Business Case: Air France

Data Science: R | Team 11 | MsBA1

12/17/2020

1.) Import of Libraries & Data Set

As a first step is important to import all the libraries as well as the data set to make the environment ready...

```
# Loading the libraries
library(readxl)
library(rstudioapi)
library(dplyr)
library(plotly)
library(pixiedust)
library(kableExtra)

# Setting working directory to source file
setwd(dirname(getActiveDocumentContext()$path))

# Reading the file
AF_df <- read_excel("_dataset/Air France Case Spreadsheet Supplement.xls",
                    sheet = "DoubleClick")
kayak_df <- read_excel("_dataset/Air France Case Spreadsheet Supplement.xls",
                    sheet = "Kayak",
                    range = "B8:H9")
```

2.) Data Massaging

As a second step we are going to massage our data by masking sure to be aware of missing values, ensuring consistency in your data and the proper structure of the different variables. Additionally our team calculated valuable additional KPIs which we could be derived from the given data.

```
# Replacing missing values with NA
AF_df[AF_df == "" ] <- NA

# Checking for missing values
sapply(AF_df, function(x) sum(is.na(x)))
```

```
##             Publisher ID           Publisher Name               Keyword ID
##                        0                        0                        0
##                  Keyword               Match Type                 Campaign
##                        0                        0                        0
##            Keyword Group                 Category            Bid Strategy
```

```
##                         0                         0                      1224
##              Keyword Type                    Status        Search Engine Bid
##                         0                         0                         0
##                    Clicks            Click Charges      Avg. Cost per Click
##                         0                         0                         0
##              Impressions      Engine Click Thru %                Avg. Pos.
##                         0                         0                         0
##           Trans. Conv. %       Total Cost/ Trans.                    Amount
##                         0                         0                         0
##               Total Cost Total Volume of Bookings
##                         0                         0
```

```
# Getting unique values for Publisher Name
publish_unique <- unique(AF_df$`Publisher Name`)

# Evaluating current unique bid strategy values
unique(AF_df$`Bid Strategy`)
```

```
## [1] NA                           "Position 2-5 Bid Strategy"
## [3] "Position 1- 3"              "Position 1-2 Target"
## [5] "Position 5-10 Bid Strategy" "Position 1-4 Bid Strategy"
## [7] "Position 1 -2 Target"       "Postiion 1-4 Bid Strategy"
## [9] "Pos 3-6"
```

```
# For loop to set bid strategies to a standardized format
for (i in 1:nrow(AF_df)) {
  if(grepl("1-2", AF_df$`Bid Strategy`[i]) == TRUE
     || grepl("1 -2", AF_df$`Bid Strategy`[i]) == TRUE
     || grepl("1- 2", AF_df$`Bid Strategy`[i]) == TRUE) {
    AF_df$`Bid Strategy`[i] <- "1-2"
  } else if (grepl("1-3", AF_df$`Bid Strategy`[i]) == TRUE
             || grepl("1 -3", AF_df$`Bid Strategy`[i]) == TRUE
             || grepl("1- 3", AF_df$`Bid Strategy`[i]) == TRUE) {
    AF_df$`Bid Strategy`[i] <- "1-3"
  } else if (grepl("1-4", AF_df$`Bid Strategy`[i]) == TRUE
             || grepl("1 -4", AF_df$`Bid Strategy`[i]) == TRUE
             || grepl("1- 4", AF_df$`Bid Strategy`[i]) == TRUE) {
    AF_df$`Bid Strategy`[i] <- "1-4"
  } else if (grepl("2-5", AF_df$`Bid Strategy`[i]) == TRUE
            || grepl("2 -5", AF_df$`Bid Strategy`[i]) == TRUE
            || grepl("2- 5", AF_df$`Bid Strategy`[i]) == TRUE) {
    AF_df$`Bid Strategy`[i] <- "2-5"
  } else if (grepl("3-6", AF_df$`Bid Strategy`[i]) == TRUE
             || grepl("3 -6", AF_df$`Bid Strategy`[i]) == TRUE
             || grepl("3- 6", AF_df$`Bid Strategy`[i]) == TRUE) {
    AF_df$`Bid Strategy`[i] <- "3-6"
  } else if (grepl("5-10", AF_df$`Bid Strategy`[i]) == TRUE
             || grepl("5 -10", AF_df$`Bid Strategy`[i]) == TRUE
             || grepl("5- 10", AF_df$`Bid Strategy`[i]) == TRUE) {
    AF_df$`Bid Strategy`[i] <- "5-10"
  } else if (is.na(AF_df$`Bid Strategy`[i]) == TRUE) {
    next
  } else {
```

```
    print("WARNING: There are Bid Strategies which weren't converted to a standardized format!")
  }
}

# Transforming bid strategies to a factor with 1 customized levels
AF_df$`Bid Strategy` <- factor(AF_df$`Bid Strategy`, levels = c("1-2", "1-3", "1-4", "2-5", "3-6", "5-1

# Re-evaluating unique bid strategy values
unique(AF_df$`Bid Strategy`)
```

```
## [1] <NA> 2-5  1-3  1-2  5-10 1-4  3-6
## Levels: 1-2 1-3 1-4 2-5 3-6 5-10
```

```
# Creating Transaction Conversion variable for Kayak
kayak_df <- within(data = kayak_df, `Trans. Conv. %` <- `Total Bookings`/Clicks)

# Creating a Profit Column
AF_df$Profit <- AF_df$Amount - AF_df$`Total Cost`
colnames(kayak_df)[7] <- "Profit"
# Creating a ROI Column
AF_df <- within(data = AF_df, ROI <- Profit/`Total Cost`)
kayak_df <- within(data = kayak_df, ROI <- Profit/`Media Cost`)
# Creating a ROA Column
AF_df <- within(data = AF_df, ROA <- Amount/`Total Cost`)
kayak_df <- within(data = kayak_df, ROA <- `Total Revenue`/`Media Cost`)


# Creating Columns for Profits per Booking / Cost per Booking / Probability of Booking
AF_df <- within(data = AF_df, Prof_Book <- Profit/`Total Volume of Bookings`)
kayak_df <- within(data = kayak_df, Prof_Book <- Profit/`Total Bookings`)
AF_df <- within(data = AF_df, Cost_Book <- `Total Cost`/`Total Volume of Bookings`)
kayak_df <- within(data = kayak_df, Cost_Book <- `Media Cost`/`Total Bookings`)
AF_df <- within(data = AF_df, Prob_of_Book <- `Trans. Conv. %` * `Engine Click Thru %`/100)

# Replacing infinite values for 0 in the dataset
AF_df[AF_df == "Inf"] <- 0
```

3.) Descriptive Statistics

```
# Creating a UDF to evaluate min, mean, standard deviation and maximum values
stats_udf <- function(x){
  my_min <- round(min(x, na.rm = T),2)
  my_mean <- round(mean(x, na.rm = T),2)
  my_sd <- round(sd(x, na.rm = T),2)
  my_max <- round(max(x, na.rm = T),2)
  return(c(my_min, my_mean, my_sd, my_max))
} # closing my_desc

# Obtaining descriptive statistics for Clicks, Impressions, Amount and Total Costs
Clicks_stats <- stats_udf(AF_df$Clicks)
Impressions_stats <- stats_udf(AF_df$Impressions)
```

```r
Amount_stats <- stats_udf(AF_df$Amount)
Tot_Cost_stats <- stats_udf(AF_df$`Total Cost`)

# Creating a dataframe with our descriptive statistics obtained
stats_df <- as.data.frame(cbind(Clicks_stats, Impressions_stats, Amount_stats, Tot_Cost_stats))
colnames(stats_df) <- c("Clicks", "Impressions", "Amount", "Total Cost")
rownames(stats_df) <- c("Minimum", "Mean", "Standard Deviation", "Maximum")

# Plotting the data frame
dust(stats_df) %>%
  sprinkle(round = 3) %>%
  kable() %>%
  kable_styling()
```

Clicks	Impressions	Amount	Total Cost
0	0	0	0
113.71	9283.52	1033.68	167.48
1062.71	169667.42	14940.37	1342.38
34012	8342415	567463.4	46188.44

```r
# Finding the Sales per Publisher
Sales <- c()
# for loop to find the Publisher Name and obtain the total sales per Publisher
for (i in 1:length(publish_unique)) {#start for loop
  Sales <- c(Sales,sum(AF_df$Amount[which(AF_df[,2] == publish_unique[i])]))
  i <- i + 1
} # end for loop

# Creating a matrix called of the Sales for each Publisher
Sales_Publisher <- data.frame(cbind(publish_unique,as.numeric(Sales)))
# Naming the columns with Publisher Names
colnames(Sales_Publisher)<- c("Publisher", "Sales")

# Plotting the data frame
dust(Sales_Publisher) %>%
  sprinkle(round = 3) %>%
  kable() %>%
  kable_styling()
```

Publisher	Sales
Yahoo - US	882288.95
MSN - Global	145524.25
Google - Global	929549.8
Overture - Global	430084.7
Google - US	1745481.8
Overture - US	347433.25
MSN - US	181549.8

4.) Visualizations

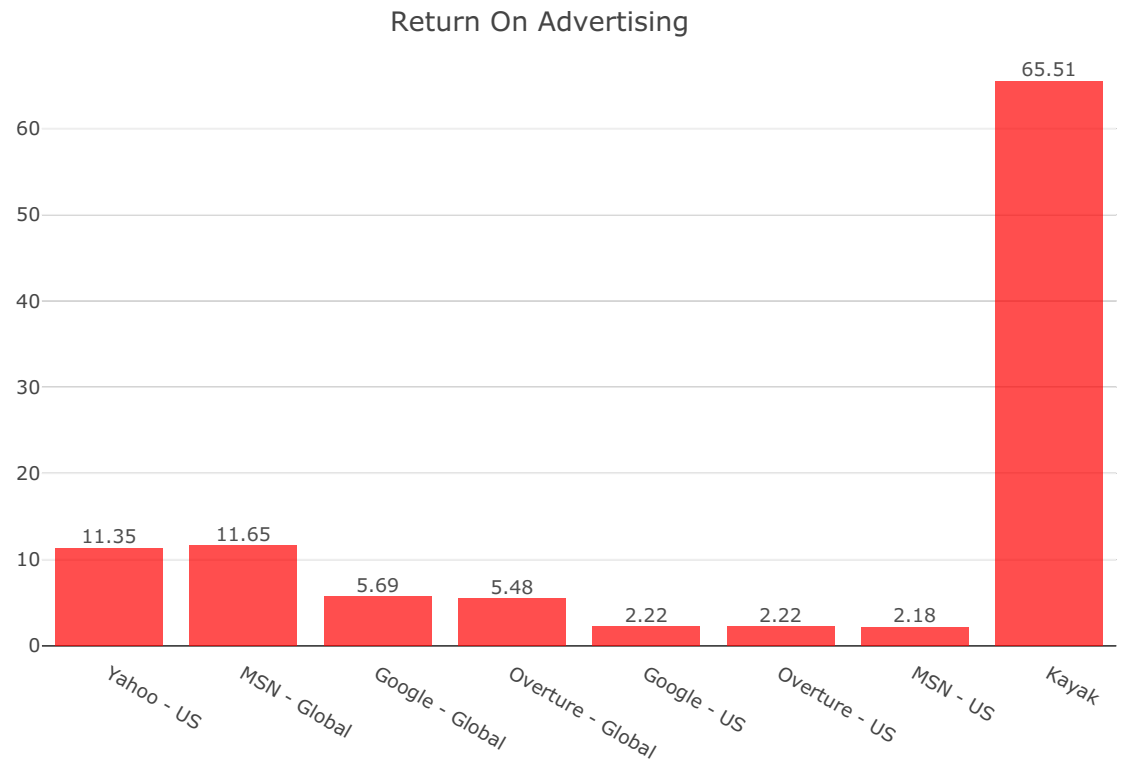
4.1.) Some Visualizations: Quick Overview about Publishers

This visualizations helped us to get a quick overview about the different publishers (for now excluding Kayak)

```
#creating the independent variable (Publisher Name) for the graph
publisher_names <- publish_unique
#creating the dependent variable (ROA) for the graph using a for loop
avg_roa <- c()
for (i in 1:length(publish_unique)) {
  mean_roa <- mean(AF_df$ROA[which(AF_df[ , 2] == publish_unique[i])])
  avg_roa <- append(avg_roa, mean_roa)
}
#adding Kayak to our data
publisher_names <- append(publisher_names,'Kayak')
avg_roa <- append(avg_roa, kayak_df$ROA)
#creating a data frame with our data
data <- data.frame(publisher_names, avg_roa )
#plotting the bar chart
data$publisher_names <- factor(data$publisher_names, levels = data[["publisher_names"]])
pub_roa <- plot_ly(data, x= ~publisher_names, y= ~avg_roa, type = "bar",
                   name = "Return on Advertising", color = I("red"), alpha = 0.7,
                   width = 750, height = 500) %>%
  add_text(text = round(avg_roa, digits = 2), textposition = 'top', color = I("black")) %>%
  layout(title = "Return On Advertising",
         xaxis = list(title = ""),
         yaxis = list(title = ""),
         showlegend = FALSE)
#printing the bar chart
```

```
pub_roa
```

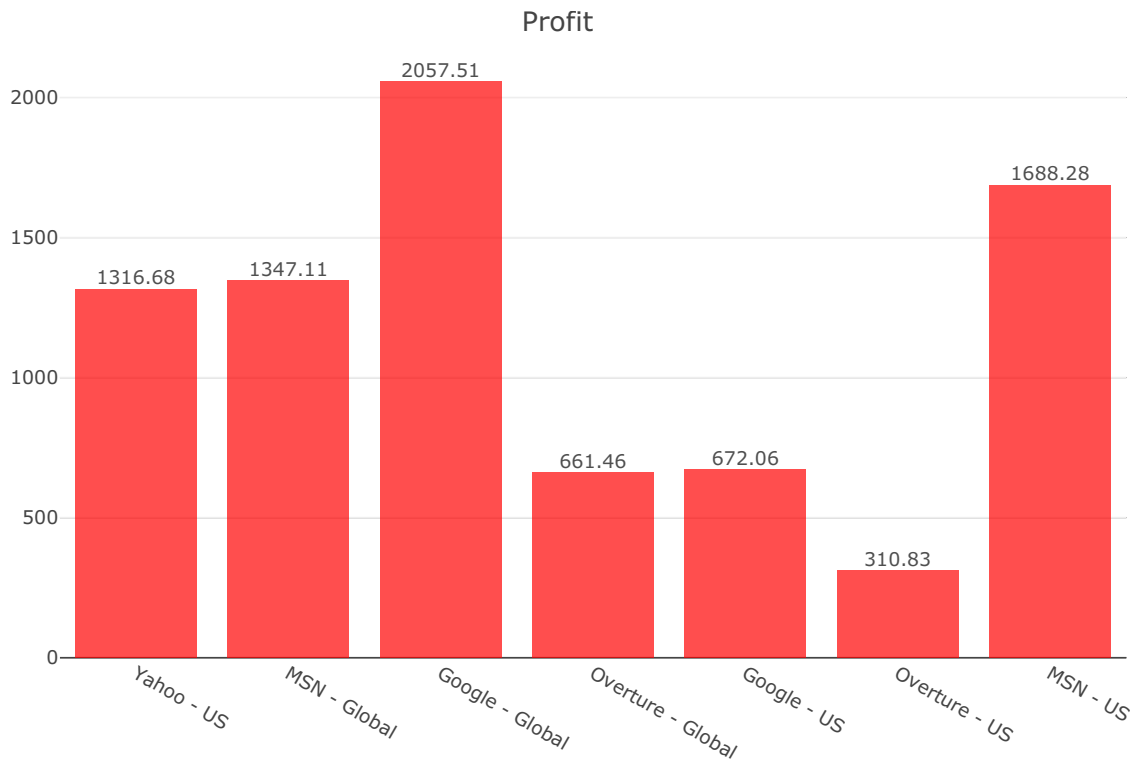
4.1.1.) Bar Chart of ROA per Publisher




```
#creating the dependent variable (Profit) for the graph using a for loop
avg_profit <- c()
for (i in 1:length(publish_unique)) {
  mean_prof <- mean(AF_df$Profit[which(AF_df[ , 2] == publish_unique[i])])
  avg_profit <- append(avg_profit, mean_prof)
}
#creating a data frame with our data
data <- data.frame(publish_unique, avg_profit)
#plotting the bar chart
data$publish_unique <- factor(data$publish_unique, levels = data[["publish_unique"]])
pub_prof <- plot_ly(data, x= ~publish_unique, y= ~avg_profit, type = "bar", name = "Profit",
                    color = I("red"), alpha = 0.7,
                    width = 750, height = 500) %>%
  add_text(text = round(avg_profit, digits = 2), textposition = 'top', color = I("black")) %>%
  layout(title = "Profit",
         xaxis = list(title = ""),
         yaxis = list(title = ""),
         showlegend = FALSE)
#printing the bar chart
```

```
pub_prof
```

4.1.2.) Bar Chart of Profit per Publisher

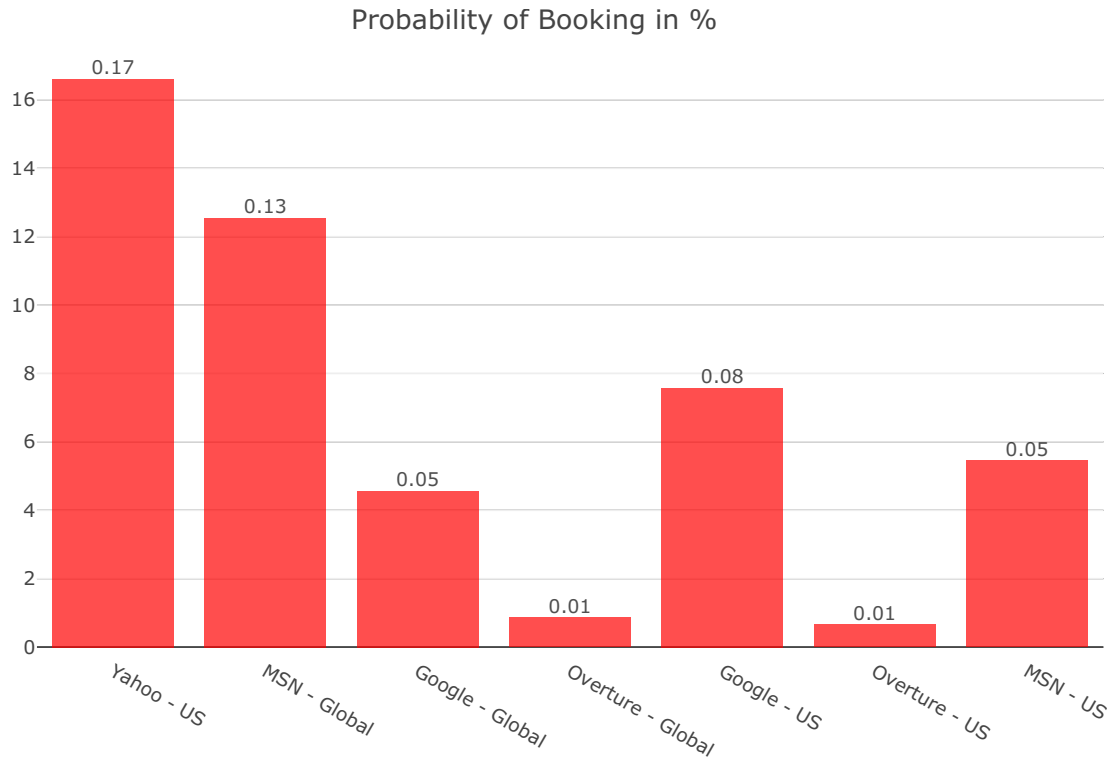



```
#creating the dependent variable (Prob.of Booking) for the graph using a for loop
prob_book <- c()
for (i in 1:length(publish_unique)) {
  mean_prob <- mean(AF_df$Prob_of_Book[which(AF_df[ , 2] == publish_unique[i])])
  prob_book <- append(prob_book, mean_prob)
}
#creating a data frame with our data
data <- data.frame(publish_unique, prob_book*100)
#plotting the bar chart
data$publish_unique <- factor(data$publish_unique, levels = data[["publish_unique"]])
pub_book <- plot_ly(data, x= ~publish_unique, y= ~prob_book*100, type = "bar",
                    name = "Probability of Booking", color = I("red"), alpha = 0.7,
                    width = 750, height = 500) %>%
  add_text(text = round(prob_book, digits = 2), textposition = 'top', color = I("black")) %>%
```

```
  layout(title = "Probability of Booking in %",
         xaxis = list(title = ""),
         yaxis = list(title = ""),
         showlegend = FALSE)
#printing the bar chart
```

```
pub_book
```

4.1.3.) Bar Chart of Prob.of Booking per Publisher



4.2.) Some Visualizations including Kayak

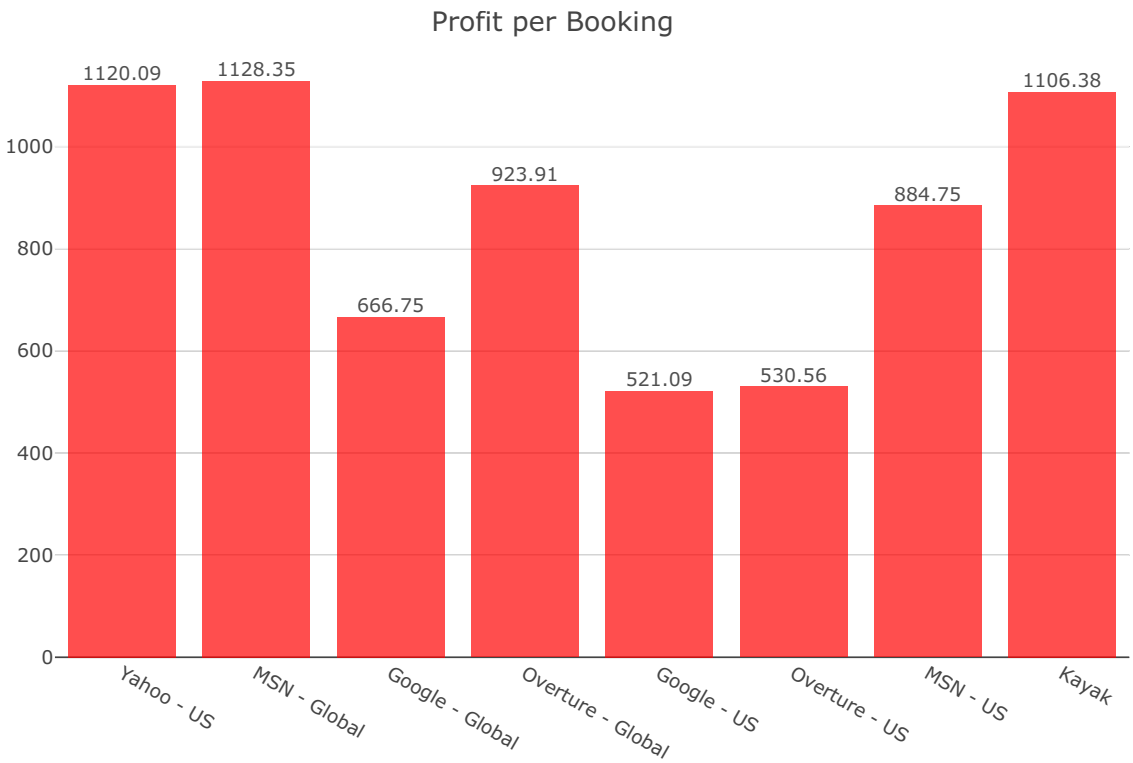
Some visualizations to better understand the performance of Kayak in comparison to the other publishers.

```
#creating the independent variable (Publisher Name) for the graph
publisher_names_prof<- publish_unique
#creating the dependent variable (Prof.per Booking) for the graph using a for loop
```

```
prof_book <- c()
for (i in 1:length(publish_unique)) {
  mean_profb <- mean(AF_df$Prof_Book[which(AF_df[ , 2] == publish_unique[i] & AF_df[ ,'Total Volume of
  prof_book <- append(prof_book, mean_profb)
}
#adding Kayak to our data
publisher_names_prof<- append(publisher_names_prof,'Kayak')
prof_book <- append(prof_book,kayak_df$Prof_Book)
#creating a data frame with our data
data <- data.frame(publisher_names_prof, prof_book)
#plotting the bar chart
data$publisher_names_prof <- factor(data$publisher_names_prof, levels = data[["publisher_names_prof"]])
pub_profb <- plot_ly(data, x= ~publisher_names_prof, y= ~prof_book, type = "bar",
                     name = "Profit per Booking", color = I("red"), alpha = 0.7,
                     width = 750, height = 500) %>%
  add_text(text = round(prof_book, digits = 2), textposition = 'top', color = I("black")) %>%
  layout(title = "Profit per Booking",
         xaxis = list(title = ""),
         yaxis = list(title = ""),
         showlegend = FALSE)
#printing the bar chart
```

```
pub_profb
```

4.2.1.) Bar Chart of Profit per Booking per Publisher (incl. Kayak)

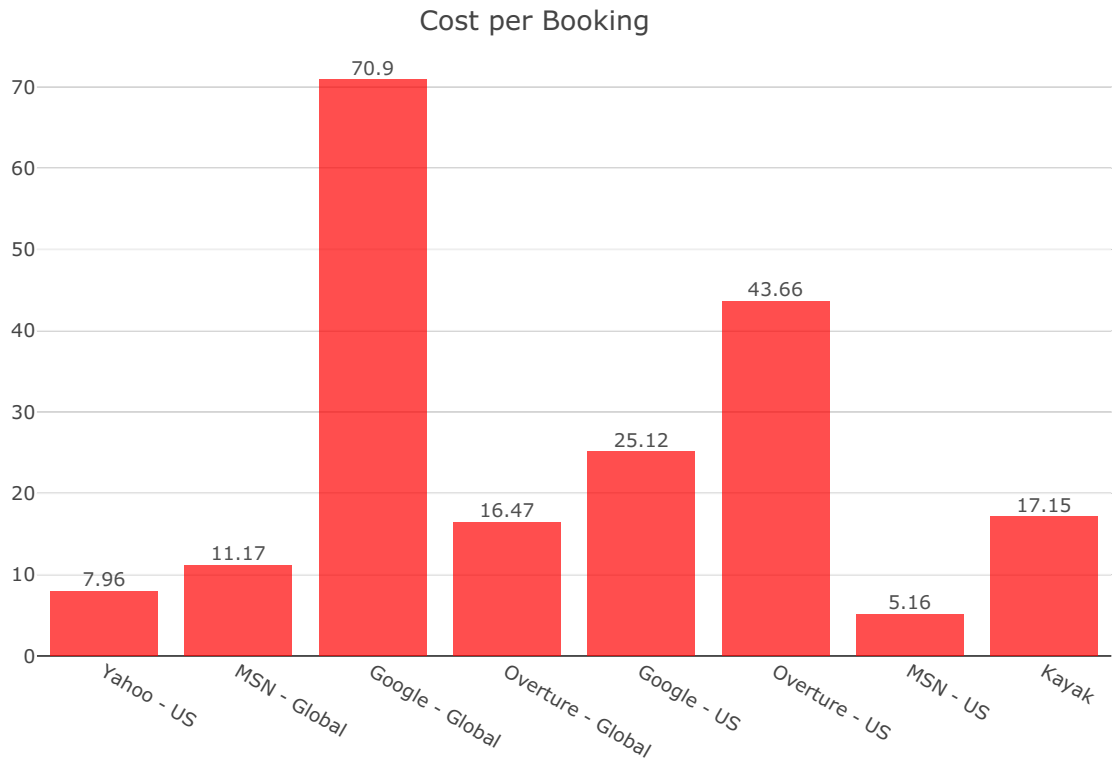



```
#creating the independent variable (Publisher Name) for the graph
publisher_names_cpb <- publish_unique
#creating the dependent variable (Prob.of Booking) for the graph using a for loop
cost_book <- c()
for (i in 1:length(publish_unique)) {
  mean_costb <- mean(AF_df$Cost_Book[which(AF_df[ , 2] == publish_unique[i])])
  cost_book<- append(cost_book, mean_costb)
}
#adding Kayak to our data
publisher_names_cpb <- append(publisher_names_cpb,'Kayak')
cost_book <- append(cost_book, kayak_df$Cost_Book)
#creating a data frame with our data
data <- data.frame(publisher_names_cpb, cost_book)
#plotting the bar chart
data$publisher_names_cpb <- factor(data$publisher_names_cpb, levels = data[["publisher_names_cpb"]])
pub_costb <- plot_ly(data, x= ~publisher_names_cpb, y= ~cost_book, type = "bar",
                     name = "Cost per Booking", color = I("red"), alpha = 0.7,
                     width = 750, height = 500) %>%
  add_text(text = round(cost_book, digits = 2), textposition = 'top', color = I("black")) %>%
  layout(title = "Cost per Booking",
         xaxis = list(title = ""),
```

```
        yaxis = list(title = ""),
        showlegend = FALSE)
#printing the bar chart
```

```
pub_costb
```

4.2.2.) Bar Chart of Cost per Booking per Publisher (incl. Kayak)

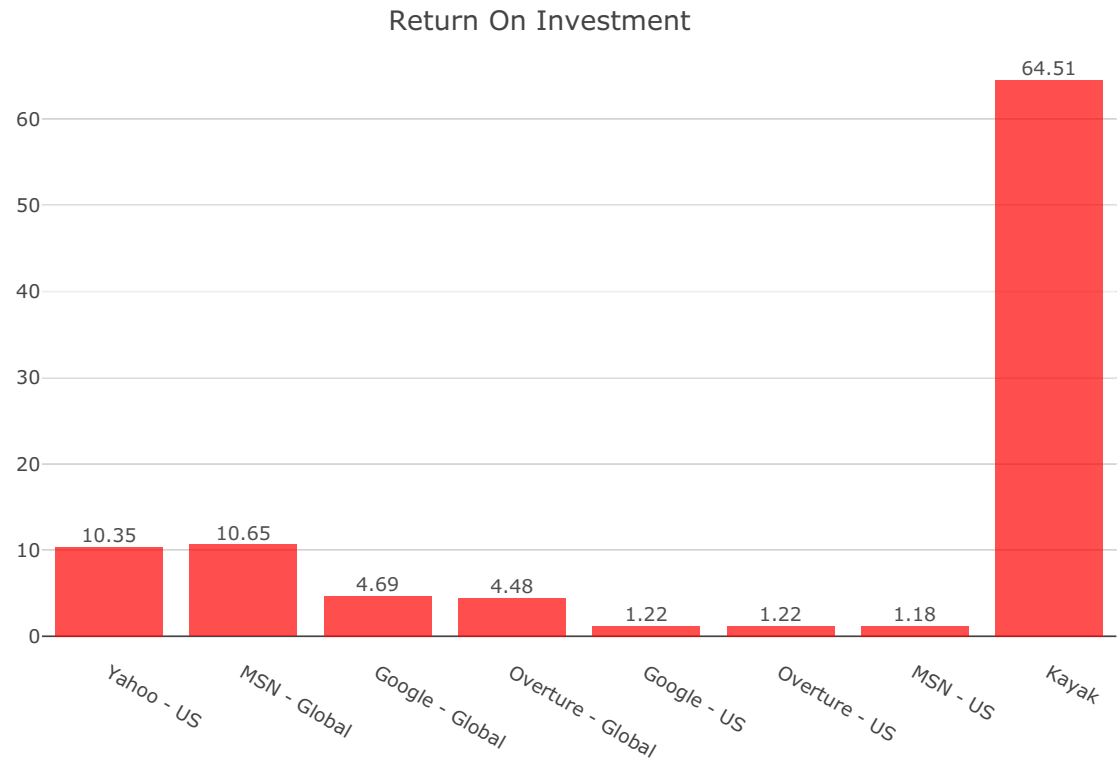



```
#creating the independent variable (Publisher Name) for the graph
publisher_names_roip <- publish_unique
#creating the dependent variable (Prob.of Booking) for the graph using a for loop
avg_roi <- c()
for (i in 1:length(publish_unique)) {
  mean_roi <- mean(AF_df$ROI[which(AF_df[ , 2] == publish_unique[i])])
  avg_roi <- append(avg_roi, mean_roi)
}
#adding Kayak to our data
```

```
publisher_names_roip <- append(publisher_names_roip,'Kayak')
avg_roi <- append(avg_roi, kayak_df$ROI)
#creating a data frame with our data
data <- data.frame(publisher_names_roip, avg_roi)
#plotting the bar chart
data$publisher_names_roip <- factor(data$publisher_names_roip, levels = data[["publisher_names_roip"]])
pub_roi <- plot_ly(data, x= ~publisher_names_roip, y= ~avg_roi, type = "bar",
                   name = "Return on investment", color = I("red"), alpha = 0.7,
                   width = 750, height = 500) %>%
  add_text(text = round(avg_roi, digits = 2), textposition = 'top', color = I("black")) %>%
  layout(title = "Return On Investment",
         xaxis = list(title = ""),
         yaxis = list(title = ""),
         showlegend = FALSE)
#printing the bar chart
```

```
pub_roi
```

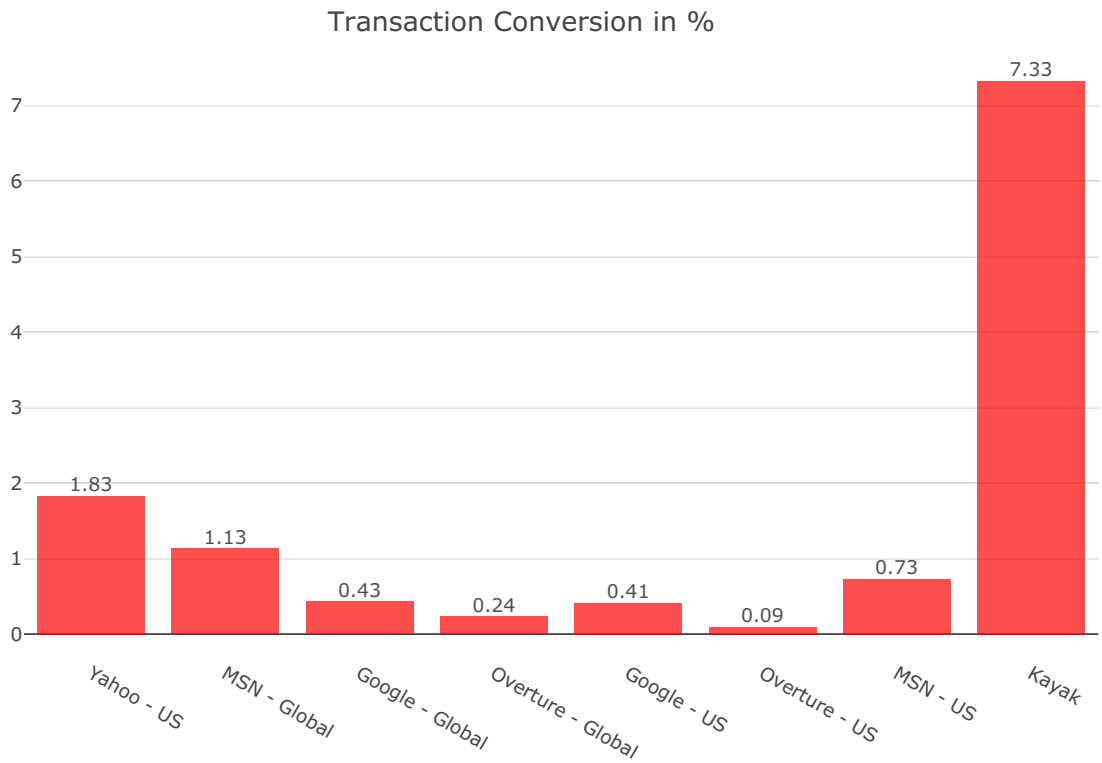
4.2.3.) Bar Chart of Return on Investment per Publisher (incl. Kayak)



```
#creating the independent variable (Publisher Name) for the graph
publisher_names_tcp <- publish_unique
#creating the dependent variable (Prob.of Booking) for the graph using a for loop
trans_conv <- c()
for (i in 1:length(publish_unique)) {
  mean_tc <- mean(AF_df$`Trans. Conv. %`[which(AF_df[ , 2] == publish_unique[i])])
  trans_conv <- append(trans_conv,mean_tc)
}
#adding Kayak to our data
publisher_names_tcp <- append(publisher_names_tcp,'Kayak')
trans_conv <- append(trans_conv, kayak_df$`Trans. Conv. %`*100)
#creating a data frame with our data
data <- data.frame(publisher_names_tcp, trans_conv )
#plotting the bar chart
data$publisher_names_tcp <- factor(data$publisher_names_tcp, levels = data[["publisher_names_tcp"]])
pub_tcr <- plot_ly(data, x= ~publisher_names_tcp, y= ~trans_conv , type = "bar",
                   name = "Transaction Conversion in %", color = I("red"), alpha = 0.7,
                   width = 750, height = 500) %>%
  add_text(text = round(trans_conv, digits = 2), textposition = 'top', color = I("black")) %>%
  layout(title = "Transaction Conversion in %",
         xaxis = list(title = ""),
         yaxis = list(title = ""),
         showlegend = FALSE)
#printing the bar chart
```

```
pub_tcr
```

4.2.4.) Bar Chart of Transaction Conv.% per Publisher (incl. Kayak)



4.3.) Additional insightful Charts (excl. Kayak)

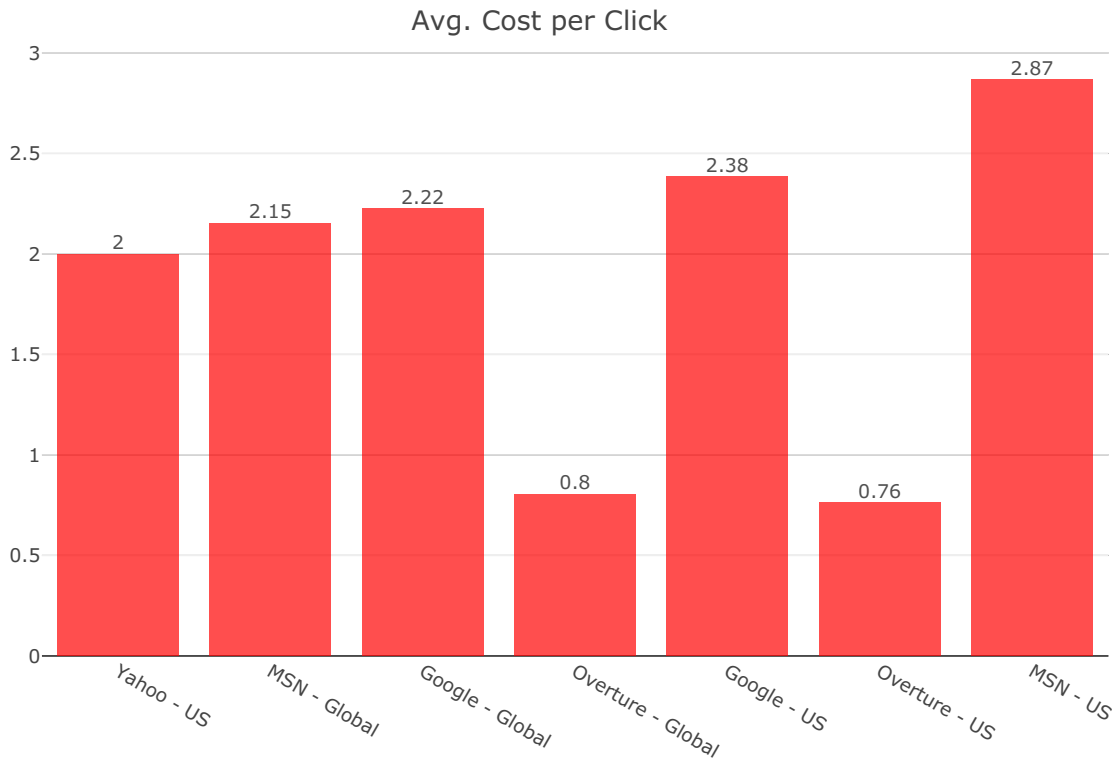
These are furter insights we investigated on. The data we collected on Kayak provided had information for these ones.

```
#creating the dependent variable (Prob.of Booking) for the graph using a for loop
avg_cpc <- c()
for (i in 1:length(publish_unique)) {
  mean_avgcpc <- mean(AF_df$`Avg. Cost per Click`[which(AF_df[ , 2] == publish_unique[i])])
  avg_cpc<- append(avg_cpc, mean_avgcpc)
}
#creating a data frame with our data
data <- data.frame(publish_unique, avg_cpc)
#plotting the bar chart
data$publish_unique <- factor(data$publish_unique, levels = data[["publish_unique"]])
pub_cpc <- plot_ly(data, x= ~publish_unique, y= ~avg_cpc, type = "bar", name = "Avg. Cost per Click",
                   color = I("red"), alpha = 0.7,
                   width = 750, height = 500) %>%
  add_text(text = round(avg_cpc, digits = 2), textposition = 'top', color = I("black")) %>%
  layout(title = "Avg. Cost per Click",
         xaxis = list(title = ""),
```

```
        yaxis = list(title = ""),
        showlegend = FALSE)
#printing the bar chart
```

```
pub_cpc
```

4.3.1) Bar Chart of Avg. Cost per Click per Publisher

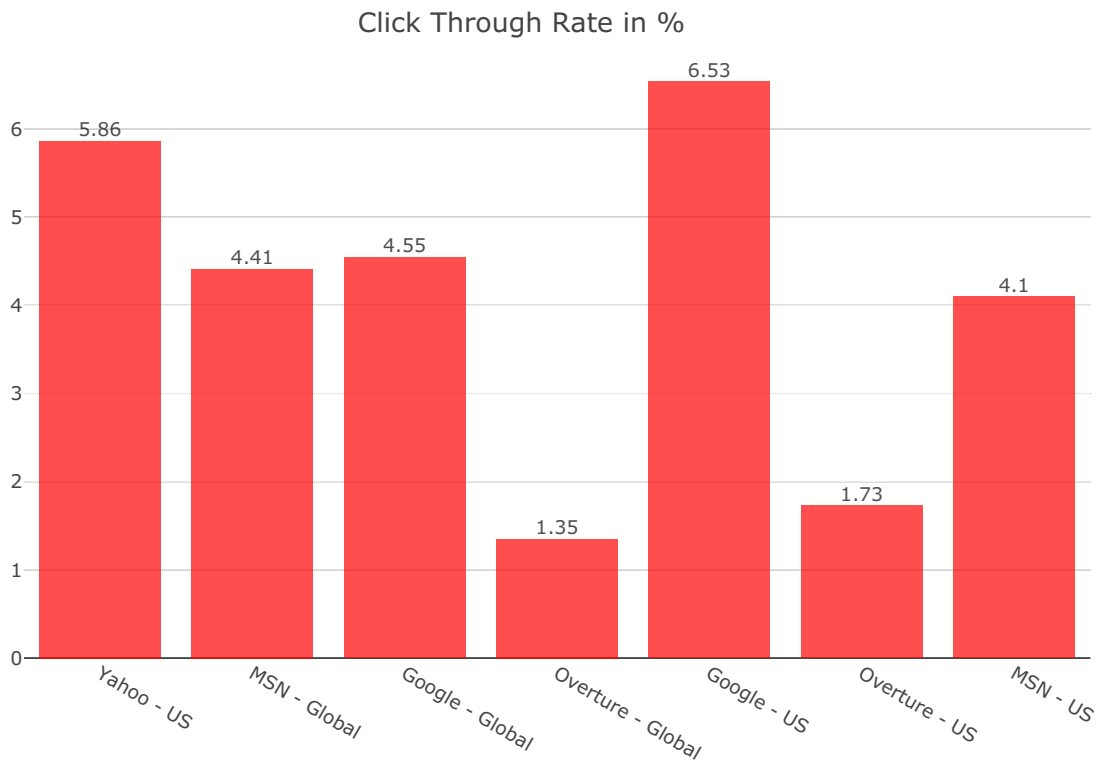



```
#creating the independent variable (Publisher Name) for the graph
publisher_names_ctr<- publish_unique
#creating the dependent variable (Prof.per Booking) for the graph using a for loop
click_tr <- c()
for (i in 1:length(publish_unique)) {
  mean_clickt <- mean(AF_df$`Engine Click Thru %`[which(AF_df[ , 2] == publish_unique[i])],round(2))
  click_tr<- append(click_tr, mean_clickt)
}
#creating a data frame with our data
```

```
data <- data.frame(publisher_names_ctr, click_tr)
#plotting the bar chart
data$publisher_names_ctr <- factor(data$publisher_names_ctr, levels = data[["publisher_names_ctr"]])
pub_ctr <- plot_ly(data, x= ~publisher_names_ctr, y= ~click_tr, type = "bar",
                   name = "Click Through Rate in %", color = I("red"), alpha = 0.7,
                   width = 750, height = 500) %>%
  add_text(text = round(click_tr, digits = 2), textposition = 'top', color = I("black")) %>%
  layout(title = "Click Through Rate in %",
         xaxis = list(title = ""),
         yaxis = list(title = ""),
         showlegend = FALSE)
#printing the bar chart
```

```
pub_ctr
```

4.3.2.) Bar Chart Click Trough.% per Publisher



4.4.) Visualizations regarding the Power of Branded Keywords

This graphs aim to better understand how well placed branded keywords can significantly contribute to better financial outcomes. As there were no keywords mentioned in the Kayak sheet - Kayak is not included

in these graphs.

```
#creating a vector with branded key words
brand_keys <- c("airfrance", "air france")
#creating a a dataf frame with branded keywords
df_branded <- AF_df[grep(paste(brand_keys, collapse = "|"), AF_df$Keyword, ignore.case = TRUE),]
#creating a a data frame with unbranded keywords
df_not_branded <- AF_df[-grep(paste(brand_keys, collapse = "|"), AF_df$Keyword, ignore.case = TRUE),]
#creating a vector with the mean booking probability of branded and unbranded key words
prob_book_vect <- c(mean(df_branded$Prob_of_Book), mean(df_not_branded$Prob_of_Book))
#creating a vector with the mean profit of branded and unbranded key words
avg_prof_vect <- c(mean(df_branded$Profit), mean(df_not_branded$Profit))
#creating a data frame with probability of booking and profit
df_branded_unbranded <- as.data.frame(cbind(prob_book_vect*100, avg_prof_vect))
#naming the columns of the data frame
colnames(df_branded_unbranded) <- c("Probability of Bookings", "Average Profits")
#naming the rows of the data frame
rownames(df_branded_unbranded) <- c("Branded Keywords", "Unbranded Keywords")

#plotting the bar chart
prob_bk_kw <- plot_ly(data = df_branded_unbranded, x = rownames(df_branded_unbranded),
                      y= ~`Probability of Bookings`, type = "bar", name = "Probability of Bookings",
                      color = I("red"), alpha = 0.7,
                      width = 750, height = 500) %>%
  add_text(text = round(prob_book_vect*100, digits = 2), textposition = 'top', color = I("black")) %>%
  layout(title = "Probability of Bookings in %",
         xaxis = list(title = ""),
         yaxis = list(title = ""),
         showlegend = FALSE)
#printing the bar chart
```

```
prob_bk_kw
```

4.4.1.) Bar Chart Probability of Booking by Branded and Unbranded Keywords


```
#plotting the bar chart
prof_kw <- plot_ly(df_branded_unbranded, x = rownames(df_branded_unbranded), y= ~`Average Profits`,
                   type = "bar", name = "Average Profits", color = I("red"), alpha = 0.7,
                   width = 750, height = 500) %>%
  add_text(text = round(avg_prof_vect, digits = 2), textposition = 'top', color = I("black")) %>%
  layout(title = "Average Profits",
         xaxis = list(title = ""),
         yaxis = list(title = ""),
         showlegend = FALSE)
#printing the bar chart
```

```
prof_kw
```

4.4.2.) Bar Chart Average Profits by Branded and Unbranded Keywords

4.5.) Visualizations regarding ROA - SEM - Cost per Click - Probabilities of Booking

These visualizations aim to better understand the correlations between the variables explained above and in particularly looking at them again in terms of the different channels. Without Kayak as there were lacking certain data.

```
#creating my pivot table
AF_df_ptrc <- AF_df %>% group_by(`Publisher Name`) %>% summarize(
  Avg_ROA = mean(ROA),
  Avg_cpc = mean(`Avg. Cost per Click`)
)
# Plotting the pivot table
dust(AF_df_ptrc) %>%
  sprinkle(round = 3) %>%
  kable() %>%
  kable_styling()
```

Publisher Name	Avg_ROA	Avg_cpc
Google - Global	5.69	2.225
Google - US	2.221	2.384
MSN - Global	11.649	2.153
MSN - US	2.181	2.867
Overture - Global	5.485	0.805
Overture - US	2.216	0.764
Yahoo - US	11.347	1.999

4.5.1.) Pivot table for ROA and Avg.Cost per Click

```
#creating my pivot table
AF_df_ptsem <- AF_df %>% group_by(`Publisher Name`) %>% summarize(
  total_records = n(),
  total_amount = sum(`Total Cost`),
  Avg_cpc = mean(`Avg. Cost per Click`),
  Avg_prob = mean(`Prob_of_Book`),
  Avg_ROA = mean(ROA)
)
# Plotting the pivot table
dust(AF_df_ptsem) %>%
  sprinkle(round = 3) %>%
  kable() %>%
  kable_styling()
```

Publisher Name	total_records	total_amount	Avg_cpc	Avg_prob	Avg_ROA
Google - Global	393	120946.712	2.225	0.046	5.69
Google - US	2071	353640.599	2.384	0.076	2.221
MSN - Global	99	12160.362	2.153	0.125	11.649
MSN - US	98	16098.487	2.867	0.055	2.181
Overture - Global	553	64295.862	0.805	0.009	5.485
Overture - US	661	141976.074	0.764	0.006	2.216
Yahoo - US	635	46197.825	1.999	0.166	11.347

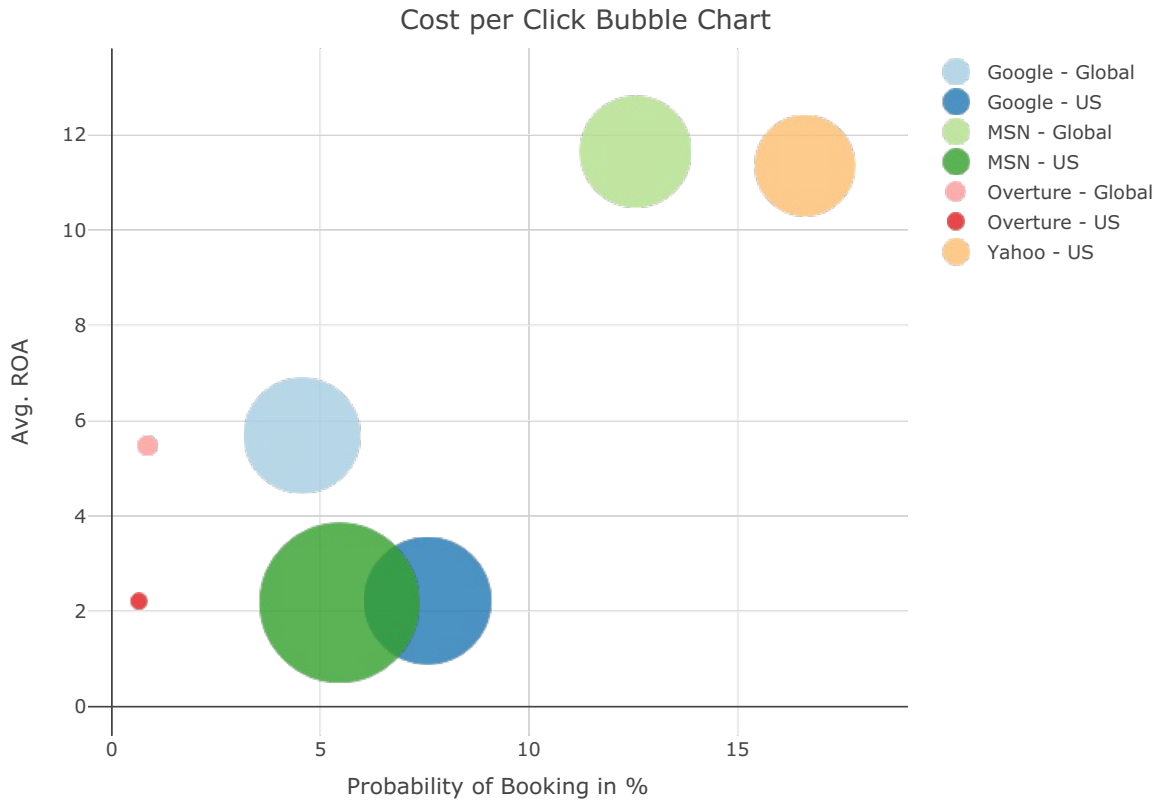
4.5.2.) Pivot table for ROA compare different Search Engine Marketing (SEM)

```
#creating my bubble chart
bub_pbr<- plot_ly(AF_df_ptsem, x = ~Avg_prob*100, y = ~Avg_ROA,
             textposition = "auto",
             type = 'scatter',
             mode = 'markers',
             size = ~Avg_cpc,
             color = ~`Publisher Name`,
             colors = 'Paired',
             marker = list(opacity = 0.8, sizemode = 'diameter'),
             width = 750, height = 500) %>%
```

```
  layout(title = 'Cost per Click Bubble Chart',
         xaxis = list(title = "Probability of Booking in %", showgrid = TRUE),
         yaxis = list(title = "Avg. ROA", showgrid = TRUE),
         showlegend = TRUE)
#printing my bubble chart
```

```
bub_pbr
```

4.5.3.) Bubble Chart Probability of Booking and Avg.ROA



4.6.) Visualizations with the subset of existing Bid Strategy

These visualizations aim to get a better understanding about the different Bid Strategies. Without Kayak as there were lacking data on on Bid Strategy.

```
# Creating my pivot table
AF_df_ptbs <- (subset(AF_df, is.na(AF_df$`Bid Strategy`) == FALSE) %>%
```

```
  group_by(`Publisher Name`, `Bid Strategy`) %>%
  summarize(
    total_records = n(),
    avg_cpc = mean(`Avg. Cost per Click`),
    avg_prof = mean(`Profit`) )
    )
# Plotting the pivot table
dust(AF_df_ptbs) %>%
  sprinkle(round = 3) %>%
  kable() %>%
  kable_styling()
```

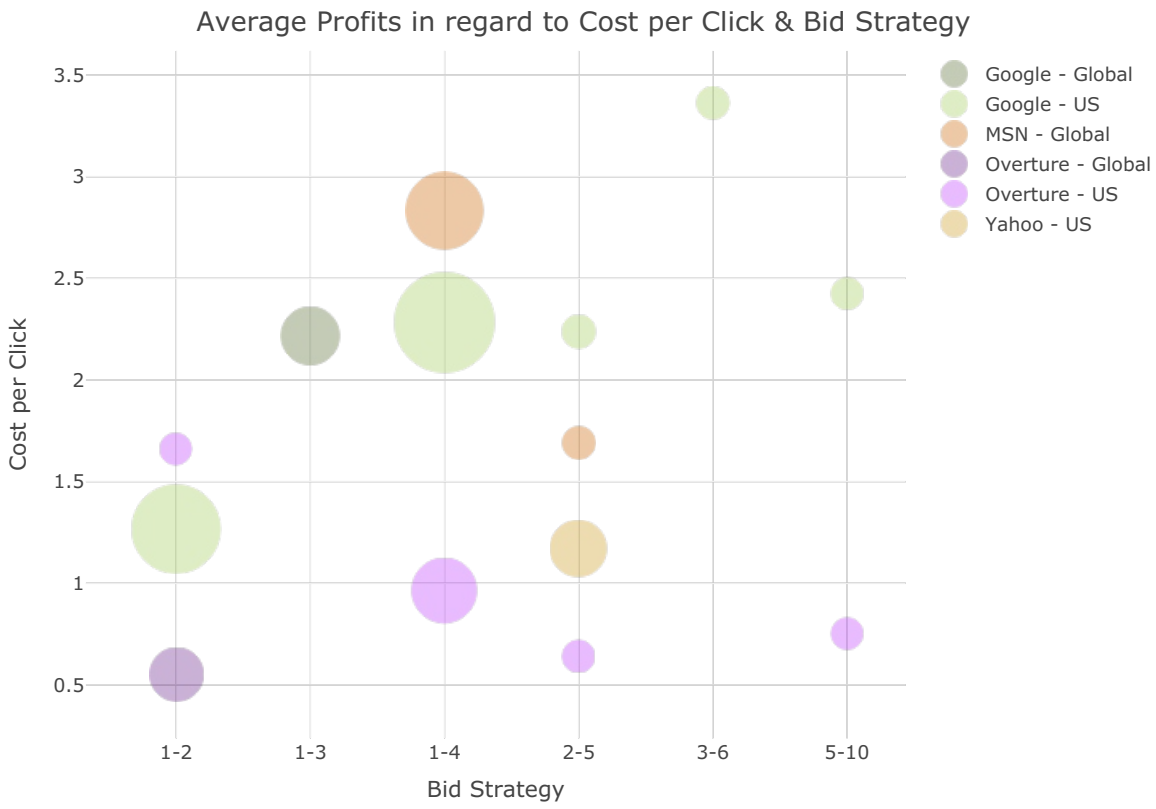
Publisher Name	Bid Strategy	total_records	avg_cpc	avg_prof
Google - Global	1-3	264	2.216	1541.251
Google - US	1-2	11	1.269	4555.233
Google - US	1-4	40	2.282	6044.135
Google - US	2-5	143	2.238	51.542
Google - US	3-6	45	3.36	12.118
Google - US	5-10	1764	2.424	-17.901
MSN - Global	1-4	40	2.833	3296.108
MSN - Global	2-5	59	1.692	25.756
Overture - Global	1-2	271	0.552	1158.492
Overture - US	1-2	3	1.658	-32.917
Overture - US	1-4	71	0.965	2106.265
Overture - US	2-5	130	0.642	-3.247
Overture - US	5-10	444	0.754	-34.233
Yahoo - US	2-5	1	1.171	1371.475

4.6.1.) Creating a pivot table of the subset where a Bid Strategy was defined

```
# Creating my bubble chart
bub_pcpcbid <- plot_ly(data = AF_df_ptbs, x = ~`Bid Strategy`, y = ~avg_cpc,
                       type = "scatter", mode = "markers",
                       color = AF_df_ptbs$`Publisher Name`,
                       colors = c("darkolivegreen", "darkolivegreen3", "darkorange3",
                                  "darkorchid4", "darkorchid1", "goldenrod3"),
                       size = ~avg_prof,
                       opacity = 0.5,
                       marker = list(sizeref = 0.05, sizemode = "area"),
                       showlegend = TRUE,
                       width = 750, height = 500) %>%
  layout(title = 'Average Profits in regard to Cost per Click & Bid Strategy',
         xaxis = list(title = "Bid Strategy", showgrid = TRUE),
         yaxis = list(title = "Cost per Click", showgrid = TRUE))
# Printing my bubble chart
```

```
bub_pcpcbid
```

4.6.2.) Bubble Chart Avg. Profits in regard to Cost per Click & Bid Strategy




```
# Creating my pivot table
AF_df_ptbsms <- (subset(AF_df, is.na(AF_df$`Bid Strategy`) == FALSE) %>%
                   group_by(`Publisher Name`, `Bid Strategy`, `Match Type`) %>%
                   summarise(
                     total_records = n(),
                     avg_cpc = mean(`ROA`))
                 )
# Plotting the pivot table
dust(AF_df_ptbsms) %>%
  sprinkle(round = 3) %>%
  kable() %>%
  kable_styling()
```

Publisher Name	Bid Strategy	Match Type	total_records	avg_cpc
Google - Global	1-3	Broad	262	2.776
Google - Global	1-3	Exact	2	384.179
Google - US	1-2	Broad	11	31.563
Google - US	1-4	Broad	40	10.444
Google - US	2-5	Broad	143	2.598
Google - US	3-6	Broad	45	0.704
Google - US	5-10	Broad	1746	1.265
Google - US	5-10	Exact	18	0
MSN - Global	1-4	Broad	40	3.485
MSN - Global	2-5	Broad	59	17.184
Overture - Global	1-2	Advanced	148	9.094
Overture - Global	1-2	Standard	123	10.609
Overture - US	1-2	Advanced	3	0
Overture - US	1-4	Advanced	21	2.256
Overture - US	1-4	Standard	50	20.228
Overture - US	2-5	Advanced	3	0
Overture - US	2-5	Standard	127	1.716
Overture - US	5-10	Advanced	197	0
Overture - US	5-10	Standard	247	0.013
Yahoo - US	2-5	Advanced	1	3.261

4.6.3.) Creating a pivot table of the subset where a Bid Strategy was defined with focus on Match Type

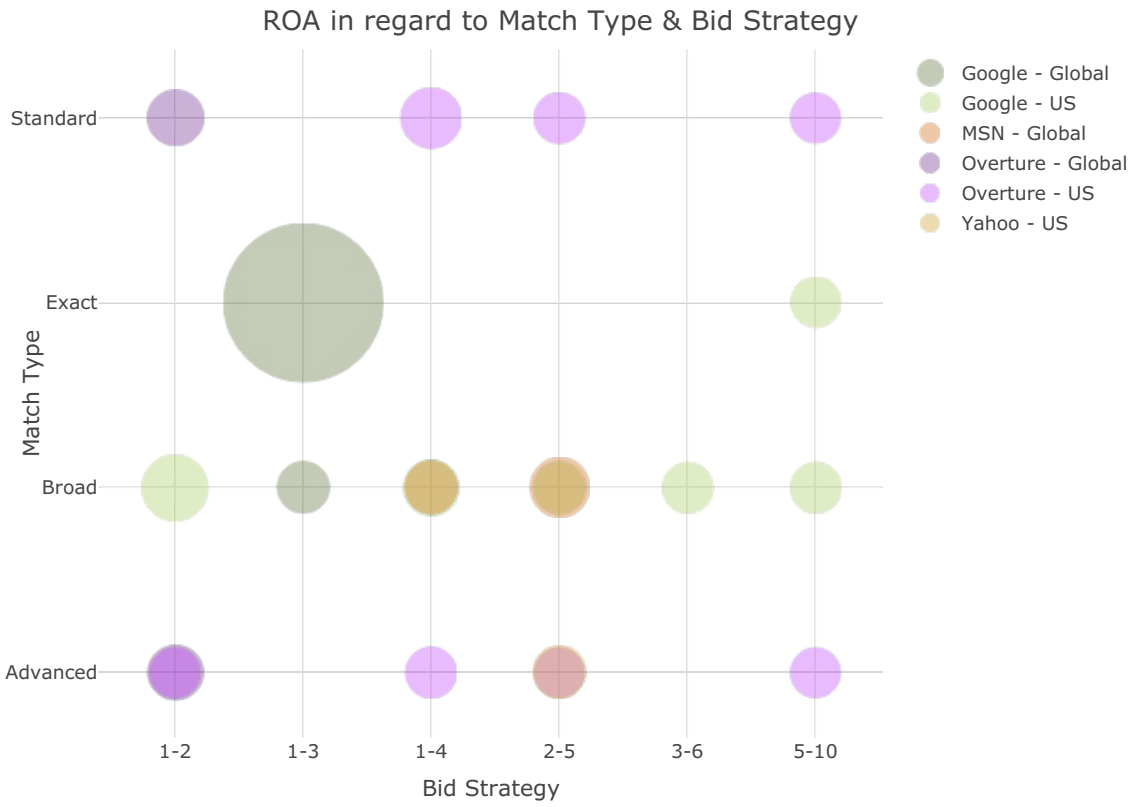
```
# Creating my bubble chart
bub_roamtbs <- plot_ly(data = AF_df_ptbsms, x = ~`Bid Strategy`, y = ~`Match Type`,
                       type = "scatter", mode = "markers",
                       color = AF_df_ptbsms$`Publisher Name`,
                       colors = c("darkolivegreen", "darkolivegreen3", "darkorange3",
                                  "darkorchid4", "darkorchid1", "goldenrod3"),
                       size = ~avg_cpc,
                       opacity = 0.5,
                       marker = list(sizeref = 0.02, sizemode = "area"),
                       showlegend = TRUE,
                       width = 750, height = 500) %>%
  layout(title = 'ROA in regard to Match Type & Bid Strategy',
         xaxis = list(title = "Bid Strategy", showgrid = TRUE),
         yaxis = list(title = "Match Type", showgrid = TRUE))

# Printing my bubble chart

# Important to understand that the apparent winner "Google - Global / Match Type: Exact / Bid Strategy:
```

```
bub_roamtbs
```

4.6.4.) Bubble Chart ROA in regard to Match Type & Bid Strategy



4.7.) Visualizations with the subset of existing Bookings

This charts aim to get a better understanding about the situation when it comes to bookings. Without Kayak as there were lacking certain data.

```
# Creating my pivot table
AF_df_ptbook <- (subset(AF_df, (AF_df$`Total Volume of Bookings` > 0))) %>%
                  group_by(`Publisher Name`) %>%
                  summarize(
                    count_of_piv = n(),
                    avg_cpc = mean(`Avg. Cost per Click`),
                    avg_cost_book = median(Cost_Book),
                    avg_prob_book = median(Prob_of_Book),
                    avg_prof_book = median(Prof_Book)
                    )
# Plotting the pivot table
dust(AF_df_ptbook) %>%
  sprinkle(round = 3) %>%
  kable() %>%
  kable_styling()
```

Publisher Name	count_of_piv	avg_cpc	avg_cost_book	avg_prob_book	avg_prof_book
Google - Global	89	2.156	191.762	0.06	609.95
Google - US	134	2.138	93.303	0.127	522.556
MSN - Global	10	1.288	61.037	0.146	834.542
MSN - US	9	1.525	39.638	0.305	823.26
Overture - Global	45	0.713	26.237	0.039	918.788
Overture - US	48	0.977	283.534	0.009	647.166
Yahoo - US	33	1.517	50.362	0.131	874

4.7.1.) Pivot table of the subset where Bookings occured

```
# Creating my bubble chart
bub_ppbcb <- plot_ly(data = AF_df_ptbook, x = ~avg_prob_book, y = ~avg_cost_book,
                type = "scatter", mode = "markers",
                color = AF_df_ptbook$`Publisher Name`,
                colors = 'Paired',
                size = ~avg_prof_book,
                marker = list(sizeref = 0.05, sizemode = "area"),
                showlegend = TRUE,
                width = 750, height = 500) %>%
  layout(title = 'Avg. Profit p.Booking to Prob. of Booking and Cost per Booking',
         xaxis = list(title = "Probability of Booking", showgrid = TRUE),
         yaxis = list(title = "Costs per Booking", showgrid = TRUE))
# Printing my bubble chart
```

```
bub_ppbcb
```

4.7.2.) Bubble Chart Avg. Profits in regards to Prob. of Booking and Cost per Booking


```
# Creating my bubble chart
bub_pbpbcc <- plot_ly(data = AF_df_ptbook, x = ~avg_prob_book, y = ~avg_cpc,
                type = "scatter", mode = "markers",
                color = AF_df_ptbook$`Publisher Name`,
                colors = 'Paired',
                size = ~avg_prof_book,
                marker = list(sizeref = 0.05, sizemode = "area"),
                showlegend = TRUE,
                width = 750, height = 500) %>%
  layout(title = 'Avg Profit p.Booking to Prob. of Booking and Cost per Click',
         xaxis = list(title = "Probability of Booking", showgrid = TRUE),
         yaxis = list(title = "Costs per Click", showgrid = TRUE))
# Printing my bubble chart
```

```
bub_pbpbcc
```

4.7.3.) Bubble Chart Avg. Profits per Booking to Probability of Booking and Cost per Click


Avg Profit p.Booking to Prob. of Booking and Cost per Click
Costs per Click
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0
0.05
0.1
0.15
0.2
0.25
0.3
Probability of Booking
Google - Global
Google - US
MSN - Global
MSN - US
Overture - Global
Overture - US
Yahoo - US