HireArt Case Study

Chuyi Guo

3/14/2019

Data Pre-processing.

```
# read in the data into system.
library(readxl)
library(dplyr)
library(ggplot2)
data = read_excel('/Users/ChuyiGuo/Desktop/HireArt/HireArt - Data Analyst Exercise 10.12.17.xlsx')
data = as.data.frame(data)
# extract month, season, year for each line of data
data$month = as.numeric(format(as.Date(data$'Date of Contact'), "%m"))
data$year = as.numeric(format(as.Date(data$'Date of Contact'), "%Y"))

# define function for getting season
# for every 3 monthes, group they as 1 season.
# i.e. month 1,2,3 is in seasion 1
get_season = function(x) {
  if (x %in% c(1,2,3)) {
  data$season = 1
} else if (x %in% c(4,5,6)) {
  data$season = 2
} else if (x %in% c(7,8, 9)) {
  data$season = 3
} else {
  data$season = 4
}
}

data$season = sapply(data$month, get_season)
# extract some of the columns
data = data[,c(colnames(data) %in% c('Client Name', "month", "year", "season"))]
```

According to the spreadsheet, a client may be contacted multiply times within one month. For this situation, the client should be counted as one, based on a monthly basis. Those duplicated records need to be taken care.

```
data2 = data[!duplicated(data),]
```

The numbers of clients for each month of each year are calculated (named n). I.e. the team contacted 9 clients in January 2014.

The numbers of clients for each season of each year are calculated (named season_per_year). I.e. the team contacted 31 clients in the first season (January, February, March) of 2014.

Also, the numbers of clients for each year are calculated (named year_total). I.e. the team contacted 186 clients in the first season (January, February, March) of 2014.

Seasonal and yearly proportions are calculated, named season_prop and year_prop, respectively. Seasonal/yearly proportion is the number of clients for a month divided by the total number of clients for the corresponding season/year. I.e. for January 2014, the Seasonal proportion is $9/31 = 0.29$ and its yearly proportion is $9/184 = 0.05$. It gives a comparative relation to a whole, other than absolute values.

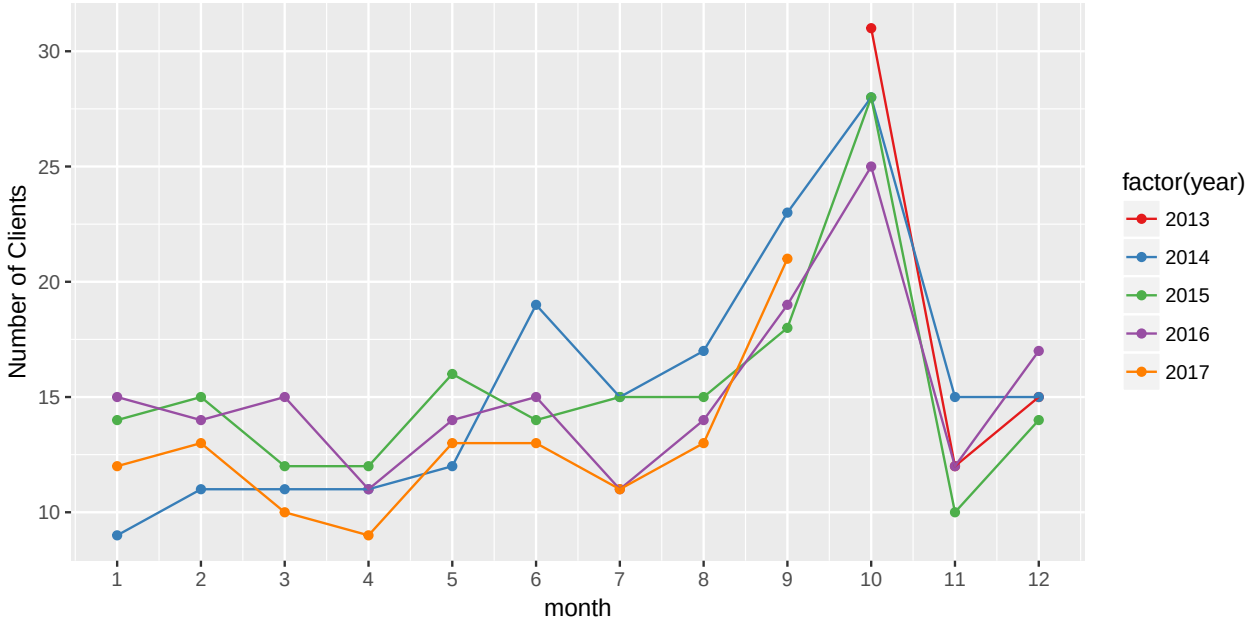
```
total = as.data.frame(
  data2 %>%
    group_by(month, season, year) %>%
    summarise(n = n()) %>%
    group_by (season, year) %>%
    mutate(season_per_year = sum(n)) %>%
    group_by(year) %>%
    mutate(year_total = sum(n)) %>%
    mutate(season_prop = n/season_per_year) %>%
    mutate(year_prop = n/year_total)
)
total
```

```
##    month season year  n season_per_year year_total season_prop  year_prop
## 1      1      1 2014  9              31        186   0.2903226 0.04838710
## 2      1      1 2015 14              41        183   0.3414634 0.07650273
## 3      1      1 2016 15              44        182   0.3409091 0.08241758
## 4      1      1 2017 12              35        115   0.3428571 0.10434783
## 5      2      1 2014 11              31        186   0.3548387 0.05913978
## 6      2      1 2015 15              41        183   0.3658537 0.08196721
## 7      2      1 2016 14              44        182   0.3181818 0.07692308
## 8      2      1 2017 13              35        115   0.3714286 0.11304348
## 9      3      1 2014 11              31        186   0.3548387 0.05913978
## 10     3      1 2015 12              41        183   0.2926829 0.06557377
## 11     3      1 2016 15              44        182   0.3409091 0.08241758
## 12     3      1 2017 10              35        115   0.2857143 0.08695652
## 13     4      2 2014 11              42        186   0.2619048 0.05913978
## 14     4      2 2015 12              42        183   0.2857143 0.06557377
## 15     4      2 2016 11              40        182   0.2750000 0.06043956
## 16     4      2 2017  9              35        115   0.2571429 0.07826087
## 17     5      2 2014 12              42        186   0.2857143 0.06451613
## 18     5      2 2015 16              42        183   0.3809524 0.08743169
## 19     5      2 2016 14              40        182   0.3500000 0.07692308
## 20     5      2 2017 13              35        115   0.3714286 0.11304348
## 21     6      2 2014 19              42        186   0.4523810 0.10215054
## 22     6      2 2015 14              42        183   0.3333333 0.07650273
## 23     6      2 2016 15              40        182   0.3750000 0.08241758
## 24     6      2 2017 13              35        115   0.3714286 0.11304348
## 25     7      3 2014 15              55        186   0.2727273 0.08064516
## 26     7      3 2015 15              48        183   0.3125000 0.08196721
## 27     7      3 2016 11              44        182   0.2500000 0.06043956
## 28     7      3 2017 11              45        115   0.2444444 0.09565217
## 29     8      3 2014 17              55        186   0.3090909 0.09139785
## 30     8      3 2015 15              48        183   0.3125000 0.08196721
## 31     8      3 2016 14              44        182   0.3181818 0.07692308
## 32     8      3 2017 13              45        115   0.2888889 0.11304348
## 33     9      3 2014 23              55        186   0.4181818 0.12365591
## 34     9      3 2015 18              48        183   0.3750000 0.09836066
## 35     9      3 2016 19              44        182   0.4318182 0.10439560
## 36     9      3 2017 21              45        115   0.4666667 0.18260870
## 37    10      4 2013 31              58         58   0.5344828 0.53448276
## 38    10      4 2014 28              58        186   0.4827586 0.15053763
## 39    10      4 2015 28              52        183   0.5384615 0.15300546
## 40    10      4 2016 25              54        182   0.4629630 0.13736264
```

```
## 41    11      4 2013 12              58          58   0.2068966 0.20689655
## 42    11      4 2014 15              58         186   0.2586207 0.08064516
## 43    11      4 2015 10              52         183   0.1923077 0.05464481
## 44    11      4 2016 12              54         182   0.2222222 0.06593407
## 45    12      4 2013 15              58          58   0.2586207 0.25862069
## 46    12      4 2014 15              58         186   0.2586207 0.08064516
## 47    12      4 2015 14              52         183   0.2692308 0.07650273
## 48    12      4 2016 17              54         182   0.3148148 0.09340659
```

Plot out the number of clients per month to get a intuitive overviews for this data set.

```
ggplot(total,aes(x = month, y = n )) +
  geom_point(aes(color = factor(year))) +
  geom_line(aes(color = factor(year))) +
  labs(y="Number of Clients") +
  scale_x_continuous(breaks = seq(1,12,1)) +
  scale_color_brewer(palette = "Set1")
```



Below shows the sum of clients for each month during the four-year period. October has the highest value, which indicates the team is likely to contact the most clients during October.

```
month_total = as.data.frame(
  total %>%
    group_by(month) %>%
    summarise(month_total = sum(n))
)
month_total
```

```
##    month month_total
## 1      1          50
## 2      2          53
## 3      3          48
## 4      4          43
## 5      5          55
## 6      6          61
## 7      7          52
```

```
## 8      8          59
## 9      9          81
## 10    10         112
## 11    11          49
## 12    12          61
```

```
month.abb[which.max(month_total$month_total)]
```

```
## [1] "Oct"
```

In order to avoid the result being affected by extreme values, the averaged seasonal proportions are calculated here. That is, for each season, take the average of its proportions during the 4-year period. Higher proportion means relatively more clients are contacted within a 3-month period.

Again, after comparing seasonal proportion, October has the highest value. It supports previous conclusion.

```
season_avg = as.data.frame(
  total %>%
  group_by(month) %>%
  summarise(avg_prop = mean(season_prop))
)

season_avg
```

```
##    month  avg_prop
## 1      1 0.3288881
## 2      2 0.3525757
## 3      3 0.3185363
## 4      4 0.2699405
## 5      5 0.3470238
## 6      6 0.3830357
## 7      7 0.2699179
## 8      8 0.3071654
## 9      9 0.4229167
## 10    10 0.5046665
## 11    11 0.2200118
## 12    12 0.2753217
```

```
month.abb[which.max(season_avg$avg_prop)]
```

```
## [1] "Oct"
```

Also, the average yearly proportions are calculated here and gives the same conclusion.

```
year_avg = as.data.frame(
  total %>%
    group_by(month) %>%
    summarise(avg_prop = mean(year_prop))
)

month.abb[which.max(year_avg$year_prop)]
```

```
## character(0)
```

Therefore, it can be concluded that October is the month that the team is likely to contact the most clients.