/SALARY is the large amount of transaction that far higher than other types. So, we will filter out the PAY/SALARY types and Outlier of age.

```
# filter Age outlier and Pay/Salary

filter_outAge_salary <- transac %>%
  filter(txn_description != "PAY/SALARY" & age < 60)
```

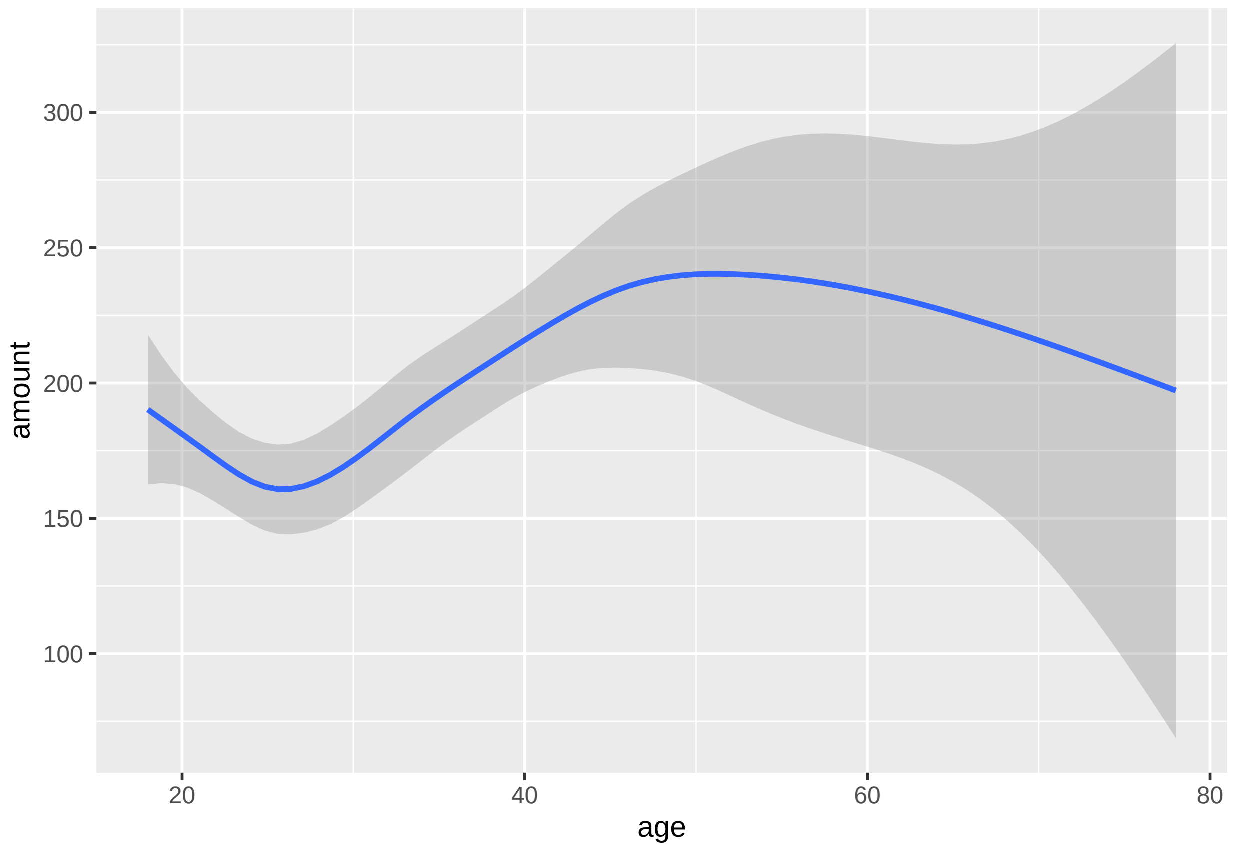
```
ggplot(filter_outAge_salary, aes(age, amount,
                    color = txn_description)) +
  geom_point()
```



Most transaction are not over 2,000 AUD.

3. Age x Amount

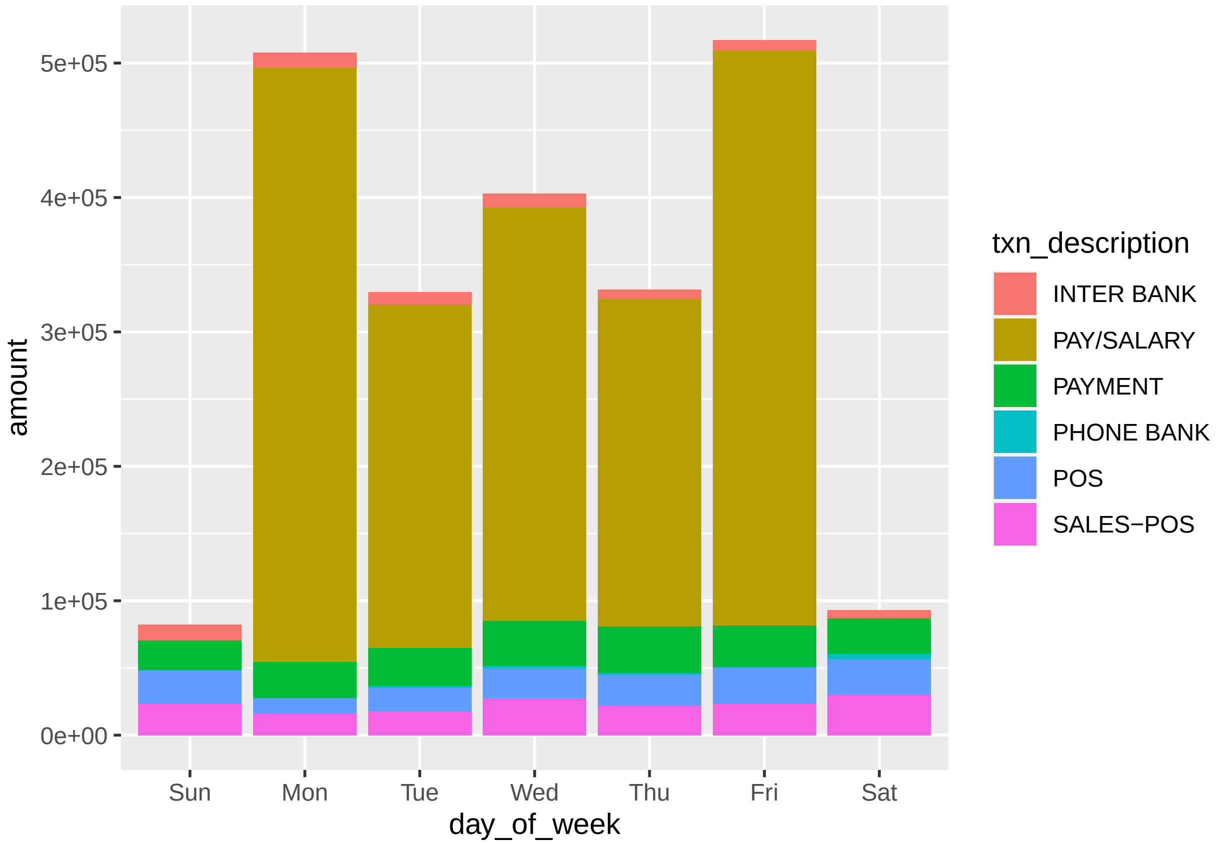
```
ggplot(transac, aes(age, amount)) +
  geom_smooth()
```



The pattern of age and amount.

4. Day of the Week and Transaction Method

```
ggplot(transac, aes(day_of_week, amount,
                    fill = txn_description)) +
  geom_col()
```



Pay/Salary proceed only weekday (Monday - Friday), No proceeding on Saturday or Sunday.

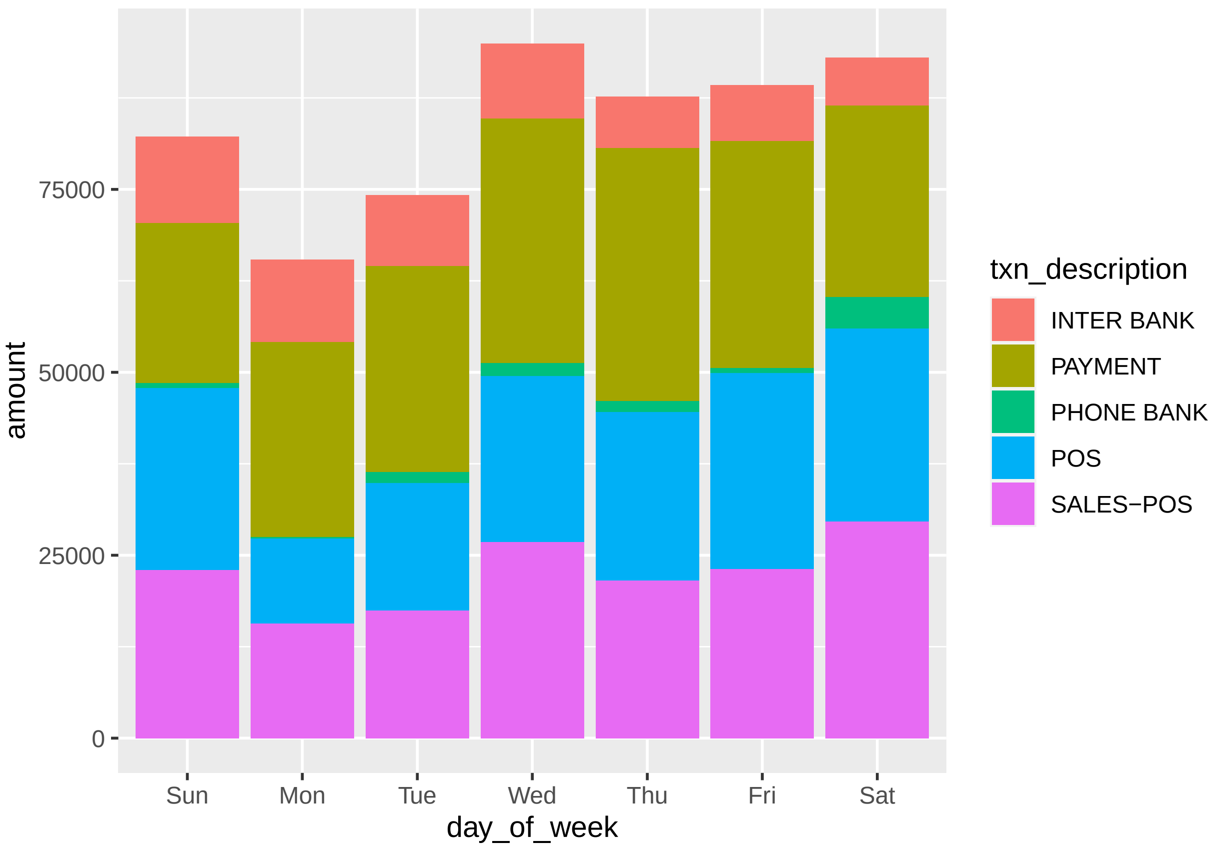
Salary Transaction is the large proportion, so we should create new variable to exclude this for analysis.

```
# Exclude Transaction type of "PAY/SALARY"
filter_out_salary <- transac %>%
  filter(txn_description != "PAY/SALARY")
```

Filter to get only PAY/SALARY Method.

```
# Filter only Transaction type of "PAY/SALARY"
filter_in_salary <- transac %>%
  filter(txn_description == "PAY/SALARY")
```

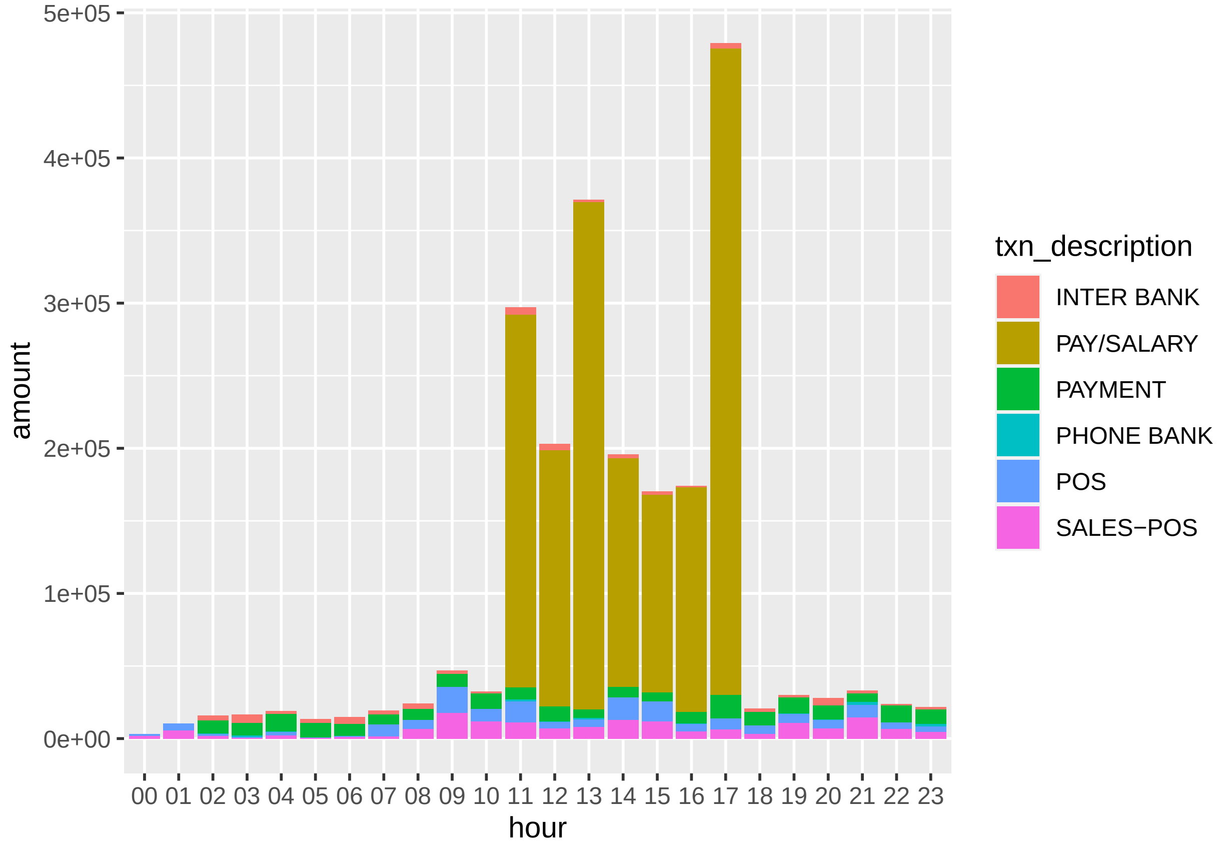
```
ggplot(filter_out_salary, aes(day_of_week, amount,
                    fill = txn_description)) +
  geom_col()
```



Excluding PAY/SALARY, There are relative proportion across weekday. POS, SALE-POS and PAYMENT are top 3 transaction types.

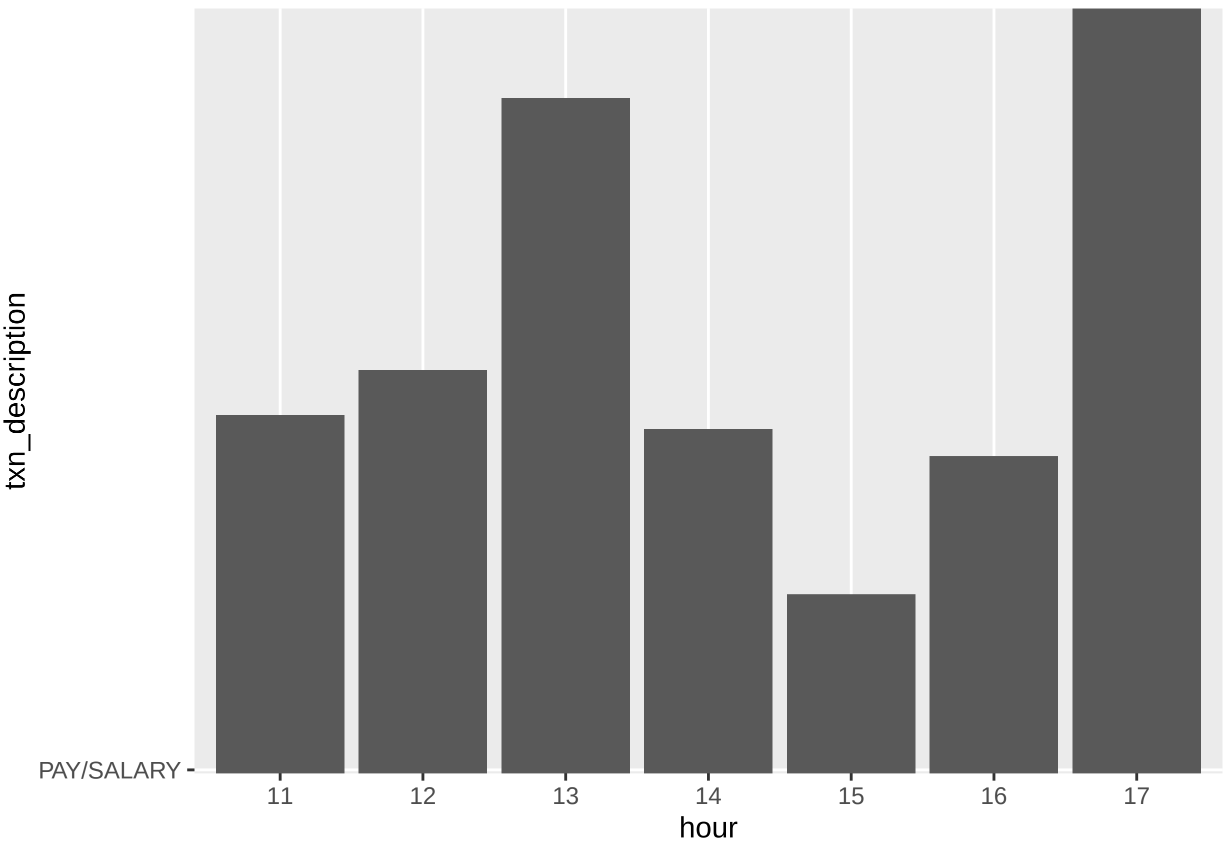
4. hour and Transaction Method

```
ggplot(transac, aes(hour, amount,
                    fill = txn_description)) +
  geom_col()
```



PAY/SALARY shows high proportion in specific time, so we will close up look at those time.

```
ggplot(filter_in_salary, aes(hour, txn_description)) +
  geom_col()
```



The Salary payment proceed between 11:00 - 17:00.

Conclusion

Data contains the transactions of 100 customers who has a Bank account with ANZ. Customer Age between 18 - 78 year old in Australia across 8 states 1609 Suburbs.

The Data collected only 3 months from 2018-08-01 to 2018-10-31, with one missing date is "2018-08-16". The number transaction is high number every Friday and Saturday, and transaction time during 09:00 - 10:00 is the peak time of Transaction across the states (more than 300+ transactions every months).

There are 6 Transaction types are POS, SALE_POS, PAY/SALARY, PAYMENT, INTER BANK, PHONE BANK. There are a large quantity of POS (3,783) and SALE-POS(3,934) transactions with relative equally. The top 3 states of POS, and SALE-POS transaction are NSW, VIC and QLD.

However, the large amount of transactions come from PAY/SALARY transaction types which customers receives the salary through the ANZ Bank account. We clearly see that this type of transactions only proceed on weekday between 11:00 - 17:00.

Save and Export data from Rstudio

```
write.csv(transac, "ANZ_clean_data.csv")
```