Team 13
A1: Air France – Business Case Presentation
Ambre Cherkaoui, Juan Pedro Ballester Vives,
Manish Kola, Yuping Li, Renata Pérez Retes.

```
library(readxl)
org_df<- read_excel("C:/Users/liyuping/Desktop/Homework/R/teamwork/Air France Case Spreadsheet Supplement.xls",
                    sheet = "DoubleClick")
View(org_df)

################## renaming columns ##############################################################################

names(org_df) <- c("pub_id", "pub_name", "keyword_id","keyword","match_type",
                   "campaign","keyword_group","category","bid_strat","keyword_type",
                   "status","se_bid","clicks","click_charges","avg_cpc","impressions",
                   "engine_click_thru_perc","avg_pos","trans_conv_perc",
                   "total_cost_per_trans","revenue","total_cost","total_booking")

################## creating dummies  ##############################################################################

#creating dummies for publisher name
org_df$pub_name_cat <- gsub("Google - US","6",org_df$pub_name)
org_df$pub_name_cat <- gsub("Google - Global","5",org_df$pub_name_cat)
org_df$pub_name_cat <- gsub("MSN - US","4",org_df$pub_name_cat)
org_df$pub_name_cat <- gsub("MSN - Global","3",org_df$pub_name_cat)
org_df$pub_name_cat <- gsub("Overture - US","2",org_df$pub_name_cat)
org_df$pub_name_cat <- gsub("Overture - Global","1",org_df$pub_name_cat)
org_df$pub_name_cat <- gsub("Yahoo - US","0",org_df$pub_name_cat)
org_df$pub_name_cat <- as.numeric(org_df$pub_name_cat) #converting characters into numeric data type
table(org_df$pub_name_cat) #checking distribution of observations

#creating dummies for publisher name
org_df$google <- gsub(".*Google.*",1,org_df$pub_name)
```

```
org_df$msn <- gsub(".*MSN.*",1,org_df$pub_name)
org_df$overture <- gsub(".*Overture.*",1,org_df$pub_name)
org_df$yahoo <- gsub("Yahoo - US",1,org_df$pub_name)
org_df$google <- as.numeric(org_df$google) #converting characters into numeric data type
org_df$msn <- as.numeric(org_df$msn)
org_df$overture <- as.numeric(org_df$overture)
org_df$yahoo <- as.numeric(org_df$yahoo)

org_df$google[is.na(org_df$google)] = 0
org_df$msn[is.na(org_df$msn)] = 0
org_df$overture[is.na(org_df$overture)] = 0
org_df$yahoo[is.na(org_df$yahoo)] = 0

org_df$google_us <- gsub("Google - US",1,org_df$pub_name)
org_df$msn_us <- gsub("MSN - US",1,org_df$pub_name)
org_df$overture_us <- gsub("Overture - US",1,org_df$pub_name)

org_df$google_us <- as.numeric(org_df$google_us) #converting characters into numeric data type
org_df$msn_us <- as.numeric(org_df$msn_us)
org_df$overture_us <- as.numeric(org_df$overture_us)

org_df$google_us[is.na(org_df$google_us)] = 0
org_df$msn_us[is.na(org_df$msn_us)] = 0
org_df$overture_us[is.na(org_df$overture_us)] = 0

#creating dummies for match type
org_df$match_type_cat <- gsub("N/A","0",org_df$match_type)
org_df$match_type_cat <- gsub("Exact","1",org_df$match_type_cat)
org_df$match_type_cat <- gsub("Standard","2",org_df$match_type_cat)
org_df$match_type_cat <- gsub("Broad","3",org_df$match_type_cat)
org_df$match_type_cat <- gsub("Advanced","4",org_df$match_type_cat)
org_df$match_type_cat <- as.numeric(org_df$match_type_cat) #converting characters into numeric data type
table(org_df$match_type_cat) #checking distribution of observations
```

```
#creating dummies for status
org_df$status_cat <- gsub("Unavailable","0",org_df$status)
org_df$status_cat <- gsub("Deactivated","1",org_df$status_cat)
org_df$status_cat <- gsub("Paused","2",org_df$status_cat)
org_df$status_cat <- gsub("Sent","3",org_df$status_cat)
org_df$status_cat <- gsub("Live","4",org_df$status_cat)
org_df$status_cat <- as.numeric(org_df$status_cat) #converting characters into numeric data type
table(org_df$status_cat) #checking distribution of observations

#creating dummies for bid strategy
org_df$bid_strat_cat <- gsub("Position 1-2 Target","6",org_df$bid_strat)
org_df$bid_strat_cat <- gsub("Position 1 -2 Target","6",org_df$bid_strat_cat)
org_df$bid_strat_cat <- gsub("Position 1- 3","5",org_df$bid_strat_cat)
org_df$bid_strat_cat <- gsub("Position 1-4 Bid Strategy","4",org_df$bid_strat_cat)
org_df$bid_strat_cat <- gsub("Postiion 1-4 Bid Strategy","4",org_df$bid_strat_cat)
org_df$bid_strat_cat <- gsub("Position 2-5 Bid Strategy","3",org_df$bid_strat_cat)
org_df$bid_strat_cat <- gsub("Pos 3-6","2",org_df$bid_strat_cat)
org_df$bid_strat_cat <- gsub("Position 5-10 Bid Strategy","1",org_df$bid_strat_cat)
org_df$bid_strat_cat[is.na(org_df$bid_strat_cat)] = 0
org_df$bid_strat_cat <- as.numeric(org_df$bid_strat_cat) #converting characters into numeric data type
table(org_df$bid_strat_cat)#checking distribution of observations

#creating dummies for campaign
org_df$campaign_cat <- gsub("Unassigned","0",org_df$campaign)
org_df$campaign_cat <- gsub(".*Air France.*","1",org_df$campaign_cat)
org_df$campaign_cat <- gsub(".*Geo Targeted.*","2",org_df$campaign_cat)
org_df$campaign_cat <- gsub("Business Class","3",org_df$campaign_cat)
org_df$campaign_cat <- gsub("French Destinations","3",org_df$campaign_cat)
org_df$campaign_cat <- gsub("General Terms","3",org_df$campaign_cat)
org_df$campaign_cat <- gsub("Google_Yearlong 2006","3",org_df$campaign_cat)
org_df$campaign_cat <- gsub("Outside Western Europe","3",org_df$campaign_cat)
org_df$campaign_cat <- gsub("Paris & France Terms","3",org_df$campaign_cat)
org_df$campaign_cat <- gsub("Western Europe Destinations","3",org_df$campaign_cat)
org_df$campaign_cat <- as.numeric(org_df$campaign_cat) #converting characters into numeric data type
```

```
table(org_df$campaign_cat) #checking distribution of observations

# creating binary y variable
org_df$booking_binary <- org_df$total_booking #copying information from total bookings

#populating new column with 1 and 0s
for(i in 1:nrow(org_df)){
  if(org_df[i,"booking_binary"]>0){
    org_df[i,"booking_binary"] =1 # business success, bookings greater than 0
  }else{
    org_df[i,"booking_binary"] =0 # business failure, bookings equal to 0
  }
} # closing i loop
table(org_df$booking_binary) # checking distribution of observations
```

```
$ pub_name_cat               : num [1:4510] 6 0 5 6 6 1 5 6 0 3 ...
$ google                     : num [1:4510] 1 0 1 1 1 0 1 1 0 0 ...
$ msn                        : num [1:4510] 0 0 0 0 0 0 0 0 0 1 ...
$ overture                   : num [1:4510] 0 0 0 0 0 1 0 0 0 0 ...
$ yahoo                      : num [1:4510] 0 1 0 0 0 0 0 0 1 0 ...
$ google_us                  : num [1:4510] 1 0 0 1 1 0 0 1 0 0 ...
$ msn_us                     : num [1:4510] 0 0 0 0 0 0 0 0 0 0 ...
$ overture_us                : num [1:4510] 0 0 0 0 0 0 0 0 0 0 ...
$ match_type_cat             : num [1:4510] 1 4 1 3 3 4 1 3 4 3 ...
$ status_cat                 : num [1:4510] 4 4 0 4 4 2 0 4 4 1 ...
$ bid_strat_cat              : num [1:4510] 0 0 0 0 0 6 5 4 0 4 ...
$ campaign_cat               : num [1:4510] 1 1 1 1 1 0 1 1 1 1 ...
$ booking_binary             : num [1:4510] 1 1 1 1 1 1 1 1 1 1 ...
$ profit                     : num [1:4510] 502455 549524 249321 244421 214503 ...
$ se_bid_norm                : num [1:4510] 1 0.273 0.182 1 1 ...
$ avg_cpc_norm               : num [1:4510] 0.0438 0.0751 0.1042 0.1589 0.0205 ...
$ impressions_norm           : num [1:4510] 0.02258 0.01943 0.00528 0.04621 0.00945 ...
$ engine_click_thru_perc_norm: num [1:4510] 0.0807 0.0737 0.1671 0.0377 0.0748 ...
$ revenue_norm               : num [1:4510] 0.909 1 0.466 0.512 0.382 ...
$ campaign_name              : chr [1:4510] "Air France Series" "Air France Series" "Air…
$ bid_strat_name             : chr [1:4510] "Unknown" "Unknown" "Unknown" "Unknown" ...
```

################ Modeling - Logistic Regression ###################################

```
# random sampling into training and testing sets
training_index <- sample(1:nrow(org_df), size=0.8*nrow(org_df))
train_set <- org_df[training_index,]
test_set <- org_df[-training_index,]
```

test_set	902 obs. of 36 variables
train_set	3608 obs. of 36 variables

```
# installing package for logistic regression
#install.packages("mlbench")
library(mlbench) #calling package to environment

lg_mod <- glm(booking_binary ~ status_cat+
                               pub_name_cat+
                               campaign_cat+
                               match_type_cat+
                               bid_strat_cat,
              data = train_set, #full data set
              family = "binomial") #specifying a classification model

summary(lg_mod) #visualizing results
```

```
Call:
glm(formula = booking_binary ~ status_cat + pub_name_cat + campaign_cat +
    match_type_cat + bid_strat_cat, family = "binomial", data = train_set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2466  -0.4546  -0.3607  -0.2662   3.0134

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    -4.47863    0.36530 -12.260  < 2e-16 ***
status_cat      0.31132    0.05260   5.919 3.24e-09 ***
pub_name_cat    0.29472    0.04744   6.212 5.24e-10 ***
campaign_cat   -0.48220    0.09305  -5.182 2.19e-07 ***
match_type_cat  0.19324    0.09908   1.950   0.0511 .
bid_strat_cat   0.25477    0.03268   7.796 6.39e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2109.5  on 3607  degrees of freedom
Residual deviance: 1958.0  on 3602  degrees of freedom
AIC: 1970

Number of Fisher Scoring iterations: 6
```

```
lg_mod_1 <- glm(booking_binary ~ status_cat+
                                 pub_name_cat+
                                 campaign_cat+
                                 bid_strat_cat,
                data = train_set, #full data set
                family = "binomial") #specifying a classification model

summary(lg_mod_1) #visualizing results
```

```
Call:
glm(formula = booking_binary ~ status_cat + pub_name_cat + campaign_cat +
    bid_strat_cat, family = "binomial", data = train_set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2482  -0.4503  -0.3633  -0.2661   3.0489

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -3.93381    0.22300 -17.640  < 2e-16 ***
status_cat     0.31919    0.05264   6.063 1.33e-09 ***
pub_name_cat   0.28164    0.04743   5.938 2.88e-09 ***
campaign_cat  -0.44762    0.09161  -4.886 1.03e-06 ***
bid_strat_cat  0.26329    0.03248   8.107 5.20e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2109.5  on 3607  degrees of freedom
Residual deviance: 1961.9  on 3603  degrees of freedom
AIC: 1971.9

Number of Fisher Scoring iterations: 6
```

```
library(caret)
prediction_test <- predict(lg_mod_1,test_set,type = "response")
summary(prediction_test)
```

```
> summary(prediction_test)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
0.009581 0.046806 0.063853 0.083347 0.101366 0.541121
```

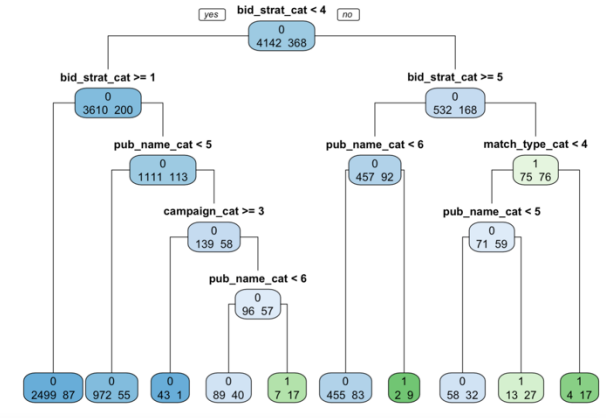
```
confusionMatrix(data = as.factor(as.numeric(prediction_test>0.1)),
                reference =as.factor(as.numeric(test_set$booking_binary)))
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 650  26
         1 193  33

               Accuracy : 0.7572
                 95% CI : (0.7279, 0.7849)
    No Information Rate : 0.9346
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1426

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.7711
            Specificity : 0.5593
         Pos Pred Value : 0.9615
         Neg Pred Value : 0.1460
             Prevalence : 0.9346
         Detection Rate : 0.7206
   Detection Prevalence : 0.7494
      Balanced Accuracy : 0.6652

       'Positive' Class : 0
```

```
prediction_train <- predict(lg_mod_1,train_set,type = "response")
confusionMatrix(data = as.factor(as.numeric(prediction_train>0.1)),
                reference =as.factor(as.numeric(train_set$booking_binary)))
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2538  122
         1  761  187

               Accuracy : 0.7553
                 95% CI : (0.7409, 0.7692)
    No Information Rate : 0.9144
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1933

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.7693
            Specificity : 0.6052
         Pos Pred Value : 0.9541
         Neg Pred Value : 0.1973
             Prevalence : 0.9144
         Detection Rate : 0.7034
   Detection Prevalence : 0.7373
      Balanced Accuracy : 0.6873

       'Positive' Class : 0
```

```
library(rpart)
library(rpart.plot)
af_tree <-
rpart(booking_binary~campaign_cat+pub_name_cat+bid_strat_cat+match_type_cat,
              data = org_df,method = "class",
              cp=0.005)
rpart.plot(af_tree,type = 1,extra = 1)
```



```
lm_google <- glm(booking_binary ~  google_us+
                                   campaign_cat+
                                   status_cat+
                                   bid_strat_cat,
                 data = train_set, #full data set
                 family = "binomial") #specifying a classification model

summary(lm_google) #visualizing results
```

```
Call:
glm(formula = booking_binary ~ google_us + campaign_cat + status_cat +
    bid_strat_cat, family = "binomial", data = train_set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.7145  -0.3753  -0.3414  -0.3305   2.5666

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -3.02416    0.18881 -16.017  < 2e-16 ***
google_us     -0.30373    0.19475  -1.560    0.119
campaign_cat   0.03597    0.08524   0.422    0.673
status_cat     0.07221    0.05218   1.384    0.166
bid_strat_cat  0.26206    0.03300   7.941 2.01e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1993.8  on 3607  degrees of freedom
Residual deviance: 1906.9  on 3603  degrees of freedom
AIC: 1916.9

Number of Fisher Scoring iterations: 5
```

```
lm_yahoo <- glm(booking_binary ~  yahoo+
                                  campaign_cat+
                                  match_type_cat+
                                  bid_strat_cat,
                data = train_set, #full data set
                family = "binomial") #specifying a classification model

summary(lm_yahoo) #visualizing results
```

```
Call:
glm(formula = booking_binary ~ yahoo + campaign_cat + match_type_cat +
    bid_strat_cat, family = "binomial", data = train_set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.7297  -0.3642  -0.3555  -0.3348   2.6513

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    -3.25142    0.30520 -10.653  < 2e-16 ***
yahoo          -0.01801    0.24110  -0.075    0.940
campaign_cat   -0.07771    0.06245  -1.244    0.213
match_type_cat  0.14284    0.09532   1.499    0.134
bid_strat_cat   0.24876    0.03431   7.251 4.14e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1993.8  on 3607  degrees of freedom
Residual deviance: 1909.2  on 3603  degrees of freedom
AIC: 1919.2

Number of Fisher Scoring iterations: 5
```

```
lm_overture <- glm(booking_binary ~  overture_us+
                                campaign_cat+
                                match_type_cat+
                                bid_strat_cat,
                   data = train_set, #full data set
                   family = "binomial") #specifying a classification model

summary(lm_overture) #visualizing results
```

```
Call:
glm(formula = booking_binary ~ overture_us + campaign_cat + match_type_cat +
    bid_strat_cat, family = "binomial", data = train_set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.7462  -0.3638  -0.3583  -0.3347   2.6314

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    -2.94860    0.33204  -8.880  < 2e-16 ***
overture_us    -0.42701    0.22888  -1.866   0.0621 .
campaign_cat   -0.16052    0.07518  -2.135   0.0328 *
match_type_cat  0.11742    0.08997   1.305   0.1919
bid_strat_cat   0.22380    0.03452   6.483 8.98e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1993.8  on 3607  degrees of freedom
Residual deviance: 1905.7  on 3603  degrees of freedom
AIC: 1915.7

Number of Fisher Scoring iterations: 5
```

```
lm_msn<- glm(booking_binary ~  msn_us+
                           campaign_cat+
                           match_type_cat+
                           bid_strat_cat,
             data = train_set, #full data set
             family = "binomial") #specifying a classification model

summary(lm_msn) #visualizing results
```

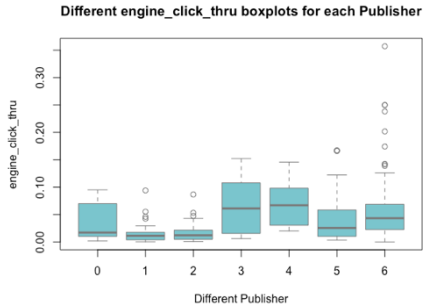
```
Call:
glm(formula = booking_binary ~ msn_us + campaign_cat + match_type_cat +
    bid_strat_cat, family = "binomial", data = train_set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.7313  -0.3997  -0.3494  -0.3310   2.6566

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    -3.30533    0.30410 -10.869  < 2e-16 ***
msn_us          0.75994    0.38877   1.955   0.0506 .
campaign_cat   -0.06452    0.06331  -1.019   0.3081
match_type_cat  0.13936    0.08972   1.553   0.1204
bid_strat_cat   0.26095    0.03341   7.811 5.65e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1993.8  on 3607  degrees of freedom
Residual deviance: 1906.0  on 3603  degrees of freedom
AIC: 1916

Number of Fisher Scoring iterations: 5
```

```
#creating a udf to normalize with min-max rescaling
normalize <- function(var){
  my_norm <- (var-min(var))/(max(var)-min(var))
  return(my_norm)
}#closing normalize function
```

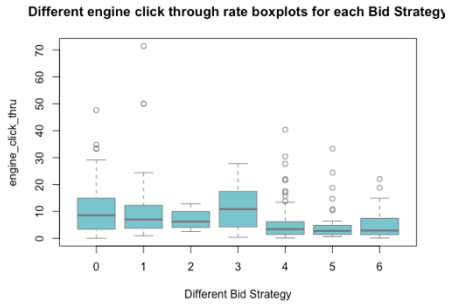
```
org_df$profit <- org_df$revenue -org_df$total_cost

org_df$se_bid_norm <- normalize(org_df$se_bid)
org_df$avg_cpc_norm <- normalize(org_df$avg_cpc)
org_df$impressions_norm <- normalize(org_df$impressions)
org_df$engine_click_thru_perc_norm <- normalize(org_df$engine_click_thru_perc)
org_df$revenue_norm <- normalize(org_df$revenue)
org_df$revenue_norm <- normalize(org_df$revenue)
```

```
 $ se_bid_norm                : num [1:4510] 1 0.273 0.182 1 1 ...
 $ avg_cpc_norm               : num [1:4510] 0.0438 0.0751 0.1042 0.1589 0.0205 ...
 $ impressions_norm           : num [1:4510] 0.02258 0.01943 0.00528 0.04621 0.00945 ...
 $ engine_click_thru_perc_norm: num [1:4510] 0.0807 0.0737 0.1671 0.0377 0.0748 ...
 $ revenue_norm               : num [1:4510] 0.909 1 0.466 0.512 0.382 ...
```

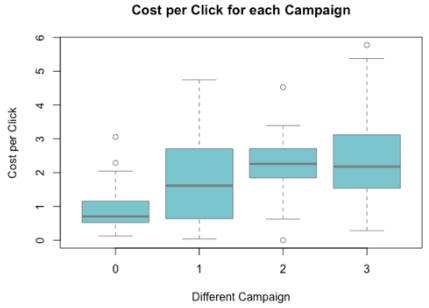
```
df_suc <- org_df[!org_df$total_booking==0,]
df_fail <- org_df[org_df$total_booking==0,]

## Insights – Boxplots, Barcharts
boxplot(engine_click_thru_perc_norm~pub_name_cat,
        data=df_suc,
        main="Different engine_click_thru boxplots for each Publisher",
        xlab="Different Publisher",
        ylab="engine_click_thru",
        col="cadetblue3",
        border="gray48"
)
```



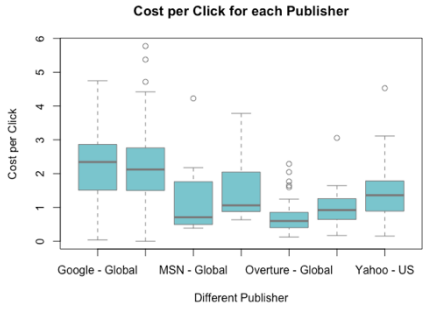
```
boxplot(engine_click_thru_perc~bid_strat_cat,
        data=df_suc,
        main="Different engine click through rate boxplots for each Bid Strategy",
        xlab="Different Bid Strategy",
        ylab="engine_click_thru",
        col="cadetblue3",
        border="gray48"
)
```



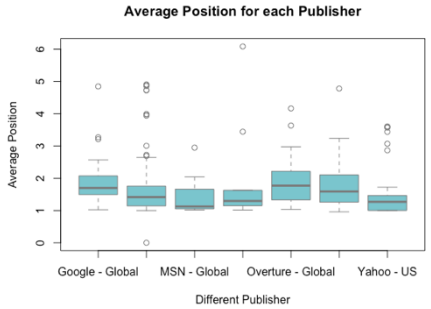

```
boxplot(avg_cpc~campaign_cat,
        data=df_suc,
        main="Cost per Click for each Campaign",
        xlab="Different Campaign",
        ylab="Cost per Click",
        col="cadetblue3",
        border="gray48"
)
```



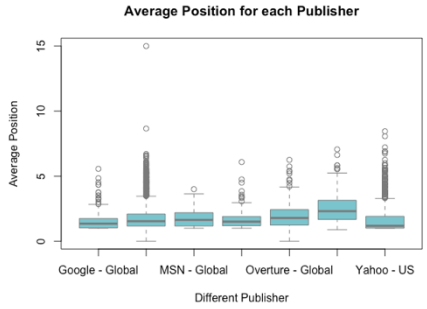

```
boxplot(avg_cpc~pub_name,
        data=df_suc,
        main="Cost per Click for each Publisher",
        xlab="Different Publisher",
        ylab="Cost per Click",
        col="cadetblue3",
        border="gray48"
)
```



```
boxplot(avg_pos~pub_name,
        data=df_suc,
        main="Average Position for each Publisher",
        xlab="Different Publisher",
        ylab="Average Position",
        col="cadetblue3",
        border="gray48"
)
```




```
boxplot(avg_pos~pub_name,
        data=org_df,
        main="Average Position for each Publisher",
        xlab="Different Publisher",
        ylab="Average Position",
        col="cadetblue3",
        border="gray48"
)
```

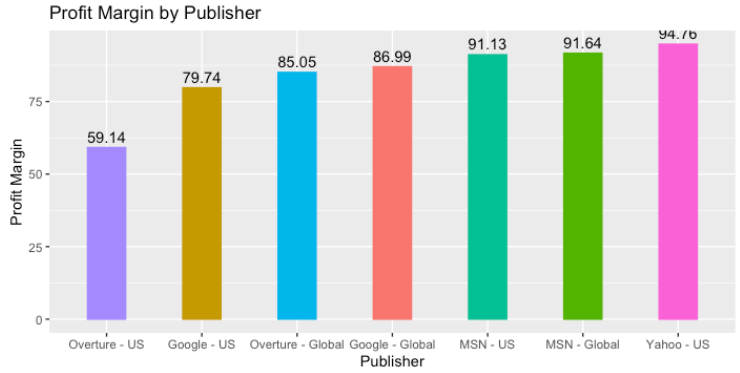



```
## Sum by publisher
sum_pub<-aggregate(cbind(revenue, total_cost, total_booking, clicks) ~ pub_name, data = org_df, FUN=sum, na.rm=TRUE)
sum_pub$profit <- sum_pub$revenue - sum_pub$total_cost
sum_pub$profit_margin <- round((sum_pub$profit/sum_pub$revenue)*100,2)

## Average return on investment
sum_pub$roi<-round(sum_pub$profit/sum_pub$total_cost,2)

## Average cost per booking
sum_pub$avg_cpb <- round(sum_pub$total_cost/sum_pub$total_booking,2)

## Conversion rate
```

```
sum_pub$convers_rate <- round((sum_pub$total_booking/sum_pub$clicks)*100,2)
sum_pub$avg_cpc <- round((sum_pub$total_cost/sum_pub$clicks),2)
```

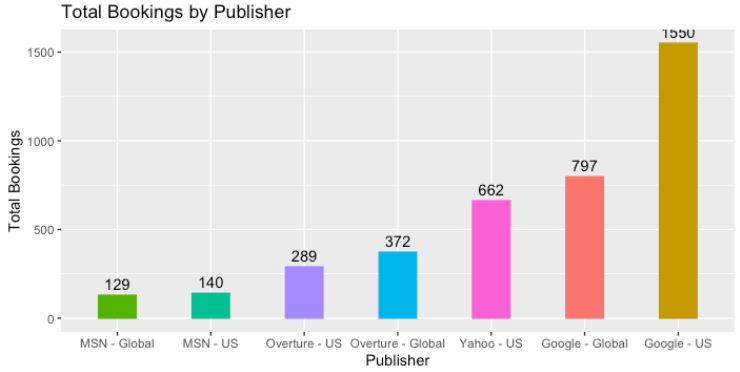
sum_bid_str	7 obs. of 11 variables
sum_cam	4 obs. of 11 variables
sum_pub	7 obs. of 11 variables

```
##Plotting
library(ggplot2)

#install.packages("tidyverse")
#installed.packages("gapminder")
#install.packages("scales")
library(tidyverse)
library(gapminder)
library(scales)

#Publisher - Profit Margin
ggplot(data = sum_pub, aes(reorder(x=pub_name,profit_margin),y=profit_margin,color=pub_name,fill=pub_name)) +
  geom_bar(stat="identity", width=0.4) + labs(x="Publisher") + labs(y="Profit Margin") + ggtitle("Profit Margin by Publisher")
+geom_text(aes(label = profit_margin), vjust = -0.5, colour = "black") +theme(legend.position="none")
```



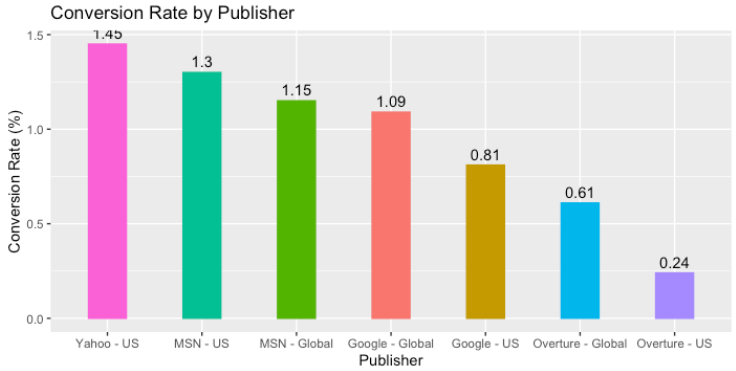
#Publisher - Total Bookings

```
ggplot(data = sum_pub, aes(reorder(x=pub_name, total_booking),y=total_booking,color=pub_name,fill=pub_name)) +
  geom_bar(stat="identity", width=0.4) + labs(x="Publisher") + labs(y="Total Bookings") + ggtitle("Total Bookings by Publisher")
+geom_text(aes(label = total_booking), vjust = -0.5, colour = "black") +theme(legend.position="none")
```

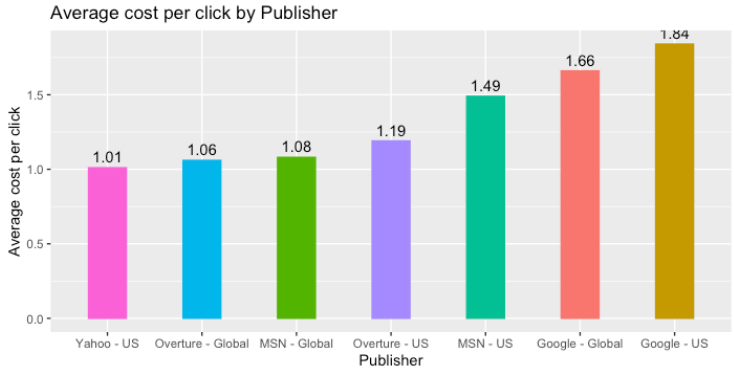


#Publisher - Conversion Rate

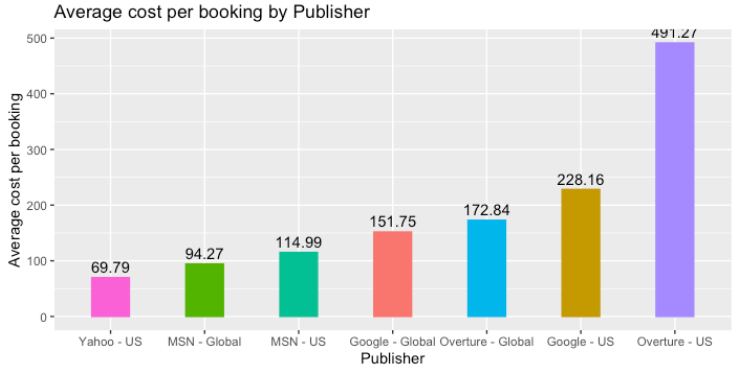
```
ggplot(data = sum_pub, aes(reorder(x=pub_name, -convers_rate),y=convers_rate,color=pub_name,fill=pub_name)) +
  geom_bar(stat="identity", width=0.4) + labs(x="Publisher") + labs(y="Conversion Rate (%)") + ggtitle("Conversion Rate by
Publisher") +geom_text(aes(label = convers_rate), vjust = -0.5, colour = "black") +theme(legend.position="none")
```



#Publisher - Average Cost per Click
```
ggplot(data = sum_pub, aes(reorder(x=pub_name, avg_cpc),y=avg_cpc,color=pub_name,fill=pub_name)) +
  geom_bar(stat="identity", width=0.4) + labs(x="Publisher") + labs(y="Average cost per click") + ggtitle("Average cost per click by Publisher") +geom_text(aes(label = avg_cpc), vjust = -0.5, colour = "black") +theme(legend.position="none")
```

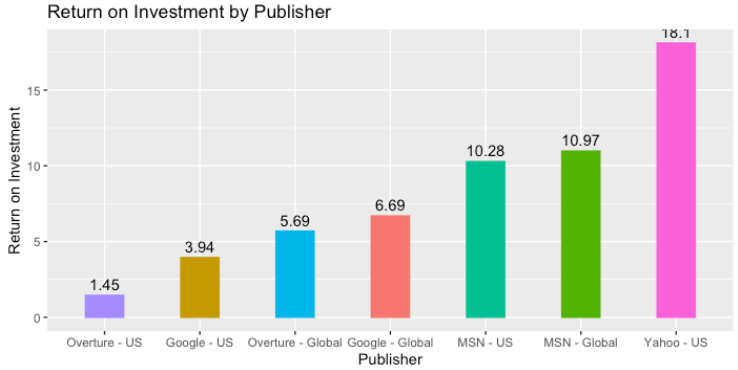


#Publisher - Average Cost per Book
```
ggplot(data = sum_pub, aes(reorder(x=pub_name, avg_cpb),y=avg_cpb,color=pub_name,fill=pub_name)) +
  geom_bar(stat="identity", width=0.4) + labs(x="Publisher") + labs(y="Average cost per booking") + ggtitle("Average cost per booking by Publisher") +geom_text(aes(label = avg_cpb), vjust = -0.5, colour = "black") +theme(legend.position="none")
```



#Publisher - Return on Investment
```
ggplot(data = sum_pub, aes(reorder(x=pub_name, roi),y=roi,color=pub_name,fill=pub_name)) +
```

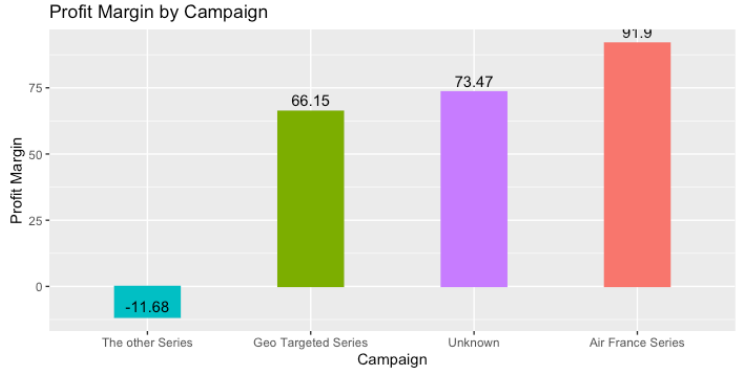
```
  geom_bar(stat="identity", width=0.4) + labs(x="Publisher") + labs(y="Return on Investment") + ggtitle("Return on Investment by Publisher") +geom_text(aes(label = roi), vjust = -0.5, colour = "black") +theme(legend.position="none")
```




```
##sum by Campaign
#change the name in text
org_df$campaign_name <- gsub(0,"Unknown",org_df$campaign_cat)
org_df$campaign_name <- gsub(1,"Air France Series",org_df$campaign_name)
org_df$campaign_name <- gsub(2,"Geo Targeted Series",org_df$campaign_name)
org_df$campaign_name <- gsub(3,"The other Series",org_df$campaign_name)
sum_cam<-aggregate(cbind(revenue, total_cost, total_booking, clicks) ~ campaign_name, data = org_df, FUN=sum, na.rm=TRUE)

sum_cam$profit <- sum_cam$revenue - sum_cam$total_cost
sum_cam$profit_margin <- round((sum_cam$profit/sum_cam$revenue)*100,2)

##Average return on investment
sum_cam$roi<-sum_cam$profit/sum_cam$total_cost

##Average cost per booking
sum_cam$avg_cpb <- sum_cam$total_cost/sum_cam$total_booking

##Conversion rate
sum_cam$convers_rate <- round((sum_cam$total_booking/sum_cam$clicks)*100,2)
sum_cam$avg_cpc <- round((sum_cam$total_cost/sum_cam$clicks),2)
```

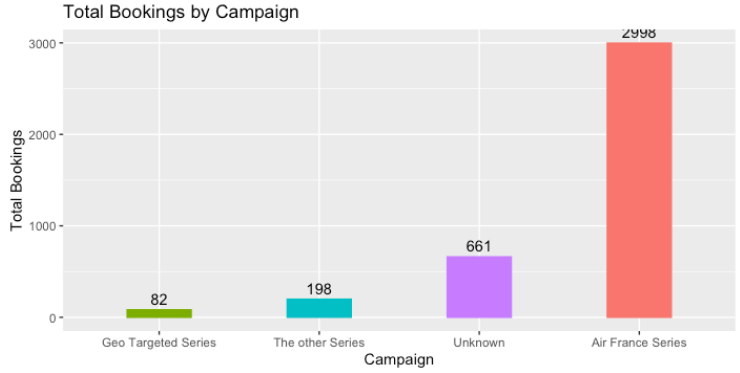
##Campaign - Profit Margin

```
ggplot(data = sum_cam, aes(reorder(x=campaign_name,profit_margin),y=profit_margin,color=campaign_name,fill=campaign_name))
+
  geom_bar(stat="identity", width=0.4) + labs(x="Campaign") + labs(y="Profit Margin") + ggtitle("Profit Margin by Campaign")
+geom_text(aes(label = profit_margin), vjust = -0.5, colour = "black") +theme(legend.position="none")
```



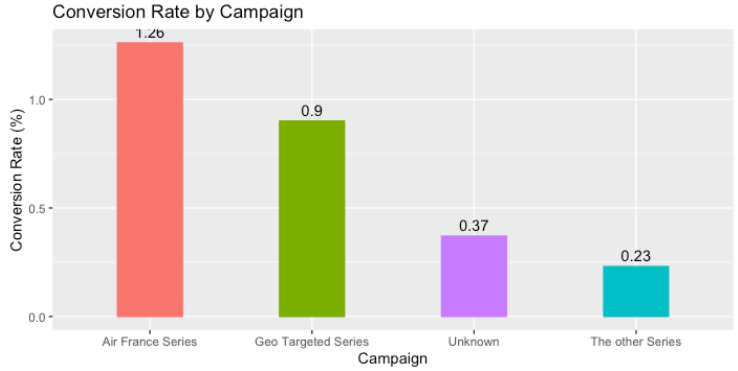
#Campaign - Total Bookings

```
ggplot(data = sum_cam, aes(reorder(x=campaign_name,
total_booking),y=total_booking,color=campaign_name,fill=campaign_name)) +
  geom_bar(stat="identity", width=0.4) + labs(x="Campaign") + labs(y="Total Bookings") + ggtitle("Total Bookings by Campaign")
+geom_text(aes(label = total_booking), vjust = -0.5, colour = "black") +theme(legend.position="none")
```



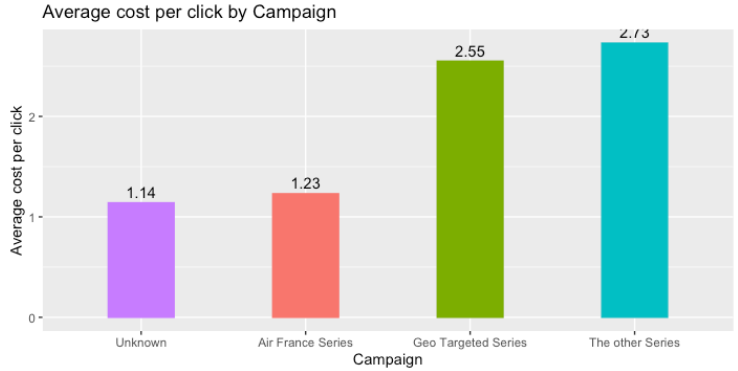
#Campaign - Conversion Rate

```
ggplot(data = sum_cam, aes(reorder(x=campaign_name, -
convers_rate),y=convers_rate,color=campaign_name,fill=campaign_name)) +
  geom_bar(stat="identity", width=0.4) + labs(x="Campaign") + labs(y="Conversion Rate (%)") + ggtitle("Conversion Rate by
Campaign") +geom_text(aes(label = convers_rate), vjust = -0.5, colour = "black") +theme(legend.position="none")
```



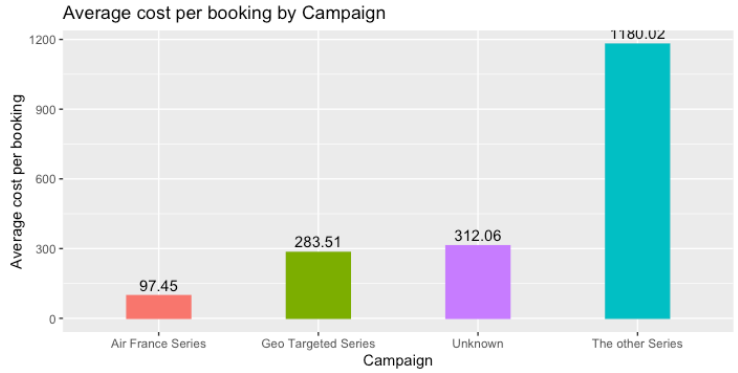
#Campaign - Average Cost per Click

```
ggplot(data = sum_cam, aes(reorder(x=campaign_name, avg_cpc),y=avg_cpc,color=campaign_name,fill=campaign_name)) +
  geom_bar(stat="identity", width=0.4) + labs(x="Campaign") + labs(y="Average cost per click") + ggtitle("Average cost per click by
Campaign") +geom_text(aes(label = avg_cpc), vjust = -0.5, colour = "black") +theme(legend.position="none")
```



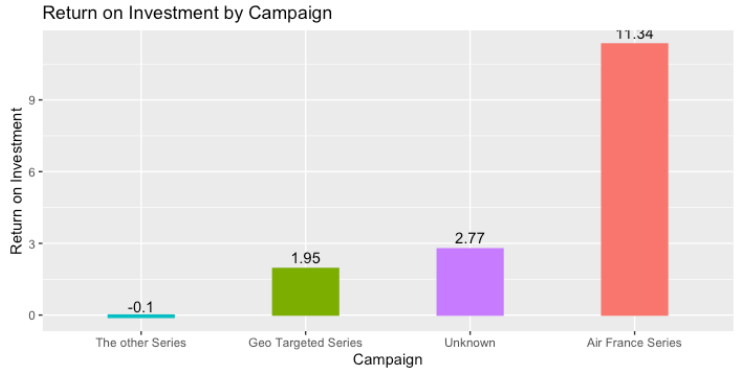
##Campaign - Average Cost per Book

```
ggplot(data = sum_cam, aes(reorder(x=campaign_name, avg_cpb),y=avg_cpb,color=campaign_name,fill=campaign_name)) +
  geom_bar(stat="identity", width=0.4) + labs(x="Campaign") + labs(y="Average cost per booking") + ggtitle("Average cost per
booking by Campaign") +geom_text(aes(label = avg_cpb), vjust = -0.5, colour = "black") +theme(legend.position="none")
```

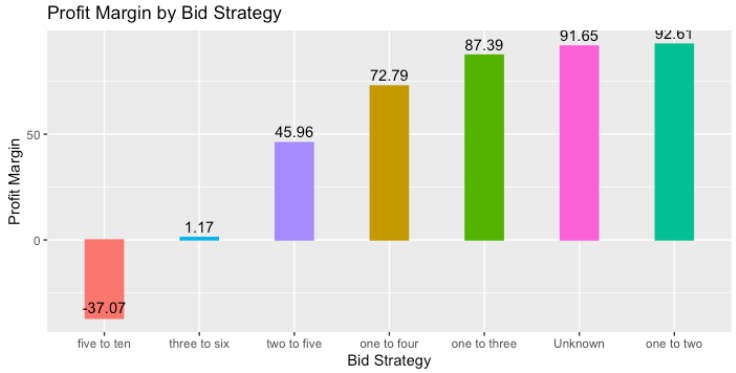


#Campaign - Return on Investment

```
ggplot(data = sum_cam, aes(reorder(x=campaign_name, roi),y=roi,color=campaign_name,fill=campaign_name)) +
  geom_bar(stat="identity", width=0.4) + labs(x="Campaign") + labs(y="Return on Investment") + ggtitle("Return on Investment by
Campaign") +geom_text(aes(label = roi), vjust = -0.5, colour = "black") +theme(legend.position="none")
```

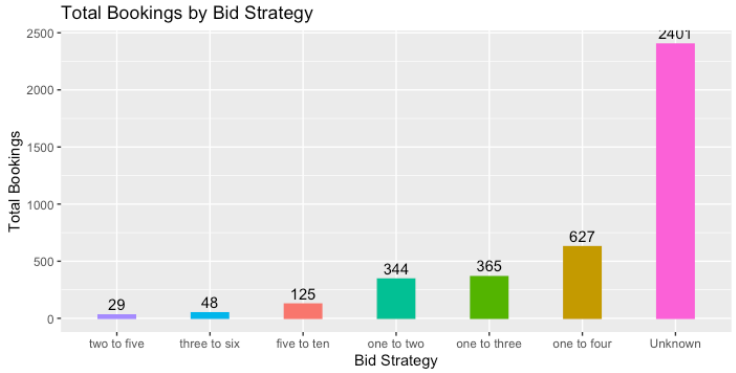


```
##sum by bid strategy
#change the name in text
org_df$bid_strat_name <- gsub(0,"Unknown",org_df$bid_strat_cat)
org_df$bid_strat_name <- gsub(1,"five to ten",org_df$bid_strat_name)
org_df$bid_strat_name <- gsub(2,"three to six",org_df$bid_strat_name)
org_df$bid_strat_name <- gsub(3,"two to five",org_df$bid_strat_name)
org_df$bid_strat_name <- gsub(4,"one to four",org_df$bid_strat_name)
org_df$bid_strat_name <- gsub(5,"one to three",org_df$bid_strat_name)
org_df$bid_strat_name <- gsub(6,"one to two",org_df$bid_strat_name)
sum_bid_str<-aggregate(cbind(revenue, total_cost, total_booking, clicks) ~ bid_strat_name, data = org_df, FUN=sum, na.rm=TRUE)

sum_bid_str$profit <- sum_bid_str$revenue - sum_bid_str$total_cost
sum_bid_str$profit_margin <- round((sum_bid_str$profit/sum_bid_str$revenue)*100,2)

##Average return on investment
sum_bid_str$roi<-round(sum_bid_str$profit/sum_bid_str$total_cost,2)

##Average cost per booking
sum_bid_str$avg_cpb <- round(sum_bid_str$total_cost/sum_bid_str$total_booking,2)

##Conversion rate
sum_bid_str$convers_rate <- round((sum_bid_str$total_booking/sum_bid_str$clicks)*100,2)
sum_bid_str$avg_cpc <- round(sum_bid_str$total_cost/sum_bid_str$clicks,2)

##Bid Strategy - Profit Margin
ggplot(data = sum_bid_str, aes(reorder(x=bid_strat_name,profit_margin),y=profit_margin,color=bid_strat_name,fill=bid_strat_name))
+
  geom_bar(stat="identity", width=0.4) + labs(x="Bid Strategy") + labs(y="Profit Margin") + ggtitle("Profit Margin by Bid Strategy")
+geom_text(aes(label = profit_margin), vjust = -0.5, colour = "black") +theme(legend.position="none")
```



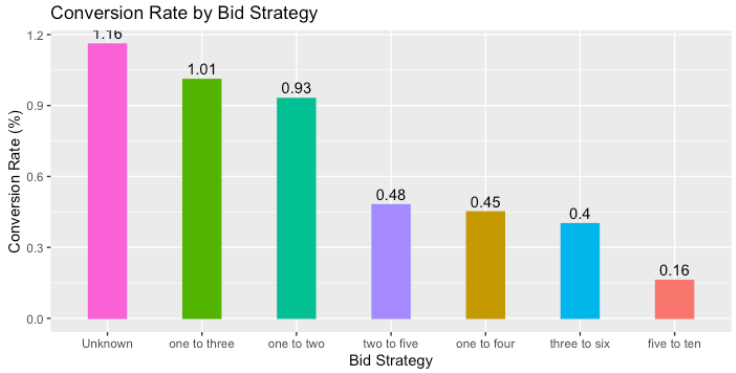
#Bid Strategy - Total Bookings

```
ggplot(data = sum_bid_str, aes(reorder(x=bid_strat_name, total_booking),y=total_booking,color=bid_strat_name,fill=bid_strat_name))
+
  geom_bar(stat="identity", width=0.4) + labs(x="Bid Strategy") + labs(y="Total Bookings") + ggtitle("Total Bookings by Bid Strategy")
+geom_text(aes(label = total_booking), vjust = -0.5, colour = "black") +theme(legend.position="none")
```



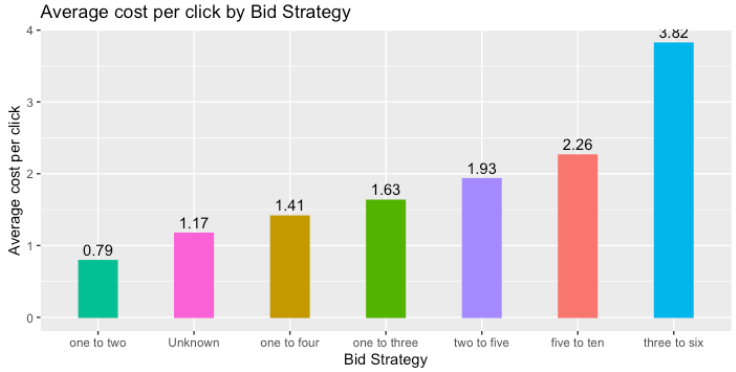
#Bid Strategy - Conversion Rate

```
ggplot(data = sum_bid_str, aes(reorder(x=bid_strat_name, -convers_rate),y=convers_rate,color=bid_strat_name,fill=bid_strat_name))
+
  geom_bar(stat="identity", width=0.4) + labs(x="Bid Strategy") + labs(y="Conversion Rate (%)") + ggtitle("Conversion Rate by Bid
Strategy") +geom_text(aes(label = convers_rate), vjust = -0.5, colour = "black") +theme(legend.position="none")
```



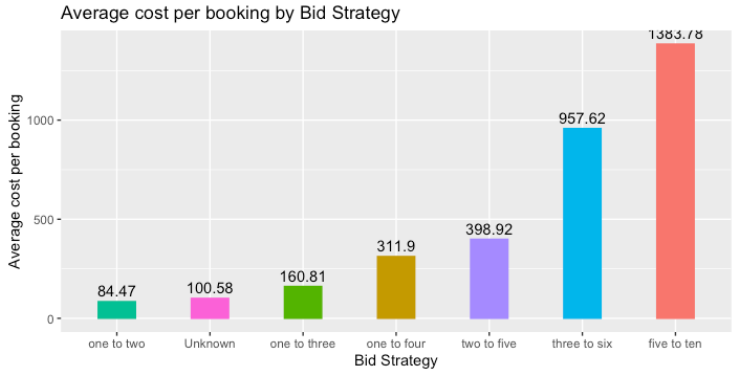
##Bid Strategy - Average Cost per Click

```
ggplot(data = sum_bid_str, aes(reorder(x=bid_strat_name, avg_cpc),y=avg_cpc,color=bid_strat_name,fill=bid_strat_name)) +
  geom_bar(stat="identity", width=0.4) + labs(x="Bid Strategy") + labs(y="Average cost per click") + ggtitle("Average cost per click by Bid Strategy") +geom_text(aes(label = avg_cpc), vjust = -0.5, colour = "black") +theme(legend.position="none")
```



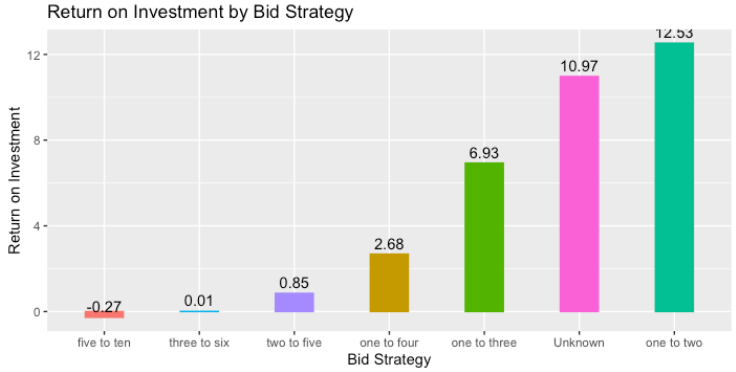
##Bid Strategy - Average Cost per Book

```
ggplot(data = sum_bid_str, aes(reorder(x=bid_strat_name, avg_cpb),y=avg_cpb,color=bid_strat_name,fill=bid_strat_name)) +
  geom_bar(stat="identity", width=0.4) + labs(x="Bid Strategy") + labs(y="Average cost per booking") + ggtitle("Average cost per booking by Bid Strategy") +geom_text(aes(label = avg_cpb), vjust = -0.5, colour = "black") +theme(legend.position="none")
```

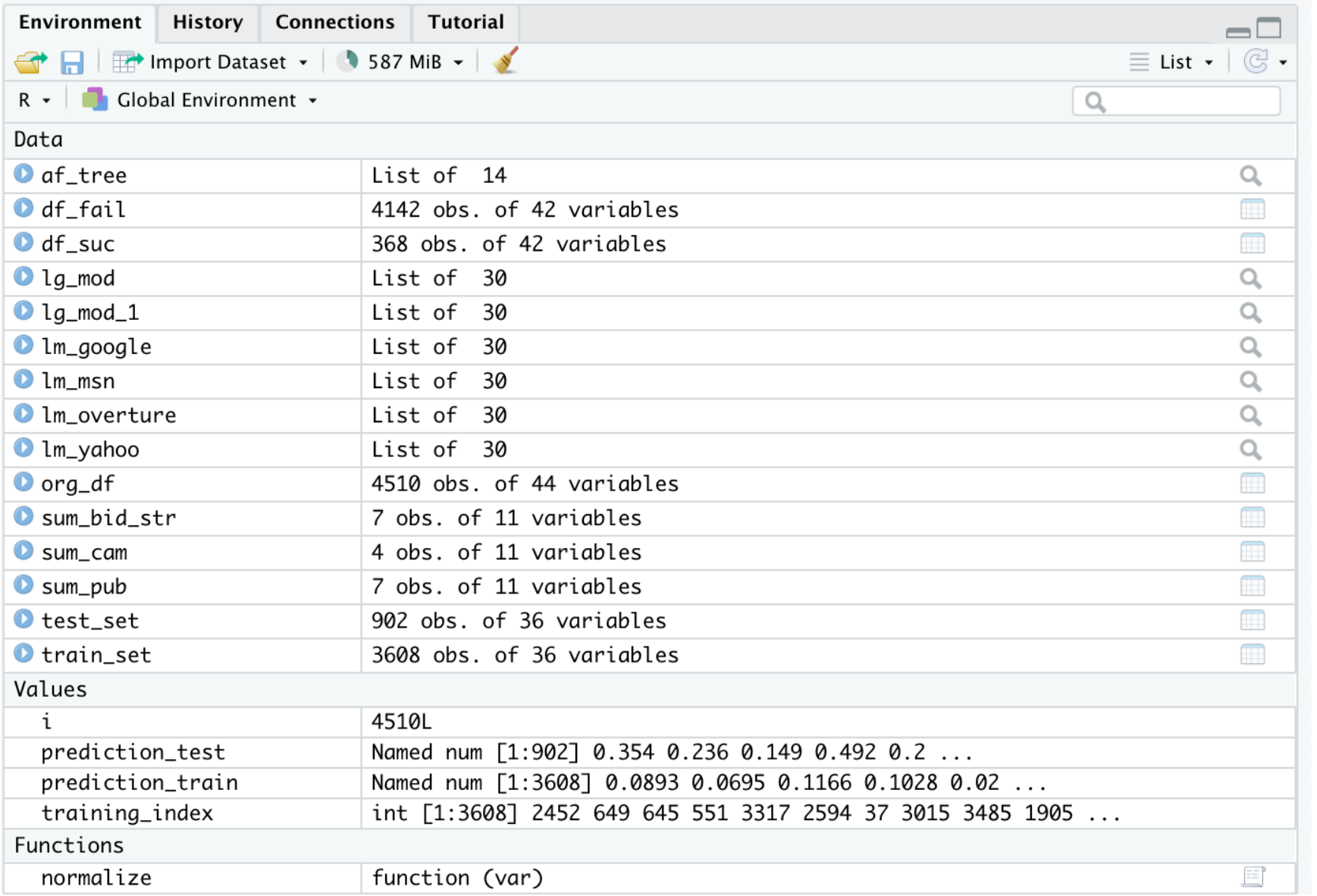


##Bid Strategy - Return on Investment

```
ggplot(data = sum_bid_str, aes(reorder(x=bid_strat_name, roi),y=roi,color=bid_strat_name,fill=bid_strat_name)) +
  geom_bar(stat="identity", width=0.4) + labs(x="Bid Strategy") + labs(y="Return on Investment") + ggtitle("Return on Investment by Bid Strategy") +geom_text(aes(label = roi), vjust = -0.5, colour = "black") +theme(legend.position="none")
```



Final Environment


Environment
History
Connections
Tutorial
Import Dataset
587 MiB
List
R
Global Environment
Data
af_tree
List of 14
df_fail
4142 obs. of 42 variables
df_suc
368 obs. of 42 variables
lg_mod
List of 30
lg_mod_1
List of 30
lm_google
List of 30
lm_msn
List of 30
lm_overture
List of 30
lm_yahoo
List of 30
org_df
4510 obs. of 44 variables
sum_bid_str
7 obs. of 11 variables
sum_cam
4 obs. of 11 variables
sum_pub
7 obs. of 11 variables
test_set
902 obs. of 36 variables
train_set
3608 obs. of 36 variables
Values
i
4510L
prediction_test
Named num [1:902] 0.354 0.236 0.149 0.492 0.2 ...
prediction_train
Named num [1:3608] 0.0893 0.0695 0.1166 0.1028 0.02 ...
training_index
int [1:3608] 2452 649 645 551 3317 2594 37 3015 3485 1905 ...
Functions
normalize
function (var)