

# DataExplorer :: CHEAT SHEET

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

```
install.packages('data.table')

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
  - Raw Counts
  - Percentages
- Data Structure
- Missing Data Profile
- Univariate Distribution
  - Histogram
  - Bar Chart (with frequency)
  - QQ Plot
- Correlation Analysis
- Principal Component Analysis

### Basic Statistics

Name	Value
Rows	234
Columns	11
Discrete columns	6
Continuous columns	5
All missing columns	0
Missing observations	0
Complete Rows	234
Total observations	2,574
Memory allocation	24.2 Kb

## Preprocessing

```
test = data.table(
  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')`

**dummify(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 function  
`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 = TRUE)`

## 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 action  
`profile_missing(test)`

## Visualizations

```
test2 = data.table(
  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 categories  
`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 features  
`plot_histogram(test2, scale_x = 'log10')`

**plot\_intro(data, ...)**  
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)`

