Predicting Bank Telemarketing Success

Group 5

2025-03-27

In the bank industry, optimizing telemarketing campaign strategies has become crucial, particularly in the aftermath of the 2008 global financial crisis. Banks are seeking ways to improve their telemarketing campaigns to increase their profit margins. We want to see if certain attributes about the client and the direct telemarketing calls made may predict whether a client subscribes to a term deposit.

Using telemarketing campaign data from a Portuguese banking institution from 2008 to 2013 (Obtained from Kaggle), we proposed a logistic regression model that can predict whether a client subscribes to a term deposit or not based on several variables given.

Load in Libraries

```
library(ltm)
library(corrplot)
library(GGally)
library(ggplot2)
library(dplyr)
library(rsample)
library(caret)
library(car)
library(gridExtra)
library(sjPlot)
```

Load in Data

```
bank <- read.csv("bank-additional-full.csv", header = TRUE, sep = ";")
```

First Glimpse

```
str(bank)
```

```
## '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" ...
```

```
sapply(bank, anyNA)
```

```
##            age            job        marital      education        default
##          FALSE          FALSE          FALSE          FALSE          FALSE
##        housing           loan        contact          month    day_of_week
##          FALSE          FALSE          FALSE          FALSE          FALSE
##       duration       campaign          pdays       previous       poutcome
##          FALSE          FALSE          FALSE          FALSE          FALSE
##   emp.var.rate cons.price.idx  cons.conf.idx      euribor3m    nr.employed
##          FALSE          FALSE          FALSE          FALSE          FALSE
##              y
##          FALSE
```

Data Cleaning

```
# Recode outcome as 0,1

# Recode Education (basics into 1 category)

bank <- bank %>% mutate(
  y = case_when(
    y == "no" ~ 0,
    y == "yes" ~ 1,
    TRUE ~ NA_real_
  ),
  education = case_when(
    education %in% c("basic.4y", "basic.6y", "basic.9y", "illiterate") ~ "basic.education",
    education == "unknown" ~ NA_character_,
    TRUE ~ education
  ),
  job = case_when(
```

```
    job %in% c("blue-collar", "housemaid") ~ "blue-collar",
    job %in% c("admin.", "management") ~ "admin/mgmt",
    job %in% c("entrepreneur","self-employed") ~ "self-employed",
    job %in% c("student", "retired", "unemployed") ~ "no income",
    job == "unknown" ~ NA_character_,
    TRUE ~ job
  )
)

# Create a new variable `nr.employed.cat` with grouped levels

bank$nr.employed.cat <- cut(
  bank$nr.employed,
  breaks = c(-Inf, 5050, 5150, 5200, Inf),  # Define breakpoints for the groups
  labels = c("Very Low", "Low", "High", "Very High"),  # Assign group labels
  right = TRUE  # Include the upper bound of each interval
)

# Check the distribution of the new variable

table(bank$nr.employed.cat)
```

```
##
##  Very Low       Low      High Very High
##      3301     10197     11456     16234
```

```
# Use & to filter rows where neither education nor job is missing

bank <- bank %>%
  filter(!is.na(education) & !is.na(job))

# Note total n is now 39,258
```

Categorical EDA

Contingency Tables

```
table(Subscription = bank$y)
```

```
## Subscription
##     0     1
## 34889  4369
```

```
table(Education = bank$education)
```

```
## Education
##     basic.education         high.school professional.course   university.degree
##               12426                9478                5231               12123
```

```r
table(Job = bank$job)
```

```
## Job
##    admin/mgmt   blue-collar     no income self-employed      services
##         12974          9818          3325          2791          3819
##    technician
##          6531
```

```r
table(Marital = bank$marital)
```

```
## Marital
## divorced  married   single  unknown
##     4417    23748    11026       67
```

```r
table(Previous_outcome = bank$poutcome)
```

```
## Previous_outcome
##     failure nonexistent     success
##        4057       33926        1275
```

```r
table(Default = bank$default)
```

```
## Default
##      no unknown     yes
##   31284    7971       3
```

```r
table(Loan = bank$loan)
```

```
## Loan
##      no unknown     yes
##   32344     947    5967
```

```r
table(Contact = bank$contact)
```

```
## Contact
##  cellular telephone
##     25030     14228
```

```r
table(Housing = bank$housing)
```

```
## Housing
##      no unknown     yes
##   17706     947   20605
```

```r
bank$month <- factor(bank$month, levels = c("jan", "feb", "mar", "apr", "may", "jun",
                                            "jul", "aug", "sep", "oct", "nov", "dec"))
table(Month = bank$month)
```

```
## Month
##   jan   feb   mar   apr   may   jun   jul   aug   sep   oct   nov   dec
##     0     0   511  2496 13156  5023  6771  5957   516   681  3980   167
```

Remove observations if they have a small amount of "unknown" if they used as RELEVANT predictors for the model development. At this point, I am going to keep these observations.

Chi-square Test Results

```
cat_vars <- c("job", "marital", "education", "default", "housing", "loan", "contact",
              "month", "day_of_week", "poutcome")
results <- data.frame(variable = character(), p_value = numeric(), chi_squared = numeric())

for (var in cat_vars) {
  cont_table <- table(bank[[var]], bank$y)
  chisq_result <- chisq.test(cont_table)
  results <- rbind(results, data.frame(variable = var, p_value = chisq_result$p.value,
                                       chi_squared = chisq_result$statistic))
}
```

```
## Warning in chisq.test(cont_table): Chi-squared approximation may be incorrect
## Warning in chisq.test(cont_table): Chi-squared approximation may be incorrect
```

```
results_sorted <- results[order(results$p_value),]
for (i in 1:nrow(results_sorted)) {
  var <- results_sorted$variable[i]
  p_val <- results_sorted$p_value[i]
  chi_sq <- results_sorted$chi_squared[i]
  cat("Chi-Square Test for", var, "\n")
  cat("Chi-Squared Value:", chi_sq, "\n")
  cat("p-value:", p_val, "\n")
  cat("-----------\n")
}
```

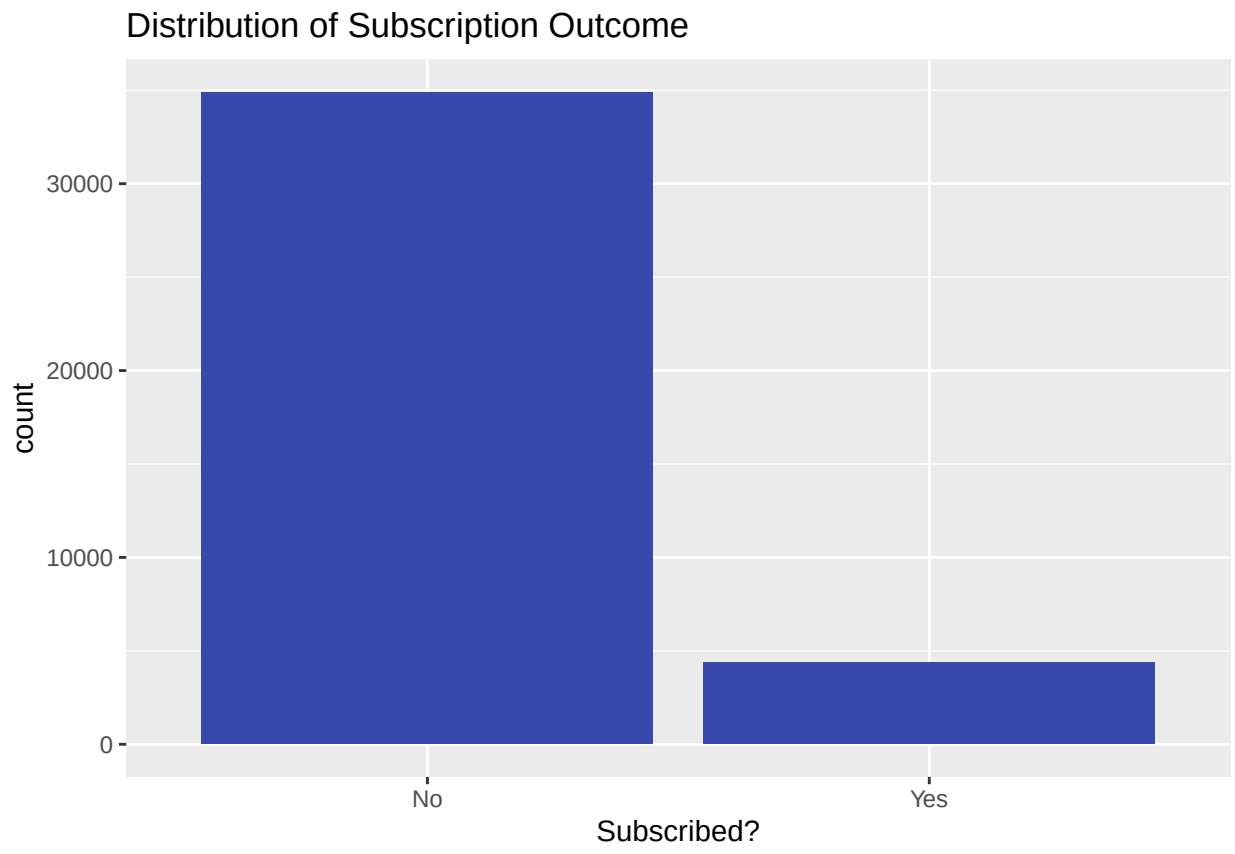
```
## Chi-Square Test for poutcome
## Chi-Squared Value: 3922.836
## p-value: 0
## -----------
## Chi-Square Test for contact
## Chi-Squared Value: 797.5792
## p-value: 1.812937e-175
## -----------
## Chi-Square Test for job
## Chi-Squared Value: 670.9772
## p-value: 9.246406e-143
## -----------
## Chi-Square Test for default
## Chi-Squared Value: 362.7572
## p-value: 1.691523e-79
## -----------
## Chi-Square Test for education
## Chi-Squared Value: 152.2818
## p-value: 8.482357e-33
## -----------
## Chi-Square Test for marital
## Chi-Squared Value: 98.69596
## p-value: 2.963949e-21
## -----------
## Chi-Square Test for day_of_week
```

```
## Chi-Squared Value: 24.06753
## p-value: 7.742347e-05
## -----------
## Chi-Square Test for housing
## Chi-Squared Value: 3.828941
## p-value: 0.1474199
## -----------
## Chi-Square Test for loan
## Chi-Squared Value: 1.440346
## p-value: 0.4866681
## -----------
## Chi-Square Test for month
## Chi-Squared Value: NaN
## p-value: NaN
## -----------
```

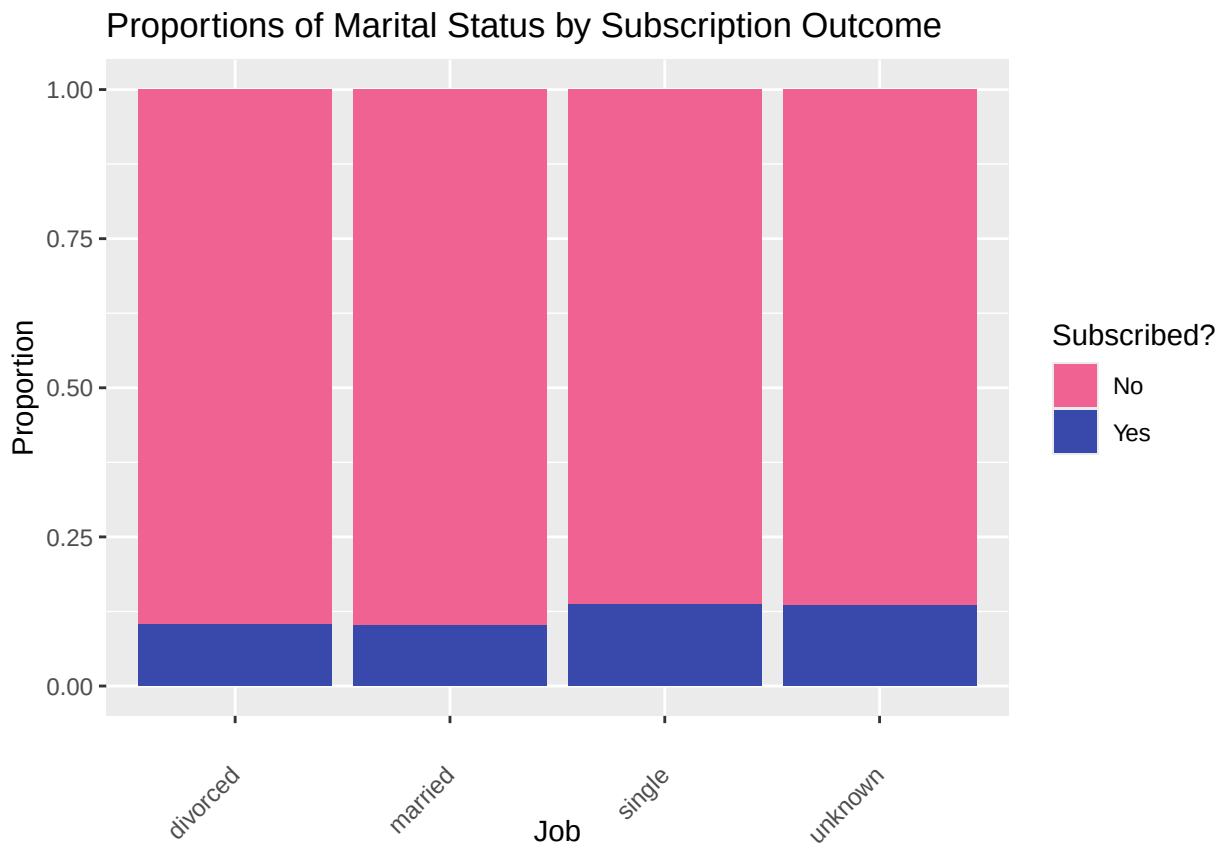
From the Chi-sq test results, I am going to consider poutcome, contact, job

Plots

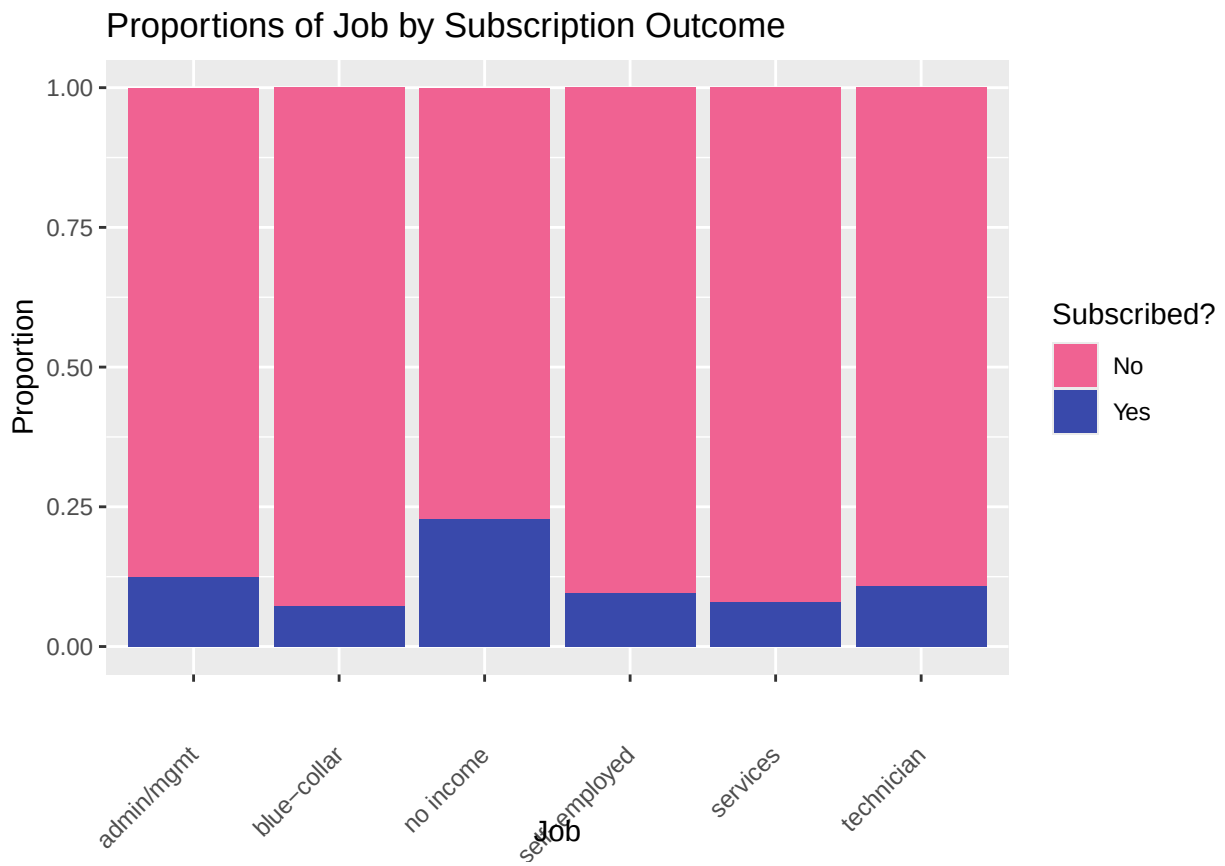
```
# Subscription
ggplot(bank, aes(x = as.factor(y))) + geom_bar(fill="#3949ab") +
  scale_x_discrete(labels = c("0" = "No", "1" = "Yes")) +
  labs(x = "Subscribed?",
       ylab = "Count",
       title ="Distribution of Subscription Outcome" )
```



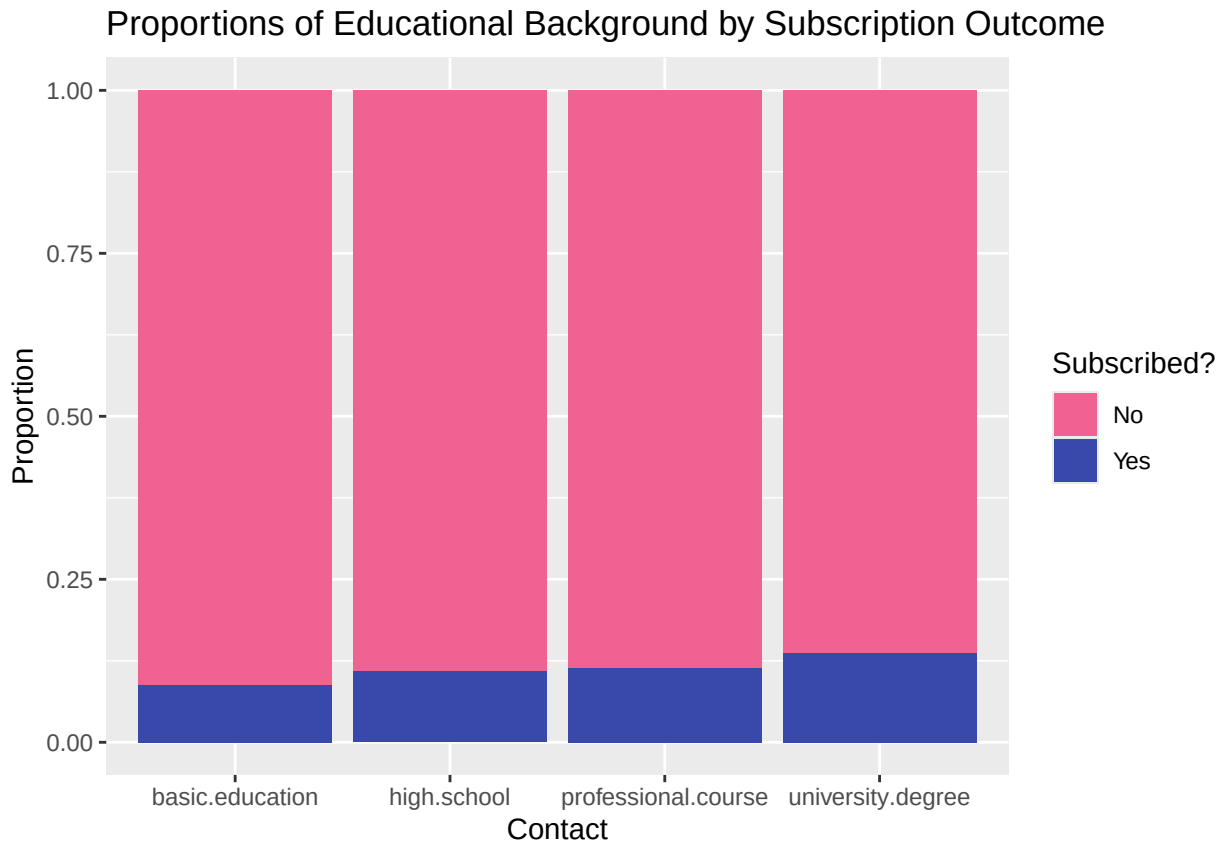

```r
# Marital
ggplot(bank, aes(x = marital, fill = factor(y))) + geom_bar(position = "fill") +
  labs(title = "Proportions of Marital Status by Subscription Outcome",
       x = "Job",
       y = "Proportion",
       fill = "Subscribed?") +
  scale_fill_manual(values = c("#f06292", "#3949ab"),
                    labels = c("No", "Yes")) +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust = 1))
```



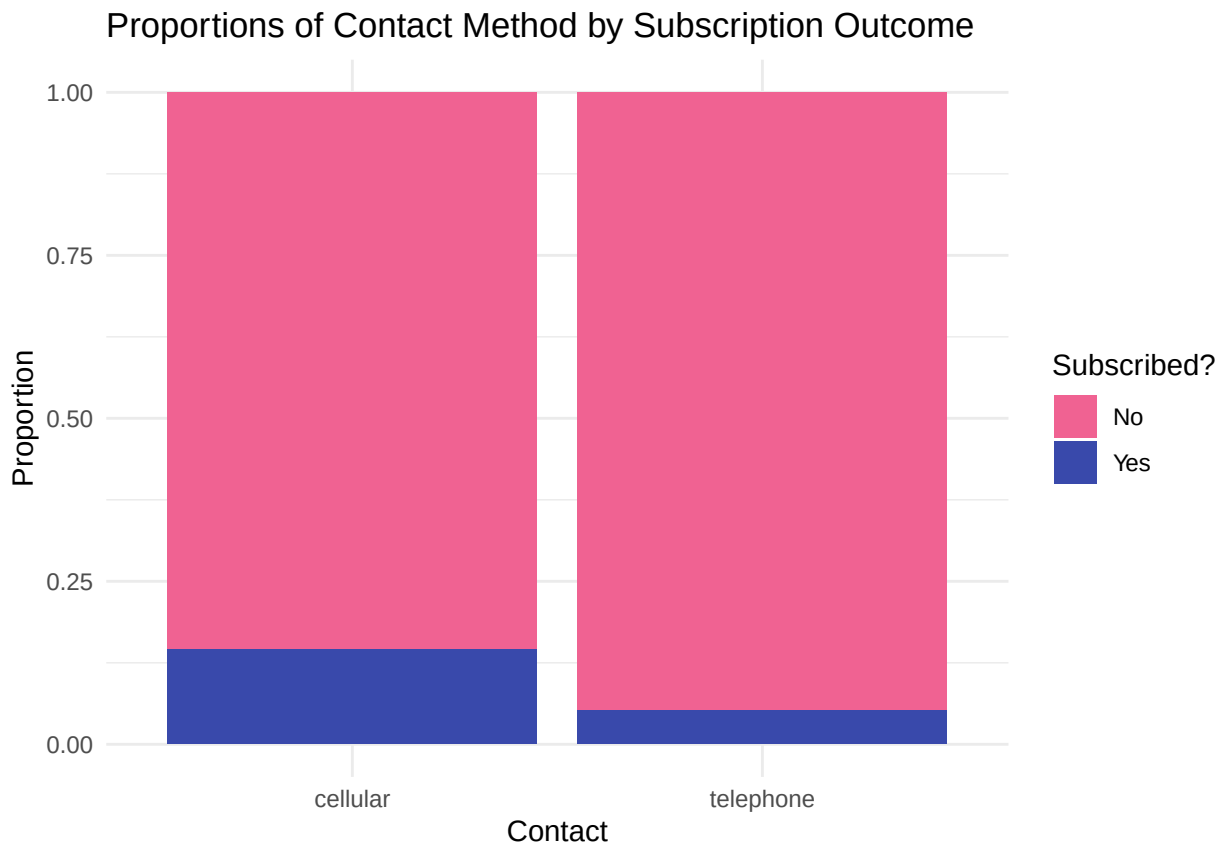

```
# Job
ggplot(bank, aes(x = job, fill = factor(y))) + geom_bar(position = "fill") +
  labs(title = "Proportions of Job by Subscription Outcome",
       x = "Job",
       y = "Proportion",
       fill = "Subscribed?") +
  scale_fill_manual(values = c("#f06292", "#3949ab"), labels = c("No", "Yes")) +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust = 1))
```



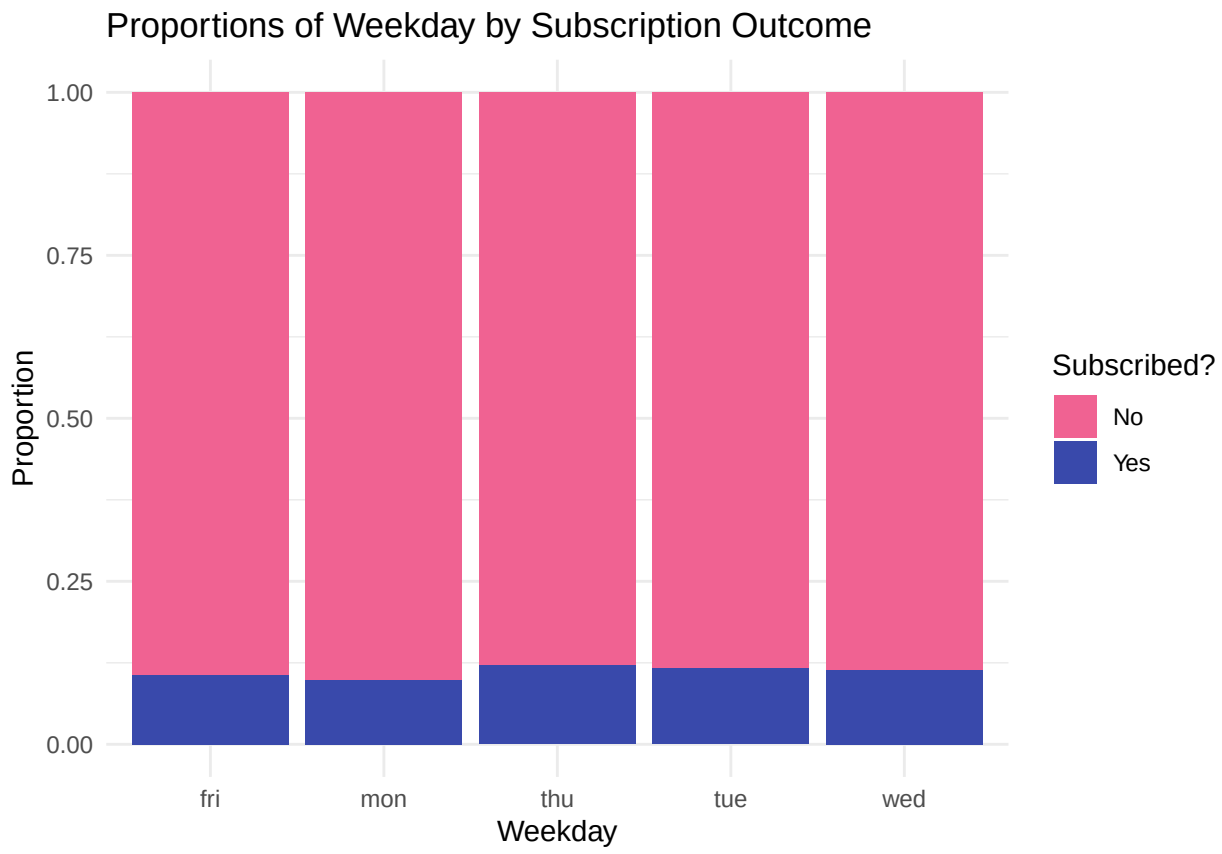

```
# Education
ggplot(bank, aes(x = education, fill = factor(y))) + geom_bar(position = "fill") +
  labs(title = "Proportions of Educational Background by Subscription Outcome",
       x = "Contact",
       y = "Proportion",
       fill = "Subscribed?") +
  scale_fill_manual(values = c("#f06292", "#3949ab"),labels = c("No", "Yes"))
```



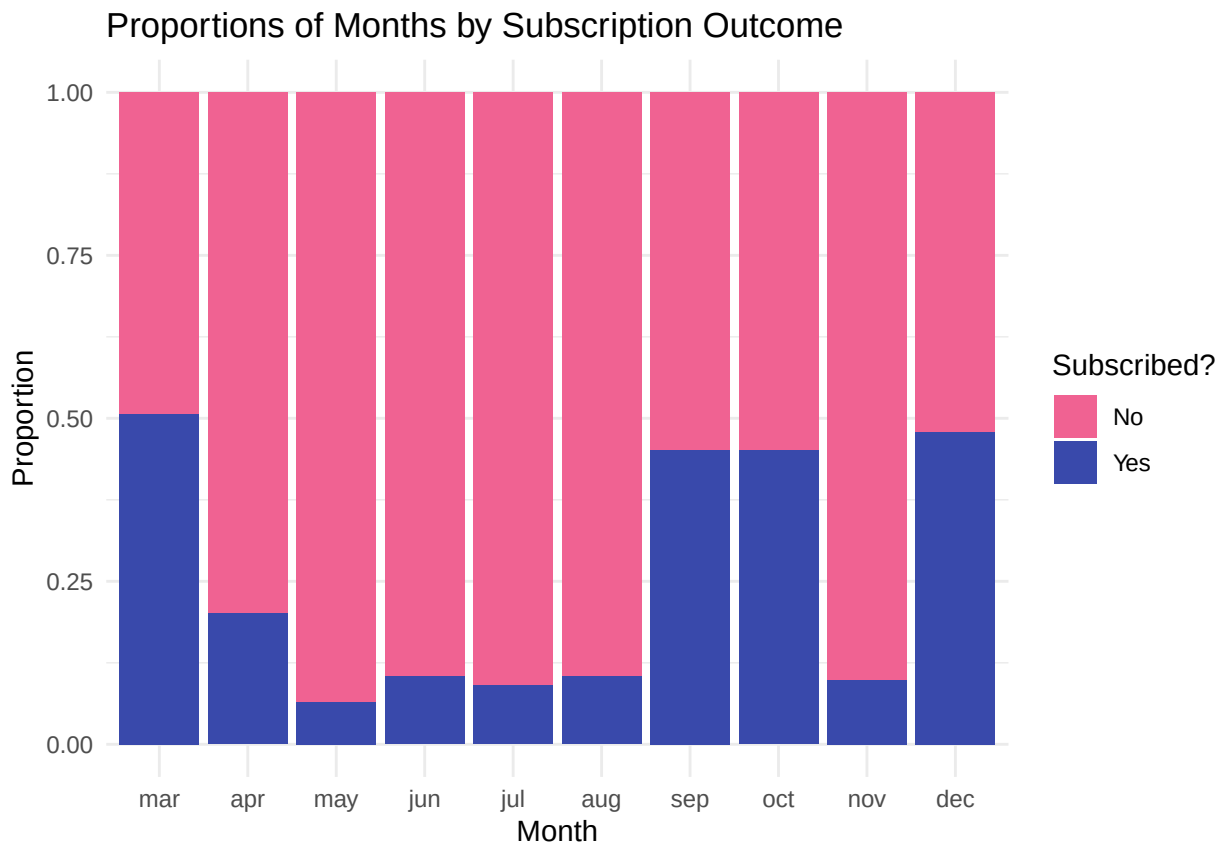

```
# Contact
ggplot(bank, aes(x = contact, fill = factor(y))) + geom_bar(position = "fill") +
  labs(title = "Proportions of Contact Method by Subscription Outcome",
       x = "Contact",
       y = "Proportion",
       fill = "Subscribed?") +
  scale_fill_manual(values = c("#f06292", "#3949ab"),labels = c("No", "Yes")) +
  theme_minimal()
```



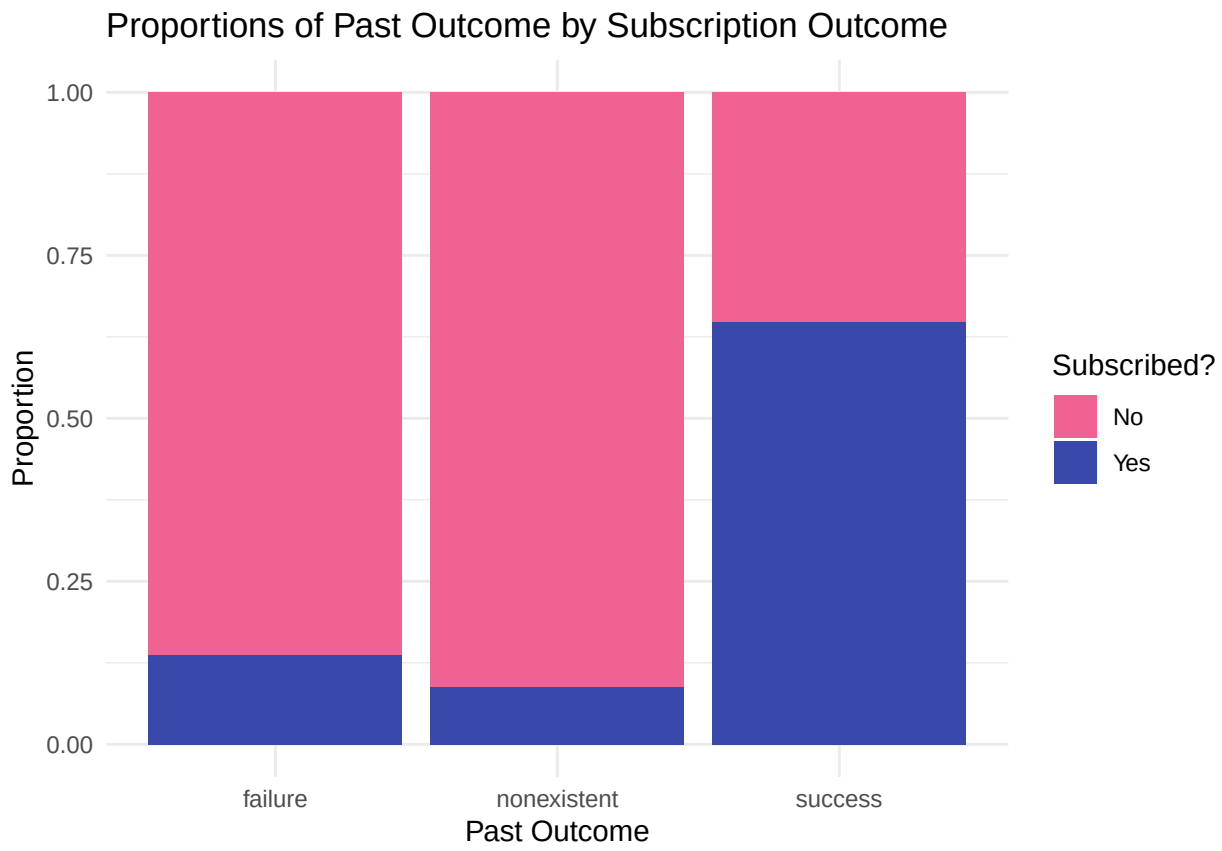

```r
# Day of Week
ggplot(bank, aes(x = day_of_week, fill = factor(y))) + geom_bar(position = "fill") +
  labs(title = "Proportions of Weekday by Subscription Outcome",
       x = "Weekday",
       y = "Proportion",
       fill = "Subscribed?") +
  scale_fill_manual(values = c("#f06292", "#3949ab"), labels = c("No", "Yes")) +
  theme_minimal()
```




```
# Month

month_order <- c("mar", "apr", "may", "jun", "jul", "aug", "sep", "oct", "nov", "dec")
bank$month <- factor(bank$month, levels = month_order, ordered = TRUE)

ggplot(bank, aes(x = month, fill = factor(y))) + geom_bar(position = "fill") +
  labs(title = "Proportions of Months by Subscription Outcome",
       x = "Month",
       y = "Proportion",
       fill = "Subscribed?") +
  scale_fill_manual(values = c("#f06292", "#3949ab"), labels = c("No", "Yes")) +
  theme_minimal()
```



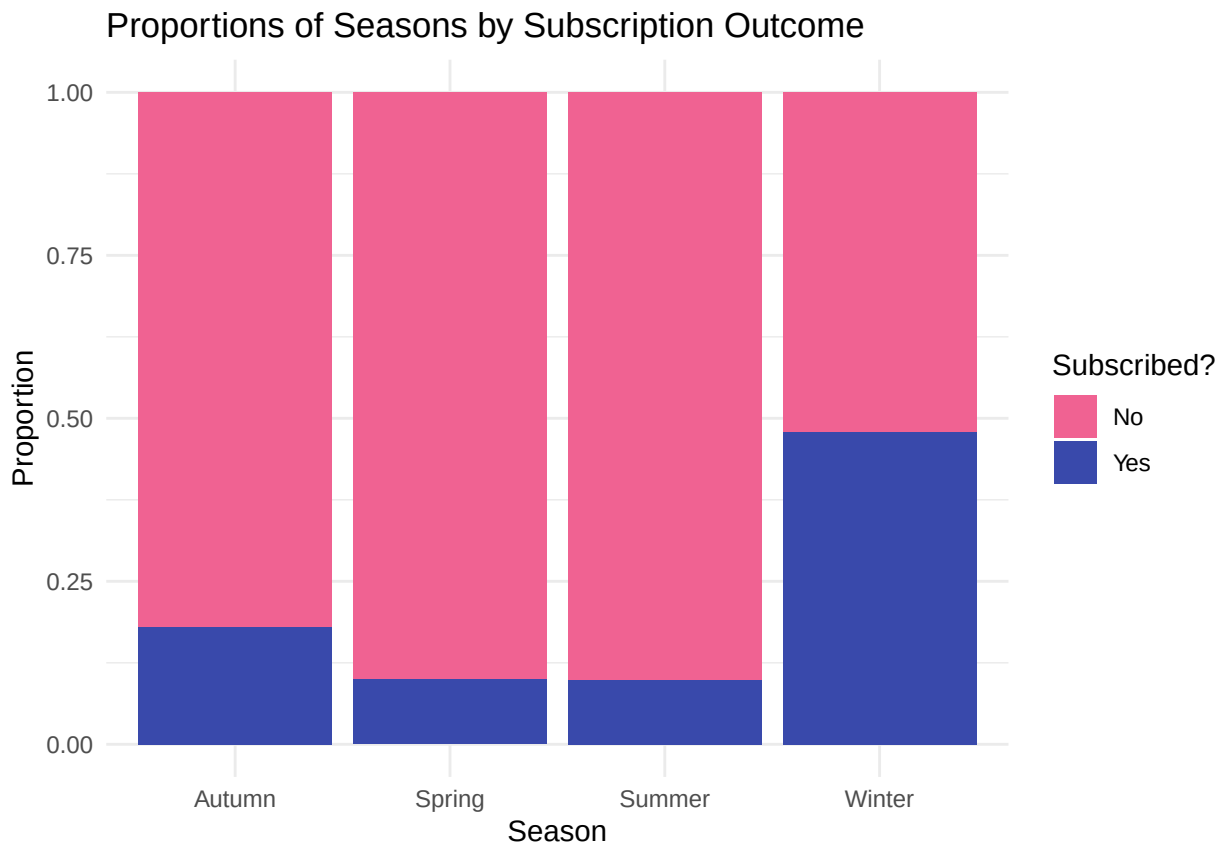

```r
# Previous Outcome

ggplot(bank, aes(x = poutcome, fill = factor(y))) + geom_bar(position = "fill") +
  labs(title = "Proportions of Past Outcome by Subscription Outcome",
       x = "Past Outcome",
       y = "Proportion",
       fill = "Subscribed?") +
  scale_fill_manual(values = c("#f06292", "#3949ab"),labels = c("No", "Yes")) +
  theme_minimal()
```



Season Variable

For an unknown reason, January and February seems to not have any data from the original dataset.

```
# Recode the month variable into seasons, excluding January and February

bank <- bank %>% mutate(
  season = case_when(
    month %in% c("mar", "apr", "may") ~ "Spring",      # March, April, May
    month %in% c("jun", "jul", "aug") ~ "Summer",      # June, July, August
    month %in% c("sep", "oct", "nov") ~ "Autumn",      # September, October, November
    month == "dec" ~ "Winter",                         # December
    TRUE ~ NA_character_                               # Exclude January and February
  )
)

ggplot(bank, aes(x = season, fill = factor(y))) + geom_bar(position = "fill") +
  labs(title = "Proportions of Seasons by Subscription Outcome",
       x = "Season",
       y = "Proportion",
       fill = "Subscribed?") +
  scale_fill_manual(values = c("#f06292", "#3949ab"), labels = c("No", "Yes")) +
  theme_minimal()
```




```
z <-chisq.test(table(bank$season))

cat("Chi-Square Test for Season", "\n")
```

```
## Chi-Square Test for Season
```

```
cat("Chi-Squared Value:", z$statistic, "\n")
```

```
## Chi-Squared Value: 22199.01
```

```
cat("p-value:", z$p.value, "\n")
```

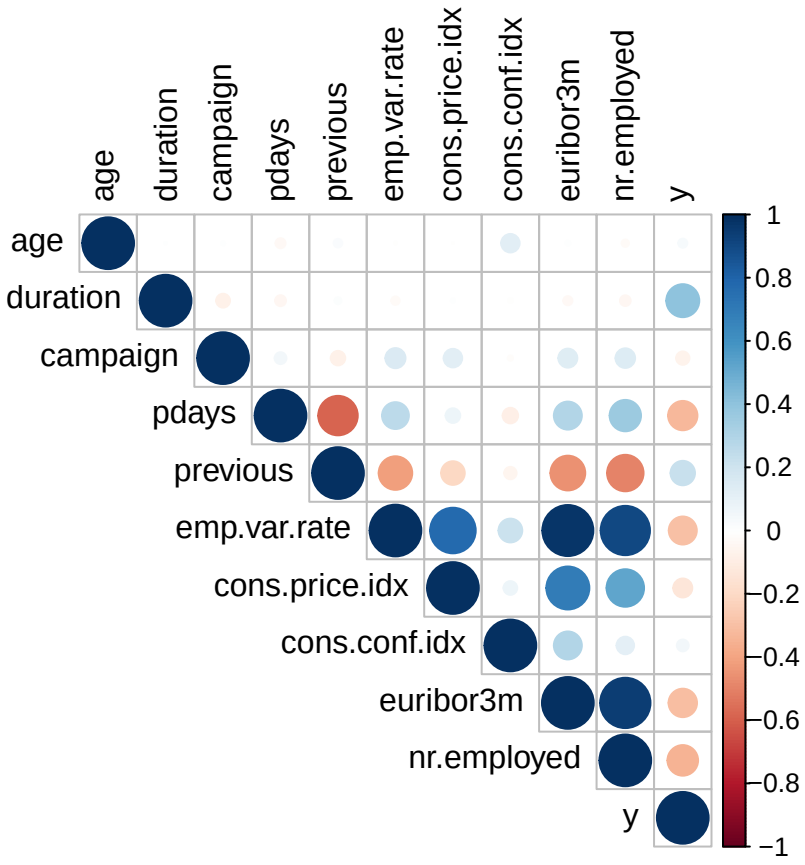
```
## p-value: 0
```

Consider "Season" as a relevant predictor for model building.

Numerical Data

Correlation Matrix

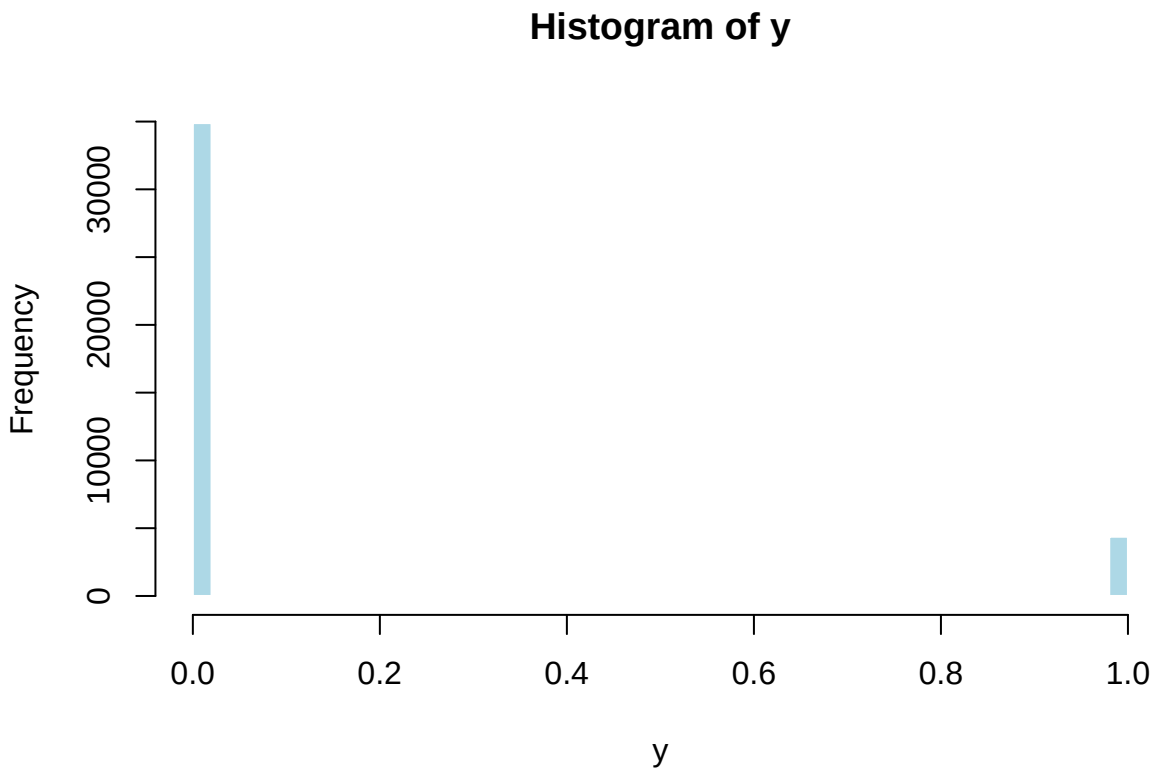
```r
numerical_columns <- colnames(bank[, sapply(bank, is.numeric)])

# Correlation Matrix for all numerical variables
cor_matrix <- cor(bank[, numerical_columns], use = "complete.obs")

# Plot correlation matrix
corrplot(cor_matrix, method = "circle", type = "upper", tl.col = "black")
```

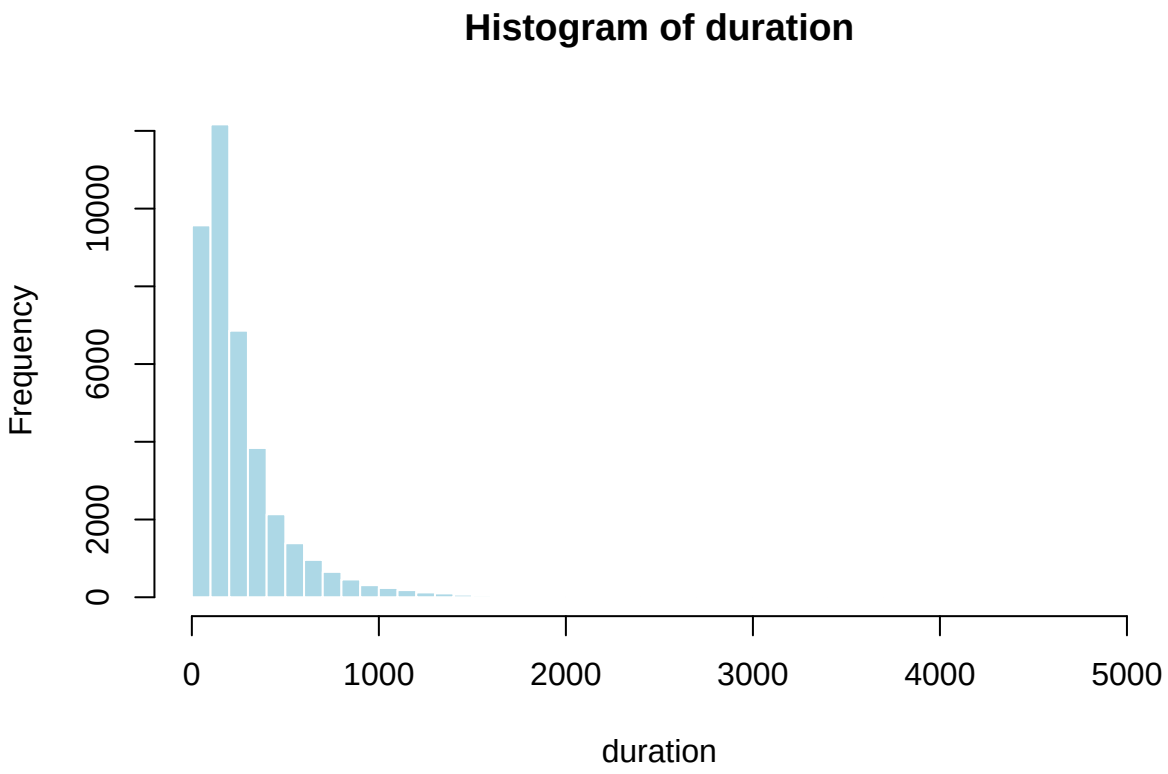



```r
# Point-biserial correlation with binary outcome (y)
pb_cor <- sapply(bank[, c(numerical_columns)], function(x) biserial.cor(x, bank$y))
pb_cor_df <- data.frame(Predictor = colnames(bank[, numerical_columns]),
                        Correlation_with_y = pb_cor)

# Top 4 Predictors
top_4_predictors <- pb_cor_df[order(abs(pb_cor_df$Correlation_with_y), decreasing = TRUE),][1:6,]
print(top_4_predictors[1:2])
```

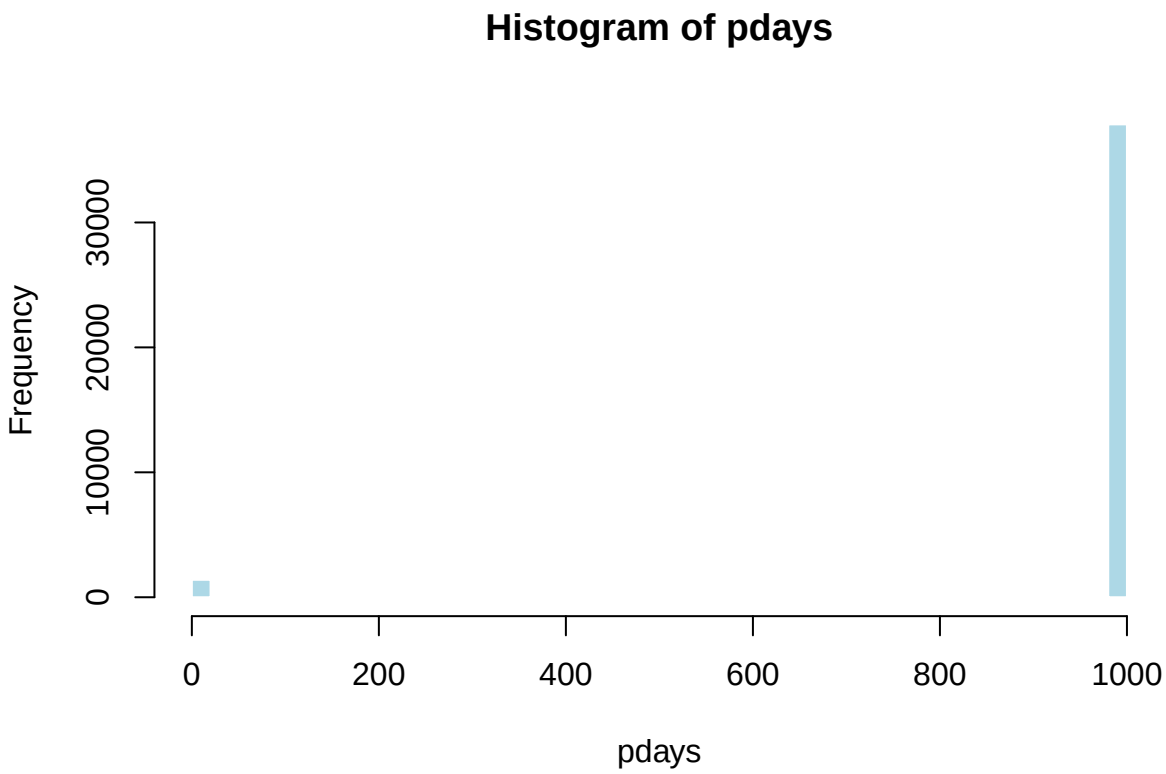
```
##                 Predictor Correlation_with_y
## y                       y         -1.0000000
## duration         duration         -0.4062711
## nr.employed   nr.employed          0.3494750
## pdays               pdays          0.3209351
## euribor3m       euribor3m          0.3020506
## emp.var.rate emp.var.rate          0.2932466
```

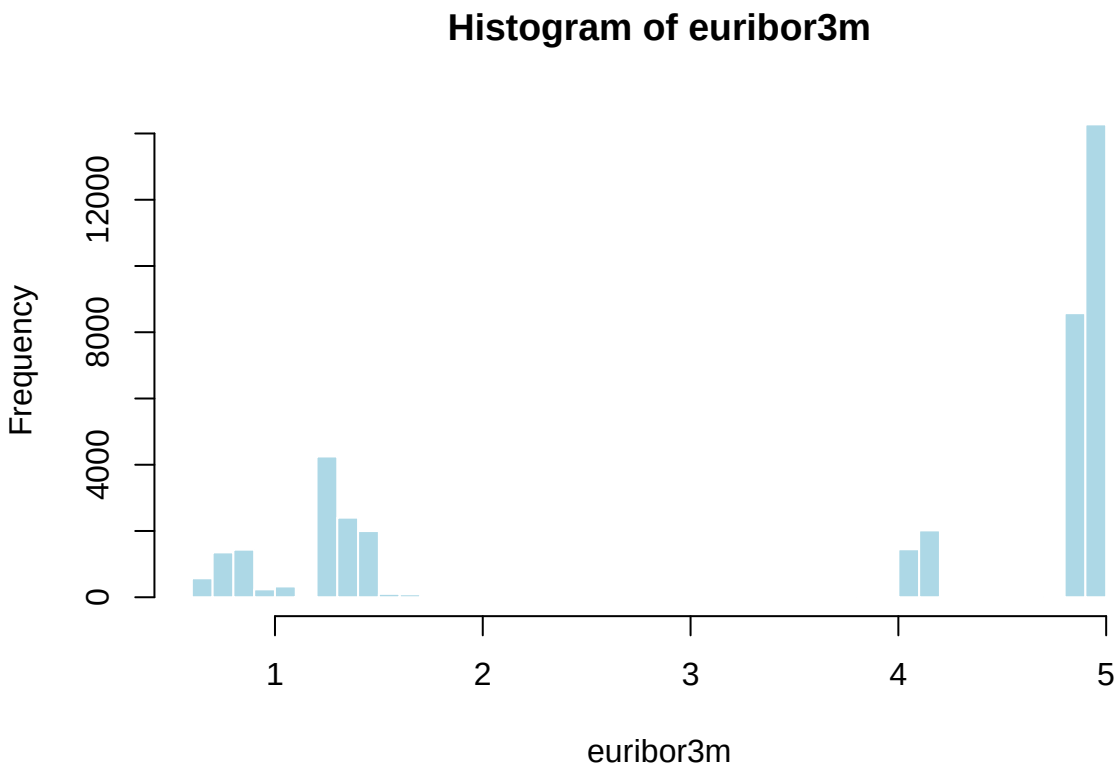
```r
for (x in top_4_predictors[,"Predictor"]) {
  hist(bank[[x]], main = paste("Histogram of", x), xlab = x, col = "lightblue",
       border = "white", breaks = 40)
}
```




Histogram of duration
Frequency
0
2000
6000
10000
0
1000
2000
3000
4000
5000
duration


Histogram of nr.employed
Frequency
0
5000
10000
15000
4950
5000
5050
5100
5150
5200
nr.employed

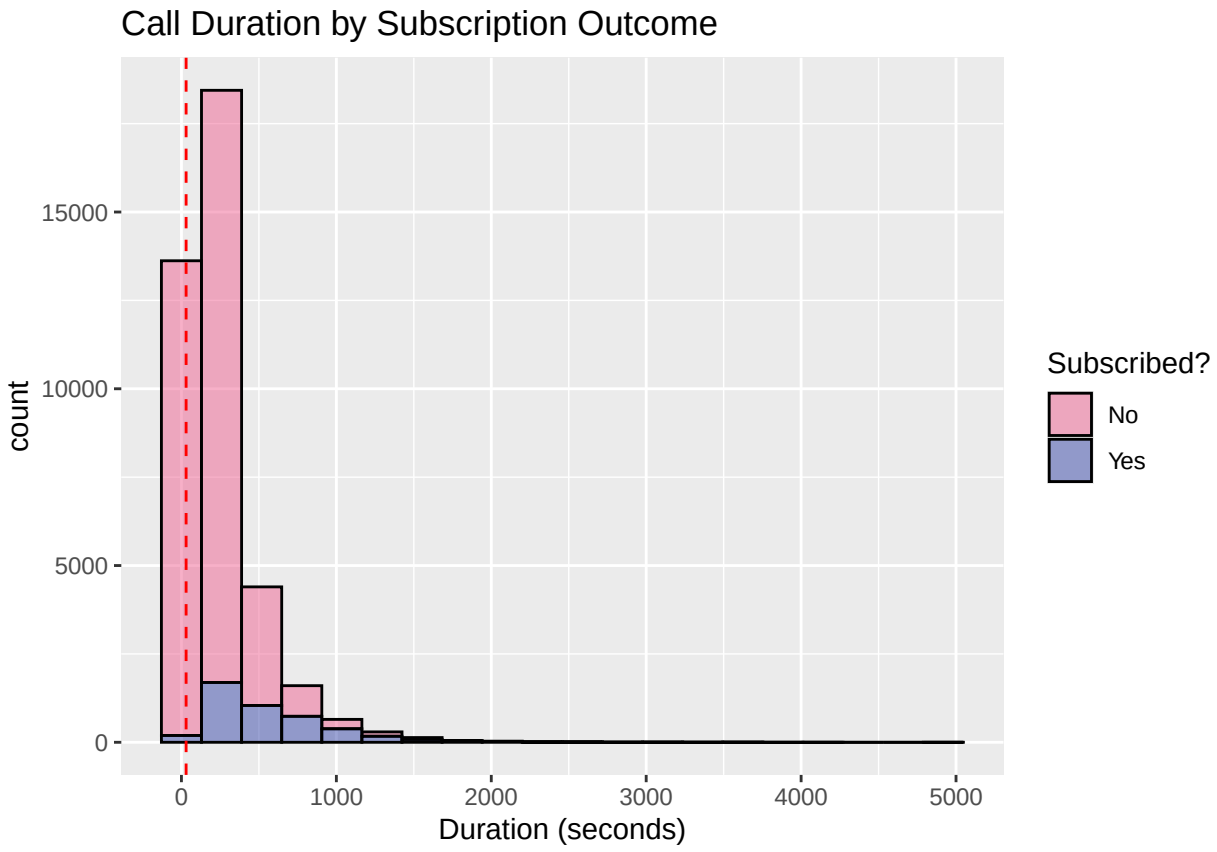

Histogram of pdays
Frequency
0
10000
20000
30000
0
200
400
600
800
1000
pdays


Histogram of euribor3m
Frequency
0
4000
8000
12000
1
2
3
4
5
euribor3m

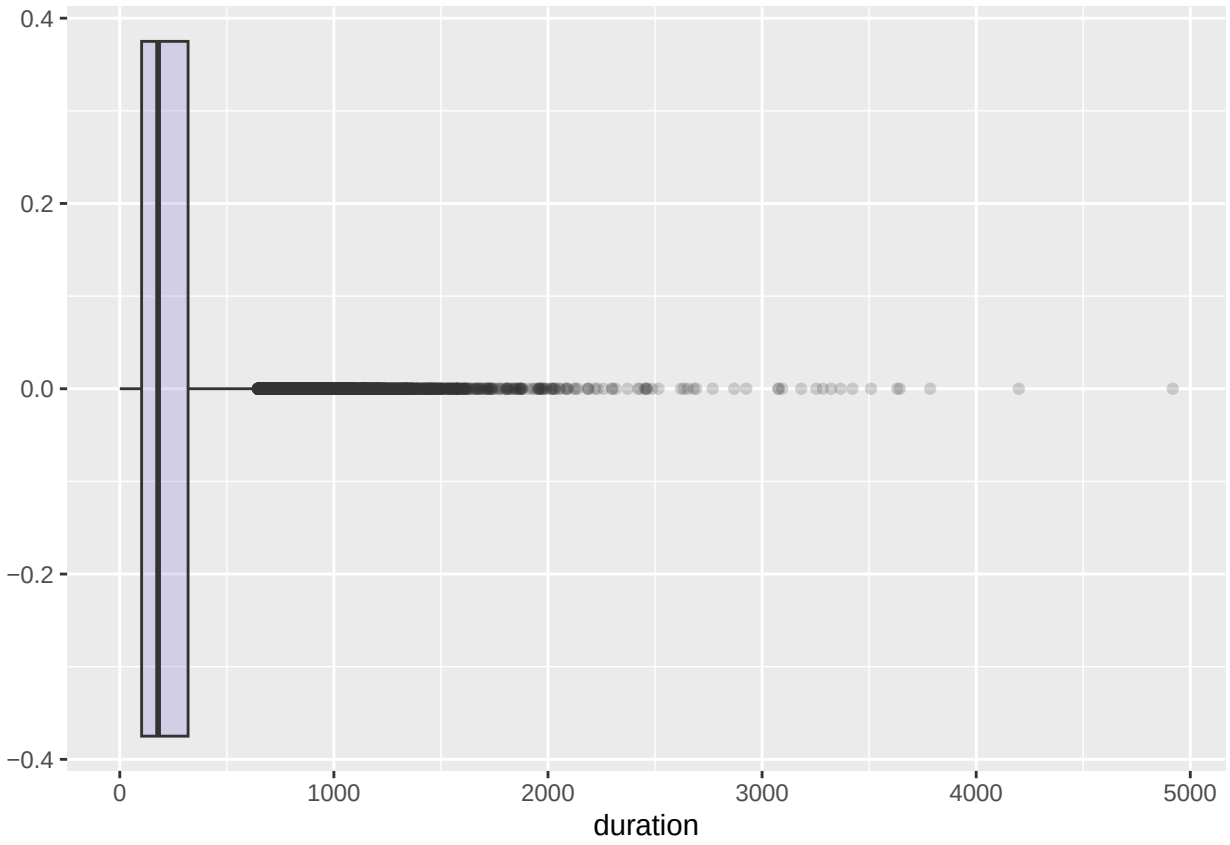
Consider Duration and Number of Employed Citizens for model building.

Duration of Calls

```
# Duration
ggplot(bank, aes(x = duration, fill = factor(y))) +
  geom_histogram(position = "stack", bins = 20, color = "black", alpha = 0.5) +
  geom_vline(xintercept = 30, linetype = "dashed", color = "red") +
  scale_fill_manual(values = c("#f06292", "#3949ab"), labels = c("No", "Yes")) +
  labs(title = "Call Duration by Subscription Outcome", x = "Duration (seconds)", fill = "Subscribed?")
```



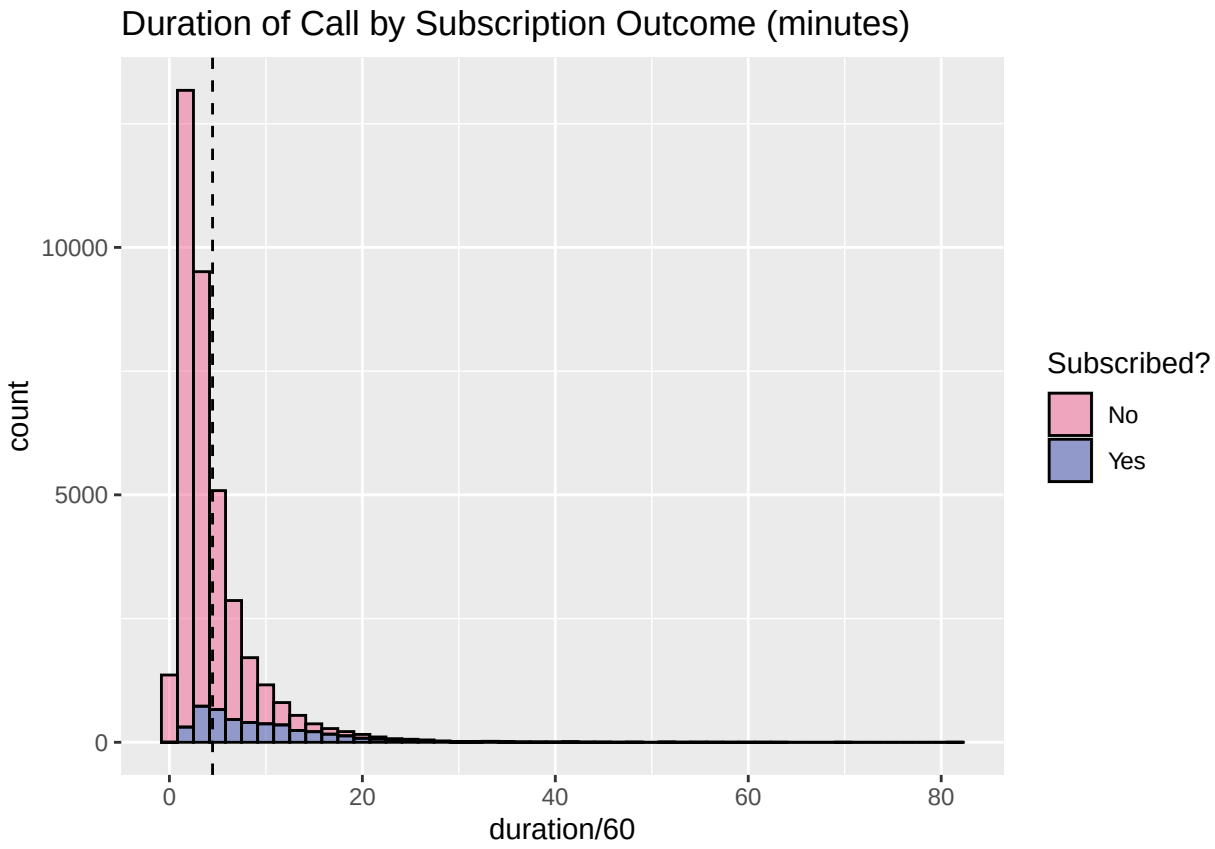

```
# Boxplot of Duration
ggplot(bank, aes(x=duration)) + geom_boxplot(fill="slateblue", alpha=0.2) + xlab("duration")
```



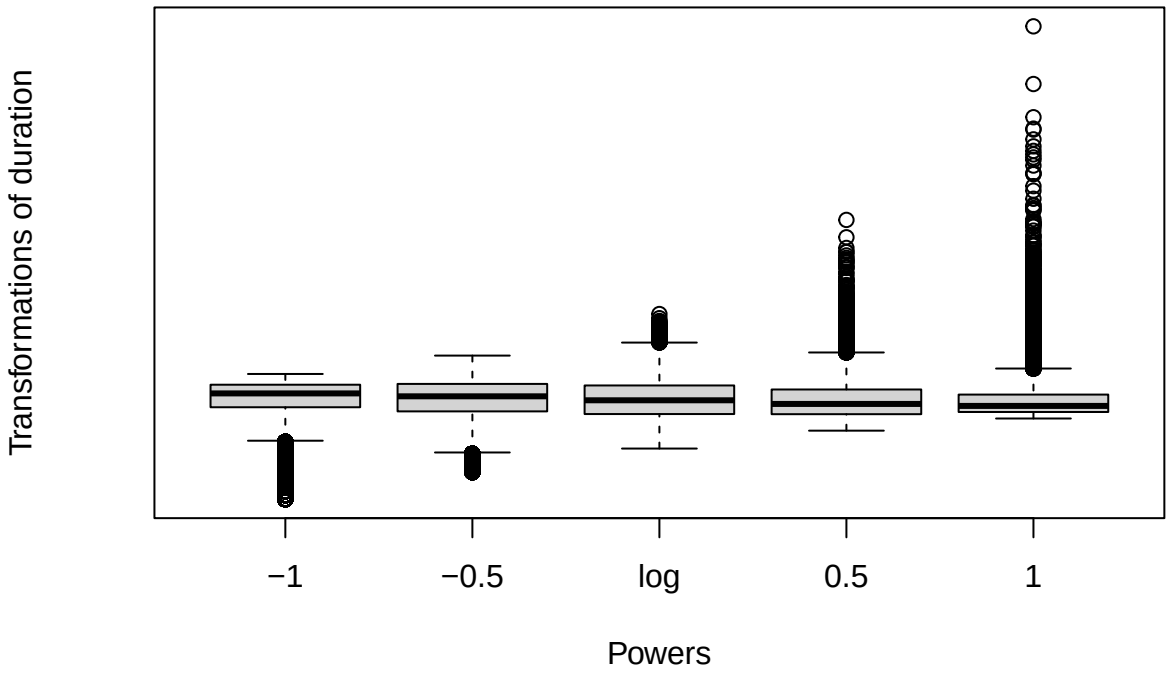

```
summary(bank$duration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   102.0   179.0   258.1   319.0  4918.0
```

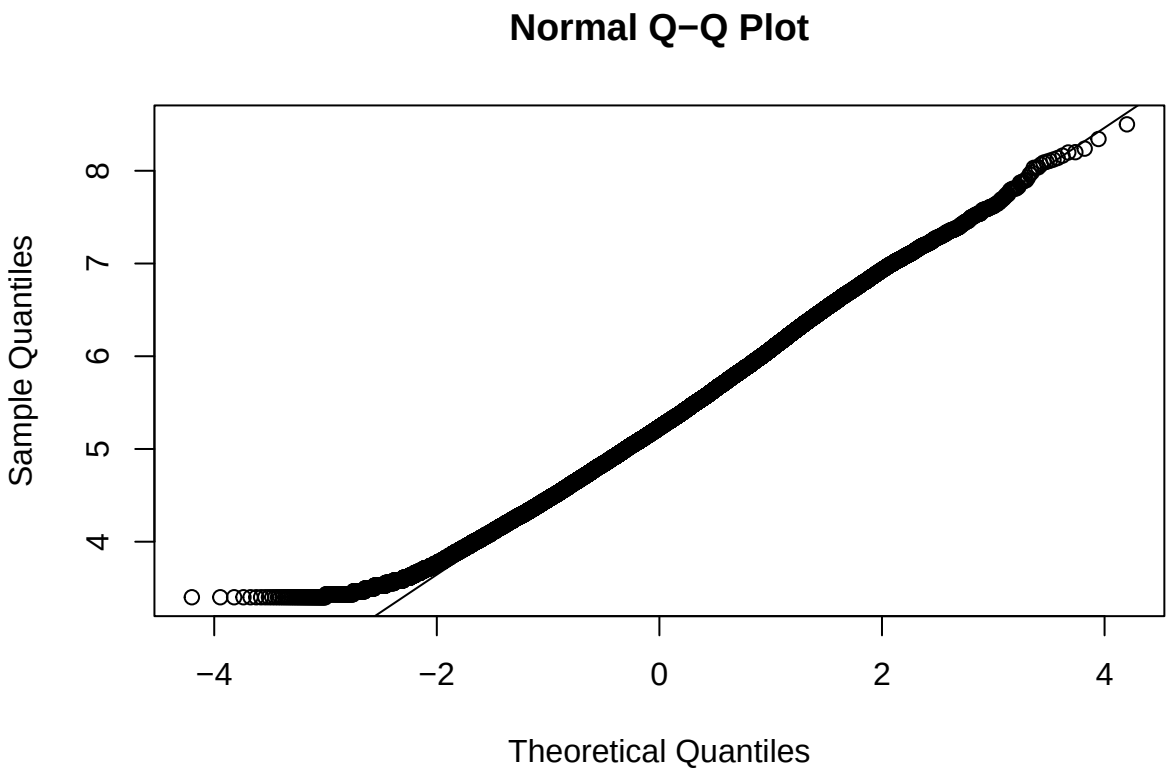
```
# Remove Duration less than 30 seconds (Likely dropped calls)
bank <- bank[bank$duration >= 30,]

# Look at Distribution by Minutes instead of Seconds
ggplot(bank, aes(x = duration/60 , fill = factor(y))) +
  geom_histogram(position = "stack", bins = 50, color = "black", alpha = 0.5) +
  geom_vline(aes(xintercept = mean(duration/60, na.rm = TRUE)), linetype = "dashed",
             color = "black") +  # Add dashed line for mean
  scale_fill_manual(values = c("#f06292", "#3949ab"), labels = c("No", "Yes")) +
  labs(title = "Duration of Call by Subscription Outcome (minutes)", fill = "Subscribed?")
```




```
# Boxcox Transformation
symbox(~duration, data = bank)
```




```
# Log transforming Duration
qqnorm(log(bank$duration))
qqline(log(bank$duration))
```




```
bank_mod <- bank # Transfer dataset to bank_mod to distinguish the additional changes

bank_mod$duration <- log(bank_mod$duration + 1)

dim(bank_mod)
```

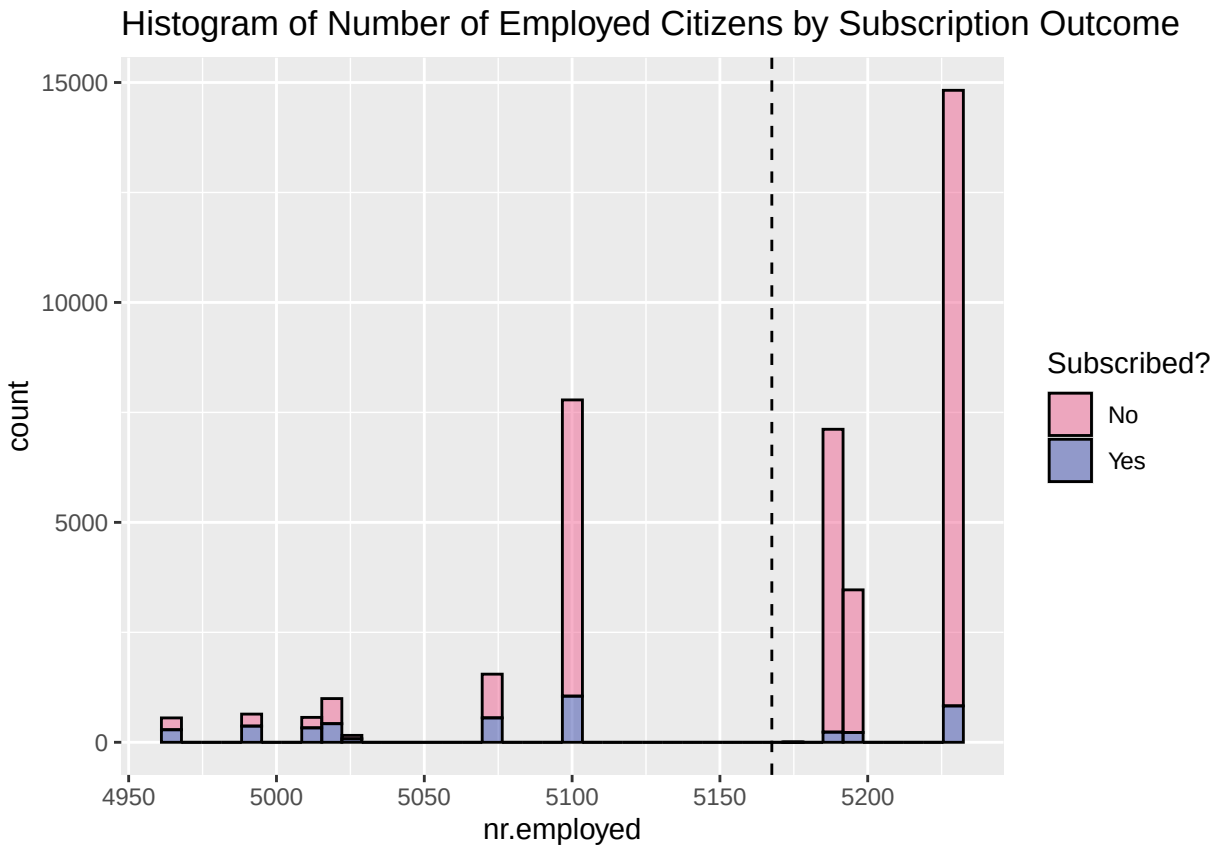
```
## [1] 37656    23
```

```
# After this code above, Total n = 37,656
```

Number of Employeed Citizens

Since the distribution of Nr.employed is very scattered, I turned it into a categorical variable

```
# nr.employed histogram

ggplot(bank_mod, aes(x = nr.employed, fill = factor(y))) +
  geom_histogram(position = "stack", bins = 40, color = "black", alpha = 0.5) +
  geom_vline(aes(xintercept = mean(nr.employed, na.rm = TRUE)), linetype = "dashed",
             color = "black") +
  scale_fill_manual(values = c("#f06292", "#3949ab"), labels = c("No", "Yes")) +
  labs(title = "Histogram of Number of Employed Citizens by Subscription Outcome", fill = "Subscribed?")
```




```
# Check the distribution of the new variable

table(bank_mod$nr.employed.cat)
```

```
##
##  Very Low       Low      High Very High
##      2912      9333     10589     14822
```

First Model

First, I decided to use 5 categorical variables and 1 numerical variable (nr.employed converted from numerical to categorical)

```
bank_mod$job <- as.factor(bank_mod$job)
bank_mod$contact <- as.factor(bank_mod$contact)
bank_mod$season <- as.factor(bank_mod$season)
bank_mod$poutcome <- as.factor(bank_mod$poutcome)
bank_mod$poutcome <- relevel(bank_mod$poutcome, ref = "nonexistent")

# Run Logistic Model
m1 <- glm(y ~ poutcome + contact + job + season + duration + nr.employed.cat,
          data = bank_mod,
          family = binomial)
summary(m1)
```

```
##
## Call:
## glm(formula = y ~ poutcome + contact + job + season + duration +
##     nr.employed.cat, family = binomial, data = bank_mod)
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -11.90828    0.20122 -59.179  < 2e-16 ***
## poutcomefailure           -0.52655    0.06531  -8.062 7.48e-16 ***
## poutcomesuccess            1.39165    0.08296  16.774  < 2e-16 ***
## contacttelephone          -0.25440    0.06137  -4.145 3.39e-05 ***
## jobblue-collar            -0.40566    0.05921  -6.851 7.33e-12 ***
## jobno income               0.18487    0.06790   2.723  0.00648 **
## jobself-employed          -0.13954    0.08724  -1.600  0.10970
## jobservices               -0.31828    0.08022  -3.968 7.26e-05 ***
## jobtechnician             -0.01113    0.06199  -0.180  0.85747
## seasonSpring              -0.50567    0.07976  -6.340 2.30e-10 ***
## seasonSummer               0.79534    0.09040   8.798  < 2e-16 ***
## seasonWinter              -0.26460    0.20613  -1.284  0.19925
## duration                   2.13517    0.03423  62.371  < 2e-16 ***
## nr.employed.catLow        -1.30829    0.07751 -16.880  < 2e-16 ***
## nr.employed.catHigh       -2.96545    0.08572 -34.594  < 2e-16 ***
## nr.employed.catVery High  -3.87038    0.09879 -39.178  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27032  on 37655  degrees of freedom
## Residual deviance: 15852  on 37640  degrees of freedom
## AIC: 15884
##
## Number of Fisher Scoring iterations: 7
```

```
# Pseudo R^2
r2_m1 <- 1 - m1$deviance/m1$null.deviance
print(r2_m1)
```

```
## [1] 0.4135798
```

```
# Look for Multicollinearity
vif(m1)
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## poutcome          1.325161  2        1.072920
## contact           1.465881  1        1.210736
## job               1.123566  5        1.011719
## season            4.534239  3        1.286523
## duration          1.267108  1        1.125659
## nr.employed.cat 7.748113  3        1.406693
```

Every predictor is significant except for job(self-employed) and season(Winter)

AIC: 15884 R^2: 0.4135798

ALL VIFs < 5 except for Nr.employed.cat

Interaction Model

This model is the same as the first model with season x duration and season x nr.employed.cat

```
# Interaction Logistic Model with relevant interactions

int <- glm(y ~ job + contact + season + poutcome + duration + nr.employed.cat +
             season:duration + season:nr.employed.cat,data = bank_mod, family = binomial)
summary(int)
```

```
##
## Call:
## glm(formula = y ~ job + contact + season + poutcome + duration +
##     nr.employed.cat + season:duration + season:nr.employed.cat,
##     family = binomial, data = bank_mod)
##
## Coefficients: (6 not defined because of singularities)
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        -8.49793    0.38239 -22.223  < 2e-16 ***
## jobblue-collar                     -0.37040    0.05982  -6.192 5.96e-10 ***
## jobno income                        0.14985    0.06825   2.196 0.028114 *
## jobself-employed                   -0.13021    0.08771  -1.485 0.137660
## jobservices                        -0.28642    0.08095  -3.538 0.000403 ***
## jobtechnician                       0.01689    0.06264   0.270 0.787445
## contacttelephone                   -0.20586    0.06681  -3.081 0.002062 **
## seasonSpring                       -2.51260    0.48718  -5.157 2.50e-07 ***
## seasonSummer                       -6.44314    0.52539 -12.263  < 2e-16 ***
## seasonWinter                        0.29212    1.71970   0.170 0.865114
## poutcomefailure                    -0.51721    0.06457  -8.010 1.15e-15 ***
## poutcomesuccess                     1.31774    0.08201  16.068  < 2e-16 ***
## duration                            1.49912    0.06867  21.832  < 2e-16 ***
## nr.employed.catLow                 -0.33157    0.12555  -2.641 0.008267 **
## nr.employed.catHigh                -2.41399    0.10463 -23.072  < 2e-16 ***
## nr.employed.catVery High           -3.67830    0.12895 -28.526  < 2e-16 ***
## seasonSpring:duration               0.54226    0.08665   6.258 3.90e-10 ***
## seasonSummer:duration               1.21064    0.09327  12.980  < 2e-16 ***
## seasonWinter:duration              -0.07100    0.30379  -0.234 0.815203
## seasonSpring:nr.employed.catLow    -1.93446    0.16833 -11.492  < 2e-16 ***
## seasonSummer:nr.employed.catLow          NA         NA      NA       NA
## seasonWinter:nr.employed.catLow          NA         NA      NA       NA
## seasonSpring:nr.employed.catHigh   -1.52402    0.17766  -8.579  < 2e-16 ***
## seasonSummer:nr.employed.catHigh         NA         NA      NA       NA
## seasonWinter:nr.employed.catHigh    1.80053    1.13622   1.585 0.113043
## seasonSpring:nr.employed.catVery High       NA         NA      NA       NA
## seasonSummer:nr.employed.catVery High       NA         NA      NA       NA
## seasonWinter:nr.employed.catVery High       NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27032  on 37655  degrees of freedom
## Residual deviance: 15521  on 37634  degrees of freedom
## AIC: 15565
##
## Number of Fisher Scoring iterations: 7
```

```
# Pseudo R^2

r2_int <- 1 - int$deviance/int$null.deviance
print(r2_int)
```

```
## [1] 0.4258156
```

AIC: 15565 R^2: 0.4258156

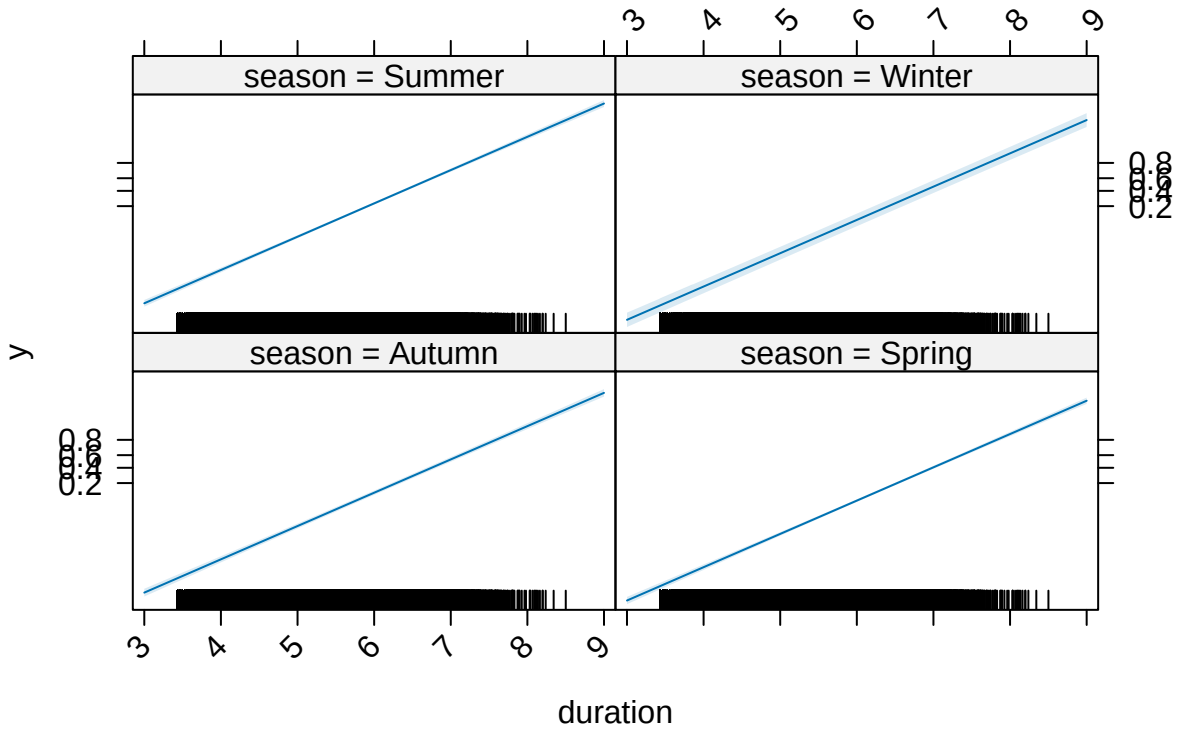
Interaction Plot

```
library(effects)
```

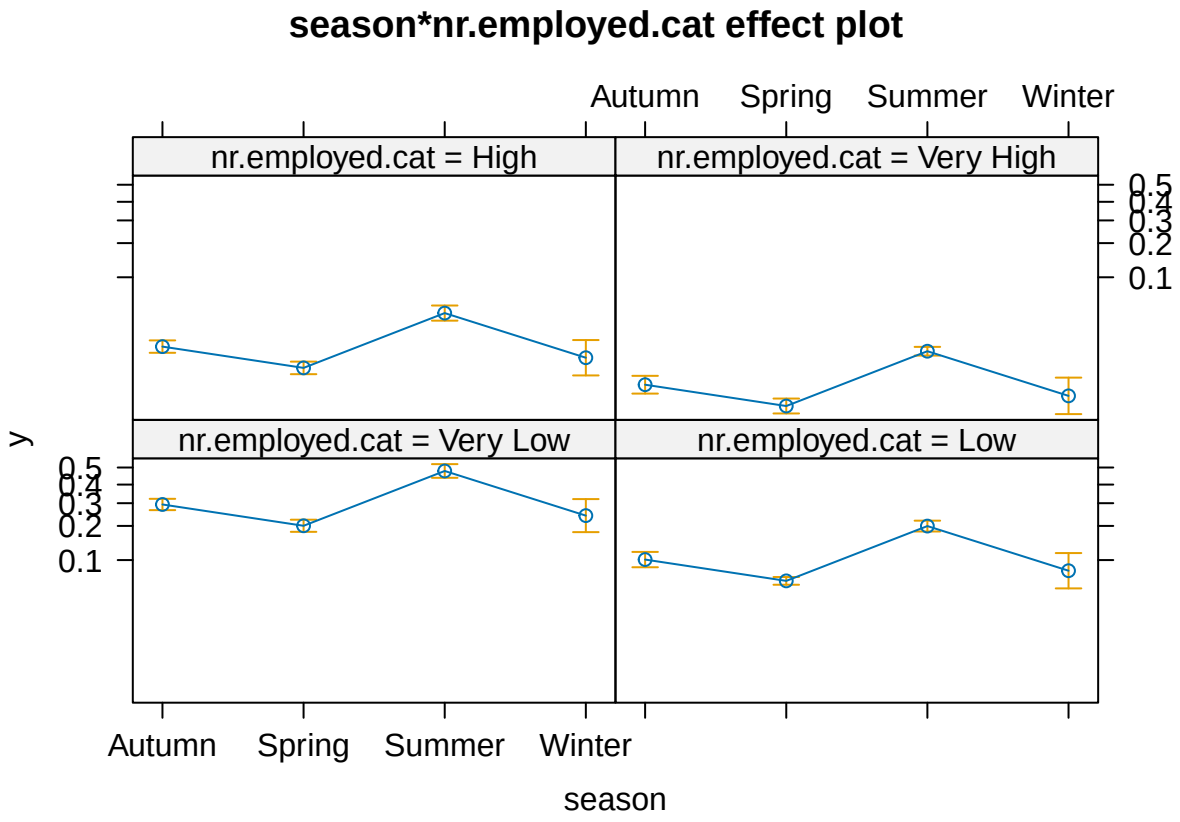
```
## Warning: package 'effects' was built under R version 4.4.3
```

```
## Use the command
##     lattice::trellis.par.set(effectsTheme())
##   to customize lattice options for effects plots.
## See ?effectsTheme for details.
```

```
int1 <- Effect(c("season", "duration"), m1)
layout(matrix(1), heights = c(10))
plot(int1, axes=list(x=list(rotate=45)))
```




```
int2 <- Effect(c("season", "nr.employed.cat"), m1)
layout(matrix(1), heights = c(10))
plot(int2)
```



LRT Comparison

```
anova(m1,int, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ poutcome + contact + job + season + duration + nr.employed.cat
## Model 2: y ~ job + contact + season + poutcome + duration + nr.employed.cat +
##     season:duration + season:nr.employed.cat
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1     37640      15852
## 2     37634      15521  6   330.75 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There's a significant difference between the First and Interaction Model

Pearson's GOF Test

```r
# Chi-Square GOF test
pearson_stat <- sum(residuals(m1, type = "pearson")^2)
df <- nrow(bank_mod) - length(coef(m1))

# Critical value
critical_value <- qchisq(0.95, df)
p_value <- 1 - pchisq(pearson_stat, df)

cat("Chi-Square GOF Test\n")
```

```
## Chi-Square GOF Test
```

```r
cat("Chi-Square Statistic:", pearson_stat, "\n")
```

```
## Chi-Square Statistic: 30085.43
```

```r
cat("Degrees of Freedom:", df, "\n")
```

```
## Degrees of Freedom: 37640
```

```r
cat("Critical Value at alpha = 0.05:", critical_value, "\n")
```

```
## Critical Value at alpha = 0.05: 38092.44
```

```r
cat("p-value:", p_value, "\n")
```

```
## p-value: 1
```

```r
if (pearson_stat > critical_value) {
  cat("Decision: Reject the null hypothesis. The model shows lack of fit.\n")
} else {
  cat("Decision: Fail to reject the null hypothesis. The model fits well.\n")
}
```

```
## Decision: Fail to reject the null hypothesis. The model fits well.
```

Forward Selection Model

```r
# Null model (no predictors)
null_model <- glm(y ~ 1, family = binomial, data = bank_mod)

# Full model (all predictors w/o conflating variables and replacing month for season)
full_model <- glm(y ~ . - month - euribor3m - cons.price.idx - previous -nr.employed + season,
                  family = binomial, data = bank_mod)

# Forward selection using AIC
forward_model <- step(null_model,
                      scope = list(lower = null_model, upper = full_model),
                      direction = "forward")
```

```
## Start:  AIC=27033.55
## y ~ 1
##
##                   Df Deviance   AIC
## + duration         1    21616 21620
## + nr.employed.cat  3    22735 22743
## + emp.var.rate     1    23894 23898
## + pdays            1    24734 24738
## + poutcome         2    24779 24785
## + contact          1    26197 26201
## + job              5    26434 26446
## + default          2    26609 26615
## + season           3    26636 26644
## + education        3    26880 26888
## + campaign         1    26890 26894
## + marital          3    26931 26939
## + cons.conf.idx    1    26943 26947
## + age              1    26999 27003
## + day_of_week      4    27009 27019
## + housing          2    27027 27033
## <none>                  27032 27034
## + loan             2    27030 27036
##
## Step:  AIC=21619.7
## y ~ duration
##
##                   Df Deviance   AIC
## + nr.employed.cat  3    16964 16974
## + emp.var.rate     1    18013 18019
## + poutcome         2    19260 19268
## + pdays            1    19263 19269
## + contact          1    20701 20707
## + job              5    20888 20902
## + default          2    21140 21148
## + season           3    21146 21156
## + education        3    21383 21393
## + cons.conf.idx    1    21462 21468
## + campaign         1    21470 21476
## + marital          3    21510 21520
## + age              1    21576 21582
## + housing          2    21606 21614
## + day_of_week      4    21605 21617
## <none>                  21616 21620
## + loan             2    21614 21622
##
## Step:  AIC=16973.57
## y ~ duration + nr.employed.cat
##
##                  Df Deviance   AIC
## + poutcome        2    16408 16422
## + season          3    16419 16435
## + pdays           1    16472 16484
## + emp.var.rate    1    16726 16738
## + job             5    16783 16803
```

```
## + cons.conf.idx  1    16822 16834
## + education      3    16856 16872
## + default        2    16886 16900
## + contact        1    16915 16927
## + campaign       1    16946 16958
## + marital        3    16944 16960
## + day_of_week    4    16945 16963
## <none>                16964 16974
## + age            1    16963 16975
## + loan           2    16963 16977
## + housing        2    16963 16977
##
## Step:  AIC=16421.71
## y ~ duration + nr.employed.cat + poutcome
##
##                 Df Deviance   AIC
## + season         3    15955 15975
## + emp.var.rate   1    16163 16179
## + job            5    16248 16272
## + cons.conf.idx  1    16284 16300
## + education      3    16310 16330
## + default        2    16334 16352
## + contact        1    16362 16378
## + pdays          1    16376 16392
## + campaign       1    16393 16409
## + marital        3    16389 16409
## + day_of_week    4    16391 16413
## <none>                16408 16422
## + age            1    16407 16423
## + loan           2    16407 16425
## + housing        2    16408 16426
##
## Step:  AIC=15974.85
## y ~ duration + nr.employed.cat + poutcome + season
##
##                 Df Deviance   AIC
## + job            5    15869 15899
## + education      3    15901 15927
## + emp.var.rate   1    15914 15936
## + default        2    15912 15936
## + pdays          1    15933 15955
## + contact        1    15934 15956
## + cons.conf.idx  1    15944 15966
## + campaign       1    15944 15966
## + marital        3    15942 15968
## + day_of_week    4    15943 15971
## <none>                15955 15975
## + age            1    15955 15977
## + loan           2    15954 15978
## + housing        2    15955 15979
##
## Step:  AIC=15899.33
## y ~ duration + nr.employed.cat + poutcome + season + job
##
```

```
##                 Df Deviance   AIC
## + emp.var.rate   1    15832 15864
## + default        2    15840 15874
## + pdays          1    15849 15881
## + contact        1    15852 15884
## + education      3    15850 15886
## + campaign       1    15859 15891
## + cons.conf.idx  1    15863 15895
## + day_of_week    4    15857 15895
## + marital        3    15862 15898
## <none>                15869 15899
## + age            1    15869 15901
## + loan           2    15869 15903
## + housing        2    15869 15903
##
## Step:  AIC=15864.1
## y ~ duration + nr.employed.cat + poutcome + season + job + emp.var.rate
##
##                 Df Deviance   AIC
## + default        2    15803 15839
## + pdays          1    15810 15844
## + contact        1    15814 15848
## + education      3    15812 15850
## + campaign       1    15822 15856
## + day_of_week    4    15821 15861
## + cons.conf.idx  1    15827 15861
## + marital        3    15824 15862
## <none>                15832 15864
## + age            1    15831 15865
## + loan           2    15831 15867
## + housing        2    15832 15868
##
## Step:  AIC=15839.35
## y ~ duration + nr.employed.cat + poutcome + season + job + emp.var.rate +
##     default
##
##                 Df Deviance   AIC
## + pdays          1    15782 15820
## + contact        1    15786 15824
## + education      3    15786 15828
## + campaign       1    15794 15832
## + day_of_week    4    15793 15837
## + cons.conf.idx  1    15799 15837
## <none>                15803 15839
## + marital        3    15798 15840
## + age            1    15803 15841
## + loan           2    15803 15843
## + housing        2    15803 15843
##
## Step:  AIC=15819.58
## y ~ duration + nr.employed.cat + poutcome + season + job + emp.var.rate +
##     default + pdays
##
##                 Df Deviance   AIC
```

```
## + contact        1    15765 15805
## + education      3    15764 15808
## + campaign       1    15772 15812
## + day_of_week    4    15771 15817
## + cons.conf.idx  1    15778 15818
## <none>                15782 15820
## + marital        3    15776 15820
## + age            1    15782 15822
## + loan           2    15781 15823
## + housing        2    15782 15824
##
## Step:  AIC=15804.74
## y ~ duration + nr.employed.cat + poutcome + season + job + emp.var.rate +
##     default + pdays + contact
##
##                 Df Deviance   AIC
## + education      3    15748 15794
## + campaign       1    15757 15799
## + day_of_week    4    15754 15802
## + cons.conf.idx  1    15762 15804
## <none>                15765 15805
## + marital        3    15759 15805
## + age            1    15765 15807
## + loan           2    15764 15808
## + housing        2    15765 15809
##
## Step:  AIC=15794.28
## y ~ duration + nr.employed.cat + poutcome + season + job + emp.var.rate +
##     default + pdays + contact + education
##
##                 Df Deviance   AIC
## + campaign       1    15740 15788
## + day_of_week    4    15738 15792
## + cons.conf.idx  1    15746 15794
## <none>                15748 15794
## + marital        3    15744 15796
## + age            1    15748 15796
## + loan           2    15747 15797
## + housing        2    15748 15798
##
## Step:  AIC=15788.25
## y ~ duration + nr.employed.cat + poutcome + season + job + emp.var.rate +
##     default + pdays + contact + education + campaign
##
##                 Df Deviance   AIC
## + day_of_week    4    15730 15786
## + cons.conf.idx  1    15737 15787
## <none>                15740 15788
## + marital        3    15736 15790
## + age            1    15740 15790
## + loan           2    15739 15791
## + housing        2    15740 15792
##
## Step:  AIC=15786.3
```

```
## y ~ duration + nr.employed.cat + poutcome + season + job + emp.var.rate +
##     default + pdays + contact + education + campaign + day_of_week
##
##                 Df Deviance   AIC
## + cons.conf.idx  1    15727 15785
## <none>               15730 15786
## + age            1    15730 15788
## + marital        3    15726 15788
## + loan           2    15729 15789
## + housing        2    15730 15790
##
## Step:  AIC=15784.96
## y ~ duration + nr.employed.cat + poutcome + season + job + emp.var.rate +
##     default + pdays + contact + education + campaign + day_of_week +
##     cons.conf.idx
##
##           Df Deviance   AIC
## <none>          15727 15785
## + age      1    15727 15787
## + marital  3    15723 15787
## + loan     2    15726 15788
## + housing  2    15727 15789
```

```
summary(forward_model)
```

```
##
## Call:
## glm(formula = y ~ duration + nr.employed.cat + poutcome + season +
##     job + emp.var.rate + default + pdays + contact + education +
##     campaign + day_of_week + cons.conf.idx, family = binomial,
##     data = bank_mod)
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -12.641992   0.504591 -25.054  < 2e-16 ***
## duration                      2.157345   0.034706  62.161  < 2e-16 ***
## nr.employed.catLow           -1.563964   0.109884 -14.233  < 2e-16 ***
## nr.employed.catHigh          -1.916608   0.192590  -9.952  < 2e-16 ***
## nr.employed.catVery High     -2.358320   0.246825  -9.555  < 2e-16 ***
## poutcomefailure              -0.574684   0.067559  -8.506  < 2e-16 ***
## poutcomesuccess               0.350736   0.239934   1.462 0.143795
## seasonSpring                  0.017005   0.114055   0.149 0.881480
## seasonSummer                  0.985969   0.105339   9.360  < 2e-16 ***
## seasonWinter                 -0.366645   0.208857  -1.755 0.079177 .
## jobblue-collar               -0.216572   0.076088  -2.846 0.004422 **
## jobno income                  0.297814   0.073429   4.056 5.00e-05 ***
## jobself-employed             -0.107425   0.088560  -1.213 0.225122
## jobservices                  -0.167968   0.086258  -1.947 0.051502 .
## jobtechnician                 0.039997   0.070453   0.568 0.570228
## emp.var.rate                 -0.414622   0.073330  -5.654 1.57e-08 ***
## defaultunknown               -0.312076   0.065838  -4.740 2.14e-06 ***
## defaultyes                   -6.495771 111.372955  -0.058 0.953490
## pdays                        -0.001082   0.000230  -4.703 2.56e-06 ***
## contacttelephone             -0.213301   0.062171  -3.431 0.000602 ***
```

```
## educationhigh.school           0.028371   0.068318   0.415 0.677939
## educationprofessional.course   0.103755   0.083306   1.245 0.212960
## educationuniversity.degree     0.246414   0.069884   3.526 0.000422 ***
## campaign                      -0.031733   0.011998  -2.645 0.008173 **
## day_of_weekmon                -0.099093   0.067994  -1.457 0.145014
## day_of_weekthu                 0.049344   0.065638   0.752 0.452202
## day_of_weektue                 0.073408   0.067447   1.088 0.276431
## day_of_weekwed                 0.088687   0.067275   1.318 0.187409
## cons.conf.idx                 -0.012182   0.006678  -1.824 0.068137 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27032  on 37655  degrees of freedom
## Residual deviance: 15727  on 37627  degrees of freedom
## AIC: 15785
##
## Number of Fisher Scoring iterations: 10
```

```
# Pseudo R^2
r2_forward <- 1 - forward_model$deviance/forward_model$null.deviance
print(r2_forward)
```

```
## [1] 0.4181998
```

Comparing model (m1) and the Final Forward Model (Forward Selection). M1 will be the selected model going forward with an of AIC: 15884 and R^2: 0.4135798

Odds Ratio Table

```
estimates <- coef(m1)
conf_intervals <- confint(m1)
```

```
## Waiting for profiling to be done...
```

```
p_values <- summary(m1)$coefficients[, "Pr(>|z|)"]

odds_bank <- data.frame(Predictor = names(coef(m1)),
                        Odds = estimates,
                        LowerBound = conf_intervals[,1],
                        UpperBound = conf_intervals[,2],
                        pvalue = p_values)

odds_bank[2:4] <- exp(odds_bank[2:4])
print(format(odds_bank, digits = 2, nsmall = 2))
```

```
##                                    Predictor    Odds LowerBound UpperBound
## (Intercept)                      (Intercept) 6.7e-06    4.5e-06    0.00001
```

```
## poutcomefailure                   poutcomefailure 5.9e-01    5.2e-01    0.67085
## poutcomesuccess                   poutcomesuccess 4.0e+00    3.4e+00    4.73373
## contacttelephone                 contacttelephone 7.8e-01    6.9e-01    0.87405
## jobblue-collar                     jobblue-collar 6.7e-01    5.9e-01    0.74830
## jobno income                         jobno income 1.2e+00    1.1e+00    1.37383
## jobself-employed                 jobself-employed 8.7e-01    7.3e-01    1.03043
## jobservices                           jobservices 7.3e-01    6.2e-01    0.85020
## jobtechnician                      jobtechnician 9.9e-01    8.8e-01    1.11630
## seasonSpring                       seasonSpring 6.0e-01    5.2e-01    0.70522
## seasonSummer                       seasonSummer 2.2e+00    1.9e+00    2.64542
## seasonWinter                       seasonWinter 7.7e-01    5.1e-01    1.15073
## duration                               duration 8.5e+00    7.9e+00    9.04993
## nr.employed.catLow             nr.employed.catLow 2.7e-01    2.3e-01    0.31457
## nr.employed.catHigh           nr.employed.catHigh 5.2e-02    4.4e-02    0.06092
## nr.employed.catVery High nr.employed.catVery High 2.1e-02    1.7e-02    0.02529
##                              pvalue
## (Intercept)                 0.0e+00
## poutcomefailure             7.5e-16
## poutcomesuccess             3.8e-63
## contacttelephone            3.4e-05
## jobblue-collar              7.3e-12
## jobno income                6.5e-03
## jobself-employed            1.1e-01
## jobservices                 7.3e-05
## jobtechnician               8.6e-01
## seasonSpring                2.3e-10
## seasonSummer                1.4e-18
## seasonWinter                2.0e-01
## duration                    0.0e+00
## nr.employed.catLow          6.3e-64
## nr.employed.catHigh        3.1e-262
## nr.employed.catVery High    0.0e+00
```

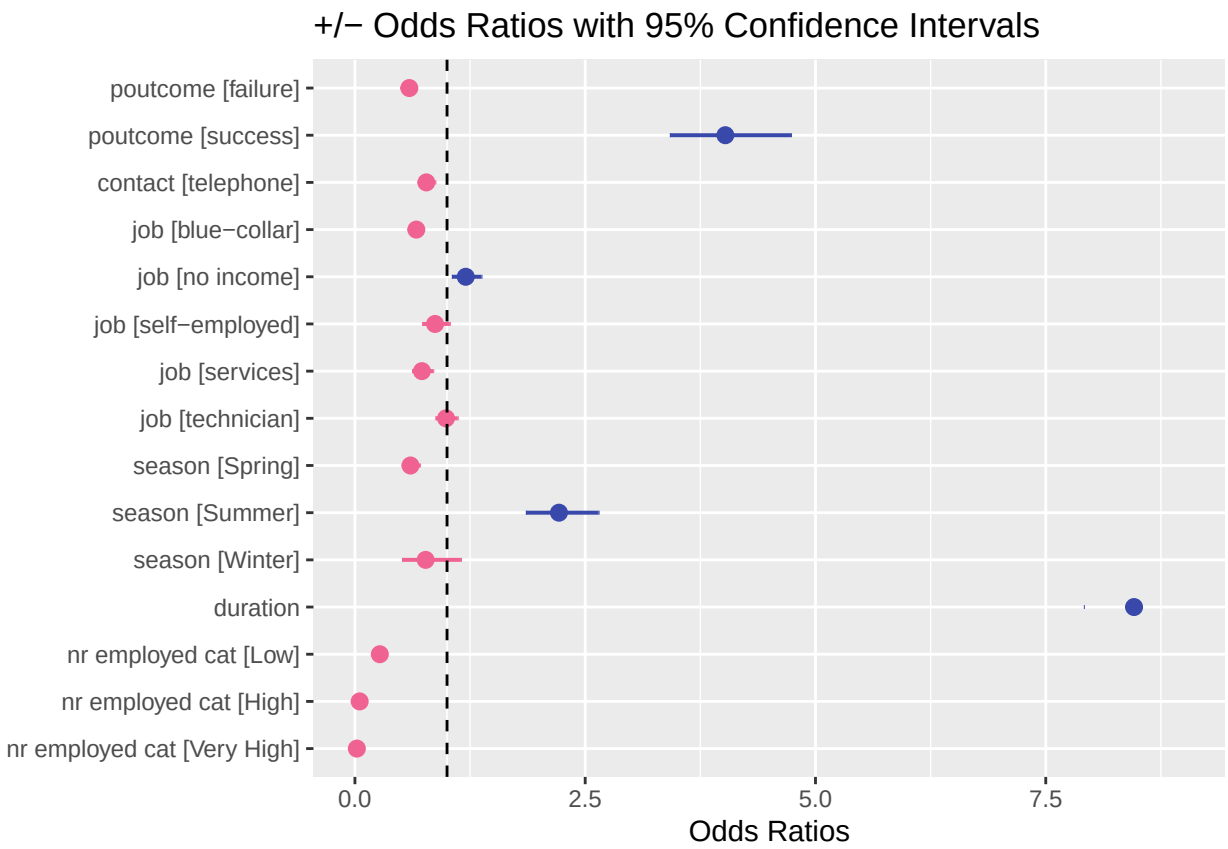
Odds Plot

```
plot_model(model = m1, title = "+/- Odds Ratios with 95% Confidence Intervals",
           colors = c("#f06292", "#3949ab")) +
  scale_y_continuous(limits = c(0, 9)) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "black", size = 0.5)
```

```
## Profiled confidence intervals may take longer time to compute.
##   Use ‘ci_method="wald"‘ for faster computation of CIs.
```

```
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```

```
## 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.
```



Cross Validation

CV First Model

```
set.seed(1)
splitdata <- rsample::initial_split(bank_mod, strata = y, prop = 0.75)
traindata <- rsample::training(splitdata)
testdata <- rsample::testing(splitdata)

traindata$y <- as.factor(traindata$y)
levels(traindata$y) <- make.names(levels(traindata$y))

cv_folds <- trainControl(method = "cv",
                         number = 5,
                         summaryFunction = twoClassSummary,
                         classProbs = TRUE,
                         savePredictions = "final")
model_cv <- train(y ~ job + contact + season + poutcome + nr.employed.cat + duration,
                  data = traindata,
                  method = "glm",
                  family = binomial,
                  trControl = cv_folds,
                  metric = "ROC")
print(model_cv)
```

```
## Generalized Linear Model
##
## 28241 samples
##     6 predictor
##     2 classes: 'X0', 'X1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 22593, 22592, 22593, 22593, 22593
## Resampling results:
##
##   ROC        Sens       Spec
##   0.9239185  0.9718806  0.412696
```

```
model_cv$results
```

```
##   parameter       ROC      Sens     Spec       ROCSD      SensSD     SpecSD
## 1      none 0.9239185 0.9718806 0.412696 0.005497567 0.001887317 0.01535527
```

CV Interaction Model

```
set.seed(1)
model_cv_int <- train(y ~ job + contact + season + poutcome + nr.employed.cat + duration +
                        season:nr.employed.cat + season:duration,
                      data = traindata,
                      method = "glm",
                      family = binomial,
                      trControl = cv_folds,
                      metric = "ROC")
print(model_cv_int)
```

```
## Generalized Linear Model
##
## 28241 samples
##     6 predictor
##     2 classes: 'X0', 'X1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 22592, 22593, 22593, 22593, 22593
## Resampling results:
##
##   ROC       Sens       Spec
##   0.927846  0.9716002  0.4307024
```

```
model_cv_int$results
```

```
##   parameter       ROC      Sens      Spec       ROCSD      SensSD     SpecSD
## 1      none 0.927846 0.9716002 0.4307024 0.003717241 0.001843233 0.01951348
```

CV Forward Model

```
set.seed(1)
model_cv_forward <- train(y ~ duration + nr.employed.cat + poutcome + season +
                            emp.var.rate + job + contact + default + pdays +
                            campaign + education + day_of_week + cons.conf.idx,
                  data = traindata,
                  method = "glm",
                  family = binomial,
                  trControl = cv_folds,
                  metric = "ROC")
print(model_cv_forward)
```

```
## Generalized Linear Model
##
## 28241 samples
##    13 predictor
##     2 classes: 'X0', 'X1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 22592, 22593, 22593, 22593, 22593
## Resampling results:
##
##   ROC        Sens       Spec
##   0.9256217  0.9720008  0.4154394
```

```
model_cv_forward$results
```

```
##   parameter       ROC      Sens      Spec       ROCSD      SensSD     SpecSD
## 1      none 0.9256217 0.9720008 0.4154394 0.003996011 0.002968612 0.01911325
```

K-Fold Comparison

```
kfold_df <- rbind("first model" = model_cv$results, "int model" = model_cv_int$results,
                  "forward model" = model_cv_forward$results)
print(kfold_df)
```

```
##               parameter       ROC      Sens      Spec       ROCSD      SensSD
## first model        none 0.9239185 0.9718806 0.4126960 0.005497567 0.001887317
## int model          none 0.9278460 0.9716002 0.4307024 0.003717241 0.001843233
## forward model      none 0.9256217 0.9720008 0.4154394 0.003996011 0.002968612
##                    SpecSD
## first model   0.01535527
## int model     0.01951348
## forward model 0.01911325
```

Using Set.Seed(1) Best ROC: Interaction Model Best Sens (True Positive): Forward Best Spec (True Negative): Interaction Best ROCSD: Interaction Best SensSD: Interaction Best SpecSD: First

All 3 models had metrics that were very comparative with strong ROC and Sensitivity score. There is a 92% probability that the model will rank a randomly chosen positive case (client subscribes) higher than a randomly chosen negative case (client does not subscribe). For sake of parsimony, we will use the First Model as our selected model of choice.

Confusion Matrix and Accuracy

```
set.seed(1)
testdata$y <- factor(testdata$y, levels = c(0, 1), labels = c("X0", "X1"))

testdata$predicted_prob <- predict(model_cv, newdata = testdata, type = "prob")[,2]
testdata$predicted_class <- ifelse(testdata$predicted_prob > 0.5, "X1", "X0")
testdata$predicted_class <- factor(testdata$predicted_class, levels = c("X0", "X1"))

confusion_matrix <- confusionMatrix(testdata$predicted_class, testdata$y, positive = "X1")
print(confusion_matrix)
```

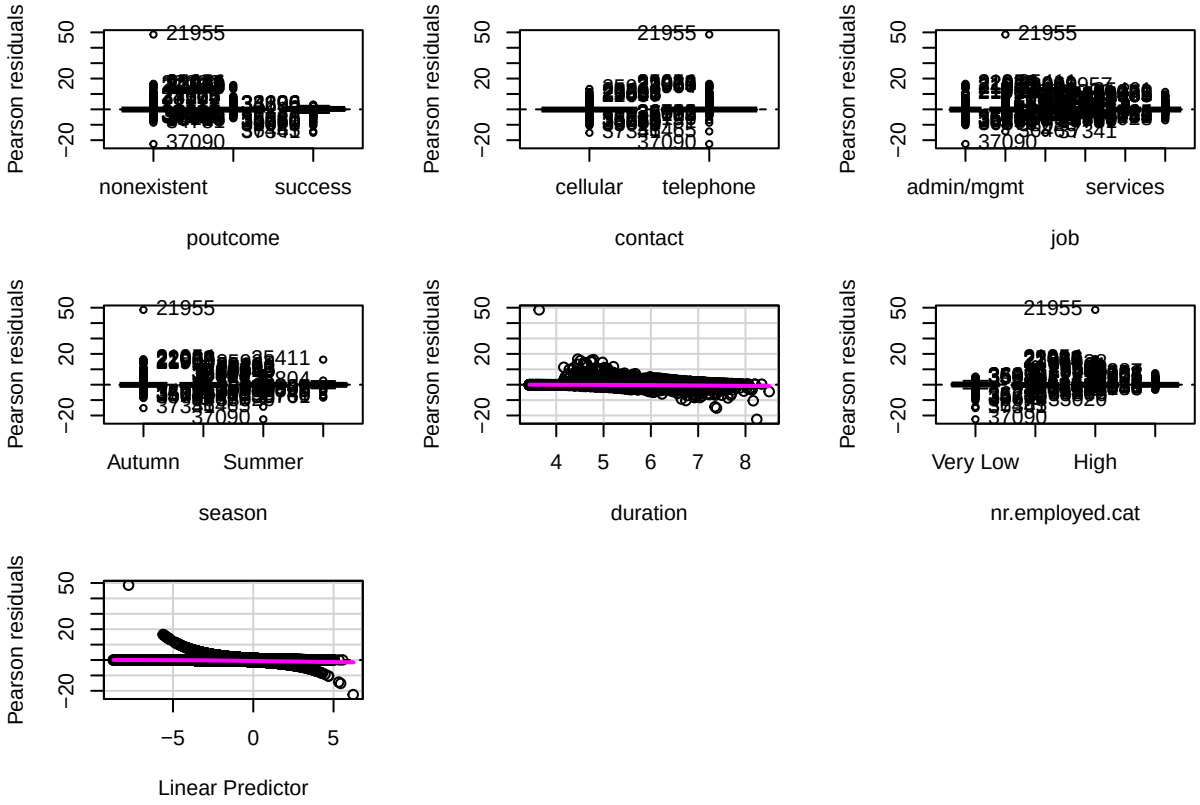
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   X0   X1
##         X0 8100  649
##         X1  222  444
##
##                Accuracy : 0.9075
##                  95% CI : (0.9015, 0.9133)
##     No Information Rate : 0.8839
##     P-Value [Acc > NIR] : 9.769e-14
##
##                   Kappa : 0.4571
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.40622
##             Specificity : 0.97332
##          Pos Pred Value : 0.66667
##          Neg Pred Value : 0.92582
##              Prevalence : 0.11609
##          Detection Rate : 0.04716
##    Detection Prevalence : 0.07074
##       Balanced Accuracy : 0.68977
##
##        'Positive' Class : X1
##
```

The First Model with 6 predictors scored an Accuracy of 90%, this means that 90% of the time, the model correctly predicts whether a bank telemarketing call results in a client subscribing to a term deposit.

The most significant predictors that influence whether a client subscribes to a term deposit is job, type of contact, season, previous outcome, number of employed citizens, and duration of the phone call.

Residual and Influence Plots

```
residualPlots(m1)
```

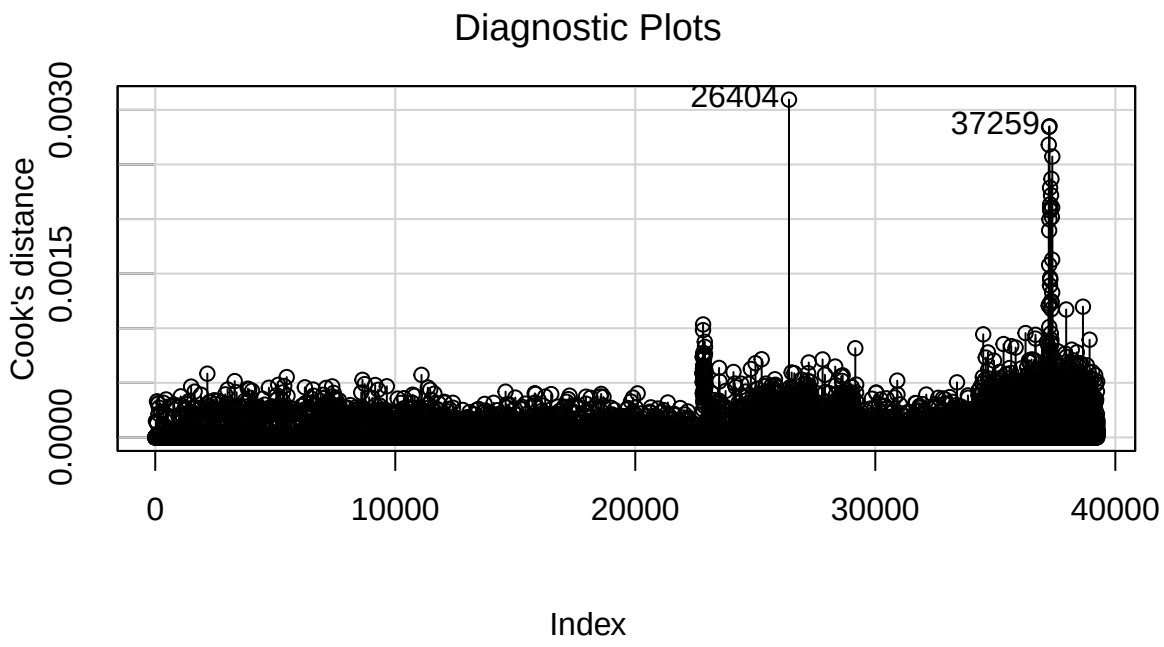



```
##                 Test stat Pr(>|Test stat|)
## poutcome
## contact
## job
## season
## duration            10.29         0.001338 **
## nr.employed.cat
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
influenceIndexPlot(m1, vars = "Cook", id.n=3)
```

```
## Warning in plot.window(...): "id.n" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "id.n" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "id.n" is not a
## graphical parameter
## Warning in axis(side = side, at = at, labels = labels, ...): "id.n" is not a
## graphical parameter
```

```
## Warning in box(...): "id.n" is not a graphical parameter

## Warning in title(...): "id.n" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "id.n" is not a
## graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "id.n" is not a
## graphical parameter
```




```
influenceIndexPlot(m1, vars = "hat", id.n=3)
```

```
## Warning in plot.window(...): "id.n" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "id.n" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "id.n" is not a
## graphical parameter
## Warning in axis(side = side, at = at, labels = labels, ...): "id.n" is not a
## graphical parameter

## Warning in box(...): "id.n" is not a graphical parameter

## Warning in title(...): "id.n" is not a graphical parameter
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "id.n" is not a
## graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "id.n" is not a
## graphical parameter
```