Data Understanding

Table of contents

Adding to the foundation of Business Understanding, the Data Understanding phase focuses on identifying, collecting, and analyzing data sets that can help the project. This phase also has four tasks:

See *Variables Overview.xlsx* for further information!

# load libraries, that might be needed  
library(knitr) # web widget  
library(tidyverse) # data manipulation  
library(data.table) # fast file reading  
library(caret) # rocr analysis  
library(ROCR) # rocr analysis  
library(kableExtra) # nice table html formating   
library(gridExtra) # arranging ggplot in grid  
library(rpart) # decision tree  
library(rpart.plot) # decision tree plotting  
library(caTools) # split

## 1. Collect initial data: Acquire the necessary data and (if necessary) load it into your analysis tool.

### **Data overview**

# Load dataset with separator ";", a header and empty cells recognized as NAs  
data <- read.csv("marketing.csv", sep = ";", header = TRUE, na.strings = "")  
  
# Check the data set  
names(data) # Displays names of attributes

[1] "age" "job" "marital" "education"   
 [5] "default" "housing" "loan" "month"   
 [9] "day\_of\_week" "campaign" "previous" "poutcome"   
[13] "emp.var.rate" "cons.price.idx" "cons.conf.idx" "euribor3m"   
[17] "y"

head(data) # Displays the first lines of the data set

age job marital education default housing loan month day\_of\_week  
1 56 housemaid married basic.4y no no no may mon  
2 57 services married high.school <NA> no no may mon  
3 37 services married high.school no yes no may mon  
4 40 admin. married basic.6y no no no may mon  
5 56 services married high.school no no yes may mon  
6 45 services married basic.9y <NA> no no may mon  
 campaign previous poutcome emp.var.rate cons.price.idx cons.conf.idx  
1 1 0 nonexistent 1.1 93.994 -36.4  
2 1 0 nonexistent 1.1 93.994 -36.4  
3 1 0 nonexistent 1.1 93.994 -36.4  
4 1 0 nonexistent 1.1 93.994 -36.4  
5 1 0 nonexistent 1.1 93.994 -36.4  
6 1 0 nonexistent 1.1 93.994 -36.4  
 euribor3m y  
1 4.857 no  
2 4.857 no  
3 4.857 no  
4 4.857 no  
5 4.857 no  
6 4.857 no

str(data) # Outputs a summary of the data structure

'data.frame': 41188 obs. of 17 variables:  
 $ 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" ...  
 $ default : chr "no" NA "no" "no" ...  
 $ housing : chr "no" "no" "yes" "no" ...  
 $ loan : chr "no" "no" "no" "no" ...  
 $ month : chr "may" "may" "may" "may" ...  
 $ day\_of\_week : chr "mon" "mon" "mon" "mon" ...  
 $ campaign : int 1 1 1 1 1 1 1 1 1 1 ...  
 $ previous : int 0 0 0 0 0 0 0 0 0 0 ...  
 $ poutcome : chr "nonexistent" "nonexistent" "nonexistent" "nonexistent" ...  
 $ emp.var.rate : num 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...  
 $ cons.price.idx: num 94 94 94 94 94 ...  
 $ cons.conf.idx : num -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...  
 $ euribor3m : num 4.86 4.86 4.86 4.86 4.86 ...  
 $ y : chr "no" "no" "no" "no" ...

nrow(data) # cross checking number or rows

[1] 41188

dim(data) # cross checking the dimension

[1] 41188 17

### Missing Variables

Variable Balance is not in the dataset !

### **Check for Duplicate Rows**

# Check for Duplicate Rows  
sum(duplicated(data))

[1] 1980

### **Missing values (NAs)**

# How many rows have missing data  
sum(!complete.cases(data))

[1] 10700

# How Many Rows Are Completely Missing Values In All Columns  
all.empty = rowSums(is.na(data))==ncol(data)  
sum(all.empty)

[1] 0

# Determine the number of missing values for each variable  
missing\_values <- sapply(data, function(x) sum(is.na(x)))  
missing\_values # Show the number of missing values for each variable

age job marital education default   
 0 330 80 1731 8597   
 housing loan month day\_of\_week campaign   
 990 990 0 0 0   
 previous poutcome emp.var.rate cons.price.idx cons.conf.idx   
 0 0 0 0 0   
 euribor3m y   
 0 0

## 2 Describe data: Examine the data and document its surface properties like data format, number of records, or field identities.

See *Variables Overview.xlsx* for further information!

### **Statistical Analysis**

#### **of numeric variables**

# statistical analysis of the data  
# statistical Summary of numeric variables  
summary\_numeric <- summary(data[, sapply(data, is.numeric)])   
summary\_numeric

age campaign previous emp.var.rate   
 Min. :17.00 Min. : 1.000 Min. :0.000 Min. :-3.40000   
 1st Qu.:32.00 1st Qu.: 1.000 1st Qu.:0.000 1st Qu.:-1.80000   
 Median :38.00 Median : 2.000 Median :0.000 Median : 1.10000   
 Mean :40.02 Mean : 2.568 Mean :0.173 Mean : 0.08189   
 3rd Qu.:47.00 3rd Qu.: 3.000 3rd Qu.:0.000 3rd Qu.: 1.40000   
 Max. :98.00 Max. :56.000 Max. :7.000 Max. : 1.40000   
 cons.price.idx cons.conf.idx euribor3m   
 Min. :92.20 Min. :-50.8 Min. :0.634   
 1st Qu.:93.08 1st Qu.:-42.7 1st Qu.:1.344   
 Median :93.75 Median :-41.8 Median :4.857   
 Mean :93.58 Mean :-40.5 Mean :3.621   
 3rd Qu.:93.99 3rd Qu.:-36.4 3rd Qu.:4.961   
 Max. :94.77 Max. :-26.9 Max. :5.045

# standard deviation of of numeric variables  
sd\_numeric <- sapply(data[, sapply(data, is.numeric)], sd)  
sd\_numeric

age campaign previous emp.var.rate cons.price.idx   
 10.4212500 2.7700135 0.4949011 1.5709597 0.5788400   
 cons.conf.idx euribor3m   
 4.6281979 1.7344474

#### **of discrete** **numeric variables (integers)**

# Frequency of the individual values for discrete numeric variables (integers)  
frequency\_int\_data <- lapply(data[, sapply(data, is.integer)], table)  
frequency\_int\_data

$age  
  
 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32   
 5 28 42 65 102 137 226 463 598 698 851 1001 1453 1714 1947 1846   
 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48   
1833 1745 1759 1780 1475 1407 1432 1161 1278 1142 1055 1011 1103 1030 928 979   
 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64   
 839 875 754 779 733 684 648 704 646 576 463 283 73 62 55 57   
 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80   
 44 57 26 33 34 47 53 34 34 32 24 34 20 27 14 31   
 81 82 83 84 85 86 87 88 89 91 92 94 95 98   
 20 17 17 7 15 8 1 22 2 2 4 1 1 2   
  
$campaign  
  
 1 2 3 4 5 6 7 8 9 10 11 12 13   
17642 10570 5341 2651 1599 979 629 400 283 225 177 125 92   
 14 15 16 17 18 19 20 21 22 23 24 25 26   
 69 51 51 58 33 26 30 24 17 16 15 8 8   
 27 28 29 30 31 32 33 34 35 37 39 40 41   
 11 8 10 7 7 4 4 3 5 1 1 2 1   
 42 43 56   
 2 2 1   
  
$previous  
  
 0 1 2 3 4 5 6 7   
35563 4561 754 216 70 18 5 1

# Unique values for discrete numeric variables (integers)  
unique\_int\_values <- sapply(data[, sapply(data, is.integer)], unique)  
unique\_int\_values

$age  
 [1] 56 57 37 40 45 59 41 24 25 29 35 54 46 50 39 30 55 49 34 52 58 32 38 44 42  
[26] 60 53 47 51 48 33 31 43 36 28 27 26 22 23 20 21 61 19 18 70 66 76 67 73 88  
[51] 95 77 68 75 63 80 62 65 72 82 64 71 69 78 85 79 83 81 74 17 87 91 86 98 94  
[76] 84 92 89  
  
$campaign  
 [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 19 18 23 14 22 25 16 17 15 20 56 39  
[26] 35 42 28 26 27 32 21 24 29 31 30 41 37 40 33 34 43  
  
$previous  
[1] 0 1 2 3 4 5 6 7

#### **of non-numeric variables**

# Frequency of individual categories for non-numeric variables including NA  
frequency\_data <- lapply(data[, sapply(data, is.character)], function(x) table(x, useNA = "always"))  
frequency\_data

$job  
x  
 admin. blue-collar entrepreneur housemaid management   
 10422 9254 1456 1060 2924   
 retired self-employed services student technician   
 1720 1421 3969 875 6743   
 unemployed <NA>   
 1014 330   
  
$marital  
x  
divorced married single <NA>   
 4612 24928 11568 80   
  
$education  
x  
 basic.4y basic.6y basic.9y high.school   
 4176 2292 6045 9515   
 illiterate professional.course university.degree <NA>   
 18 5243 12168 1731   
  
$default  
x  
 no yes <NA>   
32588 3 8597   
  
$housing  
x  
 no yes <NA>   
18622 21576 990   
  
$loan  
x  
 no yes <NA>   
33950 6248 990   
  
$month  
x  
 apr aug dec jul jun mar may nov oct sep <NA>   
 2632 6178 182 7174 5318 546 13769 4101 718 570 0   
  
$day\_of\_week  
x  
 fri mon thu tue wed <NA>   
7827 8514 8623 8090 8134 0   
  
$poutcome  
x  
 failure nonexistent success <NA>   
 4252 35563 1373 0   
  
$y  
x  
 no yes <NA>   
36548 4640 0

# Number of unique values for non-numeric variables  
unique\_data <- sapply(data[, sapply(data, is.character)], function(x) length(unique(x)))  
unique\_data

job marital education default housing loan   
 12 4 8 3 3 3   
 month day\_of\_week poutcome y   
 10 5 3 2

## 3. Explore data: Dig deeper into the data. Query it, visualize it, and identify relationships among the data.

### **Visualizations**

**Histograms**: Visualize the distribution of numeric variables.

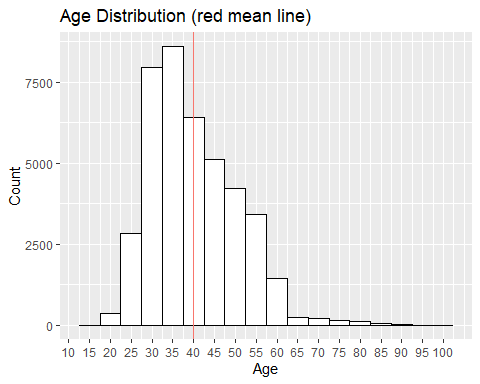
* **Density:** Density scales the height of the bars based on the relative frequency of data within each bin. It’s useful for comparing the shape of distributions across different datasets or variables, independent of the number of observations or bin width.
* **Frequency:** Frequency displays the absolute number of observations in each bin. It’s useful for understanding the exact count of observations in specific areas of the distribution, particularly for identifying outliers or regions with unusually high or low frequencies.

#### **AGE**

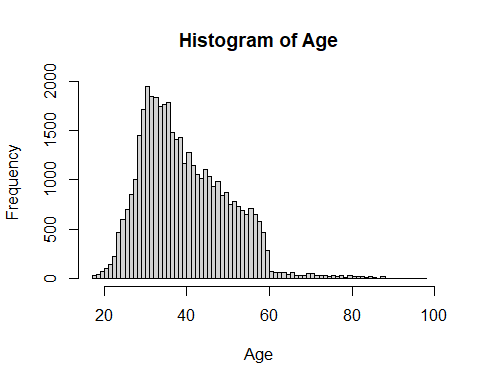
# stattistical summary of Age  
summary(data$age)

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 17.00 32.00 38.00 40.02 47.00 98.00

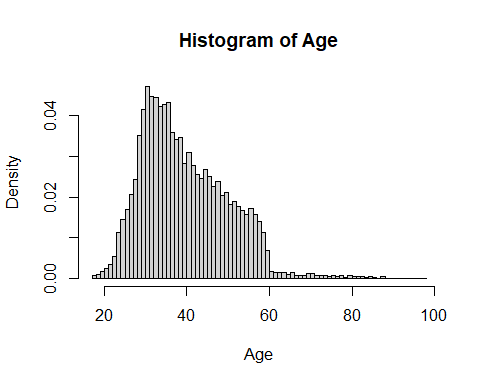
# Histogram count of Ages  
gg = ggplot (data)   
p1 = gg + geom\_histogram(aes(x=age),color="black", fill="white", binwidth = 5) +  
 ggtitle('Age Distribution (red mean line)') +  
 ylab('Count') +  
 xlab('Age') +  
 geom\_vline(aes(xintercept = mean(age), color = "red")) +  
 scale\_x\_continuous(breaks = seq(0,100,5)) +  
 theme(legend.position = "none")  
p1



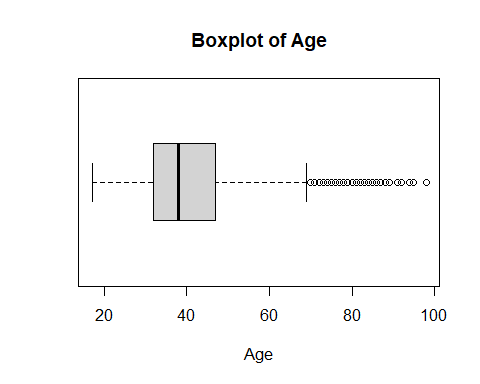
# Histogram y = Frequency, x = Age  
hist(data$age, main = "Histogram of Age", xlab = "Age", breaks = unique(data$age), freq = TRUE)



# Histogram y = Density, x = Age  
hist(data$age, main = "Histogram of Age", xlab = "Age", breaks = unique(data$age), freq = FALSE)

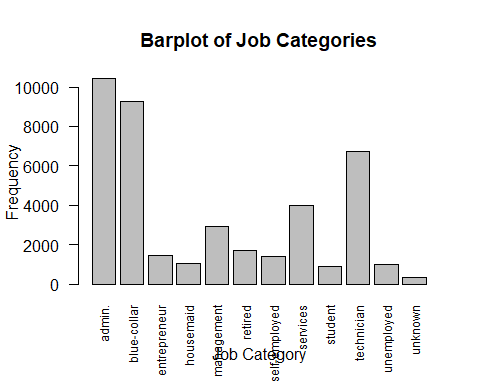


# Boxplot for Age  
boxplot(data$age, horizontal = TRUE, main = "Boxplot of Age", xlab = "Age")



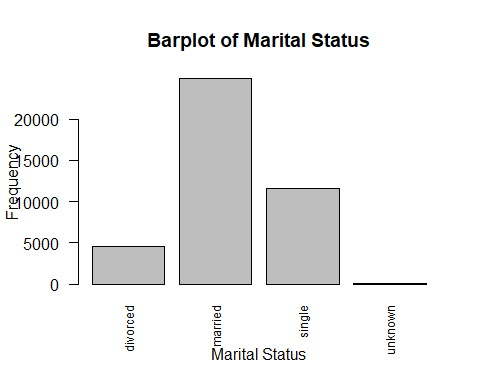
#### **JOB**

# Define Categories (so emtpy gets shown as unknown)  
categories\_job <- c('admin.', 'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired', 'self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown')  
  
# Barplot for Job Categories  
barplot(table(data$job, useNA = "always"), main = "Barplot of Job Categories", xlab = "Job Category", ylab = "Frequency", las = 2, cex.names = 0.75, names.arg = categories\_job, mgp = c(3, 1, 0))



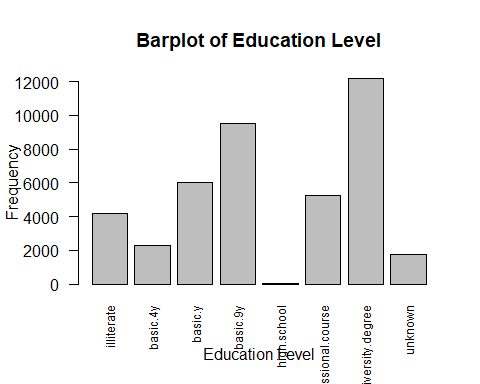
#### **MARITAL**

# Define Categories (so emtpy gets shown as unknown)  
categories\_marital <- c('divorced', 'married', 'single', 'unknown')  
  
# Barplot for Marital Status  
barplot(table(data$marital, useNA = "always"), main = "Barplot of Marital Status", xlab = "Marital Status", ylab = "Frequency", las = 2, cex.names = 0.75, names.arg = categories\_marital, mgp = c(3, 1, 0))



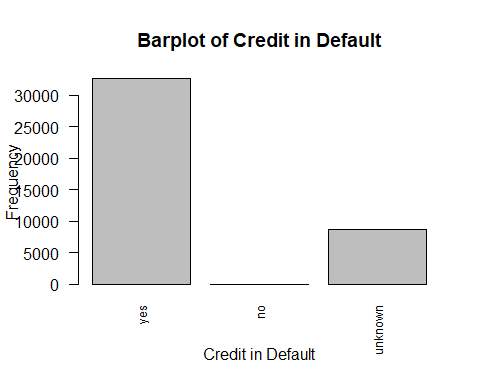
#### **EDUCATION**

# Define Categories (so emtpy gets shown as unknown)  
categories\_education <- c('illiterate', 'basic.4y', 'basic.y', 'basic.9y', 'high.school', 'professional.course', 'university.degree', 'unknown')  
  
# Barplot for Education Level  
barplot(table(data$education, useNA = "always"), main = "Barplot of Education Level", xlab = "Education Level", ylab = "Frequency", las = 2, cex.names = 0.75, names.arg = categories\_education, mgp = c(3, 1, 0))



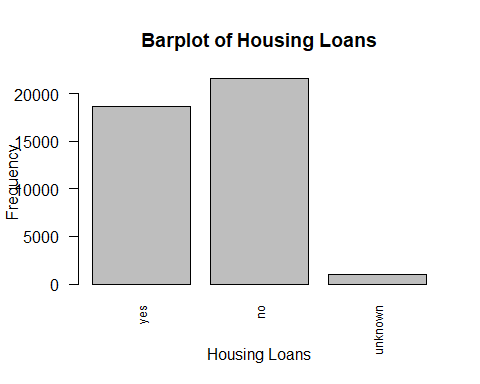
#### **DEFAULT**

# Define Categories (so emtpy gets shown as unknown)  
categories\_default <- c('yes', 'no', 'unknown')  
  
# Barplot for Credit in Default  
barplot(table(data$default, useNA = "always"), main = "Barplot of Credit in Default", xlab = "Credit in Default", ylab = "Frequency", las = 2, cex.names = 0.75, names.arg = categories\_default, mgp = c(3, 1, 0))



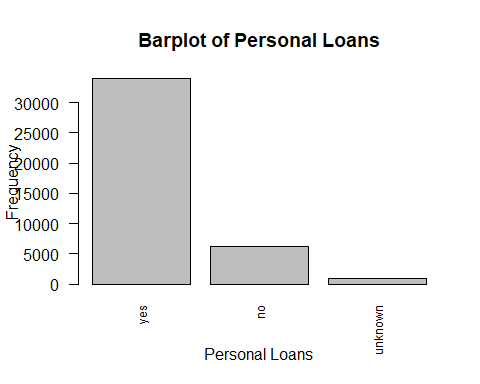
#### **HOUSING**

# Define Categories (so emtpy gets shown as unknown)  
categories\_housing <- c('yes', 'no', 'unknown')  
  
# Barplot for Housing Loans  
barplot(table(data$housing, useNA = "always"), main = "Barplot of Housing Loans", xlab = "Housing Loans", ylab = "Frequency", las = 2, cex.names = 0.75, names.arg = categories\_housing, mgp = c(3, 1, 0))



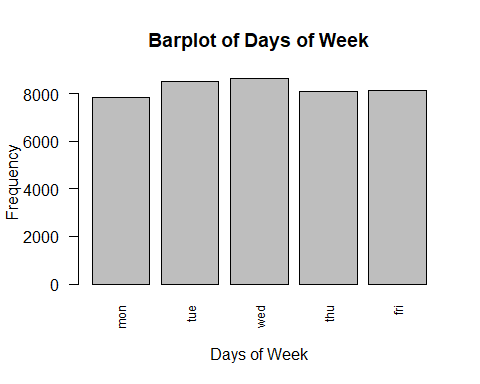
#### **LOAN**

# Define Categories (so emtpy gets shown as unknown)  
categories\_loan <- c('yes', 'no', 'unknown')  
  
# Barplot for Personal Loans  
barplot(table(data$loan, useNA = "always"), main = "Barplot of Personal Loans", xlab = "Personal Loans", ylab = "Frequency", las = 2, cex.names = 0.75, names.arg = categories\_loan, mgp = c(3, 1, 0))



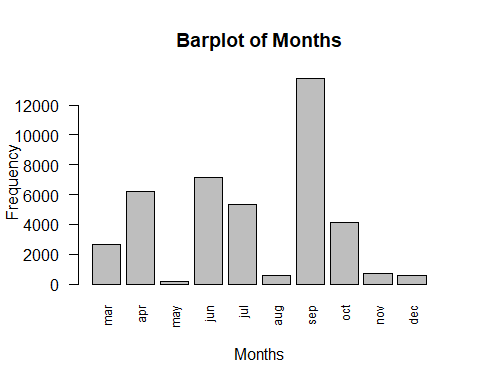
#### **DAY OF WEEK**

# Define Categories (for a nice order)  
categories\_days <- c('mon', 'tue', 'wed', 'thu', 'fri')  
  
  
# Barplot for Day of Week  
barplot(table(data$day\_of\_week), main = "Barplot of Days of Week", xlab = "Days of Week", ylab = "Frequency", las = 2, cex.names = 0.75, names.arg = categories\_days, mgp = c(3, 1, 0))



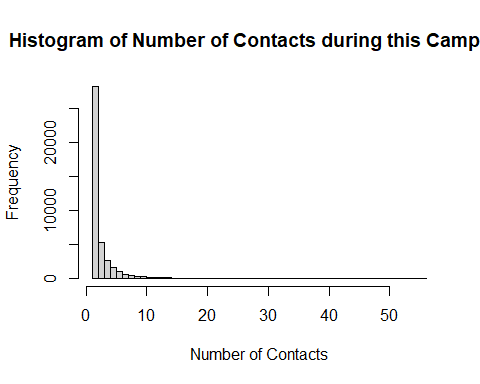
#### **MONTH**

# Define Categories (for a nice order)  
categories\_months <- c("mar", "apr", "may", "jun", "jul", "aug", "sep", "oct", "nov", "dec")  
  
# Barplot for Month  
barplot(table(data$month), main = "Barplot of Months", xlab = "Months", ylab = "Frequency", las = 2, cex.names = 0.75, names.arg = categories\_months, mgp = c(3, 1, 0))

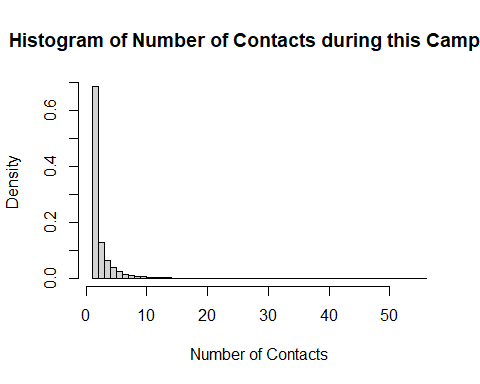


#### **CAMPAIN**

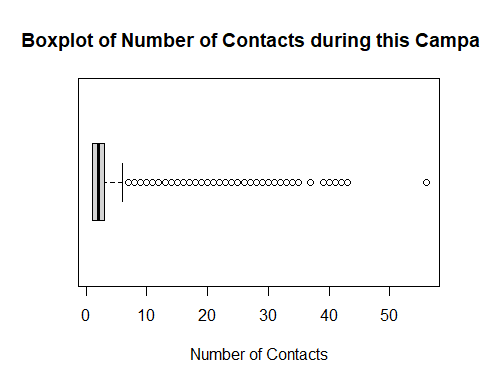
# Histogram y = Frequency, x = Campain  
hist(data$campaign, main = "Histogram of Number of Contacts during this Campain", xlab = "Number of Contacts", breaks = unique(data$campaign), freq = TRUE)



# Histogram y = Density, x = Campain  
hist(data$campaign, main = "Histogram of Number of Contacts during this Campain", xlab = "Number of Contacts", breaks = unique(data$campaign), freq = FALSE)

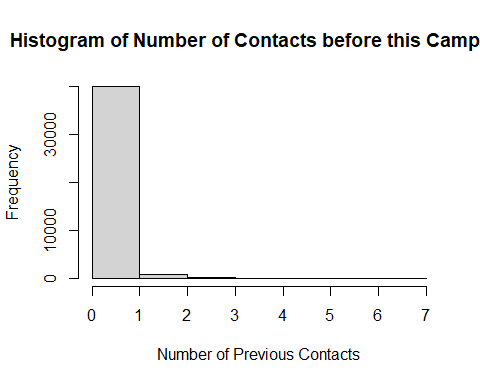


# Boxplot for Campain  
boxplot(data$campaign, horizontal = TRUE, main = "Boxplot of Number of Contacts during this Campain", xlab = "Number of Contacts")

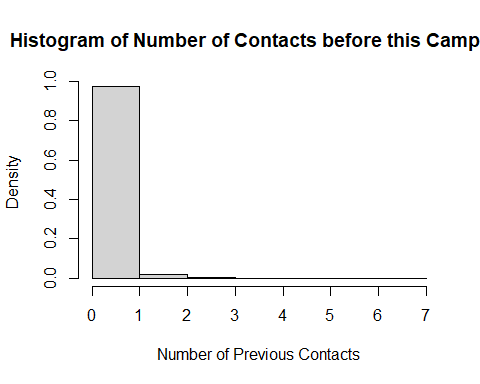


#### **PREVIOUS**

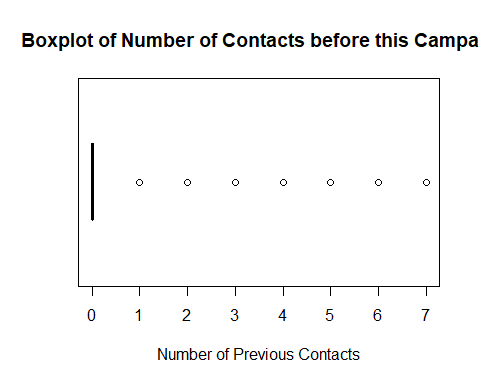
# Histogram y = Frequency, x = Previous  
hist(data$previous, main = "Histogram of Number of Contacts before this Campain", xlab = "Number of Previous Contacts", breaks = unique(data$previous), freq = TRUE)



# Histogram y = Density, x = Previous  
hist(data$previous, main = "Histogram of Number of Contacts before this Campain", xlab = "Number of Previous Contacts", breaks = unique(data$previous), freq = FALSE)

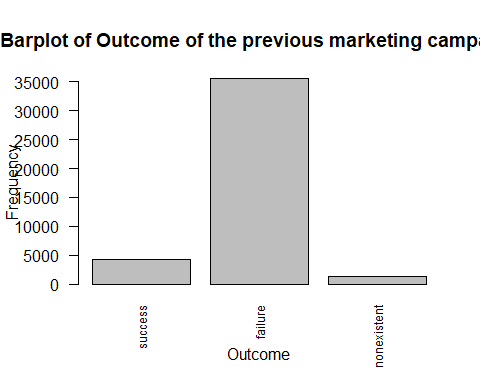


# Boxplot for Previous  
boxplot(data$previous, horizontal = TRUE, main = "Boxplot of Number of Contacts before this Campain", xlab = "Number of Previous Contacts")



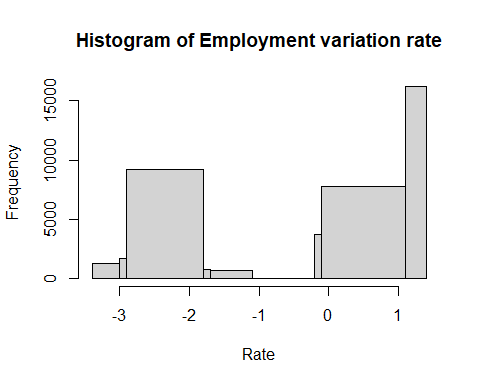
#### **POUTCOME**

# Define Categories (for a nice order)  
categories\_poutcome <- c("success", "failure", "nonexistent")  
  
# Barplot for Poutcome  
barplot(table(data$poutcome), main = "Barplot of Outcome of the previous marketing campaign", xlab = "Outcome", ylab = "Frequency", las = 2, cex.names = 0.75, names.arg = categories\_poutcome, mgp = c(3, 1, 0))

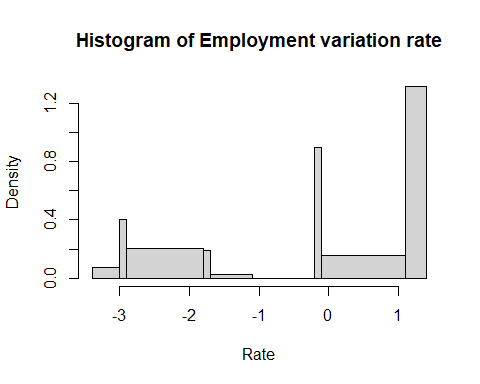


#### **Emp.var.rate =** Employment variation rate – quarterly indicator

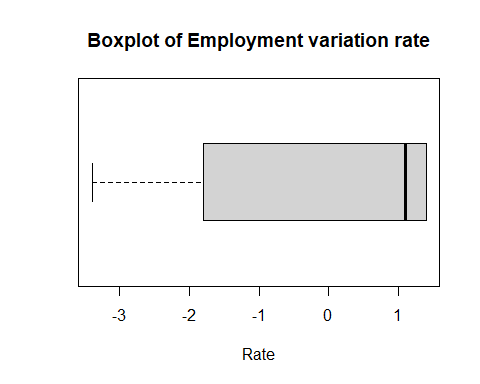
# Histogram y = Frequency, x = Emp.var.rate   
hist(data$emp.var.rate, main = "Histogram of Employment variation rate", xlab = "Rate", breaks = unique(data$emp.var.rate), freq = TRUE)



# Histogram y = Density, x = Emp.var.rate   
hist(data$emp.var.rate, main = "Histogram of Employment variation rate", xlab = "Rate", breaks = unique(data$emp.var.rate), freq = FALSE)

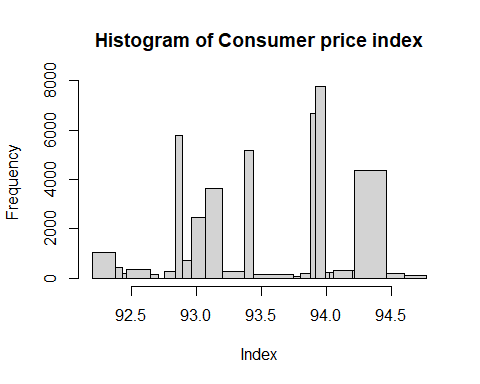


# Boxplot for Emp.var.rate   
boxplot(data$emp.var.rate, horizontal = TRUE, main = "Boxplot of Employment variation rate", xlab = "Rate")

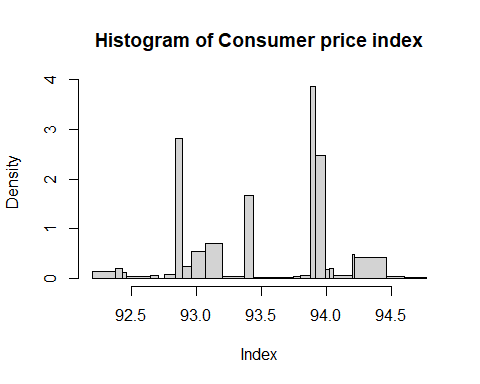


#### **Cons.price.idx =** Consumer price index – monthly indicator

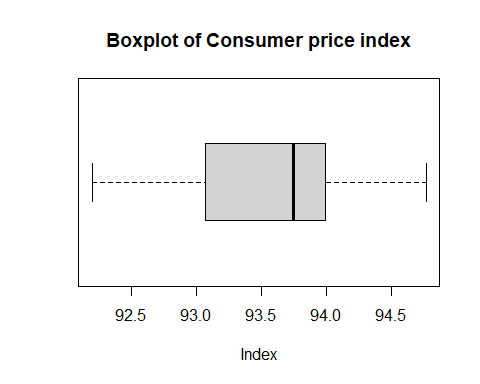
# Histogram y = Frequency, x = Cons.price.idx   
hist(data$cons.price.idx, main = "Histogram of Consumer price index", xlab = "Index", breaks = unique(data$cons.price.idx), freq = TRUE)



# Histogram y = Density, x = Cons.price.idx   
hist(data$cons.price.idx, main = "Histogram of Consumer price index", xlab = "Index", breaks = unique(data$cons.price.idx), freq = FALSE)

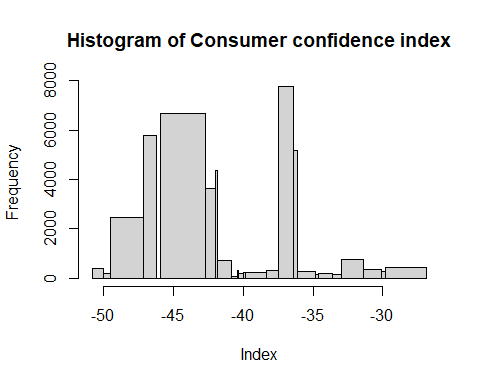


# Boxplot for Cons.price.idx  
boxplot(data$cons.price.idx, horizontal = TRUE, main = "Boxplot of Consumer price index", xlab = "Index")

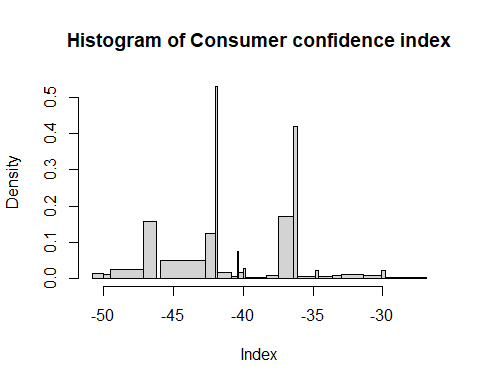


#### **Cons.conf.idx =** Consumer confidence index – monthly indicator

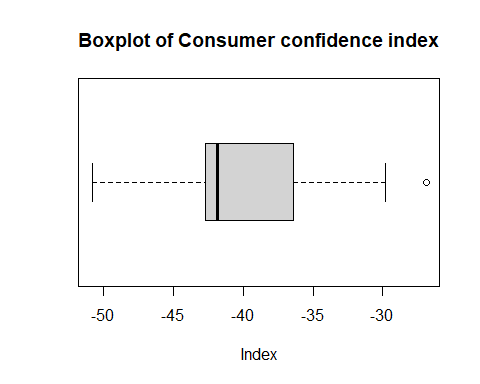
# Histogram y = Frequency, x = Cons.conf.idx   
hist(data$cons.conf.idx, main = "Histogram of Consumer confidence index", xlab = "Index", breaks = unique(data$cons.conf.idx), freq = TRUE)



# Histogram y = Density, x = Cons.conf.idx   
hist(data$cons.conf.idx, main = "Histogram of Consumer confidence index", xlab = "Index", breaks = unique(data$cons.conf.idx), freq = FALSE)

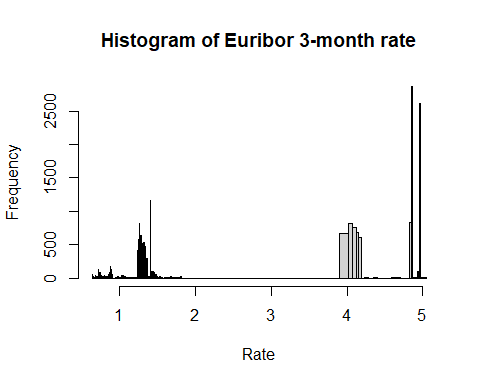


# Boxplot for Cons.price.idx  
boxplot(data$cons.conf.idx, horizontal = TRUE, main = "Boxplot of Consumer confidence index", xlab = "Index")

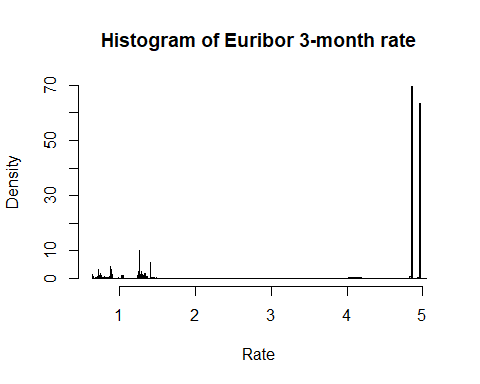


#### **Euribor3m =** Euribor 3-month rate – daily indicator

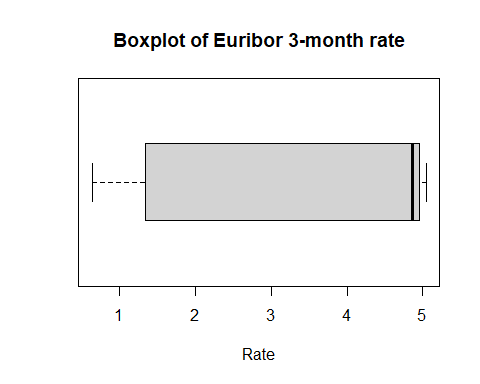
# Histogram y = Frequency, x = Euribor3m   
hist(data$euribor3m, main = "Histogram of Euribor 3-month rate", xlab = "Rate", breaks = unique(data$euribor3m), freq = TRUE)



# Histogram y = Density, x = Euribor3m   
hist(data$euribor3m, main = "Histogram of Euribor 3-month rate", xlab = "Rate", breaks = unique(data$euribor3m), freq = FALSE)

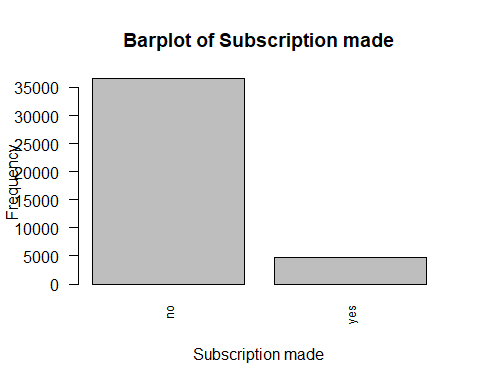


# Boxplot for Euribor3m   
boxplot(data$euribor3m, horizontal = TRUE, main = "Boxplot of Euribor 3-month rate", xlab = "Rate")



#### **Y** - Target Variable: Has the client subscribed a term deposit

# Barplot for Y  
barplot(table(data$y), main = "Barplot of Subscription made", xlab = "Subscription made", ylab = "Frequency", las = 2, cex.names = 0.75, mgp = c(3, 1, 0))



##### **Outcome Imbalance**

Observe that the dataset predicted outcome (y) is skewed towards ‘no’ with over 88.7%.

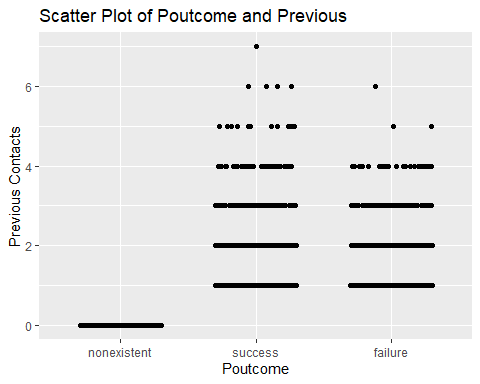
prop.table(table(data$y))

no yes   
0.8873458 0.1126542

### **Relationships between variables:**

#### previous & poutcome

# Reorder levels of poutcome  
data$poutcome <- factor(data$poutcome, levels = c("nonexistent", "success", "failure"))  
  
# Create the scatter plot  
ggplot(data, aes(x = poutcome, y = previous)) +  
 geom\_jitter(width = 0.3, height = 0) + # Add jitter to prevent overlapping  
 labs(title = "Scatter Plot of Poutcome and Previous", x = "Poutcome", y = "Previous Contacts")



##### previous = 0 & poutcome = nonexistent

# Relationship between previous = 0 & poutcome = nonexistent  
# Number of previous = 0  
count\_previous\_0 <- sum(data$previous == 0)   
count\_previous\_0

[1] 35563

# Number of poutcome ="nonexistent"  
count\_poutcome\_nonexistent <- sum(data$poutcome == "nonexistent")   
count\_poutcome\_nonexistent

[1] 35563

# Number of previous = 0 & poutcome = "nonexistent" at the same time  
count <- sum(data$poutcome[data$previous == 0] == "nonexistent")   
count

[1] 35563

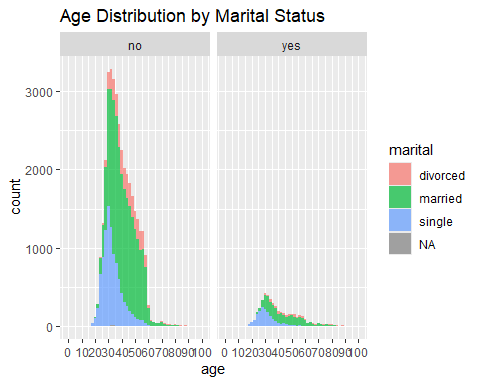
# check if relationship is true (all numbers are equal)  
realtionship <- (count\_previous\_0 == count\_poutcome\_nonexistent) && +  
 (count\_previous\_0 == count) && (count\_poutcome\_nonexistent == count)  
realtionship

[1] TRUE

### **Age Distribution vs Marital Status That Subscribes Term Deposit**

*The bulk of clients are married or divorced.* Sharp drop of clients above age 60 with marital status ‘divorced’ and ‘married’. \*Single clients drop in numbers above age 30-40.

ggplot(data, aes(x=age, fill=marital)) +   
 geom\_histogram(binwidth = 2, alpha=0.7, na.rm = TRUE) +  
 facet\_grid(cols = vars(y)) +  
 expand\_limits(x=c(0,100)) +  
 scale\_x\_continuous(breaks = seq(0,100,10)) +  
 ggtitle("Age Distribution by Marital Status")



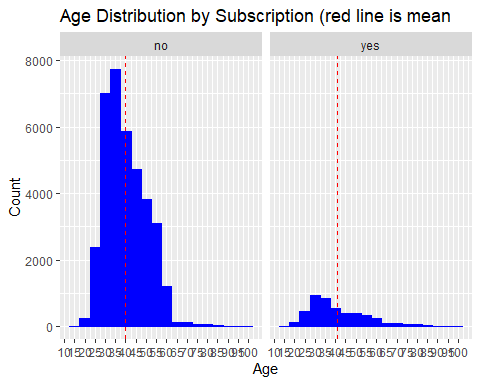
### **Age vs Subscription**

Most clients that subscribe are between age 25 to 55. Mean age for all clients is at age 40 years.

mean(data$age)

[1] 40.02406

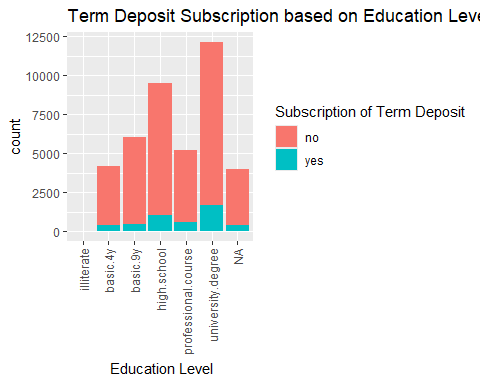
mu <- data %>% group\_by(y) %>% summarize(grp.mean=mean(age))  
  
ggplot (data, aes(x=age)) +   
 geom\_histogram(color = "blue", fill = "blue", binwidth = 5) +  
 facet\_grid(cols=vars(y)) +   
 ggtitle('Age Distribution by Subscription (red line is mean') + ylab('Count') + xlab('Age') +  
 scale\_x\_continuous(breaks = seq(0,100,5)) +  
 geom\_vline(data=mu, aes(xintercept=grp.mean), color="red", linetype="dashed")



### **Education vs Subscription**

Having higher education is seen to contribute to higher subscription of term deposit. Most clients who subscribe are from ‘high school’ and ‘university degree’ education levels.

# Define Categories (so emtpy gets shown as unknown)  
categories\_education <- c('illiterate', 'basic.4y', 'basic.y', 'basic.9y', 'high.school', 'professional.course', 'university.degree', 'unknown')  
  
ggplot(data = data, aes(x=factor(education, categories\_education), fill=y)) +  
 geom\_bar(na.rm = TRUE) +  
 ggtitle("Term Deposit Subscription based on Education Level") +  
 xlab(" Education Level") +  
 guides(fill=guide\_legend(title="Subscription of Term Deposit")) +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust=1))



The following table presents the number of subscriptions per education level. In total, 4640 clients have subscribed. The ‘percentage’ column showcases the proportion of subscribed clients relative to the total number of subscriptions, expressed as a percentage. Notably, individuals with higher education levels exhibit higher subscription rates: high school (22.22%), professional course (12.82%), and university degree (35.99%).

# categories to have the correct order  
categories\_education <- c('illiterate', 'basic.4y', 'basic.6y', 'basic.9y', 'high.school', 'professional.course', 'university.degree', 'unknown')  
  
# Filtering the data to obtain only customers with a subscription  
subscribed\_customers <- data[data$y == "yes", ]  
  
# Total number of subscribed customers  
total\_subscribed <- nrow(subscribed\_customers)  
  
# Grouping data by education level & counting subscribed customers in each category  
education\_counts <- subscribed\_customers %>%  
 group\_by(education) %>%  
 summarise(subscribed\_customers = n())  
  
# Calculating percentage of subscribed customers for each education level  
education\_counts <- education\_counts %>%  
 mutate(percentage = round(subscribed\_customers / total\_subscribed \* 100, 2))  
  
# Sorting data according to the order of categories  
education\_counts <- education\_counts[order(match(education\_counts$education, categories\_education)), ]  
  
# Printing results  
print(education\_counts)

# A tibble: 8 × 3  
 education subscribed\_customers percentage  
 <chr> <int> <dbl>  
1 illiterate 4 0.09  
2 basic.4y 428 9.22  
3 basic.6y 188 4.05  
4 basic.9y 473 10.2   
5 high.school 1031 22.2   
6 professional.course 595 12.8   
7 university.degree 1670 36.0   
8 <NA> 251 5.41

The table below displays the number of subscriptions and total clients across different education levels. The ‘percentage’ column indicates the proportion of subscribed clients per education level relative to the total number of clients per education level, presented as a percentage. Notably, individuals with higher education levels exhibit higher subscription rates, such as those with a high school education (10.84%), professional course (11.35%), and university degree (13.72%).

However, it’s essential to consider the context. For instance, the illiterate group shows a high subscription percentage (22.22%), but this is based on a small sample size of only 18 out of 41188 clients contacted. Similarly, the 14.50% subscription rate among clients with an unknown education level appears relatively high, yet this is based on a relatively small group of 1731 clients with an unknown education level.

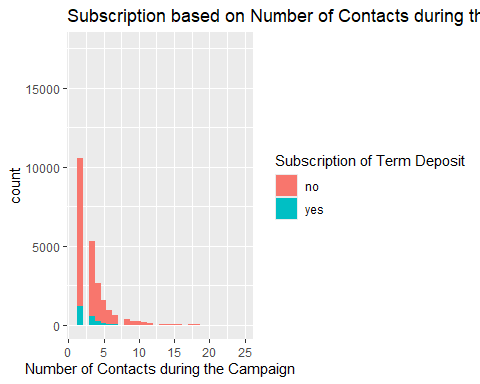
# Define order of education categories  
categories\_education <- c('illiterate', 'basic.4y', 'basic.6y', 'basic.9y', 'high.school', 'professional.course', 'university.degree', 'unknown')  
  
# Filter data to obtain only customers with successful subscriptions  
subscribed\_customers <- data[data$y == "yes", ]  
  
# Calculate total number of subscribed customers  
total\_subscribed <- nrow(subscribed\_customers)  
  
# Group data by education level & count subscribed customers in each category  
subscription\_counts <- subscribed\_customers %>%  
 group\_by(education) %>%  
 summarise(subscribed = n())  
  
# Calculate total number of customers per education level  
total\_clients <- data %>%  
 group\_by(education) %>%  
 summarise(total = n())  
  
# Merge the two datasets  
subscription\_summary <- merge(subscription\_counts, total\_clients, by = "education", all = TRUE)  
  
# Calculate percentage of subscriptions relative to total number of customers  
subscription\_summary$percentage <- round(with(subscription\_summary, subscribed / total \* 100), 2)  
  
# Adjust order of rows based on predefined education categories  
subscription\_summary <- subscription\_summary %>%  
 arrange(match(education, categories\_education))  
  
# Print results  
print(subscription\_summary)

education subscribed total percentage  
1 illiterate 4 18 22.22  
2 basic.4y 428 4176 10.25  
3 basic.6y 188 2292 8.20  
4 basic.9y 473 6045 7.82  
5 high.school 1031 9515 10.84  
6 professional.course 595 5243 11.35  
7 university.degree 1670 12168 13.72  
8 <NA> 251 1731 14.50

### **Subscription based on Number of Contact during Campaign**

It can be observed from barchart that there will be nearly no subscription beyond 10 contact during the campaign. Future campaign could improve resource utilization by setting limits to contacts during a campaign. Future campaigns can focus on first 3 contacts as it will have higher subscription rate.

ggplot(data=data, aes(x=campaign, fill=y))+  
 geom\_histogram(na.rm = TRUE, bins = 30)+  
 ggtitle("Subscription based on Number of Contacts during the Campaign")+  
 xlab("Number of Contacts during the Campaign")+  
 xlim(c(min=1,max=25)) +  
 guides(fill=guide\_legend(title="Subscription of Term Deposit"))



# Grouping data by campaign number and counting customers in each category  
campaign\_counts <- data %>%  
 group\_by(campaign) %>%  
 summarise(count\_campaign = n(),  
 count\_y\_yes = sum(y == "yes"),  
 count\_y\_no = sum(y == "no"))  
  
# Output results  
print(campaign\_counts)

# A tibble: 42 × 4  
 campaign count\_campaign count\_y\_yes count\_y\_no  
 <int> <int> <int> <int>  
 1 1 17642 2300 15342  
 2 2 10570 1211 9359  
 3 3 5341 574 4767  
 4 4 2651 249 2402  
 5 5 1599 120 1479  
 6 6 979 75 904  
 7 7 629 38 591  
 8 8 400 17 383  
 9 9 283 17 266  
10 10 225 12 213  
# ℹ 32 more rows

xxxxxx hier weiter machen

## 4. Verify data quality: How clean/dirty is the data? Document any quality issues.

17 variables (columns) and 41’188 observations (rows)

There is some missing variables (balance), and values, also wome rows are duplicated.

-> missing variable balance: is that on purpose?

-> missing values: some/all should be filled with default value (example mean).

-> delete duplicated rows.

Types: chr, int, and num

-> some types might need to be changed e.g. as.factor(), ex. : housing : Factor w/ 2 levels “no”,“yes”: 1 2