Exploratory Data Analysis

Ivor Walker

Data Preprocessing Read in the data and check the structure of the data - the separator is a semicolon.

```
bank_data <- read.csv("src/data/bank-additional-full.csv", sep = ";")
str(bank_data)
```

```
## 'data.frame':    41188 obs. of  21 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" "unknown" "no" "no" ...
##  $ housing       : chr  "no" "no" "yes" "no" ...
##  $ loan          : chr  "no" "no" "no" "no" ...
##  $ contact       : chr  "telephone" "telephone" "telephone" "telephone" ...
##  $ month         : chr  "may" "may" "may" "may" ...
##  $ day_of_week   : chr  "mon" "mon" "mon" "mon" ...
##  $ duration      : int  261 149 226 151 307 198 139 217 380 50 ...
##  $ campaign      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays         : int  999 999 999 999 999 999 999 999 999 999 ...
##  $ 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 ...
##  $ nr.employed   : num  5191 5191 5191 5191 5191 ...
##  $ y             : chr  "no" "no" "no" "no" ...
```

R reads strings as character arrays, but should be treated as factors as they represent categorical data.

```
bank_data <- bank_data %>%
  mutate(across(where(is.character), as.factor))

str(bank_data)
```

```
## 'data.frame':    41188 obs. of  21 variables:
##  $ age           : int  56 57 37 40 56 45 59 41 24 25 ...
##  $ job           : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1 8 8 1 2 10 8 ...
##  $ marital       : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ...
##  $ education     : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4 3 6 8 6 4 ...
##  $ default       : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1 1 ...
##  $ housing       : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...
##  $ loan          : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1 1 ...
```

```
##  $ contact       : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...
##  $ month         : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 7 7 7 ...
##  $ day_of_week   : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ duration      : int  261 149 226 151 307 198 139 217 380 50 ...
##  $ campaign      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays         : int  999 999 999 999 999 999 999 999 999 999 ...
##  $ previous      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome      : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ 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 ...
##  $ nr.employed   : num  5191 5191 5191 5191 5191 ...
##  $ y             : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
```

```r
# Remove duplicate columns
clean_merged_columns <- function(bank_data) {
  bank_data <- bank_data %>%
    # Remove columns ending in .x
    select(-ends_with(".x")) %>%

    # Rename .y columns back to original
    rename_with(~ gsub("\\.y$", "", .x), ends_with(".y"))

  return(bank_data)
}
```

Given the data is ordered by date from May 2008 to November 2010, we can use the day (of week) to recreate a partial unique date, then aggregate by it and see if there are any temporal patterns in the data.

```r
bank_data <- bank_data %>%
  # Mark the first row of each new day
  mutate(new_day = if_else(row_number() == 1, TRUE, day_of_week != lag(day_of_week))) %>%

  # Create a unique day ID
  mutate(day_id = cumsum(new_day)) %>%

  # Only increment the week when a new Monday starts
  mutate(week_id = cumsum(new_day & day_of_week == "mon")) %>%

  # Within each week, assign a sequential day label (e.g mon: 1)
  group_by(week_id) %>%
  ungroup() %>%

  # Now mark the first row of a new month and create month labels
  mutate(new_month = if_else(row_number() == 1, TRUE, month != lag(month, default = first(month)))) %>%

  mutate(month_id = cumsum(new_month))

# Remove helper columns
bank_data <- bank_data %>%
  select(-new_day, -new_month)

# Group day, week and month labels together
```

```r
bank_data <- bank_data %>%
  mutate(date_label = paste(day_id, week_id, month_id, sep = "-"))

excluded_features <- c("day_id", "week_id", "month_id", "date_label")

print(summary(bank_data$day_id))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0    28.0    68.0   100.1   160.0   486.0
```

Finally, I will split into training and testing data and explore the training data only. Because my data has a time-series component, I choose to split the data by day ID - train on the first n days, then test on the remaining

```r
# Define split size
train_size <- 0.8

# Split data
split_day_id <- round(train_size * max(bank_data$day_id))
training_data <- bank_data %>%
  filter(day_id <= split_day_id)
test_data <- bank_data %>%
  filter(day_id > split_day_id)
```

Exploratory Data Analysis First, we will look at the distribution of the target variable.

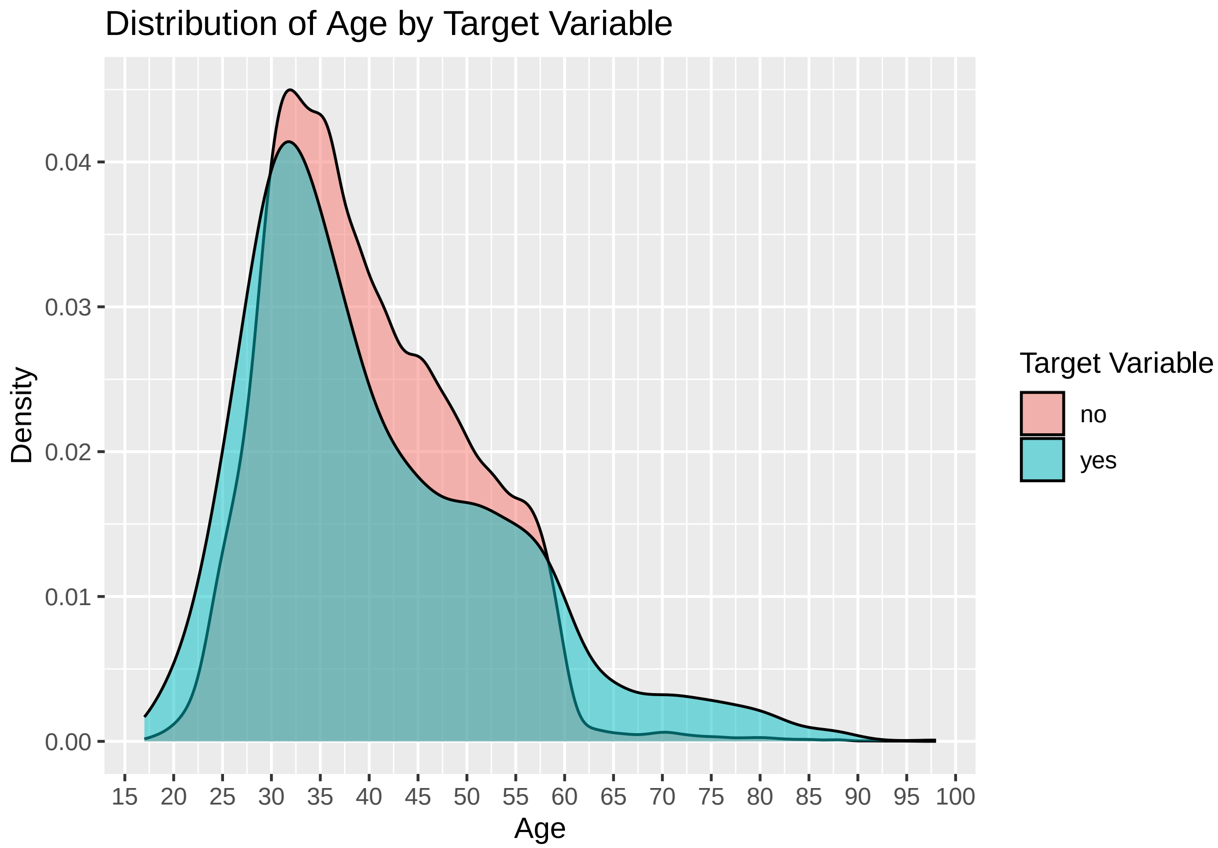
```r
summary(training_data$y)
```

```
##    no   yes
## 35984  4056
```

Our target variable has a class imbalance in favour of the "no" class.

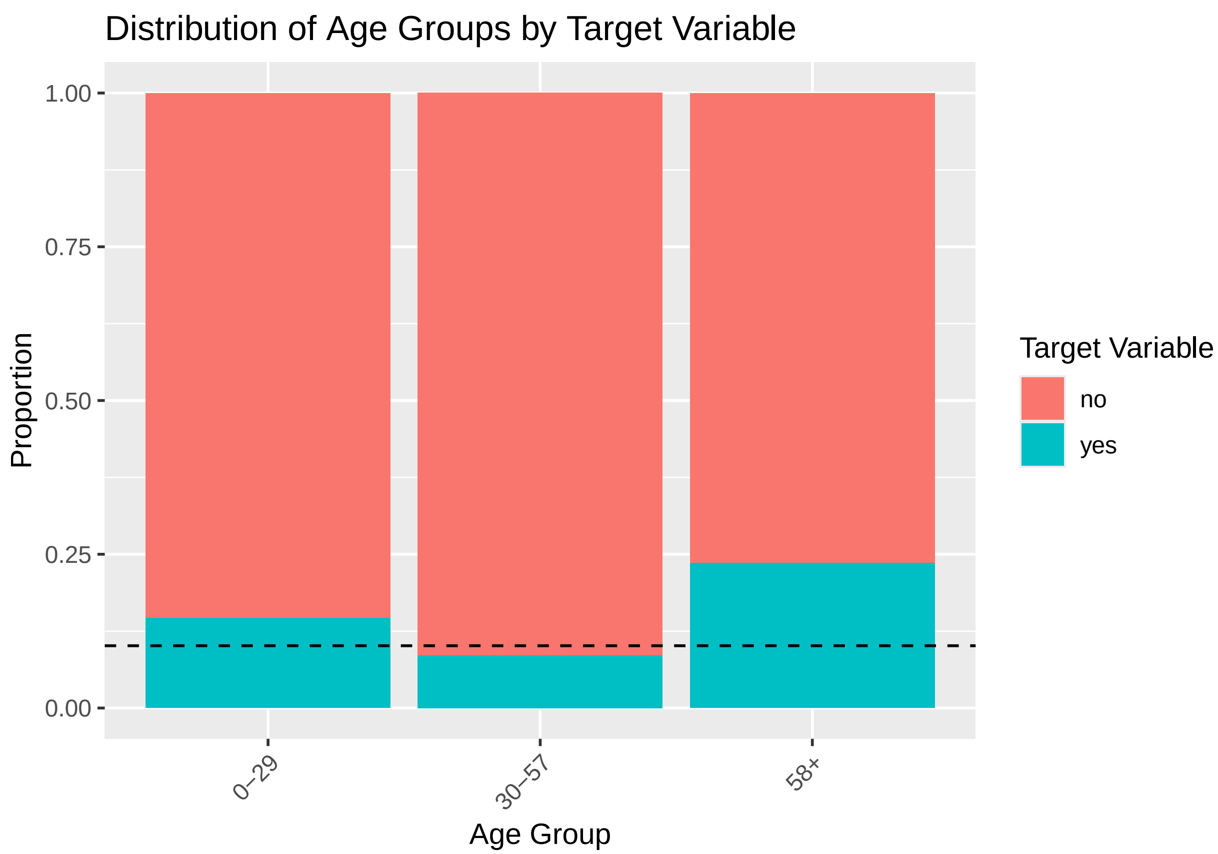
Next, we look at how all our features vary with the target variable.

```r
ggplot(training_data, aes(x = age, fill = y)) +
  geom_density(alpha = 0.5) +
  scale_x_continuous(breaks = seq(0, 100, by = 5)) +
  labs(title = "Distribution of Age by Target Variable",
       x = "Age",
       y = "Density",
       fill = "Target Variable")
```



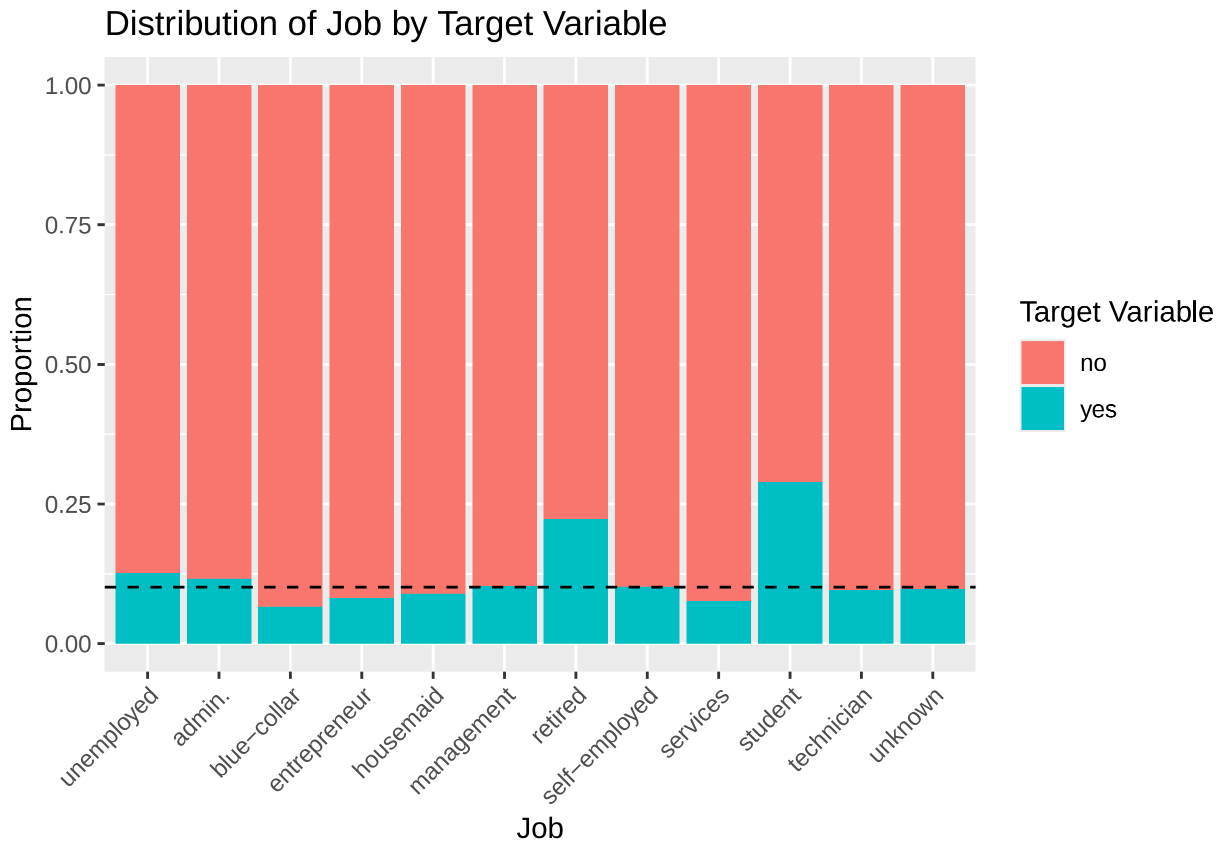
The distribution of age is different between classes. The "yes" class has a higher density between 0-29 and 58+, and the "no" class has a higher density between 30-57. I create a new factor variable by binning ages into these three categories. I use 0-29 as the reference category as it is closest to the mean density of "yes".

```r
# Create age groups
training_data <- training_data %>%
  mutate(age_group = case_when(
    age <= 29 ~ "0-29",
    age >= 30 & age <= 57 ~ "30-57",
    age >= 58 ~ "58+"
  ))

# Turn age groups into factors
training_data$age_group <- factor(training_data$age_group, levels = c("0-29", "30-57", "58+"))

# Add age to excluded features
excluded_features <- c(excluded_features, "age")

# Show distribution of age groups with a line of mean proportion and rotated x-axis labels
ggplot(training_data, aes(x = age_group, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(training_data$y == "yes"), linetype = "dashed") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of Age Groups by Target Variable",
       x = "Age Group",
       y = "Proportion",
       fill = "Target Variable")
```



Setting unemployed as the reference category allows comparison of the other job categories to a baseline.

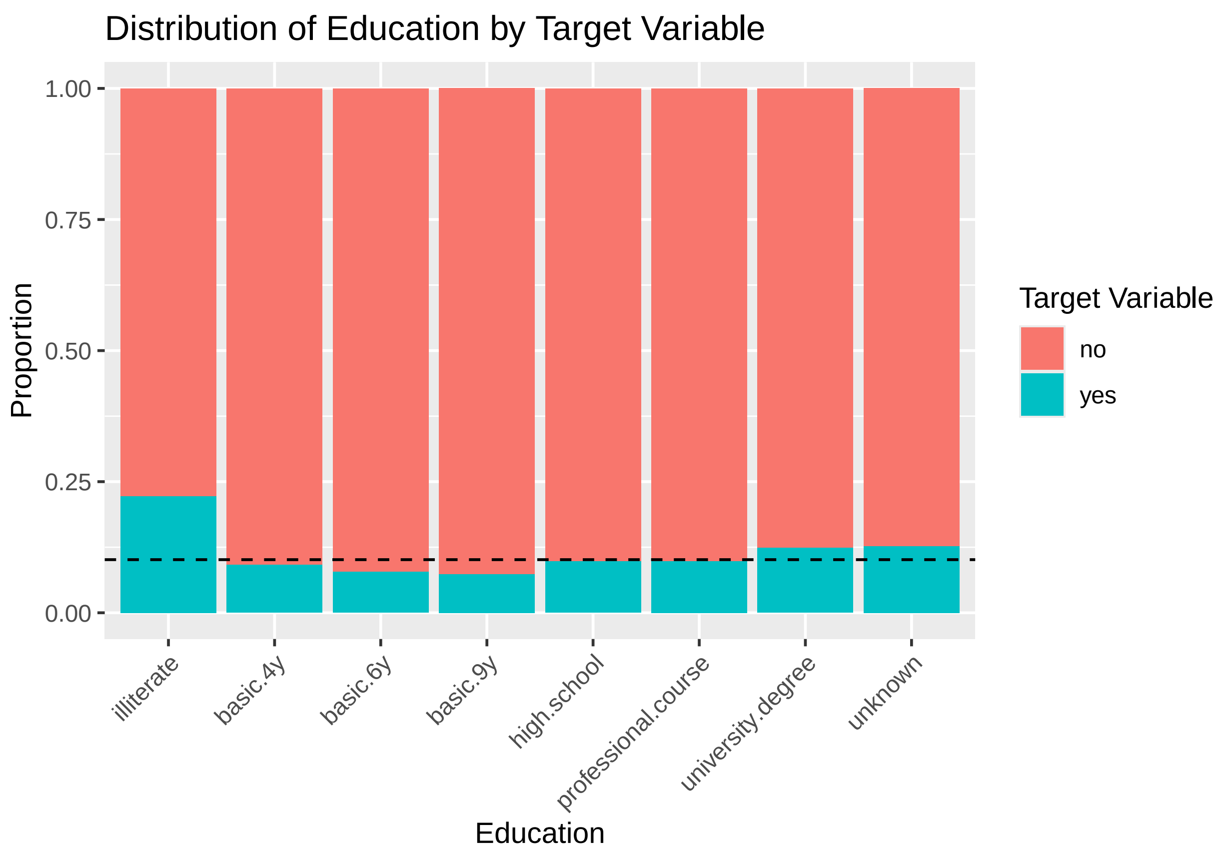
```
# Set "unemployed" as the reference category
training_data$job <- relevel(training_data$job, ref = "unemployed")

# Show distribution of job
ggplot(training_data, aes(x = job, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(training_data$y == "yes"), linetype = "dashed") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of Job by Target Variable",
       x = "Job",
       y = "Proportion",
       fill = "Target Variable")
```



Retired and students have a much higher proportion of "yes" and blue-collar has a slightly higher proportion of "no".

Education can be seen as an ordinal factor variable, so we can order the levels from lowest (illiterate) to highest (university.degree).

```
# Order education levels
training_data$education <- factor(training_data$education, levels = c("illiterate", "basic.4y", "basic.(

# Show distribution of education with a line of mean proportion and rotated x-axis labels
ggplot(training_data, aes(x = education, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(training_data$y == "yes"), linetype = "dashed") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of Education by Target Variable",
       x = "Education",
       y = "Proportion",
       fill = "Target Variable")
```

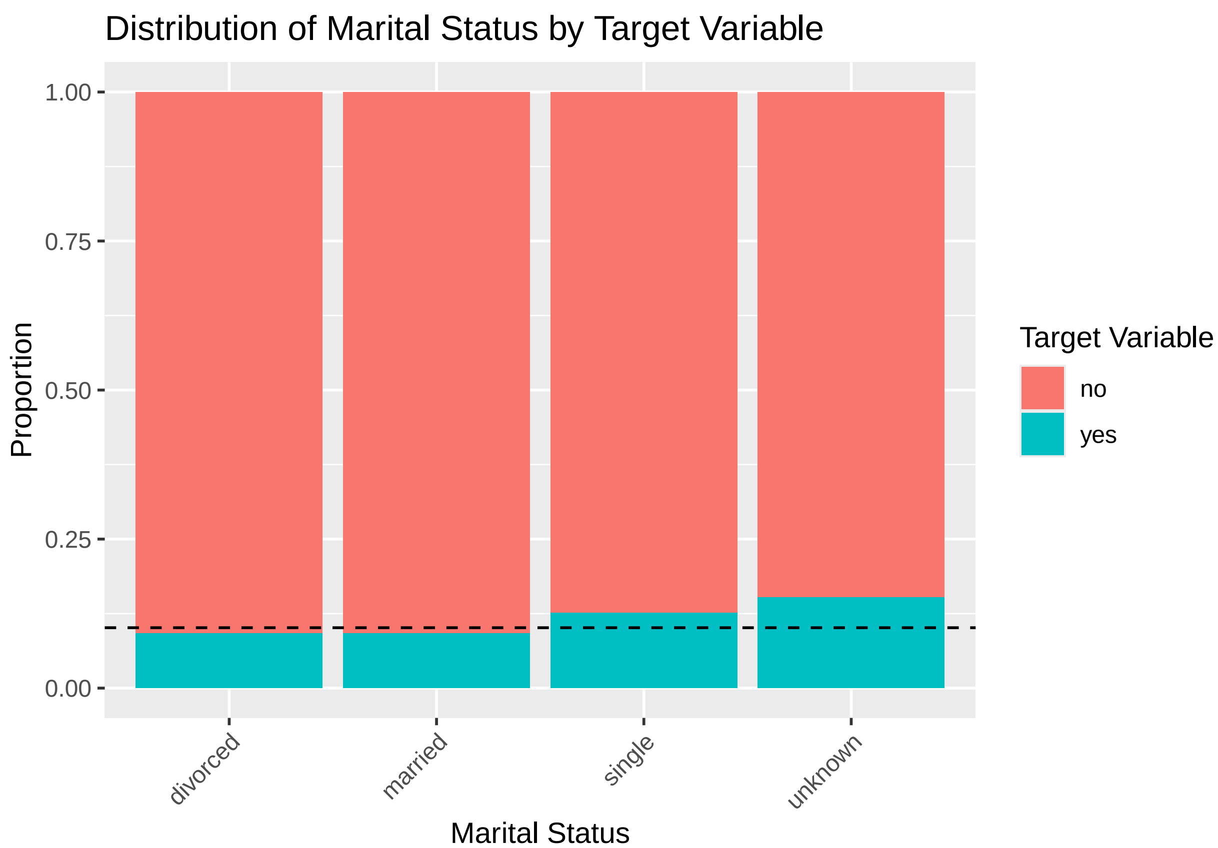



```
# Set "illiterate" as the reference category
training_data$education <- relevel(training_data$education, ref = "illiterate")

summary(training_data$education)
```

```
##          illiterate            basic.4y            basic.6y            basic.9y
##                  18                4059                2274                5976
##         high.school professional.course   university.degree             unknown
##                9265                5067               11735                1646
```

Illiterate has a much higher proportion of "yes" but is an extremely small category. "yes" falls as education level increases until high school, where it increases again.

```
# Show distribution of marital status
ggplot(training_data, aes(x = marital, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(training_data$y == "yes"), linetype = "dashed") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of Marital Status by Target Variable",
       x = "Marital Status",
       y = "Proportion",
       fill = "Target Variable")
```



The distribution of marital status is similar between classes.

```
summary(training_data$default)
```

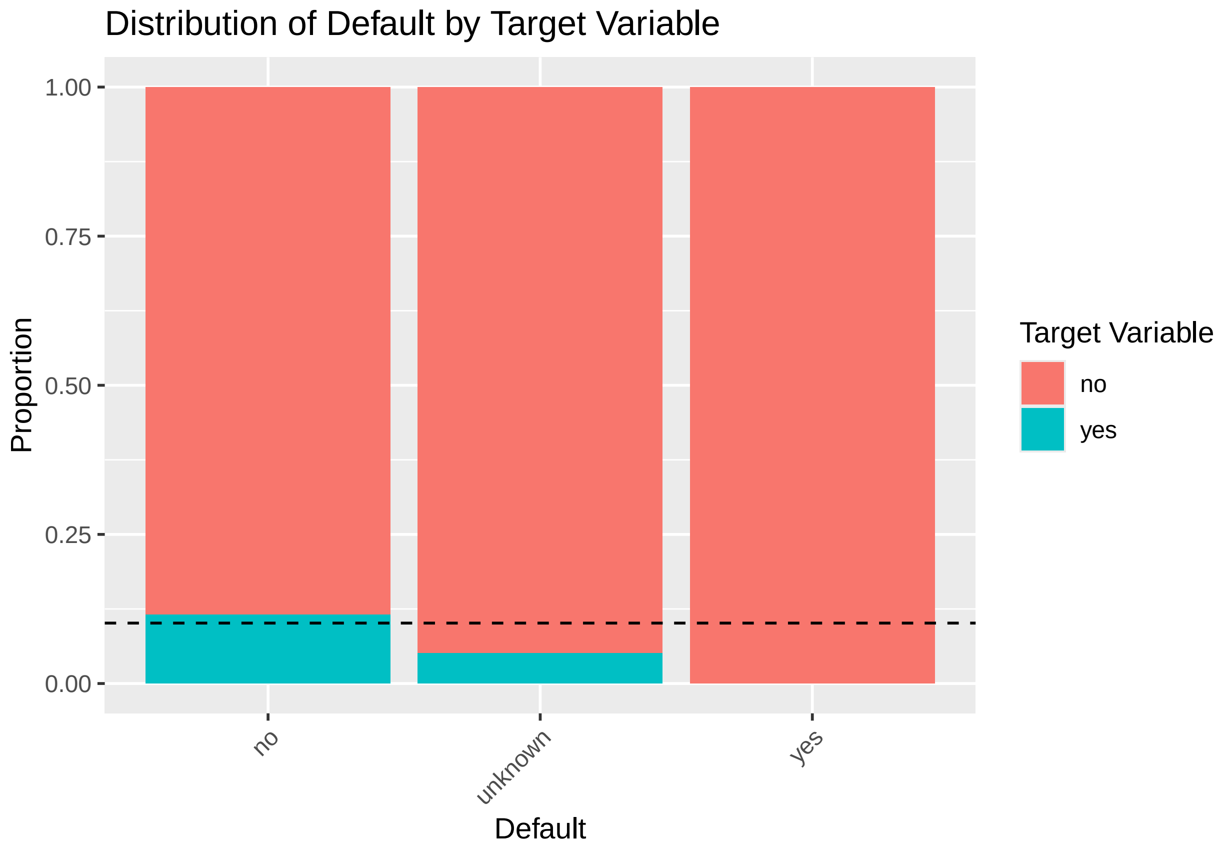
```
##      no unknown     yes
##   31460    8577       3
```

```
missing_default <- training_data %>%
  filter(default == "unknown")

nrow(missing_default) / nrow(training_data)
```

```
## [1] 0.2142108
```

This feature appears useless - there are extremely few "yes" and 21% of data on this feature is missing.

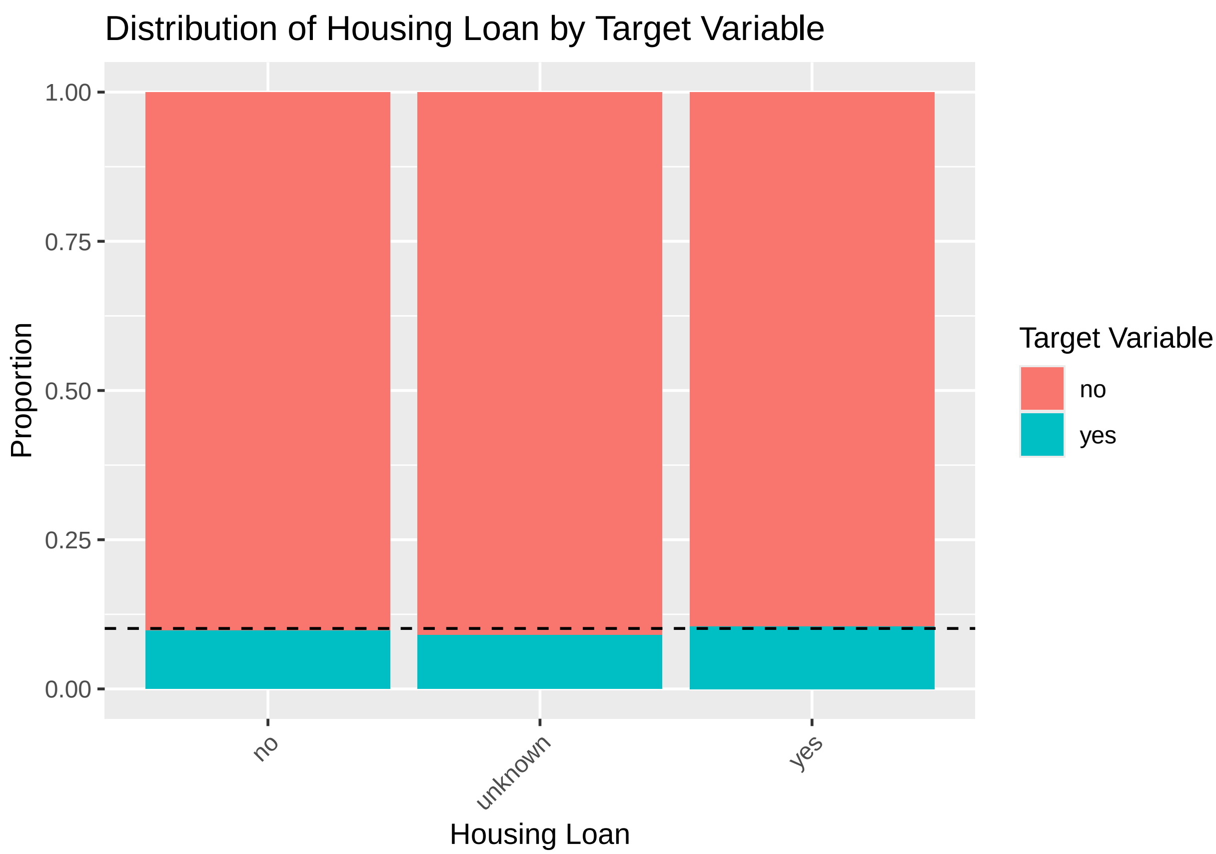
```
# Show distribution of default
ggplot(training_data, aes(x = default, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(training_data$y == "yes"), linetype = "dashed") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of Default by Target Variable",
       x = "Default",
       y = "Proportion",
       fill = "Target Variable")
```



However, “unknown” has a much lower proportion of “Yes” than “no”. People who are embarrassed to admit they have a default may be more likely to say “unknown” than “yes”. I will combine unknown and “yes”.

```
# Combine unknown and yes
training_data <- training_data %>%
  mutate(default_group = if_else((default == "unknown" | default == "yes"), "unknown_or_yes", default))

# Turn into factor
training_data$default_group <- factor(training_data$default_group, levels = c("no", "unknown_or_yes"))

# Add default to excluded features
excluded_features <- c(excluded_features, "default")
```

```
# Show distribution of housing loan
ggplot(training_data, aes(x = housing, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(training_data$y == "yes"), linetype = "dashed") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of Housing Loan by Target Variable",
       x = "Housing Loan",
       y = "Proportion",
       fill = "Target Variable")
```



House loan distribution is similar between classes.

```
# Show distribution of personal loan
ggplot(training_data, aes(x = loan, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(training_data$y == "yes"), linetype = "dashed") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of Personal Loan by Target Variable",
       x = "Personal Loan",
       y = "Proportion",
       fill = "Target Variable")
```

Distribution of personal loan is similar between classes.

```
# Show distribution of contact method
ggplot(training_data, aes(x = contact, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(training_data$y == "yes"), linetype = "dashed") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of Contact Method by Target Variable",
       x = "Contact Method",
       y = "Proportion",
       fill = "Target Variable")
```

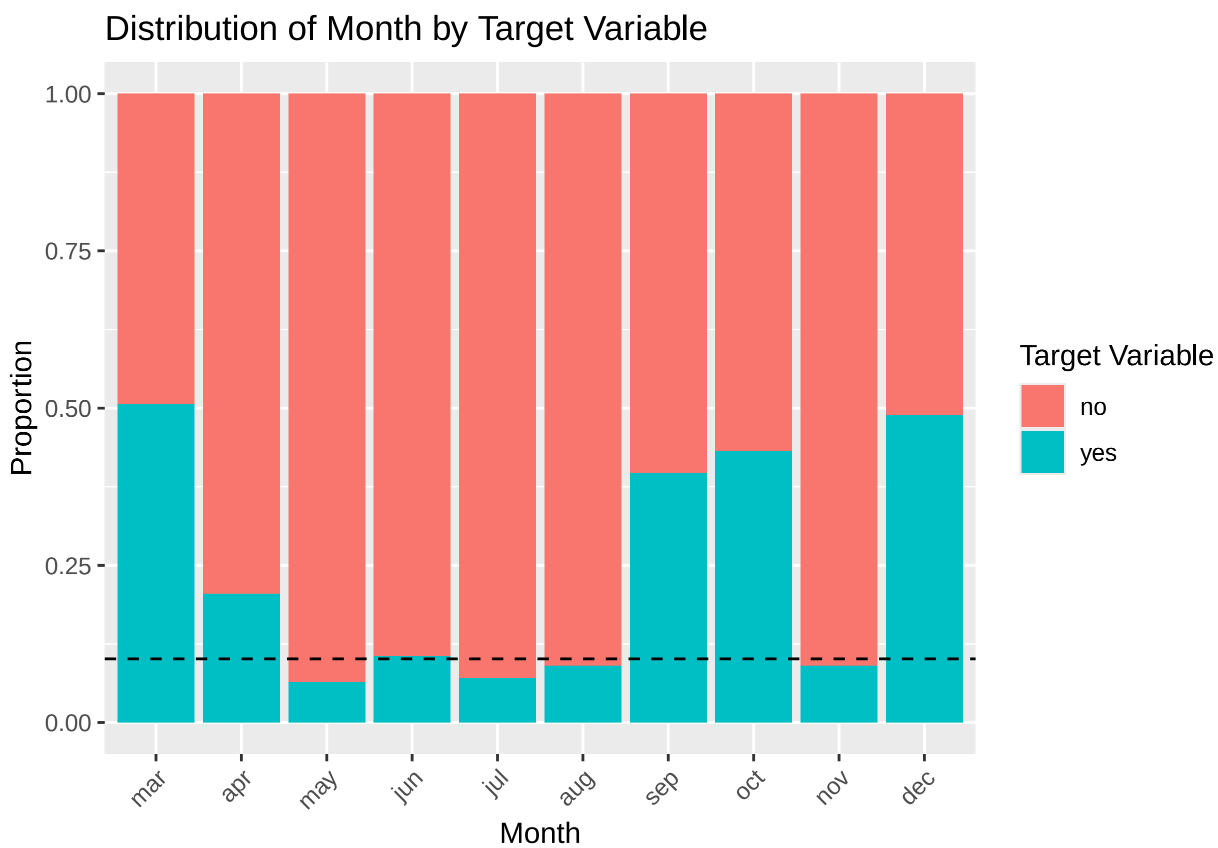
Cellular has a higher proportion of "yes" and telephone has a much higher proportion of "no".

Month is an ordinal factor variable, so we can order the levels from January to December.

```
# Order month levels
training_data$month <- factor(training_data$month, levels = c("jan", "feb", "mar", "apr", "may", "jun",

# Show frequency of each month
frequency_per_month <- table(training_data$month)
frequency_per_month
```

```
##
##   jan   feb   mar   apr   may   jun   jul   aug   sep   oct   nov   dec
##     0     0   546  2632 13769  5318  6894  5945   267   514  3973   182
```

```
# Show distribution of month and frequency of each month
ggplot(training_data, aes(x = month, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(training_data$y == "yes"), linetype = "dashed") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of Month by Target Variable",
       x = "Month",
       y = "Proportion",
       fill = "Target Variable")
```



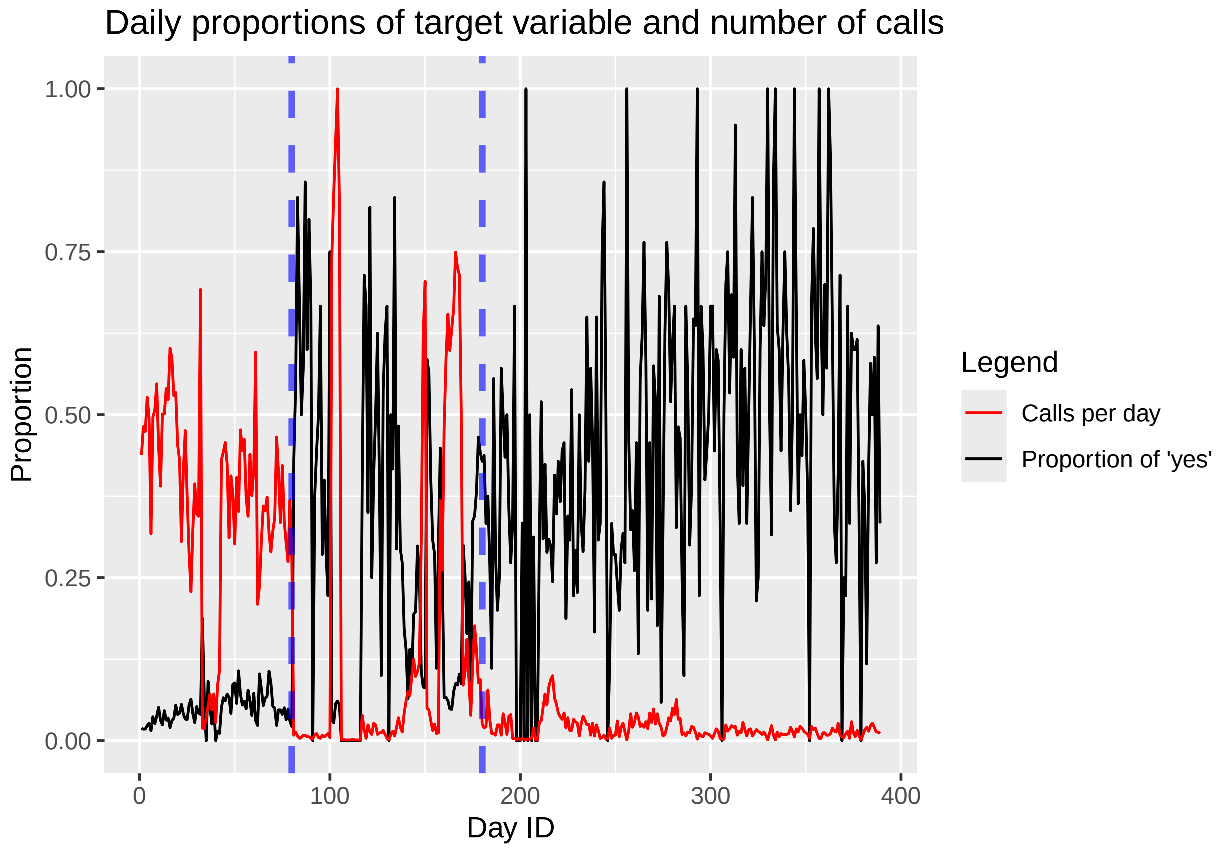
No records took place in January or February. Although there appears to be a monthly variation in target variable, the fewer records there are in a month, the higher the proportion of "yes" which suggests that some selection bias is occurring.

Now we can see the distribution of the target variable and frequency of call by day, week, and month.

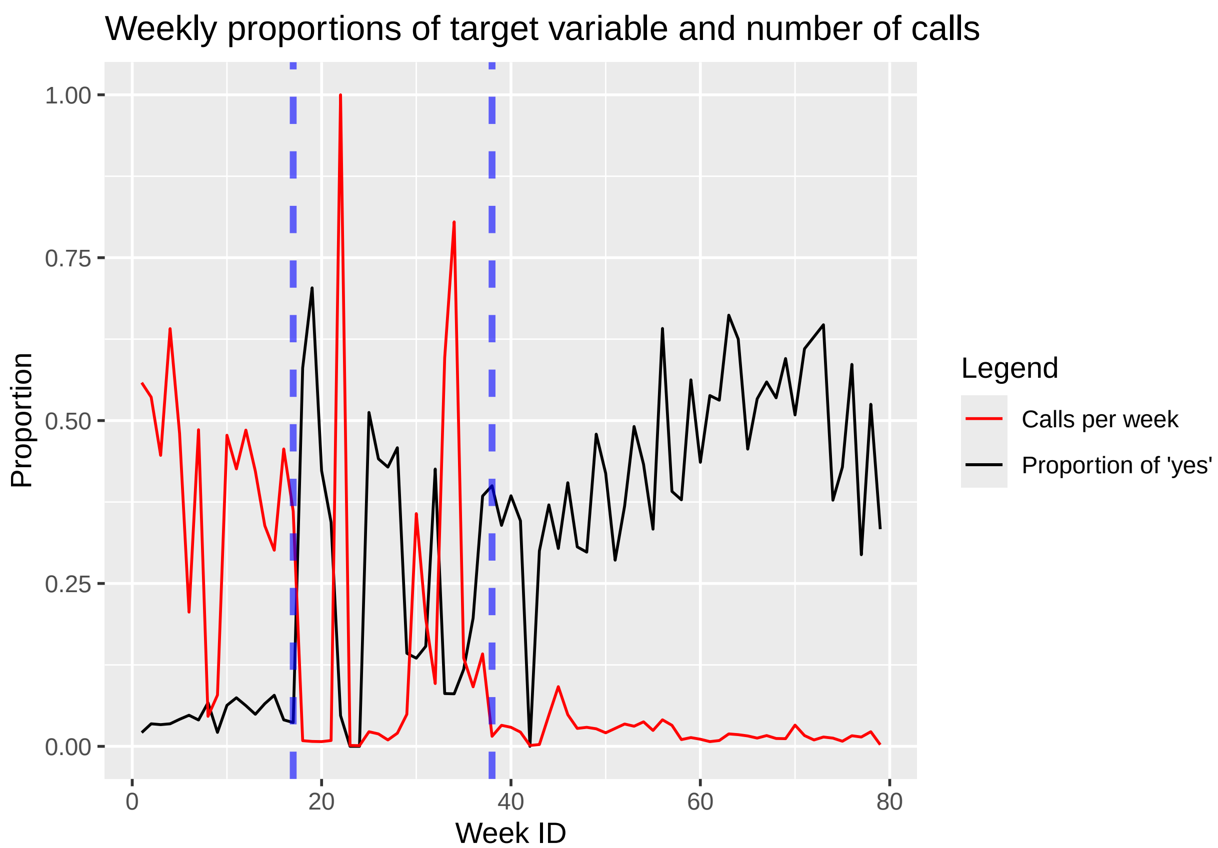
```
# Calculate number of calls per day
calls_per_day <- training_data %>% count(day_id, name = "n_calls")
benchmark_day_calls <- max(calls_per_day$n_calls)

# Add proportion of max calls per day
calls_per_day <- calls_per_day %>%
  mutate(proportion = n_calls / benchmark_day_calls)

day_seperators <- c(80, 180)

# Set seperator settings
size <- 1.2
color <- "blue"
linetype <- "dashed"
alpha <- 0.6

# Show line graph of target variable and number of calls per day with a legend
ggplot(training_data, aes(x = day_id)) +

  # Show proportion of "yes" as a line
  stat_summary(aes(y = as.numeric(y == "yes"), color = "Proportion of 'yes'"),
```

```
             fun = mean, geom = "line") +

  # Show number of calls as a line
  geom_line(data = calls_per_day, aes(x = day_id, y = n_calls / benchmark_day_calls, color = "Calls per
            show.legend = TRUE) +

  # Show seperators
  geom_vline(xintercept = day_seperators, linetype = linetype, color = color, size = size, alpha = alph

  # Change legend labels
  scale_color_manual(values = c("Proportion of 'yes'" = "black", "Calls per day" = "red")) +

  labs(
    title = "Daily proportions of target variable and number of calls",
    x = "Day ID",
    y = "Proportion",
    color = "Legend"
  )
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

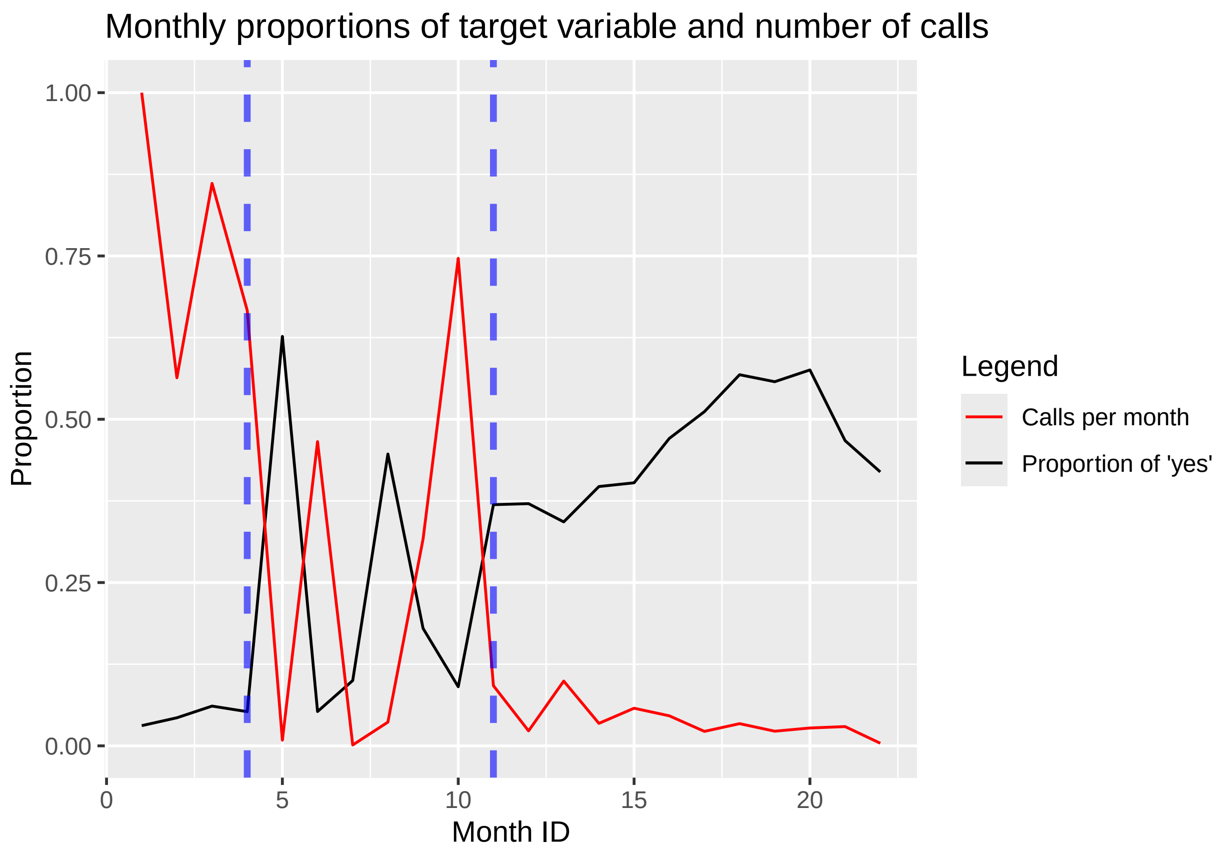


```
# Calculate number of calls per week
calls_per_week <- training_data %>% count(week_id, name = "n_calls")
benchmark_week_calls <- max(calls_per_week$n_calls)

# Add proportion of max calls per week
calls_per_week <- calls_per_week %>%
  mutate(proportion = n_calls / benchmark_week_calls)

week_seperators <- c(17, 38)

# Show line graph of target variable and number of calls per week with a legend
ggplot(training_data, aes(x = week_id)) +

  # Show proportion of "yes" as a line
  stat_summary(aes(y = as.numeric(y == "yes"), color = "Proportion of 'yes'"),
               fun = mean, geom = "line") +

  # Show number of calls as a line
  geom_line(data = calls_per_week, aes(x = week_id, y = n_calls / benchmark_week_calls, color = "Calls p
            show.legend = TRUE) +

  # Show seperators
  geom_vline(xintercept = week_seperators, linetype = linetype, color = color, size = size, alpha = alph

  # Change legend labels
  scale_color_manual(values = c("Proportion of 'yes'" = "black", "Calls per week" = "red")) +

  labs(
    title = "Weekly proportions of target variable and number of calls",
    x = "Week ID",
    y = "Proportion",
    color = "Legend"
  )
```




```
# Calculate number of calls per month
calls_per_month <- training_data %>% count(month_id, name = "n_calls")
benchmark_month_calls <- max(calls_per_month$n_calls)

# Add proportion of max calls per month
calls_per_month <- calls_per_month %>%
  mutate(proportion = n_calls / benchmark_month_calls)

month_seperators <- c(4, 11)

# Show line graph of target variable and number of calls per month with a legend
ggplot(training_data, aes(x = month_id)) +

  # Show proportion of "yes" as a line
  stat_summary(aes(y = as.numeric(y == "yes"), color = "Proportion of 'yes'"),
               fun = mean, geom = "line") +

  # Show number of calls as a line
  geom_line(data = calls_per_month, aes(x = month_id, y = n_calls / benchmark_month_calls, color = "Cal
            show.legend = TRUE) +

  # Show seperators
  geom_vline(xintercept = month_seperators, linetype = linetype, color = color, size = size, alpha = al

  # Change legend labels
  scale_color_manual(values = c("Proportion of 'yes'" = "black", "Calls per month" = "red")) +
```

```r
  labs(
    title = "Monthly proportions of target variable and number of calls",
    x = "Month ID",
    y = "Proportion",
    color = "Legend"
  )
```



There are three distinct time periods in the data. From day 1-80 (week 1-17, month 1-4) there are a large and somewhat stable level of calls but a stable near-zero proportion of "yes". From day 81-180 (week 18-38, month 5-11), the number of calls decreases and becomes more unstable but the proportion of "yes" increases and becomes unstable. After day 181, the number of calls approaches zero but the proportion of "yes" rises and becomes stable.

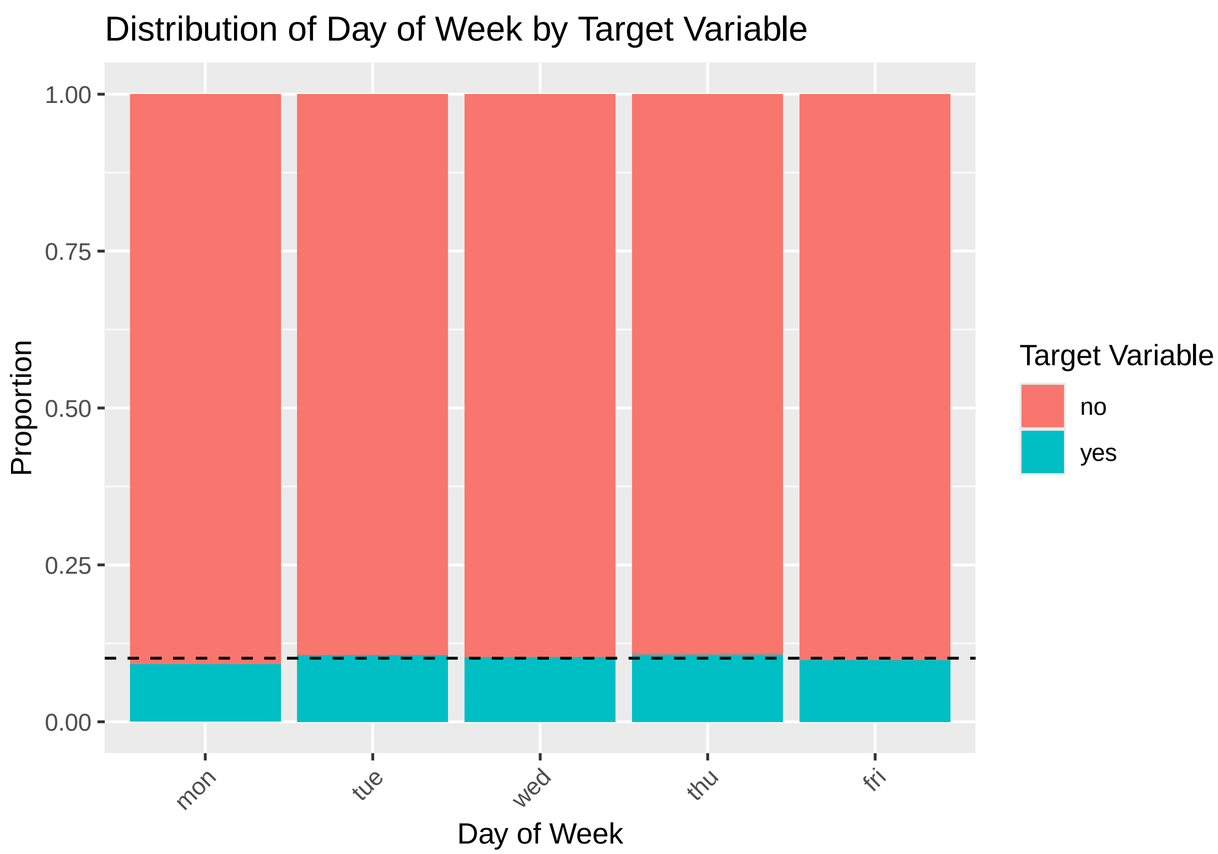
These temporal patterns suggest a change in strategy - the first period is likely a warm-up period, the second period is likely a test period, and the third period is likely a deployment period.

I will add these manually identified strategy periods, and if significant explore methods to automatically identify these periods.

```r
# Add strategy periods
training_data <- training_data %>%
  mutate(strategy_period = case_when(
    day_id <= 80 ~ "1",
    day_id > 80 & day_id <= 180 ~ "2",
    day_id > 180 ~ "3"
  ))
```

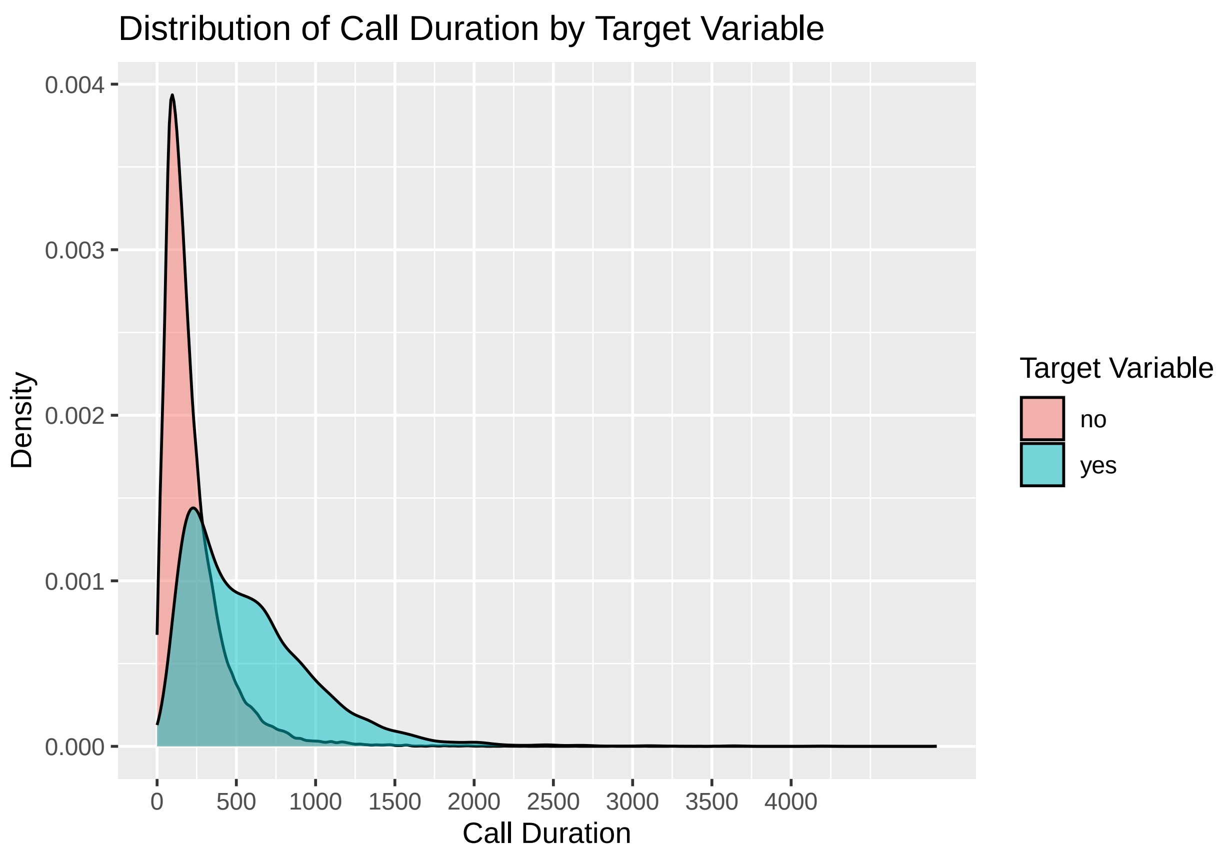
```r
# Convert to factor
training_data$strategy_period <- factor(training_data$strategy_period, levels = c("1", "2", "3"))
```

```r
# Sort day of week factor
training_data$day_of_week <- factor(training_data$day_of_week, levels = c("mon", "tue", "wed", "thu", "

# Show distribution of day of week
ggplot(training_data, aes(x = day_of_week, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(training_data$y == "yes"), linetype = "dashed") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of Day of Week by Target Variable",
       x = "Day of Week",
       y = "Proportion",
       fill = "Target Variable")
```



Day of week distribution is similar between classes.

```r
# Show distribution of call duration
ggplot(training_data, aes(x = duration, fill = y)) +
  geom_density(alpha = 0.5) +
  scale_x_continuous(breaks = seq(0, 4000, by = 500)) +
  labs(title = "Distribution of Call Duration by Target Variable",
       x = "Call Duration",
       y = "Density",
       fill = "Target Variable")
```



The distribution of call duration is drastically different between classes. Calls that last less than 250 seconds have a higher density of "no" and calls that last more than 250 seconds have a higher density of "yes". I turn this continuous variable into a factor variable by binning call duration into these two categories.

```
# Define threshold
call_duration_threshold <- 250
below_call_duration <- paste("0-", call_duration_threshold, sep = "")
above_call_duration <- paste(call_duration_threshold + 1, "+", sep = "")

# Create call duration groups
training_data <- training_data %>%
  mutate(call_duration_group = case_when(
    duration <= call_duration_threshold ~ below_call_duration,
    duration > call_duration_threshold ~ above_call_duration
  ))

# Turn call duration groups into factor
training_data$call_duration_group <- factor(training_data$call_duration_group, levels = c(below_call_du
```

Note that neither of these features can be used in the final predictive model as it is not known until after the call is finished.

```
# Add call duration group to excluded features
excluded_features <- c(excluded_features, "duration", "call_duration_group")
```

```r
# Show frequency of each campaign call
frequency_per_campaign <- table(training_data$campaign)
frequency_per_campaign
```

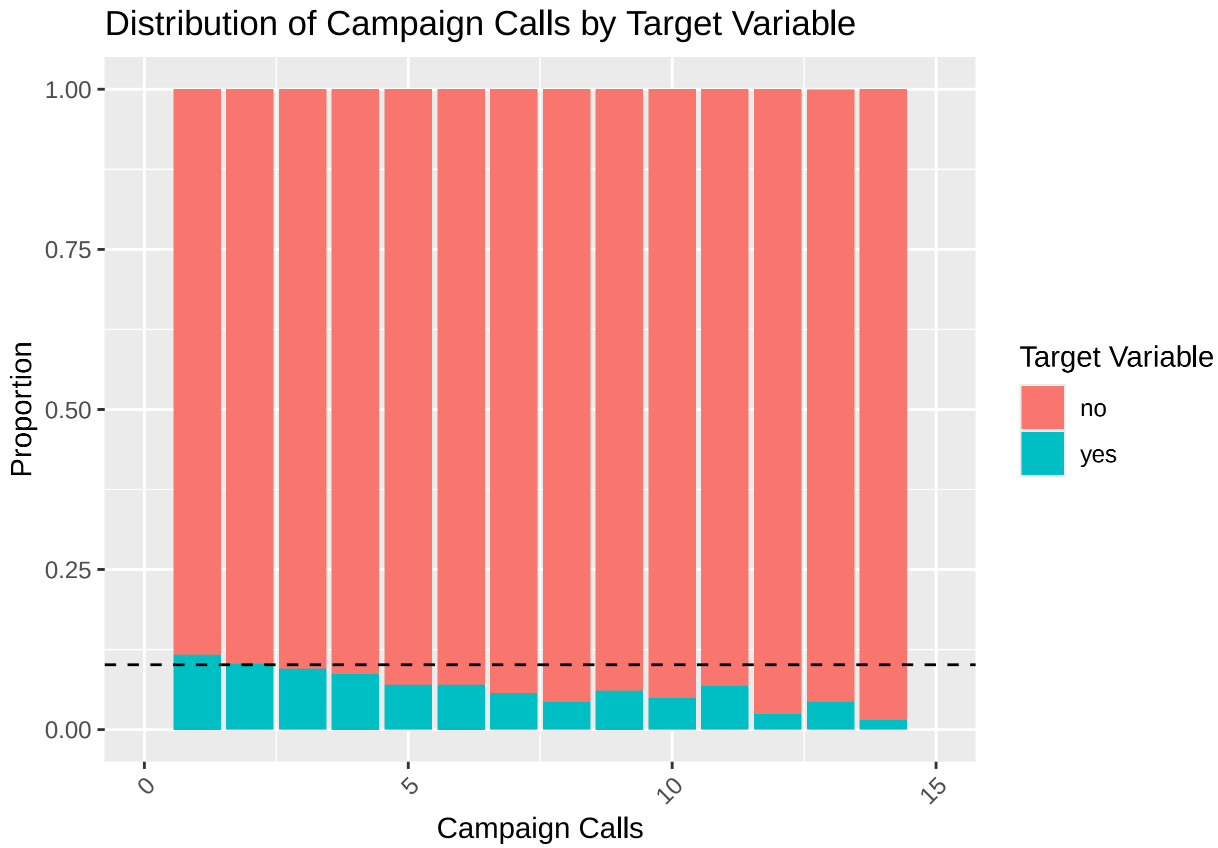
```
##
##     1     2     3     4     5     6     7     8     9    10    11    12    13
## 17044 10270  5203  2596  1576   965   619   397   279   224   176   125    92
##    14    15    16    17    18    19    20    21    22    23    24    25    26
##    69    51    50    58    33    26    30    24    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
```

The frequency distribution of campaign calls is highly left-skewed: most records have less than 10 campaign calls.

```r
campaign_calls_cutoff <- quantile(training_data$campaign, 0.99)

# Show distribution of campaign calls
ggplot(training_data, aes(x = campaign, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(training_data$y == "yes"), linetype = "dashed") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  xlim(0, campaign_calls_cutoff) +
  labs(title = "Distribution of Campaign Calls by Target Variable",
       x = "Campaign Calls",
       y = "Proportion",
       fill = "Target Variable")
```

```
## Warning: Removed 354 rows containing non-finite outside the scale range
## (‘stat_count()‘).
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_bar()‘).
```



The proportion of "yes" decreases as the number of campaign calls increases.

Number of campaign calls would be most appropriately treated as an ordinal factor variable - it is not continuous, but proportion of "yes" does decrease in proportion to increases in campaign calls.

```
# Create factor of range between min and max number of campaign calls
min_campaign_calls <- min(training_data$campaign)
max_campaign_calls <- max(training_data$campaign)
campaign_levels <- as.character(min_campaign_calls:max_campaign_calls)
training_data$campaign <- factor(training_data$campaign, levels = campaign_levels)
```

```
# 999 is the value set where no contact has been made before
no_contact <- 999
str_no_contact <- paste(no_contact)

frequency_pdays <- table(training_data$pdays)
frequency_pdays
```

```
##
##     0     1     2     3     4     5     6     7     8     9    10    11    12
##     6    22    59   343   109    36   248    53    13    38    42    24    45
##    13    14    15    16    17    18    21    22   999
##    24    14    16     2     1     2     1     1 38941
```

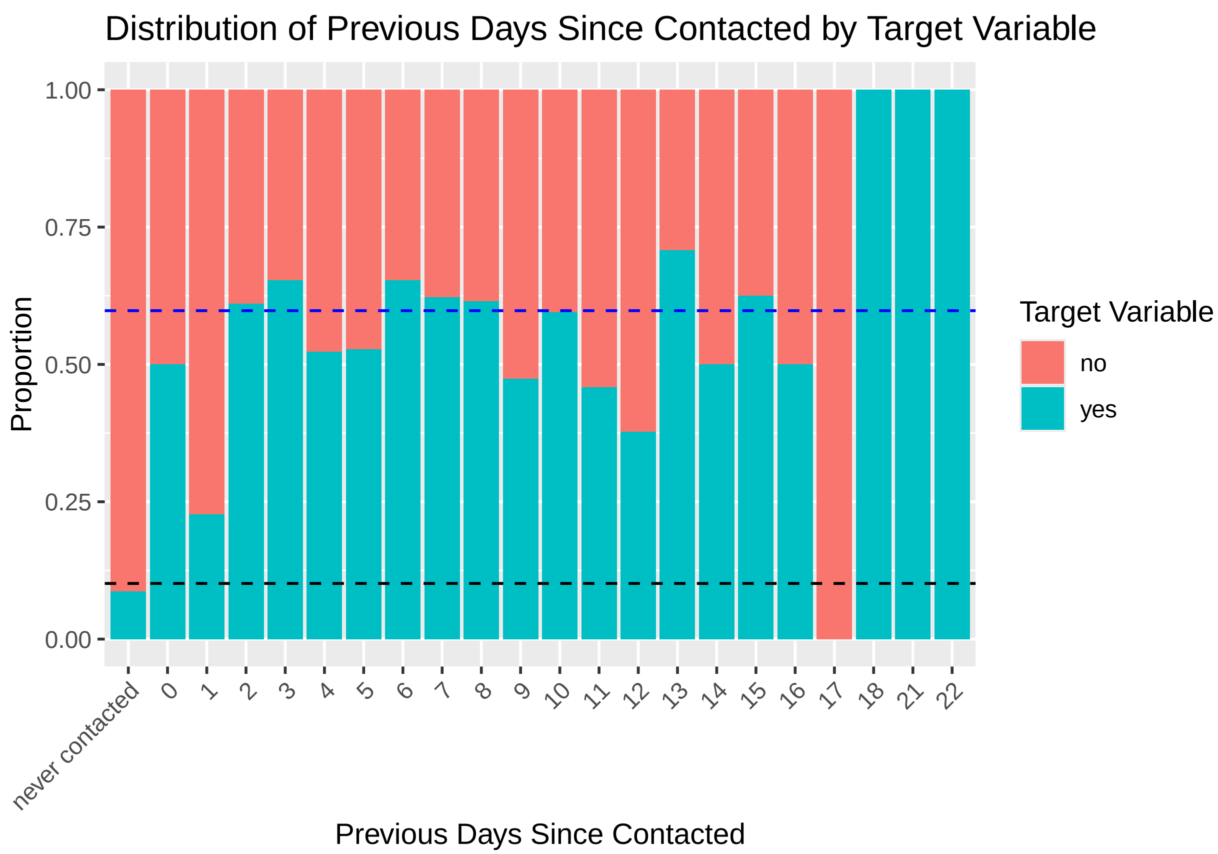
```
missing_pdays <- training_data %>%
  filter(pdays == no_contact)
nrow(missing_pdays) / nrow(training_data)
```

```
## [1] 0.9725524
```

The vast majority of customers (96%) have not been contacted before.

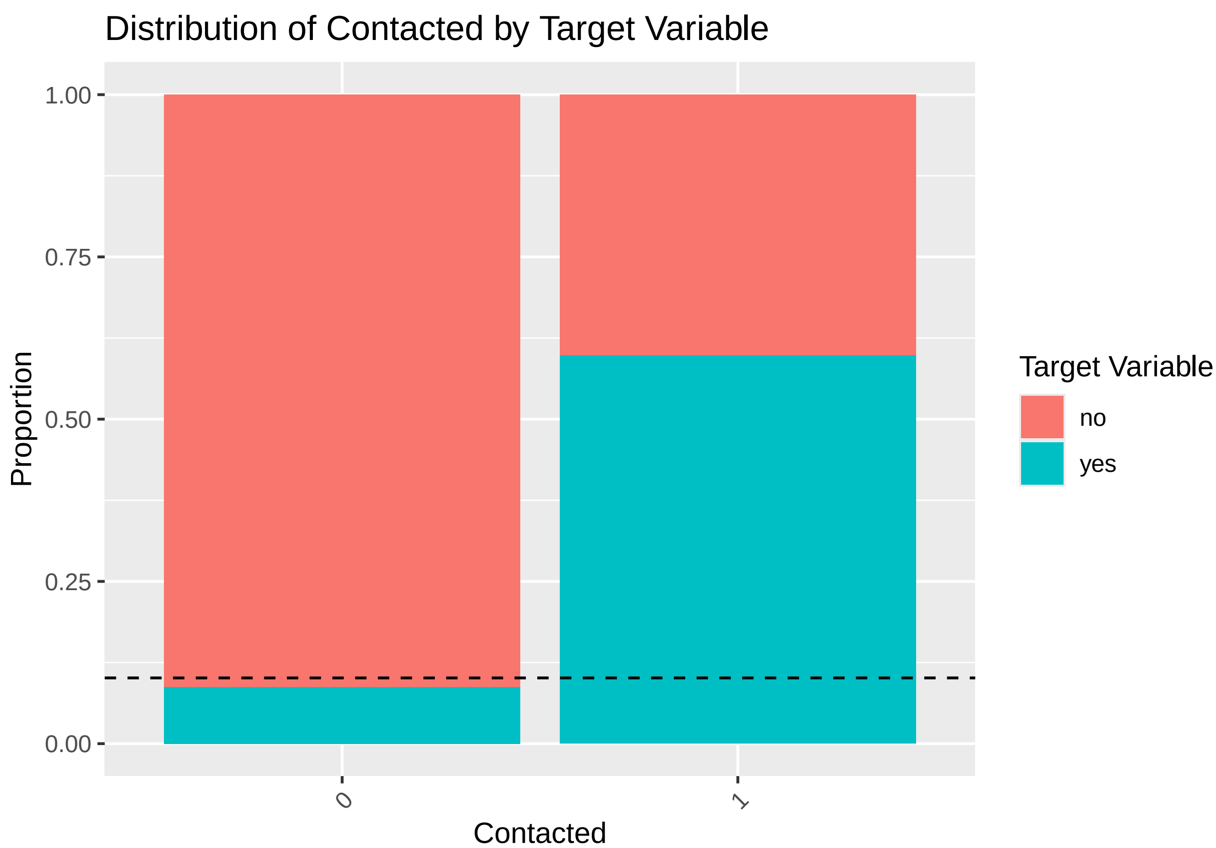
```r
# Turn pdays into a factor
min_pdays <- min(training_data$pdays)
# Get max pdays that is not the no contact value
max_pdays <- max(training_data$pdays[training_data$pdays != str_no_contact])

# Turn pdays into a factor
pdays_levels <- as.character(min_pdays:max_pdays)
training_data$pdays <- factor(training_data$pdays, levels = c(str_no_contact, pdays_levels))

# Set 999 to be the reference category
training_data$pdays <- relevel(training_data$pdays, ref = str_no_contact)

# Rename 999 to "never contacted"
levels(training_data$pdays)[levels(training_data$pdays) == str_no_contact] <- "never contacted"
```

```r
# Show distribution of pdays
ggplot(training_data, aes(x = pdays, fill = y)) +

  geom_bar(position = "fill") +

  geom_hline(yintercept = mean(training_data$y == "yes"), linetype = "dashed") +

  # Add a line to show average yes proportion of all factors apart from never contacted
  geom_hline(yintercept = mean(training_data$y[training_data$pdays != "never contacted"] == "yes"), lin

  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +

  labs(title = "Distribution of Previous Days Since Contacted by Target Variable",
       x = "Previous Days Since Contacted",
       y = "Proportion",
       fill = "Target Variable"
  )
```



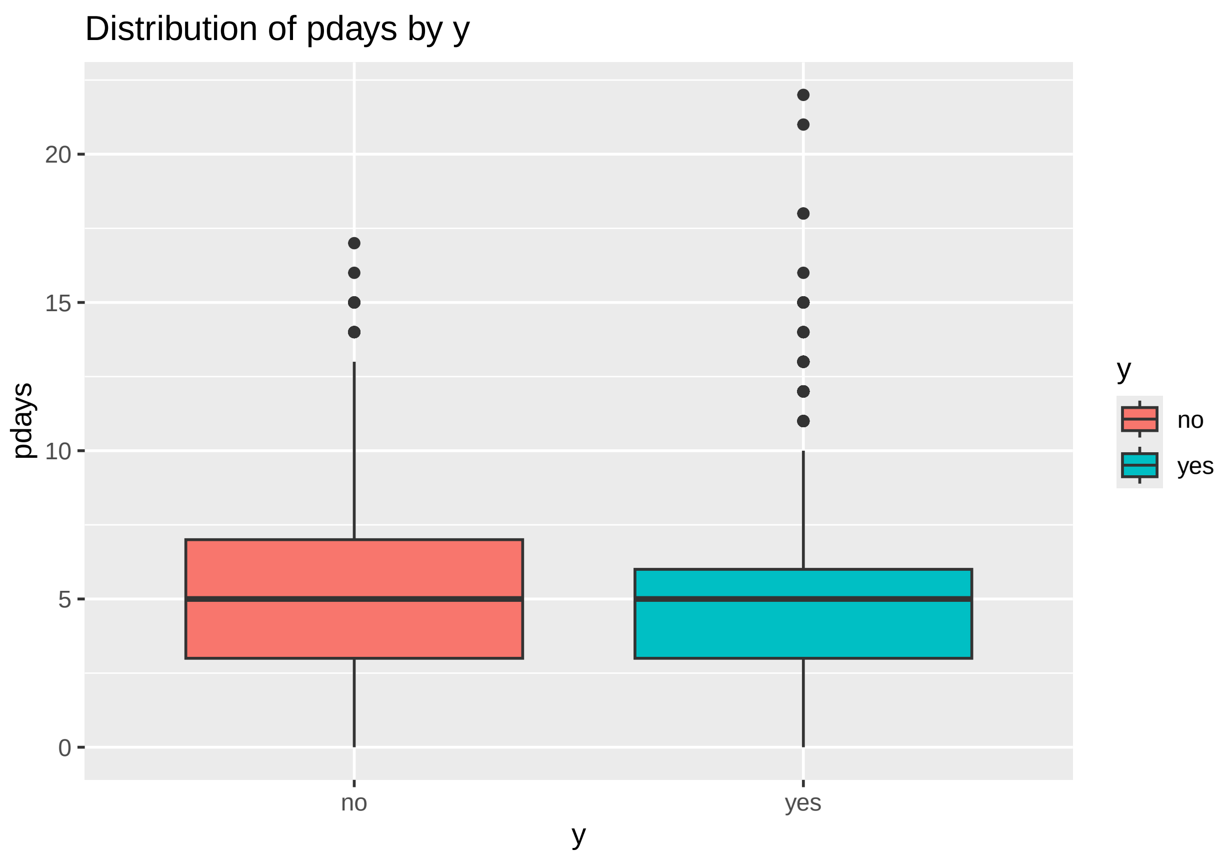
This distribution indicates that pdays may be better as a binary variable - never contacted has a lower proportion of the "Yes" class, whereas any other value has a higher proportion of the "Yes" class and this higher level appears randomly distributed around its mean (blue dotted line).

```r
# Create contacted variable
training_data <- training_data %>%
  mutate(contacted = if_else(pdays == "never contacted", 0, 1))
training_data$contacted <- factor(training_data$contacted, levels = c(0, 1))

# Show distribution of contacted
ggplot(training_data, aes(x = contacted, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(training_data$y == "yes"), linetype = "dashed") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of Contacted by Target Variable",
       x = "Contacted",
       y = "Proportion",
       fill = "Target Variable")
```



The proportion of yes is slightly lower than mean for non-contacted and much higher than mean for contacted.

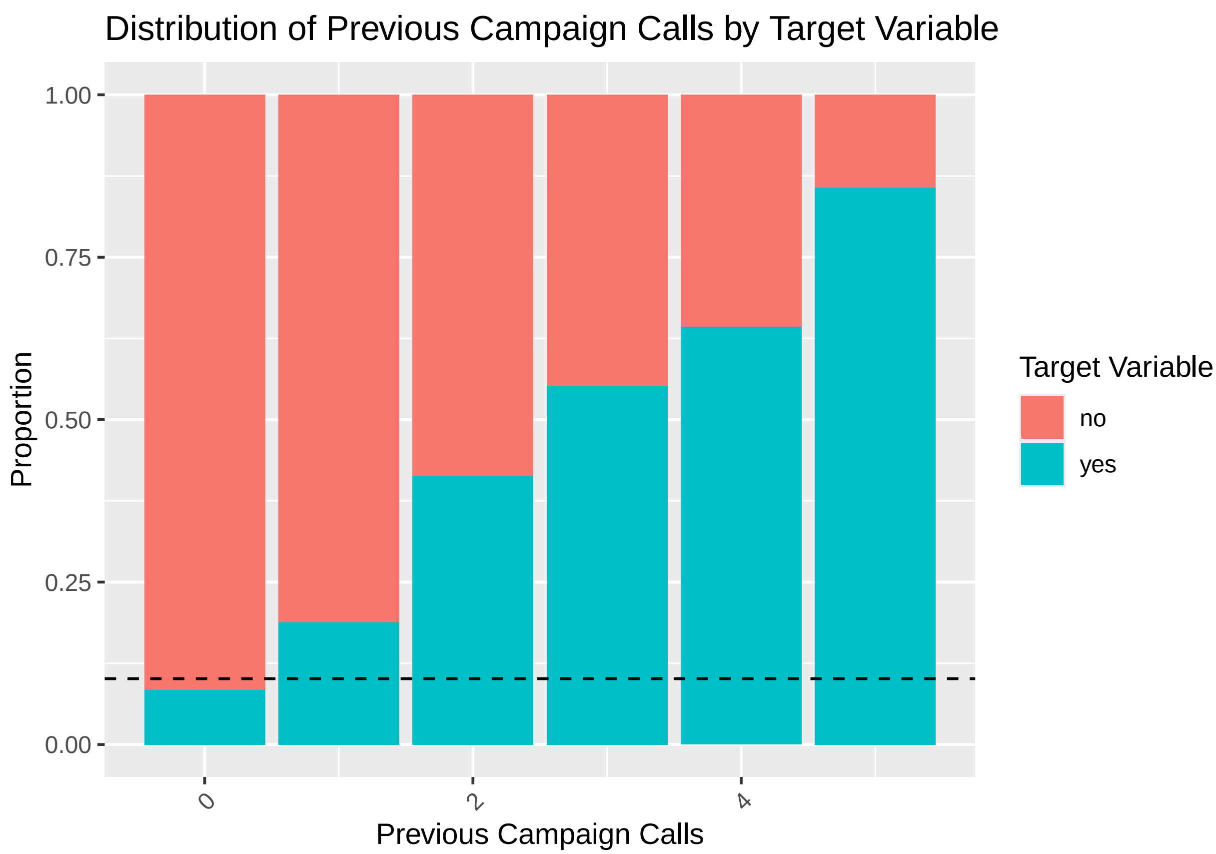
```
# For contacted rows only, extract pdays and proportion of yes
contacted_data <- training_data %>%
  filter(contacted == 1) %>%
  select(pdays, y)
contacted_data$pdays <- as.numeric(as.character(contacted_data$pdays))

# Show box plot of pdays
ggplot(contacted_data, aes(x = y, y = pdays, fill = y)) +
  geom_boxplot() +
  labs(title = "Distribution of pdays by y",
       x = "y",
       y = "pdays")
```



Within contacted records, the distribution of pdays for contacted records is similar between classes, so the exact value of pdays isn't useful.

```r
# Add pdays to excluded features
excluded_features <- c(excluded_features, "pdays")
```

```r
# Show distribution of previous
ggplot(training_data, aes(x = previous, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(training_data$y == "yes"), linetype = "dashed") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of Previous Campaign Calls by Target Variable",
       x = "Previous Campaign Calls",
       y = "Proportion",
       fill = "Target Variable")
```




```
# Show frequency of each previous campaign call
frequency_previous <- table(training_data$previous)
frequency_previous
```

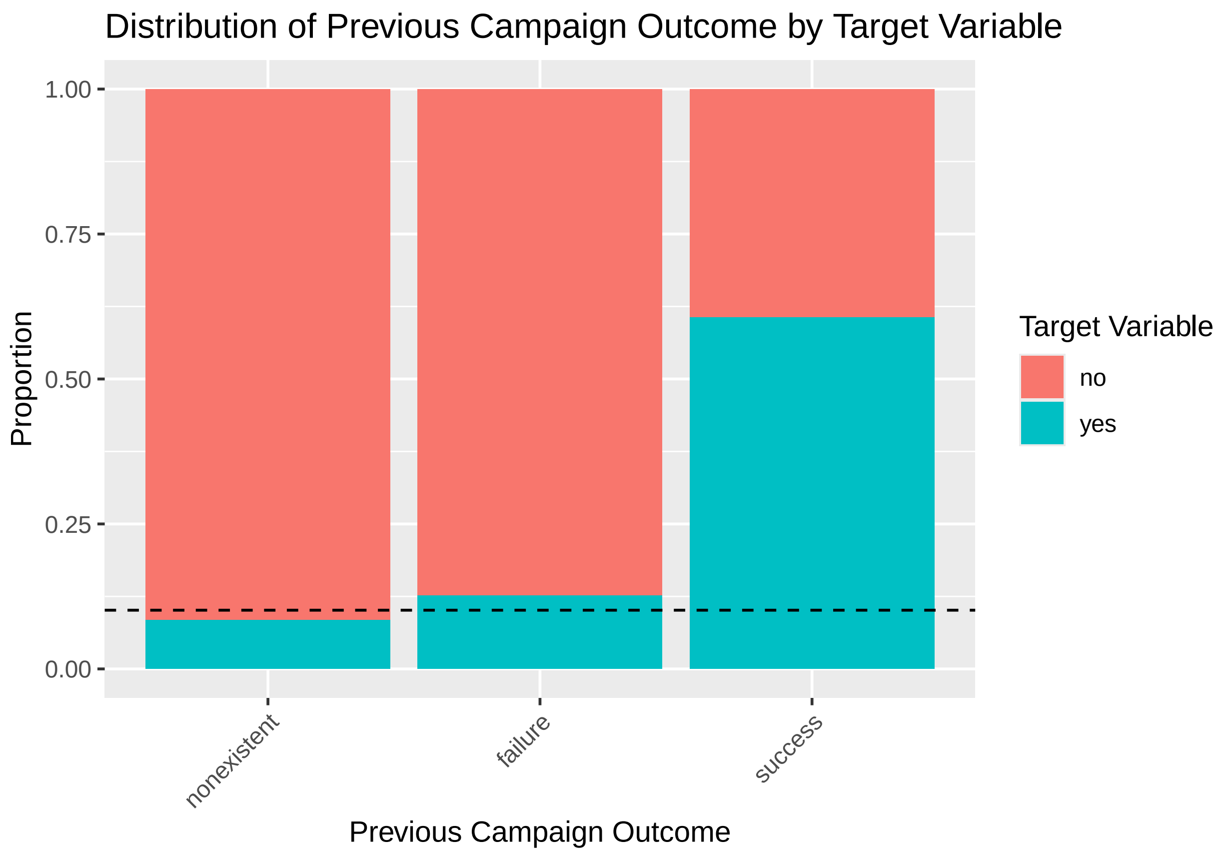
```
##
##     0     1     2     3     4     5
## 35103  4245   550   107    28     7
```

The distribution of previous campaign calls is right-skewed: 0 campaign calls has a slightly smaller proportion of “yes” than mean, 1 has a somewhat higher proportion of “yes”, and 2+ has a much higher proportion of “yes”. I treat this as an ordinal factor variable.

```
# Create factor of range between min and max number of previous campaign calls
min_previous <- min(training_data$previous)
max_previous <- max(training_data$previous)
previous_levels <- as.character(min_previous:max_previous)
training_data$previous <- factor(training_data$previous, levels = previous_levels)
```

```
# Set non-existent as the reference category
training_data$poutcome <- relevel(training_data$poutcome, ref = "nonexistent")

# Show distribution of previous campaign outcome
ggplot(training_data, aes(x = poutcome, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(training_data$y == "yes"), linetype = "dashed") +
```

```r
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of Previous Campaign Outcome by Target Variable",
       x = "Previous Campaign Outcome",
       y = "Proportion",
       fill = "Target Variable")
```

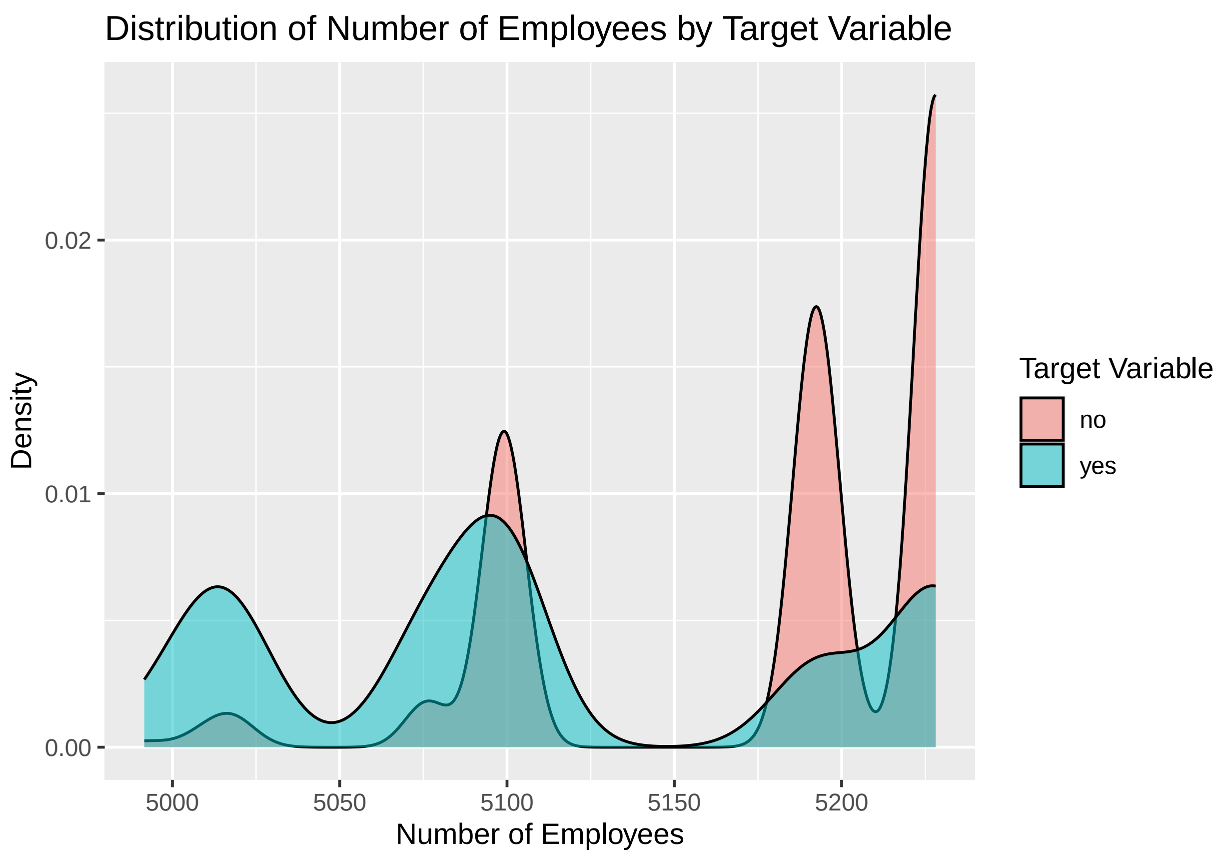


Nonexistent has a slightly lower proportion of "yes" and failure has a slightly higher proportion of "yes", but success has a much higher proportion of "yes".

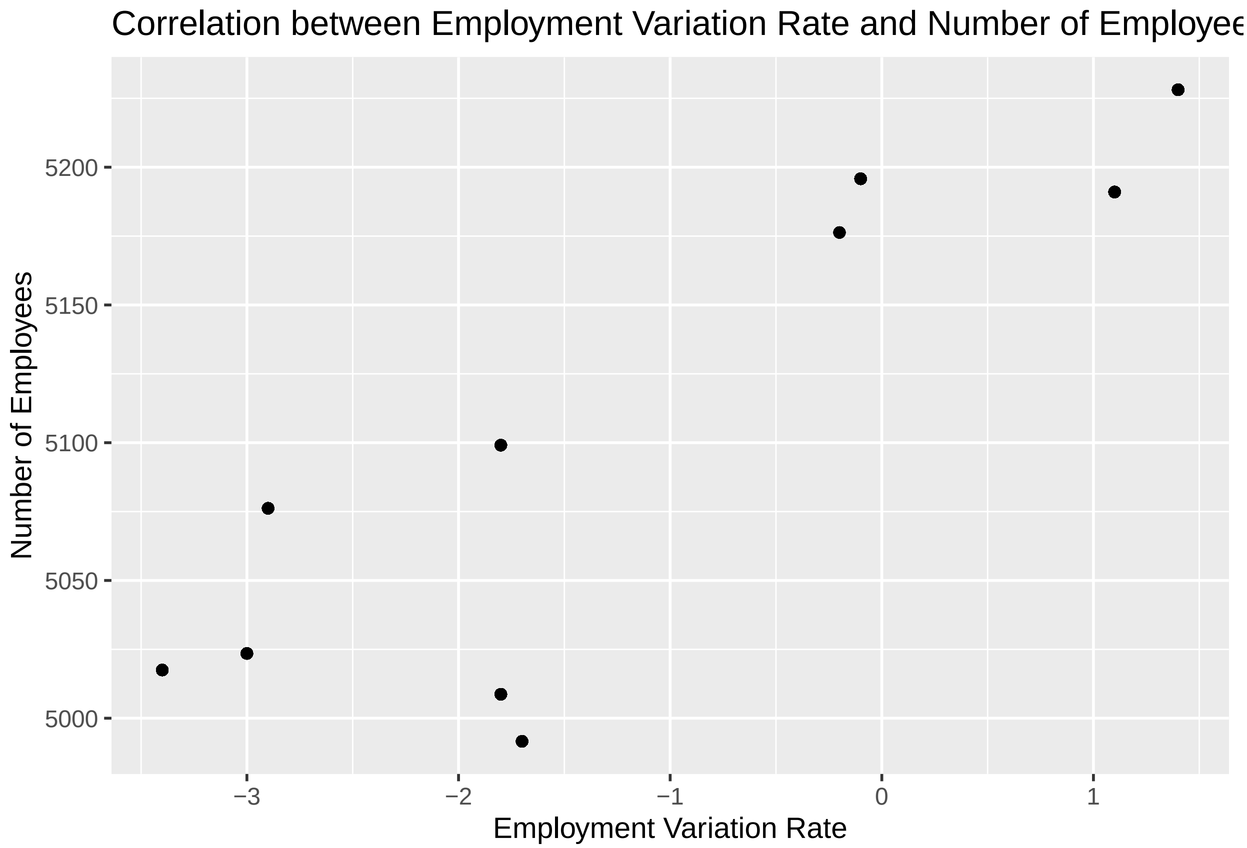
```r
# Show distribution of emp.var.rate
ggplot(training_data, aes(x = emp.var.rate, fill = y)) +
  geom_density(alpha = 0.5) +
  labs(title = "Distribution of Employment Variation Rate by Target Variable",
       x = "Employment Variation Rate",
       y = "Density",
       fill = "Target Variable")
```

Highly negative employment variation rates (up to -0.5) have higher densities of "yes" and higher employment variation rates have higher densities of "no". I keep this as a continuous variable.

```r
# Show distribution of nr.employed
ggplot(training_data, aes(x = nr.employed, fill = y)) +
  geom_density(alpha = 0.5) +
  labs(title = "Distribution of Number of Employees by Target Variable",
       x = "Number of Employees",
       y = "Density",
       fill = "Target Variable")
```

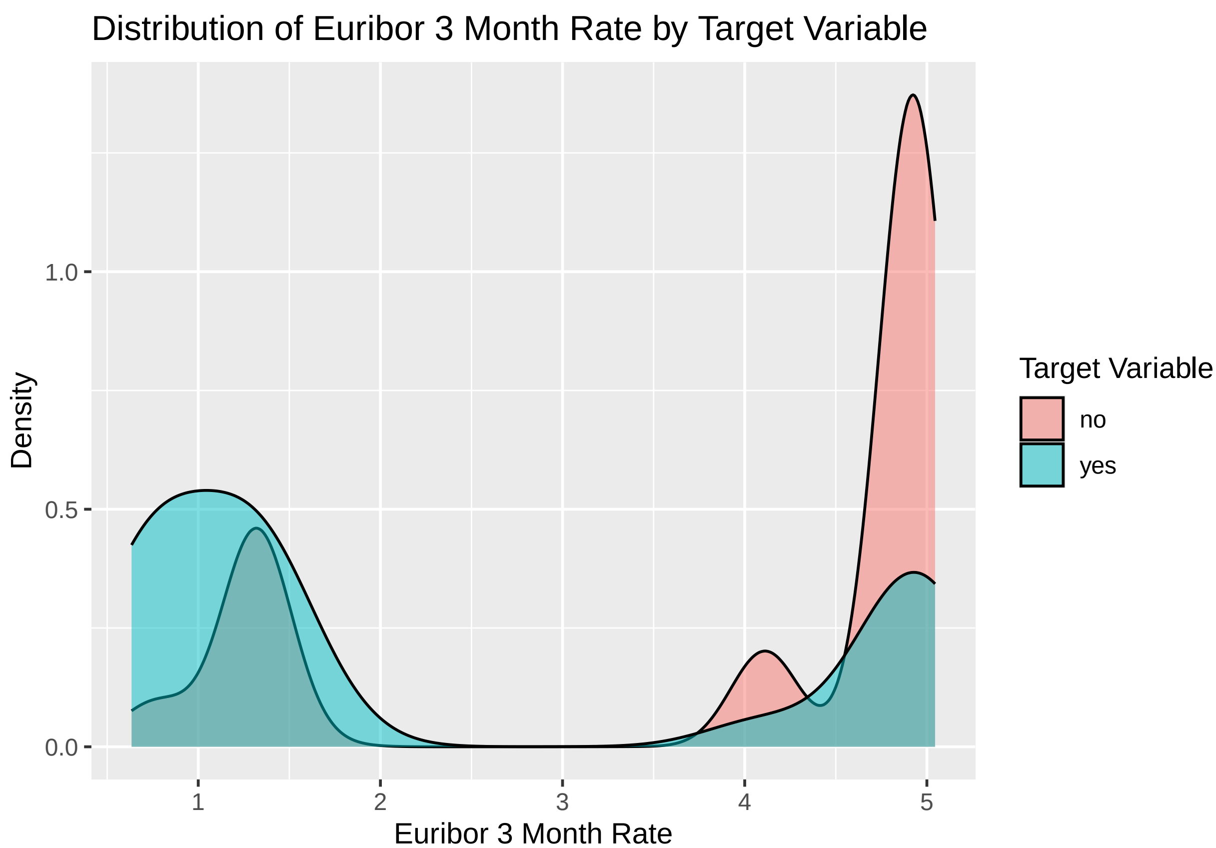



```r
# Show correlation between emp.var.rate and nr.employed
ggplot(training_data, aes(x = emp.var.rate, y = nr.employed)) +
  geom_point() +
  labs(title = "Correlation between Employment Variation Rate and Number of Employees",
       x = "Employment Variation Rate",
       y = "Number of Employees"
  )
```



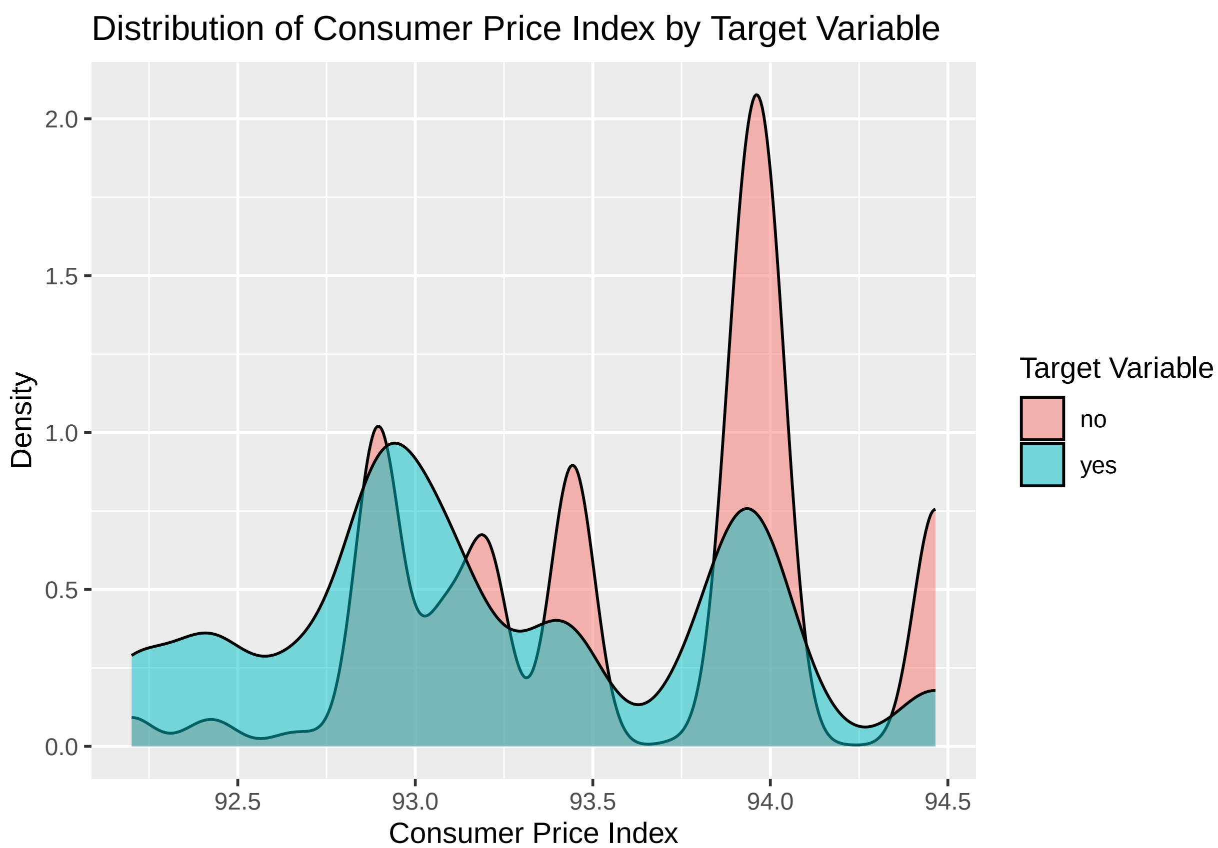
As expected, a strong positive relationship exists between the employment variation rate and the number of employees. Their distributions are similar: the lower the number of employees and variational rate, the higher the density of "yes" and the higher the number of employees and variational rate, the higher the density of "no". These variables are capturing the same variation, but I will include both for now alongside an interaction term.

```
# Show distribution of euribor3m
ggplot(training_data, aes(x = euribor3m, fill = y)) +
  geom_density(alpha = 0.5) +
  labs(title = "Distribution of Euribor 3 Month Rate by Target Variable",
       x = "Euribor 3 Month Rate",
       y = "Density",
       fill = "Target Variable")
```

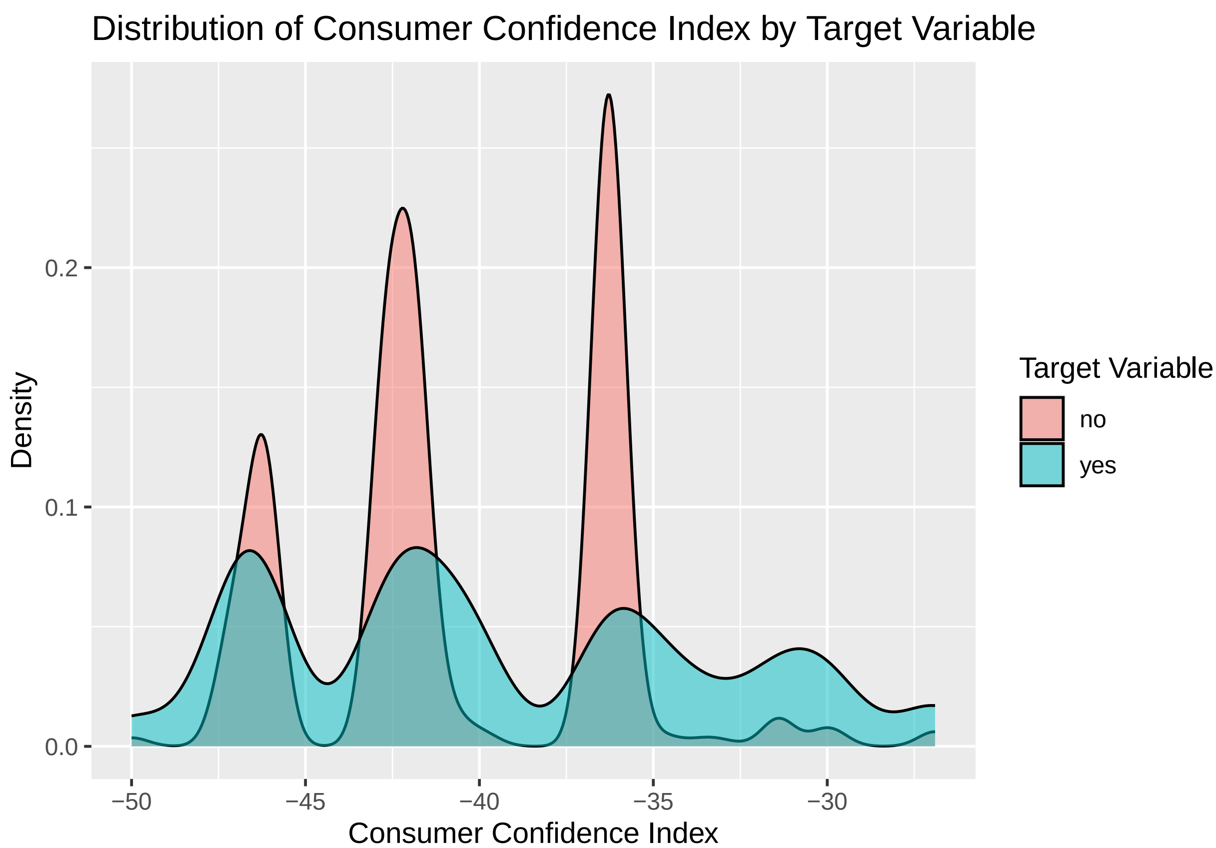


Low euribor3m rates have higher densities of "yes" and higher euribor3m rates have higher densities of "no". I keep this as a continuous variable.

```
# Show distribution of cons.price.idx
ggplot(training_data, aes(x = cons.price.idx, fill = y)) +
  geom_density(alpha = 0.5) +
  labs(title = "Distribution of Consumer Price Index by Target Variable",
       x = "Consumer Price Index",
       y = "Density",
       fill = "Target Variable")
```




```r
# Show distribution of cons.conf.idx
ggplot(training_data, aes(x = cons.conf.idx, fill = y)) +
  geom_density(alpha = 0.5) +
  labs(title = "Distribution of Consumer Confidence Index by Target Variable",
       x = "Consumer Confidence Index",
       y = "Density",
       fill = "Target Variable")
```



These variables are highly non-linear seperators of the classes but their distributions between classes are similar.

```
# Show correlation between cons.price.idx and cons.conf.idx
ggplot(training_data, aes(x = cons.price.idx, y = cons.conf.idx)) +
  geom_point() +
  labs(title = "Correlation between Consumer Price Index and Consumer Confidence Index",
       x = "Consumer Price Index",
       y = "Consumer Confidence Index"
  )
```

There is a negative, partly linear relationship between the consumer price index and consumer confidence index. Including this interaction term in the model could reduce the amount of work needed to capture the non-linear relationship of these variables with the target variable.

Modeling

```
# Initial model including all features
model_1 <- glm(y ~ ., data = training_data %>% select(-all_of(excluded_features)), family = binomial)
```

```
# Observe perfectly multicollinear variables
alias(model_1)
```

```
## Model :
## y ~ job + marital + education + housing + loan + contact + month +
##     day_of_week + campaign + previous + poutcome + emp.var.rate +
##     cons.price.idx + cons.conf.idx + euribor3m + nr.employed +
##     age_group + default_group + strategy_period + contacted
##
## Complete :
##                 (Intercept) jobadmin. jobblue-collar jobentrepreneur
## loanunknown      0           0         0              0
## poutcomesuccess  0           0         0              0
##                 jobhousemaid jobmanagement jobretired jobself-employed
## loanunknown      0            0             0          0
## poutcomesuccess  0            0             0          0
##                 jobservices jobstudent jobtechnician jobunknown maritalmarried
## loanunknown      0           0          0             0          0
```

```
## poutcomesuccess  0           0          0             0          0
##                 maritalsingle maritalunknown educationbasic.4y
## loanunknown      0             0              0
## poutcomesuccess  0             0              0
##                 educationbasic.6y educationbasic.9y educationhigh.school
## loanunknown      0                 0                 0
## poutcomesuccess  0                 0                 0
##                 educationprofessional.course educationuniversity.degree
## loanunknown      0                            0
## poutcomesuccess  0                            0
##                 educationunknown housingunknown housingyes loanyes
## loanunknown      0                1              0          0
## poutcomesuccess  0                0              0          0
##                 contacttelephone monthapr monthmay monthjun monthjul monthaug
## loanunknown      0                0        0        0        0        0
## poutcomesuccess  0                0        0        0        0        0
##                 monthsep monthoct monthnov monthdec day_of_weektue
## loanunknown      0        0        0        0        0
## poutcomesuccess  0        0        0        0        0
##                 day_of_weekwed day_of_weekthu day_of_weekfri campaign2
## loanunknown      0              0              0              0
## poutcomesuccess  0              0              0              0
##                 campaign3 campaign4 campaign5 campaign6 campaign7 campaign8
## loanunknown      0         0         0         0         0         0
## poutcomesuccess  0         0         0         0         0         0
##                 campaign9 campaign10 campaign11 campaign12 campaign13
## loanunknown      0         0          0          0          0
## poutcomesuccess  0         0          0          0          0
##                 campaign14 campaign15 campaign16 campaign17 campaign18
## loanunknown      0          0          0          0          0
## poutcomesuccess  0          0          0          0          0
##                 campaign19 campaign20 campaign21 campaign22 campaign23
## loanunknown      0          0          0          0          0
## poutcomesuccess  0          0          0          0          0
##                 campaign24 campaign25 campaign26 campaign27 campaign28
## loanunknown      0          0          0          0          0
## poutcomesuccess  0          0          0          0          0
##                 campaign29 campaign30 campaign31 campaign32 campaign33
## loanunknown      0          0          0          0          0
## poutcomesuccess  0          0          0          0          0
##                 campaign34 campaign35 campaign37 campaign39 campaign40
## loanunknown      0          0          0          0          0
## poutcomesuccess  0          0          0          0          0
##                 campaign41 campaign42 campaign43 campaign56 previous1 previous2
## loanunknown      0          0          0          0          0         0
## poutcomesuccess  0          0          0          0          1         1
##                 previous3 previous4 previous5 poutcomefailure emp.var.rate
## loanunknown      0         0         0         0               0
## poutcomesuccess  1         1         1         -1              0
##                 cons.price.idx cons.conf.idx euribor3m nr.employed
## loanunknown      0              0             0         0
## poutcomesuccess  0              0             0         0
##                 age_group30-57 age_group58+ default_groupunknown_or_yes
## loanunknown      0              0            0
```

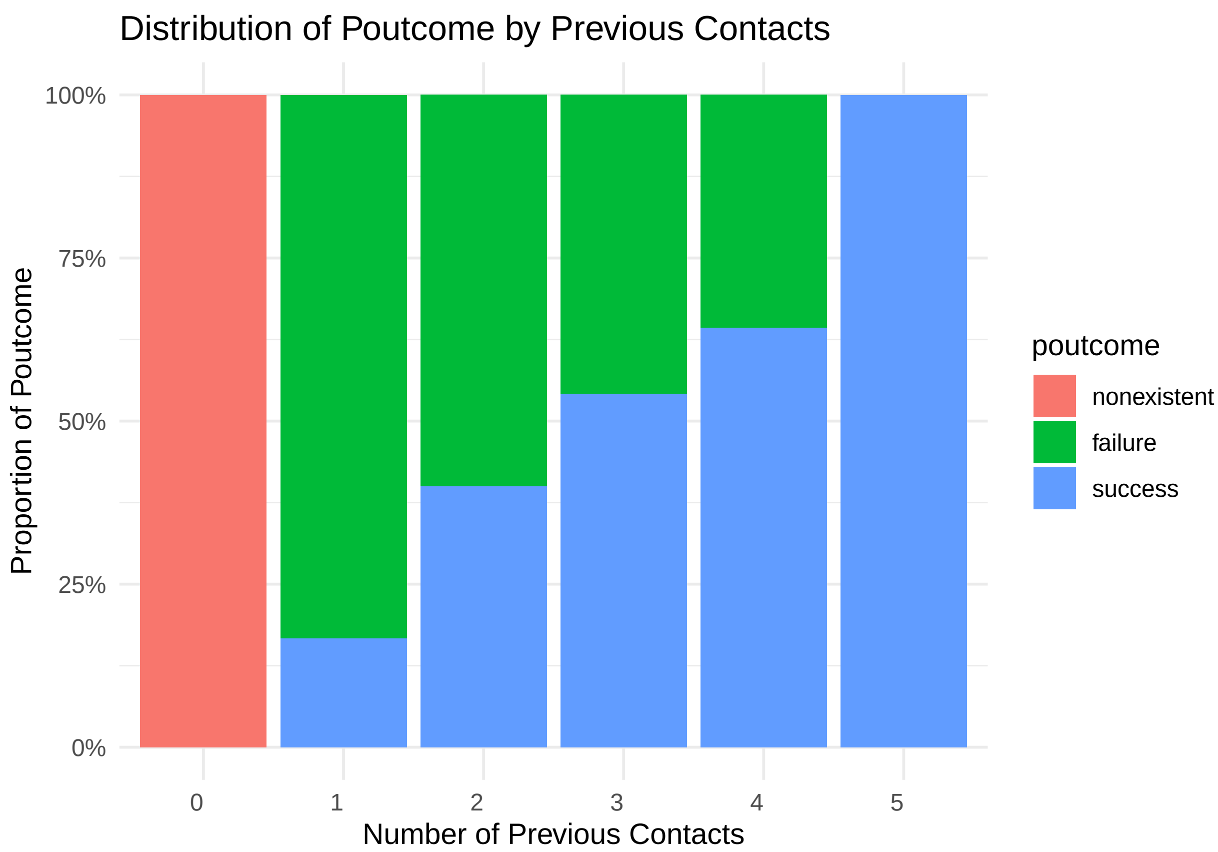
```
## poutcomesuccess  0                0            0
##                 strategy_period2 strategy_period3 contacted1
## loanunknown      0                0                0
## poutcomesuccess  0                0                0
```

The “unknown” levels of “loan” and “housing” are perfectly multicollinear, as is “poutcomesuccess” with all levels of “previous” and “poutcomefailure.”

```
# Merge loan and housing
training_data <- training_data %>%
  mutate(loan_housing_status = case_when(
    loan == "unknown" & housing == "unknown" ~ "both_unknown",
    loan == "yes" & housing == "yes" ~ "both_yes",
    loan == "yes" & housing == "no" ~ "loan_yes_housing_no",
    loan == "no" & housing == "yes" ~ "loan_no_housing_yes",
    loan == "no" & housing == "no" ~ "both_no"
  ))

# Turn loan_housing_status into a factor
training_data$loan_housing_status <- factor(training_data$loan_housing_status, levels = unique(training_
training_data$loan_housing_status <- relevel(training_data$loan_housing_status, ref = "both_unknown")

# Remove loan and housing
excluded_features <- c(excluded_features, "loan", "housing")
```

Merging loan and housing removes the multicolinearity.

```
# Create a summary table of poutcome vs previous
poutcome_previous_summary <- training_data %>%
  group_by(previous, poutcome) %>%
  summarise(count = n(), .groups = "drop")

# Create a stacked bar chart
ggplot(poutcome_previous_summary, aes(x = as.factor(previous), y = count, fill = poutcome)) +
  geom_bar(stat = "identity", position = "fill") +  # "fill" makes it a proportion-based stacked chart
  labs(x = "Number of Previous Contacts",
       y = "Proportion of Poutcome",
       title = "Distribution of Poutcome by Previous Contacts") +
  scale_y_continuous(labels = scales::percent) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 0, hjust = 1))
```



Poutcome is nonexistent when previous is 0, and is a mix of failure and success only when previous > 1.

```
# Merge poutcome and previous
training_data <- training_data %>%
  mutate(poutcome_previous = case_when(
    previous == 0 & poutcome == "nonexistent" ~ "poutcome_nonexistent_previous_0",
    previous == 1 & poutcome == "failure" ~ "poutcome_failure_previous_1",
    previous == 1 & poutcome == "success" ~ "poutcome_success_previous_1",
    previous == 2 & poutcome == "failure" ~ "poutcome_failure_previous_2",
    previous == 2 & poutcome == "success" ~ "poutcome_success_previous_2",
    previous == 3 & poutcome == "failure" ~ "poutcome_failure_previous_3",
    previous == 3 & poutcome == "success" ~ "poutcome_success_previous_3",
    previous == 4 & poutcome == "failure" ~ "poutcome_failure_previous_4",
    previous == 4 & poutcome == "success" ~ "poutcome_success_previous_4",
    previous == 5 & poutcome == "failure" ~ "poutcome_failure_previous_5",
    previous == 5 & poutcome == "success" ~ "poutcome_success_previous_5",
    previous == 6 & poutcome == "failure" ~ "poutcome_failure_previous_6",
    previous == 6 & poutcome == "success" ~ "poutcome_success_previous_6",
    previous == 7 & poutcome == "failure" ~ "poutcome_failure_previous_7",
    previous == 7 & poutcome == "success" ~ "poutcome_success_previous_7"
  ))

# Turn poutcome_previous into a factor
training_data$poutcome_previous <- factor(training_data$poutcome_previous, levels = unique(training_dat
training_data$poutcome_previous <- relevel(training_data$poutcome_previous, ref = "poutcome_nonexistent_

# Remove previous and poutcome
```

```r
excluded_features <- c(excluded_features, "previous", "poutcome")
```

Merging poutcome and previous removes the multicolinearity.

```r
# Fit model again
model_2 <- glm(y ~ ., data = training_data %>% select(-all_of(excluded_features)), family = binomial)

# Observe perfectly multicollinear variables
alias(model_2)
```

```
## Model :
## y ~ job + marital + education + contact + month + day_of_week +
##     campaign + emp.var.rate + cons.price.idx + cons.conf.idx +
##     euribor3m + nr.employed + age_group + default_group + strategy_period +
##     contacted + loan_housing_status + poutcome_previous
```

All perfectly multicollinear variables have been removed.

```r
summary(model_2)
```

```
##
## Call:
## glm(formula = y ~ ., family = binomial, data = training_data %>%
##     select(-all_of(excluded_features)))
##
## Coefficients:
##                                     Estimate Std. Error z value
## (Intercept)                       -3.093e+02  3.767e+01  -8.211
## jobadmin.                          2.875e-02  1.173e-01   0.245
## jobblue-collar                    -1.137e-01  1.224e-01  -0.929
## jobentrepreneur                   -1.359e-02  1.512e-01  -0.090
## jobhousemaid                      -6.055e-02  1.676e-01  -0.361
## jobmanagement                     -3.265e-02  1.320e-01  -0.247
## jobretired                         1.418e-01  1.467e-01   0.967
## jobself-employed                   1.143e-02  1.477e-01   0.077
## jobservices                       -8.068e-02  1.301e-01  -0.620
## jobstudent                         2.236e-01  1.510e-01   1.481
## jobtechnician                      1.960e-03  1.224e-01   0.016
## jobunknown                        -1.969e-01  2.460e-01  -0.801
## maritalmarried                     3.726e-02  6.258e-02   0.595
## maritalsingle                      1.023e-01  6.981e-02   1.465
## maritalunknown                     4.431e-01  3.664e-01   1.210
## educationbasic.4y                 -8.496e-01  6.523e-01  -1.303
## educationbasic.6y                 -7.347e-01  6.548e-01  -1.122
## educationbasic.9y                 -8.831e-01  6.517e-01  -1.355
## educationhigh.school              -8.416e-01  6.512e-01  -1.292
## educationprofessional.course      -8.317e-01  6.524e-01  -1.275
## educationuniversity.degree        -7.574e-01  6.508e-01  -1.164
## educationunknown                  -7.521e-01  6.555e-01  -1.147
## contacttelephone                  -5.583e-01  7.106e-02  -7.856
## monthapr                          -1.648e+00  1.416e-01 -11.639
## monthmay                          -1.830e+00  1.190e-01 -15.383
```

```
## monthjun                                     -2.549e+00  2.035e-01 -12.529
## monthjul                                     -1.071e+00  1.470e-01  -7.287
## monthaug                                      1.006e-01  1.747e-01   0.575
## monthsep                                     -1.115e+00  1.958e-01  -5.696
## monthoct                                     -9.986e-01  1.783e-01  -5.600
## monthnov                                     -1.771e+00  1.687e-01 -10.499
## monthdec                                     -8.174e-01  1.998e-01  -4.091
## day_of_weektue                                2.407e-01  5.934e-02   4.056
## day_of_weekwed                                3.168e-01  5.932e-02   5.341
## day_of_weekthu                                2.510e-01  5.806e-02   4.322
## day_of_weekfri                                1.853e-01  6.034e-02   3.071
## campaign2                                    -2.804e-02  4.526e-02  -0.619
## campaign3                                     5.520e-02  5.891e-02   0.937
## campaign4                                     1.815e-02  8.056e-02   0.225
## campaign5                                    -1.893e-01  1.087e-01  -1.741
## campaign6                                    -1.545e-01  1.377e-01  -1.122
## campaign7                                    -2.803e-01  1.852e-01  -1.513
## campaign8                                    -5.035e-01  2.556e-01  -1.970
## campaign9                                     9.922e-02  2.565e-01   0.387
## campaign10                                   -2.136e-01  3.175e-01  -0.673
## campaign11                                    1.325e-01  3.077e-01   0.431
## campaign12                                   -1.038e+00  5.965e-01  -1.740
## campaign13                                   -5.262e-01  5.355e-01  -0.983
## campaign14                                   -1.261e+00  1.009e+00  -1.250
## campaign15                                   -3.155e-01  7.276e-01  -0.434
## campaign16                                   -1.267e+01  2.025e+02  -0.063
## campaign17                                    2.230e-01  5.224e-01   0.427
## campaign18                                   -1.257e+01  2.515e+02  -0.050
## campaign19                                   -1.298e+01  2.726e+02  -0.048
## campaign20                                   -1.274e+01  2.642e+02  -0.048
## campaign21                                   -1.262e+01  2.948e+02  -0.043
## campaign22                                   -1.317e+01  3.273e+02  -0.040
## campaign23                                    6.481e-02  1.041e+00   0.062
## campaign24                                   -1.274e+01  3.728e+02  -0.034
## campaign25                                   -1.267e+01  5.095e+02  -0.025
## campaign26                                   -1.257e+01  5.120e+02  -0.025
## campaign27                                   -1.276e+01  4.340e+02  -0.029
## campaign28                                   -1.260e+01  5.108e+02  -0.025
## campaign29                                   -1.282e+01  4.568e+02  -0.028
## campaign30                                   -1.288e+01  5.471e+02  -0.024
## campaign31                                   -1.259e+01  5.458e+02  -0.023
## campaign32                                   -1.243e+01  7.243e+02  -0.017
## campaign33                                   -1.296e+01  7.237e+02  -0.018
## campaign34                                   -1.270e+01  8.377e+02  -0.015
## campaign35                                   -1.260e+01  6.408e+02  -0.020
## campaign37                                   -1.258e+01  1.455e+03  -0.009
## campaign39                                   -1.213e+01  1.455e+03  -0.008
## campaign40                                   -1.294e+01  1.023e+03  -0.013
## campaign41                                   -1.259e+01  1.455e+03  -0.009
## campaign42                                   -1.225e+01  1.029e+03  -0.012
## campaign43                                   -1.276e+01  1.027e+03  -0.012
## campaign56                                   -1.181e+01  1.455e+03  -0.008
## emp.var.rate                                 -2.067e+00  1.478e-01 -13.983
## cons.price.idx                                3.089e+00  2.761e-01  11.190
```

```
## cons.conf.idx                               -9.195e-03  1.426e-02  -0.645
## euribor3m                                    6.055e-01  1.369e-01   4.421
## nr.employed                                  3.382e-03  2.894e-03   1.169
## age_group30-57                              -1.185e-01  5.615e-02  -2.111
## age_group58+                                 1.226e-01  1.039e-01   1.180
## default_groupunknown_or_yes                 -2.154e-01  5.788e-02  -3.722
## strategy_period2                             3.244e-01  2.025e-01   1.602
## strategy_period3                             1.514e-01  2.295e-01   0.660
## contacted1                                   1.532e+00  2.837e-01   5.399
## loan_housing_statusboth_no                   1.438e-01  1.282e-01   1.122
## loan_housing_statusloan_no_housing_yes       1.196e-01  1.277e-01   0.936
## loan_housing_statusloan_yes_housing_no       1.420e-01  1.456e-01   0.975
## loan_housing_statusboth_yes                  7.207e-02  1.393e-01   0.517
## poutcome_previouspoutcome_failure_previous_1 -4.383e-01  6.295e-02  -6.963
## poutcome_previouspoutcome_success_previous_1 -3.058e-01  2.971e-01  -1.029
## poutcome_previouspoutcome_success_previous_2 -2.137e-01  3.266e-01  -0.654
## poutcome_previouspoutcome_failure_previous_2 -8.374e-01  1.759e-01  -4.760
## poutcome_previouspoutcome_success_previous_3 -6.247e-01  4.073e-01  -1.534
## poutcome_previouspoutcome_failure_previous_3 -8.270e-01  3.456e-01  -2.393
## poutcome_previouspoutcome_failure_previous_4 -2.034e+00  8.581e-01  -2.371
## poutcome_previouspoutcome_success_previous_4  6.687e-01  8.119e-01   0.824
## poutcome_previouspoutcome_success_previous_5  1.603e-01  1.134e+00   0.141
##                                              Pr(>|z|)
## (Intercept)                                   < 2e-16 ***
## jobadmin.                                    0.806368
## jobblue-collar                               0.352880
## jobentrepreneur                              0.928351
## jobhousemaid                                 0.717987
## jobmanagement                                0.804596
## jobretired                                   0.333787
## jobself-employed                             0.938317
## jobservices                                  0.535074
## jobstudent                                   0.138694
## jobtechnician                                0.987225
## jobunknown                                   0.423302
## maritalmarried                               0.551559
## maritalsingle                                0.142968
## maritalunknown                               0.226451
## educationbasic.4y                            0.192744
## educationbasic.6y                            0.261848
## educationbasic.9y                            0.175369
## educationhigh.school                         0.196203
## educationprofessional.course                 0.202370
## educationuniversity.degree                   0.244507
## educationunknown                             0.251211
## contacttelephone                             3.95e-15 ***
## monthapr                                      < 2e-16 ***
## monthmay                                      < 2e-16 ***
## monthjun                                      < 2e-16 ***
## monthjul                                     3.16e-13 ***
## monthaug                                     0.564972
## monthsep                                     1.22e-08 ***
## monthoct                                     2.15e-08 ***
## monthnov                                      < 2e-16 ***
```

```
## monthdec                                    4.29e-05 ***
## day_of_weektue                              4.99e-05 ***
## day_of_weekwed                              9.26e-08 ***
## day_of_weekthu                              1.54e-05 ***
## day_of_weekfri                              0.002131 **
## campaign2                                   0.535670
## campaign3                                   0.348793
## campaign4                                   0.821767
## campaign5                                   0.081607 .
## campaign6                                   0.262074
## campaign7                                   0.130223
## campaign8                                   0.048869 *
## campaign9                                   0.698860
## campaign10                                  0.501061
## campaign11                                  0.666764
## campaign12                                  0.081829 .
## campaign13                                  0.325761
## campaign14                                  0.211459
## campaign15                                  0.664616
## campaign16                                  0.950097
## campaign17                                  0.669388
## campaign18                                  0.960131
## campaign19                                  0.962034
## campaign20                                  0.961550
## campaign21                                  0.965854
## campaign22                                  0.967911
## campaign23                                  0.950347
## campaign24                                  0.972747
## campaign25                                  0.980162
## campaign26                                  0.980407
## campaign27                                  0.976555
## campaign28                                  0.980329
## campaign29                                  0.977604
## campaign30                                  0.981212
## campaign31                                  0.981601
## campaign32                                  0.986305
## campaign33                                  0.985709
## campaign34                                  0.987907
## campaign35                                  0.984311
## campaign37                                  0.993106
## campaign39                                  0.993350
## campaign40                                  0.989907
## campaign41                                  0.993101
## campaign42                                  0.990501
## campaign43                                  0.990088
## campaign56                                  0.993523
## emp.var.rate                                 < 2e-16 ***
## cons.price.idx                               < 2e-16 ***
## cons.conf.idx                               0.518952
## euribor3m                                   9.81e-06 ***
## nr.employed                                 0.242504
## age_group30-57                              0.034763 *
## age_group58+                                0.237805
## default_groupunknown_or_yes                 0.000197 ***
```

```
## strategy_period2                              0.109124
## strategy_period3                              0.509389
## contacted1                                    6.71e-08 ***
## loan_housing_statusboth_no                    0.261960
## loan_housing_statusloan_no_housing_yes        0.349250
## loan_housing_statusloan_yes_housing_no        0.329636
## loan_housing_statusboth_yes                   0.605020
## poutcome_previouspoutcome_failure_previous_1 3.32e-12 ***
## poutcome_previouspoutcome_success_previous_1 0.303385
## poutcome_previouspoutcome_success_previous_2 0.512885
## poutcome_previouspoutcome_failure_previous_2 1.93e-06 ***
## poutcome_previouspoutcome_success_previous_3 0.125070
## poutcome_previouspoutcome_failure_previous_3 0.016717 *
## poutcome_previouspoutcome_failure_previous_4 0.017748 *
## poutcome_previouspoutcome_success_previous_4 0.410115
## poutcome_previouspoutcome_success_previous_5 0.887576
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 26260  on 40039  degrees of freedom
## Residual deviance: 21211  on 39939  degrees of freedom
## AIC: 21413
##
## Number of Fisher Scoring iterations: 14
```

```r
Anova(model_2)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: y
##                    LR Chisq Df Pr(>Chisq)
## job                   14.45 11  0.2092121
## marital                3.86  3  0.2765400
## education              7.35  7  0.3937079
## contact               66.70  1  3.155e-16 ***
## month                420.23  9  < 2.2e-16 ***
## day_of_week           33.54  4  9.275e-07 ***
## campaign              52.87 41  0.1013060
## emp.var.rate         190.21  1  < 2.2e-16 ***
## cons.price.idx       125.10  1  < 2.2e-16 ***
## cons.conf.idx          0.42  1  0.5193025
## euribor3m             19.49  1  1.009e-05 ***
## nr.employed            1.37  1  0.2425292
## age_group             11.14  2  0.0038033 **
## default_group         14.30  1  0.0001557 ***
## strategy_period        3.01  2  0.2220487
## contacted             29.66  1  5.135e-08 ***
## loan_housing_status    2.24  4  0.6912700
## poutcome_previous     86.40  9  8.556e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As expected, marital and loan status are not significant.

Unexpectedly, job and age group approached significance and education was clearly insignificant. No individual levels of these factors were significant.

```
# Check multicollinearity
vif(model_2)
```

```
##                            GVIF Df GVIF^(1/(2*Df))
## job                    6.523492 11        1.088985
## marital                1.368128  3        1.053629
## education              3.182973  7        1.086217
## contact                2.529796  1        1.590533
## month               1154.897765  9        1.479590
## day_of_week            1.079551  4        1.009614
## campaign               1.100557 41        1.001169
## emp.var.rate         202.803570  1       14.240912
## cons.price.idx        83.468502  1        9.136110
## cons.conf.idx         20.736664  1        4.553753
## euribor3m            189.107074  1       13.751621
## nr.employed          149.562156  1       12.229561
## age_group              2.577742  2        1.267097
## default_group          1.128539  1        1.062327
## strategy_period      326.634669  2        4.251239
## contacted             16.508660  1        4.063085
## loan_housing_status    1.023388  4        1.002894
## poutcome_previous     20.632756  9        1.183126
```

emp.var.rate, cons.price.idx, euribor3m, nr.employed, contacted and poutcome_previous inflate standard errors so are likely multicolinear. emp.var.rate is probably not deemed significant because it is structurally multicolinear with nr.employed, so I will retain it in a predictive model.

Job, education and age_group have no multicolinearity.

```
# Get table of education level, frequency and proportion of yes
education_summary <- training_data %>%
  group_by(education) %>%
  summarise(count = n(), proportion_yes = mean(y == "yes"), .groups = "drop")
```

Education needs to be relevelled - illiterate only has 12 observations but is the base level, so has a very high standard error which makes the other levels look insignificant. A more appropriate base level would be "basic.4y".

```
education_levels <- c("basic.4y", "basic.6y", "basic.9y", "high.school", "professional.course", "univers
training_data$education <- factor(training_data$education, levels = education_levels)
```

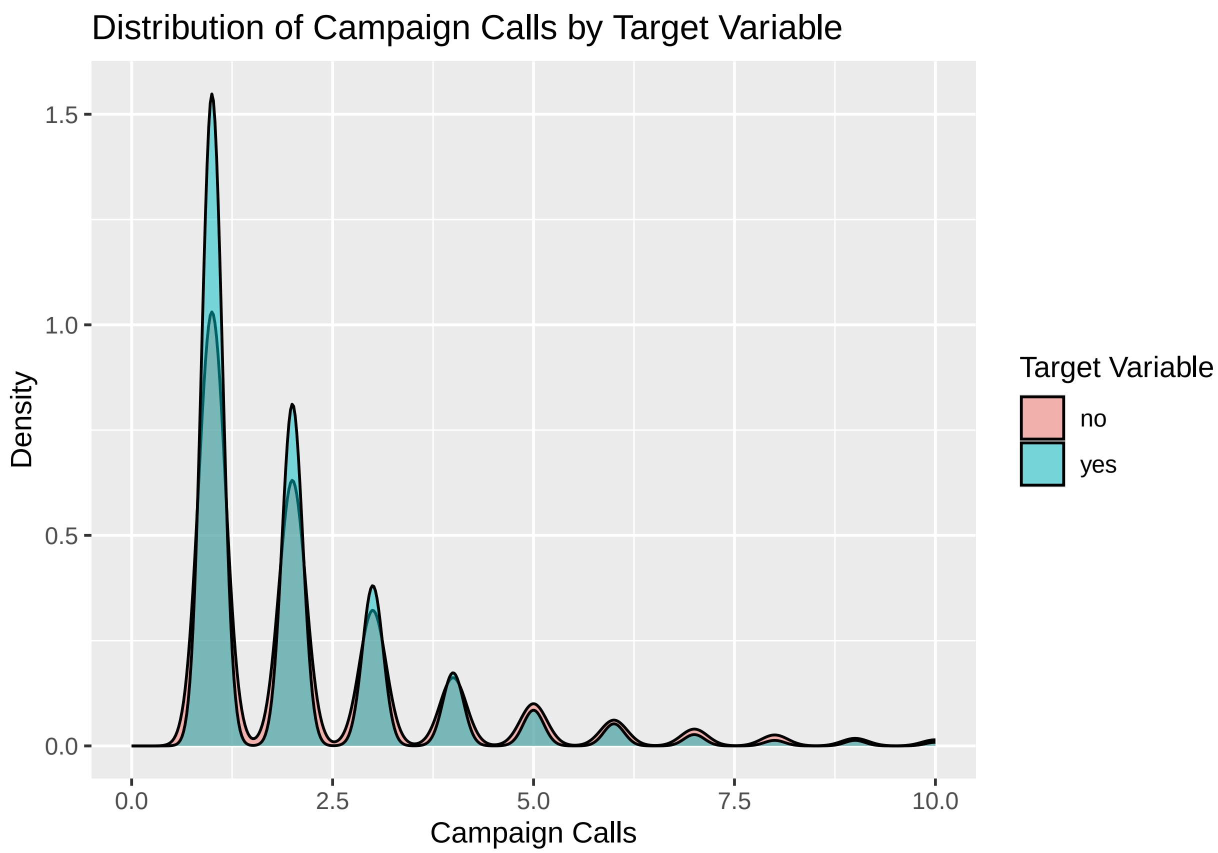
The other poorly performng factors are appropriately levelled.

Despite campaign being significant, no level past 8 is significant so I suspect the variable needs to be recast.

```
# Treat campaign as a numeric
training_data$campaign <- as.numeric(as.character(training_data$campaign))

# Restrict campaign cutoff
```

```r
campaign_calls_cutoff <- 10

# Show distribution of campaign
ggplot(training_data, aes(x = campaign, fill = y)) +
  geom_density(alpha = 0.5) +
  xlim(0, campaign_calls_cutoff) +
  labs(title = "Distribution of Campaign Calls by Target Variable",
       x = "Campaign Calls",
       y = "Density",
       fill = "Target Variable")
```

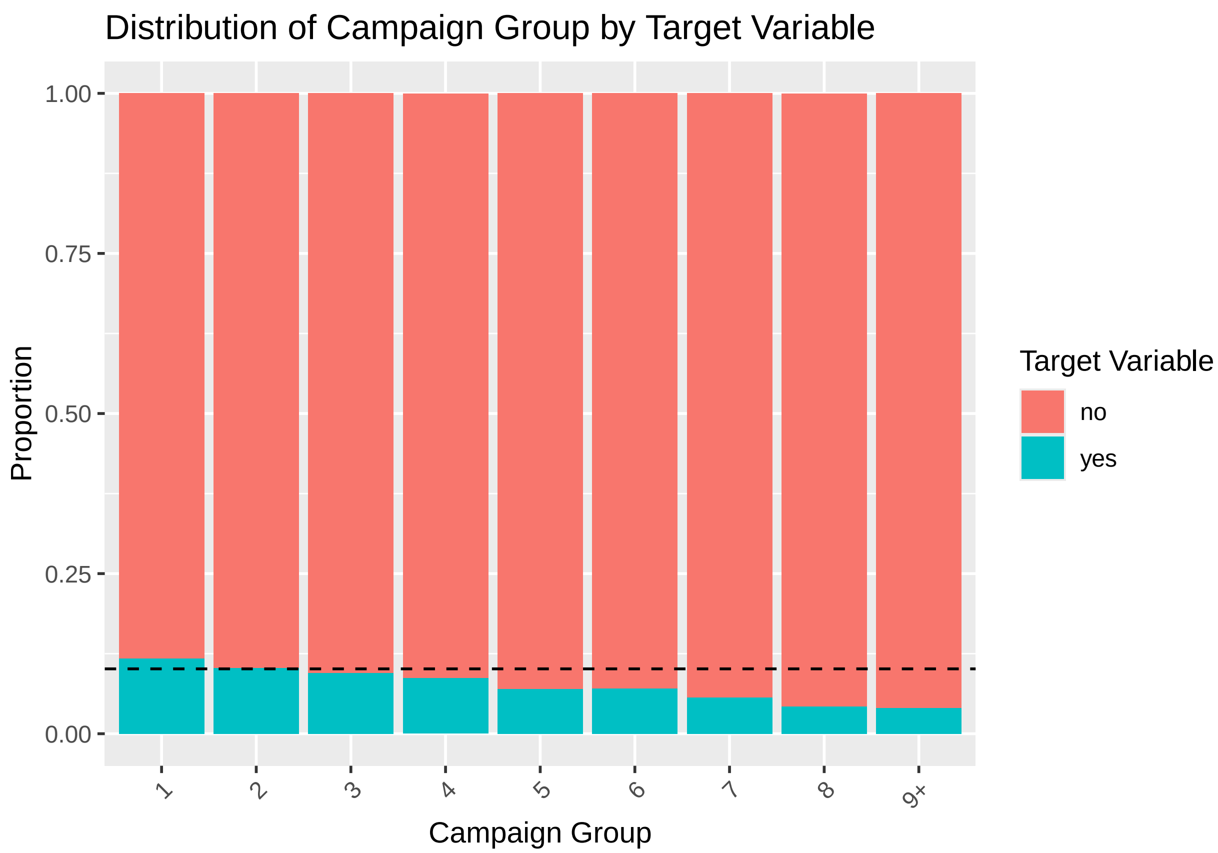
```
## Warning: Removed 867 rows containing non-finite outside the scale range
## (‘stat_density()‘).
```



Treating campaign as a continuous variable isn't appropriate. The densities of "yes" at 1, 2, 3, 4 is higher than "no" but decreasing, whereas at 5+ densities of "no" are similarly higher than "yes".

```r
# Add the continuous variable to variables to exclude
excluded_features <- c(excluded_features, "campaign")

# Try binning into 1-8 and 9+
training_data <- training_data %>%
  mutate(campaign_group = case_when(
    campaign <= 8 ~ as.character(campaign),
    campaign > 8 ~ "9+"
  ))
```

```
# Turn into factor
training_data$campaign_group <- factor(training_data$campaign_group, levels = c(as.character(1:8), "9+"
```
```
# Show distribution of campaign group
ggplot(training_data, aes(x = campaign_group, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(training_data$y == "yes"), linetype = "dashed") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of Campaign Group by Target Variable",
       x = "Campaign Group",
       y = "Proportion",
       fill = "Target Variable")
```



Rebinning campaign is a better reflection of the underlying distribution than treating it as continuous or giving each count its own level.

Similarly, the merged variable poutcome_previous has no significant levels past a previous count of 4. I will rebin this variable as well.

```
# Turn previous into numeric
training_data$previous <- as.numeric(as.character(training_data$previous))
# Rebin poutcome_previous into 1, 2, 3, 4 and 5+
training_data <- training_data %>%
  mutate(poutcome_previous = case_when(
    previous == 0 & poutcome == "nonexistent" ~ "poutcome_nonexistent_previous_0",
    previous == 1 & poutcome == "failure" ~ "poutcome_failure_previous_1",
```

```r
    previous == 1 & poutcome == "success" ~ "poutcome_success_previous_1",
    previous == 2 & poutcome == "failure" ~ "poutcome_failure_previous_2",
    previous == 2 & poutcome == "success" ~ "poutcome_success_previous_2",
    previous == 3 & poutcome == "failure" ~ "poutcome_failure_previous_3",
    previous == 3 & poutcome == "success" ~ "poutcome_success_previous_3",
    previous == 4 & poutcome == "failure" ~ "poutcome_failure_previous_4",
    previous == 4 & poutcome == "success" ~ "poutcome_success_previous_4",
    previous >= 5 & poutcome == "failure" ~ "poutcome_failure_previous_5+",
    previous >= 5 & poutcome == "success" ~ "poutcome_success_previous_5+"
  ))

# Turn poutcome_previous into a factor
training_data$poutcome_previous <- factor(training_data$poutcome_previous, levels = unique(training_data
training_data$poutcome_previous <- relevel(training_data$poutcome_previous, ref = "poutcome_nonexistent_
```

cons.price.idx and cons.conf.idx were highly significant despite their non-linear relationship with the target.

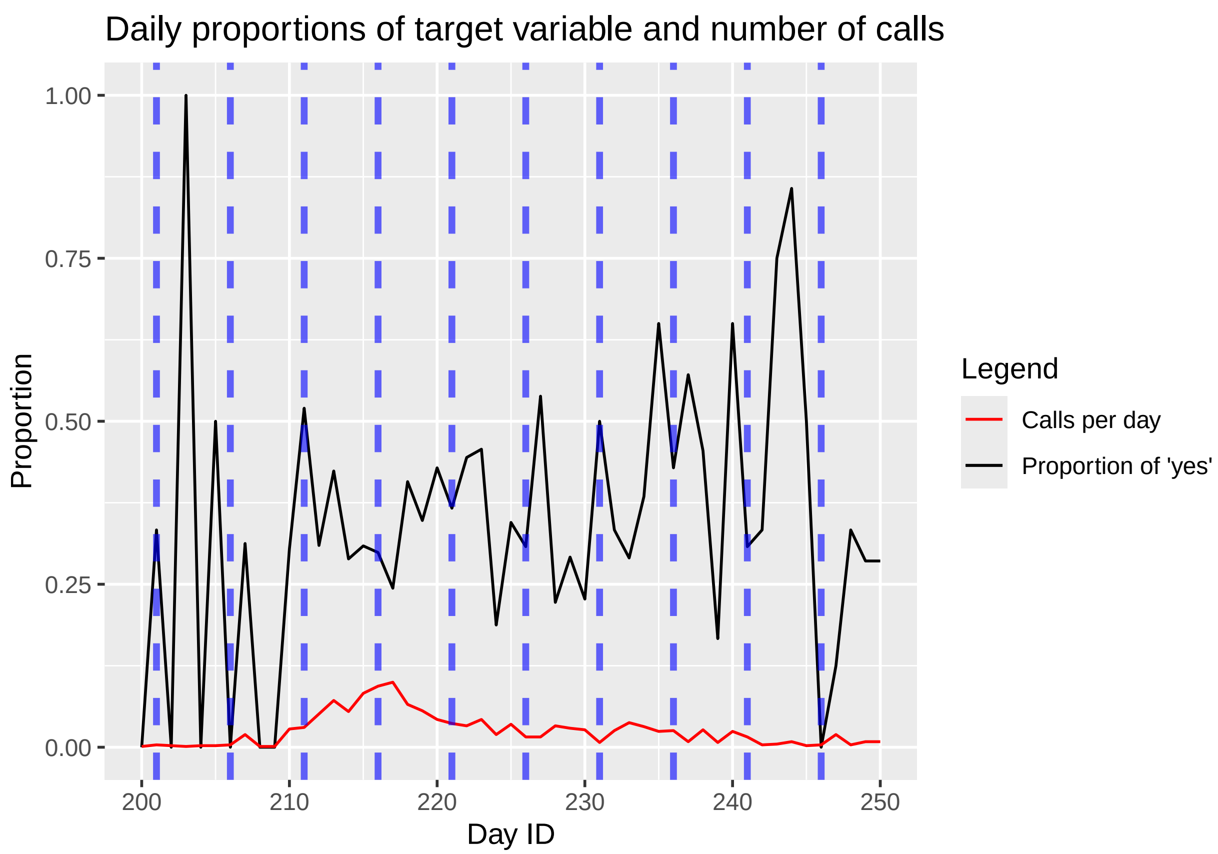
Day of month was significant at every level, but no strategy period factor level was significant. A better way to capture this non-linear relationship would be a GAM including a smooth of day_id.

Distribution of the target variable did not vary between levels of week, but every level of day of week was identified as highly significant. Similar to long-term, hort-term temporal patterns are likely not being captured properly, but day_id's smooth should capture these also.

```r
# Zoom in on daily data
zoom_interval_day <- c(200, 250)

n_days_in_week <- 5
new_weeks <- seq(1, max(training_data["day_id"]), by = n_days_in_week)

# Show daily proportions of target variable and number of calls
ggplot(training_data, aes(x = day_id)) +

  # Show proportion of "yes" as a line
  stat_summary(aes(y = as.numeric(y == "yes"), color = "Proportion of 'yes'"),
               fun = mean, geom = "line") +

  # Show number of calls as a line
  geom_line(data = calls_per_day, aes(x = day_id, y = n_calls / benchmark_day_calls, color = "Calls per
            show.legend = TRUE) +

  # Change legend labels
  scale_color_manual(values = c("Proportion of 'yes'" = "black", "Calls per day" = "red")) +

  # Limit x-axis
  xlim(zoom_interval_day) +

  # Show seperators indicating new week
  geom_vline(xintercept = new_weeks, linetype = linetype, color = color, size = size, alpha = alpha) +

  labs(
    title = "Daily proportions of target variable and number of calls",
    x = "Day ID",
    y = "Proportion",
```

```
    color = "Legend"
  )
```

```
## Warning: Removed 38971 rows containing non-finite outside the scale range
## (`stat_summary()`).

## Warning: Removed 338 rows containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 68 rows containing missing values or values outside the scale range
## (`geom_vline()`).
```



There appear to be no obvious daily patterns in the data.

I will create a GAM model with a smooth of day_id and all economic variables, a

```
# Seperate linear and nonlinear predictors
nonlinear_predictors <- c("day_id", "cons.price.idx", "cons.conf.idx", "euribor3m", "nr.employed", "emp

# Remove day_id from excluded features
excluded_features <- setdiff(excluded_features, "day_id")

get_gam_formula <- function(nonlinear_predictors, training_data, excluded_features, interactions = NA, I
  # Calculate which predictors are linear
  linear_predictor_terms <- setdiff(names(training_data), c("y", excluded_features, nonlinear_predictor
```

```r
  # Get number of unique values for each nonlinear predictor as maximum number of knots
  if(k_selection == "unique") {
    ks <- sapply(training_data %>% select(all_of(nonlinear_predictors)), function(x) length(unique(x)))
    ks <- as.vector(ks)

  # Get user-defined number of knots
  } else if(!all(is.na(ks))) {
    ks <- ks

  # Default to n knots
  } else {
    default_knots <- 10
    ks <- rep(default_knots, length(nonlinear_predictors))
  }

  # Convert unique values to array
  nonlinear_predictor_terms <- paste0("s(", nonlinear_predictors, ", k = ", ks, ")")

  # Create a formula for the model
  form_str <- paste(
    "y ~ ",
    paste(nonlinear_predictor_terms, collapse = " + "),
    " + ",
    paste(linear_predictor_terms, collapse = " + ")
  )

  # Add interactions if they exist (check if any elements of interactions aren't NA)
  if (!all(is.na(interactions))) {
    form_str <- paste(form_str, " + ", paste(interactions, collapse = " + "))
  }

  gam_formula <- as.formula(form_str)

  return(gam_formula)
}



gam_formula <- get_gam_formula(nonlinear_predictors, training_data, excluded_features, ks = c(20, 10, 1(
```

```r
# Fit model
model_3 <- gam(
  gam_formula,
  data = training_data %>% select(-all_of(excluded_features)),
  family = binomial(link = "logit")
)
```

```r
summary(model_3)
```

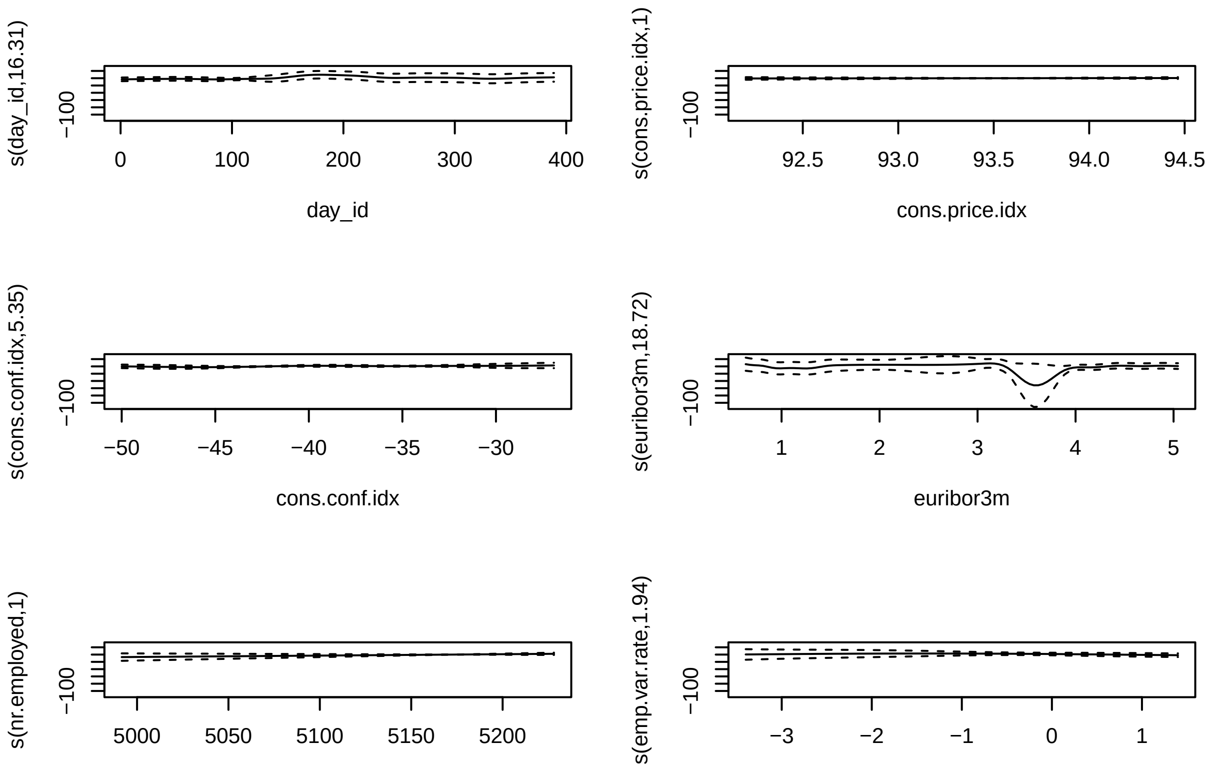
```
##
## Family: binomial
## Link function: logit
```

```
##
## Formula:
## y ~ s(day_id, k = 20) + s(cons.price.idx, k = 10) + s(cons.conf.idx,
##     k = 10) + s(euribor3m, k = 20) + s(nr.employed, k = 10) +
##     s(emp.var.rate, k = 9) + job + marital + education + contact +
##     month + day_of_week + age_group + default_group + strategy_period +
##     contacted + loan_housing_status + poutcome_previous + campaign_group
##
## Parametric coefficients:
##                                              Estimate Std. Error z value
## (Intercept)                                 -2.394242   0.936097  -2.558
## jobadmin.                                   -0.002988   0.118348  -0.025
## jobblue-collar                              -0.117057   0.123381  -0.949
## jobentrepreneur                             -0.032926   0.153242  -0.215
## jobhousemaid                                -0.129471   0.169402  -0.764
## jobmanagement                               -0.024877   0.133155  -0.187
## jobretired                                   0.082395   0.147790   0.558
## jobself-employed                            -0.012704   0.149285  -0.085
## jobservices                                 -0.063893   0.131015  -0.488
## jobstudent                                   0.126653   0.152736   0.829
## jobtechnician                               -0.035370   0.123527  -0.286
## jobunknown                                  -0.352975   0.247290  -1.427
## maritalmarried                               0.020353   0.063302   0.322
## maritalsingle                                0.088929   0.070535   1.261
## maritalunknown                               0.494205   0.365519   1.352
## educationbasic.6y                            0.146698   0.106955   1.372
## educationbasic.9y                           -0.002124   0.085679  -0.025
## educationhigh.school                        -0.001562   0.084108  -0.019
## educationprofessional.course                 0.007323   0.093740   0.078
## educationuniversity.degree                   0.055403   0.084321   0.657
## educationilliterate                          0.964031   0.641075   1.504
## educationunknown                             0.119821   0.111731   1.072
## contacttelephone                            -0.584382   0.078194  -7.474
## monthapr                                     0.623069   0.711962   0.875
## monthmay                                    -0.426163   0.887824  -0.480
## monthjun                                    -0.869898   1.020574  -0.852
## monthjul                                    -0.265757   1.000979  -0.265
## monthaug                                    -0.732983   1.446366  -0.507
## monthsep                                     2.503418   1.939085   1.291
## monthoct                                     0.905894   2.480913   0.365
## monthnov                                     1.686447   2.080532   0.811
## monthdec                                     1.358307   1.968085   0.690
## day_of_weektue                               0.262778   0.060713   4.328
## day_of_weekwed                               0.361502   0.061391   5.889
## day_of_weekthu                               0.195202   0.061720   3.163
## day_of_weekfri                               0.229285   0.063315   3.621
## age_group30-57                              -0.093161   0.057009  -1.634
## age_group58+                                 0.092944   0.105106   0.884
## default_groupunknown_or_yes                 -0.173089   0.058343  -2.967
## strategy_period2                            -0.109539   1.011989  -0.108
## strategy_period3                            -0.223127   1.074317  -0.208
## contacted1                                   1.429688   0.284590   5.024
## loan_housing_statusboth_no                   0.134093   0.129002   1.039
## loan_housing_statusloan_no_housing_yes       0.103597   0.128500   0.806
```

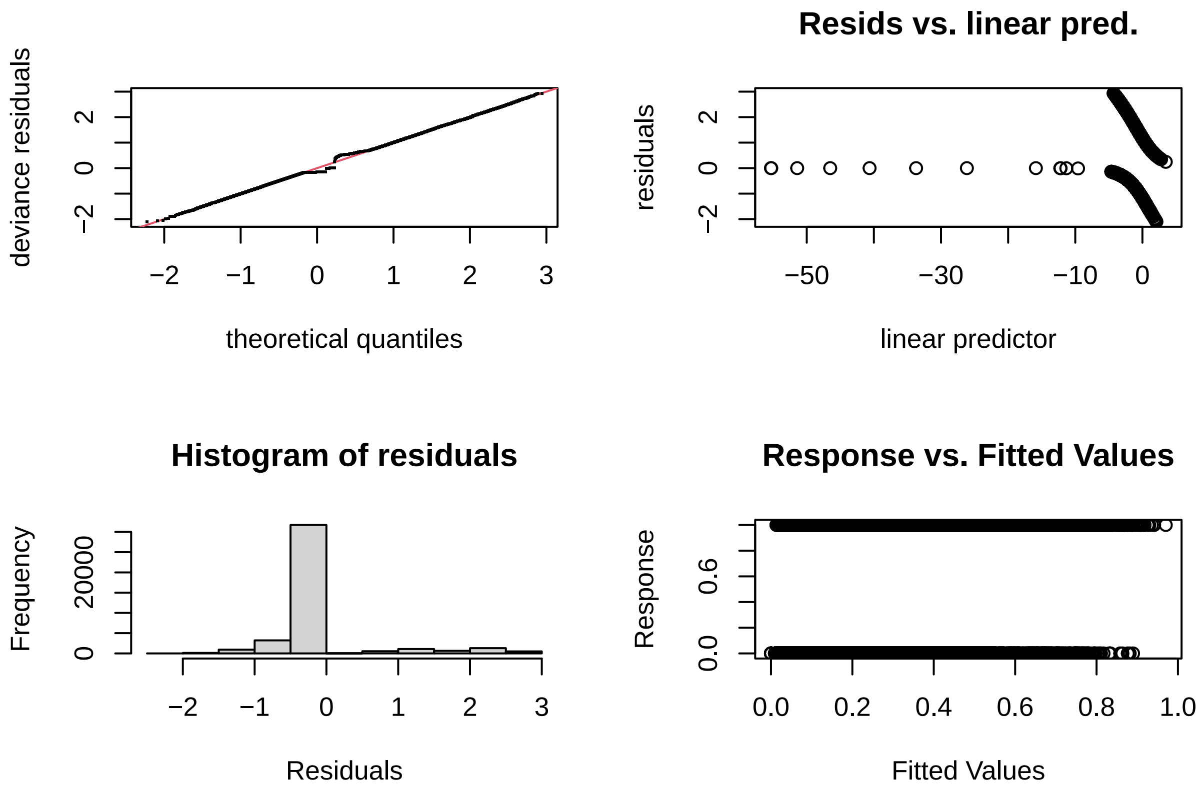
```
## loan_housing_statusloan_yes_housing_no         0.133053   0.146713   0.907
## loan_housing_statusboth_yes                    0.054561   0.140247   0.389
## poutcome_previouspoutcome_failure_previous_1  -0.380501   0.063879  -5.957
## poutcome_previouspoutcome_success_previous_1  -0.263037   0.298216  -0.882
## poutcome_previouspoutcome_success_previous_2  -0.102094   0.327767  -0.311
## poutcome_previouspoutcome_failure_previous_2  -0.774799   0.176044  -4.401
## poutcome_previouspoutcome_success_previous_3  -0.477425   0.408479  -1.169
## poutcome_previouspoutcome_failure_previous_3  -0.790794   0.343544  -2.302
## poutcome_previouspoutcome_failure_previous_4  -1.860602   0.843022  -2.207
## poutcome_previouspoutcome_success_previous_4   0.863230   0.815813   1.058
## poutcome_previouspoutcome_success_previous_5+  0.377720   1.141051   0.331
## campaign_group2                                0.000620   0.045882   0.014
## campaign_group3                                0.085693   0.059644   1.437
## campaign_group4                                0.056044   0.081313   0.689
## campaign_group5                               -0.163918   0.109793  -1.493
## campaign_group6                               -0.116668   0.139192  -0.838
## campaign_group7                               -0.262236   0.187494  -1.399
## campaign_group8                               -0.470007   0.257418  -1.826
## campaign_group9+                              -0.327875   0.146640  -2.236
##                                               Pr(>|z|)
## (Intercept)                                   0.010537 *
## jobadmin.                                     0.979855
## jobblue-collar                                0.342749
## jobentrepreneur                               0.829876
## jobhousemaid                                  0.444699
## jobmanagement                                 0.851795
## jobretired                                    0.577176
## jobself-employed                              0.932180
## jobservices                                   0.625781
## jobstudent                                    0.406976
## jobtechnician                                 0.774622
## jobunknown                                    0.153473
## maritalmarried                                0.747816
## maritalsingle                                 0.207392
## maritalunknown                                0.176355
## educationbasic.6y                             0.170192
## educationbasic.9y                             0.980222
## educationhigh.school                          0.985182
## educationprofessional.course                  0.937732
## educationuniversity.degree                    0.511152
## educationilliterate                           0.132640
## educationunknown                              0.283536
## contacttelephone                              7.81e-14 ***
## monthapr                                      0.381496
## monthmay                                      0.631221
## monthjun                                      0.394013
## monthjul                                      0.790626
## monthaug                                      0.612312
## monthsep                                      0.196693
## monthoct                                      0.715003
## monthnov                                      0.417604
## monthdec                                      0.490089
## day_of_weektue                                1.50e-05 ***
## day_of_weekwed                                3.90e-09 ***
```

```
## day_of_weekthu                               0.001563 **
## day_of_weekfri                               0.000293 ***
## age_group30-57                               0.102231
## age_group58+                                 0.376542
## default_groupunknown_or_yes                  0.003010 **
## strategy_period2                             0.913804
## strategy_period3                             0.835469
## contacted1                                   5.07e-07 ***
## loan_housing_statusboth_no                   0.298590
## loan_housing_statusloan_no_housing_yes       0.420128
## loan_housing_statusloan_yes_housing_no       0.364461
## loan_housing_statusboth_yes                  0.697251
## poutcome_previouspoutcome_failure_previous_1  2.58e-09 ***
## poutcome_previouspoutcome_success_previous_1  0.377758
## poutcome_previouspoutcome_success_previous_2  0.755434
## poutcome_previouspoutcome_failure_previous_2  1.08e-05 ***
## poutcome_previouspoutcome_success_previous_3  0.242490
## poutcome_previouspoutcome_failure_previous_3  0.021343 *
## poutcome_previouspoutcome_failure_previous_4  0.027310 *
## poutcome_previouspoutcome_success_previous_4  0.290000
## poutcome_previouspoutcome_success_previous_5+ 0.740623
## campaign_group2                              0.989219
## campaign_group3                              0.150792
## campaign_group4                              0.490670
## campaign_group5                              0.135445
## campaign_group6                              0.401929
## campaign_group7                              0.161921
## campaign_group8                              0.067873 .
## campaign_group9+                             0.025357 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                      edf Ref.df  Chi.sq p-value
## s(day_id)         16.310 17.636 165.010  <2e-16 ***
## s(cons.price.idx)  1.000  1.000   0.149   0.700
## s(cons.conf.idx)   5.348  5.785   8.435   0.282
## s(euribor3m)      18.720 18.933 142.050  <2e-16 ***
## s(nr.employed)     1.000  1.000   1.824   0.177
## s(emp.var.rate)    1.940  2.012   2.013   0.405
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.199   Deviance explained = 20.5%
## UBRE = -0.47308  Scale est. = 1         n = 40040
```

```r
# Show smooth on all non-linear plots
par(mfrow = c(3, 2))
plot(model_3)
```




```r
# Show diagnostic plots
par(mfrow = c(1, 1))

gam.check(model_3)
```




```
##
## Method: UBRE   Optimizer: outer newton
## full convergence after 16 iterations.
## Gradient range [-2.208968e-07,1.869313e-06]
## (score -0.4730841 & scale 1).
## eigenvalue range [-2.949874e-11,9.382255e-06].
## Model rank =  136 / 136
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                        k'   edf k-index p-value
## s(day_id)         19.00 16.31    0.91   0.015 *
## s(cons.price.idx)  9.00  1.00    0.93   0.145
## s(cons.conf.idx)   9.00  5.35    0.93   0.145
## s(euribor3m)      19.00 18.72    0.94   0.220
## s(nr.employed)     9.00  1.00    0.94   0.340
## s(emp.var.rate)    8.00  1.94    0.94   0.215
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Applying a smooth to day_id and all economic variables renders all levels of month insignificant except for May, so is effectively capturing the long-term temporal structure of the data. Day_of_week remains significant, so the short-term temporal structure of the data is not fully captured by the smooth of day_id. Any further time series exploration is out of the scope of this analysis, but I expect more the advanced machine learning algorithms to use these day IDs to capture this structure.

cons.price.idx, cons.conf.idx, emp.var.rate, nr.employed could be modelled as linear predictors, but day_id and euribor3m are better modelled as non-linear predictors.

```r
# Redefine non-linear predictors
nonlinear_predictors <- c("day_id", "euribor3m")
ks <- c(20, 20)
gam_formula <- get_gam_formula(nonlinear_predictors, training_data, excluded_features)
```

```r
# Fit model
model_4 <- gam(
  gam_formula,
  data = training_data %>% select(-all_of(excluded_features)),
  family = binomial(link = "logit")
)
```

```r
# Model diagnostics
summary(model_4)
```

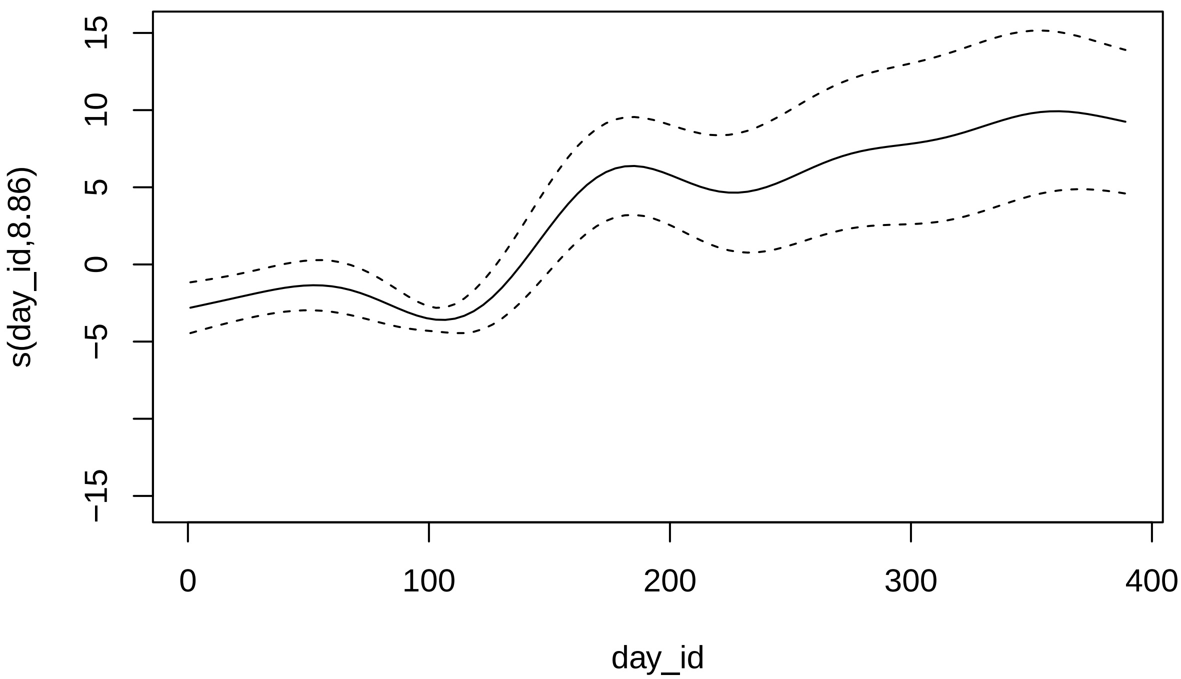
```
##
## Family: binomial
## Link function: logit
##
## Formula:
## y ~ s(day_id, k = 10) + s(euribor3m, k = 10) + job + marital +
##     education + contact + month + day_of_week + emp.var.rate +
##     cons.price.idx + cons.conf.idx + nr.employed + age_group +
##     default_group + strategy_period + contacted + loan_housing_status +
##     poutcome_previous + campaign_group
##
## Parametric coefficients:
##                                                  Estimate Std. Error z value
## (Intercept)                                    -2.006e+02  6.481e+01  -3.095
## jobadmin.                                       7.217e-03  1.181e-01   0.061
## jobblue-collar                                 -1.150e-01  1.232e-01  -0.933
## jobentrepreneur                                -3.488e-02  1.527e-01  -0.228
## jobhousemaid                                   -8.561e-02  1.683e-01  -0.509
## jobmanagement                                  -3.310e-02  1.329e-01  -0.249
## jobretired                                      1.209e-01  1.472e-01   0.821
## jobself-employed                                6.658e-03  1.487e-01   0.045
## jobservices                                    -7.584e-02  1.309e-01  -0.580
## jobstudent                                      1.684e-01  1.522e-01   1.106
## jobtechnician                                  -2.145e-02  1.233e-01  -0.174
## jobunknown                                     -2.746e-01  2.466e-01  -1.114
## maritalmarried                                  1.685e-02  6.299e-02   0.268
## maritalsingle                                   8.308e-02  7.021e-02   1.183
## maritalunknown                                  4.661e-01  3.674e-01   1.269
## educationbasic.6y                               1.350e-01  1.065e-01   1.267
## educationbasic.9y                              -1.268e-02  8.515e-02  -0.149
## educationhigh.school                            8.295e-03  8.362e-02   0.099
## educationprofessional.course                    9.834e-03  9.309e-02   0.106
## educationuniversity.degree                      6.937e-02  8.374e-02   0.828
## educationilliterate                             9.987e-01  6.418e-01   1.556
## educationunknown                                1.086e-01  1.115e-01   0.974
```

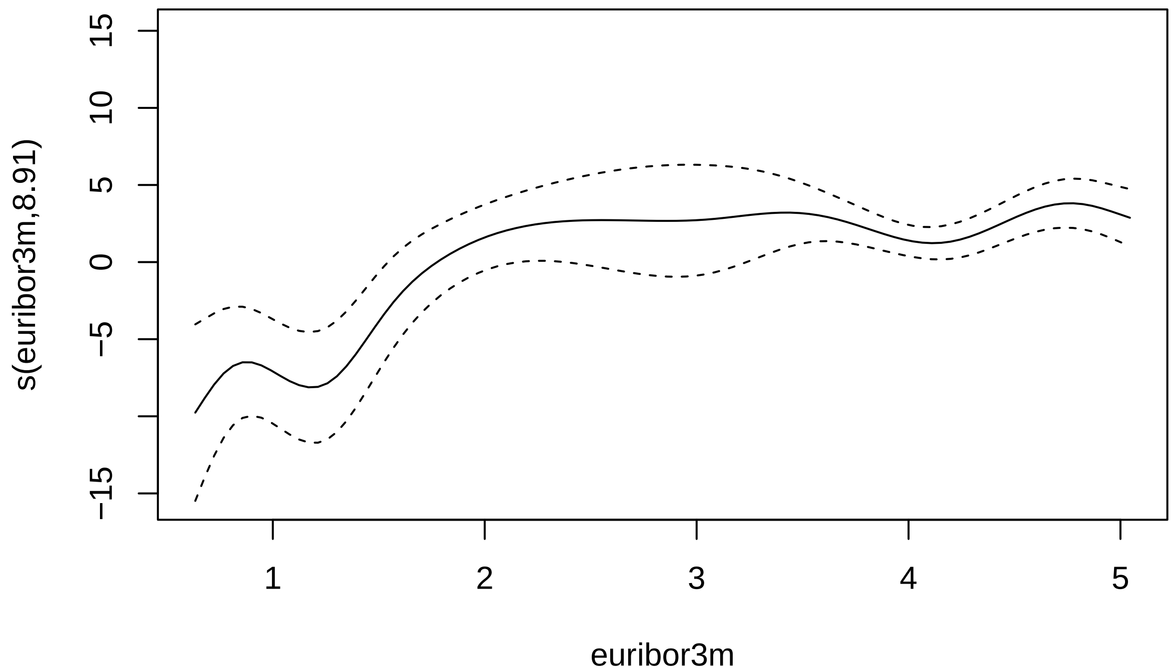
```
## contacttelephone                              -5.284e-01  7.595e-02  -6.957
## monthapr                                      -1.490e+00  2.853e-01  -5.222
## monthmay                                      -3.011e+00  3.125e-01  -9.635
## monthjun                                      -3.644e+00  4.455e-01  -8.181
## monthjul                                      -3.712e+00  4.504e-01  -8.242
## monthaug                                      -3.122e+00  6.010e-01  -5.195
## monthsep                                      -2.084e+00  5.071e-01  -4.109
## monthoct                                      -2.858e+00  6.351e-01  -4.500
## monthnov                                      -3.339e+00  6.256e-01  -5.337
## monthdec                                      -3.027e+00  6.100e-01  -4.962
## day_of_weektue                                 2.571e-01  5.990e-02   4.292
## day_of_weekwed                                 3.513e-01  6.008e-02   5.848
## day_of_weekthu                                 2.198e-01  5.952e-02   3.692
## day_of_weekfri                                 2.335e-01  6.225e-02   3.752
## emp.var.rate                                  -1.769e+00  3.504e-01  -5.048
## cons.price.idx                                 8.289e-01  5.136e-01   1.614
## cons.conf.idx                                 -2.079e-04  3.698e-02  -0.006
## nr.employed                                    2.375e-02  8.907e-03   2.666
## age_group30-57                                -9.577e-02  5.663e-02  -1.691
## age_group58+                                   1.160e-01  1.044e-01   1.111
## default_groupunknown_or_yes                   -1.864e-01  5.812e-02  -3.206
## strategy_period2                               2.253e+00  7.031e-01   3.204
## strategy_period3                               2.092e+00  7.132e-01   2.934
## contacted1                                     1.488e+00  2.840e-01   5.241
## loan_housing_statusboth_no                     1.290e-01  1.286e-01   1.003
## loan_housing_statusloan_no_housing_yes         9.934e-02  1.281e-01   0.775
## loan_housing_statusloan_yes_housing_no         1.248e-01  1.462e-01   0.854
## loan_housing_statusboth_yes                    5.329e-02  1.398e-01   0.381
## poutcome_previouspoutcome_failure_previous_1  -4.013e-01  6.350e-02  -6.319
## poutcome_previouspoutcome_success_previous_1  -3.195e-01  2.975e-01  -1.074
## poutcome_previouspoutcome_success_previous_2  -1.539e-01  3.271e-01  -0.471
## poutcome_previouspoutcome_failure_previous_2  -7.765e-01  1.755e-01  -4.424
## poutcome_previouspoutcome_success_previous_3  -5.575e-01  4.083e-01  -1.366
## poutcome_previouspoutcome_failure_previous_3  -8.174e-01  3.435e-01  -2.380
## poutcome_previouspoutcome_failure_previous_4  -2.051e+00  8.545e-01  -2.400
## poutcome_previouspoutcome_success_previous_4   8.011e-01  8.160e-01   0.982
## poutcome_previouspoutcome_success_previous_5+  3.418e-01  1.141e+00   0.300
## campaign_group2                               -1.188e-02  4.564e-02  -0.260
## campaign_group3                                6.727e-02  5.939e-02   1.133
## campaign_group4                                3.652e-02  8.106e-02   0.451
## campaign_group5                               -1.843e-01  1.095e-01  -1.683
## campaign_group6                               -1.381e-01  1.385e-01  -0.997
## campaign_group7                               -2.930e-01  1.872e-01  -1.565
## campaign_group8                               -4.784e-01  2.562e-01  -1.867
## campaign_group9+                              -3.620e-01  1.462e-01  -2.476
##                                               Pr(>|z|)
## (Intercept)                                   0.001969 **
## jobadmin.                                     0.951276
## jobblue-collar                                0.350607
## jobentrepreneur                               0.819358
## jobhousemaid                                  0.610904
## jobmanagement                                 0.803302
## jobretired                                    0.411568
## jobself-employed                              0.964293
```

```
## jobservices                                 0.562227
## jobstudent                                  0.268726
## jobtechnician                               0.861840
## jobunknown                                  0.265387
## maritalmarried                              0.789061
## maritalsingle                               0.236701
## maritalunknown                              0.204513
## educationbasic.6y                           0.205016
## educationbasic.9y                           0.881654
## educationhigh.school                        0.920973
## educationprofessional.course                0.915863
## educationuniversity.degree                  0.407417
## educationilliterate                         0.119667
## educationunknown                            0.329885
## contacttelephone                            3.48e-12 ***
## monthapr                                    1.77e-07 ***
## monthmay                                     < 2e-16 ***
## monthjun                                    2.81e-16 ***
## monthjul                                     < 2e-16 ***
## monthaug                                    2.04e-07 ***
## monthsep                                    3.97e-05 ***
## monthoct                                    6.78e-06 ***
## monthnov                                    9.45e-08 ***
## monthdec                                    6.99e-07 ***
## day_of_weektue                              1.77e-05 ***
## day_of_weekwed                              4.98e-09 ***
## day_of_weekthu                              0.000222 ***
## day_of_weekfri                              0.000175 ***
## emp.var.rate                                4.46e-07 ***
## cons.price.idx                              0.106567
## cons.conf.idx                               0.995515
## nr.employed                                 0.007669 **
## age_group30-57                              0.090792 .
## age_group58+                                0.266586
## default_groupunknown_or_yes                 0.001345 **
## strategy_period2                            0.001355 **
## strategy_period3                            0.003350 **
## contacted1                                  1.60e-07 ***
## loan_housing_statusboth_no                  0.315922
## loan_housing_statusloan_no_housing_yes      0.438074
## loan_housing_statusloan_yes_housing_no      0.392995
## loan_housing_statusboth_yes                 0.703044
## poutcome_previouspoutcome_failure_previous_1  2.64e-10 ***
## poutcome_previouspoutcome_success_previous_1  0.282762
## poutcome_previouspoutcome_success_previous_2  0.637970
## poutcome_previouspoutcome_failure_previous_2  9.68e-06 ***
## poutcome_previouspoutcome_success_previous_3  0.172092
## poutcome_previouspoutcome_failure_previous_3  0.017330 *
## poutcome_previouspoutcome_failure_previous_4  0.016402 *
## poutcome_previouspoutcome_success_previous_4  0.326199
## poutcome_previouspoutcome_success_previous_5+ 0.764467
## campaign_group2                             0.794619
## campaign_group3                             0.257309
## campaign_group4                             0.652336
```

```
## campaign_group5                                  0.092288 .
## campaign_group6                                  0.318902
## campaign_group7                                  0.117532
## campaign_group8                                  0.061885 .
## campaign_group9+                                 0.013274 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                 edf Ref.df Chi.sq p-value
## s(day_id)     8.859  8.993 132.75  <2e-16 ***
## s(euribor3m) 8.910  8.996  76.19  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.194   Deviance explained =   20%
## UBRE = -0.47087  Scale est. = 1         n = 40040
```

```
plot(model_4)
```





Modelling those economic variables as linear makes month significant again. This suggests that there are some interactions between these economic variables that capture some temporal variance, which makes sense as these economic variables are likely in-part capturing the business cycle. Unfortunately fitting interaction terms is too computationally complex, so I keep the non-linear model.

```
# Choose non-linear model as final model
model_final <- model_3
```

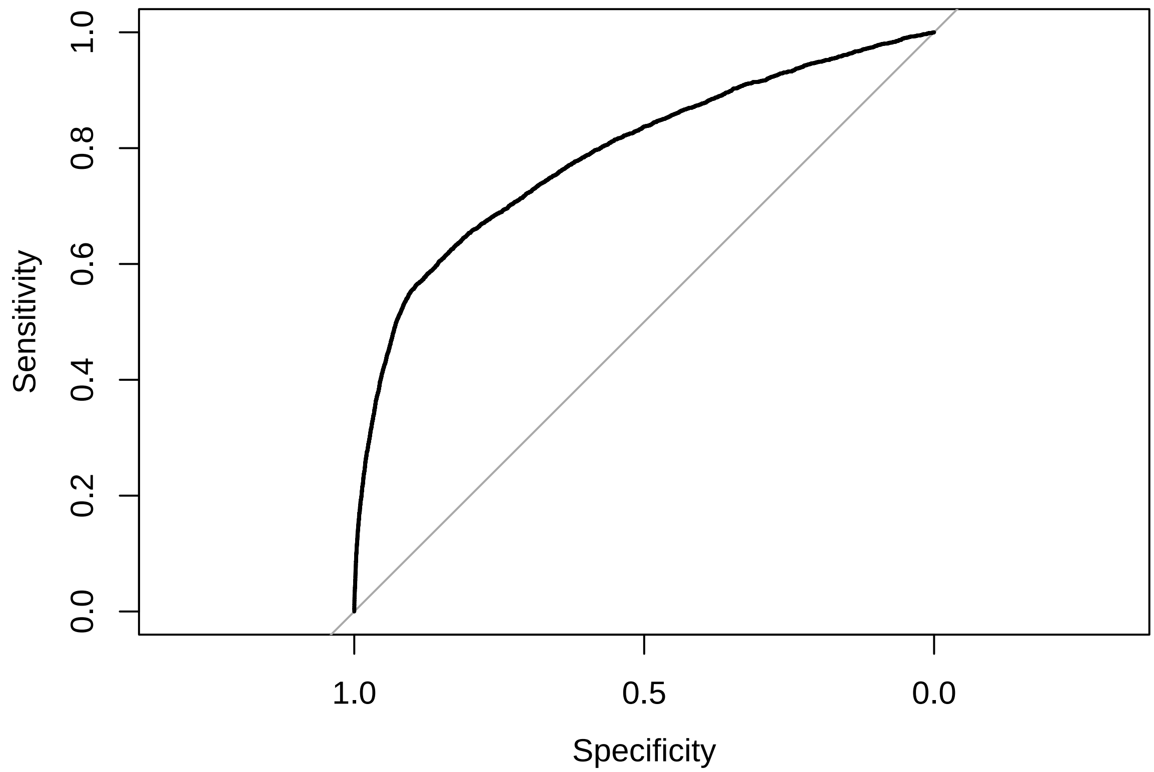
I plot an ROC curve to determine a good threshold for classification.

```
# Plot ROC curve
roc_obj <- roc(training_data$y, predict(model_final, training_data, type = "response"))
```

```
## Setting levels: control = no, case = yes
```

```
## Setting direction: controls < cases
```

```
plot(roc_obj)
```




```r
# Determine threshold
best_coords <- coords(roc_obj, "best")
best_threshold <- best_coords["threshold"][1,1]
```

I calculate the F1 score and plot a confusion matrix to evaluate the model.

```r
# Calculate F1 score
f1_score <- function(model, data) {
  # Predict on data
  predictions <- predict(model, data, type = "response")

  # Convert to binary
  predictions <- ifelse(predictions > best_threshold, "yes", "no")

  # Calculate confusion matrix
  confusion_matrix <- table(data$y, predictions)

  # Calculate precision and recall
  precision <- confusion_matrix["yes", "yes"] / sum(confusion_matrix["yes", ])
  recall <- confusion_matrix["yes", "yes"] / sum(confusion_matrix[, "yes"])

  # Calculate F1 score
  f1 <- 2 * (precision * recall) / (precision + recall)
  return(f1)
}
```

```r
# Calculate F1 score for final model
f1_score(model_final, training_data)
```

```
## [1] 0.4167659
```

```r
# Plot confusion matrix
table(training_data$y, predict(model_final, training_data, type = "response") > best_threshold)
```

```
##
##       FALSE  TRUE
##   no  30729  5255
##   yes  1605  2451
```

The confusion matrix shows that the model predicts "yes" too often when it should be predicting "no".

Significant linear predictors are: -contact -day_of_week -default_group -contacted -poutcome_previous -campaign_group

Insignificant linear predictors are: -job -education -age_group -loan_housing_status

Significant non-linear predictors are: -day_id -euribor3m

Insignificant non-linear predictors are: -emp.var.rate -cons.price.idx -cons.conf.idx -nr.employed