Data exploration and analysis report

Sílvia Maia and Pedro Antunes

30/06/2021

Data exploration and analysis report

Introduction

Within the Fraud Detection course unit we were given a real world online trasactions dataset to first analyze, and later try to predict if the transactions are fraudlent or not. This report will report the analysis part.

Dataset

We are given two datasets, a train.csv and a test.csv, in this report we will focus on the train dataset since the test dataset is the same, just does not have the *isFraud* column. The dataset describes 150000 transactions and has information about the transaction itself and its customer.

- TransactionDT: is the timedelta from a time reference.
- TransactionAmt: is the transaction cost in USD.
- ProductCD: is the product code for each transaction.
- card1-card6: Payment card information, such as card type, category, bank, country…
- addr: address.
- dist: distance.
- P_ nd (R_) emaildomain: Seller and customer email domain.
- C1-C14: Relevant aspects counters, such as how many emails were found associated with a payment card. The real meaning is masked.
- D1-D15: timedelta, such as days between previous transacitons.
- M1-M9: Combinations, such as card names and addresses.
- Vxxx: Engineering features, includes counters, ranking and other entity relationship.
- id_01-id_38: information about the customer's identity, properly masked for privacy reasons.
- DeviceType: the type of device used to make the transaction.
- DeviceInfo: deeper details about the device.

Used Libraries

```
library("dplyr")
library("lubridate")
library("ggplot2")
library("tidyr")
```

Load dataset

For faster analysis, we only use 10% of the dataset.

```
set.seed(1223)

library(data.table)

train <- fread("train.csv", sep=",", header=TRUE)

ids <- sample(1:nrow(train), 0.1*nrow(train))
# select a sample (10%) of the entire data set to allow a faster exploration of the data
ds <- train[ids,] # the working subset
```

Our approach

We first looked at the dataset and then analyzed all the columns one by one to try to understand its impact.

Summary and NAs

With a little eye survey on *ds*, we noticed that many fields have no value or are NA, so we replaced all the unvalued fields with NA to make it clear and uniform.

```
ds[ds==''] <- NA
data_na <- (colSums(is.na(ds)) / nrow(ds)) * 100
d <- data_na[data_na <5]
ds <- subset(ds, select=unlist((attributes(d))))
```

We are removing columns with percentage of NAs > 5%. The next columns with less percentage of NAs were addr1 and addr2 with about 10%, which is the double of our threshold and more than the triple of our target.

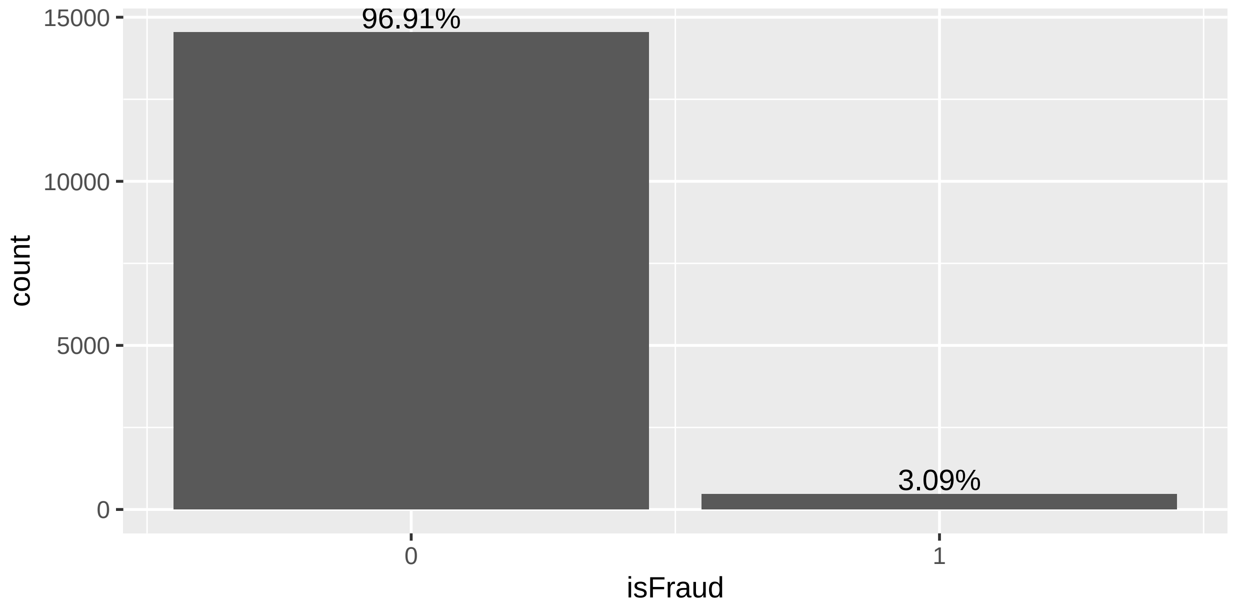
```
summary(ds)
```

We can look at which columns are nominal and which are numerals, and later get an idea of how the data is spread.

Target: *isFraud*

This is the feature that we will foresee in the second part of the work.

```
ggplot(ds, aes(x=isFraud)) + geom_bar() +
  scale_x_continuous(name="isFraud", breaks=c(0,1)) +
  geom_text(aes(label=after_stat(sprintf("%.2f%%", prop*100))), stat='count', vjust=-0.2)
```



We found that only a small percentage (about 3.09%) of transactions are fraudulent, which makes it a very unbalanced distribution.

card1-card6

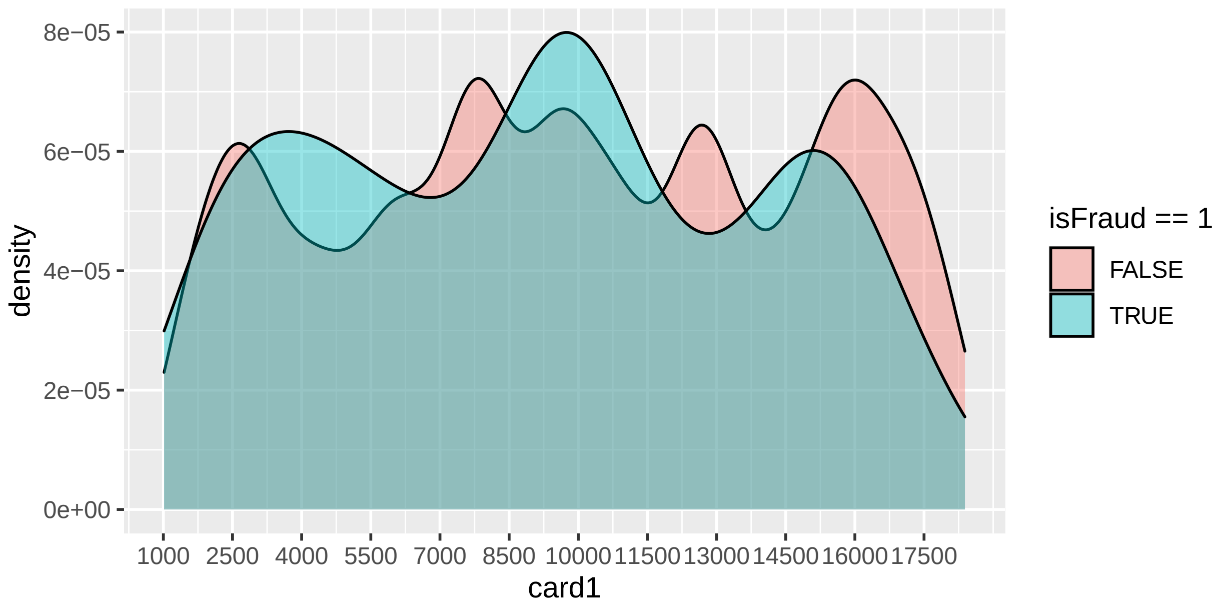
```
select(ds, c(6:11)) %>% summary()
```

```
##      card1           card2           card3          card4          
##  Min.   : 1015   Min.   :100.0   Min.   :100.0   Length:15000      
##  1st Qu.: 6019   1st Qu.:215.0   1st Qu.:150.0   Class :character  
##  Median : 9803   Median :375.0   Median :150.0   Mode  :character  
##  Mean   : 9927   Mean   :367.6   Mean   :153.2                     
##  3rd Qu.:14290   3rd Qu.:514.0   3rd Qu.:150.0                     
##  Max.   :18387   Max.   :600.0   Max.   :231.0                     
##                  NA's   :243                                       
##      card5          card6          
##  Min.   :100.0   Length:15000      
##  1st Qu.:166.0   Class :character  
##  Median :226.0   Mode  :character  
##  Mean   :200.6                     
##  3rd Qu.:226.0                     
##  Max.   :237.0                     
##  NA's   :89                        
```

With this, we saw that *card4* and *card6* are strings and there is not NA's on card1. *card2, card3 and card5* have a small percentage of NA's. Further on, we realize that *card1* range is much higher than *card2, card3 and card5.*

```
ggplot(ds) + aes(x=card1, fill=isFraud==1) + geom_density(alpha=0.4) +
  scale_x_continuous(breaks=seq(1000, 18387, 1500))
```

card1



Through this density plot, we observed that *card1* values are well scattered. However, we can evaluate interesting higher densities cases that have fraudulent percentages according to our target (*isFraud*, 3.09%).

```
d <- filter(ds, card1>=3000 & card1<=6500) %>% select(isFraud, c(6:11))
(nrow(filter(d, isFraud==1)) / nrow(d) ) * 100
```

```
## [1] 4.027386
```

4.02% cases with 3000<=card1<=6500 are fraud.

```
d <- filter(ds, card1>=8500 & card1<=12000) %>% select(isFraud, c(6:11))
(nrow(filter(d, isFraud==1)) / nrow(d) ) * 100
```

```
## [1] 4.256713
```

4.25% cases with 8500<=card1<=12000 are fraud.

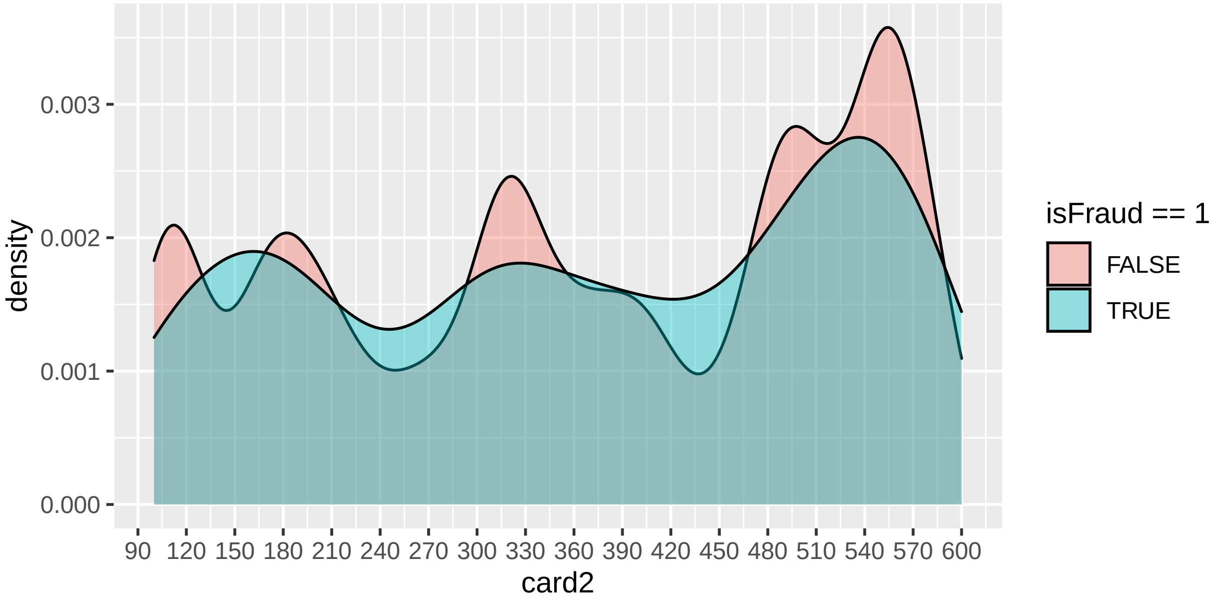
```
d <- filter(ds, card1>=13000 & card1<=14500) %>% select(isFraud, c(6:11))
(nrow(filter(d, isFraud==1)) / nrow(d) ) * 100
```

```
## [1] 2.915767
```

2.91% cases with 13000<=card1<=14500 are fraud.

```
ggplot(ds) + aes(x=card2, fill=isFraud==1) + geom_density(alpha=0.4, na.rm=TRUE) +
  scale_x_continuous(breaks=seq(0, 600, 30))
```

card2



Through this density plot, we observed that *card2* values are well scattered with a top at 555. At the value 555 we get 1.73% of fraudulent transactions (lower than our target).

```
d <- filter(ds, is.na(card2)) %>% select(isFraud, c(6:11))
(nrow(filter(d, isFraud==1)) / nrow(d) ) * 100
```

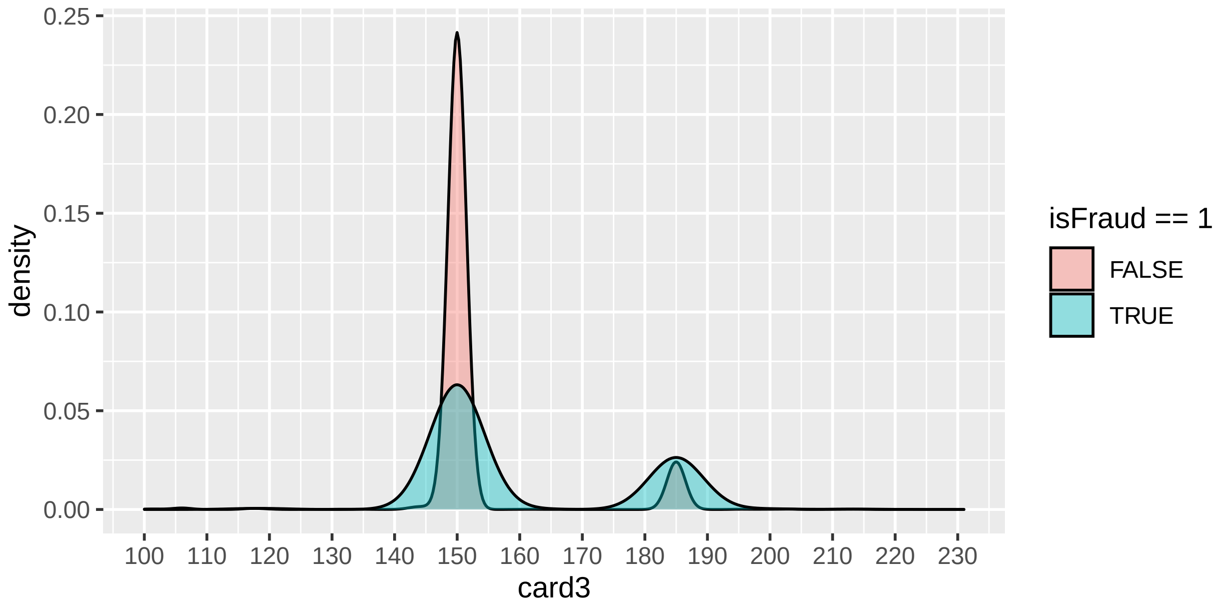
```
## [1] 7.407407
```

On *card2*, we have relevant 7.4% of NA's values that are fraud. So let's replace NA with a value outside the range of *card2*

```
ds$card2[is.na(ds$card2)] <- 0
```

```
ggplot(ds) + aes(x=card3, fill=isFraud==1) + geom_density(alpha=0.4, na.rm=TRUE) +
  scale_x_continuous(breaks=seq(100, 231, 10))
```

card3



Values on *card3* are scattered badly.

```
d <- filter(ds, card3==150) %>% select(isFraud, c(6:11))
nrow(filter(d, isFraud==1)) / nrow(d) *100
```

```
## [1] 2.419902
```

At the value 150 (that are 88.4% of card3 cases), we get 2.4% of fraudulent transactions.

```
d <- filter(ds, card3==185) %>% select(isFraud, c(6:11))
nrow(filter(d, isFraud==1)) / nrow(d) *100
```

```
## [1] 9.331476
```

At the value 185 (that are 9.57% of card3 cases), we get 9.3% of fraudulent transactions. Which is to say that all these cases are fraudulent.

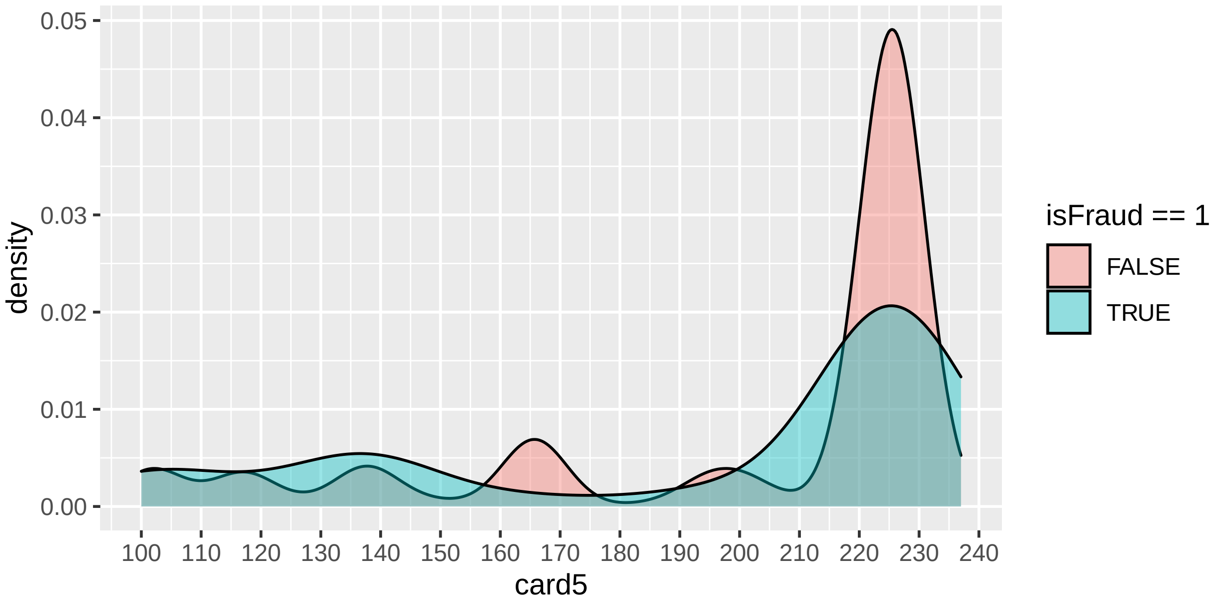
```
## [1] NaN
```

On *card3*, we not have NA's values that are fraud. So again, let's replace NA with a value outside the range of *card3*

```
ds$card2[is.na(ds$card2)] <- 0
```

```

ggplot(ds) + aes(x=card5, fill=isFraud==1) + geom_density(alpha=0.4, na.rm=TRUE) +
  scale_x_continuous(breaks=seq(100, 240, 10))
```

card5



The density of fraudulent transaction is higher at values less than or equal to 158.

```
d <- filter(ds, card5<=158) %>% select(isFraud, c(6:11))
(nrow(filter(d, isFraud==1)) / nrow(d) ) * 100
```

```
## [1] 5.265273
```

At this values, we get 5.26% of fraudulent transactions.

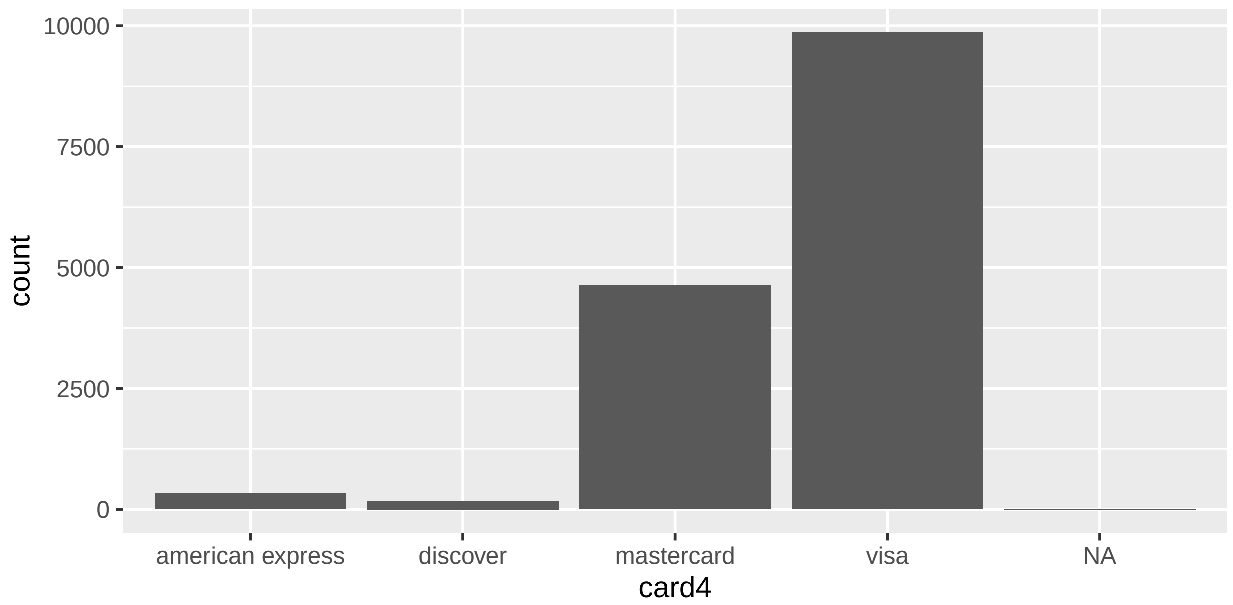
```
## [1] 3.370787
```

On *card5*, we have relevant 3.37% of NA's values that are fraud transactions. So let's replace NA with a value outside the range of *card5*

```
ds$card5[is.na(ds$card5)] <- 0
```

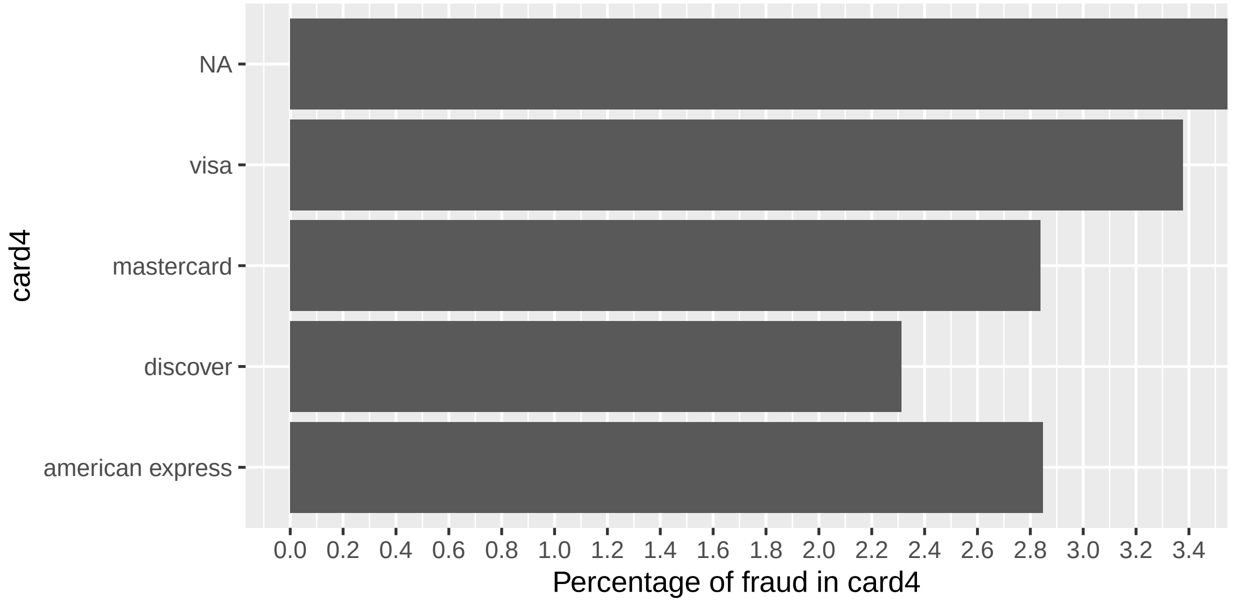
```

ggplot(ds) + aes(x=card4) + geom_bar()
```

card4



On *card4* we have different values like *american express, discover, mastercard, visa and one without name*. We analyze fraud transaction percentage of each nominal on *card4*.

```
d <- group_by(ds, card4) %>%
  mutate(fraud=sum(isFraud), notfraud=sum(isFraud==0), n=nrow(ds)) %>%
  select(card4, fraud, notfraud, n)
d <- unique(d)
ggplot(d) + aes(x=card4, y=(fraud/notfraud)*100) + coord_flip() + geom_col() +
  scale_y_continuous(name="Percentage of fraud in card4", breaks = seq(0,5,0.2))
```

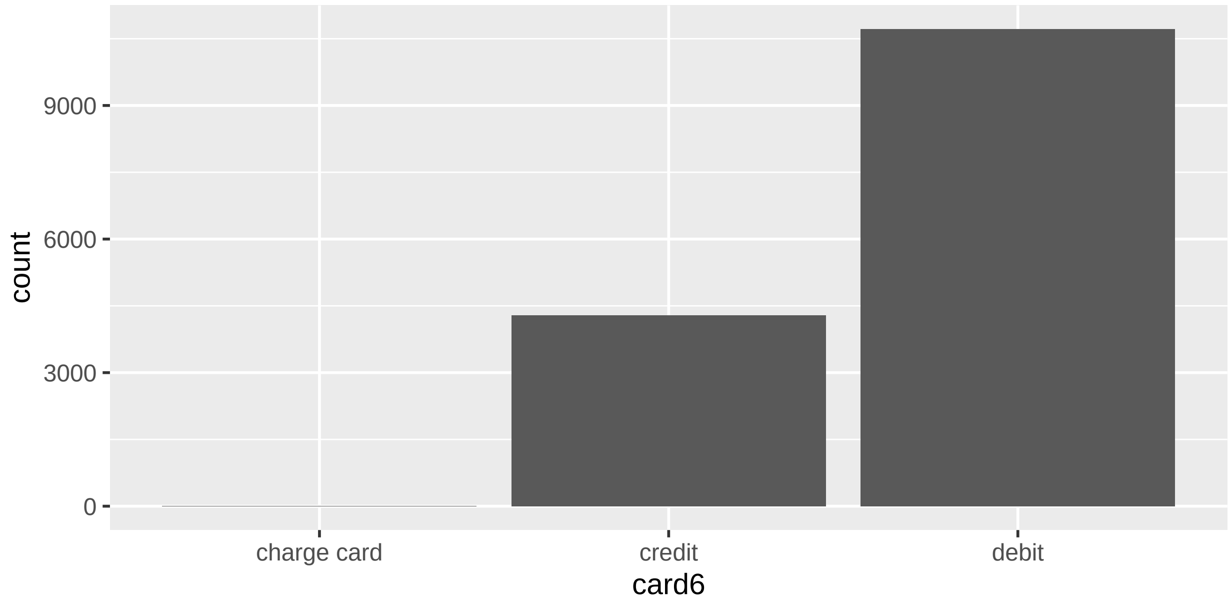


```
ds$card4[is.na(ds$card4)] <- ''
```

Then we replace the NA values with empty string ''.

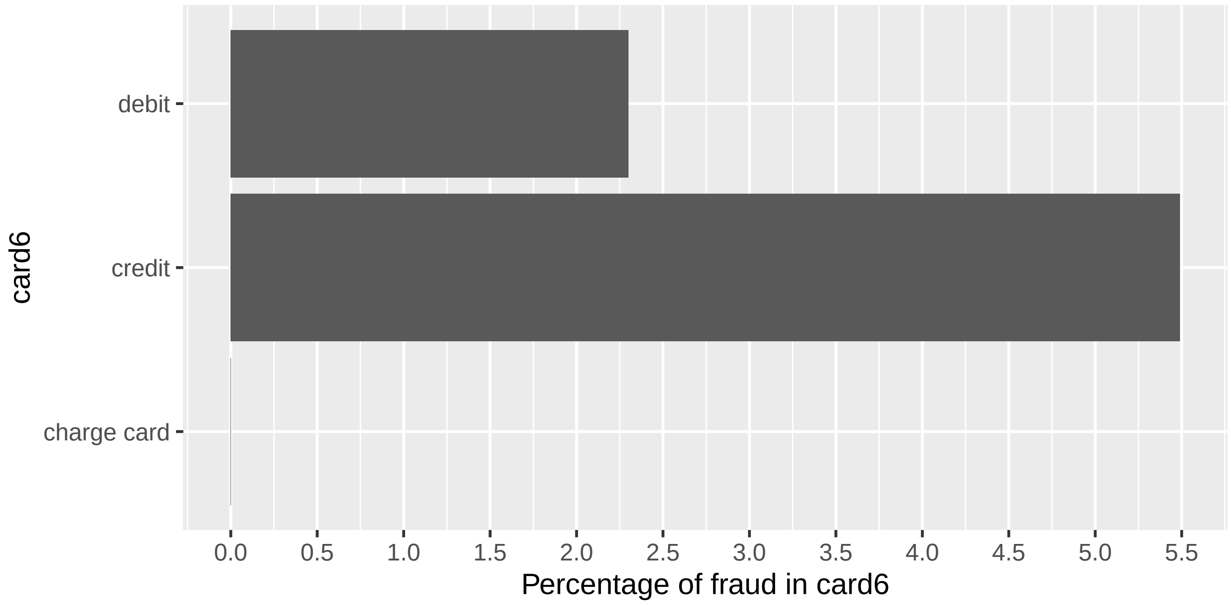
```
ggplot(ds) + aes(x=card6) + geom_bar()
```

card6



On *card6* we have different values like *charge card, credit and debit.* We analyze fraud transaction percentage of each nominal on *card6*.

```
d <- group_by(ds, card6) %>%
  mutate(fraud=sum(isFraud), notfraud=sum(isFraud==0), n=nrow(ds)) %>%
  select(card6, fraud, notfraud, n)
d <- unique(d)
ggplot(d) + aes(x=card6, y=(fraud/notfraud)*100) + coord_flip() + geom_col() +
  scale_y_continuous(name="Percentage of fraud in card6", breaks = seq(0,6,0.5))
```



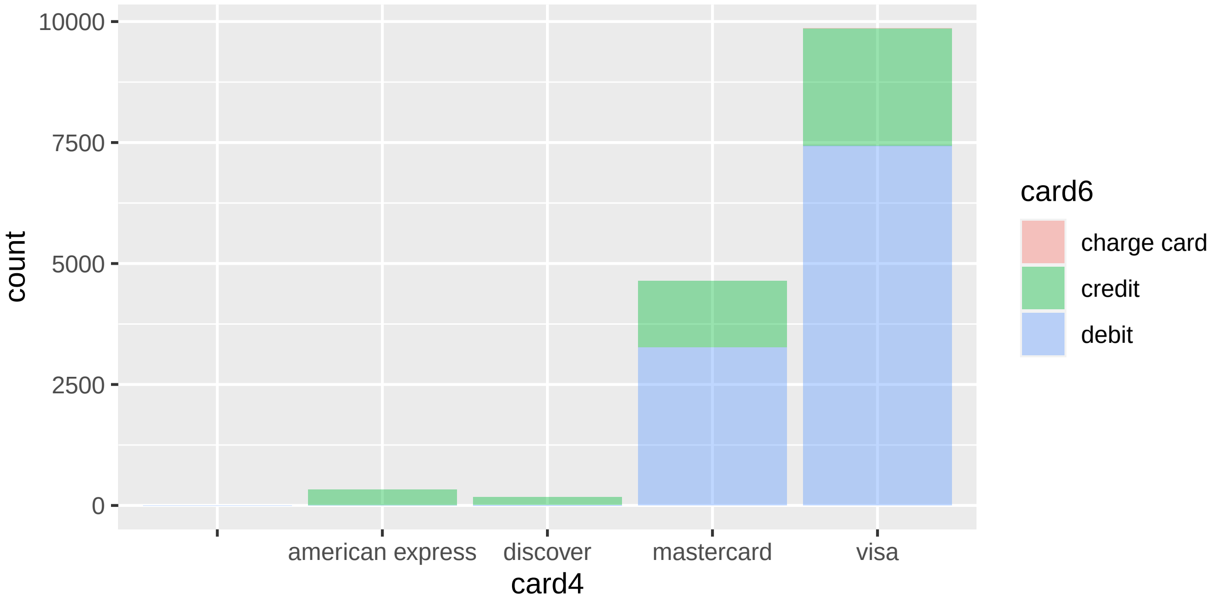
```
nrow(filter(ds, card6=='charge card'))
```

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

We only have 1 transaction by *charge card* and it is not fraudulent.

```
ggplot(ds) + aes(x=card4, fill=card6) + geom_bar(alpha=0.4)
```

card4 and card6 relationship



At *american express and discover*, the percentage of debit cards is very small. At *mastercad and visa* there are more debit cards. But there is also a significant amount of credit cards.

```
ds <- ds %>% mutate(card=group_indices(ds, card1, card2, card3, card4, card5, card6))
```

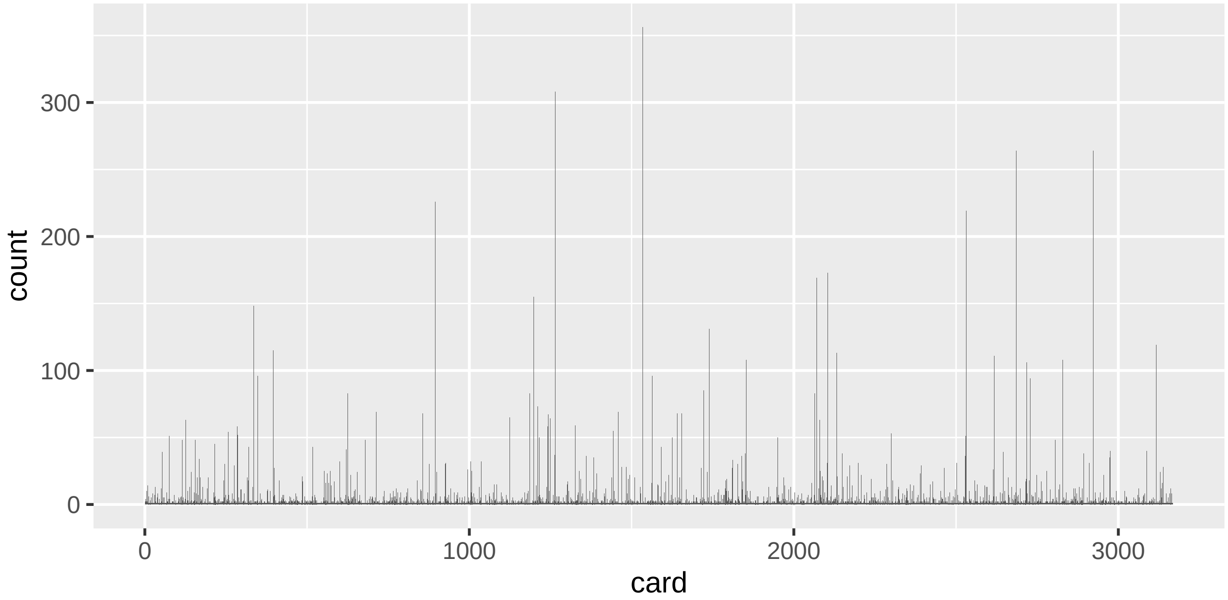
Create card ID named card We create new column named card subject to unique combination of card1-card6.

```
nrow(unique(select(ds, card)))
```

```
## [1] 3168
```

Therefore, we conclude that there are 3168 different cards for 15000 transactions.

```
ggplot(ds) + aes(x=card) + geom_bar()
```



Here we can see the number of transactions per card.

```
ds %>% group_by(card) %>% tally() %>% arrange(desc(n)) %>% slice(1:5)
```

```
## # A tibble: 5 x 2
##    card     n
##   <int> <int>
## 1  1535   356
## 2  1264   308
## 3  2686   264
## 4  2922   264
## 5   895   226
```

Top 5 cards with more transactions.

TransactionDT

```
summary(ds$TransactionDT)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   37000 1587832 2966398 3289065 5113998 7256593
```

By this values, we can see that the transactions date is on seconds and it is counting the seconds from a given reference in time (not an actual timestamp). Therefore, let's say *TransactionDT* is the seconds passed from day0. On our analyze, we use January 1, 2020 as that reference.

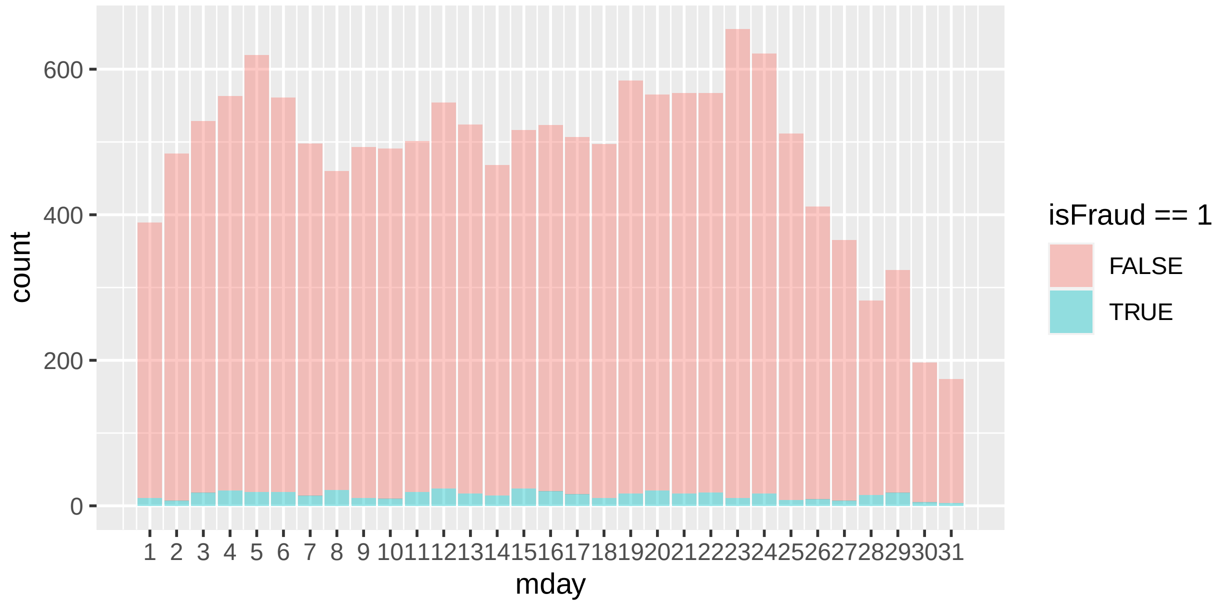
```
day0 <- dmy_hms("1/1/2020 00:00:00 GMT")
ds <- mutate(ds, hour=hour(day0+seconds(TransactionDT)))
ds <- mutate(ds, mday=mday(day0+seconds(TransactionDT)))
ds <- mutate(ds, wday=wday(day0+seconds(TransactionDT)))
```

Adding columns hour, mday, wday to represent hour, day of month and weekday of transaction.

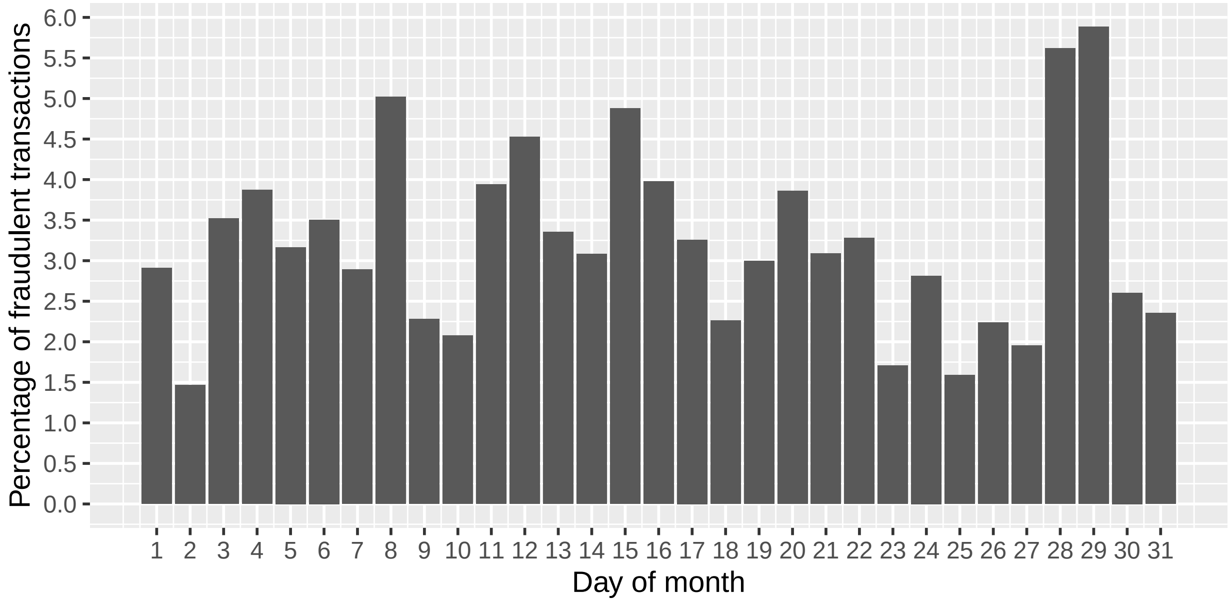
Ploting transactions per day of month, weekday and hour

```
ggplot(ds) + aes(x=mday, fill=isFraud==1) + geom_bar(alpha=0.4) +
  scale_x_continuous(breaks = seq(1,31,1))
```

Day of month



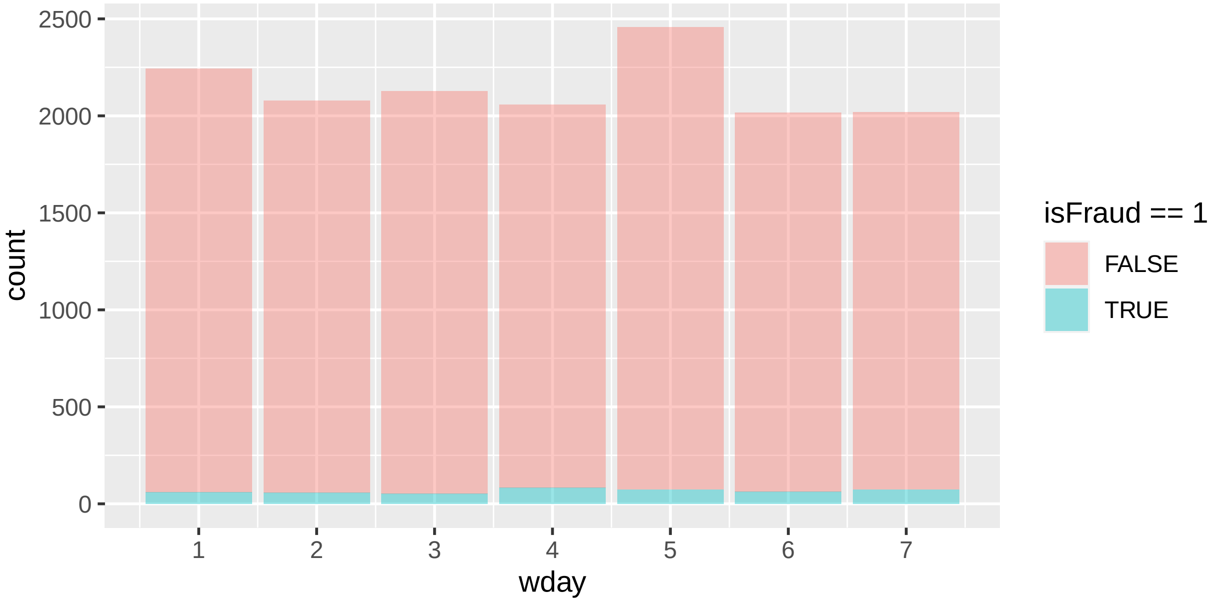
```
md <- group_by(ds, mday) %>% mutate(fraud=sum(isFraud), notfraud=sum(isFraud==0), n=nrow(ds)) %>%
  select(mday, fraud, notfraud, n)
md <- unique(md)
ggplot(md) + aes(x=mday, y=(fraud/notfraud)*100) + geom_col() +
  scale_x_continuous(name="Day of month", breaks = seq(1,31,1)) +
  scale_y_continuous(name="Percentage of fraudulent transactions", breaks = seq(0,6,0.5))
```



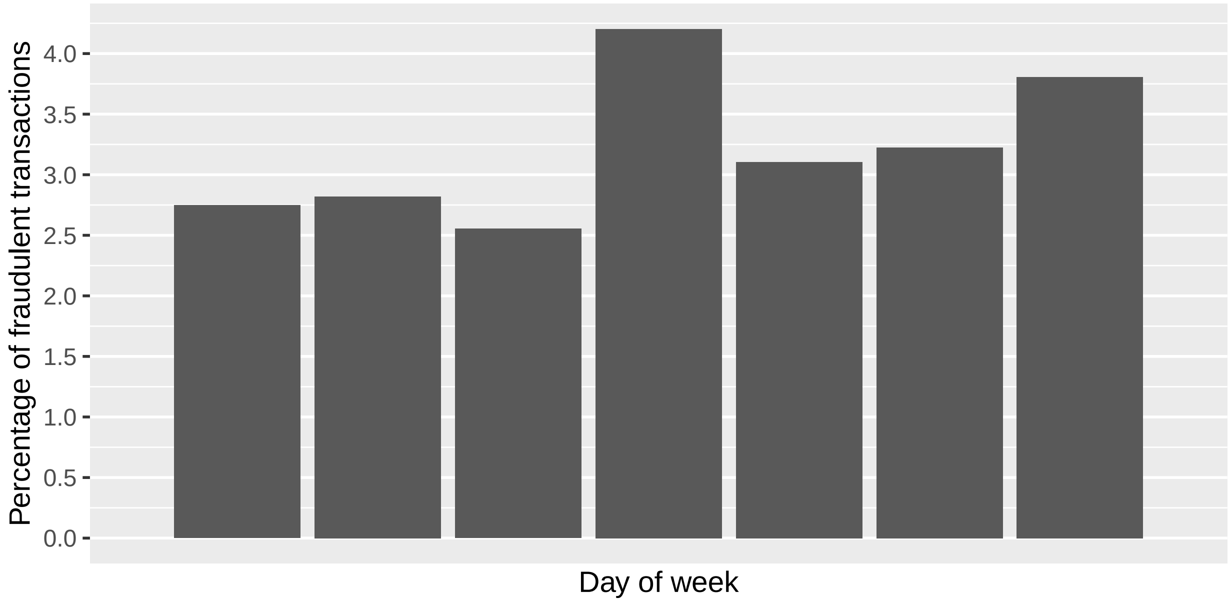
The fraudulent transactions are well spread out over the days of the month. However, by looking at the percentage of fraudulent transactions, we saw that last week of the month there's less transactions and days 28-29 have more fraudulent transactions.

```
ggplot(ds) + aes(x=wday, fill=isFraud==1) + geom_bar(alpha=0.4) +
  scale_x_continuous(breaks = seq(1,7,1))
```

Weekday



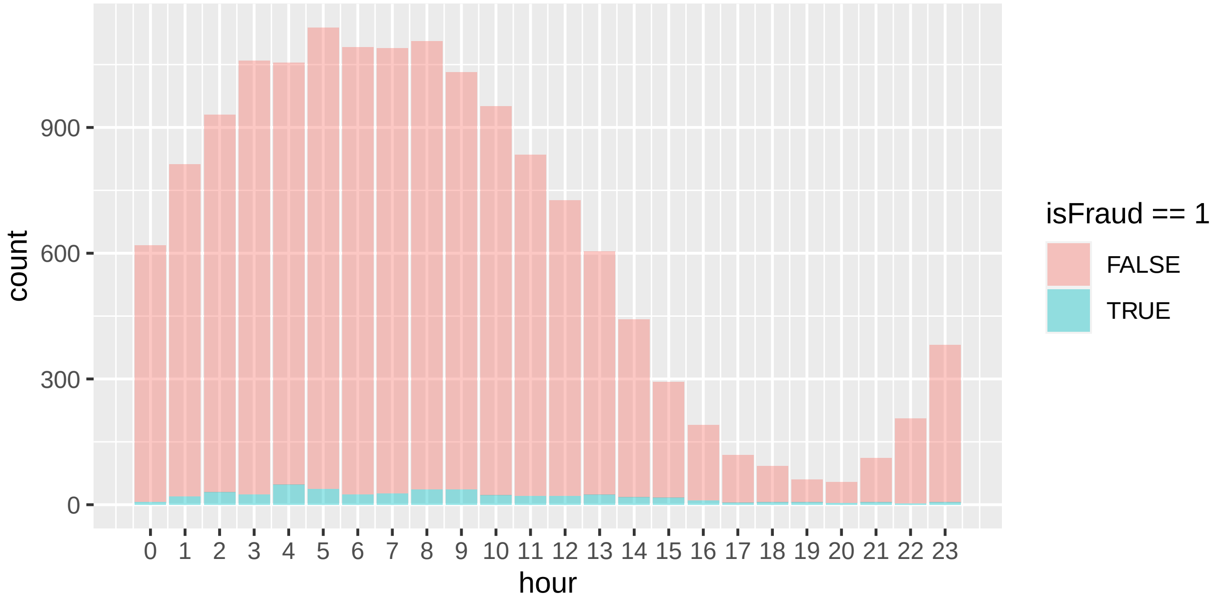
```
wd <- group_by(ds, wday) %>%
  mutate(fraud=sum(isFraud), notfraud=sum(isFraud==0), n=nrow(ds)) %>%
  select(wday, fraud, notfraud, n)
wd <- unique(wd)
ggplot(wd) + aes(x=wday, y=(fraud/notfraud)*100) + geom_col() +
  scale_x_discrete(name="Day of week", breaks = seq(1,31,1)) +
  scale_y_continuous(name="Percentage of fraudulent transactions", breaks = seq(0,6,0.5))
```



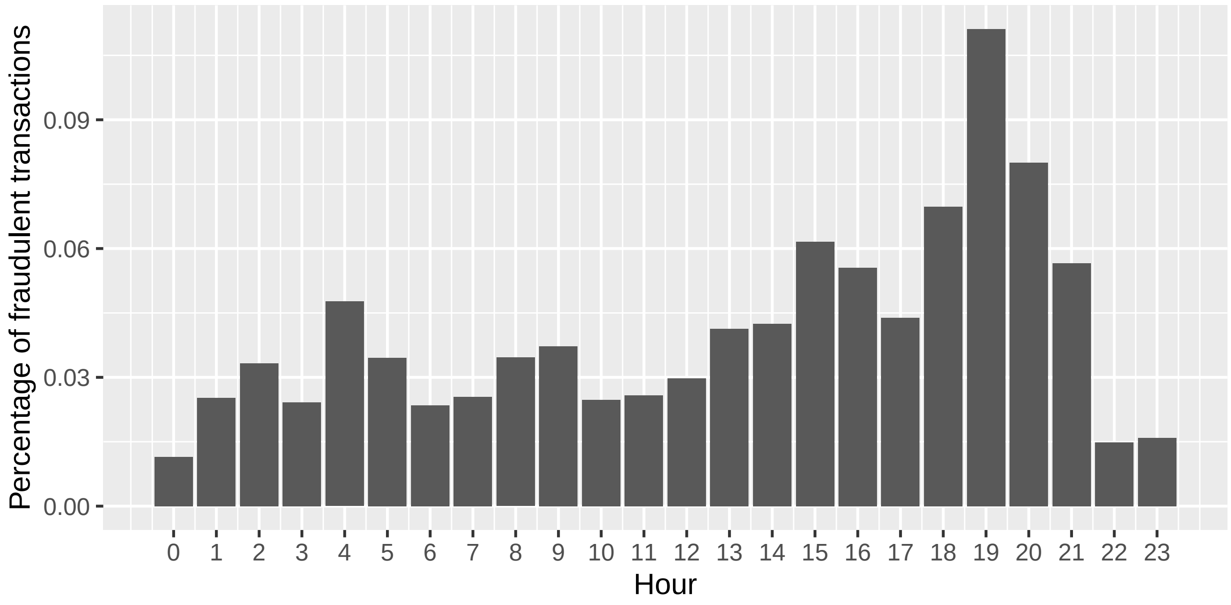
We can see that there were more transactions on Friday but more fraudulent transaction on Thursday followed by Sunday.

Hour

```
ggplot(ds) + aes(x=hour, fill=isFraud==1) + geom_bar(alpha=0.4) +
  scale_x_continuous(breaks = seq(0,23,1))
```



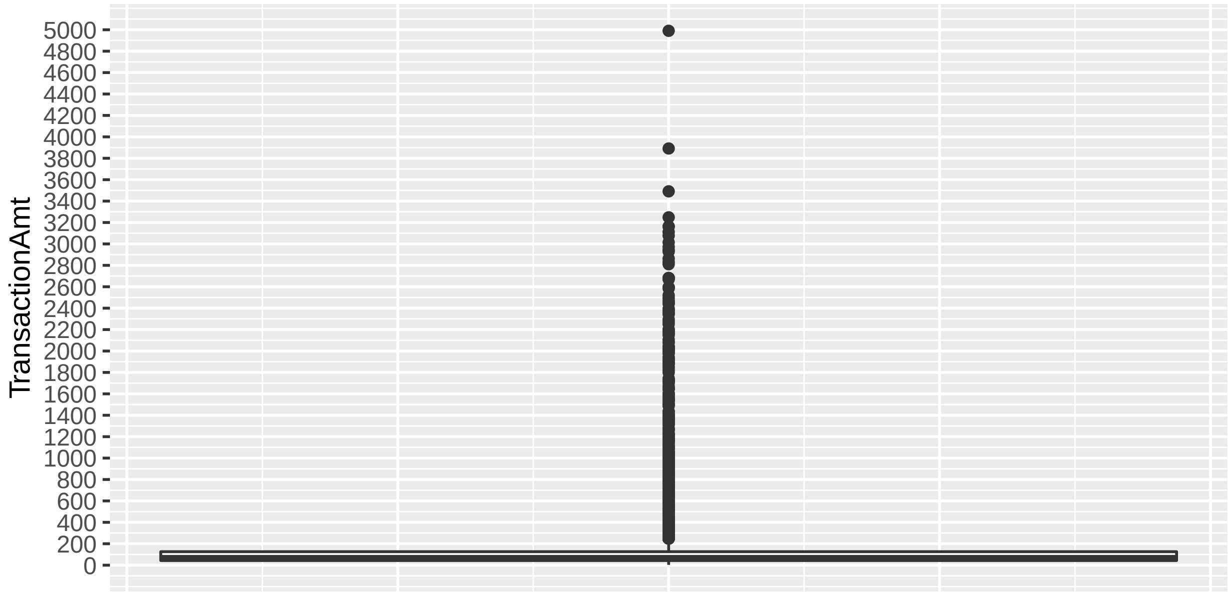
```
h <- group_by(ds, hour) %>%
  mutate(fraud=sum(isFraud), notfraud=sum(isFraud==0), n=nrow(ds)) %>%
  select(hour, fraud, notfraud, n)
h <- unique(h)
ggplot(h) + aes(x=hour, y=(fraud/notfraud)) + geom_col() +
  scale_x_continuous(name="Hour", breaks = seq(0,23,1)) +
  labs(y="Percentage of fraudulent transactions")
```



We can see that there's not a lot of transactions between 14-23, but it's where there are more percentage of fraudulent transactions.

TransactionAmt

```r
ggplot(ds) + aes(y=TransactionAmt) + geom_boxplot() +
  theme(axis.ticks.x = element_blank(), axis.text.x = element_blank()) +
  scale_y_continuous(breaks=seq(0,5000,200))
```



This box chart shows us the distribution of transactions by amount. We observe that approximately 75% of the transactions have the amount less than 200. To better visualize the distribution of transactions by amounts, we will set limits on the amounts.

```
d <- filter(ds, TransactionAmt>600)
nrow(d) / nrow(ds) * 100
```

TransactionAmt>600

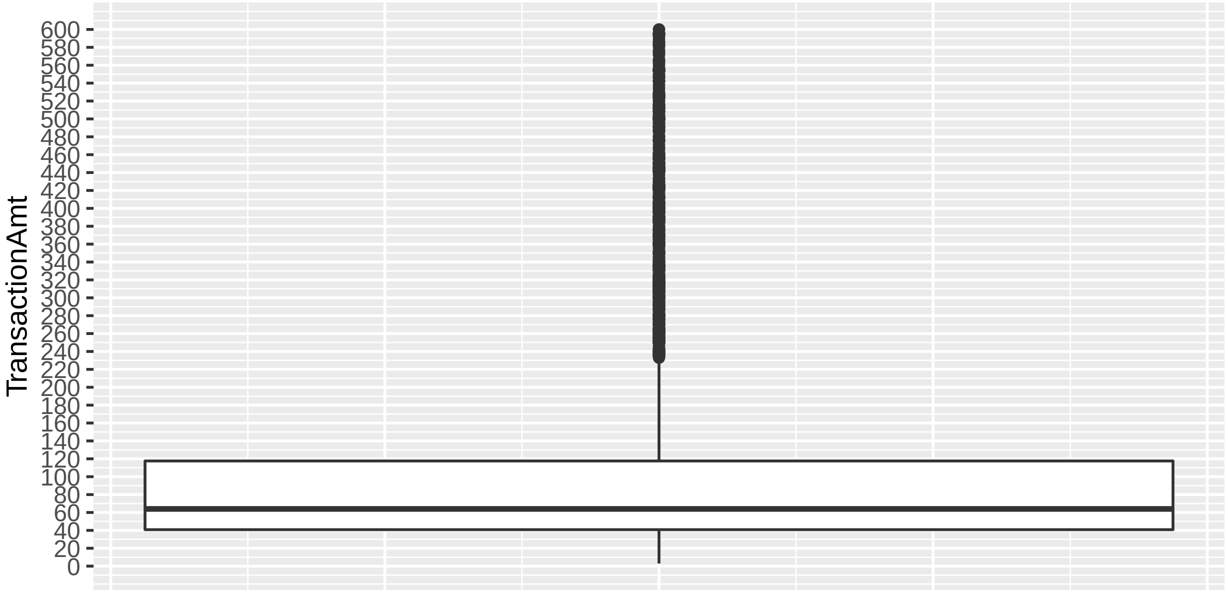
```
## [1] 2.613333
```

```
nrow(filter(d, isFraud==1)) / nrow(d) * 100
```

```
## [1] 2.55102
```

We observe that the percentage of transactions with the amount greater than 600 is 2.61% and that the percentage of fraudulent transactions with the amount in these values is 2.55%.

```
d <- filter(ds, TransactionAmt<=600)
ggplot(d) + aes(y=TransactionAmt) + geom_boxplot() +
  theme(axis.ticks.x = element_blank(), axis.text.x = element_blank()) +
  scale_y_continuous(breaks=seq(0,600,20))
```



From transactions with the amount less than or equal to 600, we observe that 25% of these transactions have an amount less than 40, 50% of these transactions have an amount between 40 and 120. With the amount greater than 120 we only have 25% of the transactions.

```
d <- filter(ds, TransactionAmt>=230 & TransactionAmt<=600)
nrow(d) / nrow(ds) * 100
```

TransactionAmt between 230 and 600

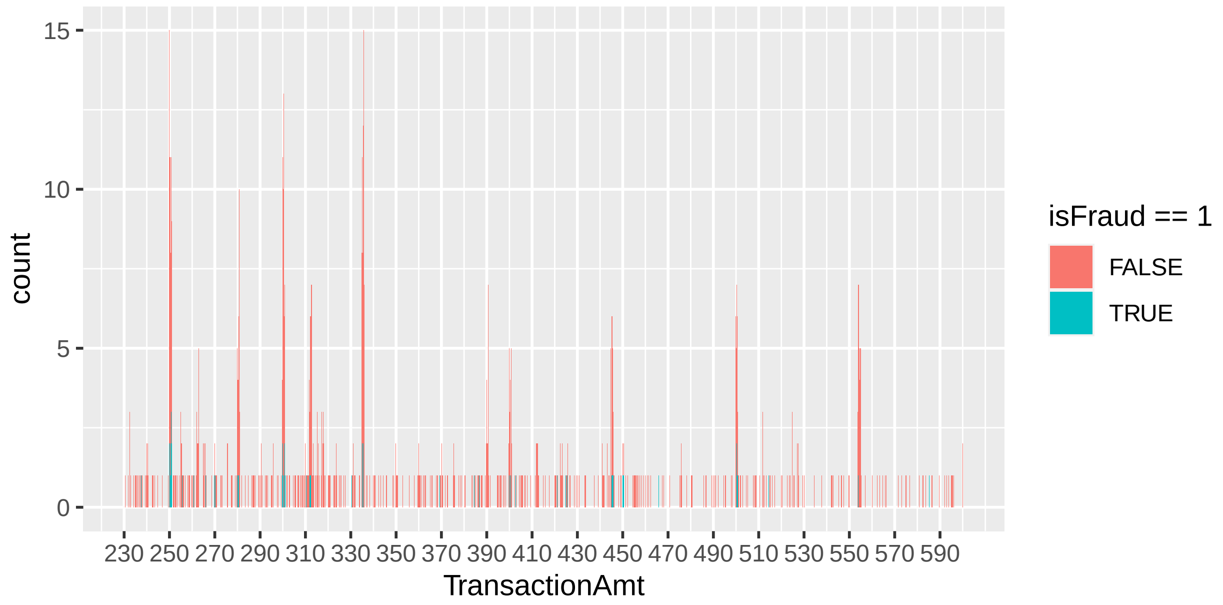
```
## [1] 8.746667
```

```
nrow(filter(d, isFraud==1)) / nrow(d) * 100
```

```
## [1] 4.72561
```

We observe that the percentage of transactions with the amount between 230 and 600 is 8.74% and that the percentage of fraudulent transactions with the amount in these values is 4.72%. In this observation, the percentage of fraudulent transactions is very high compared to our *isFraud* target.

```
ggplot(d) + aes(x=TransactionAmt, fill=isFraud==1) + geom_bar() +
  scale_x_continuous(breaks=seq(230,600, 20))
```



```
d <- filter(ds, TransactionAmt<230)
nrow(d) / nrow(ds) * 100
```

TransactionAmt < 230

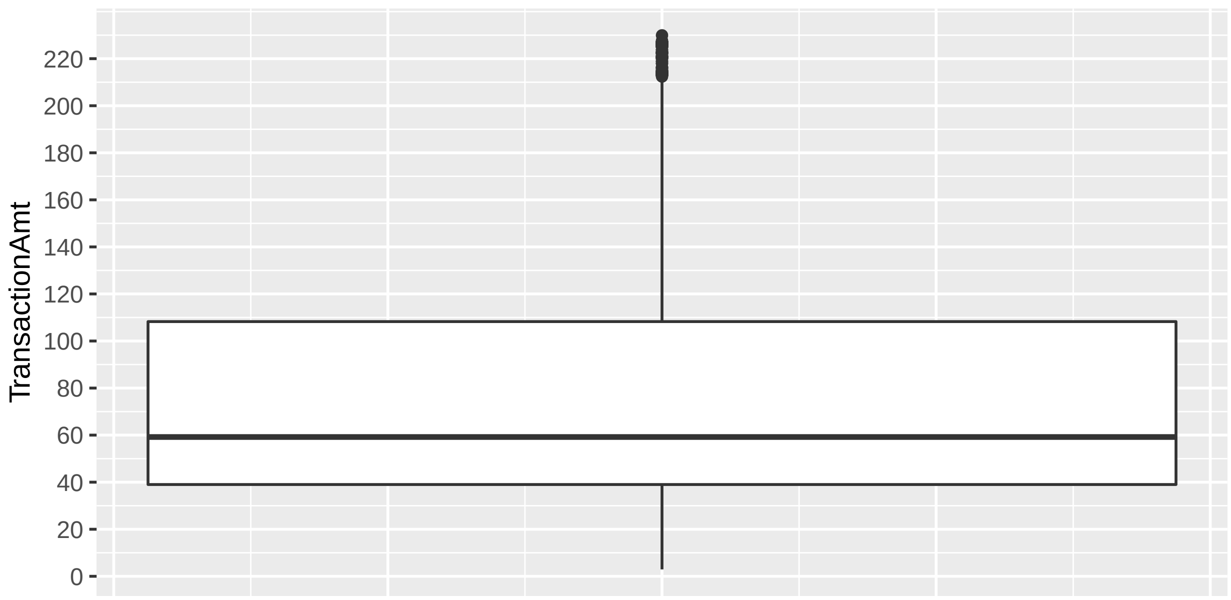
```
## [1] 88.64
```

```
nrow(filter(d, isFraud==1)) / nrow(d) * 100
```

```
## [1] 2.948255
```

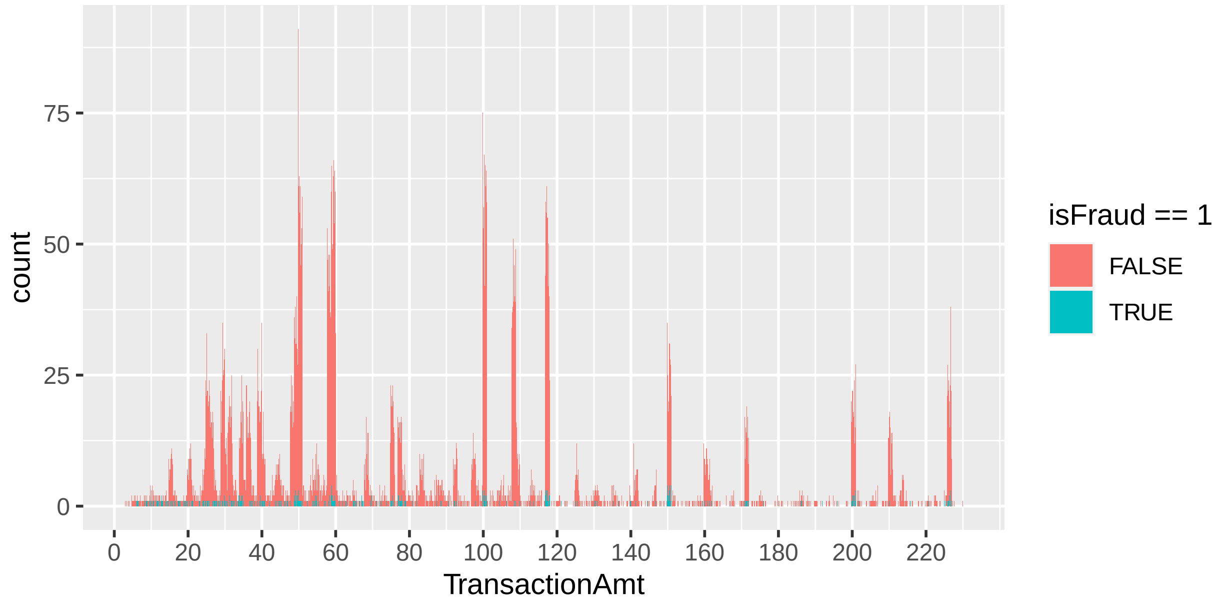
We observe that the percentage of transactions with the amount less than 230 is 88.64% and the percentage of fraudulent transactions with the amount in these values is 2.94% (which is higher than the percentage of the amount > 600). However, the percentage of fraud falls short of our *isFraud* target.

```
ggplot(d) + aes(y=TransactionAmt) + geom_boxplot() +
  theme(axis.ticks.x = element_blank(), axis.text.x = element_blank()) +
  scale_y_continuous(breaks=seq(0,230,20))
```



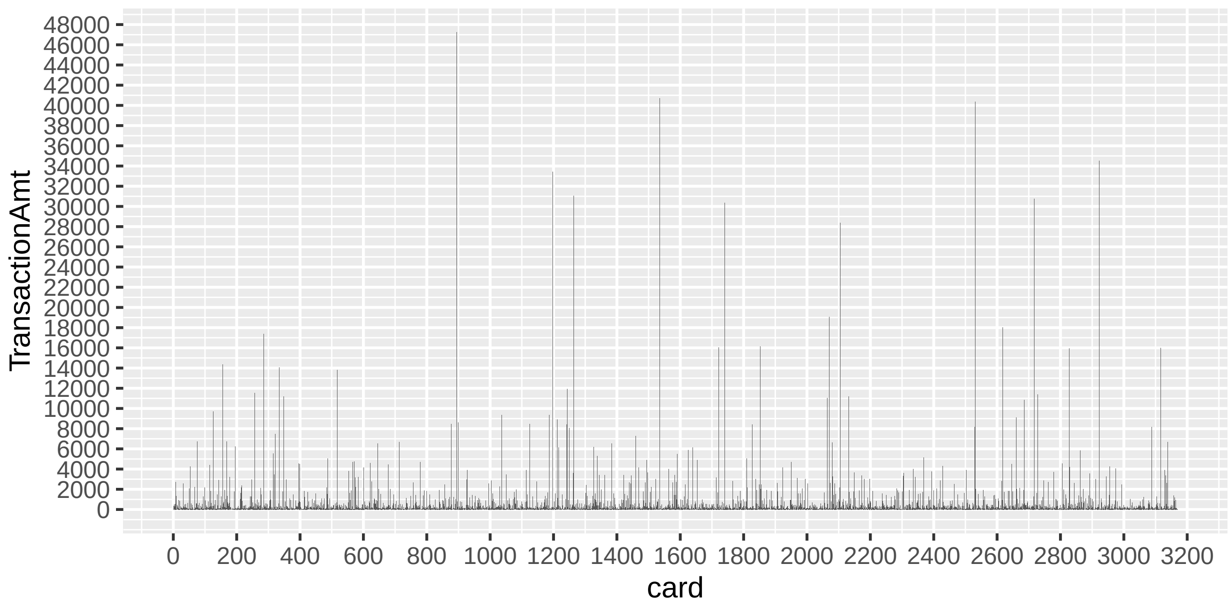
From transactions with the amount less than 230, we observe that 25% of these transactions have an amount less than 40, 50% of these transactions have an amount between 40 and 110. With the amount greater than 110 we only have 25% of the transactions. Now we can conclude that the most percentage of transactions dataset have amount between 40 and 110.

```
ggplot(d) + aes(x=TransactionAmt, fill=isFraud==1) + geom_bar() +
  scale_x_continuous(breaks=seq(0,230, 20))
```



```
ggplot(ds) + aes(x=card, y=TransactionAmt) + geom_col() +
  scale_x_continuous(breaks=seq(0,3200, 200)) +
  scale_y_continuous(breaks=seq(0,50000, 2000))
```

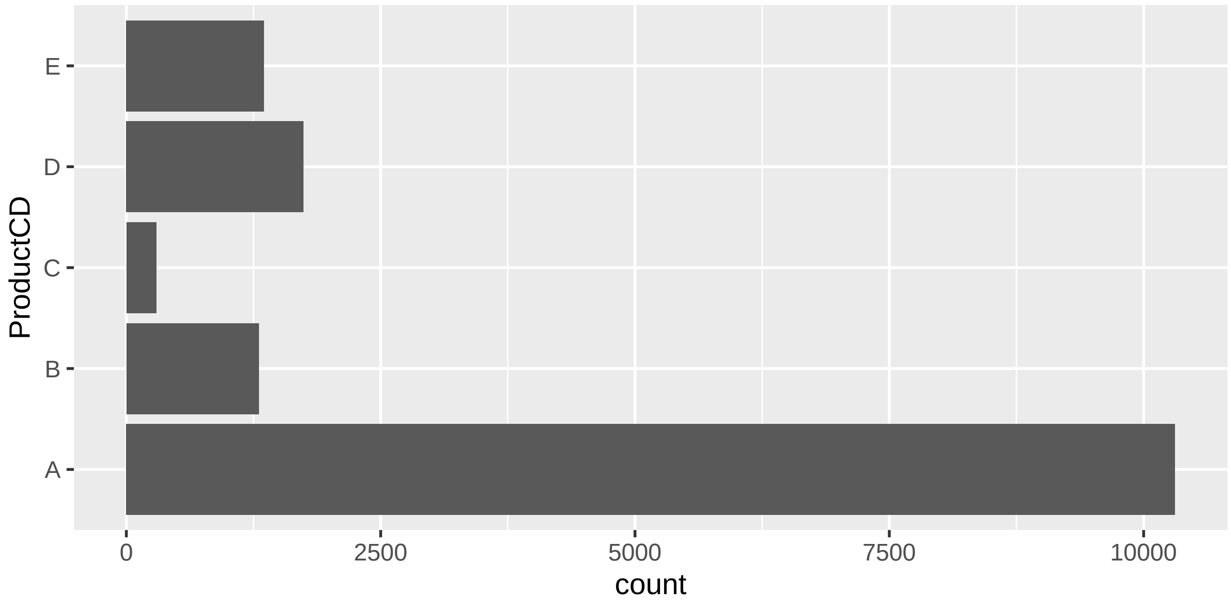
Spent amount per card



We analyze the total amount spent on transactions for each card.

ProductCD

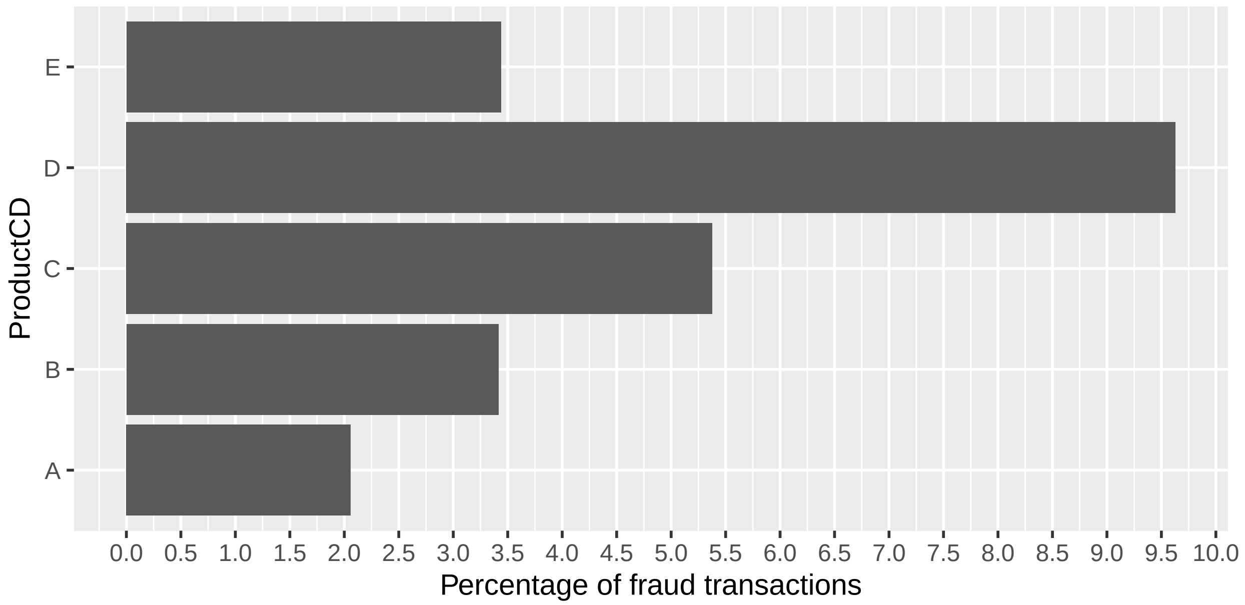
```
ggplot(ds)  + aes(x=ProductCD) + coord_flip() + geom_bar()
```



The transactions refer to 5 different products. Product A was the product that generated the most transactions (over 10000).

```
product <- group_by(ds, ProductCD) %>%
  mutate(fraud=sum(isFraud), notfraud=sum(isFraud==0), n=nrow(ds)) %>%
  select(ProductCD, fraud, notfraud, n)
product <- unique(product)
```

```
ggplot(product) + aes(x=ProductCD, y=(fraud/notfraud)*100) + coord_flip() +
  geom_col() +
  scale_y_continuous(name="Percentage of fraud transactions", breaks = seq(0,10,0.5))
```



However, we observe that product A has the lowest percentage of fraudulent transactions. Products D and C are the products that have the percentage of fraudulent transactions that is higher our target.

```
a <- filter(ds, ProductCD=='A') %>% summarise(ProductCD='A',
                                          minAmt=min(TransactionAmt),
                                          maxAmt=max(TransactionAmt),
                                          meanAmt=mean(TransactionAmt))

b <- filter(ds, ProductCD=='B') %>% summarise(ProductCD='B',
                                               minAmt=min(TransactionAmt),
                                               maxAmt=max(TransactionAmt),
                                               meanAmt=mean(TransactionAmt))

c <- filter(ds, ProductCD=='C') %>% summarise(ProductCD='C',
                                               minAmt=min(TransactionAmt),
                                               maxAmt=max(TransactionAmt),
                                               meanAmt=mean(TransactionAmt))

d <- filter(ds, ProductCD=='D') %>% summarise(ProductCD='D',
                                               minAmt=min(TransactionAmt),
                                               maxAmt=max(TransactionAmt),
                                               meanAmt=mean(TransactionAmt))

e <- filter(ds, ProductCD=='E') %>% summarise(ProductCD='E',
                                               minAmt=min(TransactionAmt),
                                               maxAmt=max(TransactionAmt),
                                               meanAmt=mean(TransactionAmt))
```

```
productAmt <- full_join(full_join(full_join(full_join(a,b),c),d),e)
```

Associate product to transaction amount

```
## Joining, by = c("ProductCD", "minAmt", "maxAmt", "meanAmt")
## Joining, by = c("ProductCD", "minAmt", "maxAmt", "meanAmt")
## Joining, by = c("ProductCD", "minAmt", "maxAmt", "meanAmt")
## Joining, by = c("ProductCD", "minAmt", "maxAmt", "meanAmt")
```

```
productAmt
```

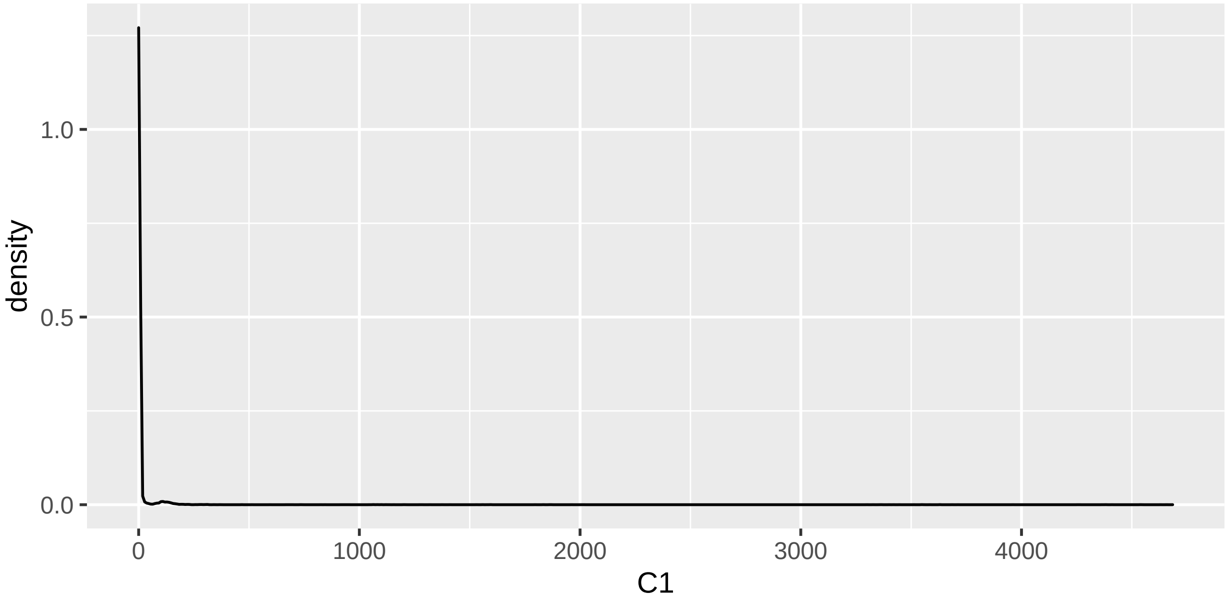
```
##   ProductCD minAmt   maxAmt   meanAmt
## 1         A   7.99 4990.470 152.70266
## 2         B  14.80  450.500  72.68710
## 3         C   4.80 1000.400  59.30901
## 4         D   2.94  486.582  43.80869
## 5         E  24.80 1800.000 176.73589
```

We look at statistics related to the amount of each product, namely minimum, maximum and the average amount.

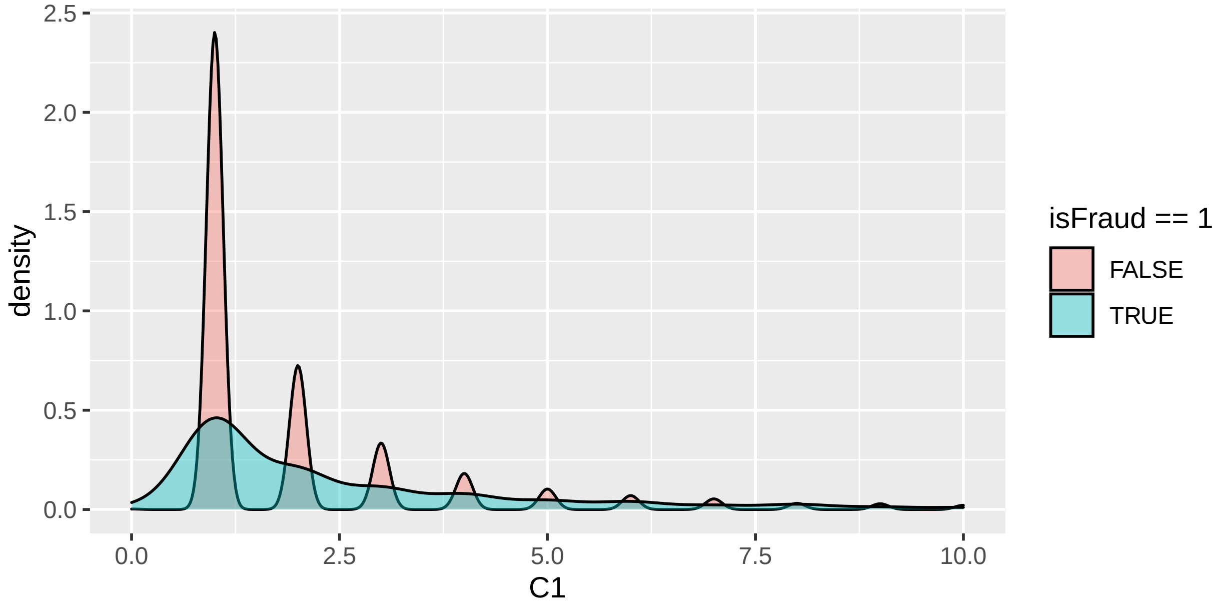
C1-C14

C1

```
ggplot(ds) + aes(x=C1) + geom_density(alpha=0.4)
```



```
ggplot(filter(ds, C1<=10)) + aes(x=C1, fill=isFraud==1) + geom_density(alpha=0.4)
```



```
nrow(filter(ds, C1<=10)) / nrow(ds) * 100
```

```
## [1] 93.4
```

```
nrow(filter(ds, C1<=10 & isFraud==1)) / nrow(filter(ds, C1<=10)) * 100
```

```
## [1] 2.76945
```

```
nrow(filter(ds, C1>10)) / nrow(ds) * 100
```

```
## [1] 6.6
```

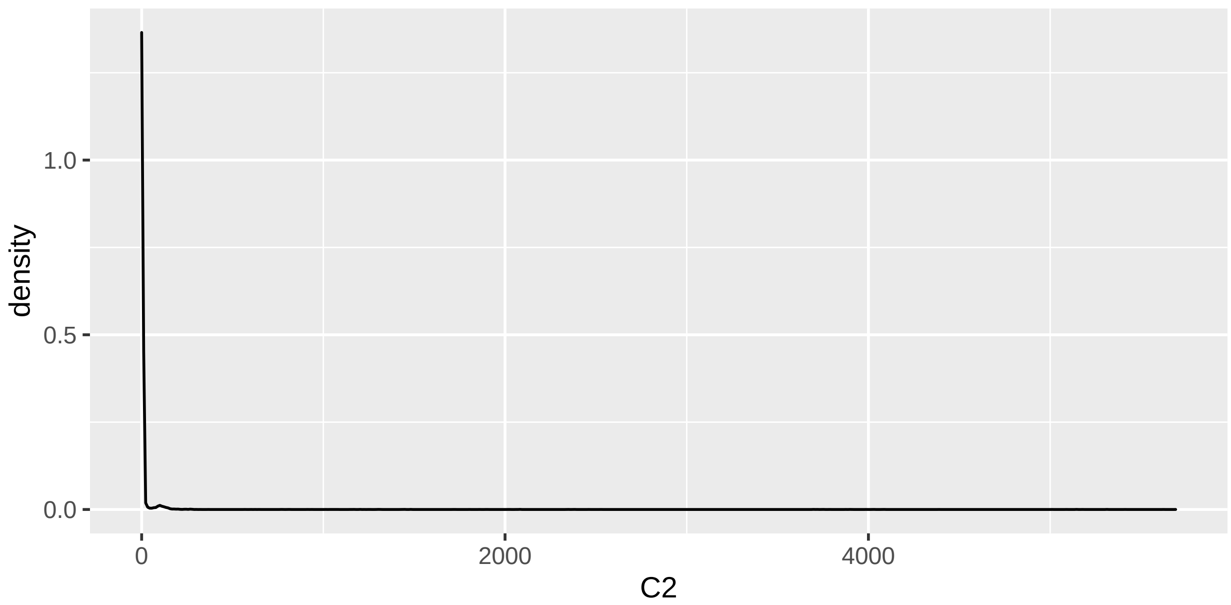
```
nrow(filter(ds, C1>10 & isFraud==1)) / nrow(filter(ds, C1>10)) * 100
```

```
## [1] 7.676768
```

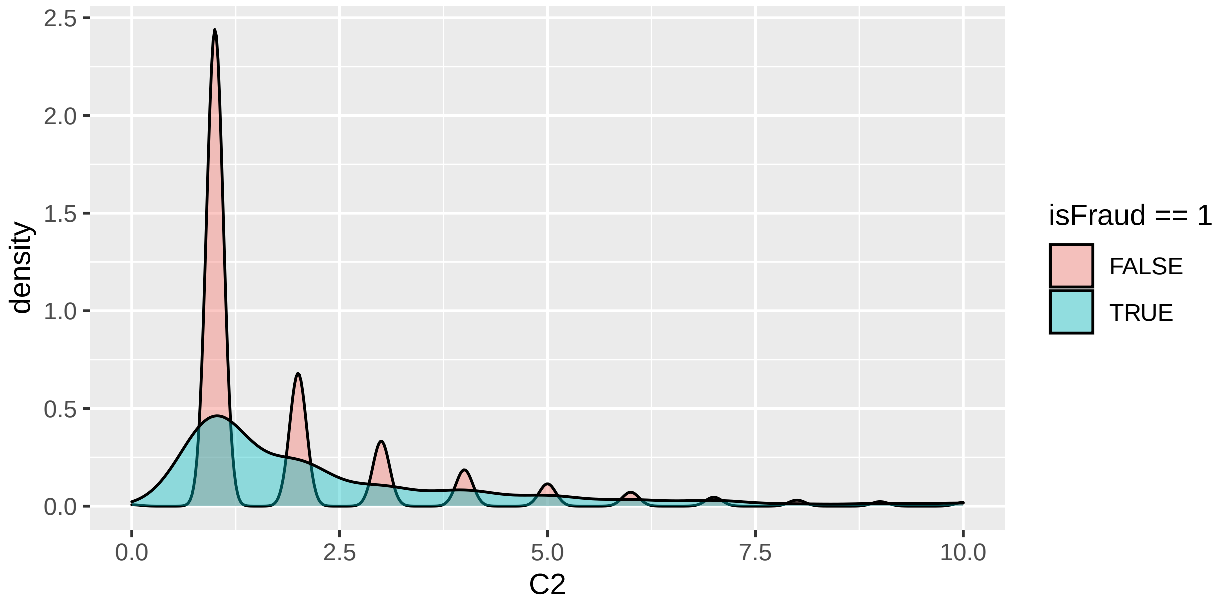
We can see that 93.4% of the cases have C1 values less than or equal to 10, but when that isn't the case, the percentage of fraudulent transactions are quite high, around 7.6%

C2

```
ggplot(ds) + aes(x=C2) + geom_density(alpha=0.4)
```



```
ggplot(filter(ds, C2<=10)) + aes(x=C2, fill=isFraud==1) + geom_density(alpha=0.4)
```



```
nrow(filter(ds, C2<=10)) / nrow(ds) * 100
```

```
## [1] 93.02667
```

```
nrow(filter(ds, C2<=10 & isFraud==1)) / nrow(filter(ds, C2<=10)) * 100
```

```
## [1] 2.780565
```

```
nrow(filter(ds, C2>10)) / nrow(ds) * 100
```

```
## [1] 6.973333
```

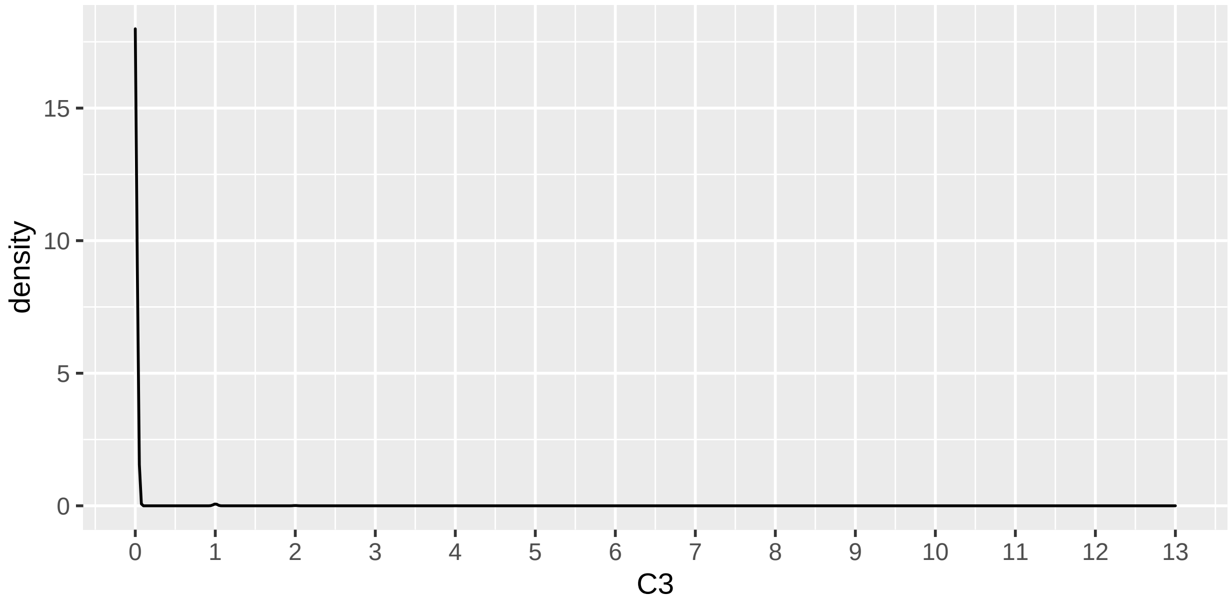
```
nrow(filter(ds, C2>10 & isFraud==1)) / nrow(filter(ds, C2>10)) * 100
```

```
## [1] 7.265774
```

We can see that C2 is very similar to C1.

C3

```
ggplot(ds) + aes(x=C3) + geom_density(alpha=0.4) +
  scale_x_continuous(breaks = seq(0,14,1))
```



```
head(unique(select(ds, C3) %>% arrange(desc(C3))))
```

```
##    C3
## 1: 13
## 2:  4
## 3:  2
## 4:  1
## 5:  0
```

```
nrow(filter(ds, C3==0)) / nrow(ds) * 100
```

```
## [1] 99.47333
```

```
nrow(filter(ds, C3==0 & isFraud==1)) / nrow(filter(ds, C3==0)) * 100
```

```
## [1] 3.109711
```

```
nrow(filter(ds, C3>0)) / nrow(ds) * 100
```

```
## [1] 0.5266667
```

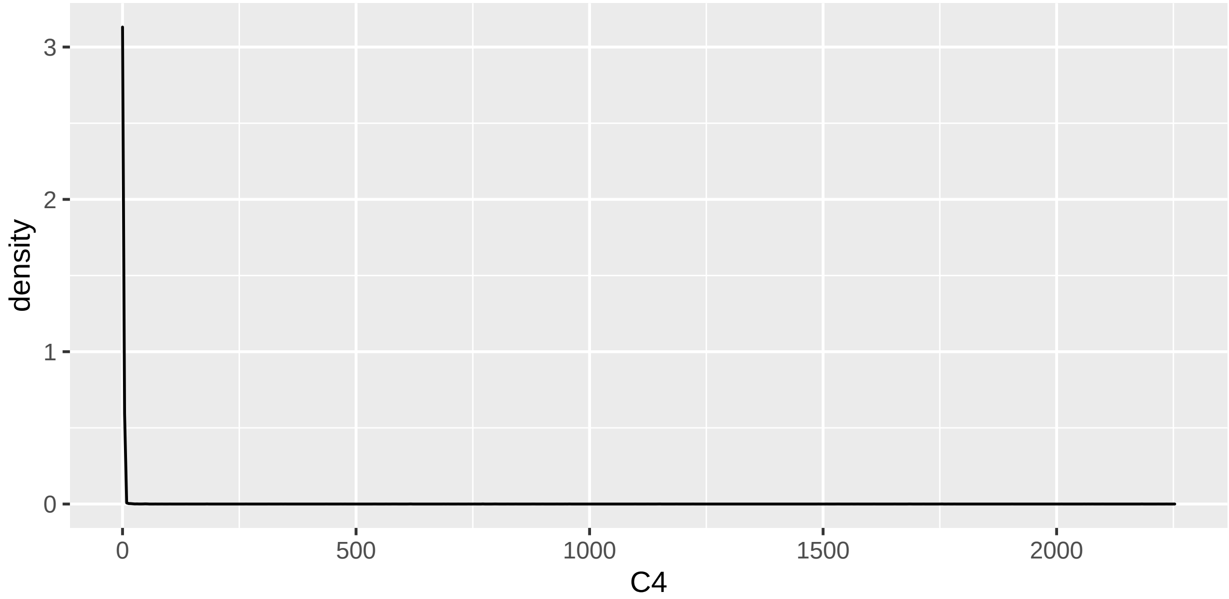
```
nrow(filter(ds, C3>0 & isFraud==1)) / nrow(filter(ds, C3>0)) * 100
```

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

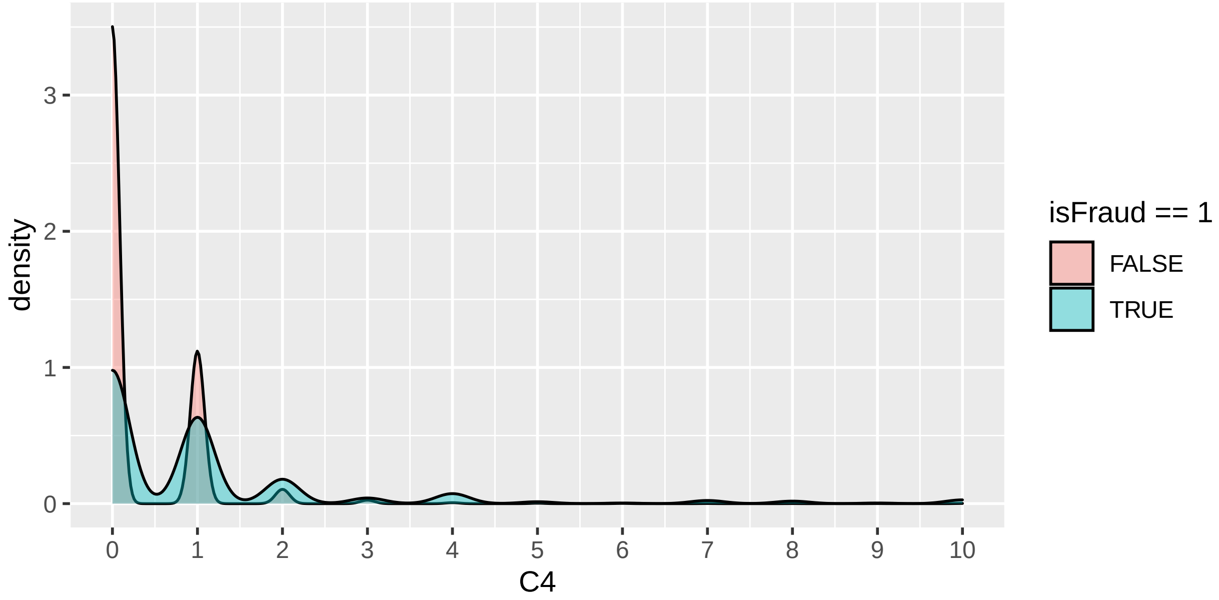
Feature C3 only has 5 different values, where more than 99% of the cases are 0 and it's where all the fraudulent transactions are.

C4

```
ggplot(ds) + aes(x=C4) + geom_density(alpha=0.4)
```



```
ggplot(filter(ds, C4<=10)) + aes(x=C4, fill=isFraud==1) +
  geom_density(alpha=0.4) +
  scale_x_continuous(breaks = seq(0,10,1))
```



```
nrow(filter(ds, C4<=10)) / nrow(ds) * 100
```

```
## [1] 99.20667
```

```
nrow(filter(ds, C4<=10 & isFraud==1)) / nrow(filter(ds, C4<=10)) * 100
```

```
## [1] 2.923191
```

```
nrow(filter(ds, C4==0)) / nrow(ds) * 100
```

```
## [1] 72.01333
```

```
nrow(filter(ds, C4==0 & isFraud==1)) / nrow(filter(ds, C4==0)) * 100
```

```
## [1] 1.971857
```

```
nrow(filter(ds, C4==1)) / nrow(ds) * 100
```

```
## [1] 23.64
```

```
nrow(filter(ds, C4==1 & isFraud==1)) / nrow(filter(ds, C4==1)) * 100
```

```
## [1] 3.891709
```

```
nrow(filter(ds, C4==2)) / nrow(ds) * 100
```

```
## [1] 2.4
```

```
nrow(filter(ds, C4==2 & isFraud==1)) / nrow(filter(ds, C4==2)) * 100
```

```
## [1] 10.83333
```

```
nrow(filter(ds, C4==3)) / nrow(ds) * 100
```

```
## [1] 0.5866667
```

```
nrow(filter(ds, C4==3 & isFraud==1)) / nrow(filter(ds, C4==3)) * 100
```

```
## [1] 10.22727
```

```
nrow(filter(ds, C4==4)) / nrow(ds) * 100
```

```
## [1] 0.2466667
```

```
nrow(filter(ds, C4==4 & isFraud==1)) / nrow(filter(ds, C4==4)) * 100
```

```
## [1] 43.24324
```

```
nrow(filter(ds, C4>=5)) / nrow(ds) * 100
```

```
## [1] 1.113333
```

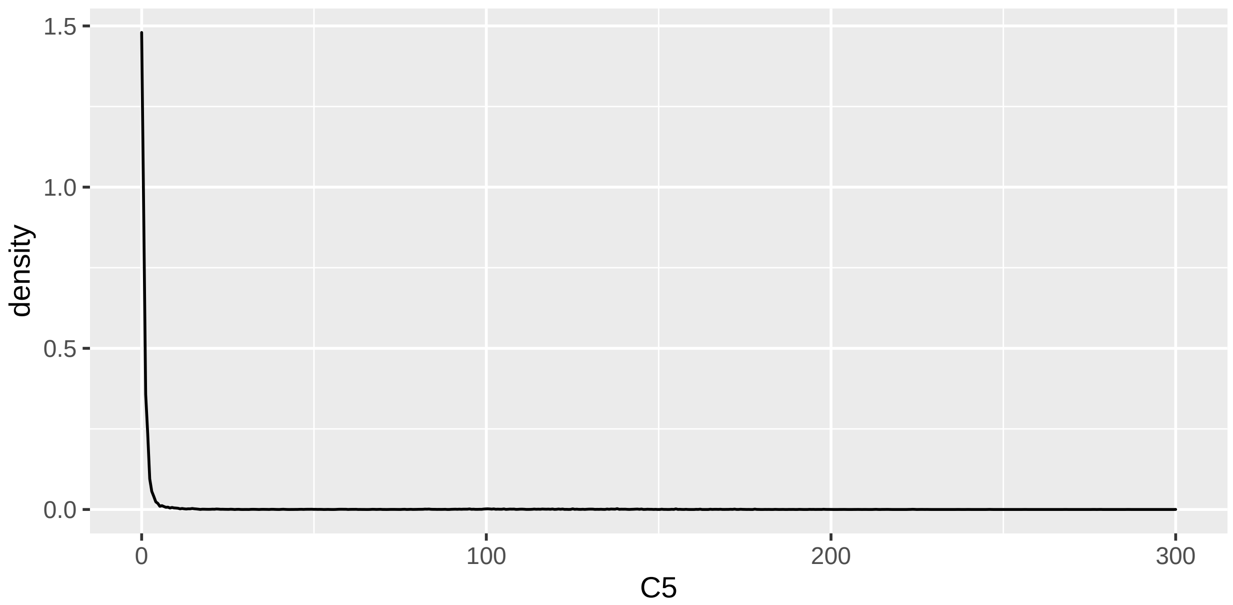
```
nrow(filter(ds, C4>=5 & isFraud==1)) / nrow(filter(ds, C4>=5)) * 100
```

```
## [1] 29.34132
```

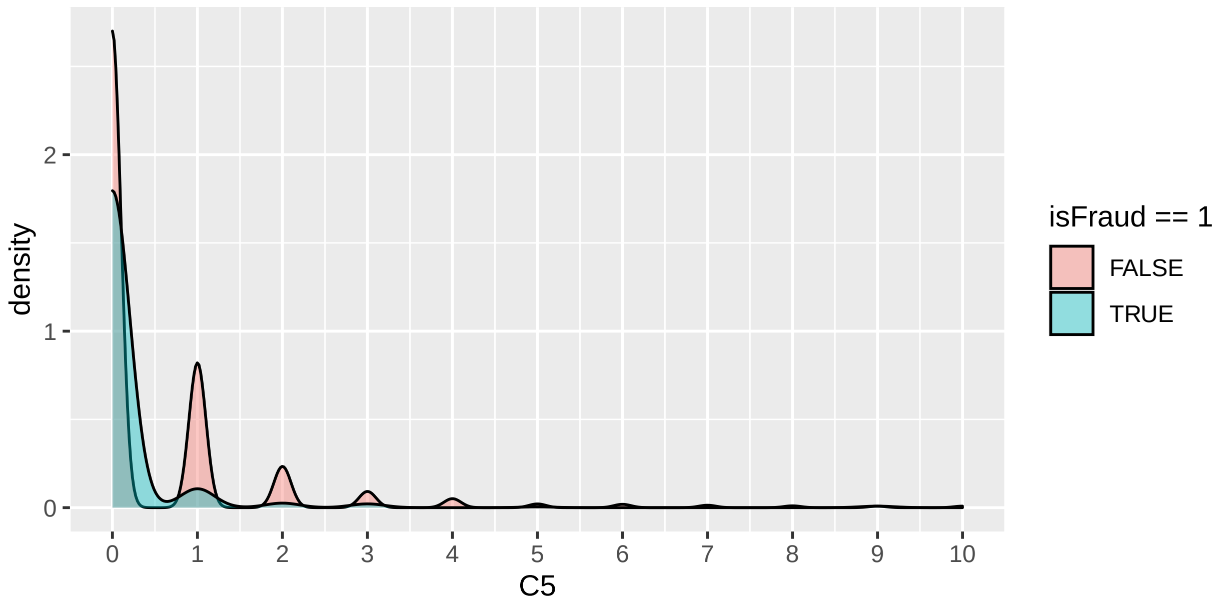
We can see that C4 is similar to C1 and C2, but when the values are greater than 0, the percentage of fraudulent transactions are way higher, special case for when C4 equals 4 the percentage is almost 50%.

C5

```
ggplot(ds) + aes(x=C5) + geom_density(alpha=0.4)
```



```
ggplot(filter(ds, C5<=10)) + aes(x=C5, fill=isFraud==1) +
  geom_density(alpha=0.4) +
  scale_x_continuous(breaks = seq(0,10,1))
```



```
nrow(filter(ds, C5<=10)) / nrow(ds) * 100
```

```
## [1] 95.52
```

```
nrow(filter(ds, C5<=10 & isFraud==1)) / nrow(filter(ds, C5<=10)) * 100
```

```
## [1] 3.196538
```

```
nrow(filter(ds, C5==0)) / nrow(ds) * 100
```

```
## [1] 65.50667
```

```
nrow(filter(ds, C5==0 & isFraud==1)) / nrow(filter(ds, C5==0)) * 100
```

```
## [1] 4.264197
```

```
nrow(filter(ds, C5>0)) / nrow(ds) * 100
```

```
## [1] 34.49333
```

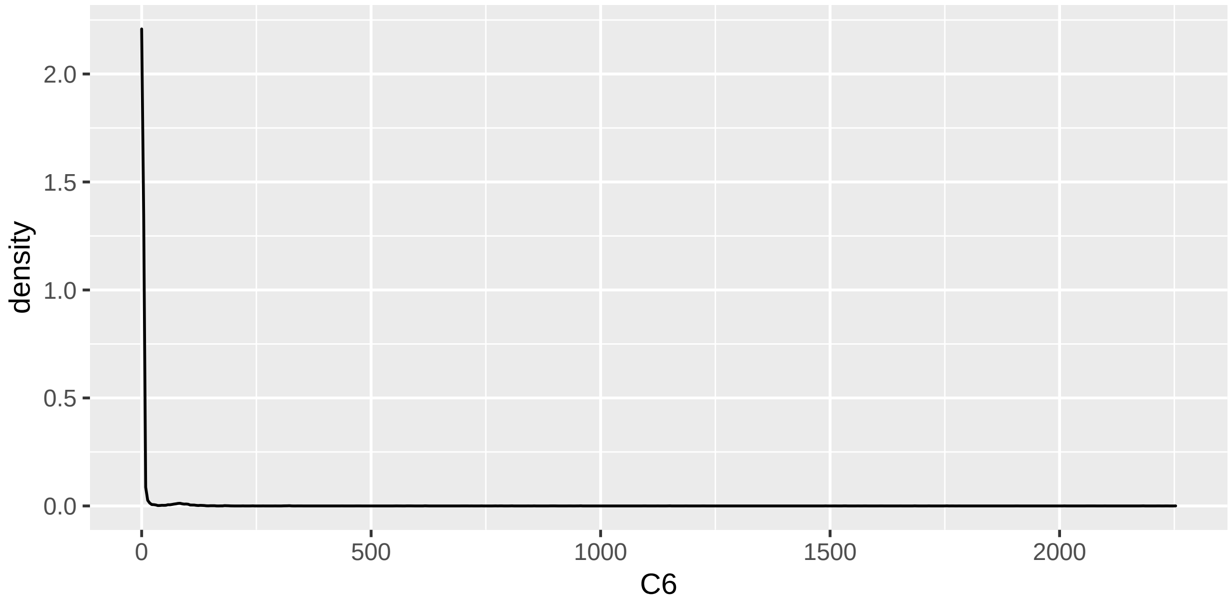
```
nrow(filter(ds, C5>0 & isFraud==1)) / nrow(filter(ds, C5>0)) * 100
```

```
## [1] 0.8697333
```

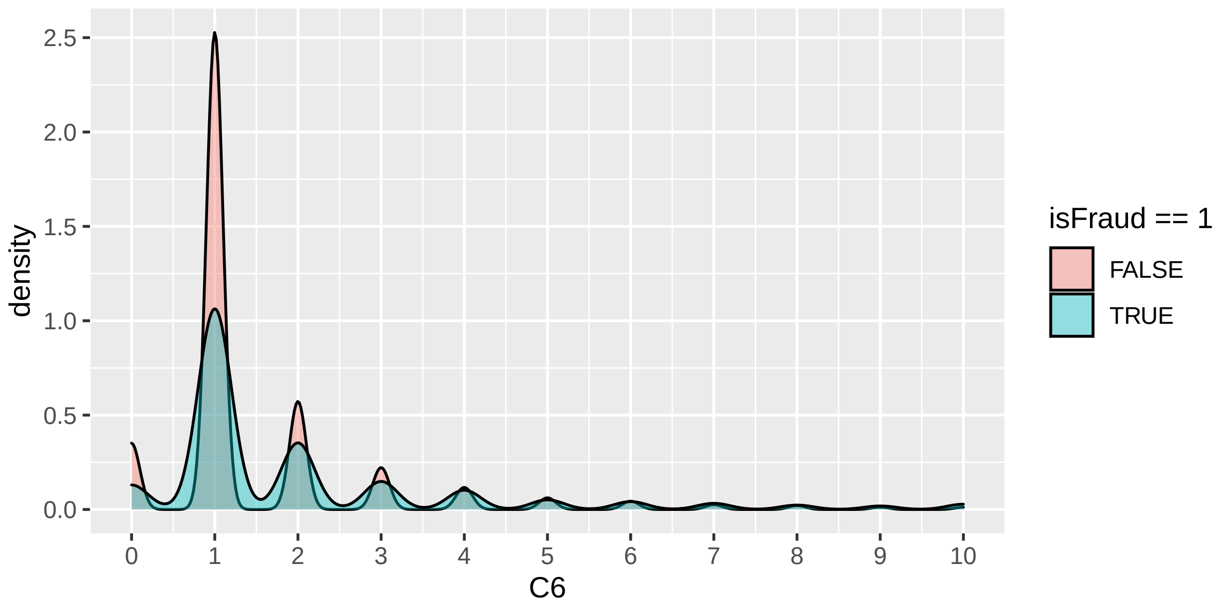
This feature has its values a little more scattered than the previous ones. When C5 equals 0 the percentage of frauds are a little bigger than the target(3.09%), and when the value is higher than 0 is just about 0.86%

C6

```
ggplot(ds) + aes(x=C6) + geom_density(alpha=0.4)
```



```
ggplot(filter(ds, C6<=10)) + aes(x=C6, fill=isFraud==1) +
  geom_density(alpha=0.4) +
  scale_x_continuous(breaks = seq(0,10,1))
```



```
nrow(filter(ds, C6<=4)) / nrow(ds) * 100
```

```
## [1] 90.22667
```

```
nrow(filter(ds, C6<=4 & isFraud==1)) / nrow(filter(ds, C6<=4)) * 100
```

```
## [1] 2.859465
```

```
nrow(filter(ds, C6==0)) / nrow(ds) * 100
```

```
## [1] 8.293333
```

```
nrow(filter(ds, C6==0 & isFraud==1)) / nrow(filter(ds, C6==0)) * 100
```

```
## [1] 2.250804
```

```
nrow(filter(ds, C6==1)) / nrow(ds) * 100
```

```
## [1] 59.93333
```

```
nrow(filter(ds, C6==1 & isFraud==1)) / nrow(filter(ds, C6==1)) * 100
```

```
## [1] 2.547275
```

```
nrow(filter(ds, C6==2)) / nrow(ds) * 100
```

```
## [1] 13.79333
```

```
nrow(filter(ds, C6==2 & isFraud==1)) / nrow(filter(ds, C6==2)) * 100
```

```
## [1] 3.673272
```

```
nrow(filter(ds, C6==3)) / nrow(ds) * 100
```

```
## [1] 5.353333
```

```
nrow(filter(ds, C6==3 & isFraud==1)) / nrow(filter(ds, C6==3)) * 100
```

```
## [1] 3.985056
```

```
nrow(filter(ds, C6==4)) / nrow(ds) * 100
```

```
## [1] 2.853333
```

```
nrow(filter(ds, C6==4 & isFraud==1)) / nrow(filter(ds, C6==4)) * 100
```

```
## [1] 5.140187
```

```
nrow(filter(ds, C6>4)) / nrow(ds) * 100
```

```
## [1] 9.773333
```

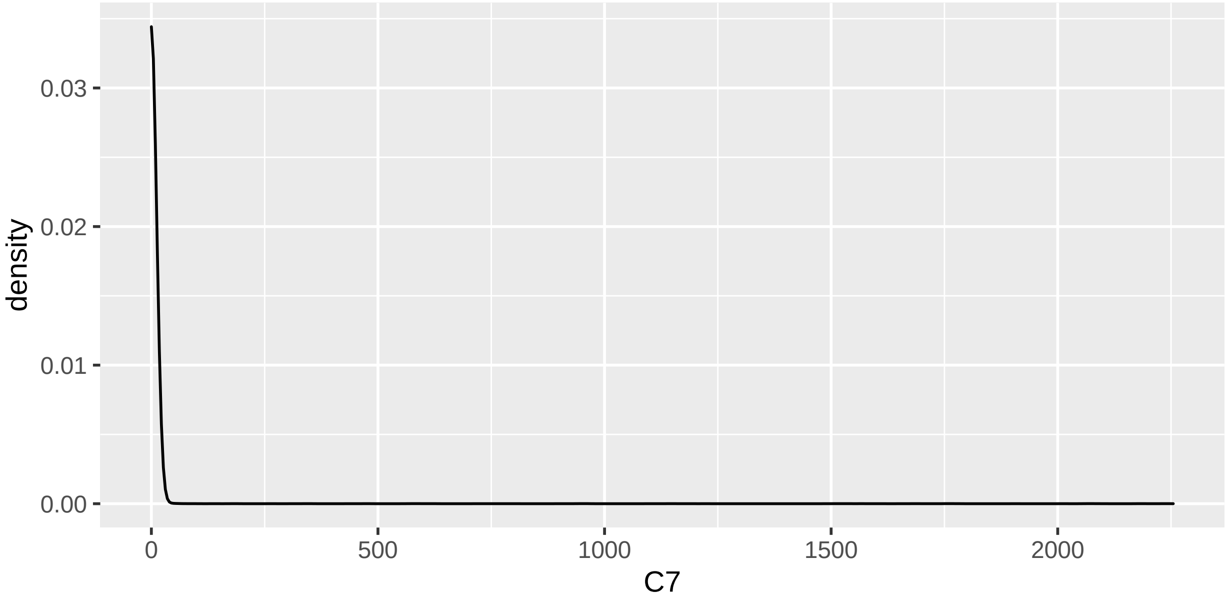
```
nrow(filter(ds, C6>4 & isFraud==1)) / nrow(filter(ds, C6>4)) * 100
```

```
## [1] 5.252387
```

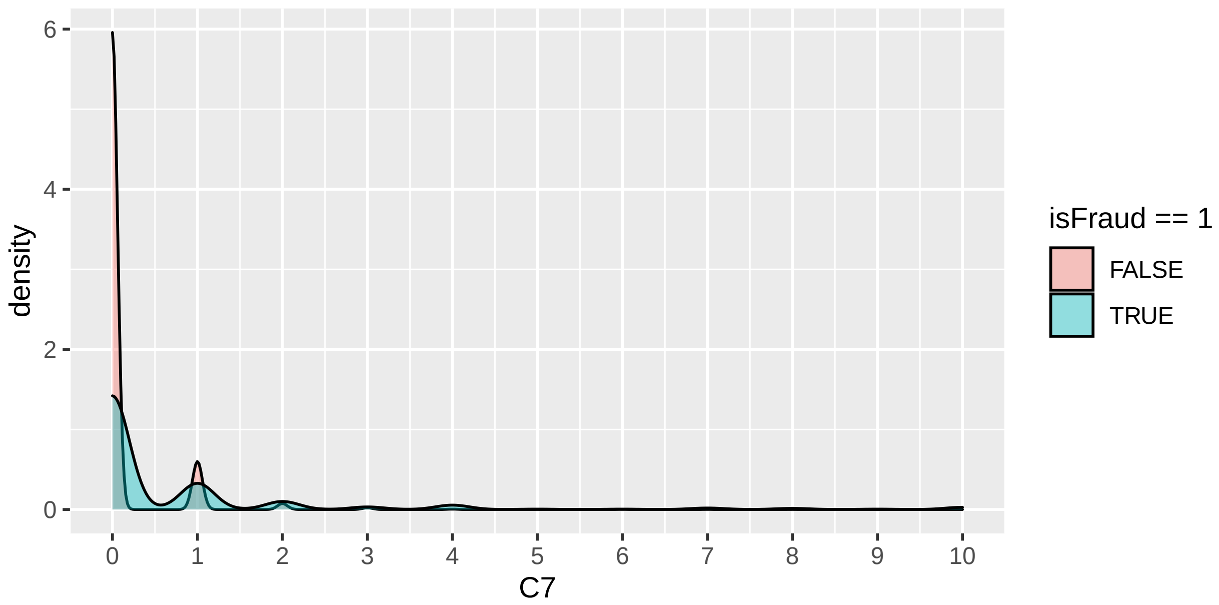
The mode of C6 is 1 instead of the other counting features that we've seen so far, but it still has a lot of cases with 0. When the values are 0 or 1 the percentage of fraudulent transactions are less than the target, but when we increase the value the percentage also increases.

C7

```
ggplot(ds) + aes(x=C7) + geom_density(alpha=0.4)
```



```
ggplot(filter(ds, C7<=10)) + aes(x=C7, fill=isFraud==1) +
  geom_density(alpha=0.4) +
  scale_x_continuous(breaks = seq(0,10,1))
```



```
nrow(filter(ds, C7<=4)) / nrow(ds) * 100
```

```
## [1] 99.2
```

```
nrow(filter(ds, C7<=4 & isFraud==1)) / nrow(filter(ds, C7<=4)) * 100
```

```
## [1] 2.856183
```

```
nrow(filter(ds, C7==0)) / nrow(ds) * 100
```

```
## [1] 88.44667
```

```
nrow(filter(ds, C7==0 & isFraud==1)) / nrow(filter(ds, C7==0)) * 100
```

```
## [1] 2.3517
```

```
nrow(filter(ds, C7==1)) / nrow(ds) * 100
```

```
## [1] 9.053333
```

```
nrow(filter(ds, C7==1 & isFraud==1)) / nrow(filter(ds, C7==1)) * 100
```

```
## [1] 5.301915
```

```
nrow(filter(ds, C7==2)) / nrow(ds) * 100
```

```
## [1] 1.206667
```

```
nrow(filter(ds, C7==2 & isFraud==1)) / nrow(filter(ds, C7==2)) * 100
```

```
## [1] 12.1547
```

```
nrow(filter(ds, C7==3)) / nrow(ds) * 100
```

```
## [1] 0.3466667
```

```
nrow(filter(ds, C7==3 & isFraud==1)) / nrow(filter(ds, C7==3)) * 100
```

```
## [1] 13.46154
```

```
nrow(filter(ds, C7==4)) / nrow(ds) * 100
```

```
## [1] 0.1466667
```

```
nrow(filter(ds, C7==4 & isFraud==1)) / nrow(filter(ds, C7==4)) * 100
```

```
## [1] 54.54545
```

```
nrow(filter(ds, C7>4)) / nrow(ds) * 100
```

```
## [1] 0.8
```

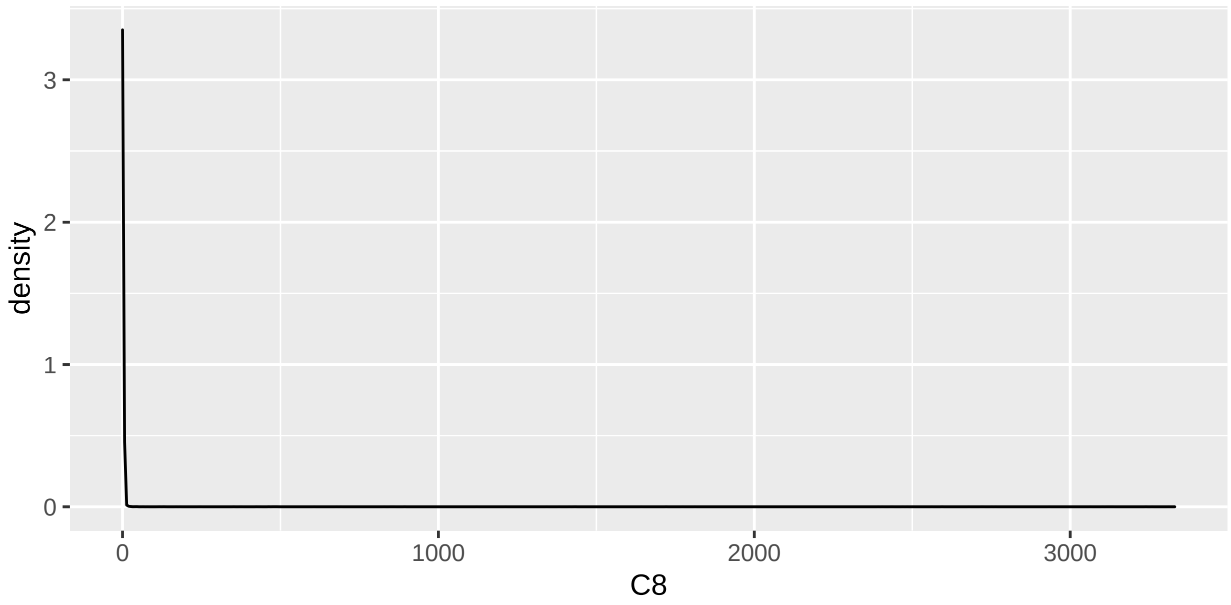
```
nrow(filter(ds, C7>4 & isFraud==1)) / nrow(filter(ds, C7>4)) * 100
```

```
## [1] 32.5
```

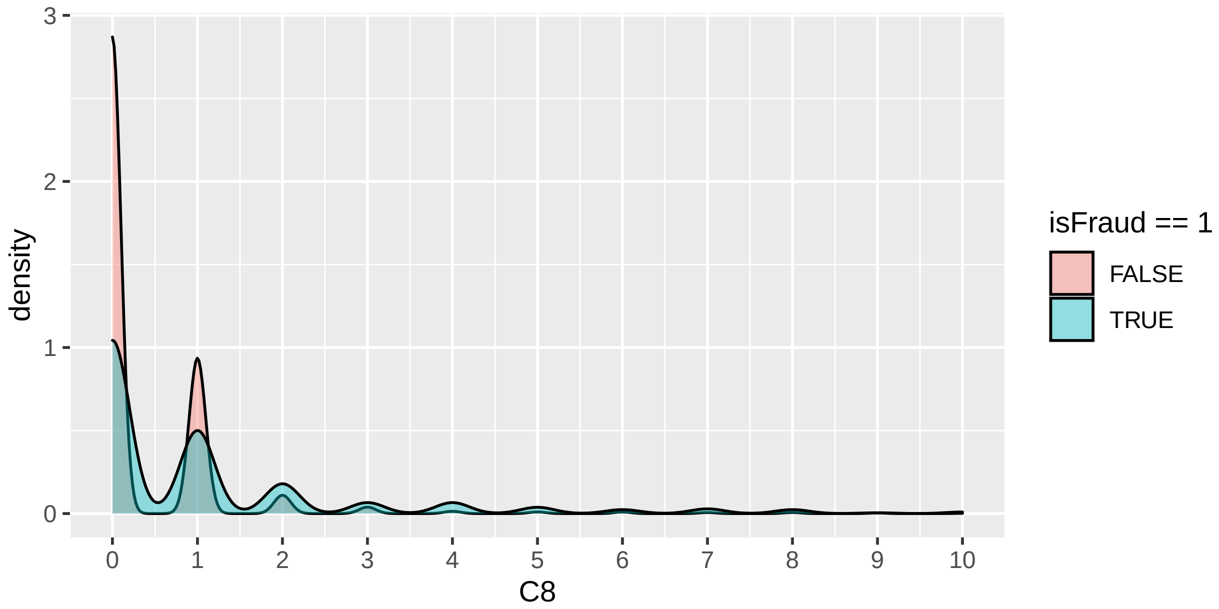
When C7 is equal to 4, the percentage of frauds is 54% which is a lot, but C4 is only 4 in a few cases in total(0.15%).

C8

```
ggplot(ds) + aes(x=C8) + geom_density(alpha=0.4)
```



```
ggplot(filter(ds, C8<=10)) + aes(x=C8, fill=isFraud==1) +
  geom_density(alpha=0.4) +
  scale_x_continuous(breaks = seq(0,10,1))
```



```
nrow(filter(ds, C8<=10)) / nrow(ds) * 100
```

```
## [1] 99.01333
```

```
nrow(filter(ds, C8<=10 & isFraud==1)) / nrow(filter(ds, C8<=10)) * 100
```

```
## [1] 2.827902
```

```
nrow(filter(ds, C8==0)) / nrow(ds) * 100
```

```
## [1] 70.54667
```

```
nrow(filter(ds, C8==0 & isFraud==1)) / nrow(filter(ds, C8==0)) * 100
```

```
## [1] 2.088452
```

```
nrow(filter(ds, C8==1)) / nrow(ds) * 100
```

```
## [1] 23.07333
```

```
nrow(filter(ds, C8==1 & isFraud==1)) / nrow(filter(ds, C8==1)) * 100
```

```
## [1] 3.062699
```

```
nrow(filter(ds, C8==2)) / nrow(ds) * 100
```

```
## [1] 2.913333
```

```
nrow(filter(ds, C8==2 & isFraud==1)) / nrow(filter(ds, C8==2)) * 100
```

```
## [1] 8.695652
```

```
nrow(filter(ds, C8==3)) / nrow(ds) * 100
```

```
## [1] 1.026667
```

```
nrow(filter(ds, C8==3 & isFraud==1)) / nrow(filter(ds, C8==3)) * 100
```

```
## [1] 9.090909
```

```
nrow(filter(ds, C8==4)) / nrow(ds) * 100
```

```
## [1] 0.42
```

```
nrow(filter(ds, C8==4 & isFraud==1)) / nrow(filter(ds, C8==4)) * 100
```

```
## [1] 22.22222
```

```
nrow(filter(ds, C8>10)) / nrow(ds) * 100
```

```
## [1] 0.9866667
```

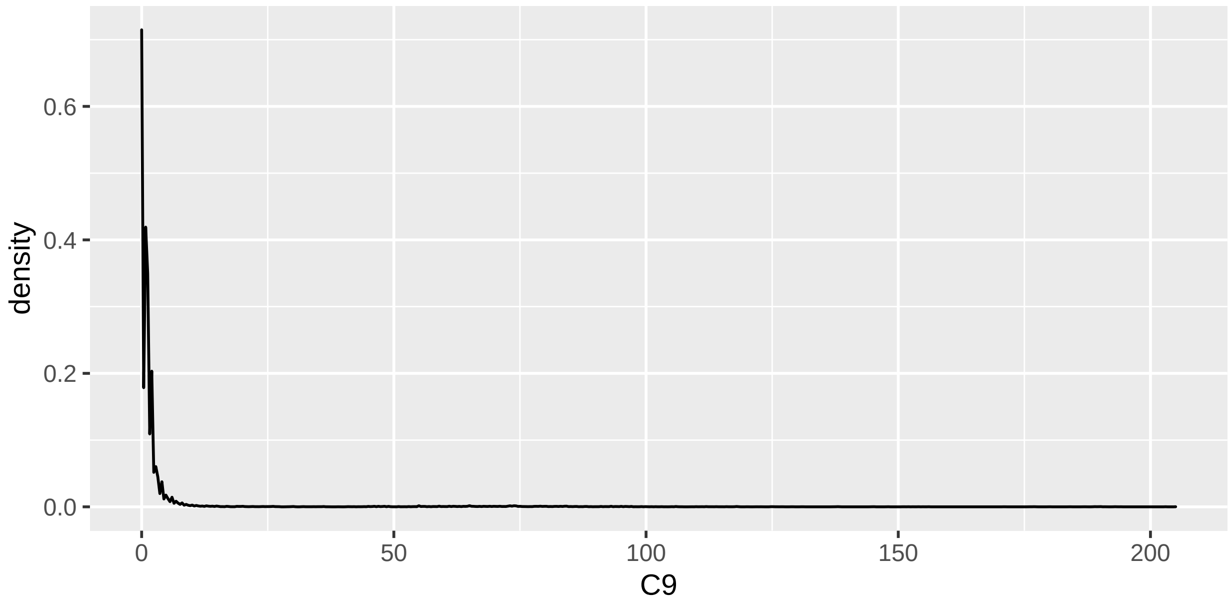
```
nrow(filter(ds, C8>10 & isFraud==1)) / nrow(filter(ds, C8>10)) * 100
```

```
## [1] 29.72973
```

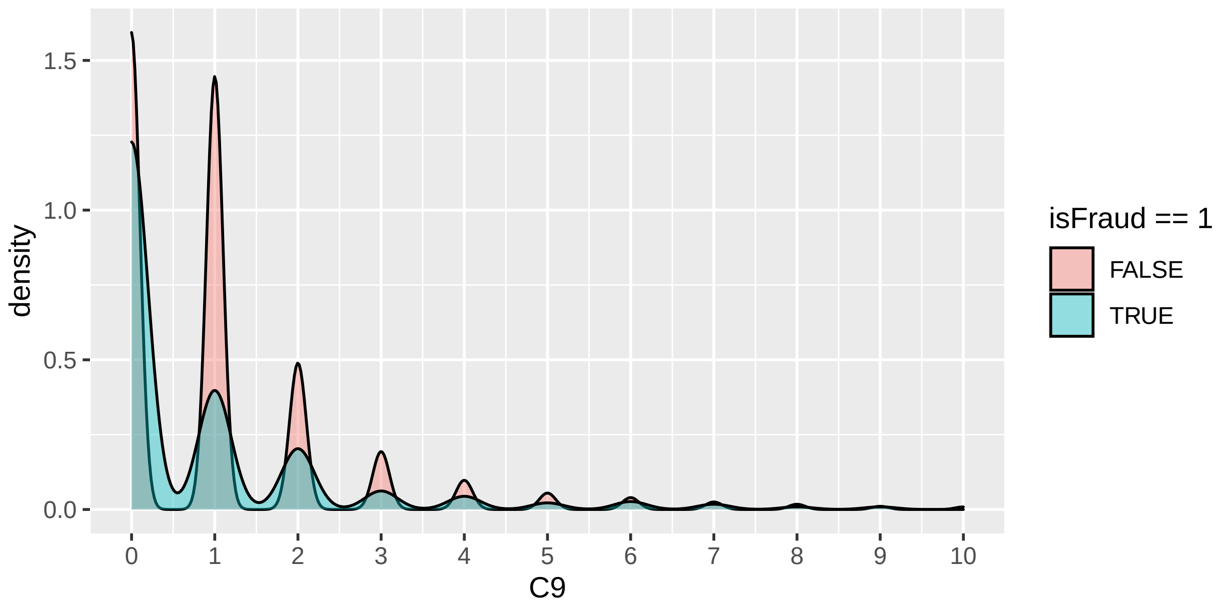
The values 0 and 1 are the ones that appear the most in feature C8, and the percentage of frauds are less than the target.

C9

```
ggplot(ds) + aes(x=C9) + geom_density(alpha=0.4)
```



```
ggplot(filter(ds, C9<=10)) + aes(x=C9, fill=isFraud==1) +
  geom_density(alpha=0.4) +
  scale_x_continuous(breaks = seq(0,10,1))
```



```
nrow(filter(ds, C9<=10)) / nrow(ds) * 100
```

```
## [1] 95.56
```

```
nrow(filter(ds, C9<=10 & isFraud==1)) / nrow(filter(ds, C9<=10)) * 100
```

```
## [1] 3.188224
```

```
nrow(filter(ds, C9==0)) / nrow(ds) * 100
```

```
## [1] 38.87333
```

```
nrow(filter(ds, C9==0 & isFraud==1)) / nrow(filter(ds, C9==0)) * 100
```

```
## [1] 4.767621
```

```
nrow(filter(ds, C9==1)) / nrow(ds) * 100
```

```
## [1] 34.22667
```

```
nrow(filter(ds, C9==1 & isFraud==1)) / nrow(filter(ds, C9==1)) * 100
```

```
## [1] 1.753019
```

```
nrow(filter(ds, C9==2)) / nrow(ds) * 100
```

```
## [1] 11.74
```

```
nrow(filter(ds, C9==2 & isFraud==1)) / nrow(filter(ds, C9==2)) * 100
```

```
## [1] 2.612152
```

```
nrow(filter(ds, C9==3)) / nrow(ds) * 100
```

```
## [1] 4.606667
```

```
nrow(filter(ds, C9==3 & isFraud==1)) / nrow(filter(ds, C9==3)) * 100
```

```
## [1] 2.026049
```

```
nrow(filter(ds, C9==4)) / nrow(ds) * 100
```

```
## [1] 2.34
```

```
nrow(filter(ds, C9==4 & isFraud==1)) / nrow(filter(ds, C9==4)) * 100
```

```
## [1] 2.849003
```

```
nrow(filter(ds, C9>4)) / nrow(ds) * 100
```

```
## [1] 8.213333
```

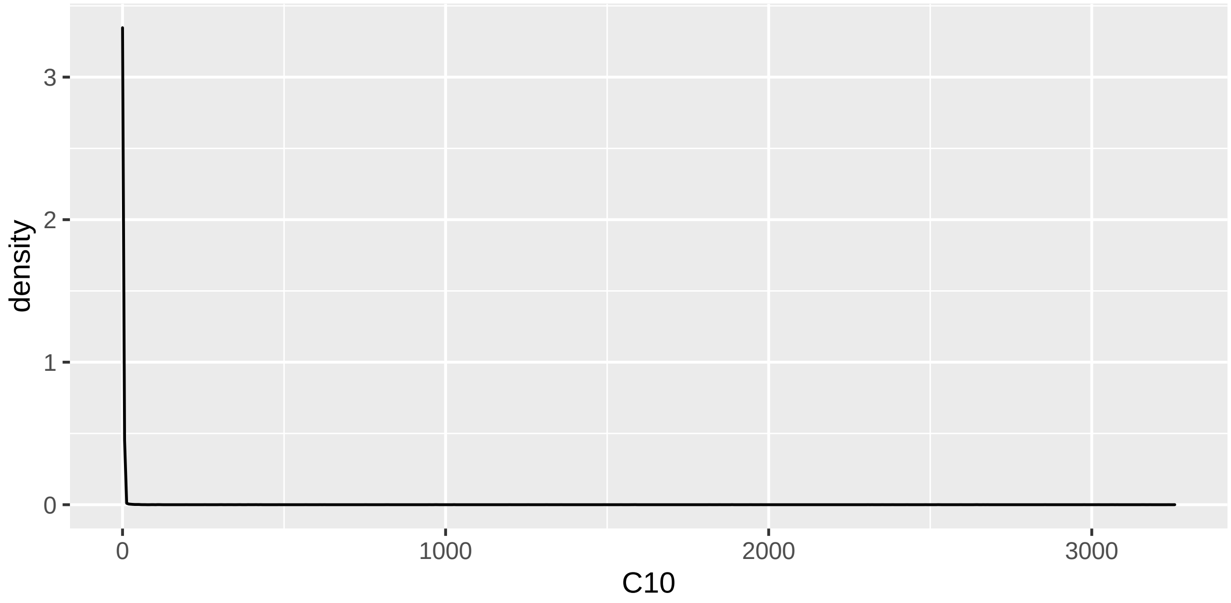
```
nrow(filter(ds, C9>4 & isFraud==1)) / nrow(filter(ds, C9>4)) * 100
```

```
## [1] 2.11039
```

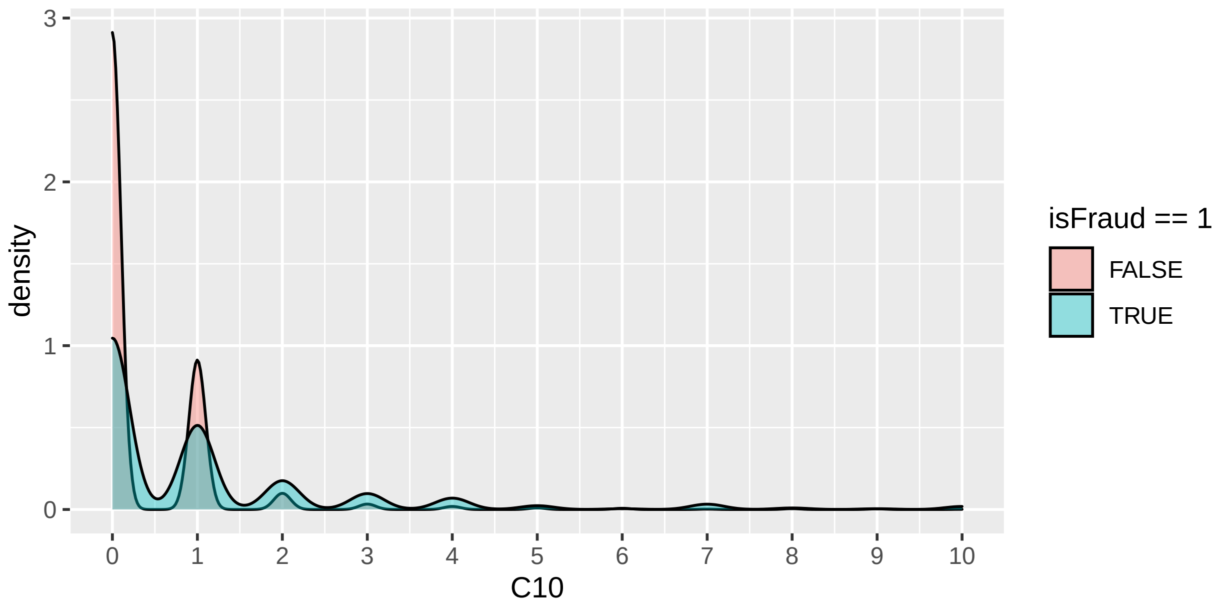
C9 has the values more scattered, when the value is 0, 39% of the times, the percentage of fraud is higher than the target, which can mean that this feature has an important role in the prediction of fraudulent transactions.

C10

```
ggplot(ds) + aes(x=C10) + geom_density(alpha=0.4)
```



```
ggplot(filter(ds, C10<=10)) + aes(x=C10, fill=isFraud==1) +
  geom_density(alpha=0.4) +
  scale_x_continuous(breaks = seq(0,10,1))
```



```
nrow(filter(ds, C10<=10)) / nrow(ds) * 100
```

```
## [1] 98.92667
```

```
nrow(filter(ds, C10<=10 & isFraud==1)) / nrow(filter(ds, C10<=10)) * 100
```

```
## [1] 2.904508
```

```
nrow(filter(ds, C10==0)) / nrow(ds) * 100
```

```
## [1] 71.48667
```

```
nrow(filter(ds, C10==0 & isFraud==1)) / nrow(filter(ds, C10==0)) * 100
```

```
## [1] 2.107619
```

```
nrow(filter(ds, C10==1)) / nrow(ds) * 100
```

```
## [1] 22.52
```

```
nrow(filter(ds, C10==1 & isFraud==1)) / nrow(filter(ds, C10==1)) * 100
```

```
## [1] 3.285968
```

```
nrow(filter(ds, C10==2)) / nrow(ds) * 100
```

```
## [1] 2.633333
```

```
nrow(filter(ds, C10==2 & isFraud==1)) / nrow(filter(ds, C10==2)) * 100
```

```
## [1] 9.620253
```

```
nrow(filter(ds, C10==3)) / nrow(ds) * 100
```

```
## [1] 0.9333333
```

```
nrow(filter(ds, C10==3 & isFraud==1)) / nrow(filter(ds, C10==3)) * 100
```

```
## [1] 15
```

```
nrow(filter(ds, C10==4)) / nrow(ds) * 100
```

```
## [1] 0.5466667
```

```
nrow(filter(ds, C10==4 & isFraud==1)) / nrow(filter(ds, C10==4)) * 100
```

```
## [1] 18.29268
```

```
nrow(filter(ds, C10>4)) / nrow(ds) * 100
```

```
## [1] 1.88
```

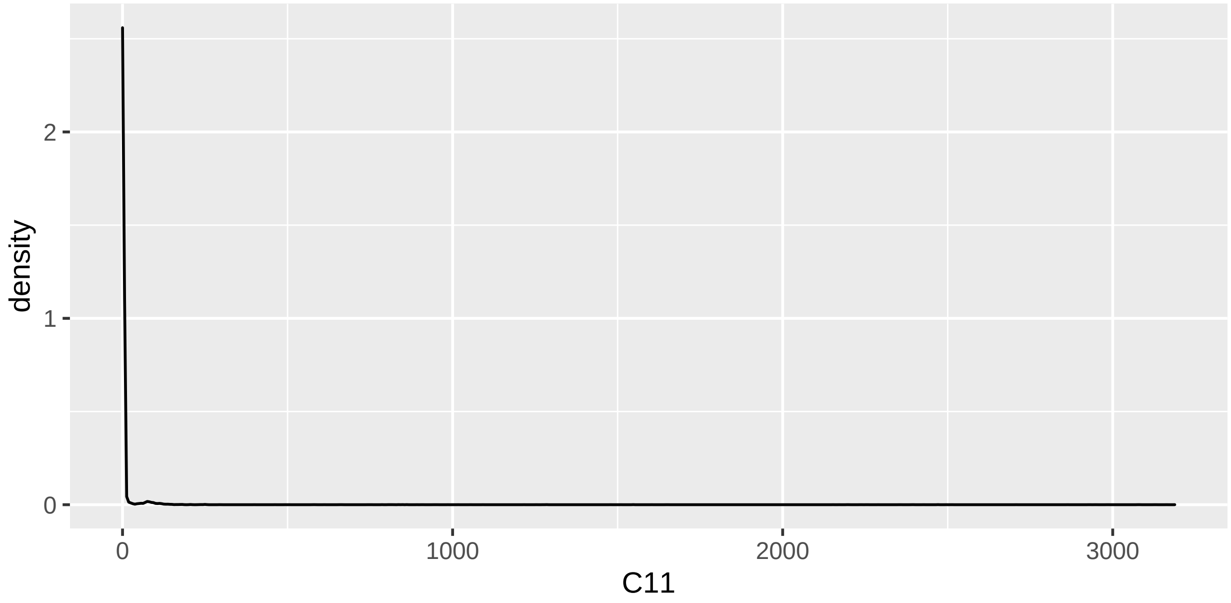
```
nrow(filter(ds, C10>4 & isFraud==1)) / nrow(filter(ds, C10>4)) * 100
```

```
## [1] 18.79433
```

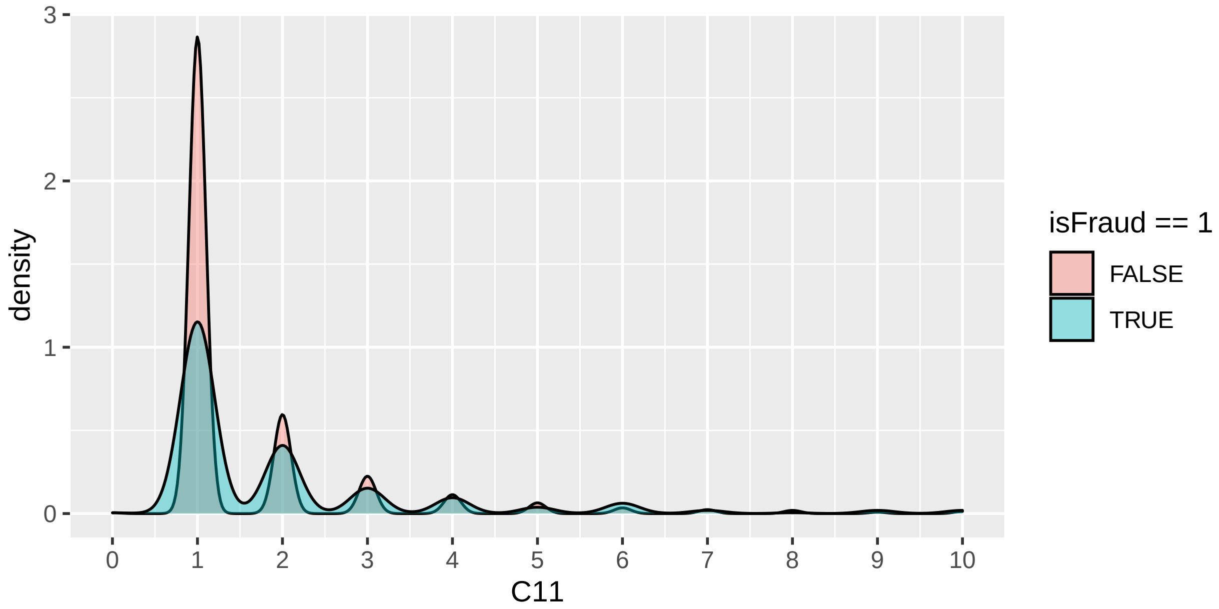
The mode of C10 is 0, and it's where the percentage of fraudulent transactions are less than the target. There is 22% of the cases where the value is 1 and the percentage of frauds is slightly higher than in the target. In the other values even though they represent less cases, the percentage of fraud is higher.

C11

```
ggplot(ds) + aes(x=C11) + geom_density(alpha=0.4)
```



```
ggplot(filter(ds, C11<=10)) + aes(x=C11, fill=isFraud==1) +
  geom_density(alpha=0.4) +
  scale_x_continuous(breaks = seq(0,10,1))
```



```
nrow(filter(ds, C11<=10)) / nrow(ds) * 100
```

```
## [1] 94.64
```

```
nrow(filter(ds, C11<=10 & isFraud==1)) / nrow(filter(ds, C11<=10)) * 100
```

```
## [1] 2.923359
```

```
nrow(filter(ds, C11==0)) / nrow(ds) * 100
```

```
## [1] 0.12
```

```
nrow(filter(ds, C11==0 & isFraud==1)) / nrow(filter(ds, C11==0)) * 100
```

```
## [1] 5.555556
```

```
nrow(filter(ds, C11==1)) / nrow(ds) * 100
```

```
## [1] 67.92
```

```
nrow(filter(ds, C11==1 & isFraud==1)) / nrow(filter(ds, C11==1)) * 100
```

```
## [1] 2.375344
```

```
nrow(filter(ds, C11==2)) / nrow(ds) * 100
```

```
## [1] 14.41333
```

```
nrow(filter(ds, C11==2 & isFraud==1)) / nrow(filter(ds, C11==2)) * 100
```

```
## [1] 3.977798
```

```
nrow(filter(ds, C11==3)) / nrow(ds) * 100
```

```
## [1] 5.426667
```

```
nrow(filter(ds, C11==3 & isFraud==1)) / nrow(filter(ds, C11==3)) * 100
```

```
## [1] 3.931204
```

```
nrow(filter(ds, C11==4)) / nrow(ds) * 100
```

```
## [1] 2.766667
```

```
nrow(filter(ds, C11==4 & isFraud==1)) / nrow(filter(ds, C11==4)) * 100
```

```
## [1] 4.819277
```

```
nrow(filter(ds, C11>4)) / nrow(ds) * 100
```

```
## [1] 9.353333
```

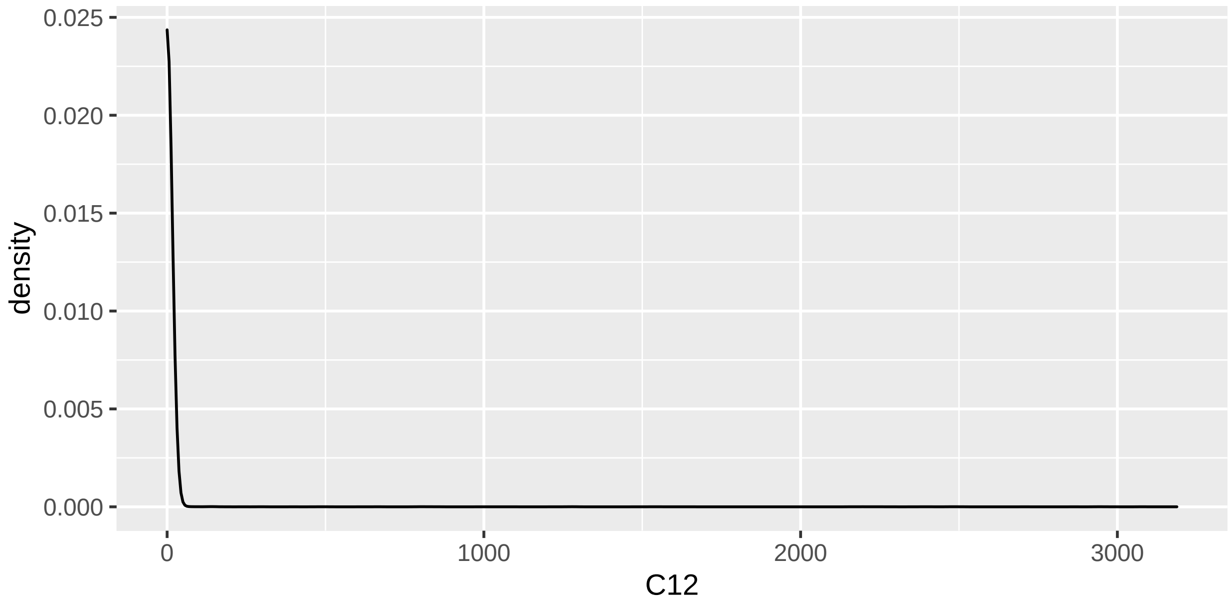
```
nrow(filter(ds, C11>4 & isFraud==1)) / nrow(filter(ds, C11>4)) * 100
```

```
## [1] 5.915895
```

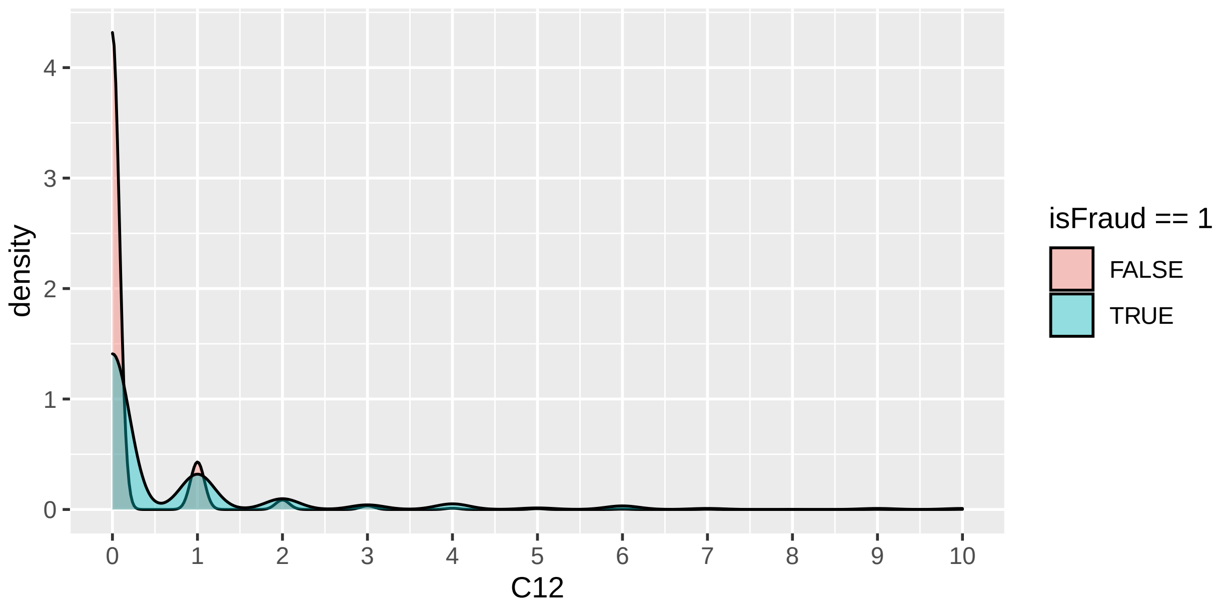
This feature is almost never 0 and its mode is 1 (68%) with the percentage of fraud being less than the target, the value 2 with 14.4% of the cases has a percentage of fraudulent transactions higher than the target.

C12

```
ggplot(ds) + aes(x=C12) + geom_density(alpha=0.4)
```



```
ggplot(filter(ds, C12<=10)) + aes(x=C12, fill=isFraud==1) +
  geom_density(alpha=0.4) +
  scale_x_continuous(breaks = seq(0,10,1))
```



```
nrow(filter(ds, C12<=10)) / nrow(ds) * 100
```

```
## [1] 99.32
```

```
nrow(filter(ds, C12<=10 & isFraud==1)) / nrow(filter(ds, C12<=10)) * 100
```

```
## [1] 2.886293
```

```
nrow(filter(ds, C12==0)) / nrow(ds) * 100
```

```
## [1] 87.12667
```

```
nrow(filter(ds, C12==0 & isFraud==1)) / nrow(filter(ds, C12==0)) * 100
```

```
## [1] 2.326115
```

```
nrow(filter(ds, C12==1)) / nrow(ds) * 100
```

```
## [1] 8.94
```

```
nrow(filter(ds, C12==1 & isFraud==1)) / nrow(filter(ds, C12==1)) * 100
```

```
## [1] 5.145414
```

```
nrow(filter(ds, C12==2)) / nrow(ds) * 100
```

```
## [1] 1.826667
```

```
nrow(filter(ds, C12==2 & isFraud==1)) / nrow(filter(ds, C12==2)) * 100
```

```
## [1] 7.664234
```

```
nrow(filter(ds, C12==3)) / nrow(ds) * 100
```

```
## [1] 0.7133333
```

```
nrow(filter(ds, C12==3 & isFraud==1)) / nrow(filter(ds, C12==3)) * 100
```

```
## [1] 8.411215
```

```
nrow(filter(ds, C12==4)) / nrow(ds) * 100
```

```
## [1] 0.2933333
```

```
nrow(filter(ds, C12==4 & isFraud==1)) / nrow(filter(ds, C12==4)) * 100
```

```
## [1] 25
```

```
nrow(filter(ds, C12>4)) / nrow(ds) * 100
```

```
## [1] 1.1
```

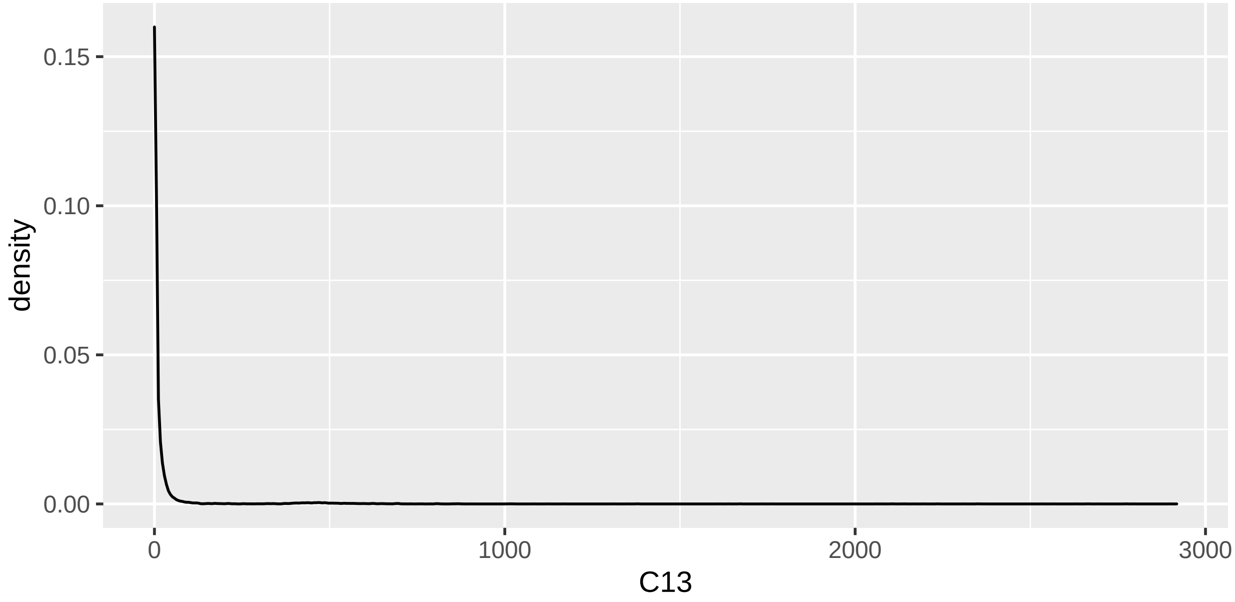
```
nrow(filter(ds, C12>4 & isFraud==1)) / nrow(filter(ds, C12>4)) * 100
```

```
## [1] 30.30303
```

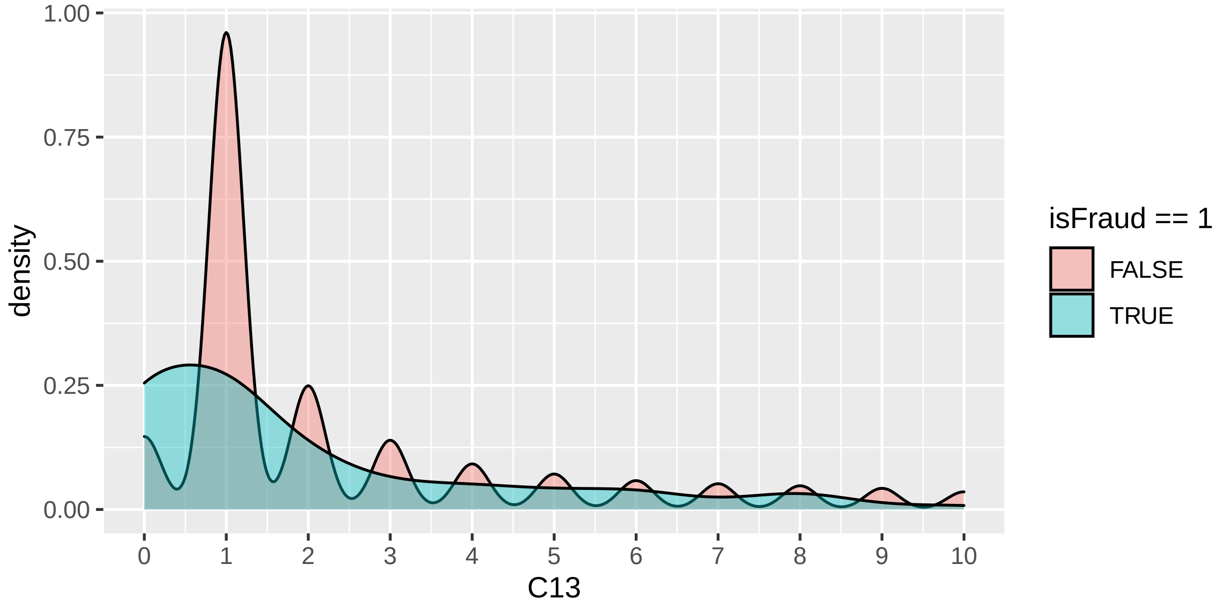
Value 1 in this feature represents 9% of the cases and its 5.15% of the cases are fraudulent.

C13

```
ggplot(ds) + aes(x=C13) + geom_density(alpha=0.4)
```



```
ggplot(filter(ds, C13<=10)) + aes(x=C13, fill=isFraud==1) +
  geom_density(alpha=0.4) +
  scale_x_continuous(breaks = seq(0,10,1))
```



```
nrow(filter(ds, C13<=10)) / nrow(ds) * 100
```

```
## [1] 74.12
```

```
nrow(filter(ds, C13<=10 & isFraud==1)) / nrow(filter(ds, C13<=10)) * 100
```

```
## [1] 3.579781
```

```
nrow(filter(ds, C13==0)) / nrow(ds) * 100
```

```
## [1] 6.36
```

```
nrow(filter(ds, C13==0 & isFraud==1)) / nrow(filter(ds, C13==0)) * 100
```

```
## [1] 12.89308
```

```
nrow(filter(ds, C13==1)) / nrow(ds) * 100
```

```
## [1] 37.04667
```

```
nrow(filter(ds, C13==1 & isFraud==1)) / nrow(filter(ds, C13==1)) * 100
```

```
## [1] 2.177434
```

```
nrow(filter(ds, C13==2)) / nrow(ds) * 100
```

```
## [1] 9.7
```

```
nrow(filter(ds, C13==2 & isFraud==1)) / nrow(filter(ds, C13==2)) * 100
```

```
## [1] 3.230241
```

```
nrow(filter(ds, C13==3)) / nrow(ds) * 100
```

```
## [1] 5.4
```

```
nrow(filter(ds, C13==3 & isFraud==1)) / nrow(filter(ds, C13==3)) * 100
```

```
## [1] 2.716049
```

```
nrow(filter(ds, C13==4)) / nrow(ds) * 100
```

```
## [1] 3.6
```

```
nrow(filter(ds, C13==4 & isFraud==1)) / nrow(filter(ds, C13==4)) * 100
```

```
## [1] 3.888889
```

```
nrow(filter(ds, C13>10)) / nrow(ds) * 100
```

```
## [1] 25.88
```

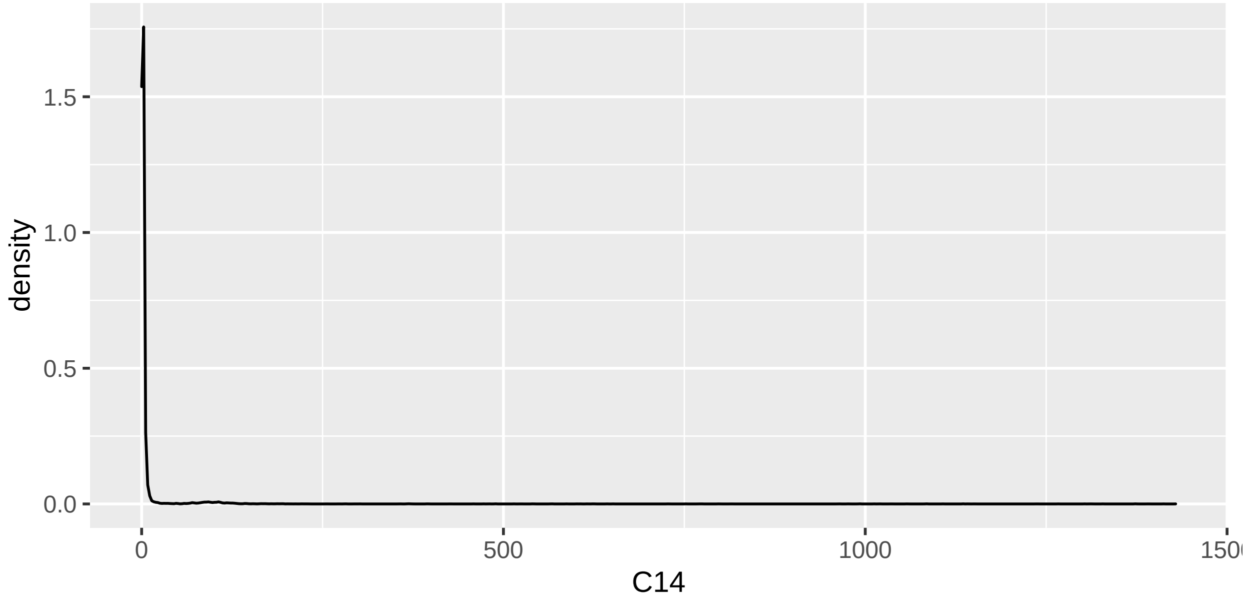
```
nrow(filter(ds, C13>10 & isFraud==1)) / nrow(filter(ds, C13>10)) * 100
```

```
## [1] 1.700155
```

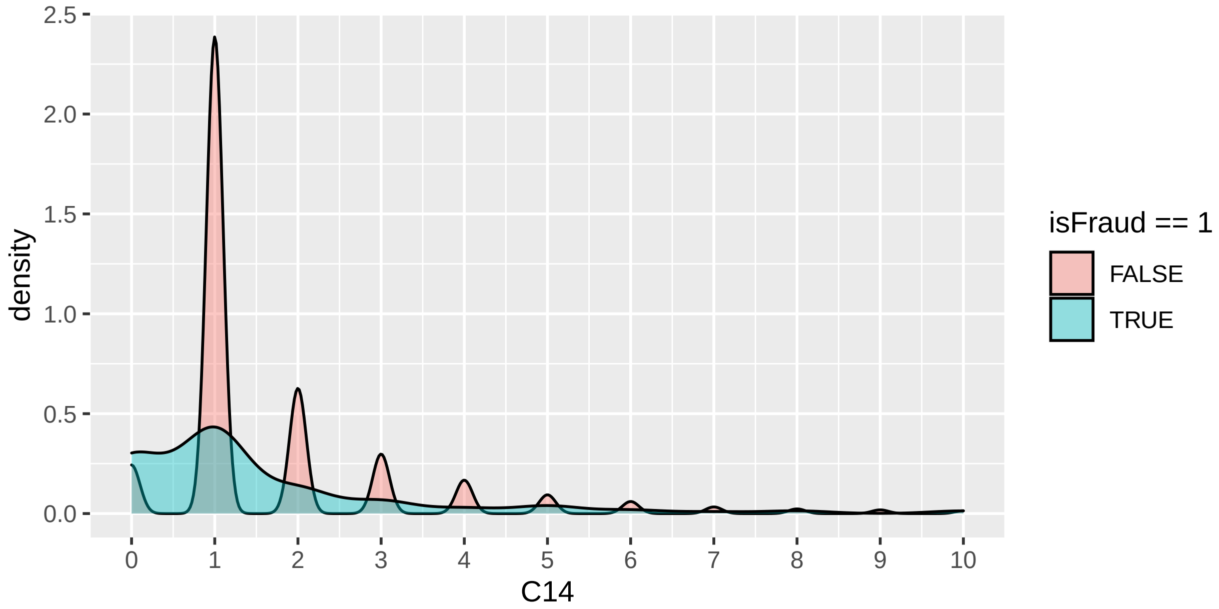
C13 has a better distribution than the others.

C14

```
ggplot(ds) + aes(x=C14) + geom_density(alpha=0.4)
```



```
ggplot(filter(ds, C14<=10)) + aes(x=C14, fill=isFraud==1) +
  geom_density(alpha=0.4) +
  scale_x_continuous(breaks = seq(0,10,1))
```



```
nrow(filter(ds, C14<=10)) / nrow(ds) * 100
```

```
## [1] 94.54667
```

```
nrow(filter(ds, C14<=10 & isFraud==1)) / nrow(filter(ds, C14<=10)) * 100
```

```
## [1] 3.151883
```

```
nrow(filter(ds, C14==0)) / nrow(ds) * 100
```

```
## [1] 6.446667
```

```
nrow(filter(ds, C14==0 & isFraud==1)) / nrow(filter(ds, C14==0)) * 100
```

```
## [1] 13.1334
```

```
nrow(filter(ds, C14==1)) / nrow(ds) * 100
```

```
## [1] 56.30667
```

```
nrow(filter(ds, C14==1 & isFraud==1)) / nrow(filter(ds, C14==1)) * 100
```

```
## [1] 2.190386
```

```
nrow(filter(ds, C14==2)) / nrow(ds) * 100
```

```
## [1] 14.88667
```

```
nrow(filter(ds, C14==2 & isFraud==1)) / nrow(filter(ds, C14==2)) * 100
```

```
## [1] 2.418271
```

```
nrow(filter(ds, C14==3)) / nrow(ds) * 100
```

```
## [1] 7.073333
```

```
nrow(filter(ds, C14==3 & isFraud==1)) / nrow(filter(ds, C14==3)) * 100
```

```
## [1] 2.63902
```

```
nrow(filter(ds, C14==4)) / nrow(ds) * 100
```

```
## [1] 3.946667
```

```
nrow(filter(ds, C14==4 & isFraud==1)) / nrow(filter(ds, C14==4)) * 100
```

```
## [1] 2.027027
```

```
nrow(filter(ds, C14>4)) / nrow(ds) * 100
```

```
## [1] 11.34
```

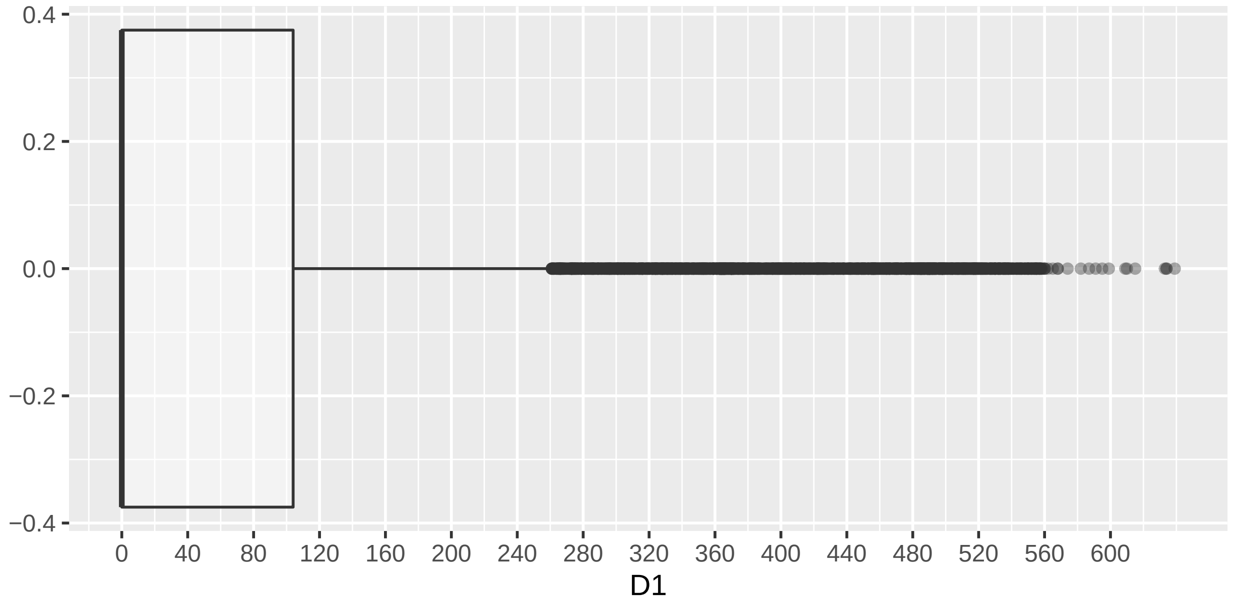
```
nrow(filter(ds, C14>4 & isFraud==1)) / nrow(filter(ds, C14>4)) * 100
```

```
## [1] 3.409759
```

C14 is similar to C13.

D1

```
ggplot(ds) + aes(x=D1) + geom_boxplot(alpha=0.4) +
  scale_x_continuous(breaks=seq(0,639,40))
```

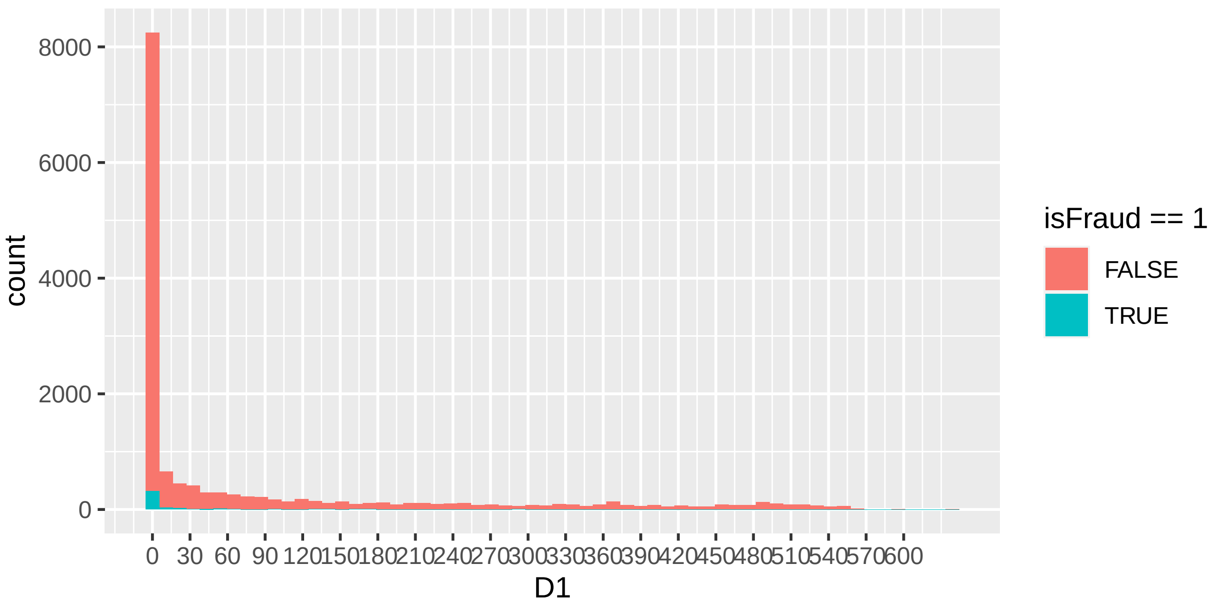


```
nrow(filter(ds, D1==0 & isFraud==1))
```

```
## [1] 261
```

We observe that there are 261 fraudulent transactions when D1 has a value of 0. More than 75% of the D1 values are between 0 and 100.

```
ggplot(ds) + aes(x=D1, fill=isFraud==1) + geom_histogram(bins = 60) +
  scale_x_continuous(breaks=seq(0,600,30))
```



Distribution of fraudulent and non-fraudulent transactions by the values of *D1*.

```
nrow(filter(ds, D1==0)) / nrow(ds) * 100
```

```
## [1] 51.04667
```

Although a maximum value of *D1* is 639, more than half of the D1 values are 0.

```
nrow(filter(ds, D1==0 & isFraud==1)) / nrow(filter(ds, D1==0)) * 100
```

```
## [1] 3.408646
```

```
nrow(filter(ds, D1>0 & isFraud==1)) / nrow(filter(ds, D1>0)) * 100
```

```
## [1] 2.764538
```

We observe that, the percentage of fraudulent transactions when the value of D1 is 0 is 3.4%. When the value of D1 is other than 0, the percentage of fraudulent transactions is 2.7% (much lower).