Team 16

Albina Kelmendi
Belinda Darko
Shanty Behl
Xavier Chauvris

Elements Presented

```
#Importing and viewing the data
library(readxl)
air_france <- read_excel("C:/Users/xavie/Downloads/Air France Case Spreadsheet Supplement.xls",
                         sheet = 2)
#Viewing Results
View(air_france)

################################################
# Exploring the data set
################################################

# Displaying the characteristics of each feature
summary(air_france)


#Checking the number of missing values in the table
apply(X = is.na(air_france), MARGIN = 2, FUN = sum) #Bid Strategy has 1224 missing values

#Counting the number of "N/A" values on the whole data frame
#install.packages("stringr")
#library(stringr)

#Counting number of N/A values per vector
sum(str_count(air_france$`Publisher ID`, "N/A"))
sum(str_count(air_france$`Publisher Name`, "N/A"))
sum(str_count(air_france$`Keyword ID`, "N/A"))
sum(str_count(air_france$Keyword, "N/A"))
sum(str_count(air_france$`Match Type`, "N/A")) #Column has a category N/A with 48 observations
sum(str_count(air_france$Campaign, "N/A"))
sum(str_count(air_france$`Keyword Group`, "N/A"))
sum(str_count(air_france$Category, "N/A"))
sum(str_count(air_france$`Keyword Type`, "N/A"))
```

```
sum(str_count(air_france$Status, "N/A"))



################################################
# Massaging the data set
################################################
#Publisher name and ID are the same
#Analyzing number of categories
table(air_france$`Publisher Name`)

#Creating new factors with numerical values
summary(air_france$`Publisher Name`) #Verifying type of data
air_france$publisher <- air_france$`Publisher Name` #Creating new column with existing data
air_france$publisher[air_france$publisher == "Google - Global"] <- 1
air_france$publisher[air_france$publisher == "Google - US"] <- 2
air_france$publisher[air_france$publisher == "MSN - Global"] <- 3
air_france$publisher[air_france$publisher == "MSN - US"] <- 4
air_france$publisher[air_france$publisher == "Overture - Global"] <- 5
air_france$publisher[air_france$publisher == "Overture - US"] <- 6
air_france$publisher[air_france$publisher == "Yahoo - US"] <- 7

#Verifying new column
table(air_france$`Publisher Name`)
table(air_france$publisher)

#Converting publisher into numeric
air_france$publisher <- as.numeric(air_france$publisher)



#Converting Match Type categories
#Number of category
table(air_france$`Match Type`)

# Creating new column
air_france$match_cat <- air_france$`Match Type`

# Reassigning values
air_france$match_cat[air_france$match_cat == "N/A"] <- 0
air_france$match_cat[air_france$match_cat == "Advanced"] <- 1
air_france$match_cat[air_france$match_cat == "Broad"] <- 2
air_france$match_cat[air_france$match_cat == "Exact"] <- 3
air_france$match_cat[air_france$match_cat == "Standard"] <- 4
```

```
#Converting to numeric
air_france$match_cat <- as.numeric(air_france$match_cat)


table(air_france$Campaign)
table(air_france$`Keyword Group`)
table(air_france$Category)
table(air_france$`Bid Strategy`)

# Creating new column
 air_france$has_bid_strat <- air_france$`Bid Strategy`

# #Reassigning values
 air_france$has_bid_strat[air_france$has_bid_strat == "Pos 3-6"] <- 1
 air_france$has_bid_strat[air_france$has_bid_strat == "Position 1- 3"] <- 1
 air_france$has_bid_strat[air_france$has_bid_strat == "Position 1-2 Target"] <- 1
 air_france$has_bid_strat[air_france$has_bid_strat == "Position 1 -2 Target"] <- 1
 air_france$has_bid_strat[air_france$has_bid_strat == "Position 1-4 Bid Strategy"] <- 1
 air_france$has_bid_strat[air_france$has_bid_strat == "Postiion 1-4 Bid Strategy"] <- 1
 air_france$has_bid_strat[air_france$has_bid_strat == "Position 2-5 Bid Strategy"] <- 1
 air_france$has_bid_strat[air_france$has_bid_strat == "Position 5-10 Bid Strategy"] <- 1
 air_france$has_bid_strat[is.na(air_france$`Bid Strategy`)] <- 0

#Checking results
table(air_france$has_bid_strat)

#Converting vector as numeric
air_france$has_bid_strat<- as.numeric(air_france$has_bid_strat)

# Checking the number of category for remaining character variables
table(air_france$`Keyword Type`)
table(air_france$Status)

#Creating new column
air_france$status_int <- air_france$Status
air_france$status_int[air_france$status_int == "Deactivated"] <- 1
air_france$status_int[air_france$status_int == "Live"] <- 2
air_france$status_int[air_france$status_int == "Paused"] <- 3
air_france$status_int[air_france$status_int == "Sent"] <- 4
air_france$status_int[air_france$status_int == "Unavailable"] <- 5

#Checking results
table(air_france$status_int)
```

```
#Converting vector as a numeric
air_france$status_int <- as.numeric(air_france$status_int)


# Chekcking final result
View(air_france)


#Existing Metrics for the campaign
# Average Cost per Click (CPC)
# Engine Click Through Rate (CTR)
# Transaction Conversion Rate (TCR)

# Creating Net Revenue Variable
air_france$net_revenue <- air_france$Amount - air_france$`Total Cost`

#checking results and missing values
View(air_france)
sum(is.na(air_france$net_revenue))


# After a quick scan, we see observation 338 is an outlier that will affect the data
# 1 observation remove out of 4510 will not have any impact on the final result
# Removing observation 338
air_france <- air_france[-338,]

# Checking Result
View(air_france)



################################################
# UDF
################################################
# Normalizing UDF
norm <- function(x){
  normalized <- (x-min(x))/(max(x)-min(x))
  return(normalized)
}#Closing loop

# Return On Advertising UDF (ROA)
roa <- function(a,b,c){
  revenues <- sum(b)
```

```
 costs <- sum(c)
 ROA <- (revenues)/costs
 return(ROA)
}#closing loop

################################################
# ROA for each Publisher
################################################
#Yahoo
yahoo_roa <- round(roa(a=air_france$`Publisher Name`[which(air_france$`Publisher Name`=="Yahoo -
US")],
            b=air_france$Amount[which(air_france$`Publisher Name`=="Yahoo - US")],
            c=air_france$`Total Cost`[which(air_france$`Publisher Name`=="Yahoo - US")]),2)

#Google US
google_us_roa <- round(roa(a=air_france$`Publisher Name`[which(air_france$`Publisher
Name`=="Google - US")],
              b=air_france$Amount[which(air_france$`Publisher Name`=="Google - US")],
              c=air_france$`Total Cost`[which(air_france$`Publisher Name`=="Google - US")]),2)


#Google Global
google_global_roa <- round(roa(a=air_france$`Publisher Name`[which(air_france$`Publisher
Name`=="Google - Global")],
                b=air_france$Amount[which(air_france$`Publisher Name`=="Google - Global")],
                c=air_france$`Total Cost`[which(air_france$`Publisher Name`=="Google - Global")]),2)


#MSN Global
msn_global_roa <- round(roa(a=air_france$`Publisher Name`[which(air_france$`Publisher
Name`=="MSN - Global")],
                  b=air_france$Amount[which(air_france$`Publisher Name`=="MSN - Global")],
                  c=air_france$`Total Cost`[which(air_france$`Publisher Name`=="MSN - Global")]),2)


#MSN US
msn_us_roa <- round(roa(a=air_france$`Publisher Name`[which(air_france$`Publisher Name`=="MSN -
US")],
                  b=air_france$Amount[which(air_france$`Publisher Name`=="MSN - US")],
                  c=air_france$`Total Cost`[which(air_france$`Publisher Name`=="MSN - US")]),2)



#Overture Global
```

```
overture_global_roa <- round(roa(a=air_france$`Publisher Name`[which(air_france$`Publisher
Name`=="Overture - Global")],
                    b=air_france$Amount[which(air_france$`Publisher Name`=="Overture - Global")],
                    c=air_france$`Total Cost`[which(air_france$`Publisher Name`=="Overture -
Global")]),2)



#Overture US
overture_us_roa <- round(roa(a=air_france$`Publisher Name`[which(air_france$`Publisher
Name`=="Overture - US")],
                    b=air_france$Amount[which(air_france$`Publisher Name`=="Overture - US")],
                    c=air_france$`Total Cost`[which(air_france$`Publisher Name`=="Overture - US")]),2)


#Implementing everything in data frame
publisher_roa <- data.frame(google_global_roa, google_us_roa, yahoo_roa, overture_global_roa,
                  overture_us_roa, msn_global_roa, msn_us_roa)

#Transposing
publisher_roa_transpose = t(publisher_roa)

#Converting to a data frame
publisher_roa_transpose <- as.data.frame(publisher_roa_transpose)

#Renaming V1 column for readability purpose
colnames(publisher_roa_transpose)[colnames(publisher_roa_transpose)=="V1"] <- "ROA"
```

	ROA
google_global_roa	7.69
google_us_roa	4.93
yahoo_roa	19.10
overture_global_roa	6.69
overture_us_roa	2.45
msn_global_roa	11.97
msn_us_roa	11.28

```
################################################
# Net Revenue Per Click for Each Publisher
################################################
#ROA udf will for for this new variable
```

```
#Yahoo
yahoo_nrc <- round(roa(a=air_france$`Publisher Name`[which(air_france$`Publisher Name`=="Yahoo - US")],
                       b=air_france$net_revenue[which(air_france$`Publisher Name`=="Yahoo - US")],
                       c=air_france$Clicks[which(air_france$`Publisher Name`=="Yahoo - US")]),2)

#Google US
google_us_nrc <- round(roa(a=air_france$`Publisher Name`[which(air_france$`Publisher Name`=="Google - US")],
                           b=air_france$net_revenue[which(air_france$`Publisher Name`=="Google - US")],
                           c=air_france$Clicks[which(air_france$`Publisher Name`=="Google - US")]),2)


#Google Global
google_global_nrc <- round(roa(a=air_france$`Publisher Name`[which(air_france$`Publisher Name`=="Google - Global")],
                               b=air_france$net_revenue[which(air_france$`Publisher Name`=="Google - Global")],
                               c=air_france$Clicks[which(air_france$`Publisher Name`=="Google - Global")]),2)


#MSN Global
msn_global_nrc <- round(roa(a=air_france$`Publisher Name`[which(air_france$`Publisher Name`=="MSN - Global")],
                            b=air_france$net_revenue[which(air_france$`Publisher Name`=="MSN - Global")],
                            c=air_france$Clicks[which(air_france$`Publisher Name`=="MSN - Global")]),2)


#MSN US
msn_us_nrc <- round(roa(a=air_france$`Publisher Name`[which(air_france$`Publisher Name`=="MSN - US")],
                        b=air_france$net_revenue[which(air_france$`Publisher Name`=="MSN - US")],
                        c=air_france$Clicks[which(air_france$`Publisher Name`=="MSN - US")]),2)



#Overture Global
overture_global_nrc <- round(roa(a=air_france$`Publisher Name`[which(air_france$`Publisher Name`=="Overture - Global")],
                                 b=air_france$net_revenue[which(air_france$`Publisher Name`=="Overture - Global")],
                                 c=air_france$Clicks[which(air_france$`Publisher Name`=="Overture - Global")]),2)
```

```
#Overture US
overture_us_nrc <- round(roa(a=air_france$`Publisher Name`[which(air_france$`Publisher
Name`=="Overture - US")],
                    b=air_france$net_revenue[which(air_france$`Publisher Name`=="Overture - US")],
                    c=air_france$Clicks[which(air_france$`Publisher Name`=="Overture - US")]),2)


#Implementing everything in data frame
publisher_nrc <- data.frame(google_global_nrc, google_us_nrc, yahoo_nrc, overture_global_nrc,
                    overture_us_nrc, msn_global_nrc, msn_us_nrc)

#Transposing
publisher_nrc_transpose = t(publisher_nrc)

#Converting to a data frame
publisher_nrc_transpose <- as.data.frame(publisher_nrc_transpose)

#Renaming V1 column for readability purpose
colnames(publisher_nrc_transpose)[colnames(publisher_nrc_transpose)=="V1"] <- "NRC"
```

	NRC
yahoo_nrc	18.34
msn_us_nrc	15.31
msn_global_nrc	11.89
google_global_nrc	11.09
google_us_nrc	7.24
overture_global_nrc	6.01
overture_us_nrc	1.72

```
##################################################
# Correlation Table
##################################################

data.frame(colnames(air_france))

#Select only numerical features based on index
air_france_corr <- air_france[,c(12:23)]

#Display Pearson correlation table
cor(air_france_corr, air_france$net_revenue, method="pearson")
```

```
                                [,1]
Search Engine Bid          0.101726967
Clicks                     0.789419333
Click Charges              0.482530250
Avg. Cost per Click       -0.052656172
Impressions                0.069240486
Engine Click Thru %        0.011393196
Avg. Pos.                 -0.030183801
Trans. Conv. %             0.015157029
Total Cost/ Trans.         0.006181546
Amount                     0.996898609
Total Cost                 0.482530250
Total Volume of Bookings   0.992107817
```

```
################################################
# Renaming columns
################################################
#Renaming columns to fit into pivot tables
colnames(air_france)[colnames(air_france)=="net_revenue"] <- "netrevenue"
colnames(air_france)[colnames(air_france)=="Publisher Name"] <- "Publishername"
colnames(air_france)[colnames(air_france)=="Total Volume of Bookings"] <- "Bookings"
colnames(air_france)[colnames(air_france) == "Match Type"] <- "matchtype"
colnames(air_france)[colnames(air_france) == "Search Engine Bid"] <- "SEB"
colnames(air_france)[colnames(air_france) == "Click Charges"] <- "ClickCharges"
colnames(air_france)[colnames(air_france) == "Avg. Cost per Click"] <- "ACPC"
colnames(air_france)[colnames(air_france) == "Total Cost/ Trans."] <- "TCPT"
colnames(air_france)[colnames(air_france) == "Total Cost"] <- "TotalCost"
colnames(air_france)[colnames(air_france) == "Engine Click Thru %"] <- "ECTR"
colnames(air_france)[colnames(air_france) == "Trans. Conv. %"] <- "TCR"
```

```
################################################
# Publishers Volume of Bookings
################################################
#install.packages('dplyr')
library(dplyr)

# Campaign performance by amount, impressions, and net revenue
campaign_perf <- air_france%>%
  group_by(Campaign)%>%
  select(Amount, Impressions, Campaign, netrevenue)%>%
  summarise(sum(Amount), sum(Impressions), sum(netrevenue))
```

	Campaign	sum(Amount)	sum(Impressions)	netrevenue
1	Air France Branded	2349870.90	1263310	2206793.0758
2	Air France Brand & French Destinations	788641.90	573159	701627.3630
3	Unassigned	777517.95	34961215	571246.0134
4	Air France Global Campaign	467981.95	1540074	405922.5002
5	Paris & France Terms	136393.55	658281	33796.1753
6	Western Europe Destinations	42103.90	588324	5921.5376
7	Geo Targeted New York	35580.15	73166	26040.4750
8	Google_Yearlong 2006	22373.70	1803463	-59585.7873
9	Geo Targeted Chicago	7144.25	12690	5562.5750
10	Geo Targeted Houston	7065.20	25338	5072.4125
11	French Destinations	6223.70	59351	-2208.0375
12	Geo Targeted DC	5191.80	19875	3920.4375
13	Geo Targeted San Francisco	3822.45	25788	1733.0375
14	Geo Targeted Seattle	2817.75	5458	1643.5000
15	Geo Targeted Boston	2461.60	16137	1065.4250
16	Geo Targeted Los Angeles	2183.65	18633	383.6625
17	General Terms	1977.95	144298	1371.4750
18	Geo Targeted Detroit	923.95	9648	204.9250
19	Geo Targeted Miami	470.05	7219	-168.4750
20	Geo Targeted Philadelphia	434.35	7643	-311.6250
21	Geo Targeted Atlanta	170.00	4428	-95.2500
22	Business Class	144.50	41878	-3124.5500
23	Geo Targeted Cincinnati	0.00	676	-33.7500
24	Outside Western Europe	0.00	8622	-597.8375

```
#Amount generated from each publisher
 publisher_rev <- air_france%>%
   group_by(Publishername)%>%
   select(netrevenue, Publishername)%>%
  summarise(sum(netrevenue))
```

	Publishername	sum(netrevenue)
1	Google - US	1391423.9
2	Yahoo - US	836091.1
3	Google - Global	808603.1
4	Overture - Global	365788.8
5	Overture - US	205457.2
6	MSN - US	165451.3
7	MSN - Global	133363.9

#Amount generated from each publisher

```
 publisher_cost <- air_france%>%
   group_by(Publishername)%>%
   select(TCPT, Publishername)%>%
   summarise(mean(TCPT))
```

	Publishername	TCPT
1	Google - Global	70.896472
2	Overture - US	43.660687
3	Google - US	25.129017
4	Overture - Global	16.468699
5	MSN - Global	11.165817
6	Yahoo - US	7.956570
7	MSN - US	5.157714

#Best match type by net revenue

```
match_netrev <- air_france%>%
  group_by(matchtype)%>%
  select(matchtype, netrevenue)%>%
  summarise(sum(netrevenue))
```

	matchtype	netrevenue
1	Broad	1511034.276
2	Advanced	1139786.538
3	Exact	984339.625
4	Standard	267550.600
5	N/A	3468.238

```
#Best match type by number of bookings
match_booking <- air_france%>%
 group_by(matchtype)%>%
 select(matchtype, Bookings)%>%
 summarise(sum(Bookings))
```

	matchtype	Bookings
1	Broad	1763
2	Advanced	987
3	Exact	850
4	Standard	336
5	N/A	2

```
#Volume of Bookings for each publisher
booking_perf <- air_france%>%
 group_by(Publishername)%>%
 select(Bookings, Publishername)%>%
 summarise(sum(Bookings))
```

	Publishername	Bookings
1	Google - US	1549
2	Google - Global	797
3	Yahoo - US	662
4	Overture - Global	372
5	Overture - US	289
6	MSN - US	140
7	MSN - Global	129

```
#Net revenue generated from each publisher
nrev_perf <- air_france%>%
 group_by(Publishername)%>%
 select(netrevenue, Publishername)%>%
 summarise(mean(netrevenue))
```

	Publishername	mean(netrevenue)
1	Google - Global	2057.5142
2	MSN - US	1688.2787
3	MSN - Global	1347.1100
4	Yahoo - US	1316.6789
5	Google - US	672.1854
6	Overture - Global	661.4626
7	Overture - US	310.8278

```
#Number of impressions for each publisher
impression_perf <- air_france%>%
  group_by(Publishername)%>%
  select(Impressions, Publishername)%>%
  summarise(mean(Impressions))
```

	Publishername	mean(Impressions)
1	Overture - Global	32366.595
2	Overture - US	25813.144
3	Google - Global	4601.338
4	Google - US	1862.652
5	MSN - US	1735.918
6	Yahoo - US	1469.835
7	MSN - Global	1413.929

```
#Total Cost per transaction, and average cost per click for each publisher
publisher_click <- air_france%>%
  group_by(Publishername)%>%
  select(Publishername, TCPT, ACPC)%>%
  summarise(mean(TCPT), mean(ACPC))
```

	Publishername	mean(TCPT)	mean(ACPC)
1	MSN - US	5.157714	2.8674701
2	Google - US	25.129017	2.3850934
3	Google - Global	70.896472	2.2249594
4	MSN - Global	11.165817	2.1529982
5	Yahoo - US	7.956570	1.9988757
6	Overture - Global	16.468699	0.8047588
7	Overture - US	43.660687	0.7639206

```
################################################
# Normalization of features
################################################
# Normalizing all numeric features
# Clicks
air_france$clicks_norm <- norm(x=air_france$Clicks)

# Clicks charges
air_france$clicks_charge_norm <- norm(x=air_france$ClickCharges)

# Average Cost Per Click
air_france$avg_cpc_norm <- norm(x=air_france$ACPC)

# Impressions
air_france$impressions_norm <- norm(x=air_france$Impressions) #

# Engine Click Thru Rate
air_france$ectr_norm <- norm(x=air_france$ECTR)

# Avg Pos
air_france$avg_pos_norm <- norm(x=air_france$`Avg. Pos.`)

# Trans. COnv. Rate
air_france$trans_conv_norm <- norm(x=air_france$TCR)

# Total cost/trans
air_france$cost_trans_norm <- norm(x=air_france$TCPT)

# Amount
air_france$amount_norm <- norm(x=air_france$Amount)

# Total Cost
air_france$tot_cost_norm <- norm(x=air_france$TotalCost)

# Total Volume of Bookings
air_france$tot_booking_norm <- norm(x=air_france$Bookings)
```

```
################################################
# Logistic Regression
################################################
air_france$binom_roa<- c() #assigning empty vector to new object

#Creating udf to create binomial variable
for(i in 1:nrow(air_france)){
 if(air_france$netrevenue[i]>0){
  air_france$binom_netrevenue[i] <- "1"
 }
 else {air_france$binom_netrevenue[i] <- "0"}
}#Closing udf

#Converting binomial variable to a numeric
air_france$binom_netrevenue <- as.numeric(air_france$binom_netrevenue)

#Creating training index
train_index <- sample(1:nrow(air_france), size=0.8*nrow(air_france))


air_france_train <- air_france[train_index,] #Train data set
air_france_test <- air_france[-train_index,] #Test data set

#Running logit regression
air_france_logit <- glm(binom_netrevenue ~ Clicks+Impressions+TotalCost+TCPT,
                        data=air_france_train, family = "binomial")


#Checking result of regression
summary(air_france_logit)
```

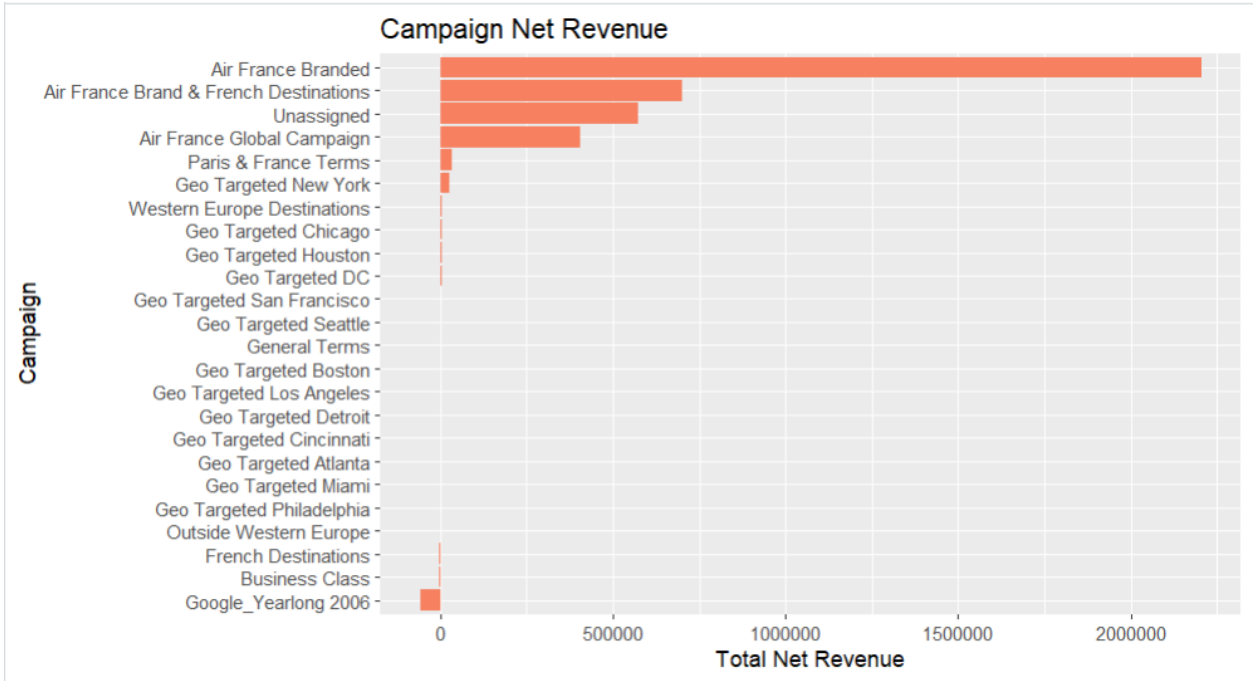
```
Call:
glm(formula = binom_netrevenue ~ Clicks + Impressions + TotalCost +
    TCPT, family = "binomial", data = air_france_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-6.8877  -0.3265  -0.3232  -0.3225   2.4765

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.933e+00  7.680e-02 -38.188  < 2e-16 ***
Clicks       4.527e-03  5.625e-04   8.049 8.35e-16 ***
Impressions -3.672e-06  6.870e-07  -5.345 9.03e-08 ***
TotalCost   -1.001e-03  2.272e-04  -4.406 1.05e-05 ***
TCPT         5.041e-04  2.534e-04   1.990   0.0466 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1816.8  on 3606  degrees of freedom
Residual deviance: 1557.4  on 3602  degrees of freedom
AIC: 1567.4

Number of Fisher Scoring iterations: 7
```

```
################################################
# Normalized Logistic Regression
################################################

#Running logit regression
air_france_logit_norm <- glm(binom_netrevenue ~ clicks_norm+impressions_norm+tot_cost_norm+
                        cost_trans_norm, data=air_france_train, family = "binomial")

#Checking results
summary(air_france_logit_norm)
```

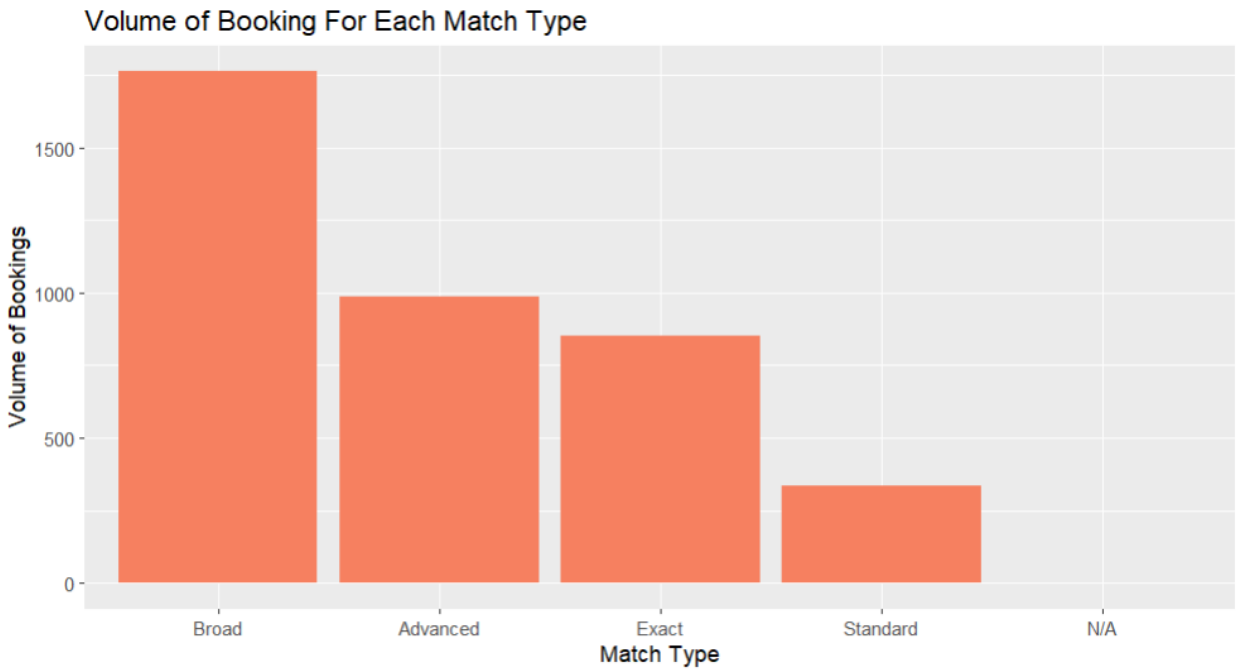
```
Call:
glm(formula = binom_netrevenue ~ clicks_norm + impressions_norm +
    tot_cost_norm + cost_trans_norm, family = "binomial", data = air_france_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-6.8877  -0.3265  -0.3232  -0.3225   2.4765

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       -2.92824    0.07666 -38.196  < 2e-16 ***
clicks_norm      153.98435   19.13082   8.049 8.35e-16 ***
impressions_norm -30.63395    5.73100  -5.345 9.03e-08 ***
tot_cost_norm    -46.23852   10.49412  -4.406 1.05e-05 ***
cost_trans_norm    4.83822    2.43159   1.990   0.0466 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1816.8  on 3606  degrees of freedom
Residual deviance: 1557.4  on 3602  degrees of freedom
AIC: 1567.4

Number of Fisher Scoring iterations: 7
```

```
################################################
# Visualization Libraries
################################################
#Importing library
library(ggplot2)

#install.packages("RColorBrewer")
library("RColorBrewer")
display.brewer.all(colorblindFriendly = FALSE)

#install.packages('dplyr')
library(dplyr)

#install.packages('forcats')
library(forcats)

#install.packages('viridis')
library(viridis)
```

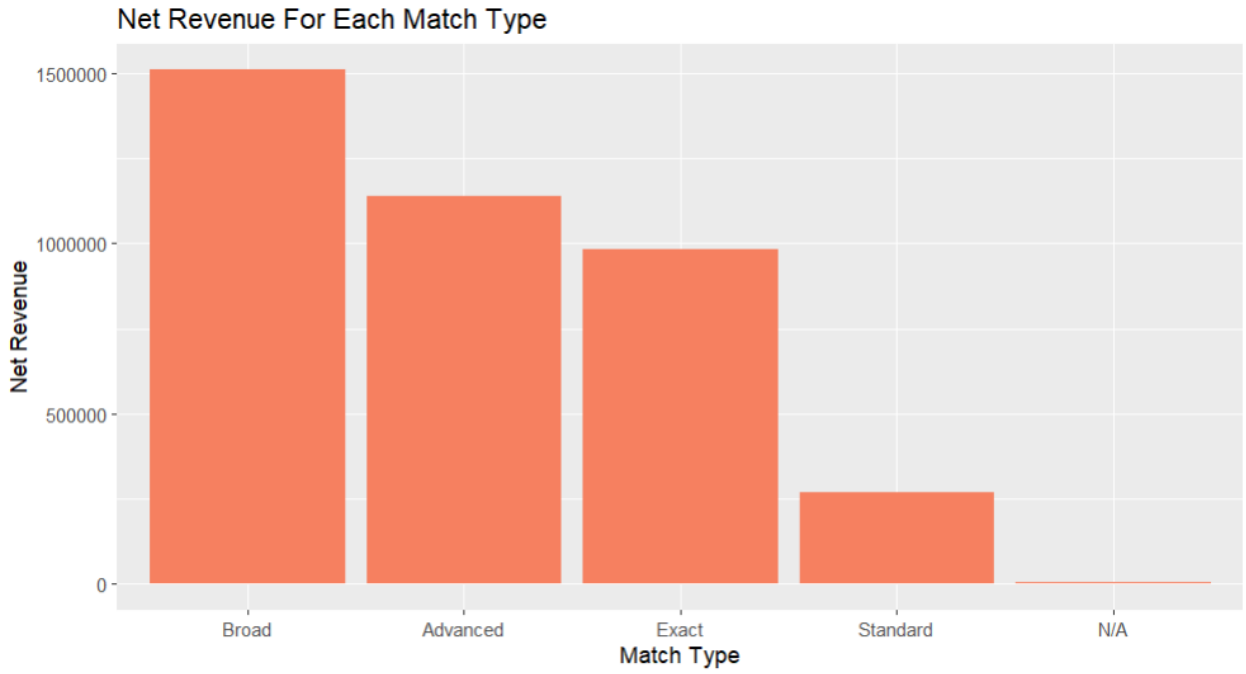
```
#install.packages('hrbrthemes')
library(hrbrthemes)


############################
# Campaign Net Revenue
############################
#Checking columns name
colnames(campaign_perf)

#Changing names
colnames(campaign_perf)[colnames(campaign_perf)=="sum(netrevenue)"] <- "netrevenue"

#Checking Results
colnames(campaign_perf)

#Calling bar graph
campaign_perf%>%                                    #calling data frame to use
 mutate(Campaign = fct_reorder(Campaign, netrevenue))%>%    #ordering data
 ggplot( aes(x=(netrevenue), y=as.factor(Campaign)))+      #assigning variables to axis
 geom_bar(stat="identity", fill="#f68060")+              #defining type of bar chart
 ggtitle("Campaign Net Revenue")+                        #graph title
 xlab("Total Net Revenue")+                              #x axis label
 ylab("Campaign")                                        #y axis label
```



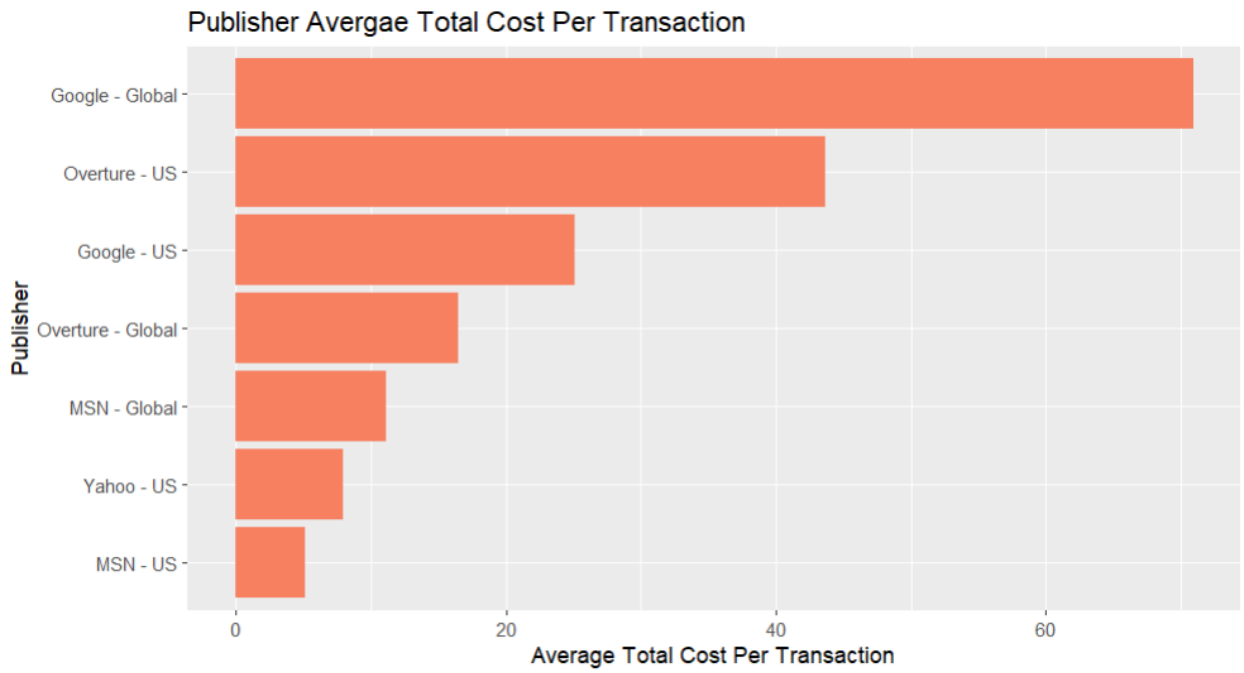
```
############################
# Match type Bookings
############################
#Checking columns name
colnames(match_booking)

#Changing columns name
colnames(match_booking)[colnames(match_booking)=="sum(Bookings)"] <- "Bookings"

#Building bar chart
match_booking%>%                                          #Data frame to take values from
 mutate(matchtype = fct_reorder(matchtype, -Bookings))%>%          #Ordering values
 ggplot( aes(x=(Bookings), y=as.factor(matchtype)))+geom_bar(stat="identity",
                                    fill="#f68060")+   #Chart characteristics
 coord_flip()+                                        #Reversing order
 ggtitle("Volume of Booking For Each Match Type")+                 #Chart title
 xlab("Volume of Bookings")+                                 #X axis title
 ylab("Match Type")                                  #Y axis title
```



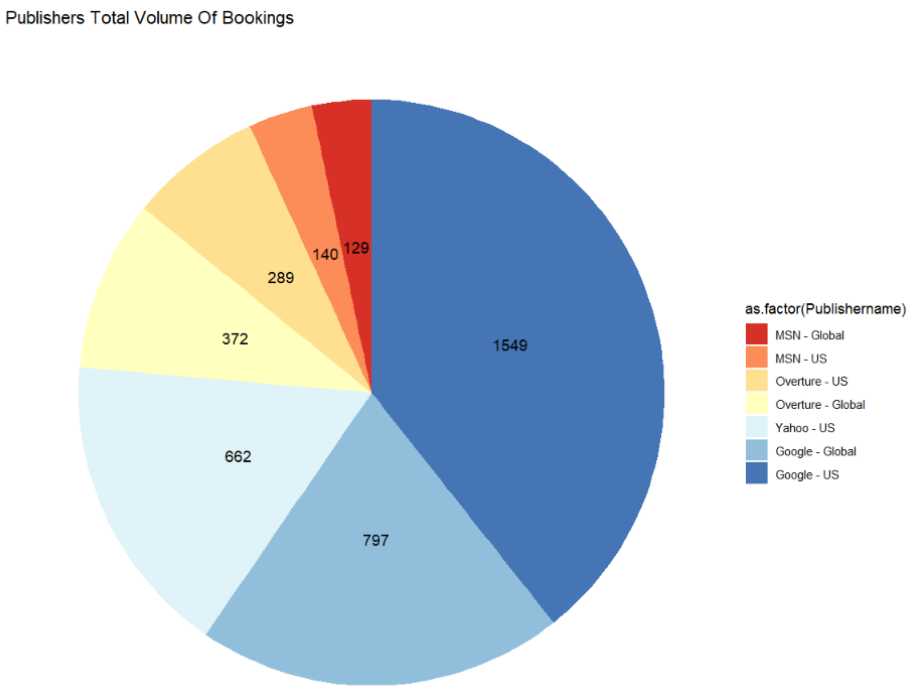
```
############################
# Match type Net Revenue
############################
#Checking columns name
colnames(match_netrev)

#Changing columns name
colnames(match_netrev)[colnames(match_netrev)=="sum(netrevenue)"] <- "netrevenue"

#Building bar chart
match_netrev%>%                                          #Data frame to take values from
 mutate(matchtype = fct_reorder(matchtype, -netrevenue))%>%          #Ordering values
 ggplot( aes(x=(netrevenue), y=as.factor(matchtype)))+geom_bar(stat="identity",
                                    fill="#f68060")+ #Chart characteristics
 coord_flip()+                                       #Reversing order
 ggtitle("Net Revenue For Each Match Type")+                  #Chart title
 xlab("Net Revenue")+                                 #X axis title
 ylab("Match Type")                                  #Y axis title
```



```
############################
# Publisher TCPT
############################
#Checking columns name
colnames(publisher_cost)

#Changing columns name
colnames(publisher_cost)[colnames(publisher_cost)=="mean(TCPT)"] <- "TCPT"

#Checking results
colnames(publisher_cost)

#Building bar chart
publisher_cost%>%                                  #calling data frame to use from
 mutate(Publishername = fct_reorder(Publishername, TCPT))%>%  #ordering data
 ggplot( aes(x=(TCPT), y=as.factor(Publishername)))+        #assigning variables to axis
 geom_bar(stat="identity", fill="#f68060")+                #defining type of bar chart
 ggtitle("Publisher Avergae Total Cost Per Transaction ")+    #graph title
 xlab("Average Total Cost Per Transaction")+                #x axis label
 ylab("Publisher")                                  #y axis label
```



```
############################
# Publisher Bookings
############################
#Checking columns name
colnames(booking_perf)

#Changing columns name
colnames(booking_perf)[colnames(booking_perf)=="sum(Bookings)"] <- "Bookings"

#Checking results
colnames(booking_perf)

#Pie chart
booking_perf%>%                                          #Data frame to take values from
 mutate(Publishername = fct_reorder(Publishername, Bookings))%>%          #Order values
 ggplot(aes(x="", y=Bookings, fill=as.factor(Publishername)))+            #Specification of feature
 geom_bar(stat="identity", width=1)+                                      #Configuration of bar chart to fit with
pie chart
 coord_polar("y", start=0)+                                               #Creating pie chart
 theme_void()+                                                          #Deleting potential background noise
 geom_text(aes(y=Bookings, label=Bookings),position = position_stack(vjust = .5),
        color="black", size=4)+                                           #Position and characteristics of legend
 scale_fill_brewer(palette="RdYlBu")+                                     #Colors of pie chart
 ggtitle("Publishers Total Volume Of Bookings")                           #Pie chart title
```



Environment

Data	
air_france	4509 obs. of 40 variable
air_france_corr	4509 obs. of 12 variable
air_france_logit	List of 30
air_france_logit_norm	List of 30
air_france_test	902 obs. of 40 variables
air_france_train	3607 obs. of 40 variable
booking_perf	7 obs. of 2 variables
campaign_perf	24 obs. of 4 variables
impression_perf	7 obs. of 2 variables
match_booking	5 obs. of 2 variables
match_netrev	5 obs. of 2 variables
nrev_perf	7 obs. of 2 variables
publisher_click	7 obs. of 3 variables
publisher_cost	7 obs. of 2 variables
publisher_nrc	1 obs. of 7 variables
publisher_nrc_transpose	7 obs. of 1 variable
publisher_rev	7 obs. of 2 variables
publisher_roa	1 obs. of 7 variables
publisher_roa_transpose	7 obs. of 1 variable

Values	
google_global_nrc	11.09
google_global_roa	7.69
google_us_nrc	7.24
google_us_roa	4.93
i	4509L
msn_global_nrc	11.89
msn_global_roa	11.97
msn_us_nrc	15.31
msn_us_roa	11.28
overture_global_nrc	6.01
overture_global_roa	6.69
overture_us_nrc	1.72
overture_us_roa	2.45
train_index	int [1:3607] 808 2561 2763 27 2479 2389 3996 1863 2501 3026 ...
yahoo_nrc	18.34
yahoo_roa	19.1
Functions	
norm	function (x)
roa	function (a, b, c)