Part 1: Introduction

- 1.Introduce our team
- 2.what is dateset we use?

  [1.1.Goal]

we get this dataset from one kaggle competition where we are challenged to analyze a Google Merchandise Store customer dataset to predict revenue per customer.

- 3.Data Fields

  [1.2.Data Fields]

- 4.library

  [1.3.Libraries]

  we use data.table, dplyr, tidyr and jsonlite packages to maniplate and preprocess our data, use ggplot2 to do visualization and use knitr(Rmarkdown) and kableExtra to present our result.

  `check.packages` is a function we wrote that can automatically check missing packages and install them.

- a glimpse into data

[2.1.Load data and have a glimpse]

as we see, it is a 'dirty' dataset, so we need to preprocess it before feeding to our statistical model or machine learning algorithm.

`device, geoNetwork, totals, trafficSource` features are in JSON format, missing value in `socialEngagementType` feature is a string "Not Socially Engaged" and `date` feature is numeric which is not correct.

Besides, it is a time seires data set, we can find a userid in different date, which need to be merged.

Part 2: Data Processing

- 1.data type conversion

  [2.3.Cast data type]

  We can use jsonlite package to parse and flatten json format feature and make it readable for us and analyzable for our model.

`:=` is assignment by reference in R (it can be used for `data.table` object), which helps to fast add, remove and update subsets of columns by reference.

so it means drop columns 'device', 'geoNetwork', 'totals', 'trafficSource' and assign to train_data

```
# drop the old json columns
train_data[, c('device', 'geoNetwork', 'totals', 'trafficSource') := NULL]
```

However, since some features use string (e.g."not available") instead of `NA` in data(so the data type of the whole column becomes `string`), we need to set them to `NA`, and then convert these features into numeric.

- 2. Missing value process

  [2.3.Preprocessing missing value]

  - 1. First, we set all unknown value to build-in type `NA`. Some of the newly parsed columns from json have various values that can be converted to NA. This includes values such as '(not set)' and 'not available in demo dataset'. Although distinguishing between these values may be useful during modeling, we are going to convert them all to NA for the purposes of visualization.
  - set function(optional) .

    set function is roughly equal to `:=`

    i -> Indicates the rows on which the values must be updated with(If not provided, implies *all rows*.)

    j -> Indicates column name or column index to be assigned value

    value -> replacement value(s) to assign by reference to `x[i, j]`.

    ```
    set(train_data, i=which(train_data[[col]] %in% na_vals), j=col,
    value=NA)
    ```

  - 3.Drop some useless features. Several of the columns newly parsed from json have only 1 unique value, e.g. 'not available in demo dataset'. These columns are obviously useless, so we drop them here.
  - 4.data type conversion

    For some features, we will need to change them to numeric by using data.table operation.

```
# character columns to convert to numeric
num_cols <- c('hits', 'pageviews', 'bounces', 'newVisits',
              'transactionRevenue')

# change columns to numeric(apply as.numeric on the whole list(column)
of each num_cols, and update the value)
train_data[, (num_cols) := lapply(.SD, as.numeric), .SDcols=num_cols]
```

- 5.simply compare final result with first one

- 6.Missing data visualization

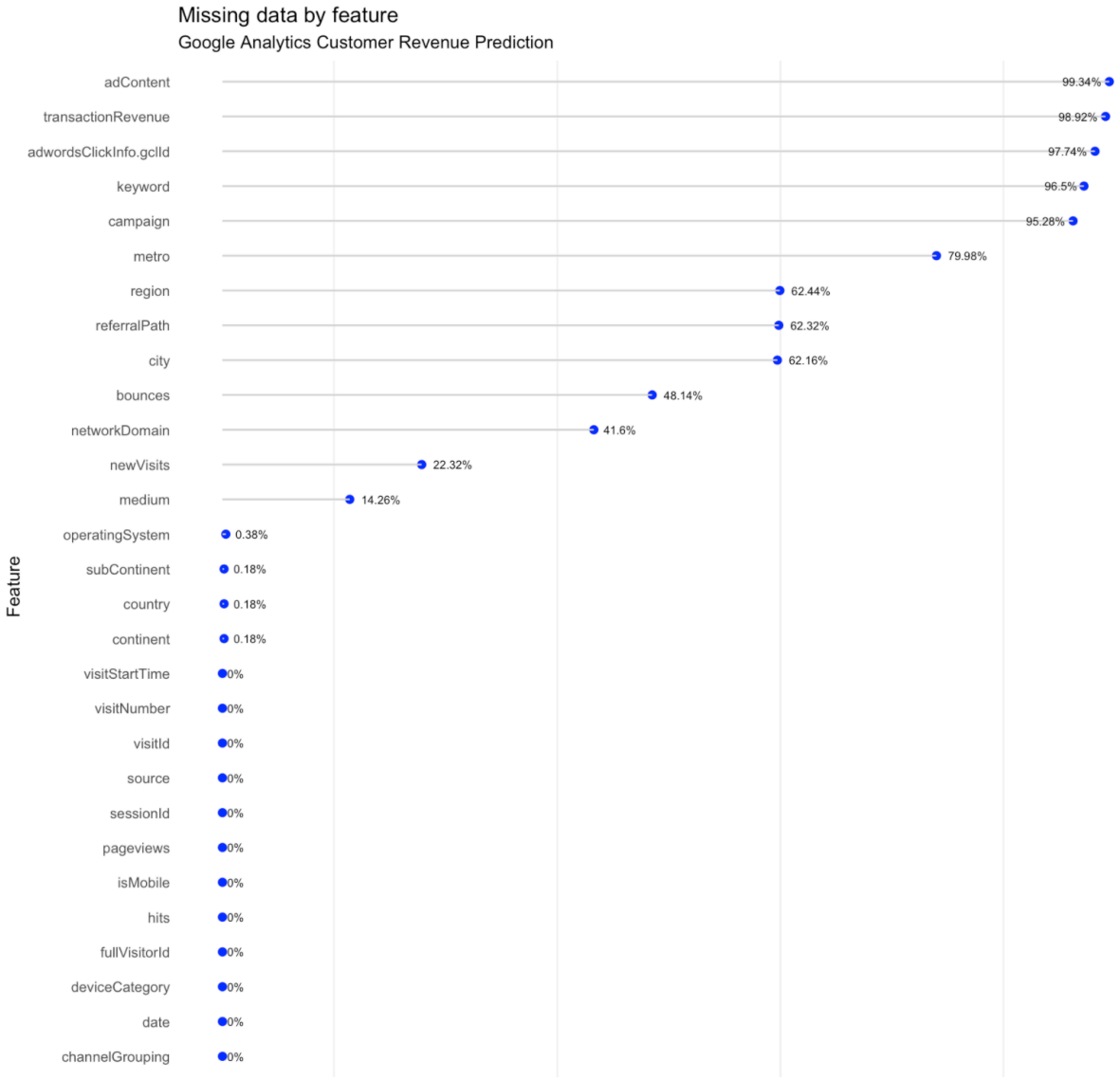
  [2.4.Missing data visualization]

  For get better visualization, we convert transactionRevenue back to unit dollars.

  ```
  # Divide transactionRevenue by 1,000,000
  train_data[, transactionRevenue := transactionRevenue / 1e+06]
  ```

  We use ggplot to draw a Cleveland Dot Plot and Bar chart

As we can see, some features have lots of missing value and some doen't. (Describe it)



Part 3: Data Exploration(I)

Data Exploration on Revenue

- 1.Data description:
  - there are 90k rows in this dataset, time period is between August 2016 and August 2017
- 2.Target data(Revenue) exploration
  - We merge revenue (grouped by UserID), sort them based on the sum of revenue and plot them. (because it is too skewed to use histgram!)
  - According to the chart, the distribution of each user's revenue is extremely skewed. Then we employ Shapiro–Wilk test on it to test whether the user's revenue are normally distributed. Apparently, it refuse the null hypothesis that the sample are came from normally distributed population.
  - The log transformation is one of the most popular among the different types of transformations used to transform skewed data to approximately conform to normality. It is widely used in financial data. Now, we try it on our data.
  - The mean of the natural log of transaction revenue appears to be around 4 and is shown a beautiful bell-shaped curve. Then, we try Shapiro–Wilk test on it, and result is

greater than 0.1 or 0.05. We now can believe these log-transformed data was generated from normally distributed population, and employ linear regression model on it in the future.

- Temporal pattern of Revenue
  - The daily revenue data are pretty volatile, but there appears to be a regular pattern here. There seems to be a regular pattern of highs and lows. We'll have to take a closer look at this. The smoothing line indicates that daily revenue, fluctuations aside, has remained fairly steady over the course of the year.
- Revenue by channel
- Revenue by device
- Revenue by geographical features
- Revenue by network domain
- Revenue by traffic

Part 4: Data Exploration(II)

Correlation analysis

- 1.intro; Correlation analysis is a statistical method used to evaluate the strength of relationship between two quantitative variables, which helps to figure out the relationship between numeric features and revenue.
- 2.Select numeric feature. Coz correlation analysis is used for analyzing quantitive features, so we select the numeric features first.

```
numeric.train_data <- train_data[, .SD, .SDcols = sapply(train_data, is.numeric)]
```

- 3. Drop useless features. (apparently, things like user_id are numeric, but useless for modeling)

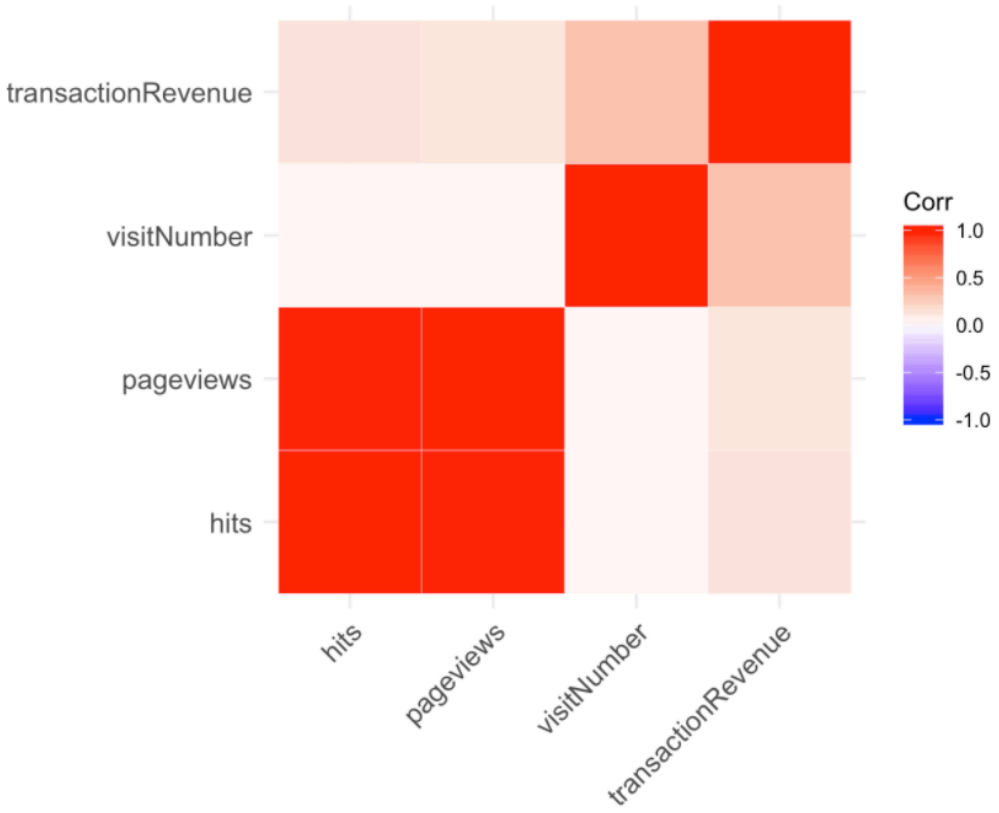
  And we know userId is not a 'real' feature, so they should be dropped. Besides newVisits and bounces are unique value feature(NA and 1), so it should be consider as a binary category feature after one-hot encoding, and ,thus, we remove it as well.

- 4.Then we calculate their correlation matrix by Pearson's method.

```
corr <- round(cor(numeric.train_data, use="pairwise.complete.obs"), 3)
show(corr, caption = "Correlation Matrix of numeric features and revenue")
```

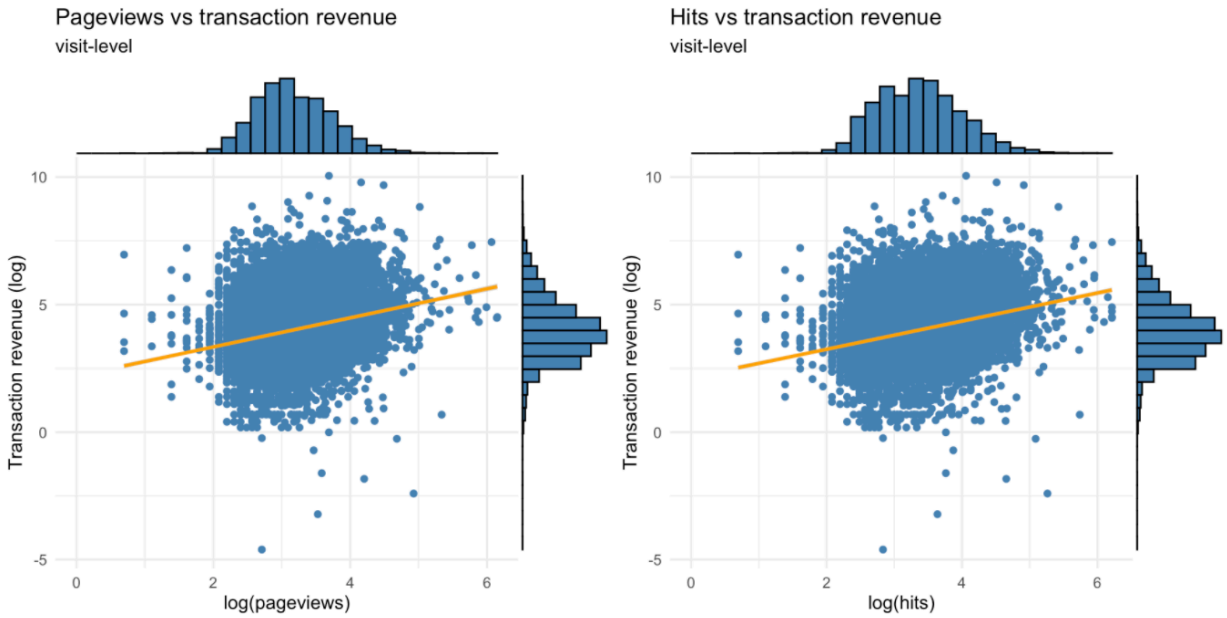
Correlation Matrix of numeric features and revenue

	visitNumber	hits	pageviews	transactionRevenue
visitNumber	1.000	0.041	0.043	0.309
hits	0.041	1.000	0.983	0.142
pageviews	0.043	0.983	1.000	0.129
transactionRevenue	0.309	0.142	0.129	1.000



- 5. Emphasize on `Pageviews and Hits` features. because they have quite high correlation coefficients.

  Pageviews and Hits are both the behaviors of users while on the site and look for anything, so it might be correlated with transaction revenue. Below are bivariate distribution plots of pageveiews versus transaction revenue, and hits versus transaction revenue.



It seems like they do have a `positive correlation`. It's not that easy to tell from the cloud of points whether there is relationship between hits and revenue and between pageviews and revenue. For the next step, we will fit a linear model to the data indicate that there is some positive correlation between the two in both cases.

- summary

In terms of further work, we will employ some techniques of feature engineering, such as bucketing, missing data imputation, feature selection and feature construction. Also, we will employ statistical regression methods, such as lasso, elastic net regression and other machine learning methods.