```
library(readr)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ purrr     1.0.2
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

```
library(dplyr)
library(lubridate)
library(ggplot2)
library(tinytex)
```

```
# Importing the Data

read_fraud <- function(wd = getwd()) {
  transactions <- list.files(path = wd, pattern = "*.csv", full.names = TRUE)

  frauddata <- lapply(transactions, function(file) {
    read_csv(file,
             col_names = TRUE,
             col_types = cols(
               CardType = col_factor(levels = c("Cr", "Dr")),
               Fraud = col_factor(levels = c("Yes", "No"))
             ))
  }) %>%
    bind_rows()

  return(frauddata)
}

transactions <- read_fraud("/Users/avaangeles/Downloads/FraudData")
transactions
```

```
## # A tibble: 5,408,478 × 4
##    TimeStamp           CardType Amount Fraud
##    <dttm>              <fct>     <dbl> <fct>
##  1 2023-01-01 00:00:05 Dr         640. No
##  2 2023-01-01 00:00:18 Cr        1500  Yes
##  3 2023-01-01 00:00:25 Cr        3822. No
##  4 2023-01-01 00:00:44 Cr        4850. No
##  5 2023-01-01 00:00:47 Dr        1500  Yes
##  6 2023-01-01 00:00:57 Dr         220. No
##  7 2023-01-01 00:00:59 Cr        1500  Yes
##  8 2023-01-01 00:01:05 Dr        3765. No
##  9 2023-01-01 00:01:24 Cr         765. No
## 10 2023-01-01 00:01:26 Cr        2594. No
## # ℹ 5,408,468 more rows
```

```
# Total loss of bank due to fraud

fraudt <- transactions %>% filter(Fraud == "Yes")
loss <-summarise(fraudt, "Total Loss" = sum(Amount, na.rm = TRUE))

print(loss)
```

```
## # A tibble: 1 × 1
##   `Total Loss`
##          <dbl>
## 1    39465197.
```

```
# The bank lost PHP 39,465,197 due to fraud transactions alone.
```

```
# Top 4 days with greatest number of fraudulent transactions
topdays <- fraudt %>%
  group_by(Date = as.Date(TimeStamp)) %>%
summarise(Fraud_Count = n()) %>%
  arrange(desc(Fraud_Count)) %>%
  slice_head(n=4)

print(topdays)
```

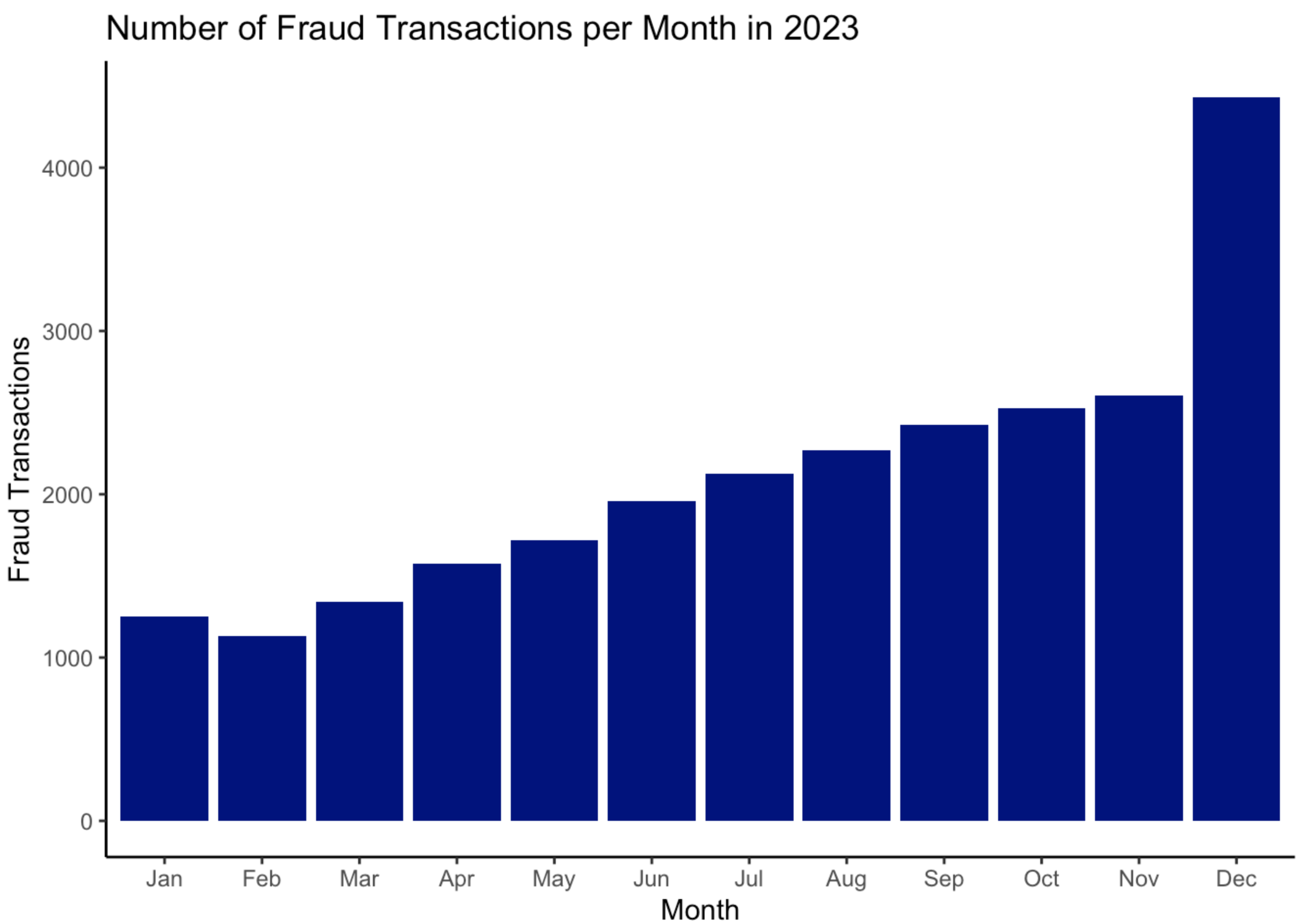
```
## # A tibble: 4 × 2
##   Date       Fraud_Count
##   <date>           <int>
## 1 2023-12-24         336
## 2 2023-12-31         333
## 3 2023-12-25         323
## 4 2023-01-01         219
```

```
# In 2023, the top four days with the most fraudulent transactions are December 24, Dece
mber 31, December 25, and January 1 respectively, which are all major holidays.
# This may be attributed to the increased spending during this season, so it is imperati
ve that fraud detection and prevention measures during the holidays, especially in Decem
ber.
# These may include requiring additional verification processes when making purchases, o
r to educate their customers on fraud tactics, whether through email or SNS.
```

```
# Visualization of number of fraud transactions per month

permonth <- transactions %>%
  filter(Fraud == "Yes") %>%
  mutate(Month = month(TimeStamp, label = TRUE)) %>%

  ggplot(aes(x = Month)) +
  geom_bar(stat = "count", fill = "navyblue") +
  labs(title = "Number of Fraud Transactions per Month in 2023",
       x = "Month",
       y = "Fraud Transactions") +
  theme_classic()
print(permonth)
```



```
# The Bar Chart shows that December had the greatest number of fraudulent transactions i
n 2023. Similar to my earlier recommendation, it is imperative that fraud detection and
prevention measures are heavily applied during this month.
```

```
# Comparison of Credit VS Debit Fraud
total_debit <- transactions %>% filter(CardType == "Dr")

total_credit <- transactions %>% filter(CardType == "Cr")

debit_fraud <- transactions %>%
  filter(Fraud == "Yes", CardType == "Dr") %>%
  nrow()/nrow(total_debit)

credit_fraud <- transactions %>%
  filter(Fraud == "Yes", CardType == "Cr") %>%
  nrow()/nrow(total_credit)

fraud_table <- data.frame(
  CardType = c("Debit", "Credit"),
  Probability = c(debit_fraud, credit_fraud)
)

print(fraud_table)
```

```
##   CardType Probability
## 1    Debit 0.006179416
## 2   Credit 0.001211453
```

```
# With a rate of 0.001%, credit cards are less prone to fraudulent transactions as compa
red to debit cards. Nonetheless, protection is imperative for all cards regardless, and
users of either of the two types must be made aware of common fraud tactic to limit frau
d overall.
```