The correlationfunnel package is a tool that enables ***efficient exploration of data, understanding relationships, and get to business insights as fast as possible.*** The results have been great so far. Students are using it as the ***EDA step prior to Machine Learning*** to ensure that features have relationship before they spend significant time tuning ML models on bad data. It’s helped me ***get to business insights 100X faster***, which has been a massive productivity boost. And, it’s a ***great communication tool*** that has helped explain business insights to executives and project stakeholders! Win-win-win.

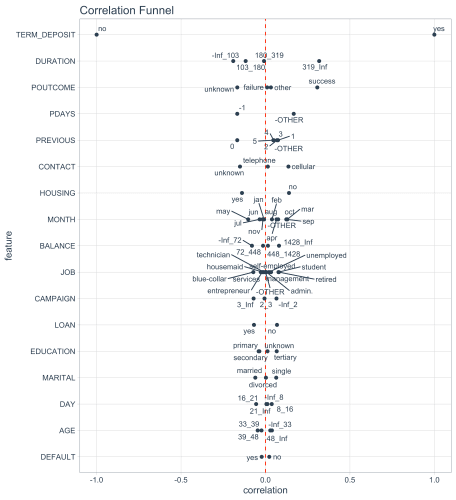
**A Critical Tool in My Machine Learning Arsenal**

[[](https://business-science.github.io/correlationfunnel/)](https://business-science.github.io/correlationfunnel/)

Why do I teach the correlationfunnel? Why is it an 80/20 tool that I use in every analysis? Why is it so great for communication?

It’s because of the ***Correlation Funnel Visualization***. The correlation funnel succinctly explains relationships to the target feature. Plus, every executive and business leader understands correlation. So it makes it super easy to explain results.

Here’s the correlation funnel that we’ll make in the **2-Minute Case Study – Bank Marketing Campaign**. It’s examining a ***Marketing Campaign for a Bank***. Without knowing much about the data, we can see that the most important features (those with highest absolute correlation) go to the top. This is the crux of business insights – We know know which features are important, without spending significant time on machine learning. Further, we can make educated guesses about the story within the data (great for communicating with stakeholders).



Correlation Funnel Visualization – Important Features go to the Top

**1. Bank Marketing Campaign Case Study – 2 Minute Example**

The following case study showcases the power of **fast exploratory correlation analysis with correlationfunnel**. The goal of the analysis is to determine which features relate to the bank’s marketing campaign goal of having customers opt into a TERM DEPOSIT (financial product).

We will see that using **3 functions**, we can quickly:

1. Transform the data into a binary format with binarize()
2. Perform correlation analysis using correlate()
3. Visualize the highest correlation features using plot\_correlation\_funnel()

**Result**:

Rather than spend hours looking at individual plots of campaign features and comparing them to which customers opted in to the TERM DEPOSIT product, in seconds we can discover which groups of customers have enrolled, drastically speeding up EDA.

**Getting Started**

First, install correlationfunnel with:

install.packages("correlationfunnel")

Next, load the following libraries.

library(correlationfunnel)

library(dplyr)

Next, collect data to analyze. We’ll use Marketing Campaign Data for a Bank that was popularized by the [UCI Machine Learning Repository](https://archive.ics.uci.edu/ml/datasets/Bank+Marketing). We can load the data with data("marketing\_campaign\_tbl").

# Use ?marketing\_campagin\_tbl to get a description of the marketing campaign features

data("marketing\_campaign\_tbl")

marketing\_campaign\_tbl %>% glimpse()

## Observations: 45,211

## Variables: 18

## $ ID "2836", "2837", "2838", "2839", "2840", "2841"…

## $ AGE 58, 44, 33, 47, 33, 35, 28, 42, 58, 43, 41, 29…

## $ JOB "management", "technician", "entrepreneur", "b…

## $ MARITAL "married", "single", "married", "married", "si…

## $ EDUCATION "tertiary", "secondary", "secondary", "unknown…

## $ DEFAULT "no", "no", "no", "no", "no", "no", "no", "yes…

## $ BALANCE 2143, 29, 2, 1506, 1, 231, 447, 2, 121, 593, 2…

## $ HOUSING "yes", "yes", "yes", "yes", "no", "yes", "yes"…

## $ LOAN "no", "no", "yes", "no", "no", "no", "yes", "n…

## $ CONTACT "unknown", "unknown", "unknown", "unknown", "u…

## $ DAY 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5…

## $ MONTH "may", "may", "may", "may", "may", "may", "may…

## $ DURATION 261, 151, 76, 92, 198, 139, 217, 380, 50, 55, …

## $ CAMPAIGN 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…

## $ PDAYS -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1…

## $ PREVIOUS 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…

## $ POUTCOME "unknown", "unknown", "unknown", "unknown", "u…

## $ TERM\_DEPOSIT "no", "no", "no", "no", "no", "no", "no", "no"…

**Response & Predictor Relationships**

Modeling and Machine Learning problems often involve a response (Enrolled in TERM\_DEPOSIT, yes/no) and many predictors (AGE, JOB, MARITAL, etc). Our job is to determine which predictors are related to the response. We can do this through **Binary Correlation Analysis**.

**Binary Correlation Analysis**

***Binary Correlation Analysis*** is the process of converting continuous (numeric) and categorical (character/factor) data to binary features. We can then perform a correlation analysis to see if there is predictive value between the features and the response (target).

**Step 1: Convert to Binary Format**

The first step is converting the continuous and categorical data into binary (0/1) format. We de-select any non-predictive features. The binarize() function then converts the features into binary features.

* **Numeric Features:** Are binned into ranges or if few unique levels are binned by their value, and then converted to binary features via one-hot encoding
* **Categorical Features**: Are binned by one-hot encoding

The result is a data frame that has only binary data with columns representing the bins that the observations fall into. There are now 74 columns that are binary (0/1). Note that the output was trimmed to the first 10 rows using slice().

marketing\_campaign\_binarized\_tbl <- marketing\_campaign\_tbl %>%

select(-ID) %>%

binarize(n\_bins = 4, thresh\_infreq = 0.01)

marketing\_campaign\_binarized\_tbl %>%

slice(1:10) %>%

knitr::kable()

| **AGE\_\_-Inf\_33** | **AGE\_\_33\_39** | **AGE\_\_39\_48** | **AGE\_\_48\_Inf** | **JOB\_\_admin.** | **JOB\_\_blue-collar** | **JOB\_\_entrepreneur** | **JOB\_\_housemaid** | **JOB\_\_management** | **JOB\_\_retired** | **JOB\_\_self-employed** | **JOB\_\_services** | **JOB\_\_student** | **JOB\_\_technician** | **JOB\_\_unemployed** | **JOB\_\_-OTHER** | **MARITAL\_\_divorced** | **MARITAL\_\_married** | **MARITAL\_\_single** | **EDUCATION\_\_primary** | **EDUCATION\_\_secondary** | **EDUCATION\_\_tertiary** | **EDUCATION\_\_unknown** | **DEFAULT\_\_no** | **DEFAULT\_\_yes** | **BALANCE\_\_-Inf\_72** | **BALANCE\_\_72\_448** | **BALANCE\_\_448\_1428** | **BALANCE\_\_1428\_Inf** | **HOUSING\_\_no** | **HOUSING\_\_yes** | **LOAN\_\_no** | **LOAN\_\_yes** | **CONTACT\_\_cellular** | **CONTACT\_\_telephone** | **CONTACT\_\_unknown** | **DAY\_\_-Inf\_8** | **DAY\_\_8\_16** | **DAY\_\_16\_21** | **DAY\_\_21\_Inf** | **MONTH\_\_apr** | **MONTH\_\_aug** | **MONTH\_\_feb** | **MONTH\_\_jan** | **MONTH\_\_jul** | **MONTH\_\_jun** | **MONTH\_\_mar** | **MONTH\_\_may** | **MONTH\_\_nov** | **MONTH\_\_oct** | **MONTH\_\_sep** | **MONTH\_\_-OTHER** | **DURATION\_\_-Inf\_103** | **DURATION\_\_103\_180** | **DURATION\_\_180\_319** | **DURATION\_\_319\_Inf** | **CAMPAIGN\_\_-Inf\_2** | **CAMPAIGN\_\_2\_3** | **CAMPAIGN\_\_3\_Inf** | **PDAYS\_\_-1** | **PDAYS\_\_-OTHER** | **PREVIOUS\_\_0** | **PREVIOUS\_\_1**PREVIOUS\_\_2 PREVIOUS\_\_3 PREVIOUS\_\_4 PREVIOUS\_\_5 PREVIOUS\_\_-OTHER POUTCOME\_\_failure POUTCOME\_\_other POUTCOME\_\_success POUTCOME\_\_unknown TERM\_DEPOSIT\_\_no TERM\_DEPOSIT\_\_yes |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 00 0 0 0 0 0 0 0 1 1 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 00 0 0 0 0 0 0 0 1 1 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 00 0 0 0 0 0 0 0 1 1 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 00 0 0 0 0 0 0 0 1 1 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 00 0 0 0 0 0 0 0 1 1 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 00 0 0 0 0 0 0 0 1 1 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 00 0 0 0 0 0 0 0 1 1 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 00 0 0 0 0 0 0 0 1 1 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 00 0 0 0 0 0 0 0 1 1 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 00 0 0 0 0 0 0 0 1 1 0 |

**Step 2: Perform Correlation Analysis**

The second step is to perform a correlation analysis between the response (target = TERM\_DEPOSIT\_yes) and the rest of the features. This returns a specially formatted tibble with the feature, the bin, and the bin’s correlation to the target. The format is exactly what we need for the next step – Producing the **Correlation Funnel**

marketing\_campaign\_correlated\_tbl <- marketing\_campaign\_binarized\_tbl %>%

correlate(target = TERM\_DEPOSIT\_\_yes)

marketing\_campaign\_correlated\_tbl

## # A tibble: 74 x 3

## feature bin correlation

##

## 1 TERM\_DEPOSIT no -1.000

## 2 TERM\_DEPOSIT yes 1.000

## 3 DURATION 319\_Inf 0.318

## 4 POUTCOME success 0.307

## 5 DURATION -Inf\_103 -0.191

## 6 PDAYS -OTHER 0.167

## 7 PDAYS -1 -0.167

## 8 PREVIOUS 0 -0.167

## 9 POUTCOME unknown -0.167

## 10 CONTACT unknown -0.151

## # … with 64 more rows

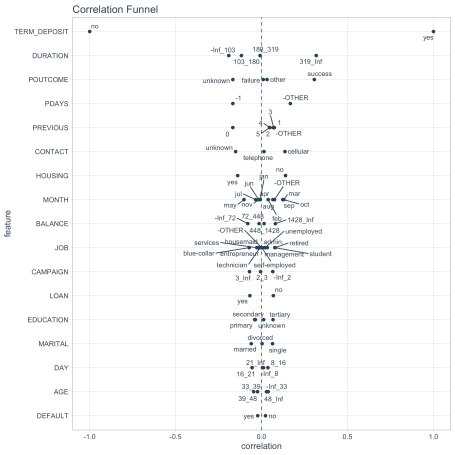
**Step 3: Visualize the Correlation Funnel**

A **Correlation Funnel** is an tornado plot that lists the highest correlation features (based on absolute magnitude) at the top of the and the lowest correlation features at the bottom. The resulting visualization looks like a Funnel.

To produce the **Correlation Funnel**, use plot\_correlation\_funnel(). Try setting interactive = TRUE to get an interactive plot that can be zoomed in on.

marketing\_campaign\_correlated\_tbl %>%

plot\_correlation\_funnel(interactive = FALSE)



**Examining the Results**

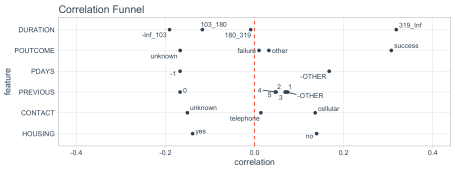
The most important features are towards the top. We can investigate these.

marketing\_campaign\_correlated\_tbl %>%

filter(feature %in% c("DURATION", "POUTCOME", "PDAYS",

"PREVIOUS", "CONTACT", "HOUSING")) %>%

plot\_correlation\_funnel(interactive = FALSE, limits = c(-0.4, 0.4))



We can see that the following prospect groups have a much greater correlation with enrollment in the TERM DEPOSIT product:

* When the “DURATION”, the amount of time a prospect is engaged in marketing campaign material, is 319 seconds or longer.
* When “POUTCOME”, whether or not a prospect has previously enrolled in a product, is “success”.
* When “CONTACT”, the medium used to contact the person, is “cellular”
* When “HOUSING”, whether or not the contact has a HOME LOAN is “no”

**2. My EDA Workflow R**

The main addition of correlationfunnel to the EDA ecosystem in R is to quickly expose feature relationships.

There are other EDA packages, but none implement the **binarization process** as a preprocessing step before correlation. This is critical to extracting insights especially in nonlinear continuous data.

**Correlation funnel requires semi-processed data meaning**

* Missing (NA) values have been treated
* Date or date-time features have been feature engineered into categorical variables
* Data is in a “clean” format (numeric data and categorical data are ready to be correlated to a Yes/No response)

Because of this, I use other packages to understand data (e.g. missing values) and correct data (preprocessing).

**EDA Packages I Use Before I Use Correlation Funnel to identify processing steps**

* GGally – The ggpairs() function is one of my all-time favorites for visualizing many features quickly. I teach GGally in **DS4B 201-R, Advanced Machine Learning and Business Consulting**, part of the 3-Course R-Track for Business.
* skimr – The skim() function is my go-to for console statistics on each of my variables. It’s like str() combined with summary() on steroids. I teach skimr in **DS4B 201-R, Advanced Machine Learning and Business Consulting**.
* Data Explorer – Automates Exploration and Data Treatment. Amazing for investigating features quickly and efficiently including by data type, missing data, feature engineering, and identifying relationships.

**Packages I use to implement preprocessing**

* recipes – This is one of the most important R packages in my exploratory and machine learning workflow. It’s use to create ***preprocessing pipelines*** – A series of steps to correct data and prepare for machine learning, correlation analysis, etc. I teach recipes in **DS4B 201-R, Advanced Machine Learning and Business Consulting**.

**Other Great EDA Packages (Notable Mentions)**

* VisDat – Similar in concept to Data Explorer. Utilities for visualizing missing data
* naniar – For understanding missing data.
* UpSetR – For generating upset plots

**3. What do you think of Correlation Funnel?**

Let me know what to do you think of correlationfunnel in the ***comments***. I’m interested in getting ***your feedback*** on how it works on different data sets. I’ve tried it on a ton, but that doesn’t mean that it’s perfect. Let me know your results – good or bad.