Assignment 1 – Model answers

Problem 1

To run the attribution model, you need to first install and load the "ChannelAttribution" package and then run the "heuristics_models" function on the data. This will provide you with the output containing the number of conversions and total conversion value per channel.

You can also obtain a more reader-friendly presentation of the results by printing the number of conversions and conversion values separately. In the output, columns 2, 4, and 6 contain the numbers of conversions, while 3, 5 and 7 contain the monetary value of the conversions. We can save the number conversions and conversion values per channel into separate objects

```
model_conversions <- model[,c(1,2,4,6)]
model_conversions
```

```
      channel_name first_touch_conversions last_touch_conversions linear_touch_conversions
1        Affiliate                     308                     339                321.15278
2          Display                      96                       0                 68.19959
3       Newsletter                      37                     125                 65.74713
4 Price_comparison                      89                      98                 91.81667
5              SEO                     211                     260                236.76634
6            Other                      30                      26                 28.58333
7              SEA                     928                     851                886.23416
8      Retargeting                       9                       9                  9.50000
```

```
model_conversionvalue <- model[,c(1,3,5,7)]
model_conversionvalue
```

```
      channel_name first_touch_value last_touch_value linear_touch_value
1        Affiliate         21694.670       23807.9650          22597.830
2          Display          6747.347           0.0000           4781.804
3       Newsletter          2629.857        8842.2408           4672.154
4 Price_comparison          6179.352        6689.8022           6323.110
5              SEO         14377.487       17973.5081          16287.696
6            Other          2101.650        1886.3698           2025.755
7              SEA         65948.307       60459.3069          62943.294
8      Retargeting           630.990         650.4683            678.018
```

In light of these results, the following interpretations can be made about different channels

Channel	Role in the path to purchase
Affiliate	Similar revenue attribution across models. Therefore, affiliate is 'touched' across the entire journey, and thus seems to be important in creating awareness, increasing consideration, as well as in motivating ultimate purchase decisions.
Display	Display seems to be mainly feeding the 'top of the funnel', since there is no revenue attributed to the channel on a last-touch basis.
Retargeting	Unlike (traditional) display, retargeting seems to work as a means to 'convert' prospective customers into paying customers. This is understandable since the ads are shown to people that have already indicated an interest in the agency's offering by visiting their website. However, the number of conversions is quite low.
Newsletter	Newsletter seems to be a strong driver of final purchase decisions, since the revenue attributed is higher on a last-touch basis than first-touch basis.

SEA	SEA brings in most of the revenue, according to all of the attribution models.
SEO	Compared with SEA, SEO is stronger in terms of briging in revenues on a last-touch basis. These results might suggest that SEA is helping bring new customers to the funnel while SEO is important in keeping customers in the funnel. This might occur, for example, because the travel agency is using general keywords or competitor keywords to bid for advertising.
Price_comparison	Similarly to affiliate marketing, price comparisong sites are 'touched' across the entire journey, and thus seems to be important in creating awareness, increasing consideration, as well as in motivating ultimate purchase decisions.
Other	It is hard to say anything about 'other' based on the data since several channels are included here.

Problem 2

The number of clicks for all channels in the data can be obtained by the following code:

```
sum(session_data$channel=="Affiliate")
sum(session_data$channel=="SEA")
sum(session_data$channel=="Newsletter")
sum(session_data$channel=="Price_comparison")
sum(session_data$channel=="SEO")
sum(session_data$channel=="Other")
sum(session_data$channel=="SEA")
sum(session_data$channel=="Retargeting")
```

For example, there are 57007 SEA clicks in the data. These can be assigned to variables in the customary style, e.g.,

```
click_Affiliate <- sum(session_data$channel=="Affiliate")
```

This is not a very efficient process. Moreover, if new channels are added to the data, the script needs to be edited accordingly. A better way is to use a 'for loop'. (You can learn more about loops via https://www.r-bloggers.com/how-to-write-the-first-for-loop-in-r/ or by taking the *Intermediate R* course on DataDamp: https://www.datacamp.com/courses/intermediate-r.)

We first obtain the channel names, and store them into a vector:

```
channel_names <- model$channel_name
channel_names
```

```
[1] Affiliate        Display          Newsletter       Price_comparison
[5] SEO              Other            SEA              Retargeting
8 Levels: Affiliate Display Newsletter Other ... SEO
```

Next we create a new vector where we will store the number of clicks. We initialize the vector as a set of zeros using the rep() function. The function length() below serves to make the vector as long as there are different channels. We add this vector to the data frame model created earlier (where we stored the attribution modeling results):

```
model$clicks <- rep(0,length(channel_names))
```

The for-loop is as follows:

```
for (i in 1:length(channel_names))
{
      model$clicks[i] <- sum(session_data$channel==channel_names[i])
}
```

The '(i in 1:length(channel_names))' tells R that within the loop we will first set the variable i at 1 and then increase it by one until we reach length(channel_names), which in this case is 8. Using the length() function instead of the number 8 makes the script more flexible with respect to changes in the underlying data.

R then runs the code within the {curved brackets} 8 times (or, more generally, length(channel_names) times), and each time increases the variable i by one. In the first iteration, the number of affiliate clicks is calculated since

```
channel_names[1]
```

```
[1] Affiliate
8 Levels: Affiliate Display Newsletter Other ... SEO
```

In the last iteration of the loop, the number of retargeting clicks is counted.
After running the loop, the number of channels is obtained as follows:

```
model
```

```
      channel_name first_touch_conversions first_touch_value last_touch_conversions
1        Affiliate                     308         21694.670                     339
2          Display                      96          6747.347                       0
3       Newsletter                      37          2629.857                     125
4 Price_comparison                      89          6179.352                      98
5              SEO                     211         14377.487                     260
6            Other                      30          2101.650                      26
7              SEA                     928         65948.307                     851
8      Retargeting                       9           630.990                       9
  last_touch_value linear_touch_conversions linear_touch_value clicks
1       23807.9650                321.15278          22597.830   6130
2           0.0000                 68.19959           4781.804 135432
3        8842.2408                 65.74713           4672.154   7234
4        6689.8022                 91.81667           6323.110   9262
5       17973.5081                236.76634          16287.696  16942
6        1886.3698                 28.58333           2025.755   6323
7       60459.3069                886.23416          62943.294  57007
8         650.4683                  9.50000            678.018    961
```

We can omit the the columns that are not relevant in this problem as follows:

```
model[,c("channel_name","clicks")]
```

```
      channel_name clicks
1        Affiliate   6130
2          Display 135432
3       Newsletter   7234
4 Price_comparison   9262
5              SEO  16942
6            Other   6323
7              SEA  57007
8      Retargeting    961
```

Problem 3

Total click-based cost can be calculated as follows:

```
cost_per_click_total <- cost_data$cost_per_click*model$clicks
```

In Problem 4, affiliate marketing cost is based on commission rather than clicks. Therefore, we create a new vector where we replace the first item with a new value:

```
cost_Problem_4 <- cost_per_click_total
cost_Problem_4[1] <- model$last_touch_value[1]*cost_data$comission_fee[1]
```

These cost data should be compared to revenue attributions. We take a subset of columns from the model dataframe as follows (see Problem 1):

```
model_conversionvalue <- model[,c(1,3,5,7)]
```

Subsequently, we can calculate the ROI for different channels as follows:

```
first_touch_ROI <-(model_conversionvalue$first_touch_value-
cost_Problem_4)/cost_Problem_4
last_touch_ROI <-(model_conversionvalue$last_touch_value-
cost_Problem_4)/cost_Problem_4
linear_touch_ROI <-(model_conversionvalue$linear_touch_value-
cost_Problem_4)/cost_Problem_4
ROI_results <-
data.frame(channel=model_conversionvalue$channel_name,first_touch_ROI,last_
touch_ROI,linear_touch_ROI)
ROI_results
```

```
           channel first_touch_ROI last_touch_ROI linear_touch_ROI
1        Affiliate       0.8224716       1.000000        0.8983420
2          Display      -0.8576545      -1.000000       -0.8991206
3       Newsletter              NA             NA               NA
4            Other              NA             NA               NA
5 Price_comparison              NA             NA               NA
6      Retargeting       0.8759922       0.9339032        1.0158110
7              SEA      -0.2287695      -0.2929604       -0.2639115
8              SEO              NA             NA               NA
```

Affiliate marketing and retargeting, on the one hand, have positive ROI. On the other hand, regardless of the attribution model used, display advertising and SEA are ROI-negative, display ads more so than SEA. Considering SEA, one explanation is that the current keywords and targeting options are a little bit off, bringing people to the funnel that are unlikely to buy, bringing down the ROI as well. Alternatively, the website may be poorly designed, leading to low conversion rates.

Problem 4

Above, we have already calculated the click-based cost for Affiliate, which was stored in cost_per_click_total. We can use this vector and recycle the code above to re-calculate the ROI for affiliate:

```
cost_Problem_5 <- cost_per_click_total
first_touch_ROI <-(model_conversionvalue$first_touch_value-cost_Problem_5)/
cost_Problem_5
```

```
last_touch_ROI <-(model_conversionvalue$last_touch_value-cost_Problem_5)/
cost_Problem_5
linear_touch_ROI <-
(model_conversionvalue$linear_touch_valuecost_Problem_5)/cost_Problem_5
ROI_results_Problem_5 <
data.frame(channel=model_conversionvalue$channel_name,first_touch_ROI,last_
touch_ROI,linear_touch_ROI)
```

We can get the relevant results by simply running 'ROI_results' and 'ROI_results_Problem_5' and comparing the relevant rows. For clener results, we can bind the relevant rows and print the results as follows:

```
ROI_comparison <- rbind(ROI_results[1,],ROI_results_Problem_5[1,])
rownames(ROI_comparison) <- c("Comission","Pay-per-click")
ROI_comparison
```

```
                channel first_touch_ROI last_touch_ROI linear_touch_ROI
Commission     Affiliate       0.8224716      1.0000000        0.8983420
Pay-per-click  Affiliate       0.7695489      0.9419221        0.8432162
```

The current commission-based approach produces slightly better ROIs regardless of the attribution model used. Therefore, the firm should not adopt the pay-per-click model.