MMMデータセットの探索的データ分析

ソースコード:

https://github.com/yu-ya-tanaka/Robyn-Community-Japan-Resource/tree/main/MMM_EDA_template

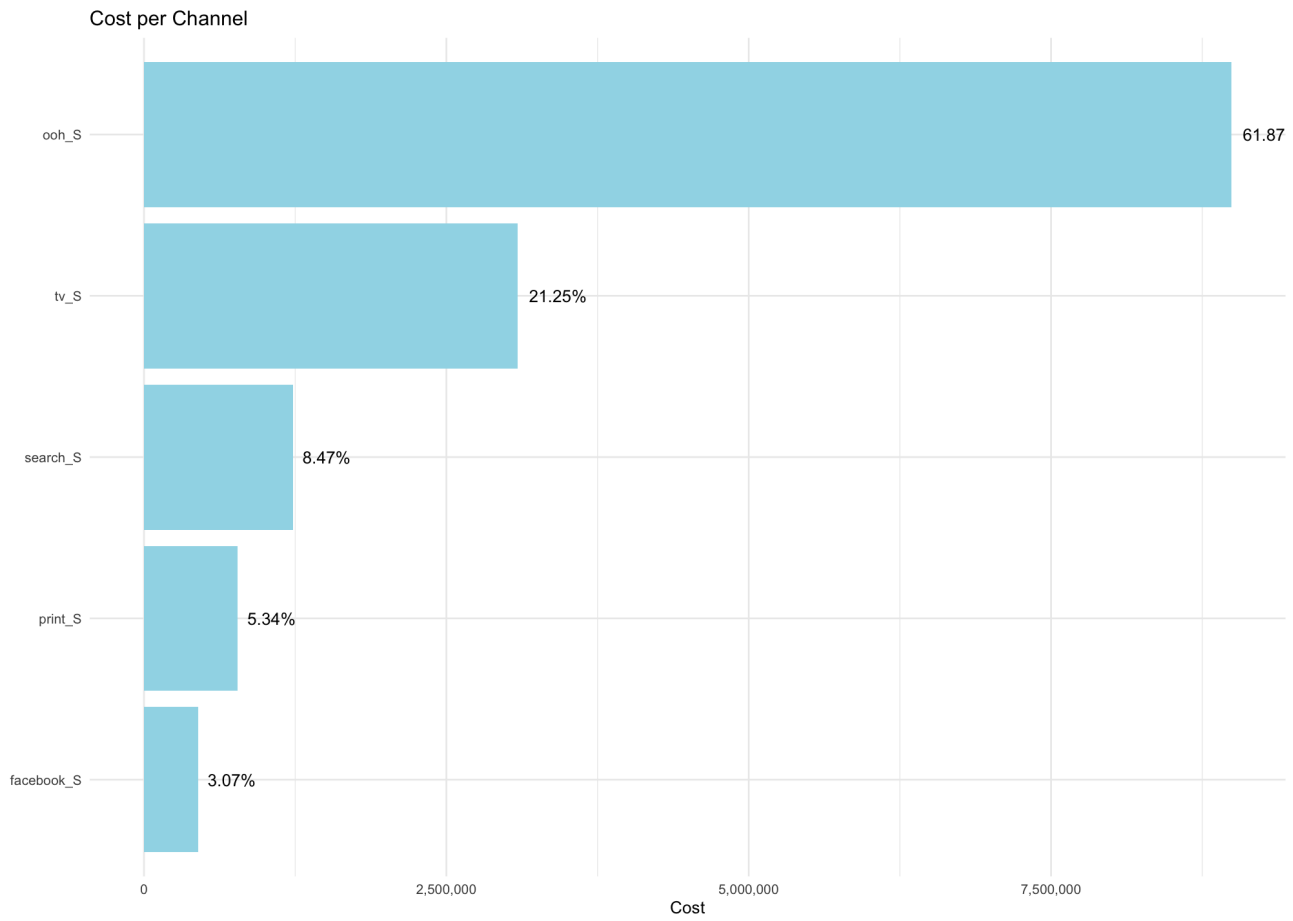
空白とゼロのレコードをカウント
目的：データセットがMMMでの分析に適するか判断する
　　（空白がある場合はそれを埋める方法を検討し、ゼロが多い場合は除外を検討する）

```
###### 5. データチェック) 空白とゼロのレコードをカウント
count_summary <- data %>%
  summarize(across(
    .cols = everything(),
    .fns = list(
      zeros = ~sum(.x == 0, na.rm = TRUE),           # 0の個数
      blanks = ~sum(is.na(.x) | .x == "", na.rm = TRUE), # NAまたは空文字の個数
      others = ~sum(!(.x == 0 | is.na(.x) | .x == ""), na.rm = TRUE) # それ以外の個数
    ),
    .names = "{.col}___{.fn}"  # 新しいカラム名のフォーマット
  )) %>%
  pivot_longer(
    cols = everything(),
    names_to = c("column", "type"),
    names_sep = "___",
    values_to = "count"
  ) %>%
  pivot_wider(
    names_from = type,
    values_from = count
  ) %>%
  arrange(column)  # カラム名でソート

print(count_summary, n=100)
```

	column	zeros	blanks	others
	<chr>	<int>	<int>	<int>
1	DATE	0	0	0
2	competitor_sales_B	0	0	208
3	events	0	0	208
4	facebook_I	106	0	102
5	facebook_S	107	0	101
6	newsletter	0	0	208
7	ooh_S	123	0	85
8	print_S	121	0	87
9	revenue	0	0	208
10	search_S	32	0	176
11	search_clicks_P	32	0	176
12	tv_S	116	0	92

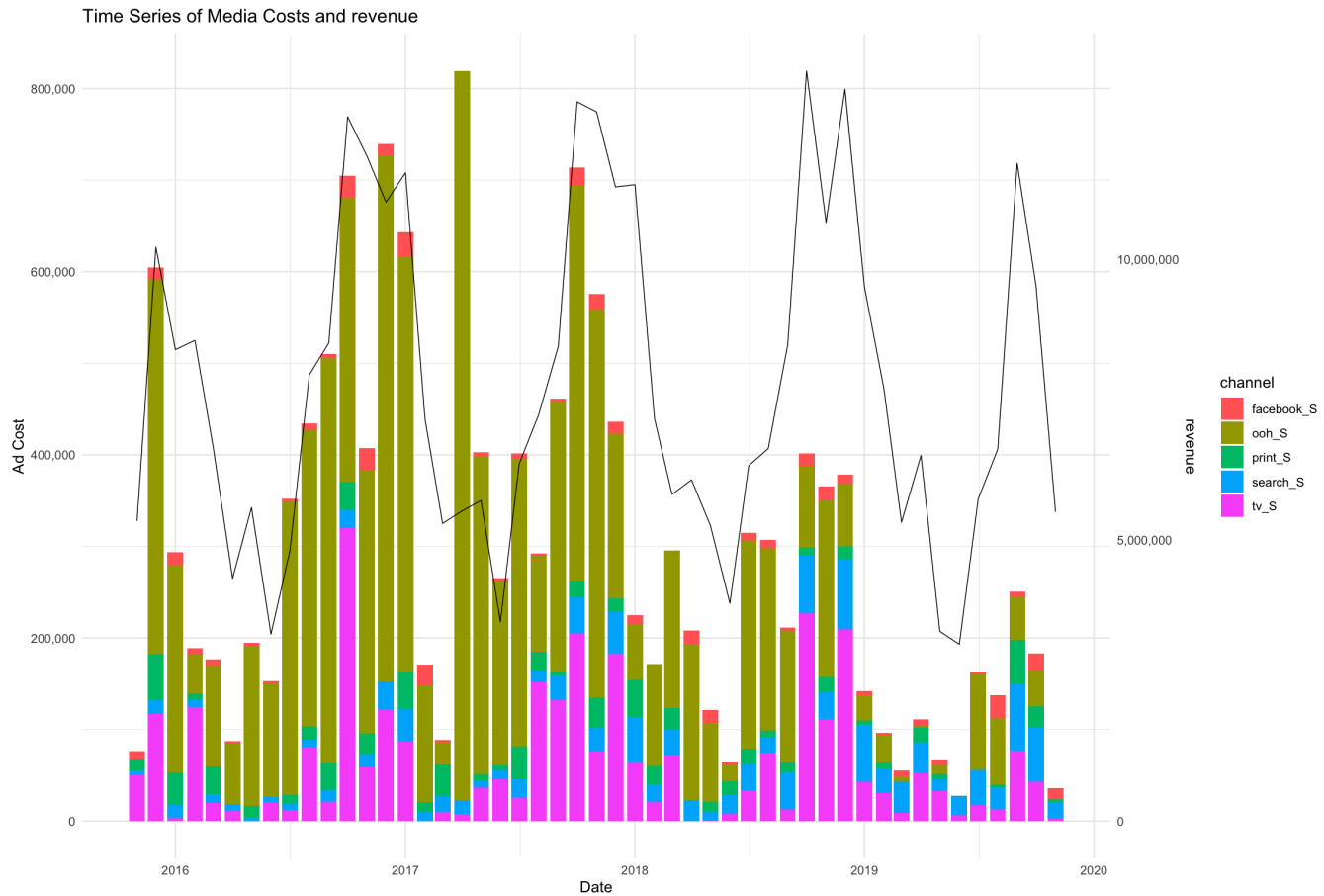
メディア毎の出稿金額の割合の可視化
目的：どのメディアの出向割合が高いかを把握する

```
##### 可視化1. メディア毎のコスト内訳を確認
# ggplot2でのプロット作成
p <- ggplot(total_cost, aes(y = reorder(channel, total_cost), x = total_cost)) +
  geom_col(fill = "lightblue") +  # 棒グラフを描画
  geom_text(aes(label = paste0(round(percentage, 2), "%")), hjust = -0.2, nudge_x = 50) +
  labs(title = "Cost per Channel", x = "Cost", y = "") +
  theme_minimal() +
  scale_x_continuous(labels = comma)

p
# ggplotly(p)
```



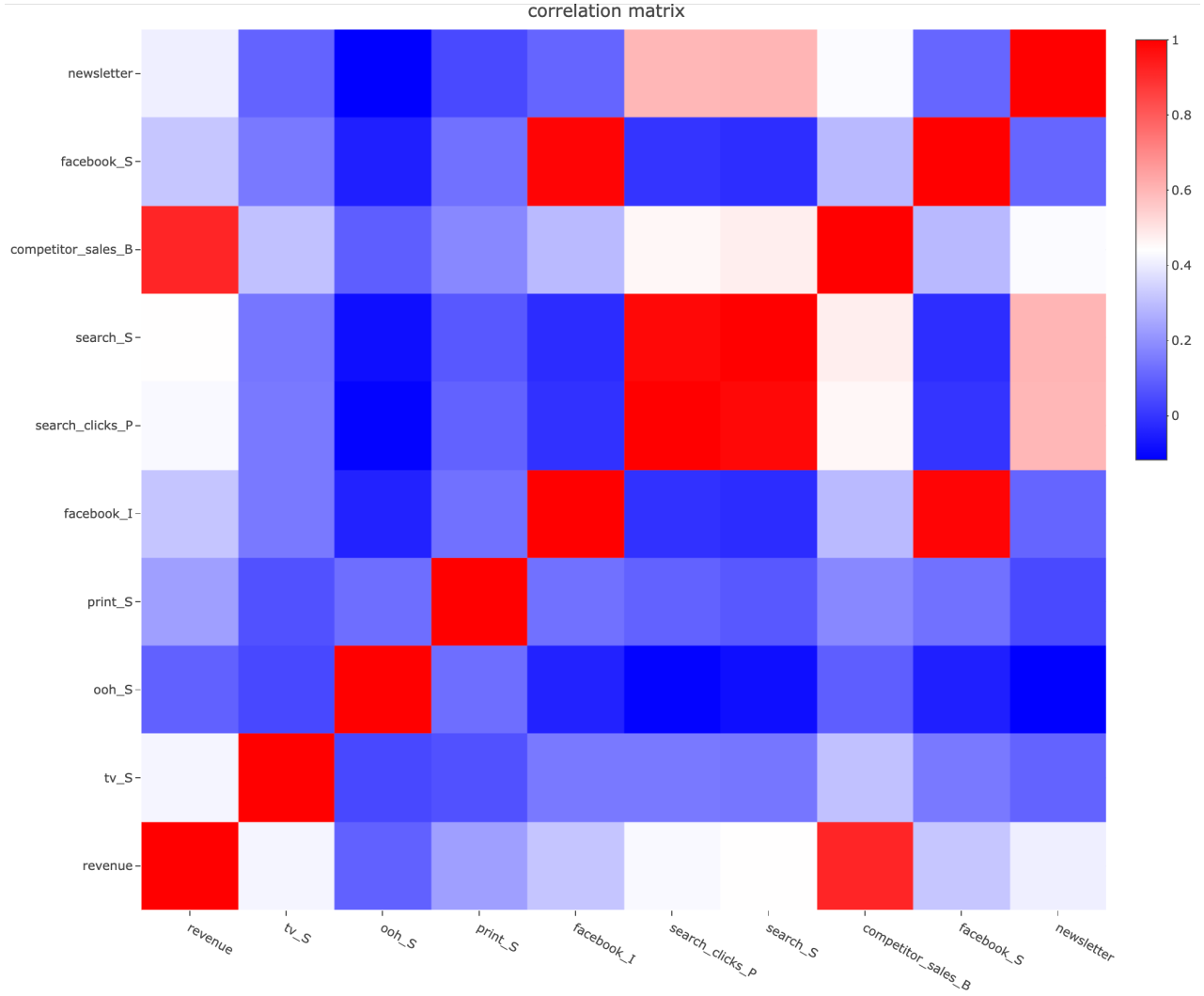
目的変数（KGI）とコスト内訳の時系列推移
目的：KGIとコスト内訳の推移から、どのメディアの出稿金額が増えたときにKGIが変化するかの傾向を把握する

```r
##### 可視化2. KGIとコスト全体の時系列推移を確認
# グラフ作成関数の定義
# 1.コストを積み上げで表現
create_time_series_plot <- function(data, date_col, kgi_col, cost_cols) {...}
# 2.コストを全体を100%とした時の内訳で表現
create_time_series_plot_breakdown <- function(data, date_col, kgi_col, cost_cols) {...}

# 週次/月次への変換を行わない場合
p <- create_time_series_plot(data, date_col, kgi_col, cost_cols)
p
# ggplotly(p)
```



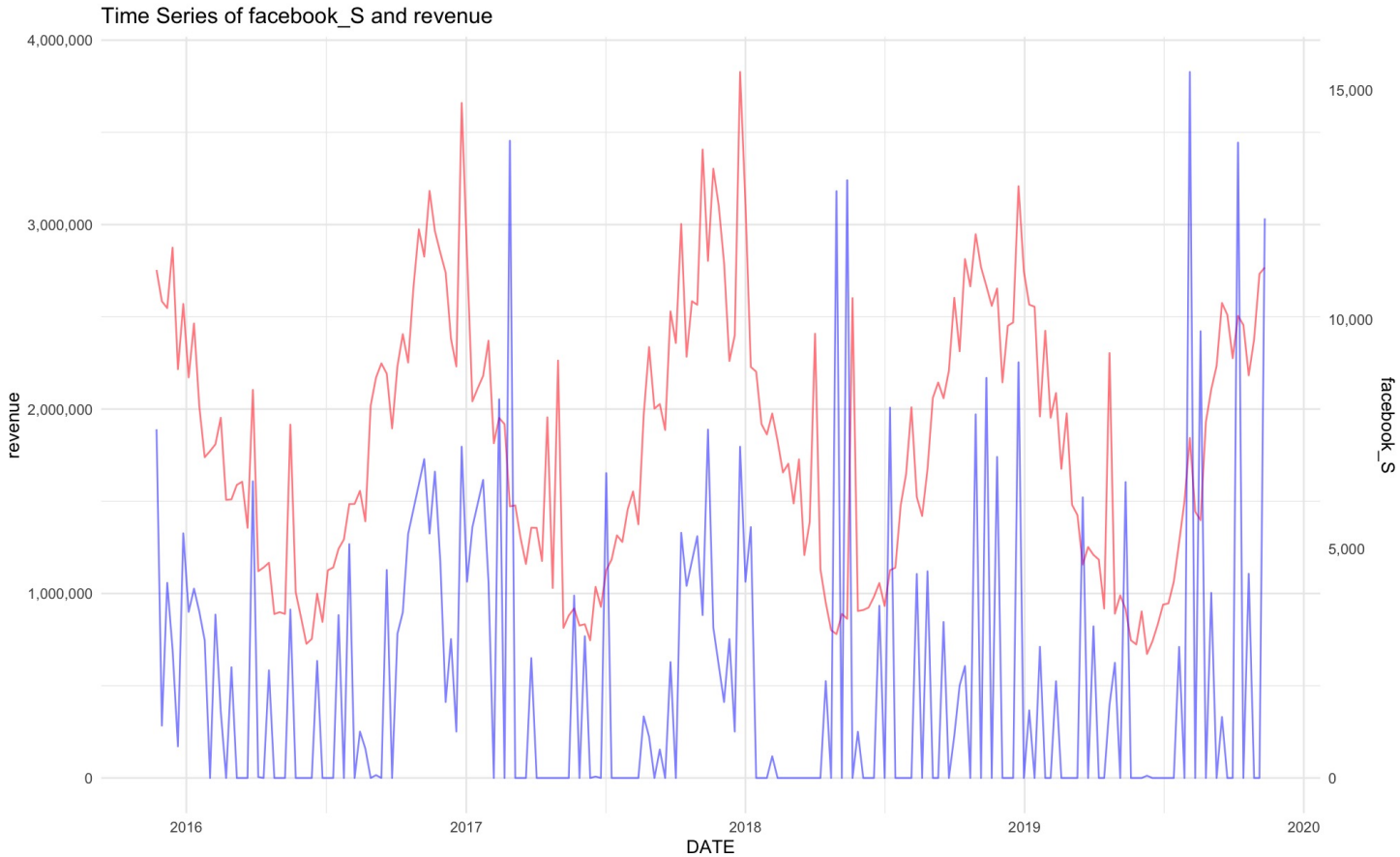
相関行列
目的：相関の高い変数の組み合わせを可視化し、モデリングで利用する変数を検討する
　　（相関の高い変数のうち一つを除外する）

```
##### 可視化3. 相関行列
cor_matrix <- data %>%
  select(-date_col) %>%
  select_if(is.numeric) %>%
  cor()

p <- plot_ly(x = colnames(cor_matrix),
             y = rownames(cor_matrix),
             z = cor_matrix,
             type = "heatmap",
             colors = colorRamp(c("blue", "white", "red")))
p <- p %>% layout(title = 'correlation matrix')
p
```



目的変数と説明変数の時系列推移
目的：目的変数と説明変数の関係を把握する

```
##### 可視化4. KGI x 一つの説明変数の組み合わせで推移を確認
# 日付と数字のカラムに限定
data_date_and_numeric <- data %>%
  select_if(is.numeric) %>%
  mutate(DATE=data[[date_col]]) %>%
  select(DATE, everything())

# カラム数を取得
num_cols <- ncol(data_date_and_numeric) - 2

# 各メディアコストごとにグラフを作成
for (i in 1:num_cols) {
  # メディアコストのカラム名を取得
  cost_col <- names(data_date_and_numeric)[i + 2]  # Date と KGI をスキップ

  # scaling
  max_cost <- data_date_and_numeric[[cost_col]] %>% max(na.rm = TRUE)
  max_KGI <- data_date_and_numeric[[kgi_col]] %>% max(na.rm = TRUE)
                    max_KGI / max_cos
```



ヒストグラム
目的：データのばらつきを把握する
（ばらつきが小さい変数の場合、MMMでモデリングがうまくできない場合があるため）

```r
##### 可視化5. ヒストグラムで変数毎にばらつきを確認
# 日付と数字のカラムに限定
data_only_numeric <- data %>%
  select_if(is.numeric)

# 各数値カラムごとにヒストグラムを作成
for (i in 1:ncol(data_only_numeric)) {
  # 数値カラムのカラム名を取得
  col_name <- names(data_only_numeric)[i]

  # ヒストグラム用のデータの最大値を取得
  max_count <- max(table(data_only_numeric[[col_name]]))

  # ggplotでヒストグラムとカーネル密度推定を描画
  p <- ggplot(data_only_numeric, aes_string(x = col_name)) +
    geom_histogram(bins = 30, fill = "blue", color = "black", alpha = 0.7) +
    theme_minimal() +
    labs(title = paste("Histogram of", col_name), x = col_name, y = "Frequency") +
    scale_x_continuous(labels = scales::comma)

  # プロットを表示
  print(col_name)
  print(p)
}
```