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("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 ...
```

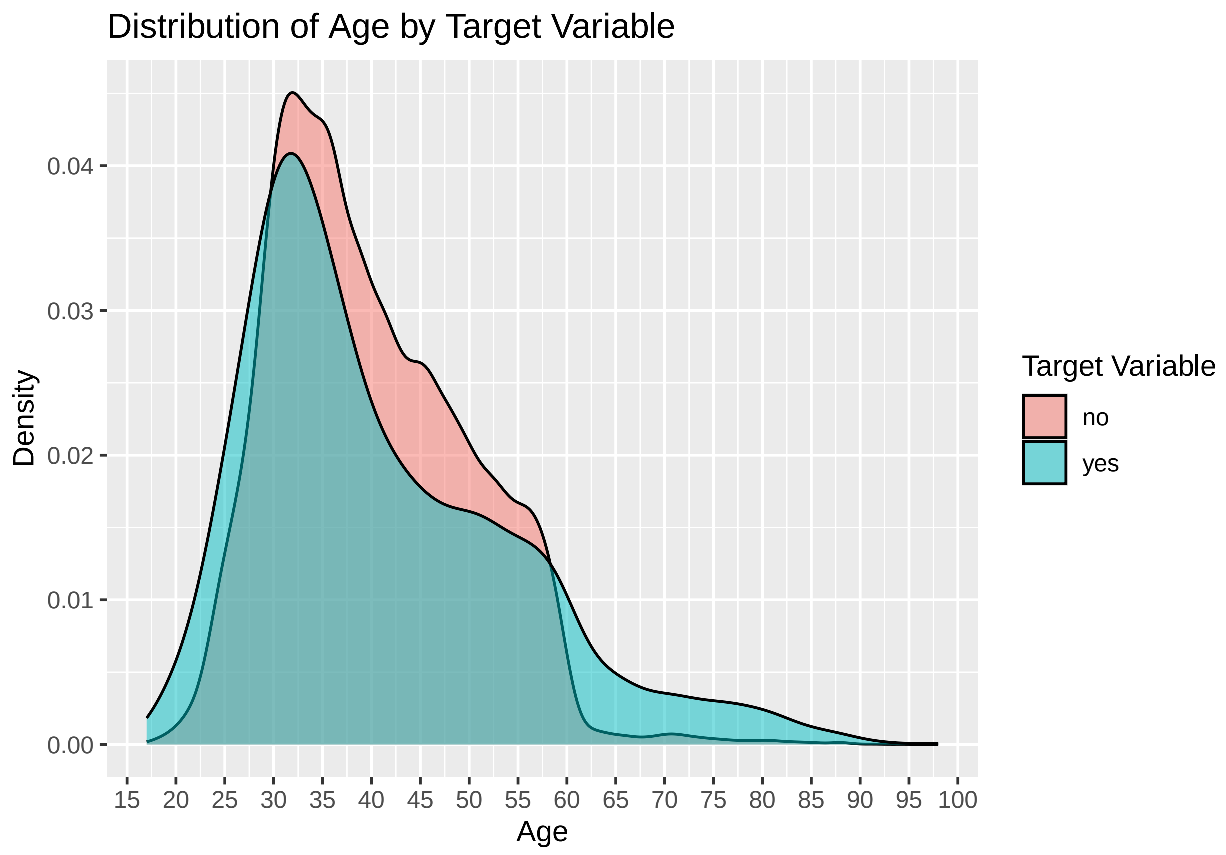
Exploratory Data Analysis First, we will look at the distribution of the target variable.

```
summary(bank_data$y)
```

```
##    no   yes
## 36548  4640
```

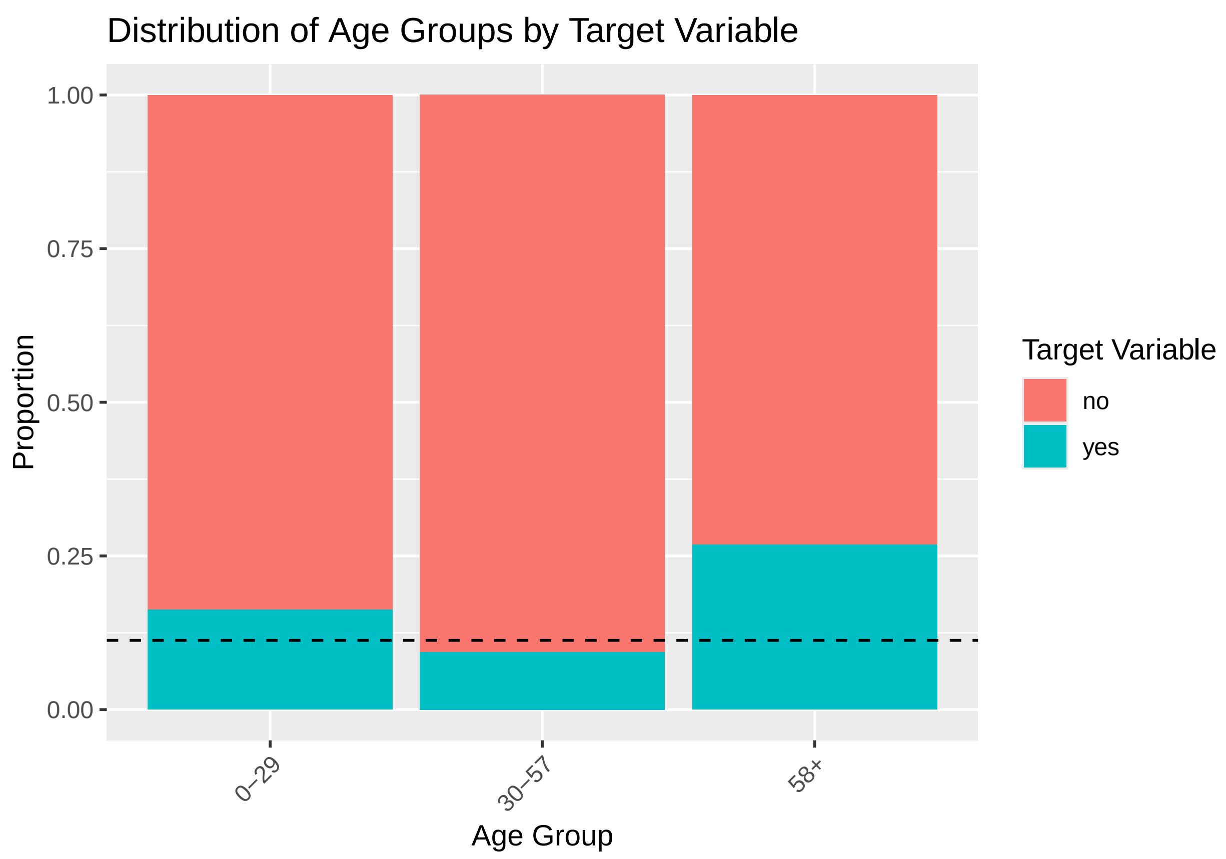
Our target variable has a class imbalance in favour of the “no” class. Next, we look at how all our features vary with the target variable.

```
ggplot(bank_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")
```



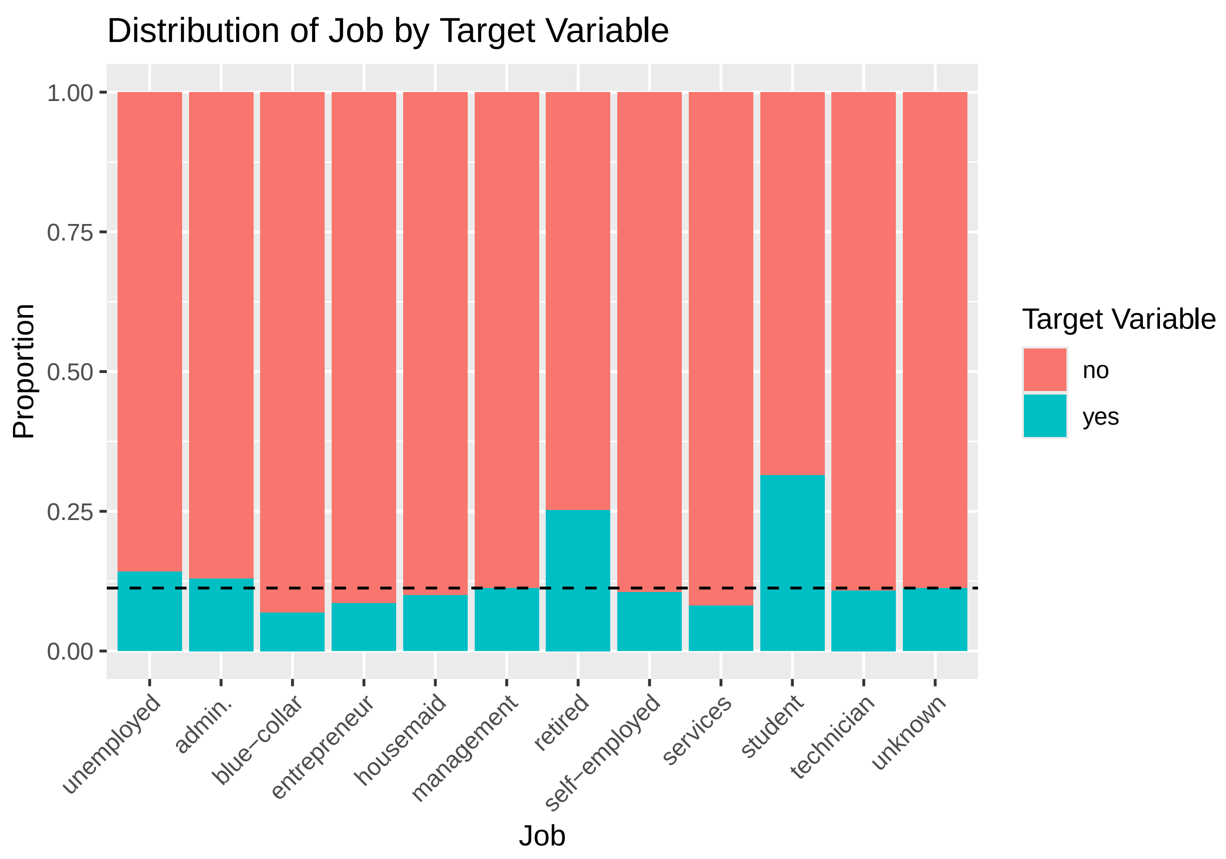
The distribution of age is different between classes. The “yes” class has a higher density between 0-29 and 58+, and the “no” class has a higher density between 30-57. I create a new factor variable by binning ages into these three categories. I use 0-29 as the reference category as it is closest to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

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

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

# Show distribution of age groups with a line of mean proportion and rotated x-axis labels
ggplot(bank_data, aes(x = age_group, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(bank_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.

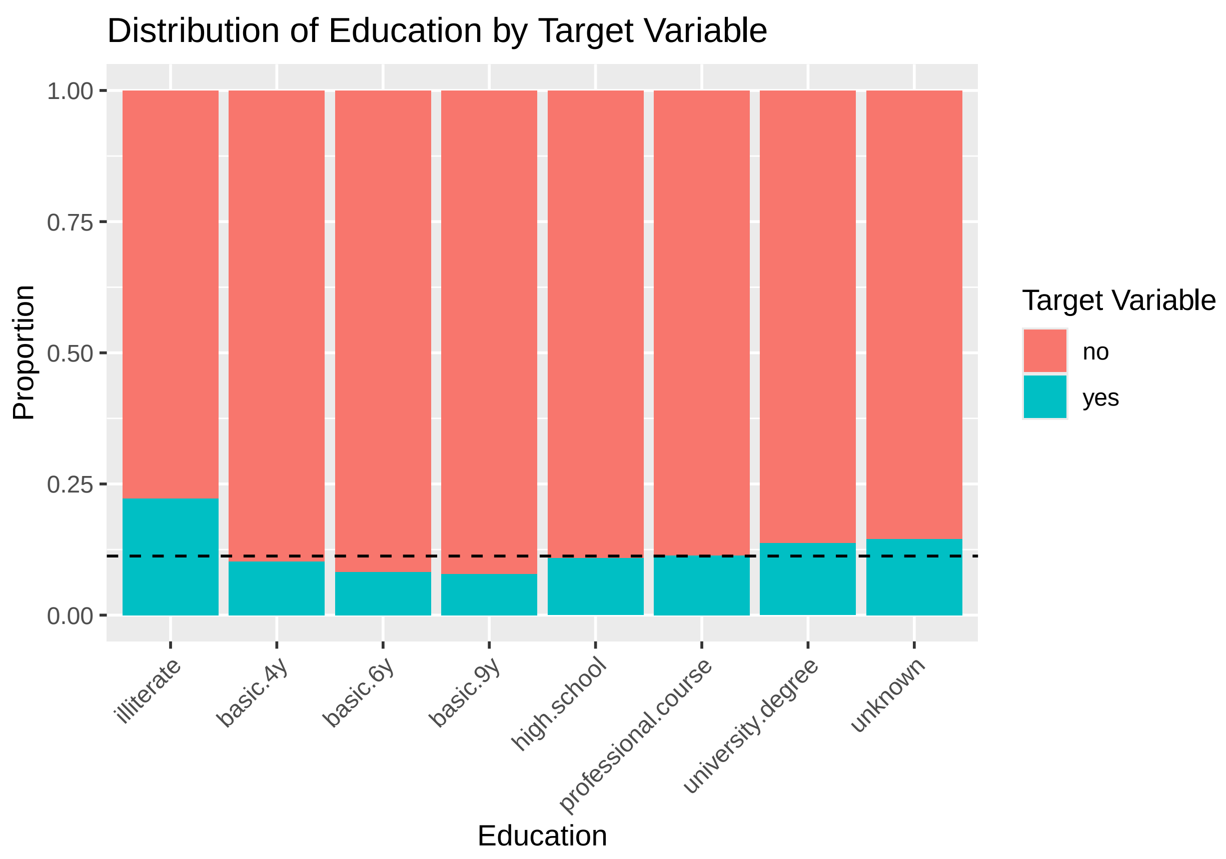
```r
# Set "unemployed" as the reference category
bank_data$job <- relevel(bank_data$job, ref = "unemployed")

# Show distribution of job
ggplot(bank_data, aes(x = job, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(bank_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).

```r
# Order education levels
bank_data$education <- factor(bank_data$education, levels = c("illiterate", "basic.4y", "basic.6y", "ba

# Show distribution of education with a line of mean proportion and rotated x-axis labels
ggplot(bank_data, aes(x = education, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(bank_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")
```

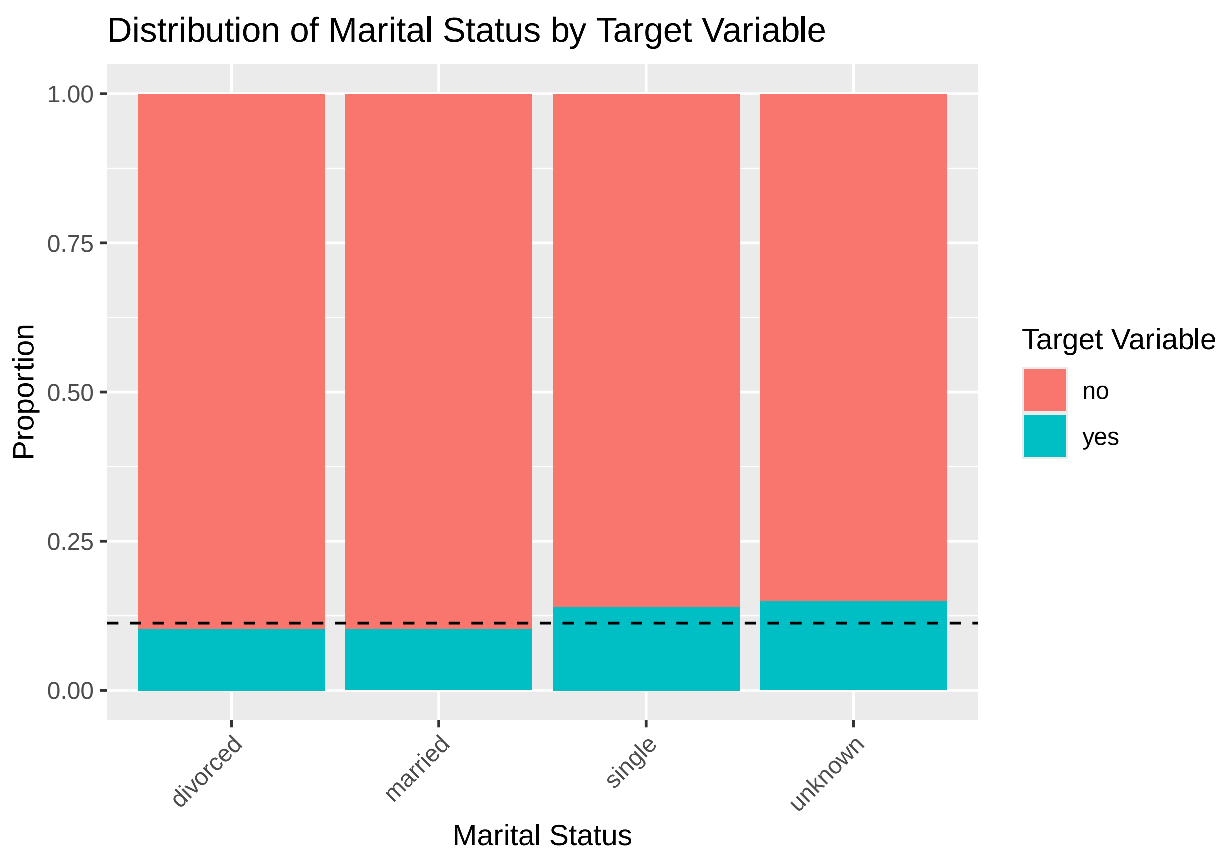


```r
# Set "illiterate" as the reference category
bank_data$education <- relevel(bank_data$education, ref = "illiterate")

summary(bank_data$education)
```

```
##          illiterate            basic.4y            basic.6y            basic.9y
##                  18                4176                2292                6045
##         high.school professional.course   university.degree             unknown
##                9515                5243               12168                1731
```

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

```r
# Show distribution of marital status
ggplot(bank_data, aes(x = marital, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(bank_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.

```r
summary(bank_data$default)
```

```
##      no unknown     yes
##   32588    8597       3
```

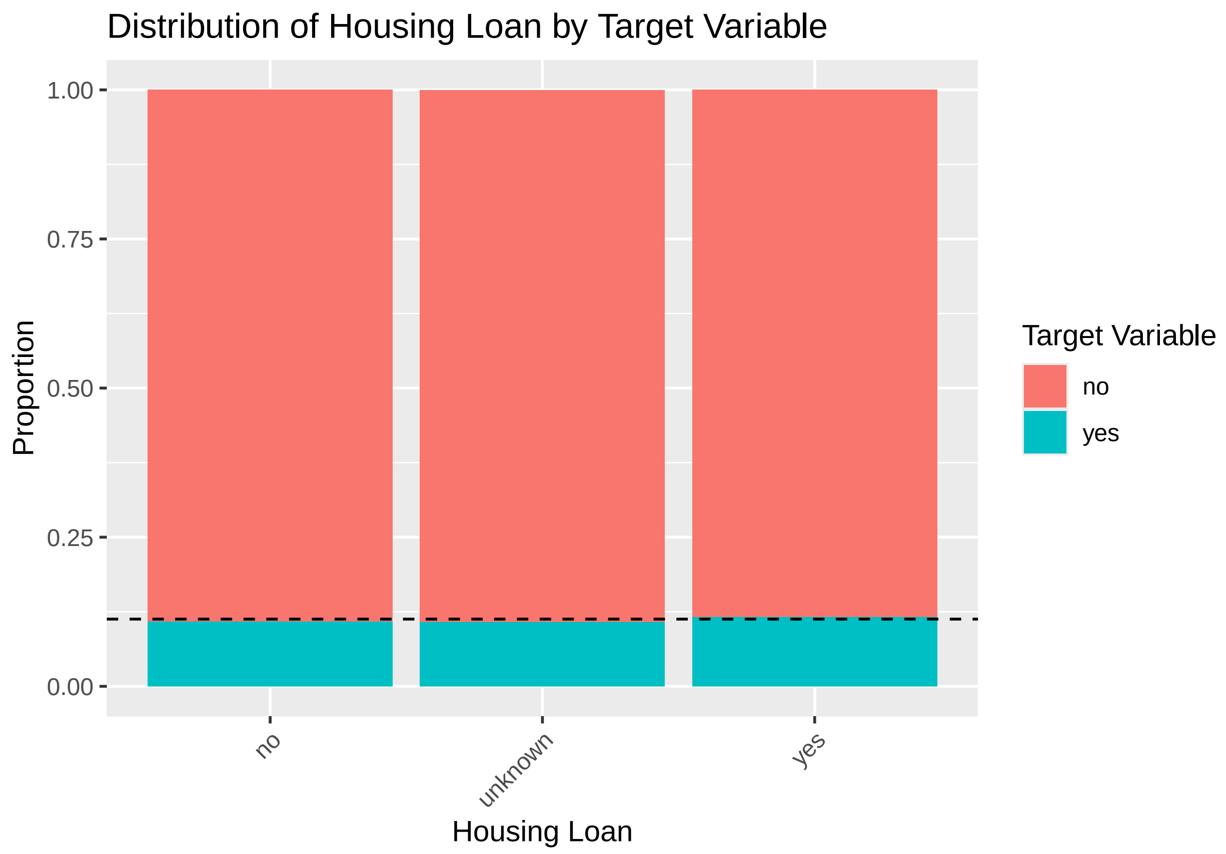
```r
missing_default <- bank_data %>%
  filter(default == "unknown")

nrow(missing_default) / nrow(bank_data)
```

```
## [1] 0.2087258
```

This feature is useless - there are extremely few "yes" and 21% of data on this feature is missing - so we will drop it.

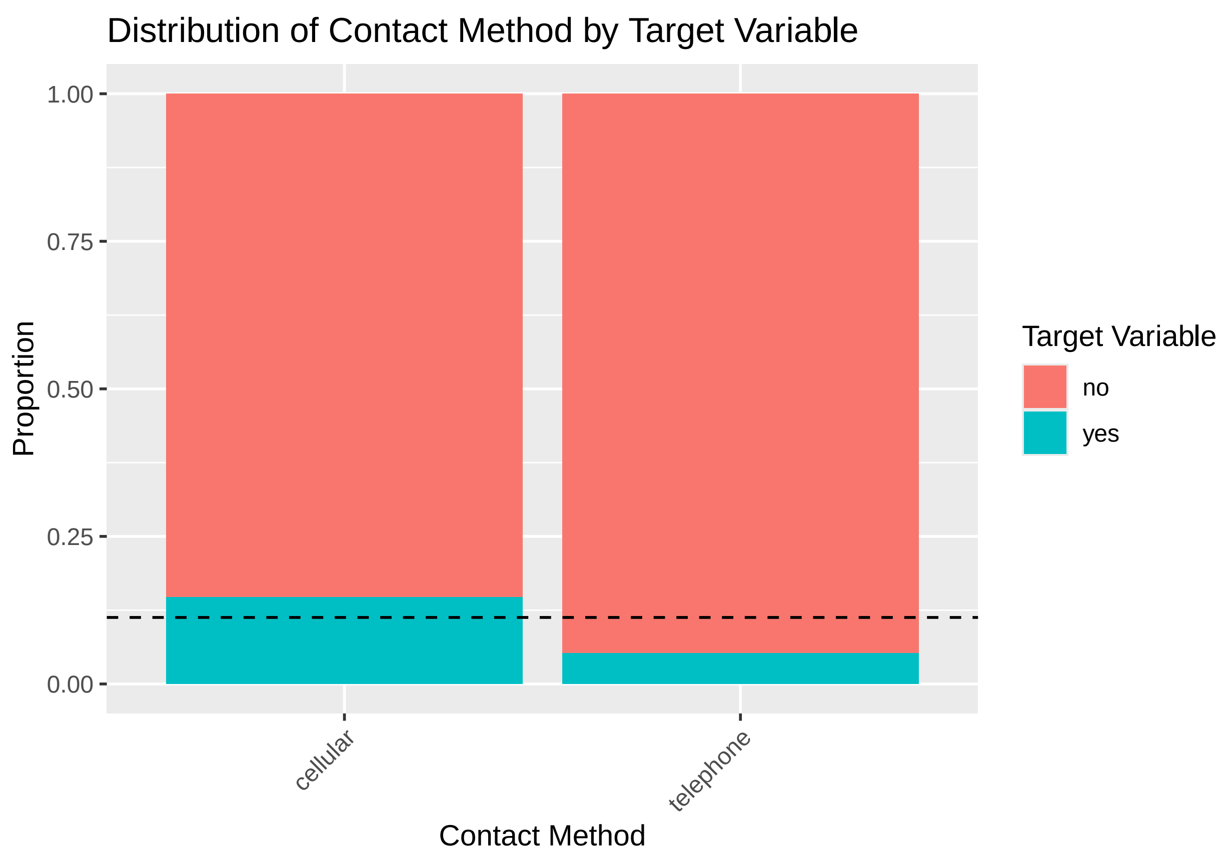
```r
# Drop default
bank_data <- bank_data %>%
  select(-default)
```

```r
# Show distribution of housing loan
ggplot(bank_data, aes(x = housing, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(bank_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.

```r
# Show distribution of contact method
ggplot(bank_data, aes(x = contact, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(bank_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")
```



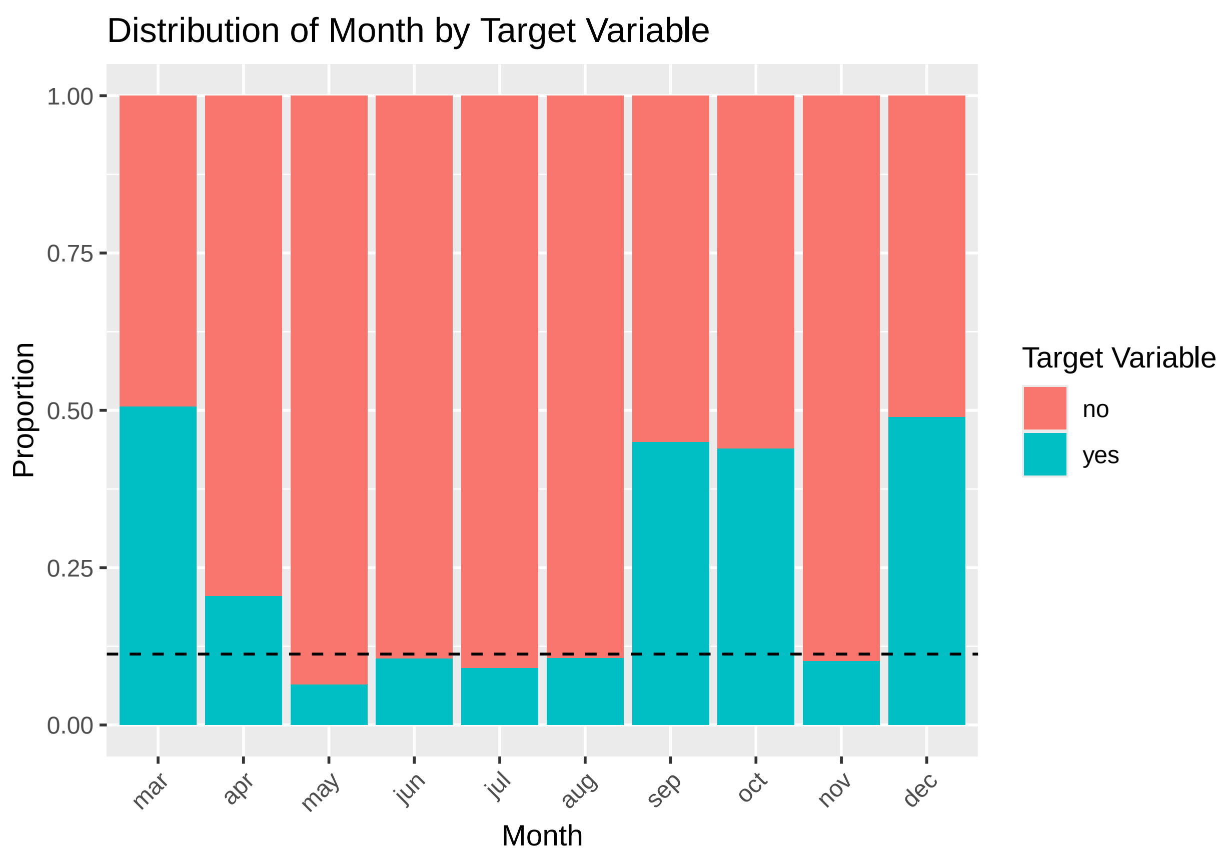
Cellular has a higher proportion of "yes" and telephone has a much higher proportion of "no".

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

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

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

```
##
##   jan   feb   mar   apr   may   jun   jul   aug   sep   oct   nov   dec
##     0     0   546  2632 13769  5318  7174  6178   570   718  4101   182
```

```
# Show distribution of month and frequency of each month
ggplot(bank_data, aes(x = month, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(bank_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.

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.

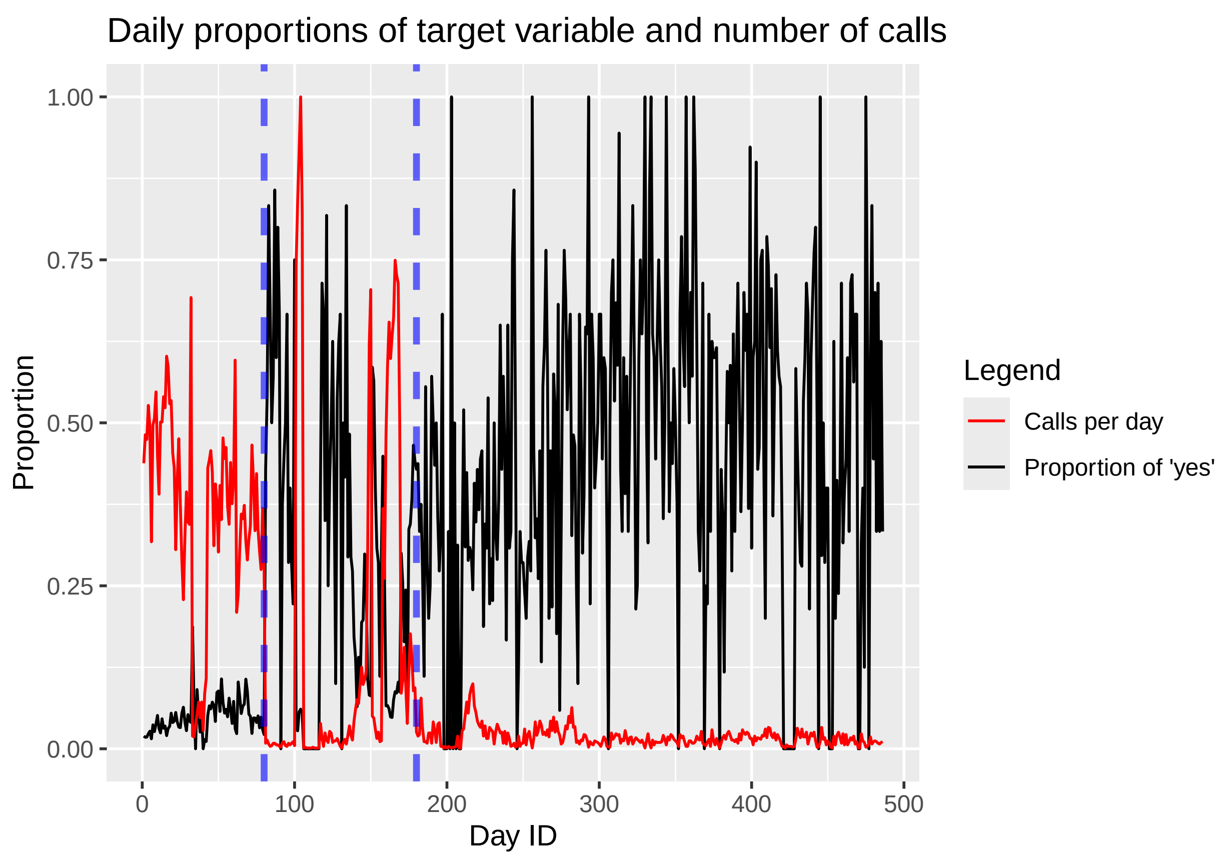
```
bank_data <- bank_data %>%
  # Mark the first row of each new day
  mutate(new_day = if_else(row_number() == 1, TRUE, day_of_week != lag(day_of_week))) %>%

  # 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))) %>%
  mutate(month_id = cumsum(new_month))

# Remove helper columns
bank_data <- bank_data %>%
```

```r
  select(-new_day, -new_month)

# Group day, week and month labels together
bank_data <- bank_data %>%
  mutate(date_label = paste(day_id, week_id, month_id, sep = "-"))
```

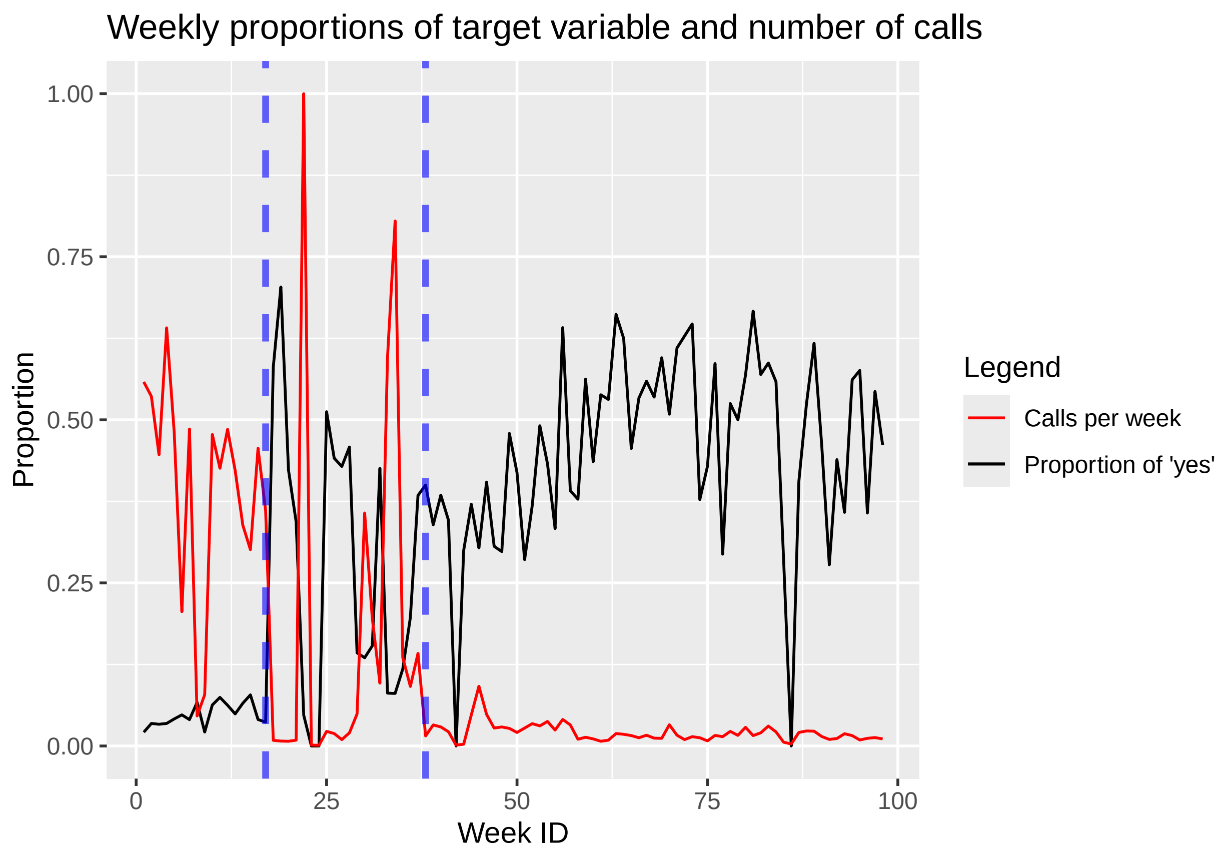
Now we can see the distribution of the target variable and frequency of call by day, week, and month.

```r
# Calculate number of calls per day
calls_per_day <- bank_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(bank_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.
```

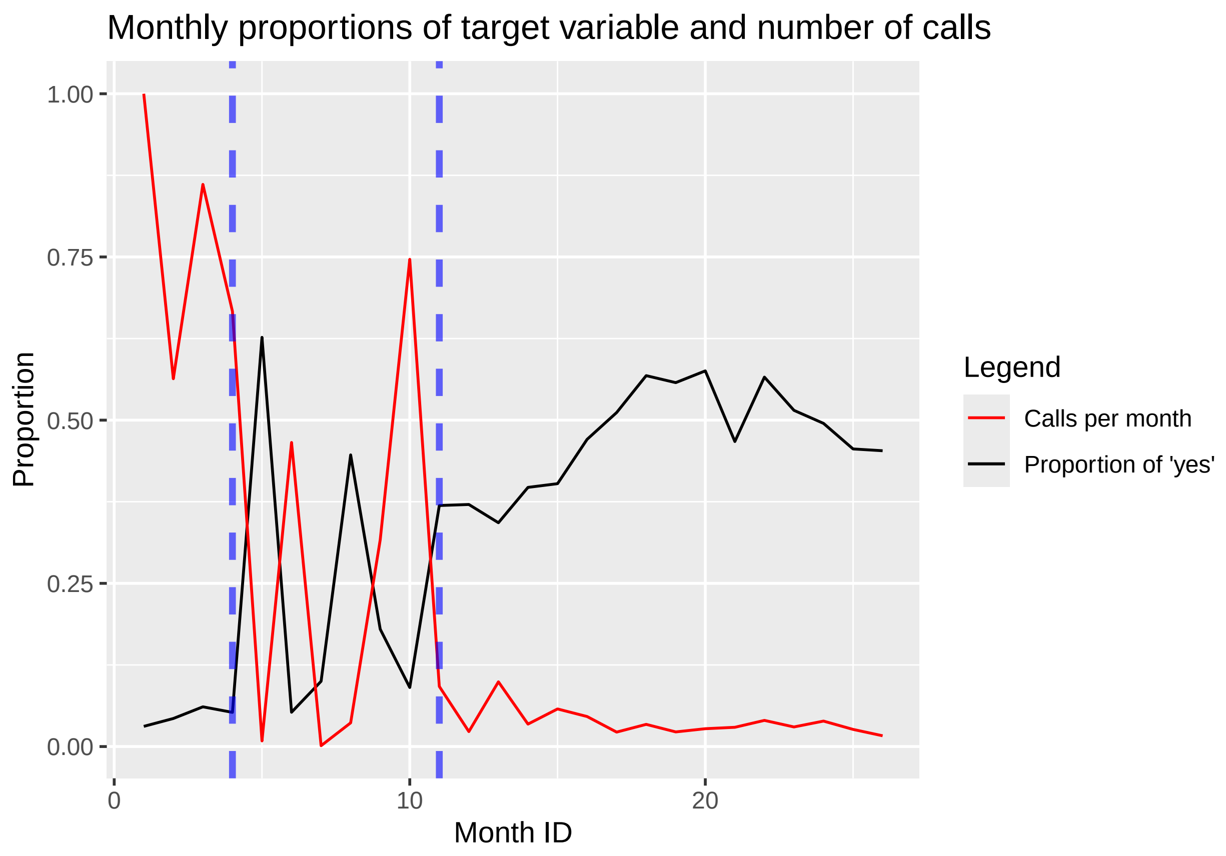


```
# Calculate number of calls per week
calls_per_week <- bank_data %>% count(week_id, name = "n_calls")
benchmark_week_calls <- max(calls_per_week$n_calls)

# 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(bank_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 <- bank_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(bank_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")) +

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.

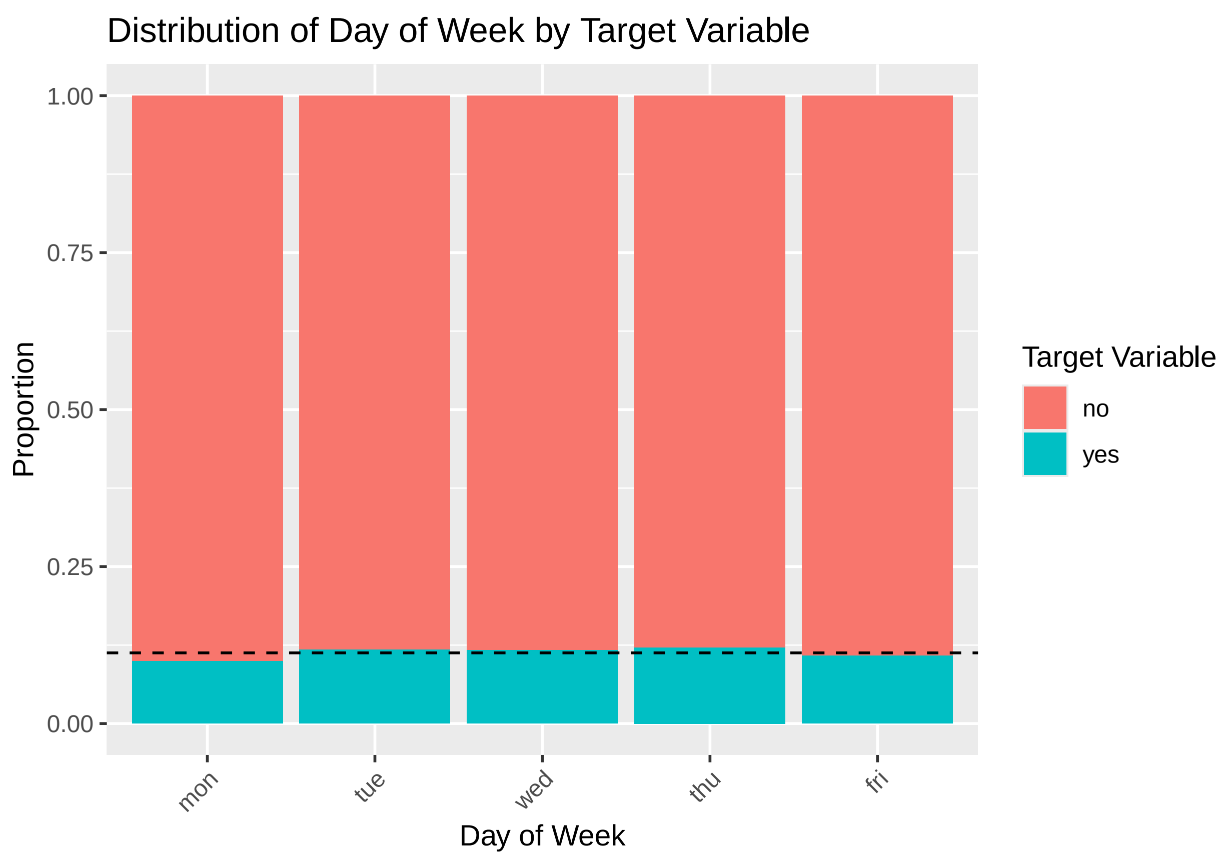
Any seasonality in this time period is completely dominated by effects from these strategy shifts, so month of year is not useful for now but measuring seasonality may be useful in future datasets where strategy shifts do not occur.

```r
# Remove month of year
bank_data <- bank_data %>%
  select(-month)
```

We can add lags of these proportions aggregated by month to the data to measure the effects of changes in strategy.

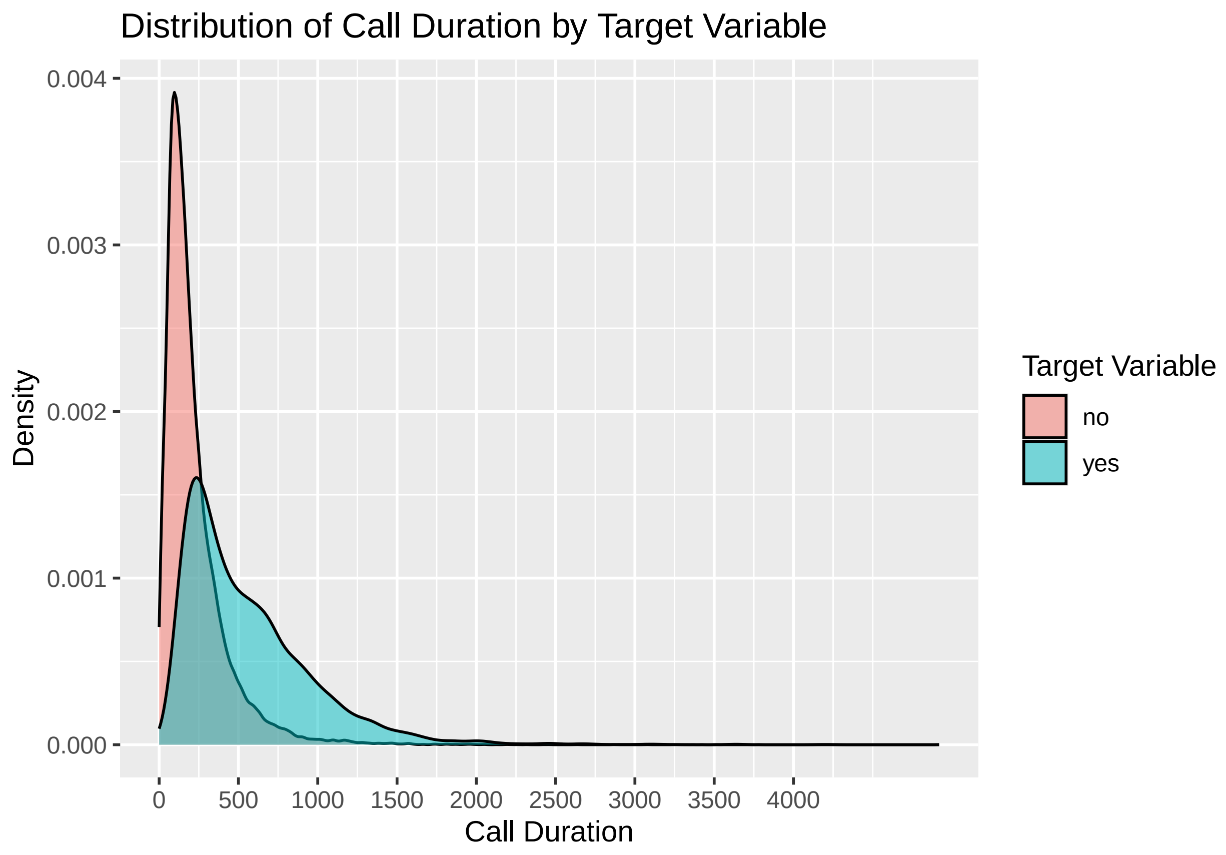
```r
# Compute proportion of "yes" per month
proportion_yes <- bank_data %>%
  group_by(month_id) %>%
  summarise(proportion_yes = mean(y == "yes", na.rm = TRUE)) %>%
  ungroup()

# Define number of lags
n_lags <- 3

# Generated lagged columns dynamically, and fill missing values with 0
for (i in 1:n_lags) {
  proportion_yes <- proportion_yes %>%
    mutate(!!paste0("lag_proportion_yes_month_", i) := coalesce(lag(proportion_yes, i), 0))
}

# Merge lagged columns back into the data
bank_data <- bank_data %>%
  left_join(proportion_yes, by = "month_id")
```

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

# Show distribution of day of week
ggplot(bank_data, aes(x = day_of_week, fill = y)) +
  geom_bar(position = "fill") +
  geom_hline(yintercept = mean(bank_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(bank_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.