Mai_Final

Duc Viet Mai

2022-10-29

Task 1: Abstract

The data set contains 41,188 observations of 21 banking customers' information variables. After reviewing each attribute closely and providing a summary of the chosen dataset, I concluded with four intriguing hypotheses regarding banking loaning activities:

1. Customers who currently study at basic education would borrow more than others in form of a student loan to go into university.
2. Customers who own a house would borrow more to pay a mortgage than those who rent outside.
3. Customers with low-paid jobs would likely to loan much more than those with high-paid jobs to pay for their increasing living expenses.
4. Consumer Price Index and Consumer Confidence Index are closely related and are indicators for changes in loaning activities.

```
df <- read.csv('https://raw.githubusercontent.com/ALY6000/datasets/main/kaggle/bankmarketing.csv')

# Installing packages in the console:
#install.packages("grid")
#install.packages("vcd")


# Calling libraries
library(grid)
library(vcd)

#Cleaning dataset
newdf <- data.frame(df$loan, df$housing, df$default,
                    df$age, df$job, df$marital,
                    df$education, df$cons.conf.idx,
                    df$cons.price.idx, df$nr.employed,
                    df$month, df$euribor3m,
                    df$day_of_week, df$duration, df$contact)
colnames(newdf) <- c("loan", "housing", "default",
                     "age", "job", "marital",
                     "education", "CPI",
                     "CCI", "years_of_emmployment",
                     "month", "eurobankrate",
                     "day_of_week","duration",
                     "contact")
newdf <- newdf[newdf$loan != "unknown"
```

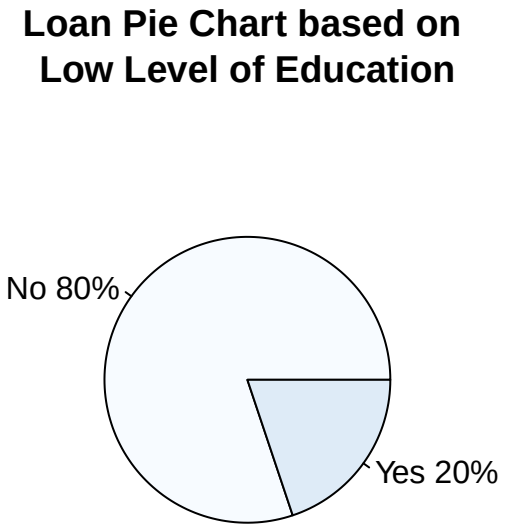
```
                & newdf$housing != "unknown"
                & newdf$default != "unknown"
                & newdf$job != "unknown"
                & newdf$marital != "unknown"
                & newdf$education != "unknown"
                & newdf$contact != "unknown",]

newdf$loan_code <- ifelse(newdf$loan == "yes", 1, 0)
newdf$years_of_emmployment <- newdf$years_of_emmployment/365

cleandf <- subset(newdf,newdf$age < quantile(newdf$age, 0.975)
                  & newdf$CPI < quantile(newdf$CPI, 0.975))
```

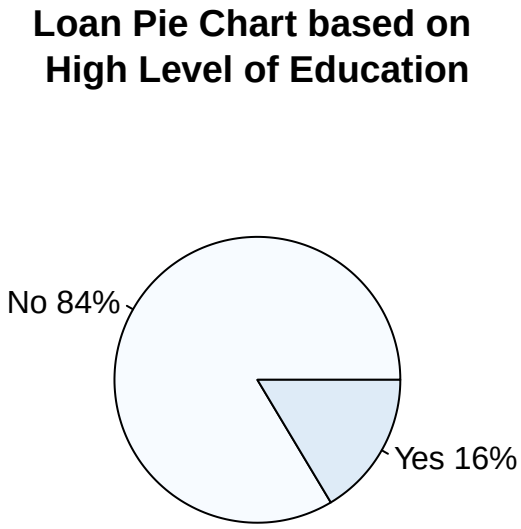
Task 2: Key findings

After having clean data set, I am slicing down the data by comparing the loaning activities based on education, job types, and accommodation. I have created three pie charts showing customers loaning percentage when they have a good education, high-paid jobs, and houses and when they have low education, low-paid jobs, and no accommodation. The results are fascinating when we find a 3-5 percent difference between these data groups. Compared to before cleaning the data set, the percentage difference is 2-4, one percentage smaller than after cleaning.

1. Undoubtedly, lower level of education customers will likely loan more because they want to pay for their future university tuition, indicating that the first hypothesis might be true.

```
# Before cleaning
newdf_edu1 <- newdf[newdf$age < 30
                  &newdf$education == c("basic.4y","basic.6y"),]

newdf_edu2 <- newdf[newdf$age < 30
                   &newdf$education == c("university.degree","professional.course"),]

opar <- par(mfrow=c(1,2))

x <- c(sum(newdf_edu1$loan_code == 0), sum(newdf_edu1$loan_code == 1))
label <- c("No","Yes")
pct <- round(x/sum(x)*100)
label <- paste(label, pct)
label <- paste(label,"%",sep = "")
pie(x, labels = label, col = blues9,
      main="Loan Pie Chart based on \nLow Level of Education")

x0 <- c(sum(newdf_edu2$loan_code == 0), sum(newdf_edu2$loan_code == 1))
label0 <- c("No","Yes")
pct0 <- round(x0/sum(x0)*100)
label0 <- paste(label0, pct0)
label0 <- paste(label0,"%",sep = "")
pie(x0, labels = label0, col = blues9,
      main="Loan Pie Chart based on \nHigh Level of Education")
mtext("Before Cleansing Data", outer = TRUE, line =-22, col = "red")
```

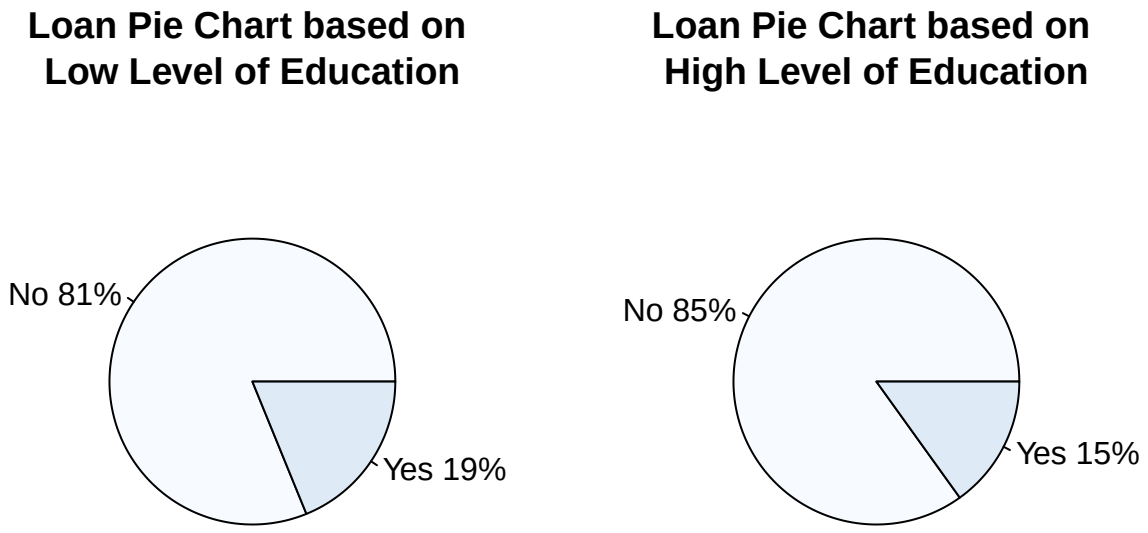




Before Cleansing Data

```
par(opar)

# After cleaning
opar4 <- par(mfrow=c(1,2))

cleandf_edu1 <- cleandf[cleandf$age < 30
                        &cleandf$education == c("basic.4y","basic.6y")
                    ,]
cx <- c(sum(cleandf_edu1$loan_code == 0), sum(cleandf_edu1$loan_code == 1))
clabel <- c("No","Yes")
cpct <- round(cx/sum(cx)*100)
clabel <- paste(clabel, cpct)
clabel <- paste(clabel,"%",sep = "")
pie(cx, labels = clabel, col = blues9,
    main = "Loan Pie Chart based on \nLow Level of Education")

cleandf_edu2 <- cleandf[cleandf$age < 30
                     &cleandf$education == c("university.degree","professional.course")
                     ,]
cx0 <- c(sum(cleandf_edu2$loan_code == 0), sum(cleandf_edu2$loan_code == 1))
clabel0 <- c("No","Yes")
cpct0 <- round(cx0/sum(cx0)*100)
clabel0 <- paste(clabel0, cpct0)
clabel0 <- paste(clabel0,"%",sep = "")
pie(cx0, labels = clabel0, col = blues9,
    main = "Loan Pie Chart based on \nHigh Level of Education")
```

```
mtext("After Cleansing Data", outer = TRUE, line =-22, col = "red")
```



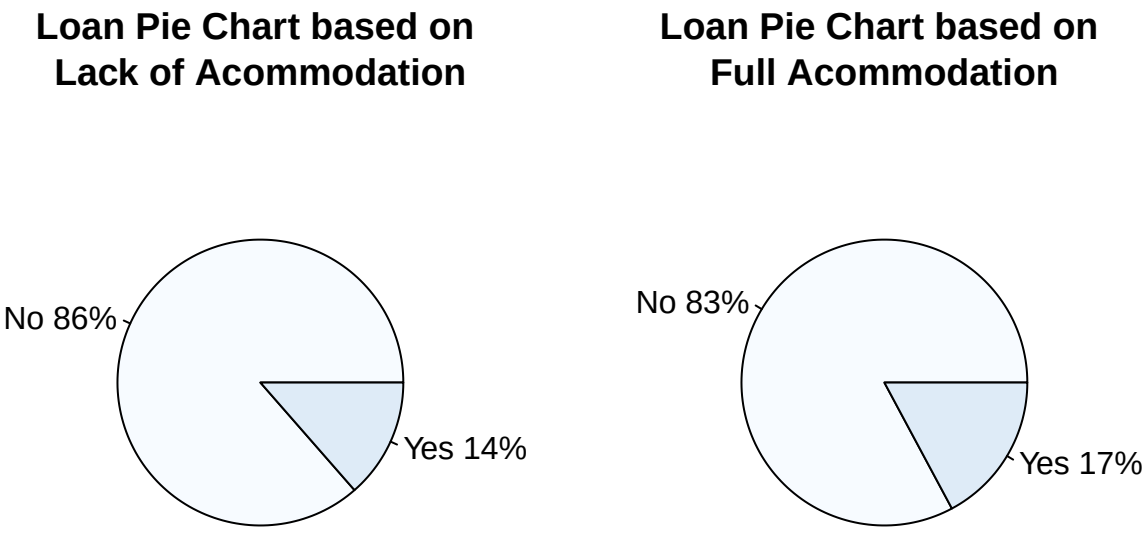
After Cleansing Data

```
par(opar4)
```

2. Similarly, lower and working-class customers (no accommodation/renting) will likely loan less than the middle and upper (own a house) because they do not have to pay a mortgage loan, implying that the second hypothesis might be true.

```
# Before cleaning
opar1 <- par(mfrow=c(1,2))

newdf_house1 <- newdf[newdf$age > 30 & newdf$age < 60
                      &newdf$housing == "no",]
x1 <- c(sum(newdf_house1$loan_code == 0), sum(newdf_house1$loan_code == 1))
label1 <- c("No","Yes")
pct1 <- round(x1/sum(x1)*100)
label1 <- paste(label1, pct1)
label1 <- paste(label1,"%",sep = "")
pie(x1, labels = label1, col = blues9,
      main = "Loan Pie Chart based on \nLack of Acommodation")

newdf_house2 <- newdf[newdf$age > 30 & newdf$age < 60
                      &newdf$housing == "yes",]
x2 <- c(sum(newdf_house2$loan_code == 0), sum(newdf_house2$loan_code == 1))
```

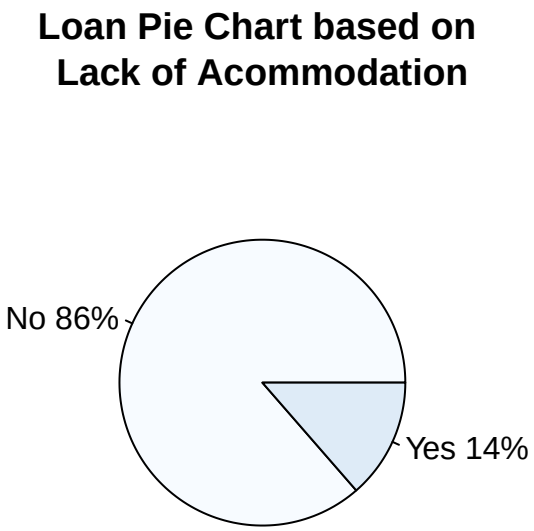
```
label2 <- c("No","Yes")
pct2 <- round(x2/sum(x2)*100)
label2 <- paste(label2, pct2)
label2 <- paste(label2,"%",sep = "")
pie(x2, labels = label2, col = blues9,
      main = "Loan Pie Chart based on \nFull Acommodation")
mtext("Before Cleansing Data", outer = TRUE, line =-22, col = "red")
```

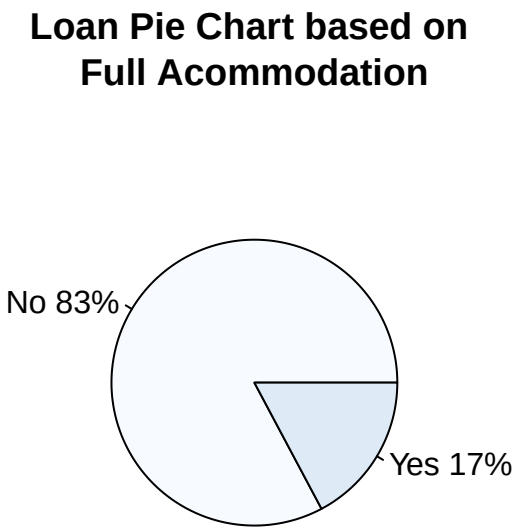


Before Cleansing Data

```
par(opar1)

# After cleaning
opar5 <- par(mfrow=c(1,2))
cleandf_house1 <- cleandf[cleandf$age > 30 & cleandf$age < 60
                      &cleandf$housing == "no",]
cx1 <- c(sum(cleandf_house1$loan_code == 0), sum(cleandf_house1$loan_code == 1))
clabel1 <- c("No","Yes")
cpct1 <- round(cx1/sum(cx1)*100)
clabel1 <- paste(clabel1, cpct1)
clabel1 <- paste(clabel1,"%",sep = "")
pie(cx1, labels = clabel1, col = blues9,
    main = "Loan Pie Chart based on \nLack of Acommodation")

cleandf_house2 <- cleandf[cleandf$age > 30 & cleandf$age < 60
                      &cleandf$housing == "yes",]
cx2 <- c(sum(cleandf_house2$loan_code == 0), sum(cleandf_house2$loan_code == 1))
```

```
clabel2 <- c("No","Yes")
cpct2 <- round(cx2/sum(cx2)*100)
clabel2 <- paste(clabel2, cpct2)
clabel2 <- paste(clabel2,"%",sep = "")
pie(cx2, labels = clabel2, col = blues9,
    main = "Loan Pie Chart based on \nFull Acommodation")
mtext("After Cleansing Data", outer = TRUE, line =-22, col = "red")
```





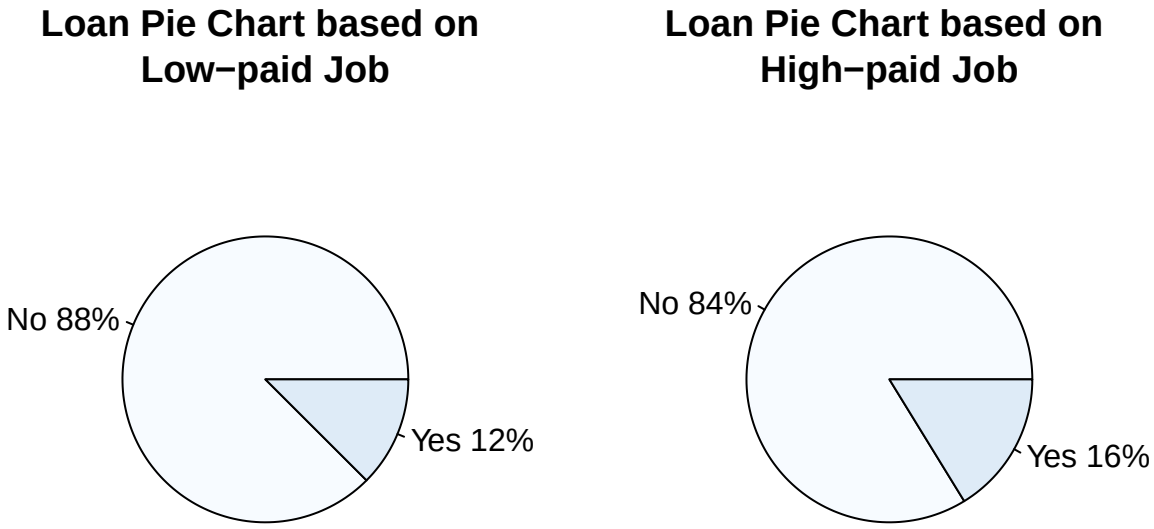
After Cleansing Data

```
par(opar5)
```

3. Surprisingly, the lower working class also has a lower loaning approval rate than the middle and upper class, which might be due to their high default rate or credibility. This information suggests that my third hypotheses is false.

```
# Before cleaning
opar2 <- par(mfrow=c(1,2))

newdf_job1 <- newdf[newdf$job == c("self-employed","housemaid"),]
x3 <- c(sum(newdf_job1$loan_code == 0), sum(newdf_job1$loan_code == 1))
label3 <- c("No","Yes")
pct3 <- round(x3/sum(x3)*100)
label3 <- paste(label3, pct3)
label3 <- paste(label3,"%",sep = "")
pie(x3, labels = label3, col = blues9,
```

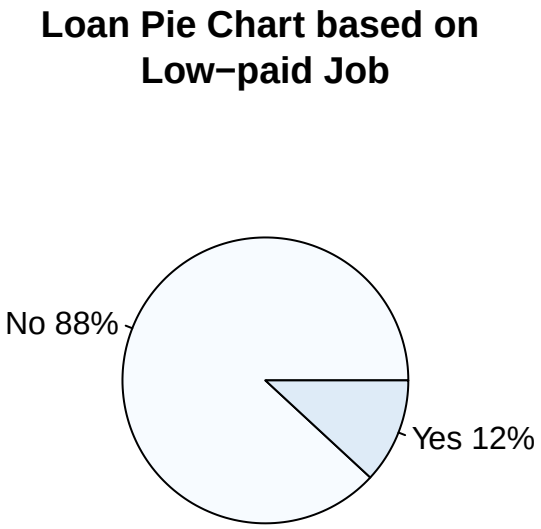
```
      main = "Loan Pie Chart based on \nLow-paid Job")

newdf_job2 <- newdf[newdf$job == c("blue-collar","admin."),]
x4 <- c(sum(newdf_job2$loan_code == 0), sum(newdf_job2$loan_code == 1))
label4 <- c("No","Yes")
pct4 <- round(x4/sum(x4)*100)
label4 <- paste(label4, pct4)
label4 <- paste(label4,"%",sep = "")
pie(x4, labels = label4, col = blues9,
      main = "Loan Pie Chart based on \nHigh-paid Job")
mtext("Before Cleansing Data", outer = TRUE, line =-22, col = "red")
```

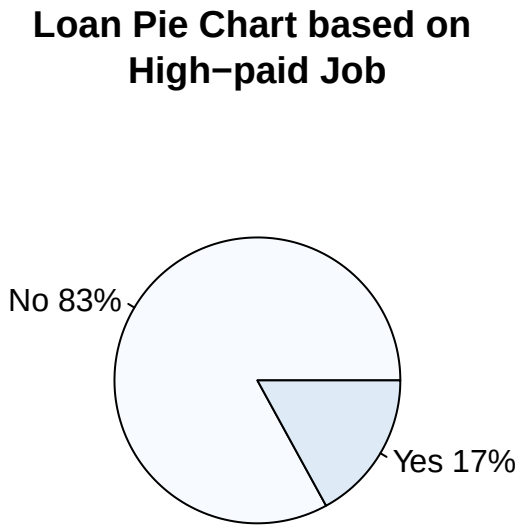


Before Cleansing Data

```
par(opar2)

# After cleaning
opar6 <- par(mfrow=c(1,2))
cleandf_job1 <- cleandf[cleandf$job == c("self-employed","housemaid"),]
cx3 <- c(sum(cleandf_job1$loan_code == 0), sum(cleandf_job1$loan_code == 1))
clabel3 <- c("No","Yes")
cpct3 <- round(cx3/sum(cx3)*100)
clabel3 <- paste(clabel3, cpct3)
clabel3 <- paste(clabel3,"%",sep = "")
pie(cx3, labels = clabel3, col = blues9,
    main = "Loan Pie Chart based on \nLow-paid Job")
```

```
cleandf_job2 <- cleandf[cleandf$job == c("blue-collar","admin."),]
cx4 <- c(sum(cleandf_job2$loan_code == 0), sum(cleandf_job2$loan_code == 1))
clabel4 <- c("No","Yes")
cpct4 <- round(cx4/sum(cx4)*100)
clabel4 <- paste(clabel4, cpct4)
clabel4 <- paste(clabel4,"%",sep = "")
pie(cx4, labels = clabel4, col = blues9,
      main = "Loan Pie Chart based on \nHigh-paid Job")
mtext("After Cleansing Data", outer = TRUE, line =-22, col = "red")
```





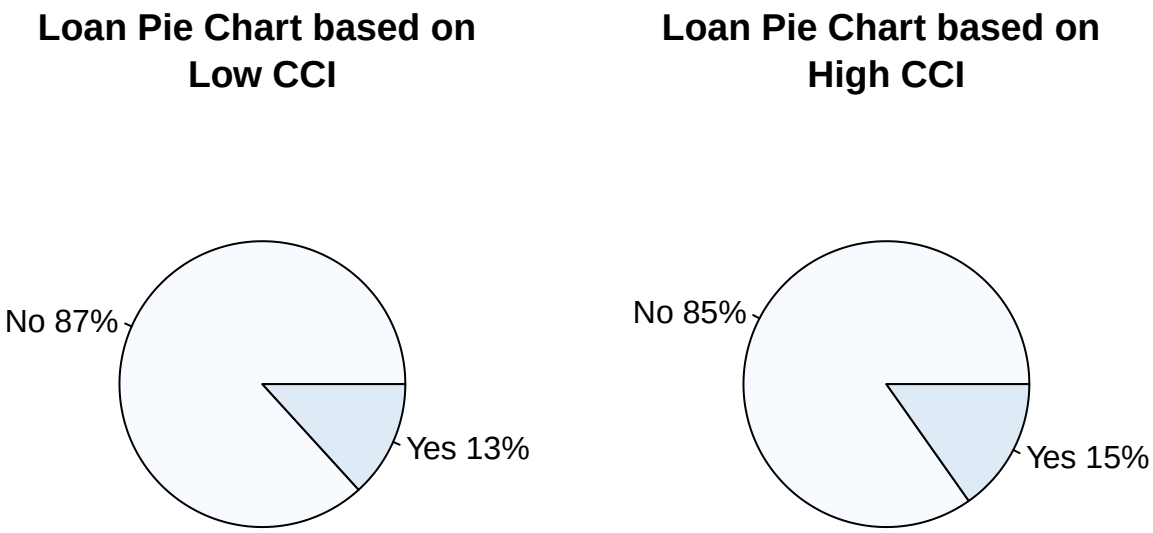
After Cleansing Data

```
par(opar6)
```

4. I then create a scatter plot chart to observe the correlation between the CCI and CPI levels. And I found that people who are pessimistic about the future economy will likely spend less and loan less. In contrast, optimistic customers spend more and hence loan more.

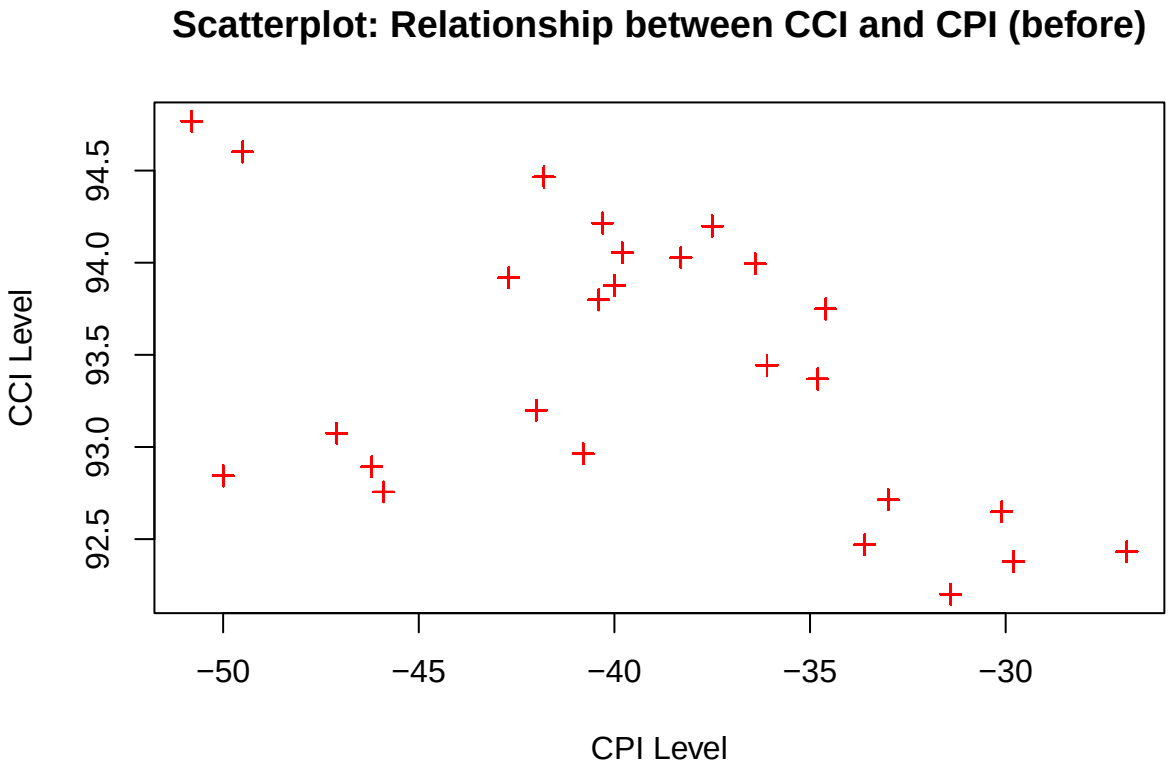
```
# Before cleaning
opar3 <- par(mfrow=c(1,2))

newdf_default1 <- newdf[newdf$CCI < 92.5,]
x5 <- c(sum(newdf_default1$loan_code == 0), sum(newdf_default1$loan_code == 1))
label5 <- c("No","Yes")
pct5 <- round(x5/sum(x5)*100)
label5 <- paste(label5, pct5)
```

```r
label5 <- paste(label5,"%",sep = "")
pie(x5, labels = label5, col = blues9,
      main = "Loan Pie Chart based on \nLow CCI")

newdf_default2 <- newdf[newdf$CCI > 94,]
x6 <- c(sum(newdf_default2$loan_code == 0), sum(newdf_default2$loan_code == 1))
label6 <- c("No","Yes")
pct6 <- round(x6/sum(x6)*100)
label6 <- paste(label6, pct6)
label6 <- paste(label6,"%",sep = "")
pie(x6, labels = label6, col = blues9,
    main = "Loan Pie Chart based on \nHigh CCI")
mtext("Before Cleansing Data", outer = TRUE, line =-22, col = "red")
```

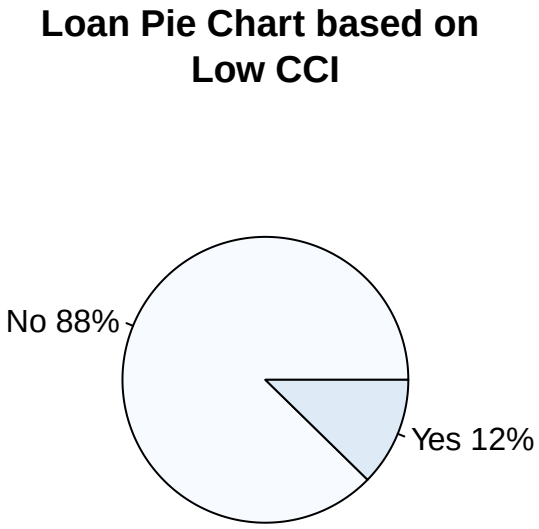


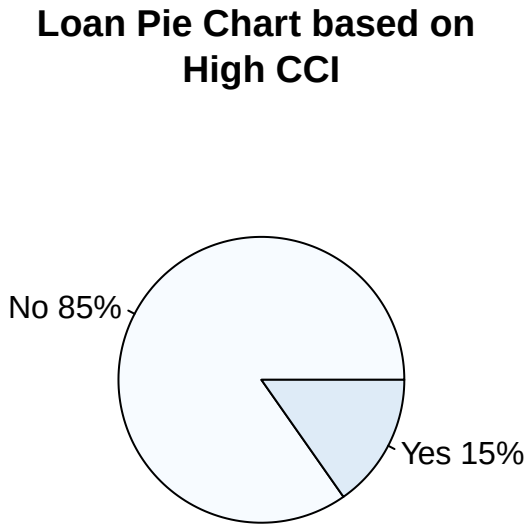
Before Cleansing Data

```r
par(opar3)

plot(newdf$CPI, newdf$CCI, xlab = "CPI Level", ylab = "CCI Level",
     main = "Scatterplot: Relationship between CCI and CPI (before)",
     col = "red", pch = 3)
```



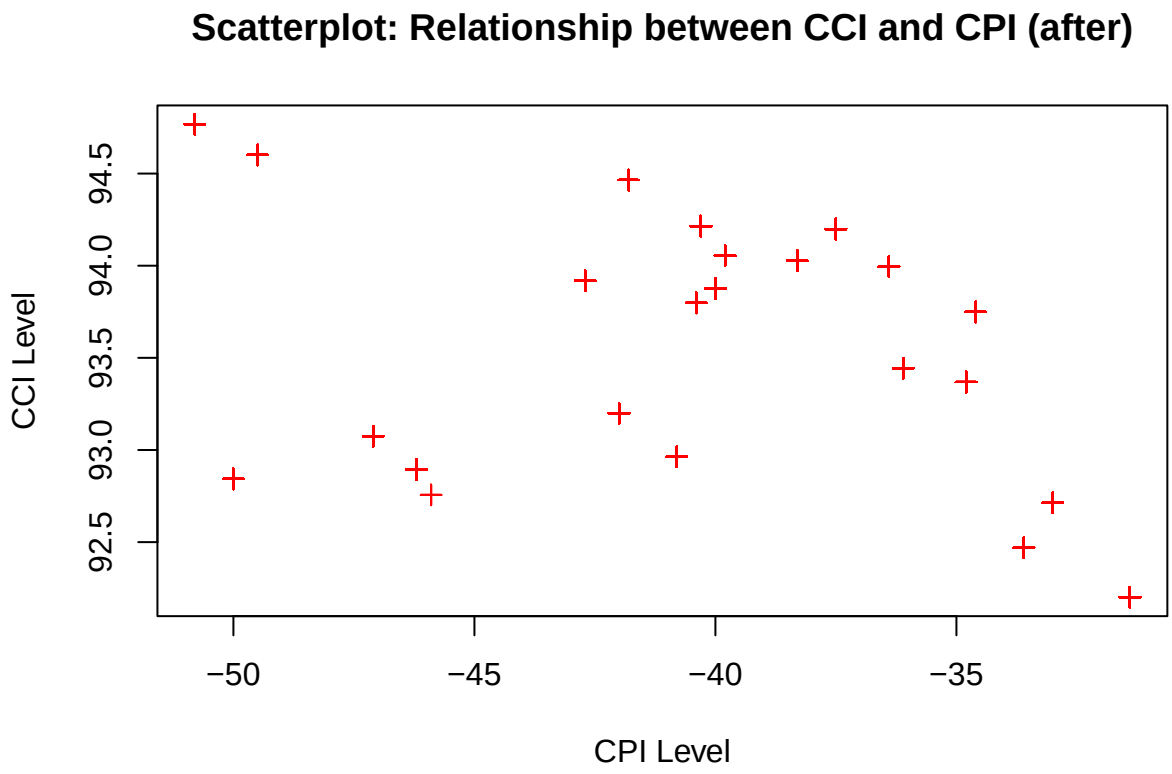

```
# After cleaning
opar7 <- par(mfrow=c(1,2))

cleandf_CCI1 <- cleandf[cleandf$CCI < 92.5,]
cx5 <- c(sum(cleandf_CCI1$loan_code == 0), sum(cleandf_CCI1$loan_code == 1))
clabel5 <- c("No","Yes")
cpct5 <- round(cx5/sum(cx5)*100)
clabel5 <- paste(clabel5, cpct5)
clabel5 <- paste(clabel5,"%",sep = "")
pie(cx5, labels = clabel5, col = blues9,
    main = "Loan Pie Chart based on \nLow CCI")

cleandf_CCI2 <- cleandf[cleandf$CCI > 94,]
cx6 <- c(sum(cleandf_CCI2$loan_code == 0), sum(cleandf_CCI2$loan_code == 1))
clabel6 <- c("No","Yes")
cpct6 <- round(cx6/sum(cx6)*100)
clabel6 <- paste(clabel6, cpct6)
clabel6 <- paste(clabel6,"%",sep = "")
pie(cx6, labels = clabel6, col = blues9,
      main = "Loan Pie Chart based on \nHigh CCI")
mtext("After Cleansing Data", outer = TRUE, line =-22, col = "red")
```





After Cleansing Data

```
par(opar7)

plot(cleandf$CPI, cleandf$CCI, xlab = "CPI Level", ylab = "CCI Level",
     main = "Scatterplot: Relationship between CCI and CPI (after)",
     col = "red", pch = 3)
```



Task 3: Change one repetitive code block with a for-loop

(I think about maybe inflation could affect the bank loan rates. So I compute monthly inflation rates using the CPI from each month.)

```
inflation <- function(a,b){
  c <- ((b-a)/a)*100
}
rate <- c(inflation(mean(cleandf$CPI[cleandf$month=="apr"]),
                mean(cleandf$CPI[cleandf$month=="aug"])),
inflation(mean(cleandf$CPI[cleandf$month=="aug"]),
                mean(cleandf$CPI[cleandf$month=="dec"])),
inflation(mean(cleandf$CPI[cleandf$month=="dec"]),
                mean(cleandf$CPI[cleandf$month=="jul"])),
inflation(mean(cleandf$CPI[cleandf$month=="jul"]),
                mean(cleandf$CPI[cleandf$month=="jun"])),
inflation(mean(cleandf$CPI[cleandf$month=="jun"]),
                mean(cleandf$CPI[cleandf$month=="mar"])),
inflation(mean(cleandf$CPI[cleandf$month=="mar"]),
                mean(cleandf$CPI[cleandf$month=="may"])),
inflation(mean(cleandf$CPI[cleandf$month=="may"]),
                mean(cleandf$CPI[cleandf$month=="nov"])),
inflation(mean(cleandf$CPI[cleandf$month=="nov"]),
                mean(cleandf$CPI[cleandf$month=="oct"])),
```

```
inflation(mean(cleandf$CPI[cleandf$month=="oct"]),
               mean(cleandf$CPI[cleandf$month=="sep"])))

for (n in rate) {
    message(sprintf("%.2f ",n), "is the inflation rate") # changed to msg
}
```

```
## -23.27 is the inflation rate

## -4.34 is the inflation rate

## 24.26 is the inflation rate

## -1.89 is the inflation rate

## 3.38 is the inflation rate

## -4.22 is the inflation rate

## 2.86 is the inflation rate

## 10.55 is the inflation rate

## -19.78 is the inflation rate
```

Task 4: Double-checking effects of oddities

The before-effect was all the pie charts I have shown above. I assume that the after-effect will count all the unknown values in every situation, meaning that every pie chart or histogram would show another bar or another portion for unknown values. For instance, if I present a histogram for loan total with the unknowns, it would depict three bars for each month instead of two. Another change is the significant drop of ten thousand observations in the data set. Fortunately, my hypotheses do not require me to calculate the values for unknown values, as I did not aim to learn about the effects of the unknown values from the beginning.

```
df1 <- read.csv('https://raw.githubusercontent.com/ALY6000/datasets/main/kaggle/bankmarketing.csv',
                na.string=c("unknown",0))
newdf1 <- data.frame(df$loan, df$housing, df$default,
                     df$age, df$job, df$marital,
                     df$education, df$cons.conf.idx,
                     df$cons.price.idx, df$nr.employed,
                     df$month, df$euribor3m,
                     df$day_of_week, df$duration, df$contact)
colnames(newdf1) <- c("loan", "housing", "default",
                      "age", "job", "marital",
                      "education", "CPI",
                      "CCI", "years_of_emmployment",
                      "month", "eurobankrate",
                      "day_of_week","duration",
                      "contact")
str(newdf1)
```

```
## 'data.frame':    41188 obs. of  15 variables:
##  $ loan                : chr  "no" "no" "no" "no" ...
##  $ housing             : chr  "no" "no" "yes" "no" ...
##  $ default             : chr  "no" "unknown" "no" "no" ...
##  $ age                 : int  56 57 37 40 56 45 59 41 24 25 ...
##  $ job                 : chr  "housemaid" "services" "services" "admin." ...
##  $ marital             : chr  "married" "married" "married" "married" ...
##  $ education           : chr  "basic.4y" "high.school" "high.school" "basic.6y" ...
##  $ CPI                 : num  -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...
##  $ CCI                 : num  94 94 94 94 94 ...
##  $ years_of_emmployment: num  5191 5191 5191 5191 5191 ...
##  $ month               : chr  "may" "may" "may" "may" ...
##  $ eurobankrate        : num  4.86 4.86 4.86 4.86 4.86 ...
##  $ day_of_week         : chr  "mon" "mon" "mon" "mon" ...
##  $ duration            : int  261 149 226 151 307 198 139 217 380 50 ...
##  $ contact             : chr  "telephone" "telephone" "telephone" "telephone" ...
```

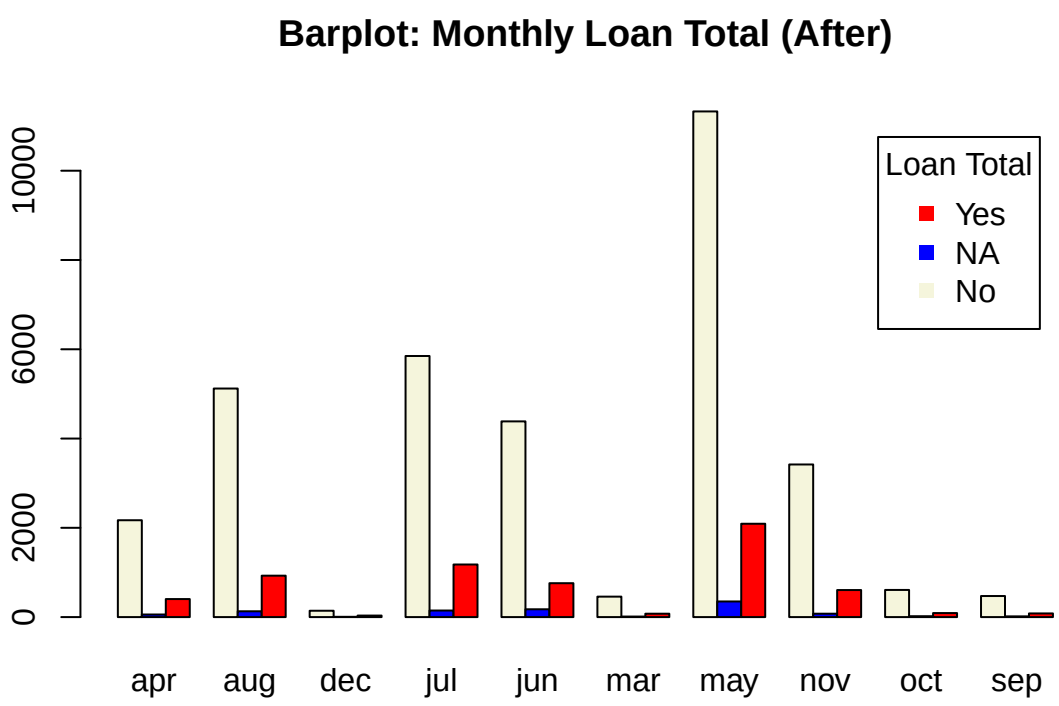
```
str(newdf)
```

```
## 'data.frame':    30488 obs. of  16 variables:
##  $ loan                : chr  "no" "no" "no" "yes" ...
##  $ housing             : chr  "no" "yes" "no" "no" ...
##  $ default             : chr  "no" "no" "no" "no" ...
##  $ age                 : int  56 37 40 56 59 24 25 25 29 57 ...
##  $ job                 : chr  "housemaid" "services" "admin." "services" ...
##  $ marital             : chr  "married" "married" "married" "married" ...
##  $ education           : chr  "basic.4y" "high.school" "basic.6y" "high.school" ...
##  $ CPI                 : num  -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...
##  $ CCI                 : num  94 94 94 94 94 ...
##  $ years_of_emmployment: num  14.2 14.2 14.2 14.2 14.2 ...
##  $ month               : chr  "may" "may" "may" "may" ...
##  $ eurobankrate        : num  4.86 4.86 4.86 4.86 4.86 ...
##  $ day_of_week         : chr  "mon" "mon" "mon" "mon" ...
##  $ duration            : int  261 226 151 307 139 380 50 222 137 293 ...
##  $ contact             : chr  "telephone" "telephone" "telephone" "telephone" ...
##  $ loan_code           : num  0 0 0 1 0 0 0 0 1 0 ...
```

```
# Example of the before and after effect
counts1 <- table(newdf1$loan, newdf1$month)
counts1
```

```
##
##             apr   aug   dec   jul   jun   mar   may   nov   oct   sep
##   no       2170  5120   142  5849  4384   458 11328  3419   608   472
##   unknown    58   130     6   147   175    12   350    76    20    16
##   yes       404   928    34  1178   759    76  2091   606    90    82
```

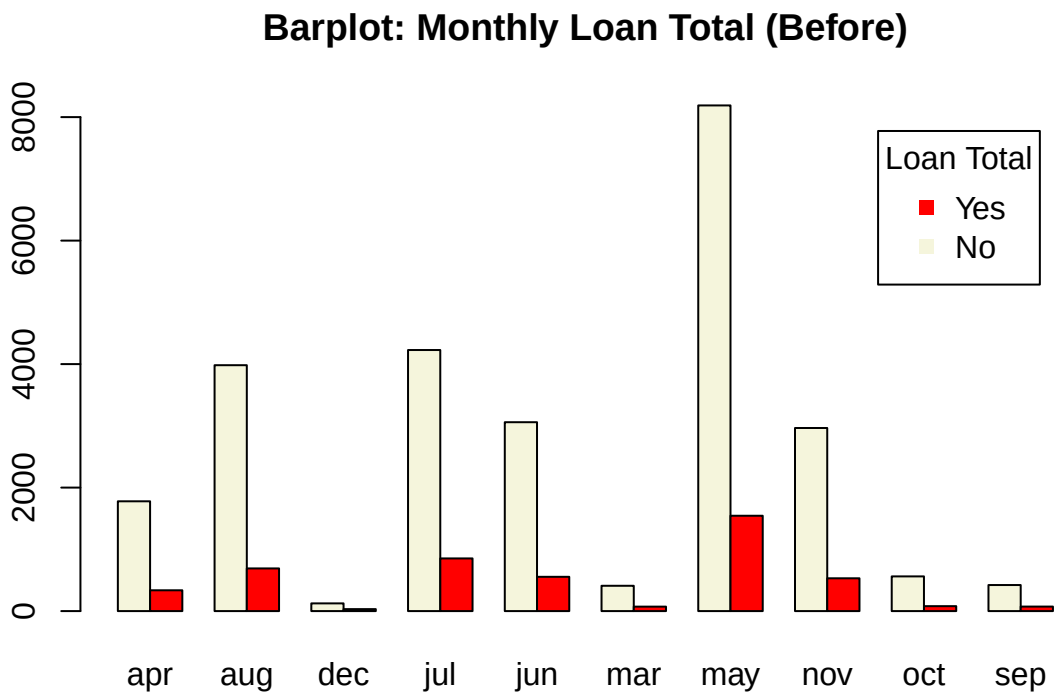
```
barplot(counts1, beside = TRUE, col = c("beige","blue","red"),
        main = "Barplot: Monthly Loan Total (After)")
legend("topright", inset = .05,
       title = "Loan Total", c("Yes","NA","No"),
       col = c("red","blue","beige"), pch = c(15))
```




```
counts2 <- table(newdf$loan, newdf$month)
counts2
```

```
##
##        apr  aug  dec  jul  jun  mar  may  nov  oct  sep
##   no  1778 3982  125 4228 3059  410 8189 2965  562  422
##   yes  337  691   32  853  555   72 1544  531   80   73
```

```
barplot(counts2, beside = TRUE, col = c("beige","red"),
        main = "Barplot: Monthly Loan Total (Before)")
legend("topright", inset = .05,
       title = "Loan Total", c("Yes","No"),
       col = c("red","beige"), pch = c(15))
```



Task 5 & 6: A for-loop to automatically summarize huge datasets

```
t1<-read.csv('https://www.dropbox.com/s/vp44yozebx5xgok/bdiag.csv?dl=1')
t2<-read.csv('https://raw.githubusercontent.com/ALY6000/datasets/main/kaggle/train.csv')

summarize_df1 <- function( a )
{
  for (i in colnames( a )) {
    if (is.character(unlist(a[i]))) {
      cat(i,":", length(unique(unlist(a[i])))/2,
          "example values are", head(unique(unlist(a[i]))), "\n")
    } else if (is.integer(unlist(a[i]))) {
      cat(i, ":", min(unlist(a[i])), "and", max(unlist(a[i])),
          "are its min and max values, with", which.max(unlist(a[i])),
      "being the most common value", "\n")
    } else if (is.double(unlist(a[i]))) {
      cat(i, ":", "its mean and interquartile range are",
          mean(unlist(a[i])), "and", IQR(unlist(a[i])), ", respectively", "\n")
    }
  }
}

summarize_df1(df)
```

```
## age : 17 and 98 are its min and max values, with 38453 being the most common value
## job : 6 example values are housemaid services admin. blue-collar technician retired
## marital : 2 example values are married single divorced unknown
## education : 4 example values are basic.4y high.school basic.6y basic.9y professional.course unknown
## default : 1.5 example values are no unknown yes
## housing : 1.5 example values are no yes unknown
## loan : 1.5 example values are no yes unknown
## contact : 1 example values are telephone cellular
## month : 5 example values are may jun jul aug oct nov
## day_of_week : 2.5 example values are mon tue wed thu fri
## duration : 0 and 4918 are its min and max values, with 24092 being the most common value
## campaign : 1 and 56 are its min and max values, with 4108 being the most common value
## pdays : 0 and 999 are its min and max values, with 1 being the most common value
## previous : 0 and 7 are its min and max values, with 41082 being the most common value
## poutcome : 1.5 example values are nonexistent failure success
## emp.var.rate : its mean and interquartile range are 0.0818855 and 3.2 , respectively
## cons.price.idx : its mean and interquartile range are 93.57566 and 0.919 , respectively
## cons.conf.idx : its mean and interquartile range are -40.5026 and 6.3 , respectively
## euribor3m : its mean and interquartile range are 3.621291 and 3.617 , respectively
## nr.employed : its mean and interquartile range are 5167.036 and 129 , respectively
## y : 1 example values are no yes
```

```
summarize_df1(t1)
```

```
## id : 8670 and 911320502 are its min and max values, with 465 being the most common value
## diagnosis : 1 example values are M B
## radius_mean : its mean and interquartile range are 14.12729 and 4.08 , respectively
## texture_mean : its mean and interquartile range are 19.28965 and 5.63 , respectively
## perimeter_mean : its mean and interquartile range are 91.96903 and 28.93 , respectively
## area_mean : its mean and interquartile range are 654.8891 and 362.4 , respectively
## smoothness_mean : its mean and interquartile range are 0.09636028 and 0.01893 , respectively
## compactness_mean : its mean and interquartile range are 0.104341 and 0.06548 , respectively
## concavity_mean : its mean and interquartile range are 0.08879932 and 0.10114 , respectively
## concave.points_mean : its mean and interquartile range are 0.04891915 and 0.05369 , respectively
## symmetry_mean : its mean and interquartile range are 0.1811619 and 0.0338 , respectively
## fractal_dimension_mean : its mean and interquartile range are 0.06279761 and 0.00842 , respectively
## radius_se : its mean and interquartile range are 0.4051721 and 0.2465 , respectively
## texture_se : its mean and interquartile range are 1.216853 and 0.6401 , respectively
## perimeter_se : its mean and interquartile range are 2.866059 and 1.751 , respectively
## area_se : its mean and interquartile range are 40.33708 and 27.34 , respectively
## smoothness_se : its mean and interquartile range are 0.007040979 and 0.002977 , respectively
## compactness_se : its mean and interquartile range are 0.02547814 and 0.01937 , respectively
## concavity_se : its mean and interquartile range are 0.03189372 and 0.02696 , respectively
## concave.points_se : its mean and interquartile range are 0.01179614 and 0.007072 , respectively
## symmetry_se : its mean and interquartile range are 0.0205423 and 0.00832 , respectively
## fractal_dimension_se : its mean and interquartile range are 0.003794904 and 0.00231 , respectively
## radius_worst : its mean and interquartile range are 16.26919 and 5.78 , respectively
## texture_worst : its mean and interquartile range are 25.67722 and 8.64 , respectively
## perimeter_worst : its mean and interquartile range are 107.2612 and 41.29 , respectively
## area_worst : its mean and interquartile range are 880.5831 and 568.7 , respectively
## smoothness_worst : its mean and interquartile range are 0.1323686 and 0.0294 , respectively
## compactness_worst : its mean and interquartile range are 0.254265 and 0.1919 , respectively
## concavity_worst : its mean and interquartile range are 0.2721885 and 0.2684 , respectively
## concave.points_worst : its mean and interquartile range are 0.1146062 and 0.09647 , respectively
```

```
## symmetry_worst : its mean and interquartile range are 0.2900756 and 0.0675 , respectively
## fractal_dimension_worst : its mean and interquartile range are 0.08394582 and 0.02062 , respectively
```

```
summarize_df1(t2)
```

```
## Id : 1 and 1460 are its min and max values, with 1460 being the most common value
## MSSubClass : 20 and 190 are its min and max values, with 10 being the most common value
## MSZoning : 2.5 example values are RL RM C (all) FV RH
## LotFrontage : NA and NA are its min and max values, with 935 being the most common value
## LotArea : 1300 and 215245 are its min and max values, with 314 being the most common value
## Street : 1 example values are Pave Grvl
## Alley : 1.5 example values are NA Grvl Pave
## LotShape : 2 example values are Reg IR1 IR2 IR3
## LandContour : 2 example values are Lvl Bnk Low HLS
## Utilities : 1 example values are AllPub NoSeWa
## LotConfig : 2.5 example values are Inside FR2 Corner CulDSac FR3
## LandSlope : 1.5 example values are Gtl Mod Sev
## Neighborhood : 12.5 example values are CollgCr Veenker Crawfor NoRidge Mitchel Somerst
## Condition1 : 4.5 example values are Norm Feedr PosN Artery RRAe RRNn
## Condition2 : 4 example values are Norm Artery RRNn Feedr PosN PosA
## BldgType : 2.5 example values are 1Fam 2fmCon Duplex TwnhsE Twnhs
## HouseStyle : 4 example values are 2Story 1Story 1.5Fin 1.5Unf SFoyer SLvl
## OverallQual : 1 and 10 are its min and max values, with 59 being the most common value
## OverallCond : 1 and 9 are its min and max values, with 186 being the most common value
## YearBuilt : 1872 and 2010 are its min and max values, with 379 being the most common value
## YearRemodAdd : 1950 and 2010 are its min and max values, with 158 being the most common value
## RoofStyle : 3 example values are Gable Hip Gambrel Mansard Flat Shed
## RoofMatl : 4 example values are CompShg WdShngl Metal WdShake Membran Tar&Grv
## Exterior1st : 7.5 example values are VinylSd MetalSd Wd Sdng HdBoard BrkFace WdShing
## Exterior2nd : 8 example values are VinylSd MetalSd Wd Shng HdBoard Plywood Wd Sdng
## MasVnrType : 2.5 example values are BrkFace None Stone BrkCmn NA
## MasVnrArea : NA and NA are its min and max values, with 298 being the most common value
## ExterQual : 2 example values are Gd TA Ex Fa
## ExterCond : 2.5 example values are TA Gd Fa Po Ex
## Foundation : 3 example values are PConc CBlock BrkTil Wood Slab Stone
## BsmtQual : 2.5 example values are Gd TA Ex NA Fa
## BsmtCond : 2.5 example values are TA Gd NA Fa Po
## BsmtExposure : 2.5 example values are No Gd Mn Av NA
## BsmtFinType1 : 3.5 example values are GLQ ALQ Unf Rec BLQ NA
## BsmtFinSF1 : 0 and 5644 are its min and max values, with 1299 being the most common value
## BsmtFinType2 : 3.5 example values are Unf BLQ NA ALQ Rec LwQ
## BsmtFinSF2 : 0 and 1474 are its min and max values, with 323 being the most common value
## BsmtUnfSF : 0 and 2336 are its min and max values, with 225 being the most common value
## TotalBsmtSF : 0 and 6110 are its min and max values, with 1299 being the most common value
## Heating : 3 example values are GasA GasW Grav Wall OthW Floor
## HeatingQC : 2.5 example values are Ex Gd TA Fa Po
## CentralAir : 1 example values are Y N
## Electrical : 3 example values are SBrkr FuseF FuseA FuseP Mix NA
## X1stFlrSF : 334 and 4692 are its min and max values, with 1299 being the most common value
## X2ndFlrSF : 0 and 2065 are its min and max values, with 1183 being the most common value
## LowQualFinSF : 0 and 572 are its min and max values, with 186 being the most common value
## GrLivArea : 334 and 5642 are its min and max values, with 1299 being the most common value
## BsmtFullBath : 0 and 3 are its min and max values, with 739 being the most common value
## BsmtHalfBath : 0 and 2 are its min and max values, with 598 being the most common value
```

```
## FullBath : 0 and 3 are its min and max values, with 12 being the most common value
## HalfBath : 0 and 2 are its min and max values, with 189 being the most common value
## BedroomAbvGr : 0 and 8 are its min and max values, with 636 being the most common value
## KitchenAbvGr : 0 and 3 are its min and max values, with 49 being the most common value
## KitchenQual : 2 example values are Gd TA Ex Fa
## TotRmsAbvGrd : 2 and 14 are its min and max values, with 636 being the most common value
## Functional : 3.5 example values are Typ Min1 Maj1 Min2 Mod Maj2
## Fireplaces : 0 and 3 are its min and max values, with 167 being the most common value
## FireplaceQu : 3 example values are NA TA Gd Fa Ex Po
## GarageType : 3.5 example values are Attchd Detchd BuiltIn CarPort NA Basment
## GarageYrBlt : NA and NA are its min and max values, with 379 being the most common value
## GarageFinish : 2 example values are RFn Unf Fin NA
## GarageCars : 0 and 4 are its min and max values, with 421 being the most common value
## GarageArea : 0 and 1418 are its min and max values, with 1299 being the most common value
## GarageQual : 3 example values are TA Fa Gd NA Ex Po
## GarageCond : 3 example values are TA Fa NA Gd Po Ex
## PavedDrive : 1.5 example values are Y N P
## WoodDeckSF : 0 and 857 are its min and max values, with 54 being the most common value
## OpenPorchSF : 0 and 547 are its min and max values, with 1329 being the most common value
## EnclosedPorch : 0 and 552 are its min and max values, with 198 being the most common value
## X3SsnPorch : 0 and 508 are its min and max values, with 206 being the most common value
## ScreenPorch : 0 and 480 are its min and max values, with 1329 being the most common value
## PoolArea : 0 and 738 are its min and max values, with 1424 being the most common value
## PoolQC : 2 example values are NA Ex Fa Gd
## Fence : 2.5 example values are NA MnPrv GdWo GdPrv MnWw
## MiscFeature : 2.5 example values are NA Shed Gar2 Othr TenC
## MiscVal : 0 and 15500 are its min and max values, with 347 being the most common value
## MoSold : 1 and 12 are its min and max values, with 5 being the most common value
## YrSold : 2006 and 2010 are its min and max values, with 17 being the most common value
## SaleType : 4.5 example values are WD New COD ConLD ConLI CWD
## SaleCondition : 3 example values are Normal Abnorml Partial AdjLand Alloca Family
## SalePrice : 34900 and 755000 are its min and max values, with 692 being the most common value
```

```
# I use the which.max() function instead of mode() function (Sources: Tutorialspoint)
```

References

1. Consumer Confidence Index (CCI). (September, 2022). OECD Data. https://data.oecd.org/leadind/consumer-confidence-index-cci.htm

2. Kabacoff, R. I. (May 2015). Chapter 6: Basic graphs. R in action: Data analysis and graphics in R (2nd ed). Manning. ISBN: 978-1-617-29138-8.

3. MISH. (February, 2022). What's the relationship between the CPI and increases in banklending ?. Mish Talk. https://mishtalk.com/economics/whats-the-relationship-between-the-cpi-and-increases-in-bank-lending

4. Pandit, S. (September, 2022). https://www.kaggle.com/datasets/shrutipandit707/bankmarketing

5. Sarveshwar, K.I. (September, 2016). Sarveshwar Inani's blog. http://learningeconometrics.blogspot.com/2016/09/four-moments-of-distribution-mean.html

6. Turney, S. (June, 2022). What is Kurtosis? Definition, examples & formula. Scribbr. https://www.scribbr.com/statistics/kurtosis/

7. Tutorialspoint. How to find the most frequent factor value in an R dataframe column. https://www.tutorialspoint.com/how-to-find-the-most-frequent-factor-value-in-an-r-data-frame-column#:~:text=More%20Detail-,To%20find%20the%20most%20frequent%20factor%20value%20in%20an%20R,which

URL Colab + Github

Colab: https://colab.research.google.com/drive/1paiQVDjTwoljB6yVNG-dprMSH951hkzg#scrollTo=1zFPmTX8FyUc

Github: https://github.com/rickmai99/ALY6000.git