Causal Inference Project - Appendix

Sam Musch, Pardha Pitchikala, Patrick Seng, Sameeksha Aithal, Xiangke Chen

Contents

1 Data Quality Check

Load the data set

```
setwd("C:\\Users\\pardh\\Downloads\\Studies\\3-Spring\\CI\\Project")

data <-read.csv("FINAL_DATA.csv")
```

Missing Values

There are no missing values in our data.

```
sum(is.na(data))
```

```
## [1] 0
```

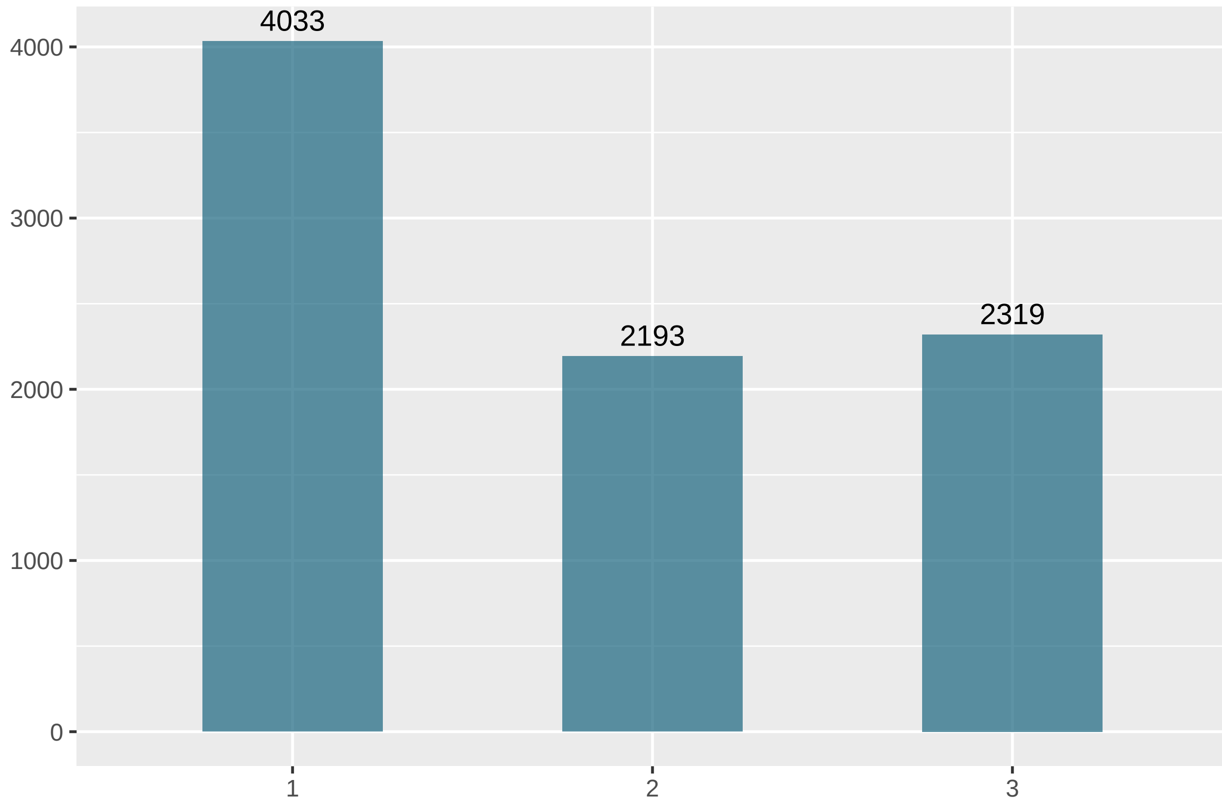
2 Exploratory Data Analysis

2.1 Customer distribution

The number of customers in each campaign are different. If these groups' attributes are different, we might need to consider using **Matching techniques** to make sure that independant variables are balanced across campaigns.

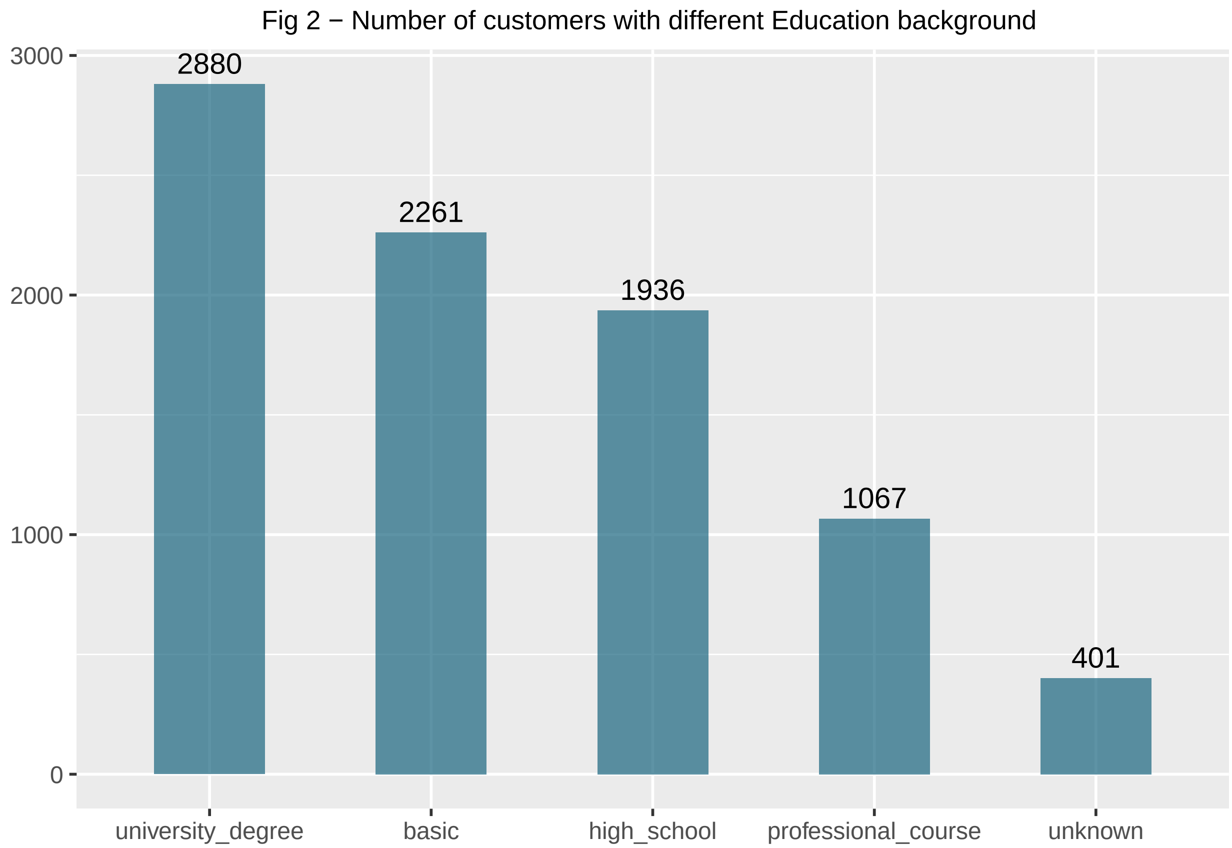
```
data_1 <- data %>% group_by(campaign) %>%
            summarise(num_customers = n()) %>%
            ungroup()

ggplot(data_1, aes(x = factor(campaign), y = num_customers)) +
        geom_bar(stat = "identity" , width=0.5, fill = rgb(0.1,0.4,0.5,0.7)) +
        ggtitle("Fig 1 - Number of customers targeted for each campaign") +
        geom_text(aes(label = num_customers,vjust = -0.5)) +
        theme(plot.title = element_text(color="black", size=10, hjust = 0.5)) +
        theme(axis.title.y = element_blank(), axis.title.x = element_blank() )
```

Fig 1 − Number of customers targeted for each campaign



2.2 Distrbution of customers with different education background

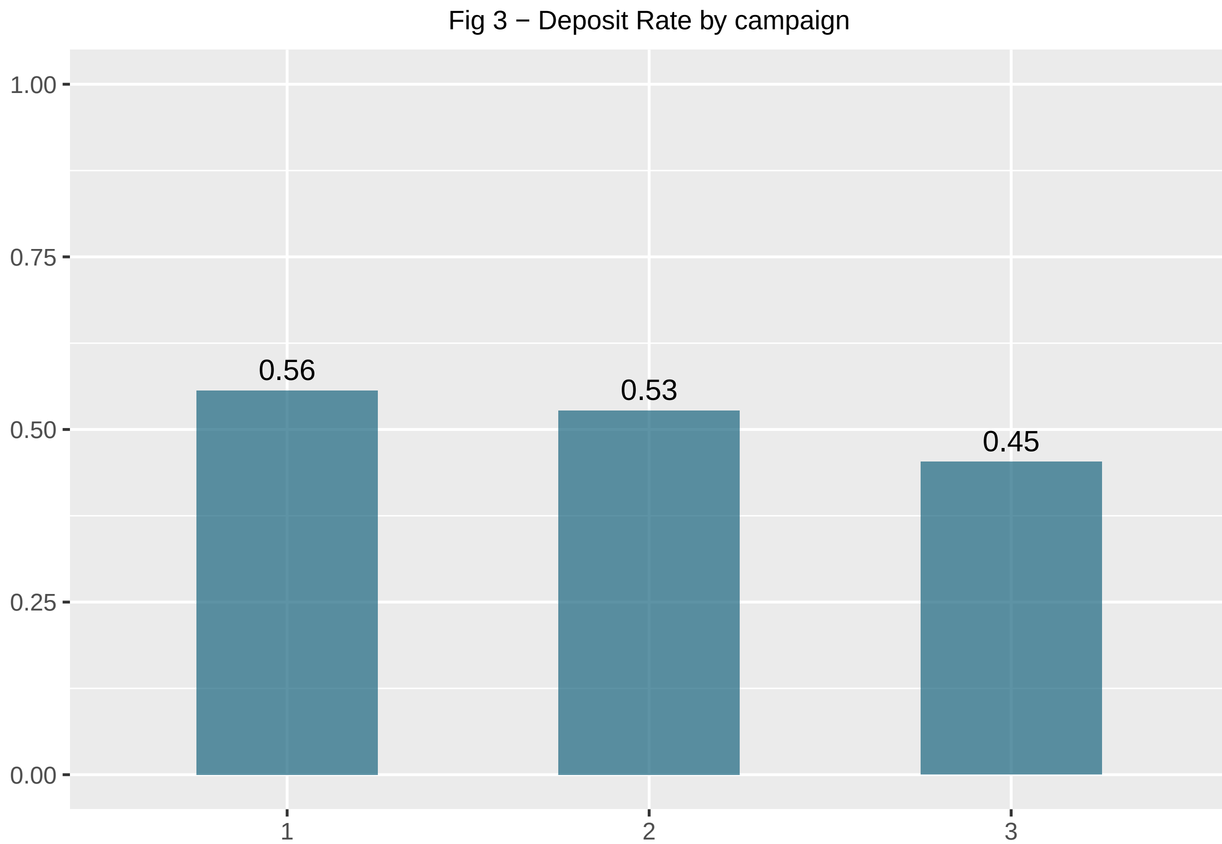
```
data %>% group_by(education) %>%
  summarise(fre = n()) %>%
  arrange(desc(fre)) %>%
  mutate(education=factor(education, levels=education)) %>%
  ggplot(aes(x = factor(education),width=0.5, y = fre)) +
  ggtitle("Fig 2 - Number of customers with different Education background ") +
  geom_bar( stat = "identity",  fill = rgb(0.1,0.4,0.5,0.7)) +
  theme(plot.title = element_text(color="black", size=10, hjust = 0.5)) +
  geom_text(aes(label = round(fre,2),vjust = -0.5)) +
   theme(axis.title.y = element_blank(), axis.title.x = element_blank() )
```



2.3 Deposit rate across campaigns

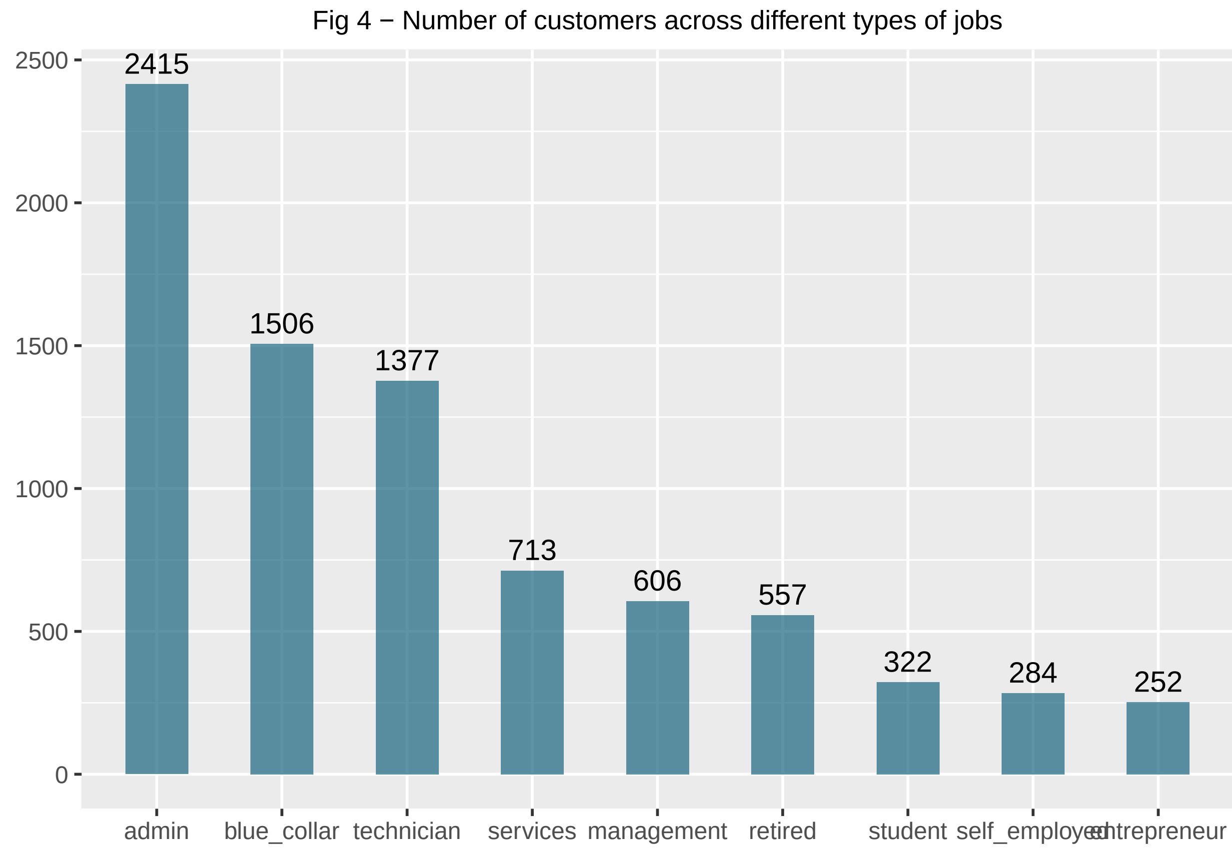
```
# generate data for the plot
group <- data %>% group_by(campaign) %>%
  summarise(succ = sum(y),
            ttl = n(),
            ratio = succ / ttl) %>%
  select(-succ,-ttl)
```

```
ggplot(group, aes(x = factor(campaign), y = ratio)) +
  geom_bar(stat = "identity", width = 0.5, position = "dodge",
           fill = rgb(0.1,0.4,0.5,0.7)) +
  ggtitle('Fig 3 - Deposit Rate by campaign') +
  xlab("Campaign") +
  ylab('Success Rate') +
  theme(plot.title = element_text(color="black", size=10, hjust = 0.5)) +
  ylim(0,1) +
  geom_text(aes(label = round(ratio,2),vjust = -0.5)) +
  theme(axis.title.y = element_blank(), axis.title.x = element_blank())
```



2.4 Distrbution of customers across different types of Jobs

```
data %>% group_by(job) %>%
  summarise(fre = n()) %>%
  filter(fre > 250) %>%
  arrange(desc(fre)) %>%
  mutate(job=factor(job, levels=job)) %>%
  ggplot(aes(x = factor(job),width=0.5, y = fre)) +
  geom_bar( stat = "identity",  fill = rgb(0.1,0.4,0.5,0.7)) +
  ggtitle('Fig 4 - Number of customers across different types of jobs') +
  theme(plot.title = element_text(color="black", size=10, hjust = 0.5)) +
  geom_text(aes(label = round(fre,2),vjust = -0.5)) +
  theme(axis.title.y = element_blank(), axis.title.x = element_blank())
```



2.5 Regression before matching

We are running a simple regression to check if different groups of customers behave differently across groups.

```
# set campaing 3 as base level

data_1 = data %>% select(-cons.conf.idx, -DATE)
data_1$campaign = as.factor(data_1$campaign)

# filter data for specific campaigns
data_c1 = data_1 %>% filter(campaign != 2)
data_c2= data_1 %>% filter(campaign != 1)

# Simple regression: C1 vs C3
data_c1 <- within(data_c1, campaign <- relevel(campaign, ref = 3))
mod1 = glm(y ~ campaign +                                 # Treatment vs. Control
             age + marital + housing + job + education +  # Demographic Factors
             previous + poutcome + default,               # Financial Factors
             data = data_c1, family = 'binomial')
summary(mod1)
```

```
##
## Call:
## glm(formula = y ~ campaign + age + marital + housing + job +
##     education + previous + poutcome + default, family = "binomial",
```

```
##     data = data_c1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6648  -1.0735   0.1945   1.1625   1.8184
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -1.215286   0.267799  -4.538 5.68e-06 ***
## campaign1                     0.279360   0.056576   4.938 7.90e-07 ***
## age                           0.004490   0.003228   1.391  0.16429
## maritalmarried                0.062016   0.091081   0.681  0.49594
## maritalsingle                 0.320914   0.102916   3.118  0.00182 **
## maritalunknown                0.504068   0.546262   0.923  0.35613
## housingunknown                0.014061   0.188466   0.075  0.94053
## housingyes                   -0.019629   0.055516  -0.354  0.72366
## jobblue_collar               -0.213283   0.104687  -2.037  0.04162 *
## jobentrepreneur               0.046326   0.164483   0.282  0.77821
## jobhousemaid                 -0.013751   0.189199  -0.073  0.94206
## jobmanagement                -0.027048   0.116562  -0.232  0.81650
## jobretired                    1.048225   0.158654   6.607 3.92e-11 ***
## jobself_employed             -0.039164   0.151746  -0.258  0.79634
## jobservices                  -0.133284   0.112445  -1.185  0.23589
## jobstudent                    1.266364   0.200325   6.322 2.59e-10 ***
## jobtechnician                -0.072949   0.092510  -0.789  0.43038
## jobunemployed                 0.247568   0.183407   1.350  0.17707
## jobunknown                    0.171060   0.318346   0.537  0.59103
## educationhigh_school          0.027687   0.094006   0.295  0.76836
## educationprofessional_course  0.116760   0.112839   1.035  0.30079
## educationuniversity_degree    0.192558   0.095407   2.018  0.04356 *
## educationunknown              0.234774   0.147673   1.590  0.11187
## previous                      0.687367   0.143889   4.777 1.78e-06 ***
## poutcomenonexistent           0.540628   0.186811   2.894  0.00380 **
## poutcomesuccess               2.385267   0.191592  12.450  < 2e-16 ***
## defaultunknown               -0.701135   0.081746  -8.577  < 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: 8796.8  on 6351  degrees of freedom
## Residual deviance: 7707.9  on 6325  degrees of freedom
## AIC: 7761.9
##
## Number of Fisher Scoring iterations: 5
```

```r
# Simple regression: C2 vs C3
data_c2 <- within(data_c2, campaign <- relevel(campaign, ref = 3))
mod2 = glm(y ~ campaign +                                 # Treatment vs. Control
             age + marital + housing + job + education +  # Demographic Factors
             previous + poutcome + default,               # Financial Factors
             data = data_c2, family = 'binomial')
summary(mod2)
```

```
##
## Call:
## glm(formula = y ~ campaign + age + marital + housing + job +
##     education + previous + poutcome + default, family = "binomial",
##     data = data_c2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8558  -1.0633  -0.6729   1.1977   1.8818
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -0.741696   0.291765  -2.542   0.0110 *
## campaign2                     0.178316   0.064634   2.759   0.0058 **
## age                           0.005475   0.003795   1.443   0.1491
## maritalmarried               -0.054243   0.108973  -0.498   0.6187
## maritalsingle                 0.066529   0.122214   0.544   0.5862
## maritalunknown               -0.221241   0.907187  -0.244   0.8073
## housingunknown               -0.182798   0.219638  -0.832   0.4053
## housingyes                    0.095318   0.065489   1.455   0.1455
## jobblue_collar               -0.210969   0.123199  -1.712   0.0868 .
## jobentrepreneur              -0.299437   0.193985  -1.544   0.1227
## jobhousemaid                 -0.310321   0.218895  -1.418   0.1563
## jobmanagement                -0.229816   0.136718  -1.681   0.0928 .
## jobretired                    0.937602   0.186716   5.022 5.13e-07 ***
## jobself_employed              0.105011   0.181548   0.578   0.5630
## jobservices                  -0.034963   0.127987  -0.273   0.7847
## jobstudent                    1.364817   0.230589   5.919 3.24e-09 ***
## jobtechnician                -0.062297   0.111365  -0.559   0.5759
## jobunemployed                 0.324431   0.215110   1.508   0.1315
## jobunknown                   -0.671852   0.449423  -1.495   0.1349
## educationhigh_school         -0.017892   0.110775  -0.162   0.8717
## educationprofessional_course  0.043602   0.133046   0.328   0.7431
## educationuniversity_degree    0.177769   0.110961   1.602   0.1091
## educationunknown              0.115629   0.178156   0.649   0.5163
## previous                      0.352270   0.137993   2.553   0.0107 *
## poutcomenonexistent           0.197921   0.192266   1.029   0.3033
## poutcomesuccess               2.096314   0.197849  10.596  < 2e-16 ***
## defaultunknown               -0.793109   0.094526  -8.390  < 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: 6252.9  on 4511  degrees of freedom
## Residual deviance: 5557.4  on 4485  degrees of freedom
## AIC: 5611.4
##
## Number of Fisher Scoring iterations: 5
```

3 Matching

Below we are performing some data transformations for doing regression and matching.

```r
# create dummy columns for categorical variables
bank_dummy_p <- dummy_cols(data %>% select(-c("DATE", "cons.conf.idx")) )

# deselect the unnecessary columns
bank_dummy_req_cols <- bank_dummy_p %>%
                            select(-c('job',    'marital',  'education',
                                      'default',    'housing',  'loan',
                                      'month',  'poutcome',
                                      'job_self_employed','marital_unknown',
                                      "default_unknown",
                                      "loan_unknown", "loan_yes", "month_sep",
                                      "poutcome_success"))


# create a data frame for campaings 1 and 3
data_13 <- bank_dummy_req_cols %>%
                      filter(campaign %in% c(1,3)) %>%
                      mutate(campaign = ifelse(campaign == 3, 0, 1))

# create a data frame for campaings 2 and 3
data_23 <- bank_dummy_req_cols %>%
                      filter(campaign %in% c(2,3)) %>%
                      mutate(campaign = ifelse(campaign == 3, 0, 1))
```

3.1 Campaigns 1 vs 3

Below we are trying Propensity Score matching for camapings 1 and 3

```r
match_13 <- matchit(campaign ~ age +  previous +  job_admin +
                      job_technician +  job_management +
                      job_blue_collar +  job_retired +  job_services +
                      job_student + job_unknown +  job_housemaid +
                      job_unemployed +  job_entrepreneur +
                      marital_divorced +  marital_married +  marital_single +
                      education_university_degree +
                      education_professional_course +  education_high_school +
                      education_unknown +  education_basic +  default_no +
                      housing_no +  housing_unknown +  housing_yes +
                      poutcome_failure +
                      poutcome_nonexistent,
                      data = data_13, ratio = 1, method = "nearest",
                      replace = TRUE ,distance = "logit", calliper = 0.001)

# Extract the matched data from matching output (m.out_2)
matched_data_13 <- match.data(match_13)
```

Below output shows how much balance we achieved through PSM

```r
# number of records matched
summary(match_13)[["nn"]]
```

```
##             Control Treated
```

```
## All          2319    4033
## Matched      1599    4033
## Unmatched     720       0
## Discarded       0       0
```

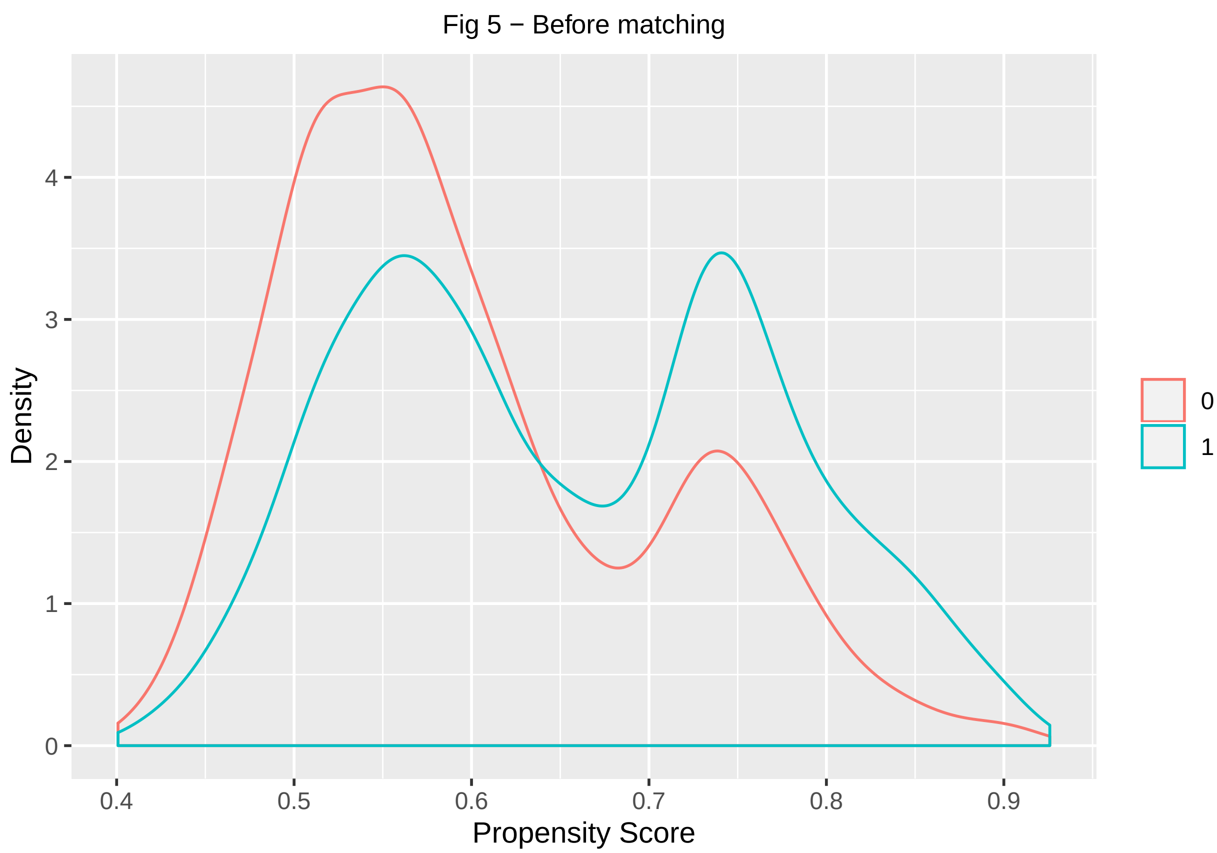
```r
# select list of Covariates to perform T test on data_23 before matching
list_of_covariates <- c('age', 'previous', 'job_admin', 'job_technician',
                        'job_management', 'job_blue_collar', 'job_retired',
                        'job_services', 'job_student', 'job_unknown',
                        'job_housemaid', 'job_unemployed', 'job_entrepreneur',
                        'marital_divorced', 'marital_married', 'marital_single',
                        'education_university_degree',
                        'education_professional_course', 'education_high_school',
                        'education_unknown', 'education_basic', 'default_no',
                        'housing_no', 'housing_unknown', 'housing_yes', 'loan_no',
                        'month_apr', 'month_aug',
                        'month_dec', 'month_jul', 'month_jun', 'month_mar',
                        'month_may', 'month_nov', 'month_oct', 'poutcome_failure',
                        'poutcome_nonexistent')

# Create a dummy DataFrame to store the results of T test of data_13 before matching
t_test_13_before <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(t_test_13_before) <- c('Covariate', 'p_value_before')

# Run T test for each selected covariate
for (covariate in list_of_covariates){
  t <- t.test(data_13[,covariate] ~ campaign, data = data_13 )
  t_test_13_before[nrow(t_test_13_before) + 1,] = c(covariate,round(t[["p.value"]],2))
}


# Create a dummy DataFrame to store the results of T test of data_13 after matching
t_test_13_after <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(t_test_13_after) <- c('Covariate', 'p_value_after')

# Run T test for each selected covariate
for (covariate in list_of_covariates){
  t <- t.test(matched_data_13[,covariate] ~ campaign, data = matched_data_13 )
  t_test_13_after[nrow(t_test_13_after) + 1,] = c(covariate,round(t[["p.value"]],2))
}

merge_data_13 <- t_test_13_before %>%
              inner_join(t_test_13_after, c("Covariate" = "Covariate")) %>%
              filter( (p_value_before <= 0.2) & (p_value_after != 0) )

print(merge_data_13)
```

```
##           Covariate p_value_before p_value_after
## 1       job_retired           0.01          0.16
## 2      job_services           0.01          0.12
## 3       job_student           0.15          0.32
## 4   education_basic           0.13          0.11
## 5        default_no              0          0.03
## 6         month_mar           0.07          0.43
```

```
## 7 poutcome_failure              0          0.03
```

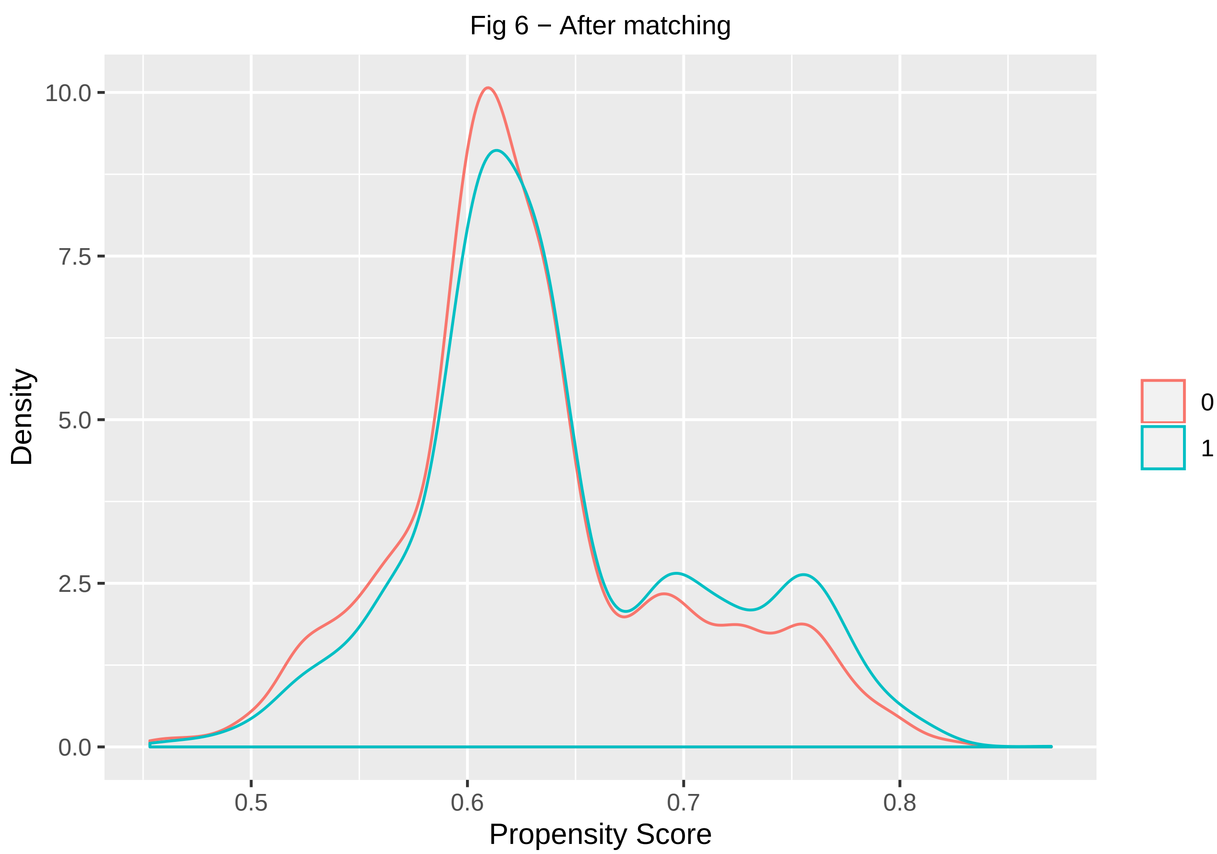
Plot densities of propensity scores for camapaigns 1 and 3 before matching

```r
# plot before matching
pscore_before_13 = glm(campaign ~ ., data = data_13, family = "binomial" )$fitted.values
data_13$pscore <- pscore_before_13

ggplot(data_13, aes(x = pscore, col = factor(campaign))) +
  geom_density() +
  xlab("Propensity Score") + ylab("Density") +
  ggtitle("Fig 5 - Before matching") +
  theme(legend.title = element_blank()) +
  theme(plot.title = element_text(color="black", size=10, hjust = 0.5))
```



Plot densities of propensity scores for camapaigns 1 and 3 after matching

```r
# plot after matching
ggplot(matched_data_13, aes(x = distance, col = factor(campaign))) +
  geom_density() +
  xlab("Propensity Score") + ylab("Density") +
  ggtitle("Fig 6 - After matching") +
  theme(legend.title = element_blank()) +
  theme(plot.title = element_text(color="black", size=10, hjust = 0.5))
```



Regression after matching

```
model_13_after <- glm( y ~ ., data = matched_data_13 %>%
                                select(-c( "distance", "month_apr", "month_aug",
                                          "month_dec",
                                          "month_jul", "month_jun",
                                          "month_mar","month_may",
                                          "month_nov", "month_oct", "loan_no"))
                              , family = 'binomial')

summary(model_13_after)
```

```
##
## Call:
## glm(formula = y ~ ., family = "binomial", data = matched_data_13 %>%
##     select(-c("distance", "month_apr", "month_aug", "month_dec",
##         "month_jul", "month_jun", "month_mar", "month_may", "month_nov",
##         "month_oct", "loan_no")))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6638  -1.1006   0.2468   1.1462   1.8176
##
## Coefficients: (2 not defined because of singularities)
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      1.499602   0.731951   2.049  0.04048 *
```

```
## age                             0.004947   0.003470   1.425  0.15405
## campaign                        0.187980   0.064146   2.931  0.00338 **
## previous                        0.684004   0.145403   4.704 2.55e-06 ***
## job_admin                       0.025824   0.163836   0.158  0.87476
## job_blue_collar                -0.254157   0.177252  -1.434  0.15161
## job_entrepreneur                0.017732   0.225772   0.079  0.93740
## job_housemaid                   0.017008   0.253068   0.067  0.94642
## job_management                 -0.014791   0.189545  -0.078  0.93780
## job_retired                     1.043531   0.221180   4.718 2.38e-06 ***
## job_services                   -0.123481   0.189538  -0.651  0.51473
## job_student                     1.281649   0.260412   4.922 8.58e-07 ***
## job_technician                 -0.094623   0.172649  -0.548  0.58364
## job_unemployed                  0.237145   0.243662   0.973  0.33043
## job_unknown                     0.172670   0.380526   0.454  0.65000
## marital_divorced               -0.849989   0.623200  -1.364  0.17260
## marital_married                -0.785851   0.617552  -1.273  0.20319
## marital_single                 -0.535781   0.618842  -0.866  0.38661
## education_basic                -0.277957   0.159674  -1.741  0.08172 .
## education_high_school          -0.254428   0.159555  -1.595  0.11080
## education_professional_course -0.167603   0.174744  -0.959  0.33749
## education_university_degree   -0.114689   0.159332  -0.720  0.47164
## education_unknown                     NA         NA      NA       NA
## default_no                      0.700554   0.089389   7.837 4.61e-15 ***
## housing_no                      0.042737   0.059072   0.723  0.46939
## housing_unknown                 0.143992   0.205812   0.700  0.48416
## housing_yes                           NA         NA      NA       NA
## poutcome_failure               -2.411223   0.197049 -12.237  < 2e-16 ***
## poutcome_nonexistent           -1.813432   0.262171  -6.917 4.61e-12 ***
## weights                         0.128601   0.075197   1.710  0.08723 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7777.1  on 5631  degrees of freedom
## Residual deviance: 6800.6  on 5604  degrees of freedom
## AIC: 6856.6
##
## Number of Fisher Scoring iterations: 5
```

3.2 Campaigns 2 vs 3

Below we are trying Propensity Score matching for camapings 2 and 3

```r
match_23 <- matchit(campaign ~ age +  previous +  job_admin +
                      job_technician +  job_management +
                      job_blue_collar +  job_retired +  job_services +
                      job_student + job_unknown +  job_housemaid +
                      job_unemployed +  job_entrepreneur +
                      marital_divorced +  marital_married +
                      marital_single +  education_university_degree +
                      education_professional_course +  education_high_school +
                      education_unknown +  education_basic +  default_no +
```

```r
                      housing_no +  housing_unknown +  housing_yes +
                      poutcome_failure +  poutcome_nonexistent,
                      data = data_23, ratio = 1, method = "nearest",
                      replace = TRUE ,distance = "logit", calliper = 0.02)

# Extract the matched data from matching output (m.out_2)
matched_data_23 <- match.data(match_23)
```

Below output shows how much balance we achieved through PSM

```r
# number of records matched
summary(match_23)[["nn"]]
```

```
##           Control Treated
## All          2319    2193
## Matched      1180    2193
## Unmatched    1139       0
## Discarded       0       0
```

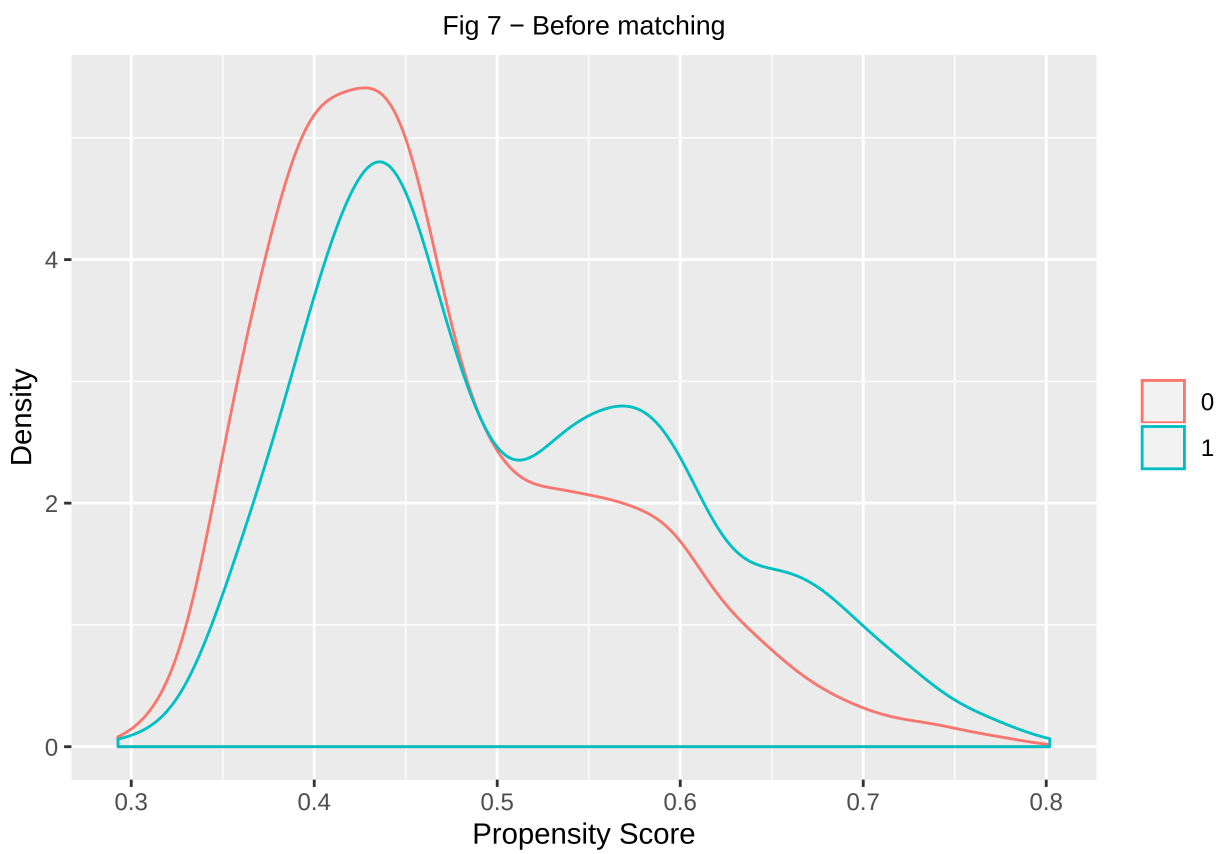
```r
# Create a dummy DataFrame to store the results of T test of data_13 before matching
t_test_23_before <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(t_test_23_before) <- c('Covariate', 'p_value_before')

# Run T test for each selected covariate
for (covariate in list_of_covariates){
  t <- t.test(data_23[,covariate] ~ campaign, data = data_23 )
  t_test_23_before[nrow(t_test_23_before) + 1,] = c(covariate,round(t[["p.value"]],2))
}


# Create a dummy DataFrame to store the results of T test of data_13 after matching
t_test_23_after <- data.frame(matrix(ncol = 2, nrow = 0))
colnames(t_test_23_after) <- c('Covariate', 'p_value_after')

# Run T test for each selected covariate
for (covariate in list_of_covariates){
  t <- t.test(matched_data_23[,covariate] ~ campaign, data = matched_data_23 )
  t_test_23_after[nrow(t_test_23_after) + 1,] = c(covariate,round(t[["p.value"]],2))
}

merge_data_23 <- t_test_23_before %>%
              inner_join(t_test_23_after, c("Covariate" = "Covariate")) %>%
              filter( (p_value_before <= 0.2) & (p_value_after != 0) )

print(merge_data_23)
```

```
##                    Covariate p_value_before p_value_after
## 1                  job_admin            0.1          0.29
## 2             job_management           0.09          0.42
## 3                job_retired            0.1          0.01
## 4                job_student           0.14          0.73
## 5           marital_divorced           0.15          0.13
```

```
## 6  education_university_degree           0.2          0.98
## 7        education_high_school           0.12          0.63
## 8                   default_no              0          0.94
## 9                    month_aug              0          0.02
## 10                   month_jun              0          0.12
## 11                   month_mar           0.18          0.79
## 12                   month_may           0.01          0.04
## 13            poutcome_failure              0          0.06
```

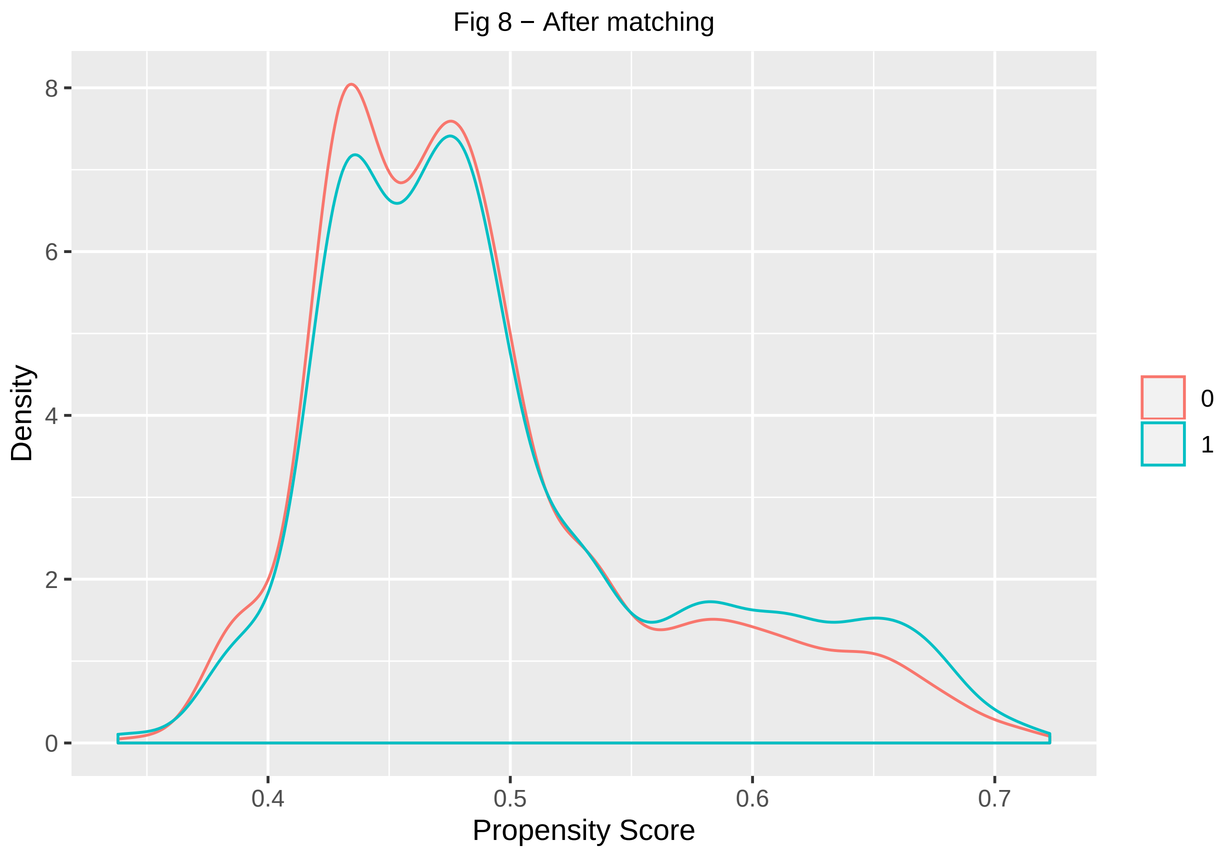
Plot densities of propensity scores for camapaigns 1 and 3 before matching

```r
# plot before matching
pscore_before_23 = glm(campaign ~ ., data = data_23,
                       family = "binomial" )$fitted.values
data_23$pscore <- pscore_before_23

ggplot(data_23, aes(x = pscore, col = factor(campaign))) +
  geom_density() +
  xlab("Propensity Score") + ylab("Density") +
  ggtitle("Fig 7 - Before matching") +
  theme(legend.title = element_blank()) +
  theme(plot.title = element_text(color="black", size=10, hjust = 0.5))
```



Plot densities of propensity scores for camapaigns 1 and 3 after matching

```r
# plot after matching
ggplot(matched_data_23, aes(x = distance, col = factor(campaign))) +
  geom_density() +
  xlab("Propensity Score") + ylab("Density") +
  ggtitle("Fig 8 - After matching") +
  theme(legend.title = element_blank()) +
  theme(plot.title = element_text(color="black", size=10, hjust = 0.5))
```



Regression after matching

```r
model_23_after <- glm( y ~ ., data = matched_data_23 %>%
                              select(-c( "distance", "month_apr", "month_aug",
                                         "month_dec",
                                         "month_jul", "month_jun",
                                         "month_mar","month_may",
                                         "month_nov", "month_oct", "loan_no"))
                             , family = 'binomial')

summary(model_23_after)
```

```
##
## Call:
## glm(formula = y ~ ., family = "binomial", data = matched_data_23 %>%
##     select(-c("distance", "month_apr", "month_aug", "month_dec",
##         "month_jul", "month_jun", "month_mar", "month_may", "month_nov",
```

```
##         "month_oct", "loan_no")))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7778  -1.0716   0.2439   1.1566   1.8760
##
## Coefficients: (2 not defined because of singularities)
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -0.647532   1.246636  -0.519  0.60347
## age                           0.004817   0.004441   1.085  0.27803
## campaign                      0.116517   0.078529   1.484  0.13788
## previous                      0.322437   0.150177   2.147  0.03179 *
## job_admin                    -0.229757   0.225775  -1.018  0.30885
## job_blue_collar              -0.469645   0.243971  -1.925  0.05423 .
## job_entrepreneur             -0.534093   0.295586  -1.807  0.07078 .
## job_housemaid                -0.524030   0.317209  -1.652  0.09853 .
## job_management               -0.523886   0.253552  -2.066  0.03881 *
## job_retired                   0.638157   0.296907   2.149  0.03161 *
## job_services                 -0.301506   0.255954  -1.178  0.23881
## job_student                   1.244034   0.335625   3.707  0.00021 ***
## job_technician               -0.261865   0.237444  -1.103  0.27009
## job_unemployed                0.217006   0.312703   0.694  0.48770
## job_unknown                  -1.396095   0.602939  -2.315  0.02059 *
## marital_divorced              1.530368   1.166589   1.312  0.18958
## marital_married               1.403504   1.160559   1.209  0.22653
## marital_single                1.450633   1.160731   1.250  0.21139
## education_basic              -0.252035   0.216878  -1.162  0.24519
## education_high_school        -0.304499   0.218140  -1.396  0.16275
## education_professional_course -0.175770   0.234957  -0.748  0.45440
## education_university_degree   -0.016306   0.216028  -0.075  0.93983
## education_unknown                    NA         NA      NA       NA
## default_no                    0.866064   0.115189   7.519 5.53e-14 ***
## housing_no                   -0.061185   0.076407  -0.801  0.42326
## housing_unknown              -0.157658   0.255416  -0.617  0.53706
## housing_yes                          NA         NA      NA       NA
## poutcome_failure             -2.039849   0.209097  -9.756  < 2e-16 ***
## poutcome_nonexistent         -1.821476   0.281329  -6.475 9.51e-11 ***
## weights                       0.274295   0.100513   2.729  0.00635 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4674.3  on 3372  degrees of freedom
## Residual deviance: 4104.0  on 3345  degrees of freedom
## AIC: 4160
##
## Number of Fisher Scoring iterations: 5
```

3.3 Heterogeneity between campaigns 1 and 3 after matching the data

```r
# create a row number
matched_data_13_r <- matched_data_13 %>% mutate(id = row_number())

# converting education dummy columns in to a single column
edu_list <- c("id","education_basic", "education_high_school",
              "education_professional_course",
              "education_university_degree", "education_unknown")


matched_data_13_edu <- matched_data_13_r %>% select(edu_list) %>%
                        gather("education", "value", -id ) %>%
                        filter(value == 1) %>%
                        select(-value) %>%
                        mutate(education = str_replace(education,"education_",""))

# converting Job dummy columns in to a single column
job_list <- c("id","job_admin", "job_blue_collar", "job_entrepreneur",
              "job_housemaid", "job_management", "job_retired",
              "job_services", "job_student", "job_technician",
              "job_unemployed", "job_unknown")

matched_data_13_job <- matched_data_13_r %>% select(job_list) %>%
                        gather("job", "value", -id ) %>%
                        filter(value == 1) %>%
                        select(-value) %>%
                        mutate(job = str_replace(job,"job_",""))

# converting marital status dummy columns in to a single column
mar_status <- c("id","marital_divorced", "marital_married", "marital_single")

matched_data_13_mar <- matched_data_13_r %>% select(mar_status) %>%
                        gather("marital_status", "value", -id ) %>%
                        filter(value == 1) %>%
                        select(-value) %>%
                        mutate(marital_status = str_replace(marital_status,"marital_",""))

# converting Housing  dummy columns in to a single column
hous_list <- c("id","housing_no", "housing_yes", "housing_unknown")

matched_data_13_hous <- matched_data_13_r %>% select(hous_list) %>%
                        gather("housing", "value", -id ) %>%
                        filter(value == 1) %>%
                        select(-value) %>%
                        mutate(housing = str_replace(housing,"housing_",""))

# converting poutcome  dummy columns in to a single column
pout_list <- c("id","poutcome_failure", "poutcome_nonexistent")

matched_data_13_pout <- matched_data_13_r %>% select(pout_list) %>%
                        gather("poutcome", "value", -id ) %>%
                        filter(value == 1) %>%
                        select(-value) %>%
                        mutate(poutcome = str_replace(poutcome,"poutcome_",""))
```

```r
# Merge all data frames into a single data frame
merged_het_13_after <- matched_data_13_r %>%
                    select("id","y" ,"age", "campaign", "previous",
                           "default_no", "weights") %>%
                    left_join(matched_data_13_edu, c("id" = "id")) %>%
                    left_join(matched_data_13_job, c("id" = "id")) %>%
                    left_join(matched_data_13_mar, c("id" = "id")) %>%
                    left_join(matched_data_13_hous, c("id" = "id")) %>%
                    left_join(matched_data_13_pout, c("id" = "id"))

# C1 vs C3 heterogeneity

merged_het_13_after$job <- as.factor(merged_het_13_after$job)
merged_het_13_after <- within(merged_het_13_after, job <- relevel(job, ref = 2))

# check if any ionteraction terms are significant
mod_het_13_after = glm(y ~ campaign +
                           age + marital_status + housing + job + education +
                           previous + poutcome + default_no + job*campaign,
                           data = merged_het_13_after, family = 'binomial')

summary(mod_het_13_after)
```

```
##
## Call:
## glm(formula = y ~ campaign + age + marital_status + housing +
##     job + education + previous + poutcome + default_no + job *
##     campaign, family = "binomial", data = merged_het_13_after)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4452  -1.1107  -0.7399   1.1652   1.7849
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -1.916738   0.316750  -6.051 1.44e-09 ***
## campaign                     -0.003773   0.146672  -0.026 0.979476
## age                           0.005352   0.003603   1.485 0.137416
## marital_statusmarried         0.078348   0.101301   0.773 0.439278
## marital_statussingle          0.316556   0.114757   2.758 0.005807 **
## housingunknown                0.102736   0.213475   0.481 0.630334
## housingyes                   -0.039276   0.061101  -0.643 0.520351
## jobadmin                      0.144313   0.175187   0.824 0.410073
## jobentrepreneur              -0.116174   0.352942  -0.329 0.742036
## jobhousemaid                 -0.036016   0.427500  -0.084 0.932859
## jobmanagement                 0.068362   0.259915   0.263 0.792537
## jobretired                    1.225948   0.284617   4.307 1.65e-05 ***
## jobservices                   0.229228   0.223875   1.024 0.305877
## jobstudent                    1.213973   0.396820   3.059 0.002219 **
## jobtechnician                -0.111864   0.195731  -0.572 0.567647
## jobunemployed                 0.307040   0.446321   0.688 0.491493
## jobunknown                   -0.216943   0.892593  -0.243 0.807968
## educationhigh_school          0.031567   0.103473   0.305 0.760310
```

```
## educationprofessional_course  0.112266   0.126702   0.886 0.375586
## educationuniversity_degree    0.160118   0.107604   1.488 0.136745
## educationunknown              0.262894   0.163207   1.611 0.107223
## previous                      0.593811   0.162760   3.648 0.000264 ***
## poutcomenonexistent           0.484834   0.211056   2.297 0.021609 *
## default_no                    0.702380   0.091447   7.681 1.58e-14 ***
## campaign:jobadmin             0.222389   0.188214   1.182 0.237374
## campaign:jobentrepreneur      0.555713   0.407197   1.365 0.172339
## campaign:jobhousemaid         0.426708   0.489249   0.872 0.383116
## campaign:jobmanagement        0.270310   0.291108   0.929 0.353118
## campaign:jobretired           0.089643   0.322541   0.278 0.781069
## campaign:jobservices         -0.186330   0.261196  -0.713 0.475616
## campaign:jobstudent           0.436694   0.462895   0.943 0.345478
## campaign:jobtechnician        0.397403   0.214798   1.850 0.064296 .
## campaign:jobunemployed        0.352372   0.502101   0.702 0.482807
## campaign:jobunknown           1.246874   0.986308   1.264 0.206164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6647.2  on 4798  degrees of freedom
## Residual deviance: 6302.5  on 4765  degrees of freedom
##   (833 observations deleted due to missingness)
## AIC: 6370.5
##
## Number of Fisher Scoring iterations: 4
```

3.4 Heterogeneity between campaigns 2 and 3 after matching the data

```r
# create a row number
matched_data_23_r <- matched_data_23 %>% mutate(id = row_number())

# converting education dummy columns in to a single column
matched_data_23_edu <- matched_data_23_r %>% select(edu_list) %>%
                        gather("education", "value", -id ) %>%
                        filter(value == 1) %>%
                        select(-value) %>%
                        mutate(education = str_replace(education,"education_",""))

# converting Job dummy columns in to a single column
matched_data_23_job <- matched_data_23_r %>% select(job_list) %>%
                        gather("job", "value", -id ) %>%
                        filter(value == 1) %>%
                        select(-value) %>%
                        mutate(job = str_replace(job,"job_",""))

# converting marital status dummy columns in to a single column
matched_data_23_mar <- matched_data_23_r %>% select(mar_status) %>%
                        gather("marital_status", "value", -id ) %>%
                        filter(value == 1) %>%
                        select(-value) %>%
```

```
                         mutate(marital_status = str_replace(marital_status,"marital_",""))

# converting Housing dummy columns in to a single column
matched_data_23_hous <- matched_data_23_r %>% select(hous_list) %>%
                         gather("housing", "value", -id ) %>%
                         filter(value == 1) %>%
                         select(-value) %>%
                         mutate(housing = str_replace(housing,"housing_",""))

# converting poutcome dummy columns in to a single column
matched_data_23_pout <- matched_data_23_r %>% select(pout_list) %>%
                         gather("poutcome", "value", -id ) %>%
                         filter(value == 1) %>%
                         select(-value) %>%
                         mutate(poutcome = str_replace(poutcome,"poutcome_",""))

# merging all dataframes into a single data frame
merged_het_23_after <- matched_data_23_r %>%
                     select("id","y" ,"age", "campaign", "previous", "default_no", "weights") %>%
                     left_join(matched_data_23_edu, c("id" = "id")) %>%
                     left_join(matched_data_23_job, c("id" = "id")) %>%
                     left_join(matched_data_23_mar, c("id" = "id")) %>%
                     left_join(matched_data_23_hous, c("id" = "id")) %>%
                     left_join(matched_data_23_pout, c("id" = "id"))

# C2 vs C3 heterogeneity
merged_het_23_after$job <- as.factor(merged_het_23_after$job)
merged_het_23_after <- within(merged_het_23_after, job <- relevel(job, ref = 2))

# check if any ionteraction terms are significant
mod_het_23_after = glm(y ~ campaign +
                            age + marital_status + housing + job + education +
                            previous + poutcome + default_no + job*campaign,
                            data = merged_het_23_after, family = 'binomial')

summary(mod_het_23_after)
```

```
##
## Call:
## glm(formula = y ~ campaign + age + marital_status + housing +
##     job + education + previous + poutcome + default_no + job *
##     campaign, family = "binomial", data = merged_het_23_after)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0122  -1.0828  -0.7178   1.2059   1.9327
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -1.752804   0.394802  -4.440 9.01e-06 ***
## campaign                     -0.092442   0.181348  -0.510  0.61023
## age                           0.005418   0.004680   1.158  0.24699
## marital_statusmarried        -0.116058   0.139762  -0.830  0.40631
```

```
## marital_statussingle         -0.036248   0.155161  -0.234  0.81529
## housingunknown               -0.124463   0.264651  -0.470  0.63815
## housingyes                    0.078555   0.079580   0.987  0.32358
## jobadmin                     -0.069298   0.209615  -0.331  0.74095
## jobentrepreneur               0.070862   0.379004   0.187  0.85168
## jobhousemaid                  0.049378   0.407865   0.121  0.90364
## jobmanagement                -0.024203   0.297874  -0.081  0.93524
## jobretired                    0.931199   0.363065   2.565  0.01032 *
## jobservices                  -0.046779   0.274208  -0.171  0.86454
## jobstudent                    1.398495   0.443630   3.152  0.00162 **
## jobtechnician                -0.013263   0.237205  -0.056  0.95541
## jobunemployed                 0.609971   0.422093   1.445  0.14843
## jobunknown                   -0.305211   0.758879  -0.402  0.68755
## educationhigh_school         -0.019376   0.133115  -0.146  0.88427
## educationprofessional_course  0.142917   0.165894   0.861  0.38897
## educationuniversity_degree    0.281031   0.137479   2.044  0.04094 *
## educationunknown              0.196433   0.226998   0.865  0.38685
## previous                      0.476837   0.203288   2.346  0.01900 *
## poutcomenonexistent           0.384585   0.264629   1.453  0.14614
## default_no                    0.846063   0.118582   7.135 9.69e-13 ***
## campaign:jobadmin             0.436325   0.234221   1.863  0.06248 .
## campaign:jobentrepreneur     -0.268122   0.464753  -0.577  0.56400
## campaign:jobhousemaid        -0.118391   0.515256  -0.230  0.81827
## campaign:jobmanagement        0.039944   0.347356   0.115  0.90845
## campaign:jobretired           0.278940   0.427833   0.652  0.51441
## campaign:jobservices          0.315386   0.323423   0.975  0.32948
## campaign:jobstudent           0.366828   0.544873   0.673  0.50080
## campaign:jobtechnician        0.257787   0.267993   0.962  0.33609
## campaign:jobunemployed        0.250819   0.521974   0.481  0.63086
## campaign:jobunknown          -1.238950   1.300333  -0.953  0.34069
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3970.6  on 2880  degrees of freedom
## Residual deviance: 3751.5  on 2847  degrees of freedom
##   (492 observations deleted due to missingness)
## AIC: 3819.5
##
## Number of Fisher Scoring iterations: 4
```

4 Difference in Difference

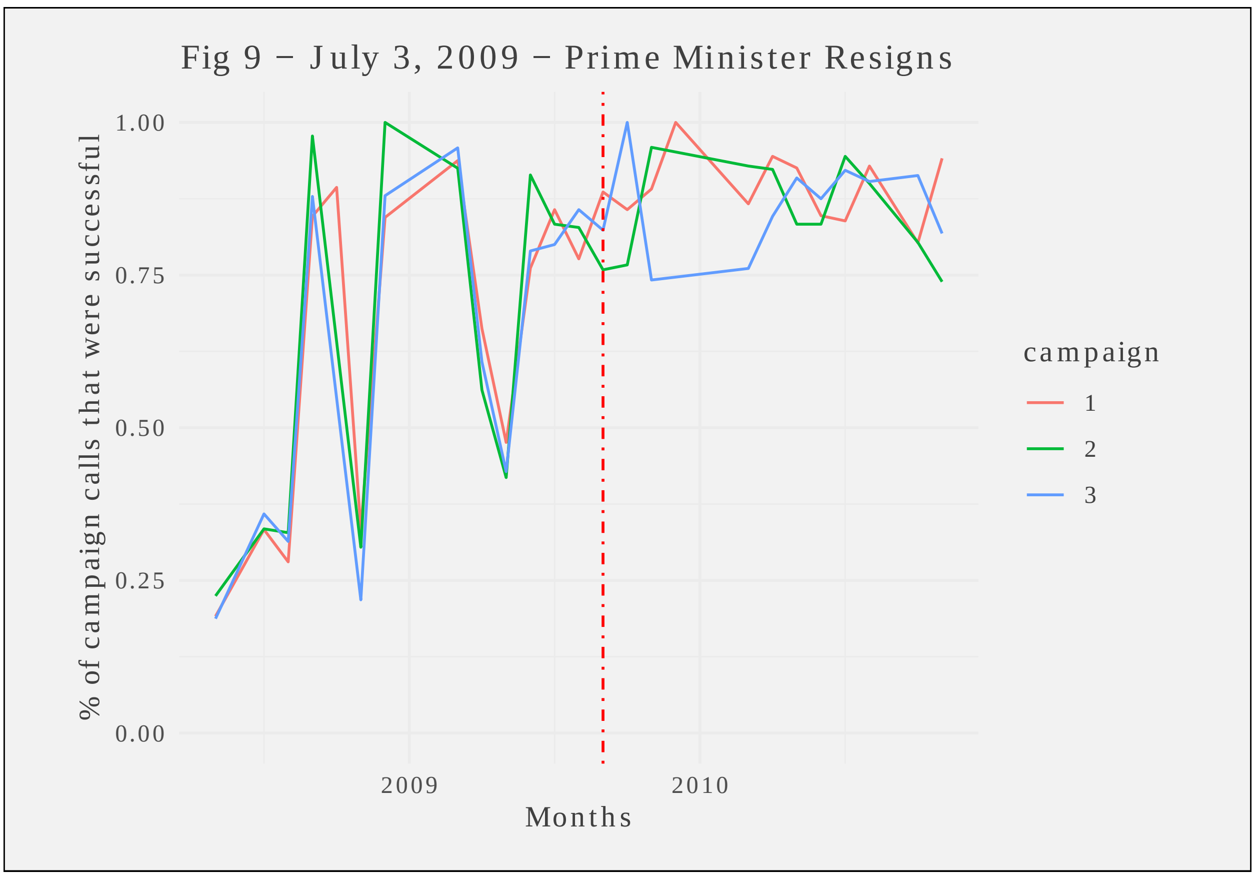
Data preparation

```
# Make sure date is in correct form
data <- data %>% mutate(DATE = mdy(DATE))


# Implement resignation date
resignation_date <- mdy('07-03-2009')
```

```r
data <- data %>% mutate(after = ifelse(DATE > resignation_date, 1, 0))


# Month-by-month success of each campaign
df <- data %>% group_by(per_year = year(DATE),
                        per_month = month(DATE),
                        campaign = factor(campaign)) %>%
  summarize(averaging = mean(y),
            counting = n())


# Make easier to plot
df$Date <- as.yearmon(paste(df$per_year, df$per_month), "%Y %m")
```

Analyse call success rate across time before and after the news

```r
ggplot(data=df, aes(x=Date, y = averaging, color = campaign)) +
  geom_line() + ylim(0,1) +
  xlab("Months") + ylab("% of campaign calls that were successful") +
  ggtitle('Fig 9 - July 3, 2009 - Prime Minister Resigns') +
  geom_vline(xintercept = as.numeric(df$Date[38]), linetype=4, colour="red") +
  theme_ilo()
```



4.1 Regression for Difference in Difference

```r
# setting campaign 3 as reference
data$campaign <- factor(data$campaign)
data <- within(data, campaign <- relevel(campaign, ref = 3))

# Set camapign 3 as 0 and rest of the campaigns as 1
data_123 <- data %>%
  mutate(treat = ifelse(campaign == 3, 0, 1))

# setting campaign 3 as reference
data_123 <- within(data_123, campaign <- relevel(campaign, ref = 3))
data_123$treat <- factor(data_123$treat)

# DiD regresion to chech the impact on campaings after the news
did_one <- glm(y ~ treat + after + treat * after,
               data = data_123, family = "binomial")

summary(did_one)
```

```
##
## Call:
## glm(formula = y ~ treat + after + treat * after, family = "binomial",
##     data = data_123)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9840  -1.0865   0.5486   1.2712   1.3965
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.50156    0.04667 -10.746  < 2e-16 ***
## treat1       0.28386    0.05513   5.149 2.62e-07 ***
## after        2.31912    0.15802  14.676  < 2e-16 ***
## treat1:after -0.31273    0.17659  -1.771   0.0766 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 11831  on 8544  degrees of freedom
## Residual deviance: 10601  on 8541  degrees of freedom
## AIC: 10609
##
## Number of Fisher Scoring iterations: 4
```

4.2 Placebo effect

We wanted to move the resignation date of president few months back artificially and check the treatment effect

```r
# moving the prime minister resigantion data 3 months prior
resignation_plc_date <- mdy('04-03-2009')

data_placebo = data_123 %>%
               mutate(after_placebo = ifelse(DATE > resignation_plc_date, 1, 0))

# DiD with placebo data
did_basic_placebo_one = glm(y ~ treat + after_placebo + treat * after_placebo,
                            data = data_placebo,
                            family = "binomial")

summary(did_basic_placebo_one)
```

```
##
## Call:
## glm(formula = y ~ treat + after_placebo + treat * after_placebo,
##     family = "binomial", data = data_placebo)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6160  -1.0253   0.7950   0.9061   1.4596
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -0.64255    0.05381 -11.941  < 2e-16 ***
## treat1                 0.27377    0.06372   4.297 1.73e-05 ***
## after_placebo          1.32055    0.09254  14.270  < 2e-16 ***
## treat1:after_placebo  0.03800    0.10781   0.353    0.724
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 11831  on 8544  degrees of freedom
## Residual deviance: 10909  on 8541  degrees of freedom
## AIC: 10917
##
## Number of Fisher Scoring iterations: 4
```