DATA 622 Assignment 1: Exploratory Data Analysis

Kevin Kirby

2025-02-27

```
library(tidyverse)
library(caret)
library(ggplot2)
library(ggcorrplot)
library(dplyr)
```

Overview

The assignment states: "This assignment focuses on one of the most important aspects of data science, Exploratory Data Analysis (EDA). Many surveys show that data scientists spend 60-80% of their time on data preparation. EDA allows you to identify data gaps & data imbalances, improve data quality, create better features and gain a deep understanding of your data before doing model training - and that ultimately helps train better models. In machine learning, there is a saying -"better data beats better algorithms" - meaning that it is more productive to spend time improving data quality than improving the code to train the model."

A Portuguese bank conducted a marketing campaign (phone calls) to predict if a client will subscribe to a term deposit The records of their efforts are available in the form of a dataset. The objective here is to apply machine learning techniques to analyze the dataset and figure out most effective tactics that will help the bank in next campaign to persuade more customers to subscribe to the bank's term deposit. Download the Bank Marketing Dataset."

This report contains the following sections: * Exploratory Data Analysis * Algorithm Selection * Pre-processing

I downloaded the dataset and uploaded the files to my Google Cloud Platform instance. These are public URLs that automatically download the file if clicked.

```
bank_additional_full_df <- read_csv2('https://storage.googleapis.com/data_science_masters_files/data_622
bank_additional_names_txt <- readLines('https://storage.googleapis.com/data_science_masters_files/data_6
bank_additional_df <- read_csv2('https://storage.googleapis.com/data_science_masters_files/data_622/bank
bank_full_df <- read_csv2('https://storage.googleapis.com/data_science_masters_files/data_622/bank%2Bmar
bank_names_txt <- readLines('https://storage.googleapis.com/data_science_masters_files/data_622/bank%2Bm
bank_df <- read_csv2('https://storage.googleapis.com/data_science_masters_files/data_622/bank%2Bmarketin
```

Exploratory Data Analysis

Correlation

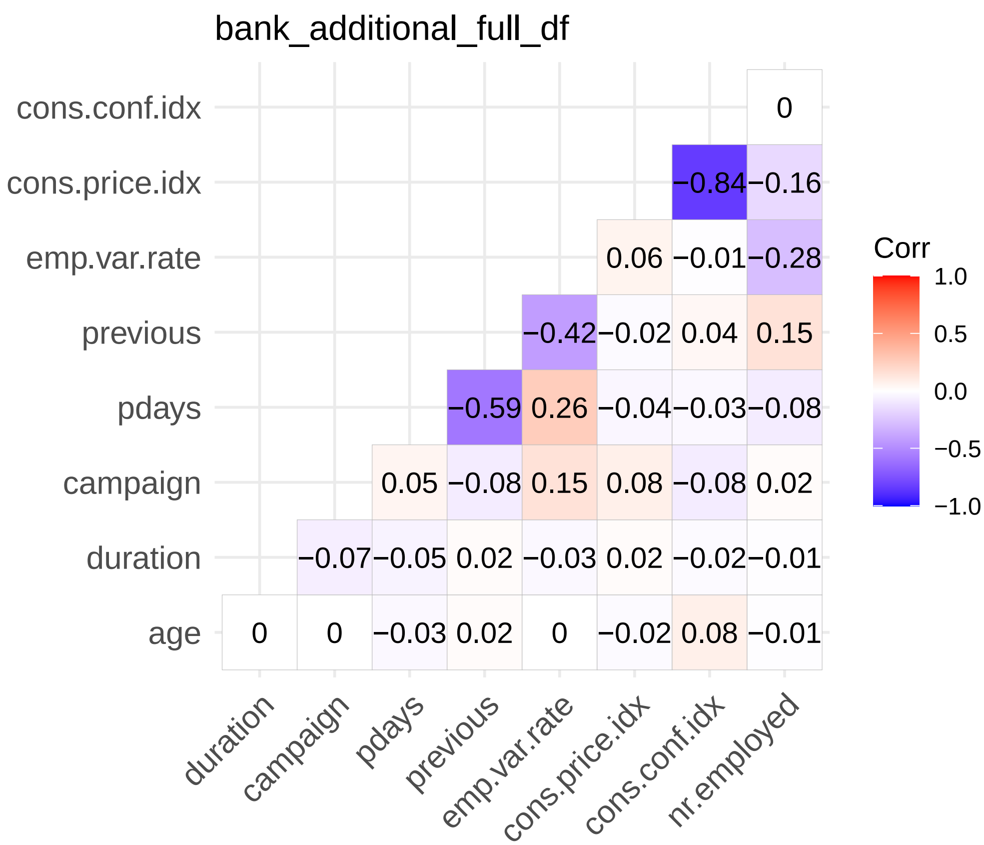
I will start with a review of correlation:

```
bank_addfull_num_df <- bank_additional_full_df[, sapply(bank_additional_full_df, is.numeric)]
bank_addfull_cat_df <- bank_additional_full_df[, sapply(bank_additional_full_df, is.factor)]
bank_addfull_num_matrix <- cor(bank_addfull_num_df, use = "pairwise.complete.obs")
bank_addfull_high_corr <- colnames(bank_addfull_num_df)[findCorrelation(bank_addfull_num_matrix, cutoff
```

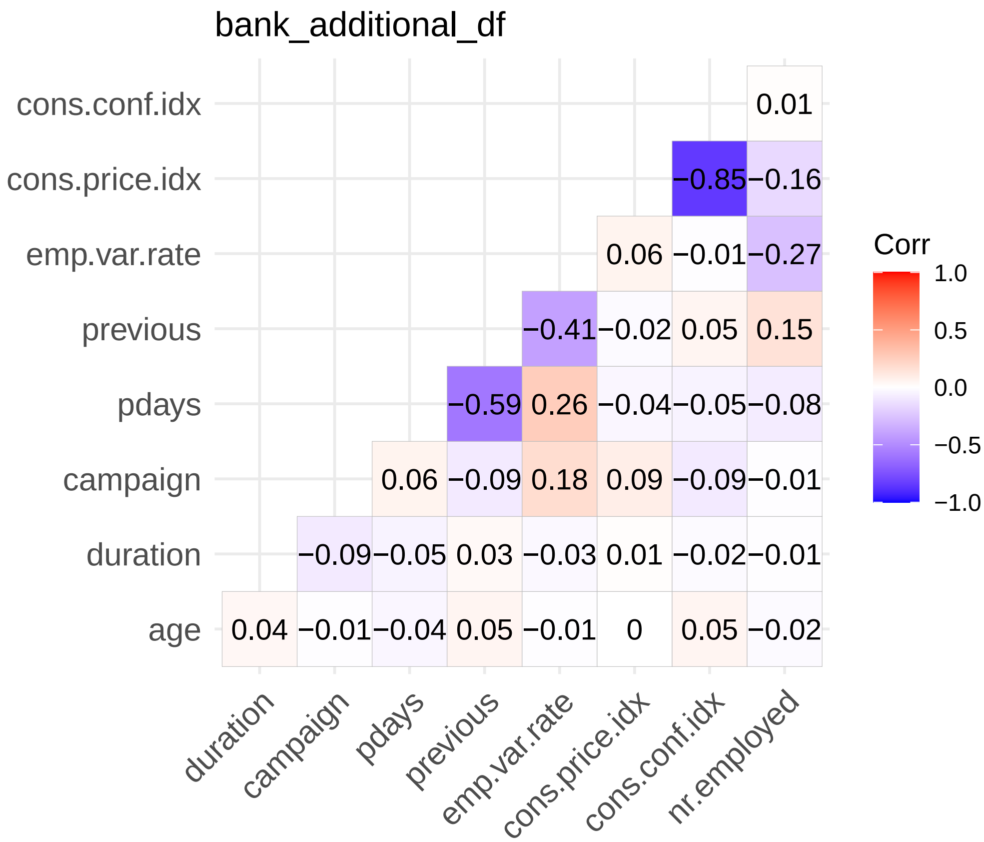
```r
bank_additional_num_df <- bank_additional_df[, sapply(bank_additional_df, is.numeric)]
bank_additional_cat_df <- bank_additional_df[, sapply(bank_additional_df, is.factor)]
bank_additional_num_matrix <- cor(bank_additional_num_df, use = "pairwise.complete.obs")
bank_additional_high_corr <- colnames(bank_additional_num_df)[findCorrelation(bank_additional_num_matri

bank_full_num_df <- bank_full_df[, sapply(bank_full_df, is.numeric)]
bank_full_cat_df <- bank_full_df[, sapply(bank_full_df, is.factor)]
bank_full_num_matrix <- cor(bank_full_num_df, use = "pairwise.complete.obs")
bank_full_high_corr <- colnames(bank_full_num_df)[findCorrelation(bank_full_num_matrix, cutoff = 0.75)]


bank_num_df <- bank_df[, sapply(bank_df, is.numeric)]
bank_cat_df <- bank_df[, sapply(bank_df, is.factor)]
bank_num_matrix <- cor(bank_num_df, use = "pairwise.complete.obs")
bank_high_corr <- colnames(bank_num_df)[findCorrelation(bank_num_matrix, cutoff = 0.75)]
```

The following features have reasonably high correlation, where reasonably high is greater than or equal to 75%: * Consumer Price Idex and Consumer Confidence Index

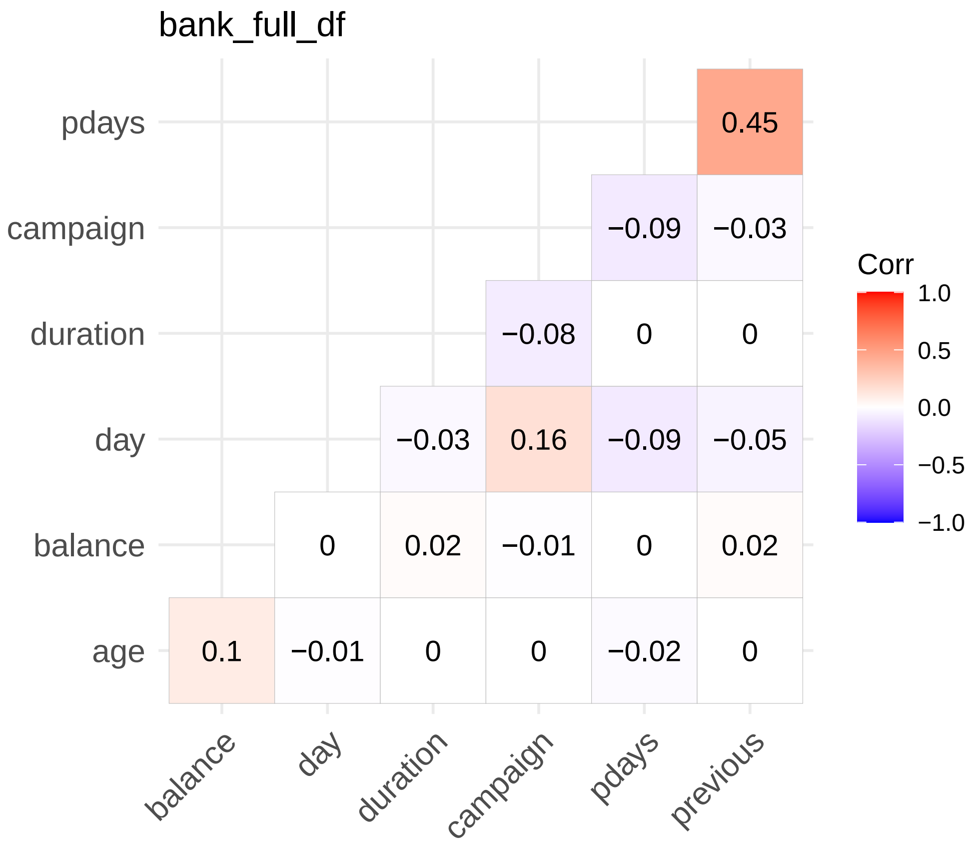
```r
ggcorrplot(bank_addfull_num_matrix, type = "lower", lab = TRUE) + ggtitle("bank_additional_full_df")
```



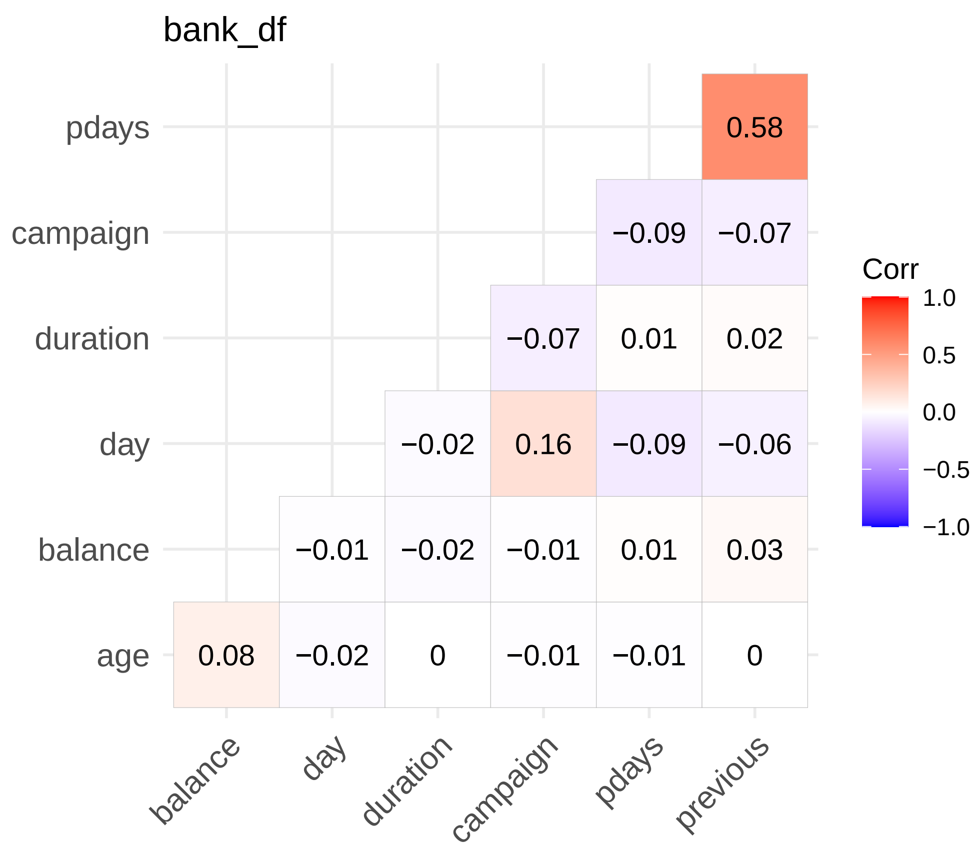

```r
ggcorrplot(bank_additional_num_matrix, type = "lower", lab = TRUE) + ggtitle("bank_additional_df")
```




```
ggcorrplot(bank_full_num_matrix, type = "lower", lab = TRUE) + ggtitle("bank_full_df")
```



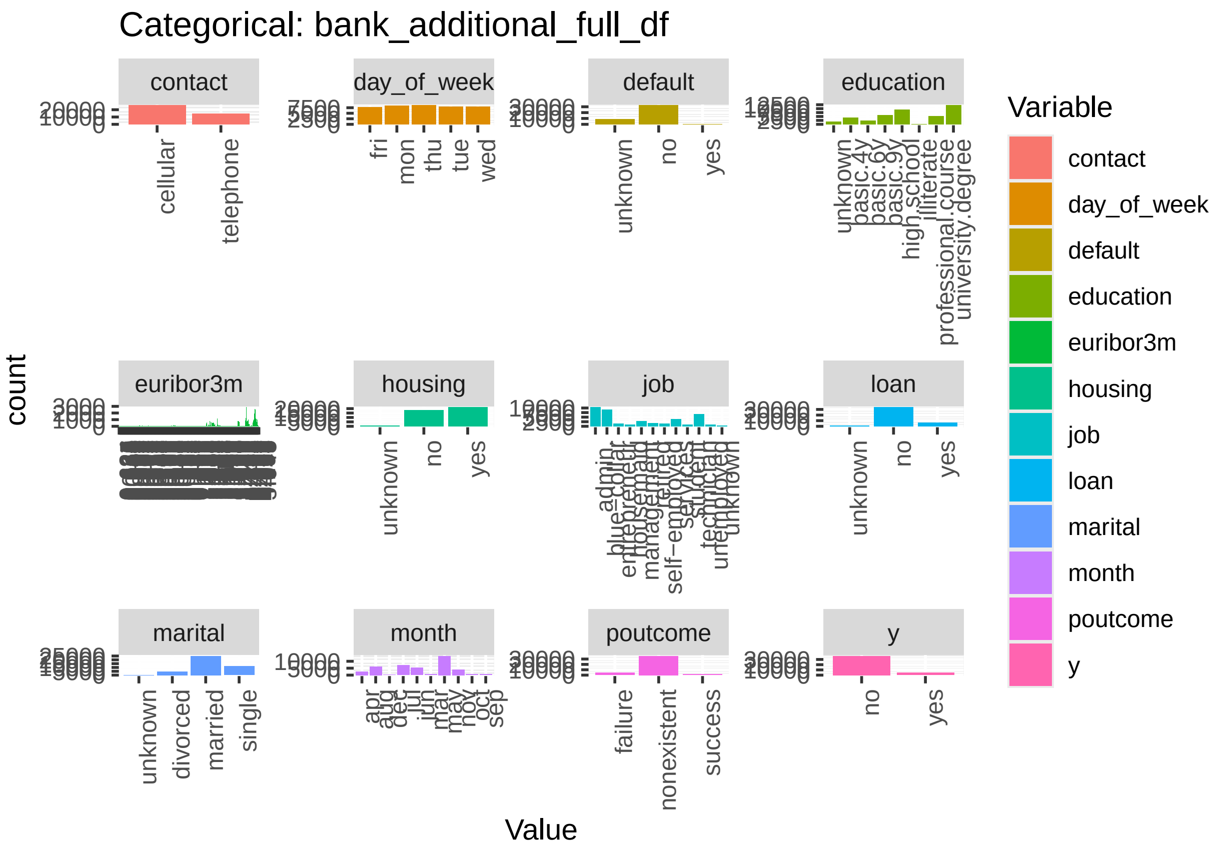

```r
ggcorrplot(bank_num_matrix, type = "lower", lab = TRUE) + ggtitle("bank_df")
```



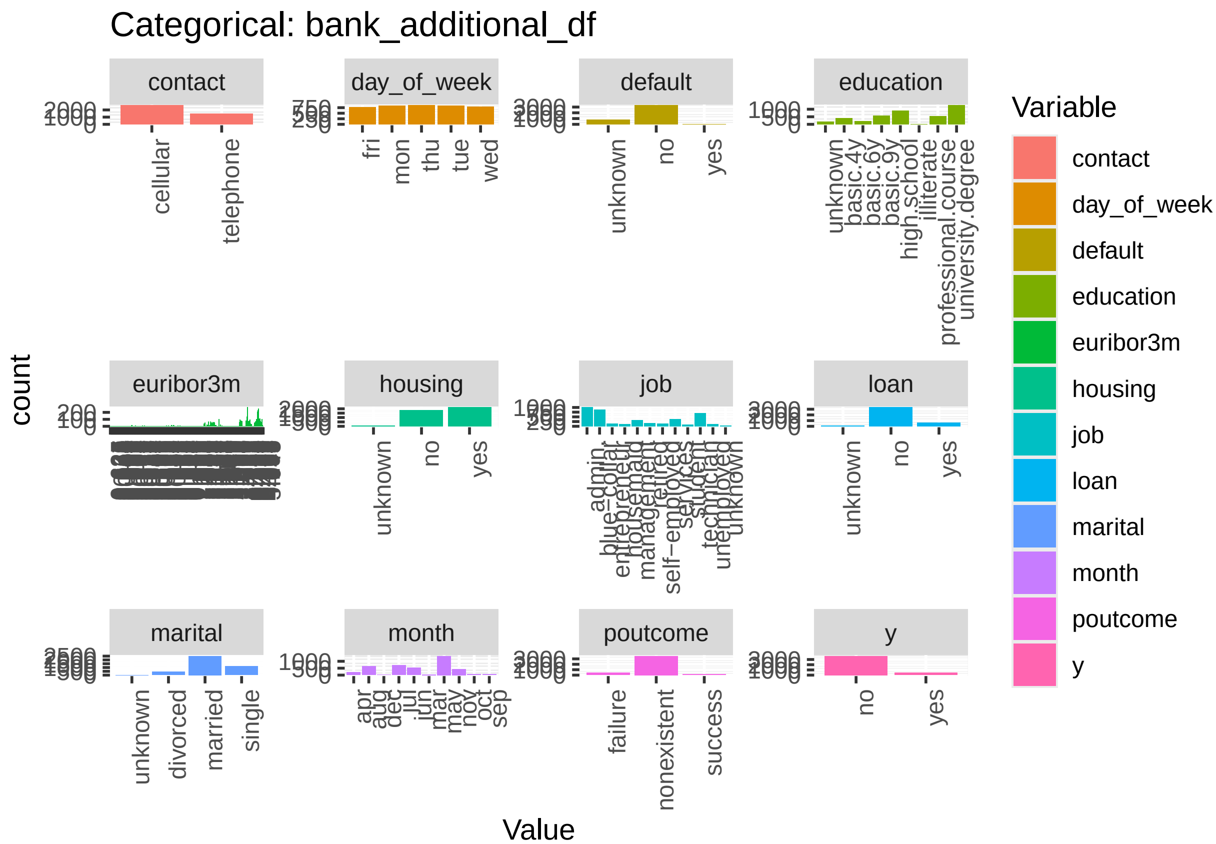
Overall distribution, patterns, trends

The datasets show a relatively normal distribution of the age of people contacted while the typical number of contacts and duration. It's interesting that the biggest push for campaigns was during May, I wonder if that's a popular time of year for new banking services in Portugal. I was first confused by the "admin" job category and why it was so high but then I realized it mudt be the category for non-management white collar workers.

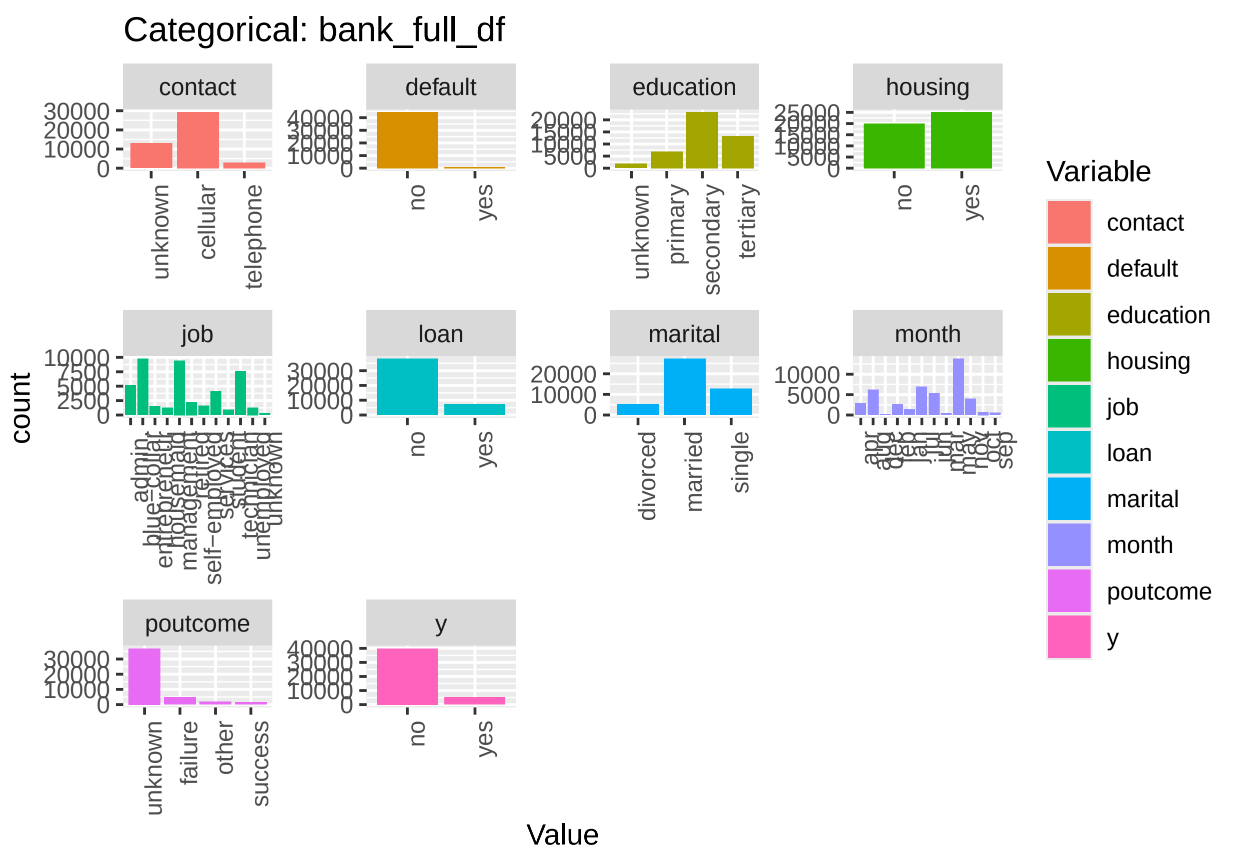
```
cat_chart <- function(df, title) {
  df <- df %>% mutate(across(where(is.character), as.factor))
  cat_df <- df %>% select(where(is.factor))

  if (ncol(cat_df) == 0) {
    message("No categorical variables")
    return(NULL)
  }

  cat_df %>%
    pivot_longer(everything(), names_to = "Variable", values_to = "Value") %>%
    ggplot(aes(x = Value, fill = Variable)) +
    geom_bar() +
    facet_wrap(~ Variable, scales = "free") +
    ggtitle(title) +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
}

cat_chart(bank_additional_full_df, "Categorical: bank_additional_full_df")
```



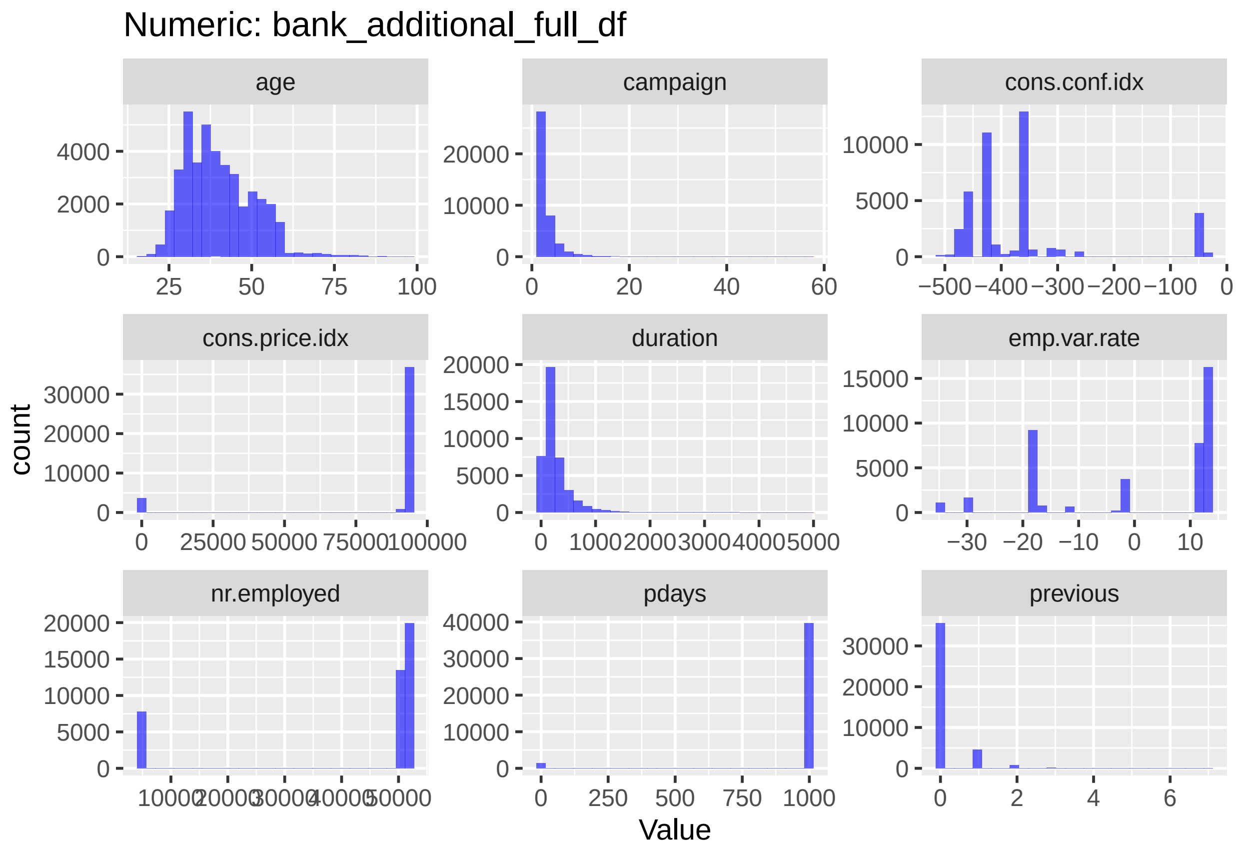

```
cat_chart(bank_additional_df, "Categorical: bank_additional_df")
```



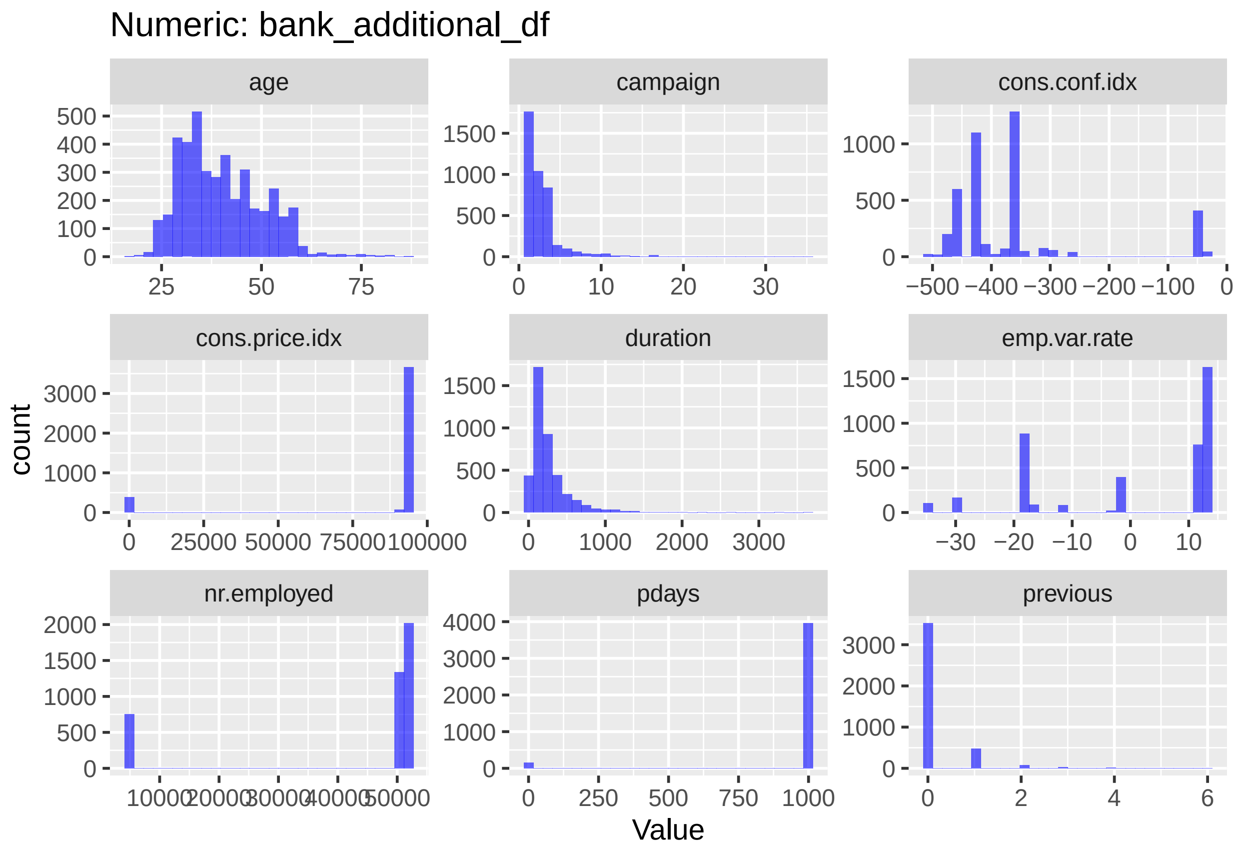

```r
cat_chart(bank_full_df, "Categorical: bank_full_df")
```



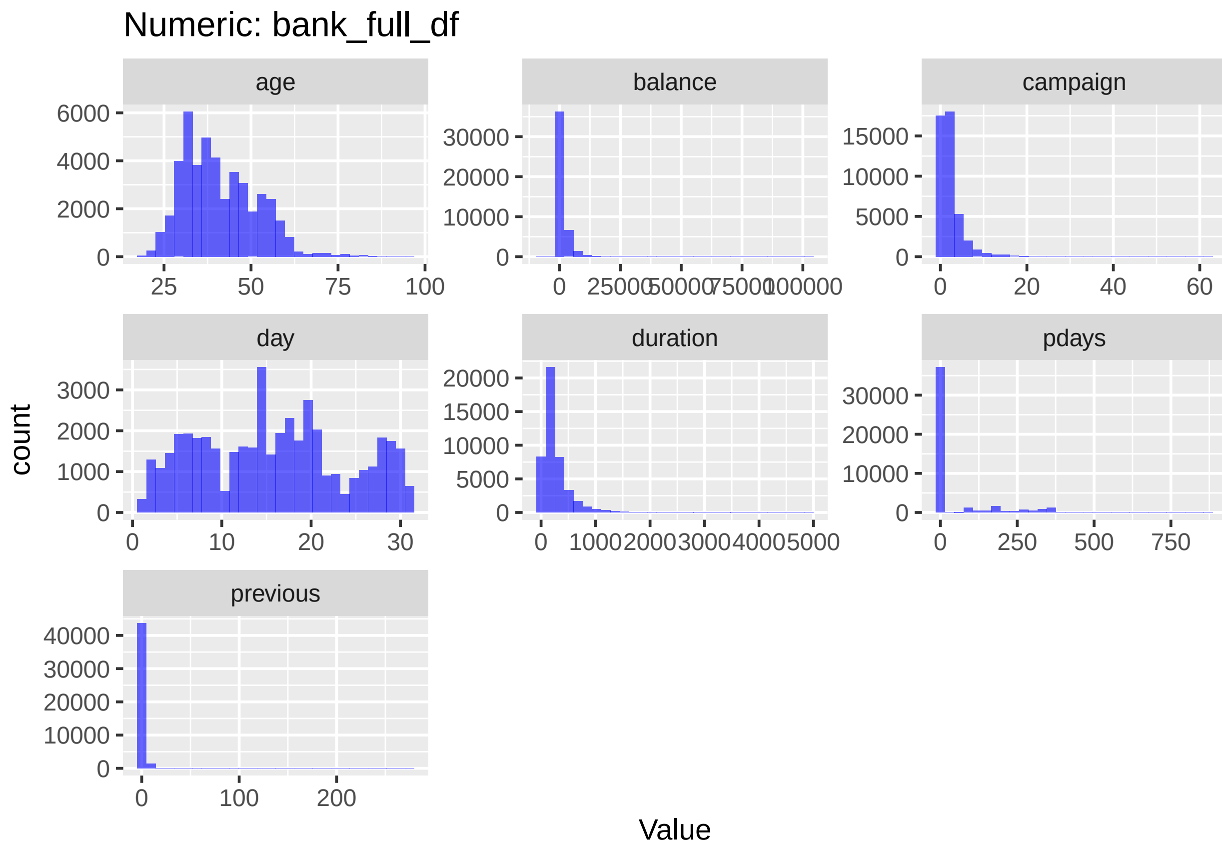

```r
dist_chart <- function(df, title) {
  df %>%
    select(where(is.numeric)) %>%
    pivot_longer(everything(), names_to = "Variable", values_to = "Value") %>%
    ggplot(aes(x = Value)) +
    geom_histogram(bins = 30, fill = "blue", alpha = 0.6) +
    facet_wrap(~ Variable, scales = "free") +
    ggtitle(title)
}

dist_chart(bank_additional_full_df, "Numeric: bank_additional_full_df")
```




```
dist_chart(bank_additional_df, "Numeric: bank_additional_df")
```




```
dist_chart(bank_full_df, "Numeric: bank_full_df")
```



Outliers

```
find_out <- function(df) {
  num_df <- df[, sapply(df, is.numeric)]
  lapply(num_df, function(x) boxplot.stats(x)$out)
}

summ_out <- function(df) {
  num_df <- df[, sapply(df, is.numeric)]
  data.frame(Variable = names(num_df),
             Outlier_Count = sapply(num_df, function(x) length(boxplot.stats(x)$out)))
}

out_addfull <- find_out(bank_additional_full_df)
summ_addfull <- summ_out(bank_additional_full_df)

cat("bank_additional_full_df outlier count:\n")
```

```
## bank_additional_full_df outlier count:
```

```
print(summ_addfull)
```

```
##                      Variable Outlier_Count
## age                       age           469
## duration             duration          2963
## campaign             campaign          2406
## pdays                   pdays          1515
```

```
## previous               previous          5625
## emp.var.rate       emp.var.rate             0
## cons.price.idx cons.price.idx          3616
## cons.conf.idx   cons.conf.idx          4282
## nr.employed       nr.employed          7763
```

```r
out_additional <- find_out(bank_additional_df)
summ_additional <- summ_out(bank_additional_df)

cat("bank_additional_df outlier count:\n")
```

```
## bank_additional_df outlier count:
```

```r
print(summ_additional)
```

```
##                       Variable Outlier_Count
## age                        age            39
## duration              duration           291
## campaign              campaign           235
## pdays                    pdays           160
## previous              previous           596
## emp.var.rate      emp.var.rate             0
## cons.price.idx cons.price.idx           386
## cons.conf.idx    cons.conf.idx           455
## nr.employed        nr.employed           758
```

```r
out_full <- find_out(bank_full_df)
summ_full <- summ_out(bank_full_df)

cat("bank_full_df outlier count:\n")
```

```
## bank_full_df outlier count:
```

```r
print(summ_full)
```

```
##          Variable Outlier_Count
## age           age           487
## balance   balance          4729
## day           day             0
## duration duration          3235
## campaign campaign          3064
## pdays       pdays          8257
## previous previous          8257
```

```r
out_bank <- find_out(bank_df)
summ_bank <- summ_out(bank_df)

cat("bank_df outlier count:\n")
```

```
## bank_df outlier count:
```

```r
print(summ_bank)
```

```
##          Variable Outlier_Count
## age           age            38
## balance   balance           506
## day           day             0
## duration duration           330
```

```
## campaign campaign           318
## pdays       pdays           816
## previous previous           816
```

There are a lot of outliers in the duration and campaign variables, speaking to the long tail of marketing camppaigns. The high number for previous contacts also suggests widely variety in how many times a given person is contacted.

Central tendency and spread

The bank balances have the largest spread, covering 9.2 million. While age has a very tight interquartile range of 15, which reaffirms the above seen normal looking distribution, the campaign value is of two along with a median and SD of 2.5 idicates extremely high clustering of values around 2.

```r
options(scipen = 999)

cat("bank_additional_full_df outlier count:\n")
```

```
## bank_additional_full_df outlier count:
```

```r
bank_additional_full_df %>%
  summarise(across(where(is.numeric), list(
    mean = ~mean(., na.rm = TRUE),
    median = ~median(., na.rm = TRUE),
    sd = ~sd(., na.rm = TRUE),
    var = ~var(., na.rm = TRUE),
    IQR = ~IQR(., na.rm = TRUE),
    range = ~max(., na.rm = TRUE) - min(., na.rm = TRUE)
  ))) %>%
  pivot_longer(cols = everything(), names_to = "Metric", values_to = "Value")
```

```
## # A tibble: 54 x 2
##    Metric            Value
##    <chr>             <dbl>
##  1 age_mean           40.0
##  2 age_median         38
##  3 age_sd             10.4
##  4 age_var           109.
##  5 age_IQR            15
##  6 age_range          81
##  7 duration_mean     258.
##  8 duration_median   180
##  9 duration_sd       259.
## 10 duration_var    67226.
## # i 44 more rows
```

```r
cat("bank_additional_df outlier count:\n")
```

```
## bank_additional_df outlier count:
```

```r
bank_additional_df %>%
  summarise(across(where(is.numeric), list(
    mean = ~mean(., na.rm = TRUE),
    median = ~median(., na.rm = TRUE),
    sd = ~sd(., na.rm = TRUE),
    var = ~var(., na.rm = TRUE),
    IQR = ~IQR(., na.rm = TRUE),
```

```r
    range = ~max(., na.rm = TRUE) - min(., na.rm = TRUE)
  ))) %>%
  pivot_longer(cols = everything(), names_to = "Metric", values_to = "Value")
```

```
## # A tibble: 54 x 2
##    Metric             Value
##    <chr>              <dbl>
##  1 age_mean            40.1
##  2 age_median          38
##  3 age_sd              10.3
##  4 age_var            106.
##  5 age_IQR             15
##  6 age_range           70
##  7 duration_mean      257.
##  8 duration_median    181
##  9 duration_sd        255.
## 10 duration_var     64874.
## # i 44 more rows
```

```r
cat("bank_full_df outlier count:\n")
```

```
## bank_full_df outlier count:
```

```r
bank_full_df %>%
  summarise(across(where(is.numeric), list(
    mean = ~mean(., na.rm = TRUE),
    median = ~median(., na.rm = TRUE),
    sd = ~sd(., na.rm = TRUE),
    var = ~var(., na.rm = TRUE),
    IQR = ~IQR(., na.rm = TRUE),
    range = ~max(., na.rm = TRUE) - min(., na.rm = TRUE)
  ))) %>%
  pivot_longer(cols = everything(), names_to = "Metric", values_to = "Value")
```

```
## # A tibble: 42 x 2
##    Metric               Value
##    <chr>                <dbl>
##  1 age_mean              40.9
##  2 age_median            39
##  3 age_sd                10.6
##  4 age_var              113.
##  5 age_IQR               15
##  6 age_range             77
##  7 balance_mean        1362.
##  8 balance_median       448
##  9 balance_sd          3045.
## 10 balance_var      9270599.
## # i 32 more rows
```

```r
cat("bank_df outlier count:\n")
```

```
## bank_df outlier count:
```

```r
bank_df %>%
  summarise(across(where(is.numeric), list(
    mean = ~mean(., na.rm = TRUE),
```

```
    median = ~median(., na.rm = TRUE),
    sd = ~sd(., na.rm = TRUE),
    var = ~var(., na.rm = TRUE),
    IQR = ~IQR(., na.rm = TRUE),
    range = ~max(., na.rm = TRUE) - min(., na.rm = TRUE)
  ))) %>%
  pivot_longer(cols = everything(), names_to = "Metric", values_to = "Value")
```

```
## # A tibble: 42 x 2
##    Metric              Value
##    <chr>               <dbl>
##  1 age_mean             41.2
##  2 age_median           39
##  3 age_sd               10.6
##  4 age_var             112.
##  5 age_IQR              16
##  6 age_range            68
##  7 balance_mean       1423.
##  8 balance_median      444
##  9 balance_sd         3010.
## 10 balance_var     9057922.
## # i 32 more rows
```

Algorithm Selection

Logistic Regression: works well for classification problems and can handle categorical features when properly encoded/

Pros: * Interpretable and easy to implement * Works well with linearly separable data and provides probabilistic outputs

Cons: * Struggles with non-linear relationships unless feature engineering is applied * Requires categorical encoding.

K-Nearest Neighbors (KNN): A non-parametric model that can handle both numeric and categorical features.

Pros: * Non-parametric, meaning it can capture complex relationships without assuming a distribution. * Works well with mixed data types if properly preprocessed.

Cons: * Computationally expensive with large datasets due to distance calculations. * Performance depends heavily on the choice of k and feature scaling.

Overall, I would recommend using K-Nearest Neighbors (KNN). This is because the datasets have both both numeric and categorical features and needs a non-parametric model that can capture complex relationships. I just need to be mindful of feature scaling and ensure the appropriate k value is picked.

The data had labels and allowed me to choose a supervised learning model that could be trained towards known values. if there aren't labels, I would have leaned towards a clustering model, like K-means.

For fewer than 1,000 data points, I would pick logistic regression and focus on binary outputs from the sparse availability. Otherwise, overfitting becomes a risk.

Pre-processing

- The metrics need to be normalized, with different types of metrics getting different normalization
  - Robust scaling for metrics like Balance, Campaign, and Duration where there are heavy skews

  - Standard z-score normalization for metrics like Age where you see something resembling a normal curve

- Missing values would be imputed where possible
- For age, a combination of values for Job, Housing, and Marital can be used to create. reasonable assumption of someone's age. The value imputed would be the median age based on these three points
- I would drop unknowns for Housing and Marital since those look relatively small
- I would not need to change the size of this dataset because it's small to begin with. if it was 100X larger, I would randomly sample 10% to start and see if that was enough.