library(readxl)  
library(tidyverse)

## -- Attaching packages -------------------------------------------------------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0 v purrr 0.3.4  
## v tibble 3.0.1 v dplyr 0.8.5  
## v tidyr 1.0.3 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

## -- Conflicts ----------------------------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(lubridate)

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:dplyr':  
##   
## intersect, setdiff, union

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

library(funModeling)

## Loading required package: Hmisc

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':  
##   
## src, summarize

## The following objects are masked from 'package:base':  
##   
## format.pval, units

## funModeling v.1.9.3 :)  
## Examples and tutorials at livebook.datascienceheroes.com  
## / Now in Spanish: librovivodecienciadedatos.ai

library(sf)

## Linking to GEOS 3.8.0, GDAL 3.0.4, PROJ 6.3.1

library(rnaturalearth)  
library(rnaturalearthdata)

ONCE THE LIBRARIES ARE LOADED, OUR NEXT STEP WILL BE LOADING THE DATASET AND TAKING A LOOK AT THE DATA

df = read\_xlsx("ANZ synthesised transaction dataset.xlsx", col\_names = TRUE)

## Warning in read\_fun(path = enc2native(normalizePath(path)), sheet\_i = sheet, :  
## Expecting numeric in C3052 / R3052C3: got 'THE DISCOUNT CHEMIST GROUP'

## Warning in read\_fun(path = enc2native(normalizePath(path)), sheet\_i = sheet, :  
## Expecting numeric in C4360 / R4360C3: got 'LAND WATER & PLANNING East Melbourne'

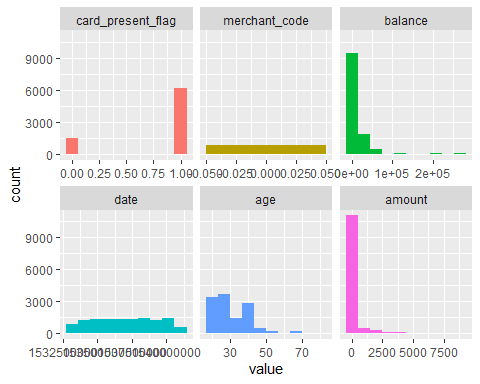
df\_status(df)

## variable q\_zeros p\_zeros q\_na p\_na q\_inf p\_inf type  
## 1 status 0 0.00 0 0.00 0 0 character  
## 2 card\_present\_flag 1523 12.65 4326 35.92 0 0 numeric  
## 3 bpay\_biller\_code 883 7.33 11160 92.67 0 0 numeric  
## 4 account 0 0.00 0 0.00 0 0 character  
## 5 currency 0 0.00 0 0.00 0 0 character  
## 6 long\_lat 0 0.00 0 0.00 0 0 character  
## 7 txn\_description 0 0.00 0 0.00 0 0 character  
## 8 merchant\_id 0 0.00 4326 35.92 0 0 character  
## 9 merchant\_code 883 7.33 11160 92.67 0 0 numeric  
## 10 first\_name 0 0.00 0 0.00 0 0 character  
## 11 balance 0 0.00 0 0.00 0 0 numeric  
## 12 date 0 0.00 0 0.00 0 0 POSIXct/POSIXt  
## 13 gender 0 0.00 0 0.00 0 0 character  
## 14 age 0 0.00 0 0.00 0 0 numeric  
## 15 merchant\_suburb 0 0.00 4326 35.92 0 0 character  
## 16 merchant\_state 0 0.00 4326 35.92 0 0 character  
## 17 extraction 0 0.00 0 0.00 0 0 character  
## 18 amount 0 0.00 0 0.00 0 0 numeric  
## 19 transaction\_id 0 0.00 0 0.00 0 0 character  
## 20 country 0 0.00 0 0.00 0 0 character  
## 21 customer\_id 0 0.00 0 0.00 0 0 character  
## 22 merchant\_long\_lat 0 0.00 4326 35.92 0 0 character  
## 23 movement 0 0.00 0 0.00 0 0 character  
## unique  
## 1 2  
## 2 2  
## 3 1  
## 4 100  
## 5 1  
## 6 100  
## 7 6  
## 8 5725  
## 9 1  
## 10 80  
## 11 12006  
## 12 91  
## 13 2  
## 14 33  
## 15 1609  
## 16 8  
## 17 9442  
## 18 4457  
## 19 12043  
## 20 1  
## 21 100  
## 22 2703  
## 23 2

df$bpay\_biller\_code = as.character(df$bpay\_biller\_code)

plot\_num(df)

## Warning: attributes are not identical across measure variables; they will be  
## dropped

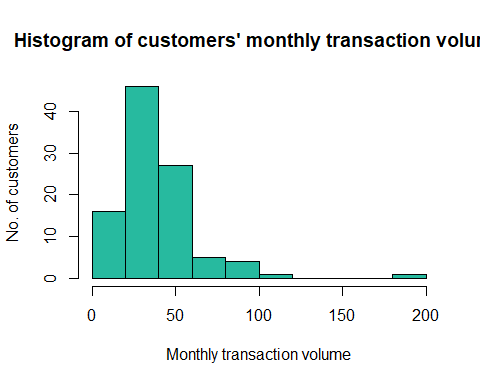


EXTRACTING INFORMATION FROM THE DATES (using libridate library)

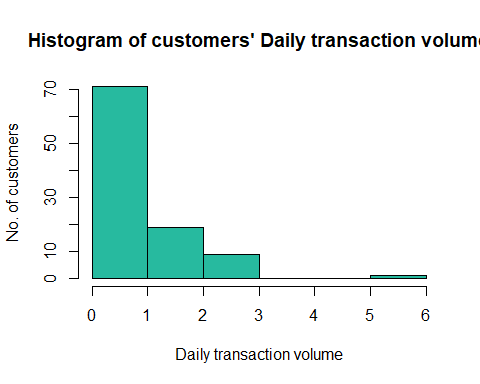
#EXTRACTING WEEKS AND HOURS FROM DATE AND EXTRACTION RESPECTIVELY  
df$weekday = weekdays(df$date)  
df$hour = as.character(df$extraction)  
df$hour = substr(df$hour,12,19)  
df$hour = hour(as.POSIXct(df$hour, format = "%H:%M:%S"))

WE NOW LOOK AT THE TRANSACTIONAL TRENDS

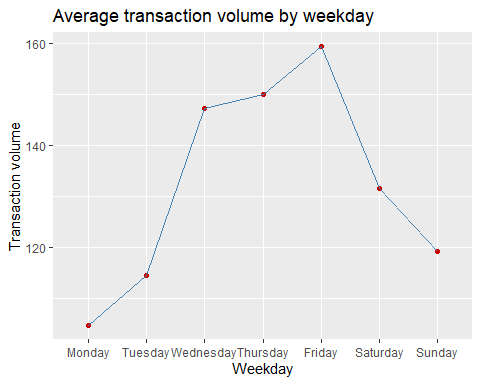
#AVERAGE MONTHLY TRANSACTION DISTRIBUTION  
df\_mon <- df %>% group\_by(customer\_id) %>% summarise(mon\_avg\_vol = round(n()/3,0))  
  
hist(df\_mon$mon\_avg\_vol,  
 xlab= 'Monthly transaction volume', ylab='No. of customers', main = "Histogram of customers' monthly transaction volume",col = "#27ba9f")



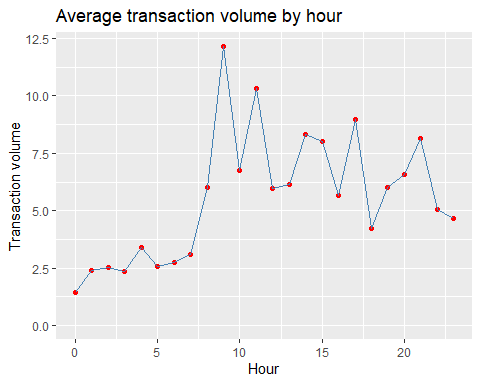
#AVERAGE DAILY TRANSACTION DISTRUBUTION  
df\_day <- df %>% group\_by(customer\_id) %>% summarise(da\_avg\_vol = round(n()/91,0))  
  
hist(df\_day$da\_avg\_vol, xlab= 'Daily transaction volume', ylab='No. of customers', main = "Histogram of customers' Daily transaction volume",col = "#27ba9f")



#AVERAGE TRANSACTION BY WEEKDAY  
df\_wk <- df%>% select(date, weekday) %>%  
 group\_by(date, weekday) %>%  
 summarise(daily\_avg\_vol =n()) %>%  
 group\_by(weekday) %>%  
 summarise(avg\_vol=mean(daily\_avg\_vol,na.rm=TRUE ))  
df\_wk <- df\_wk %>% mutate(weekday = factor(weekday, levels = c("Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"))) %>% arrange(weekday)  
   
ggplot(df\_wk,aes(x=weekday, y=avg\_vol)) +geom\_point(colour = "#CC0000")+geom\_line(aes(group = 1), colour = "Steelblue") +   
 ggtitle('Average transaction volume by weekday') +  
 labs(x='Weekday',y='Transaction volume')



#AVERAGE TRANSACTION PLOTTED BY HOURS  
df\_hr <- df %>%  
 select(date,hour) %>%  
 group\_by(date,hour) %>%  
 summarise(trans\_vol=n()) %>%  
 group\_by(hour) %>%  
 summarise(trans\_vol\_per\_hr = mean(trans\_vol,na.rm=TRUE))  
  
ggplot(df\_hr,aes(x=hour,y=trans\_vol\_per\_hr))+geom\_point(colour ="Red")+geom\_line(aes(group = 1), colour ="Steelblue")+ ggtitle('Average transaction volume by hour') + labs(x='Hour',y='Transaction volume') + expand\_limits( y = 0)

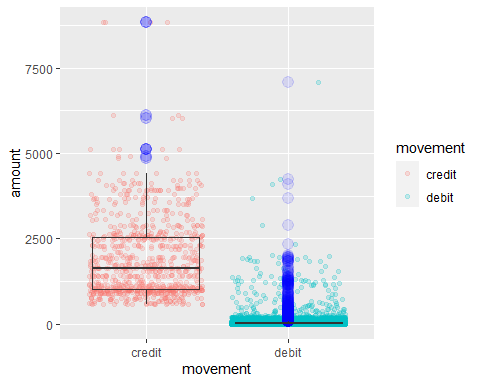


PLOTTING OUTLIERS IN THE TRANSACTIONAL DATA

tapply(df$amount, df$movement, summary)

## $credit  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 576 1014 1626 1899 2539 8836   
##   
## $debit  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.10 15.19 26.93 52.57 45.00 7081.09

ggplot(data = df, aes(x=movement, y=amount)) +  
 geom\_point(aes(color=movement), alpha=0.2, position ="jitter") +  
 geom\_boxplot(outlier.size=4, outlier.colour='blue', alpha=0.1)



WE CAN GET THE LONGITUDE AND LATITUDE FOR THE CUSTOMERS AND THE MERCHANTS

#Extracting Latitude and Longitudes  
#1. Customer location  
cust\_loc = data.frame(df$long\_lat)  
cust\_loc = data.frame(separate(cust\_loc, col = df.long\_lat, into = c("cust\_long","cust\_lat"), sep = " "))  
  
#2. Merchant location  
merc\_loc = data.frame(df$merchant\_long\_lat)  
merc\_loc = data.frame(separate(merc\_loc, col = df.merchant\_long\_lat, into = c("merc\_long","merc\_lat"), sep = " "))  
  
df\_loc <- cbind(df$customer\_id,cust\_loc,merc\_loc)  
colnames(df\_loc)[which(names(df\_loc)=="df$customer\_id")] <- "customer\_id"  
  
df\_loc$cust\_long = as.numeric(df\_loc$cust\_long)  
df\_loc$cust\_lat = as.numeric(df\_loc$cust\_lat)  
df\_loc$merc\_lat = as.numeric(df\_loc$merc\_lat)  
df\_loc$merc\_long = as.numeric(df\_loc$merc\_long)

#PLOTTING THE CUSTOMER AND THE MERCHANT  
merc\_map <- function (id ){  
   
 l = subset (df\_loc[,c("customer\_id","merc\_long","merc\_lat")], customer\_id == id)  
 l <- l[c("merc\_long","merc\_lat")]  
   
 k = subset (df\_loc[,c("customer\_id","cust\_long","cust\_lat")], customer\_id == id)  
 k <- unique(k[c("cust\_long","cust\_lat")])  
   
 ggplot(data = ne\_countries(scale = "medium", returnclass = "sf")) +  
 geom\_sf(fill = "#5a5a5a") +  
 coord\_sf(xlim = c(113,154), ylim = c(-44,-10), expand = FALSE) +  
 geom\_point(data = l, aes(x = merc\_long, y = merc\_lat),na.rm=TRUE, size = 1, shape = 4, colour ="#08e8de" ) +  
 geom\_point(data = k, aes(x = cust\_long, y = cust\_lat),na.rm=TRUE, size = 2, shape = 24, fill = "#f13f09") + xlab("Longitude") + ylab("Latitude") + scale\_color\_manual(values = c("steelblue", "red"),labels = c("Merchant", "Customer"))  
}

merc\_map(id ='CUS-1669695324') #EXAMPLE

