Ryan McNamara (rjm2232) Matthieu Schulz (mhs2216)



Basics of data.table

DataExplorer is designed to work with data.tables but also works with data.frames. Functions in **DataExplorer** will update data.tables in place but will return a new data.frame.

Install data.table package for R

install.packages('data.table')

Create **data.table** (similar to data.frame)

```
dt = data.table(
      a = c(-4, -9, 2, 73, 3),
      b = c(4, 2, 0, 100, -2),
      c = c("cat", "dog", "cat", "fish", "fish"),
      d = c(8, 7, 2, 10, 2)
```

data.table Syntax

Subsetting Rows

dt[name %in% c('b', 'c') & id > 0]

Selecting Columns

dt[,.(b,c)]

Compute on Columns

dt[,.(mean(b), sum(d))]

Rename Columns

dt[,.(mean_id = mean(b), score = d)]

Grouping using by

dt[,.(mean(d)), by = .(c)]

Sorting using keyby

dt[, .(c), keyby = .(d)]

Expressions in by

dt[, .(mean(b)), by = .(d > 7)]

Multiple columns using .SD

dt[, lapply(.SD, mean), by = .(c)]

Create Report

DataExplorer allows you to create a summary report of a data.table using only two functions:

configure_report and create_report. For more detailed explanations on what can be included in the report see the last column of this cheat sheet.

```
configure_report(
       add introduce = TRUE.
       add plot intro = TRUE.
       add_plot_str = TRUE,
       add_plot_missing = TRUE,
       add_plot_histogram = TRUE,
       add_plot_density = FALSE,
       add_plot_qq = TRUE,
       add_plot_bar = TRUE,
       add_plot_correlation = TRUE,
       add plot prcomp = TRUE.
       add_plot_boxplot = TRUE,
       add_plot_scatterplot = TRUE,
```

Note: Other arguments include plot configuration and theme configuration.

```
create_report(
        data,
        output_format = html_document(...),
        output_file = 'report.html',
        output_dir = getwd(),
        y = NULL
        config = configure_report(),
        report_title = 'Data Profiling Report',
```

Note: Other arguments include other arguments to be passed to render

```
Data Profiling Report
Basic Statistics
```

Preprocessing

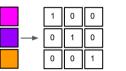
a = rnorm(100, 10, 5),b = as.factor(sample(c("cat", "dog", "fish"), 100, replace = TRUE)), c = sample(c(1, 3, NA), 100, replace = TRUE),d = c(rep("c1", 60), rep("c2", 25), rep("c3", 10), rep("c4", 4), NA)

Table Operations



drop_columns(data, ind)

Drops specified columns drop_columns(test, 'c')



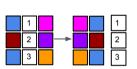
dummifv(data, maxcat, select)

One-hot encodes specified columns dummify(test, ind = 'b')



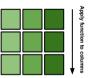
set_missing(data, value, exclude = NULL)

Set missing values set missing(test, value = 2, exclude = 'd')



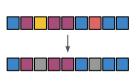
split columns(data, binary as factor = FALSE)

Split data into numeric and categorical columns split = split columns(test) split\$discrete, split\$continuous



update_columns(data, ind, what)

Update specific columns using a update columns(test, c('a', 'c'), log)



group category(data, feature, threshold, measure, update = FALSE, category_name = 'OTHER'. exclude = NULL)

Group sparse categories for discrete feature based on a threshold group_category(test, 'd', 0.1, update =

Summary Operations

introduce(data)

Get basic information for input data: rows, columns, discrete_columns, continuous columns, etc. introduce(test)

profile missing(data)

Get missing value profile: frequency, percentage, suggested profile_missing(test)

Visualizations

a = c(rnorm(95, 10, 5), rep(NA, 5)),

b = as.factor(sample(c("car", "boat", "tank"), 100, replace = TRUE)),

c = sample(c(4, 9, NA), 100, replace = TRUE),

d = c(rep("a1", 50), rep("a2", 32), rep("a3", 8), rep("a4", 9), NA)



plot_bar(data, with, by, by_position, maxcat, order_bar, binary_as_factor, ...)

Bar Chart for discrete features, based on frequency or another continuous feature plot_bar(test2)



plot_boxplot(data, by, binary_as_factor,

Generates a boxplot for each continuous feature based on a selected feature plot boxplot(test, by = 'b')



plot_correlation(data, type, maxcat, ...)

Creates a correlation heatmap for all discrete plot correlation(test2)

plot_density(data, binary_as_factor, ...)



Creates plot density estimates for each of the continuous features

plot density(test2)

plot_histogram(data, binary_as_factor, Generates a histogram for all of the continuous

plot histogram(test2, scale x = (log10))





Plots basic information (uses introduce()) for the

data being inputted plot_intro(test2)



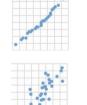
plot_missing(data, ...) Generates a boxplot for each continuous

feature based on a selected feature. plot missing(test2)



plot_prcomp(data, maxcat, nrow, ncol, ...)

Generates the visualization of prcomp plot prcomp(na.omit(test2))



plot_qq(data, by, sample_rows, ...) Creates a Quantile-Quantile plot for each of the

continuous features plot_qq(test2)

plot scatterplot(data, ...)

Generates a scatterplot for all features based on a selected feature plot_scatterplot(test2, by = 'a')



plot_str(data, type, max_level_, print network, ...)

Generates a D3 network graph to visualize data structure plot_str(test2)

