R Tools for Automated Exploratory Data Analysis

Mateusz Staniak (mtst@mstaniak.pl)
Wrocław 21 I 2020

Navigation

Current Issue
Accepted articles
Archive
R News
News and Notes
Submissions
Reviews and Proofreading
Editorial Board

Subscribe

RSS Feed

ISSN: 2073-4859

The R Journal: accepted article

This article will be copy edited and may be changed before publication.

The Landscape of R Packages for Automated Exploratory Data Analysis

Mateusz Staniak and Przemysław Biecek

Abstract The increasing availability of large but noisy data sets with a large number of heterogeneous variables leads to the increasing interest in the automation of common tasks for data analysis. The most time-consuming part of this process is the Exploratory Data Analysis, crucial for better domain understanding, data cleaning, data validation, and feature engineering. There is a growing number of libraries that attempt to automate some of the typical Exploratory Data Analysis tasks to make the search for new insights easier and faster. In this paper, we present a systematic review of existing tools for Automated Exploratory Data Analysis (autoEDA). We explore the features of fifteen popular R packages to identify the parts of analysis that can be effectively automated with the current tools and to point out new directions for further autoEDA development.

Received: ; online 2019-08-17, supplementary material, (1.6 Kb)
CRAN packages: cranlogs, radiant, visdat, archivist, xtable, arsenal, DataExplorer, dataMaid, dlookr, ExPanDaR, explore, shiny, exploreR, funModeling, inspectdf, RtutoR, SmartEDA, data.table, summarytools, knitr, ggplot2, xray, tableone, describer, skimr, prettyR, Hmisc, ggfortify, autoplotly, gpairs, GGally, survminer, cr17, DALEX, iml
CRAN Task Views implied by cited CRAN packages: ReproducibleResearch, MissingData, TeachingStatistics, WebTechnologies, Bayesian, ClinicalTrials, Econometrics, Finance, Graphics, HighPerformanceComputing, Multivariate, OfficialStatistics, Phylogenetics, SocialSciences, Survival

Faces of automation

- Crunchbase lists over 5,000 startups who are relying on machine learning for their main and ancillary applications, products and services today.
- 81% of machine learning startups Crunchbase tracks have had two funding rounds or less with seed, angel and early-stage rounds being the most common.
- According to KPMG's Venture Pulse Report,

  venture capital (VC) investment in artificial intelligence almost doubled in 2017, attracting $12B compared to $6B in 2016.
- Q2'18 was a second-straight record quarter for total Artificial Intelligence (AI) funding with total investments exceeding $2.3B including eight mega-rounds over $100M according to the latest PwC/CB Insights MoneyTree Report from Q2 2018

https://www.forbes.com/sites/louiscolumbus/2018/08/26/25-machine-learning-startups-to-watch-in-2018/

D/SRUPTION

/ Articles / Resources / Magazine / Events / Partners / About Us

4) Benevolent AI – pharmaceuticals

Benevolent AI is one of many startups disrupting the pharmaceutical industry. The company applies machine learning to improve the way that medicines are discovered, developed, tested and brought to market via several different steps. These include processing and modelling bioscience data, to give scientists hypotheses and ideas to explore; understanding the biology of a disease; finding the best responders for specific drug treatments in patients ahead of clinical trials; and designing molecules to ensure that drugs have the best chance of efficacy in patients. This not only speeds up the discovery and delivery of treatments, but also ensures they are more effective. Benevolent AI is based in London, with a research facility in Cambridge, and further offices in New York and Belgium. In 2018 the startup raised $115m, bringing up its funding total to more than $200m.

5) Hunters.AI – cybersecurity

Hunters.AI is an Israeli startup on a mission to protect the world from cybersecurity threats. Its AI-driven threat hunting technology constantly searches an organisation's systems for security breaches, and identifies attacks as soon as they are attempted. This gives companies the best chance to mitigate damage and isolate any serious risks. The technology also provides a full report of any incidents which do occur, detailing the timeline, location, risk level, target and any recommended actions. Hunters.AI was founded in 2018 and raised $5.4m in its first seed round.

6) Pony.ai – autonomous vehicles

https://disruptionhub.com/10-machine-learning-startups-transforming-industries/

Development of a Machine Learning Model Predicting an ICU Admission for Patients with Elective Surgery and Its Prospective Validation in Clinical Practice

Article Aug 2019

Stefanie Jauk · Diether Kramer · Günther Stark · [...] · Johann Kainz

Frequent utilization of the Intensive Care Unit (ICU) is associated with higher costs and decreased availability for patients who urgently need it. Common risk assessment tool, like the ASA score, lack objectivity and do account only for some influencing parameters. The ai.....

21 Reads

Request full-text Recommend Follow Share

Predicting Chemical Reaction Barriers with a Machine Learning Model

Article Mar 2019

Aayush R. Singh · Brian A. Rohr · Joseph A. Gauthier · Jens K. Nørskov

In the past few decades, tremendous advances have been made in the understanding of catalysis at solid surfaces. Despite this, most discoveries of materials for improved catalytic performance are made by a slow trial and error process in an experimental laboratory.... ...

57 Reads · 2 Citations

Request full-text Recommend Follow Share

Part of the project: Optimal Machine Learning model for software defect prediction

Research from: Tripti Lamba's Lab

Optimal Machine learning Model for Software Defect Prediction

Article Full-text available Feb 2019

Tripti Lamba · Dr. kavita · A.K. Mishra

Machine Learning is a division of Artificial Intelligence which builds a system that learns from the data. Machine learning has the capability of taking the raw data from the repository which can do the computation and can predict... ...

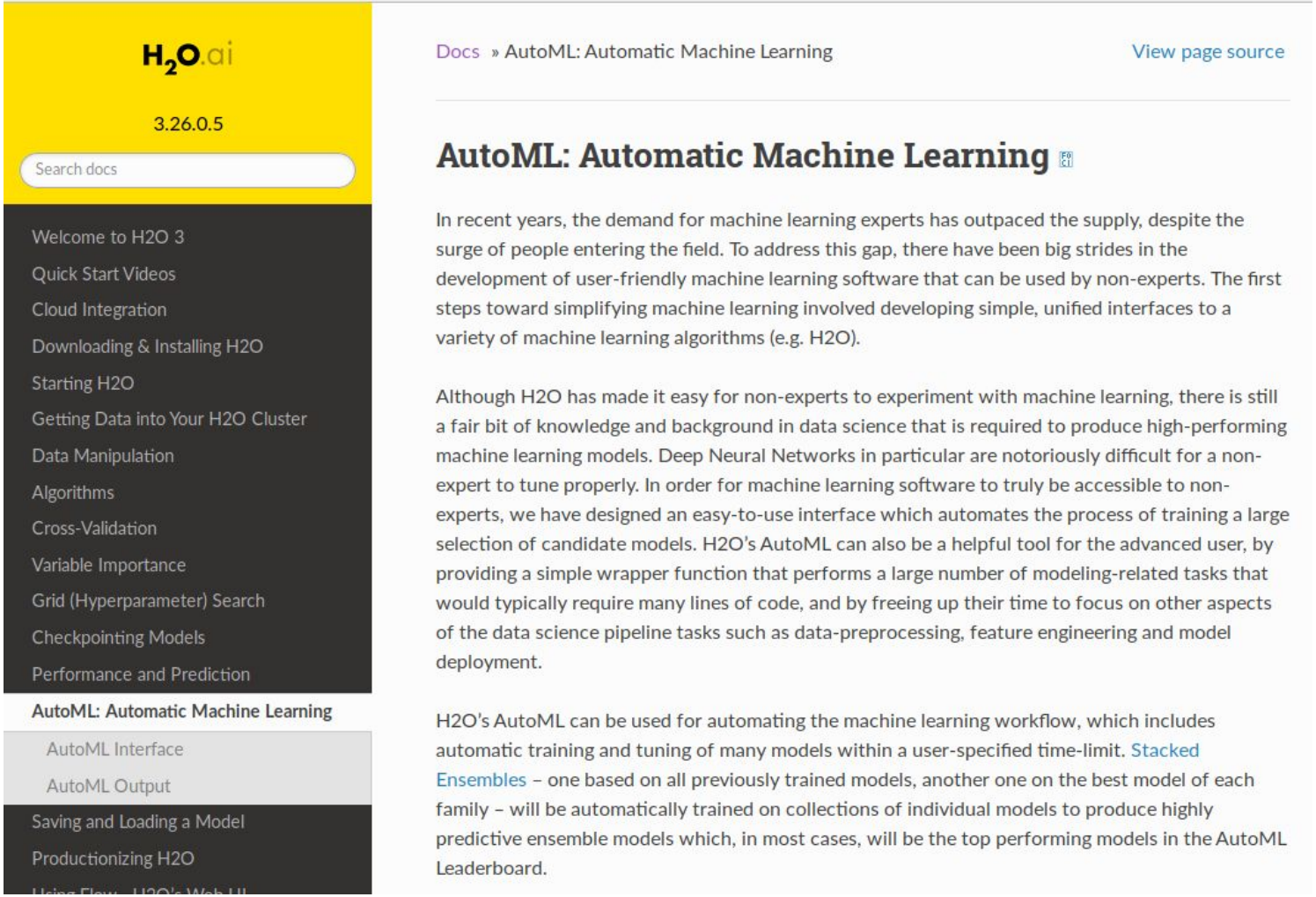
1 Recommendation · 101 Reads

Download Recommend Follow Share

Source

MACHINE LEARNING MODELS FOR PREDICTING FINANCIAL DISTRESS

https://www.researchgate.net



http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html

```
java -cp autoweka.jar weka.classifiers.meta.AutoWEKAClassifier
    -timeLimit 5 -t iris.arff -no-cv
```

Figure 2: Command-line call for running Auto-WEKA with a time limit of 5 minutes on training dataset `iris.arff`. Auto-WEKA performs cross-validation internally, so we disable WEKA's cross-validation (`-no-cv`). Running with `-h` lists the available options.

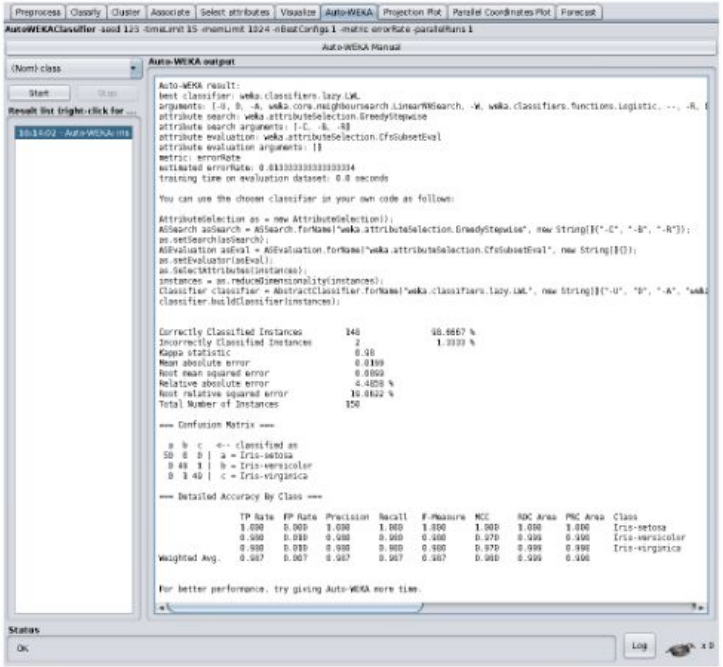


Figure 3: Example Auto-WEKA run on the iris dataset. The resulting best classifier along with its parameter settings is printed first, followed by its performance. While Auto-WEKA runs, it logs to the status bar how many configurations it has evaluated so far.

http://www.cs.ubc.ca/labs/beta/Projects/autoweka/papers/16-599.pdf

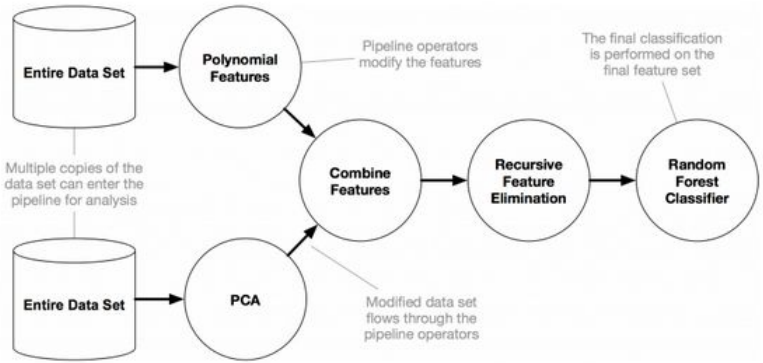
TPOT

The Tree-Based Pipeline Optimization Tool (TPOT) was one of the very first AutoML methods and open-source software packages developed for the data science community. TPOT was developed by Dr. Randal Olson while a postdoctoral student with Dr. Jason H. Moore at the Computational Genetics Laboratory of the University of Pennsylvania and is still being extended and supported by this team.

The goal of TPOT is to automate the building of ML pipelines by combining a flexible expression tree representation of pipelines with stochastic search algorithms such as genetic programming. TPOT makes use of the Python-based scikit-learn library as its ML menu.

Several peer-reviewed papers have been published on TPOT. Our first paper in 2016 won a best paper award at the EvoStar computer science conference. Our second paper in 2016 won a best paper award at the GECCO computer science conference. We showed in a 2017 paper presented at the GECCO conference how TPOT could be adapted to the analysis of big data from genetic studies of common human diseases. This paper was nominated for a best paper award. Here is our latest paper on some new operators to facilitate scaling TPOT to big data. Please contact us for reprints of these papers and others. These can also be found on arXiv.

The TPOT software is open-source, programmed in Python, and available on GitHub.



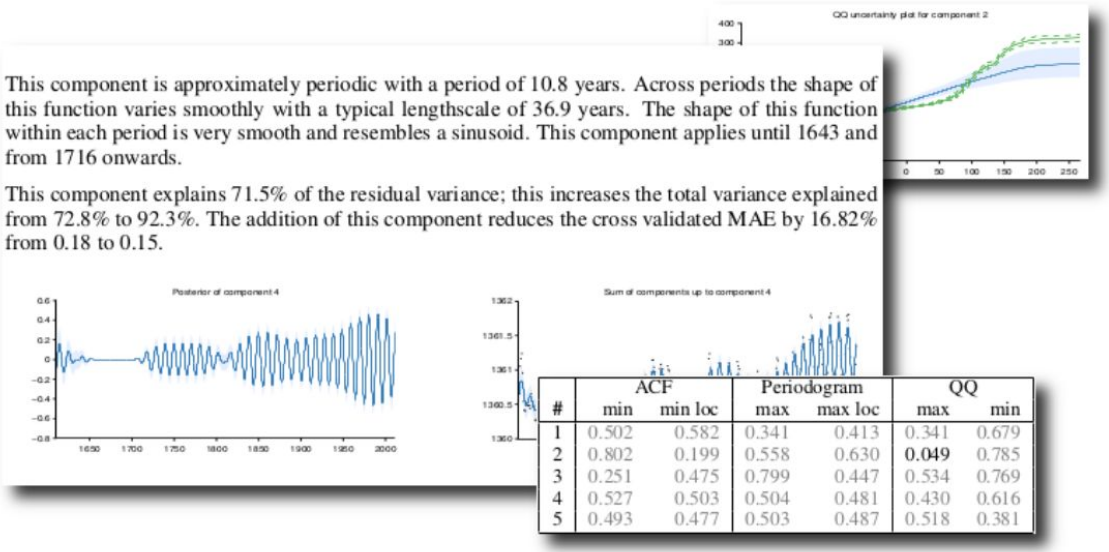
http://automl.info/tpot/

The Automatic Statistician

An artificial intelligence for data science

Example analyses

We have examples of autogenerated reports that you can look at.



#	ACF min	ACF min loc	Periodogram max	Periodogram max loc	QQ max	QQ min
1	0.502	0.582	0.341	0.413	0.341	0.679
2	0.802	0.199	0.558	0.630	0.049	0.785
3	0.251	0.475	0.799	0.447	0.534	0.769
4	0.527	0.503	0.504	0.481	0.430	0.616
5	0.493	0.477	0.503	0.487	0.518	0.381

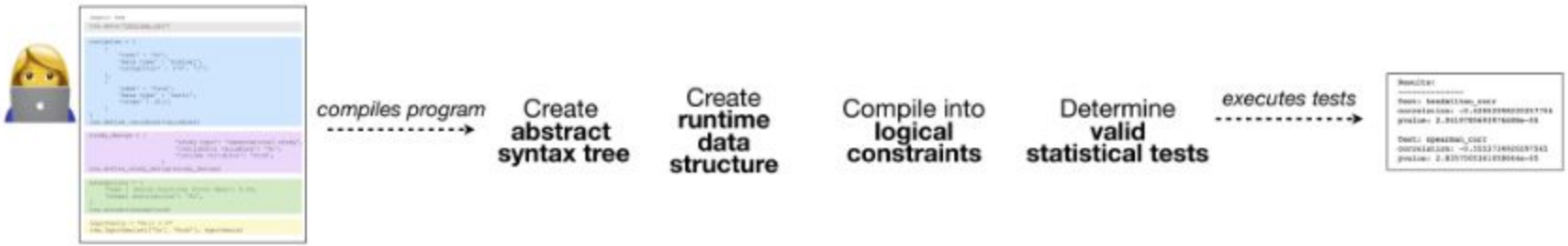
Tea: A High-level Language and Runtime System for Automating Statistical Analyses

Eunice Jun, Maureen Daum, Jared Roesch
University of Washington

Sarah E. Chasins
University of California, Berkeley

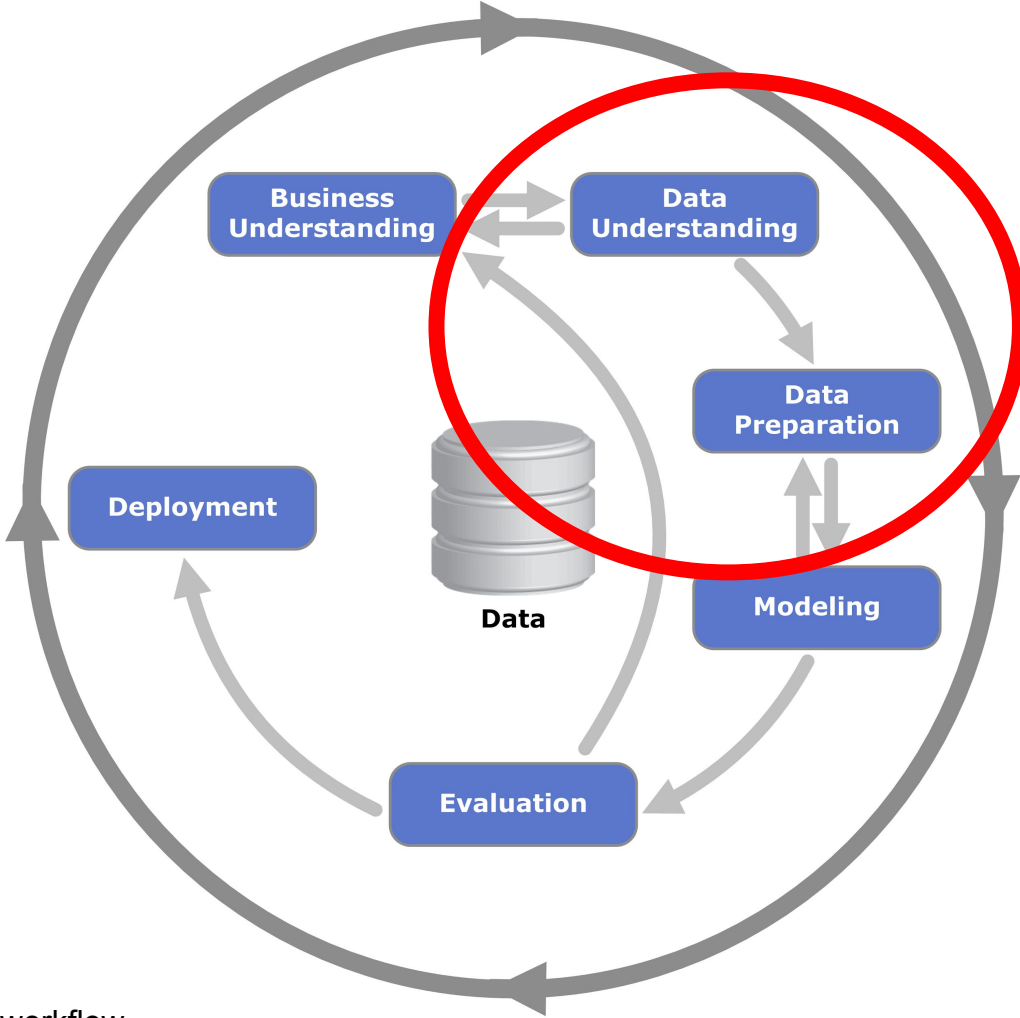
Emery D. Berger
University of Massachusetts Amherst

Rene Just, Katharina Reinecke
University of Washington



What main methodology are you using for your analytics, data mining, or data science projects ? [200 votes total] 2014 poll 2007 poll	
CRISP-DM (86)	43% 42%
My own (55)	27.5% 19%
SEMMA (17)	8.5% 13%
Other, not domain-specific (16)	8% 4%
KDD Process (15)	7.5% 7.3%
My organizations' (7)	3.5% 5.3%
A domain-specific methodology (4)	2% 4.7%
None (0)	0% 4.7%

https://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html

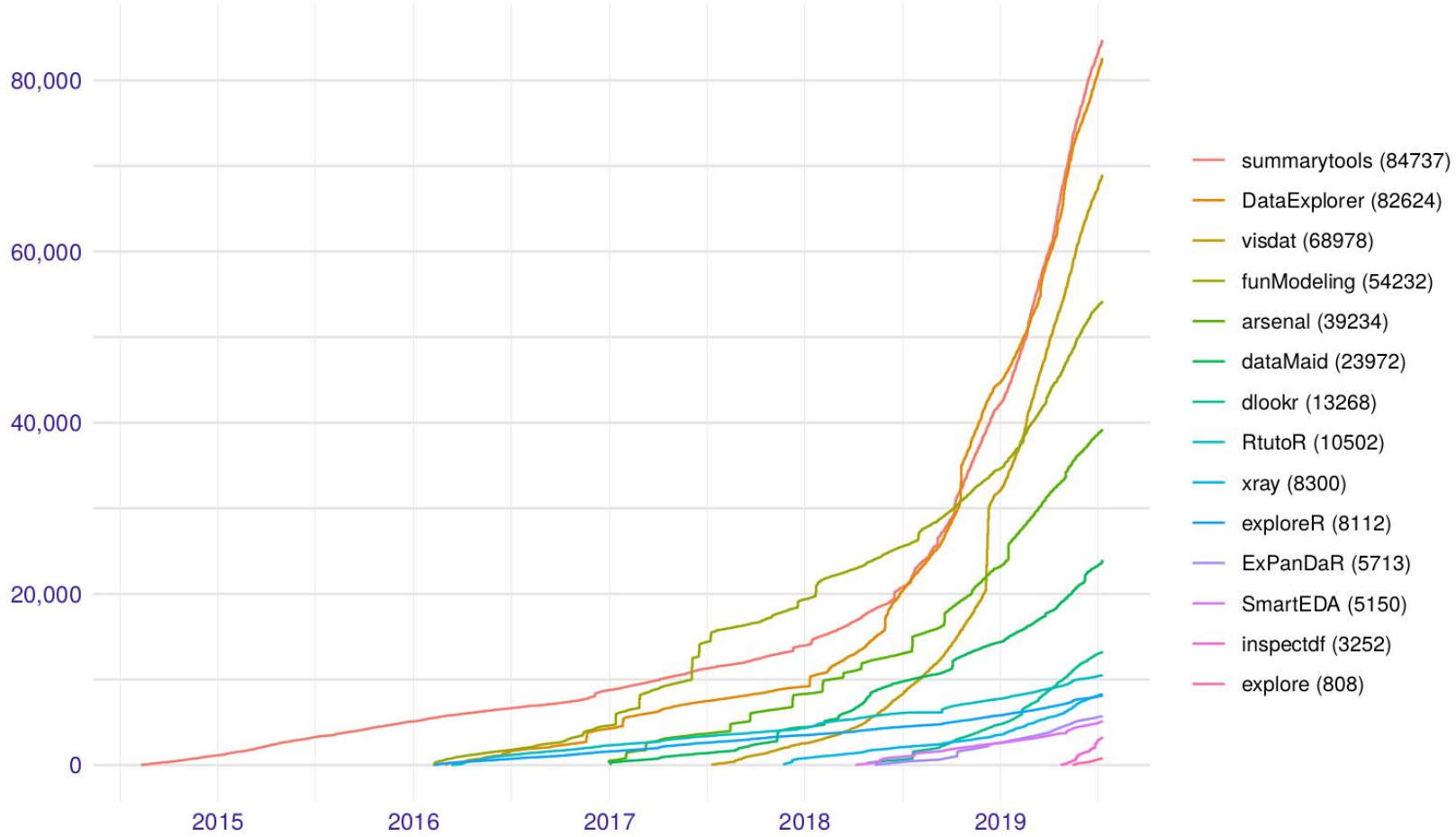


CRISP-DM workflow

https://www.datasciencecentral.com/profiles/blogs/crisp-dm-a-standard-methodology-to-ensure-a-good-outcome

Total number of downloads

Based on CRAN statistics



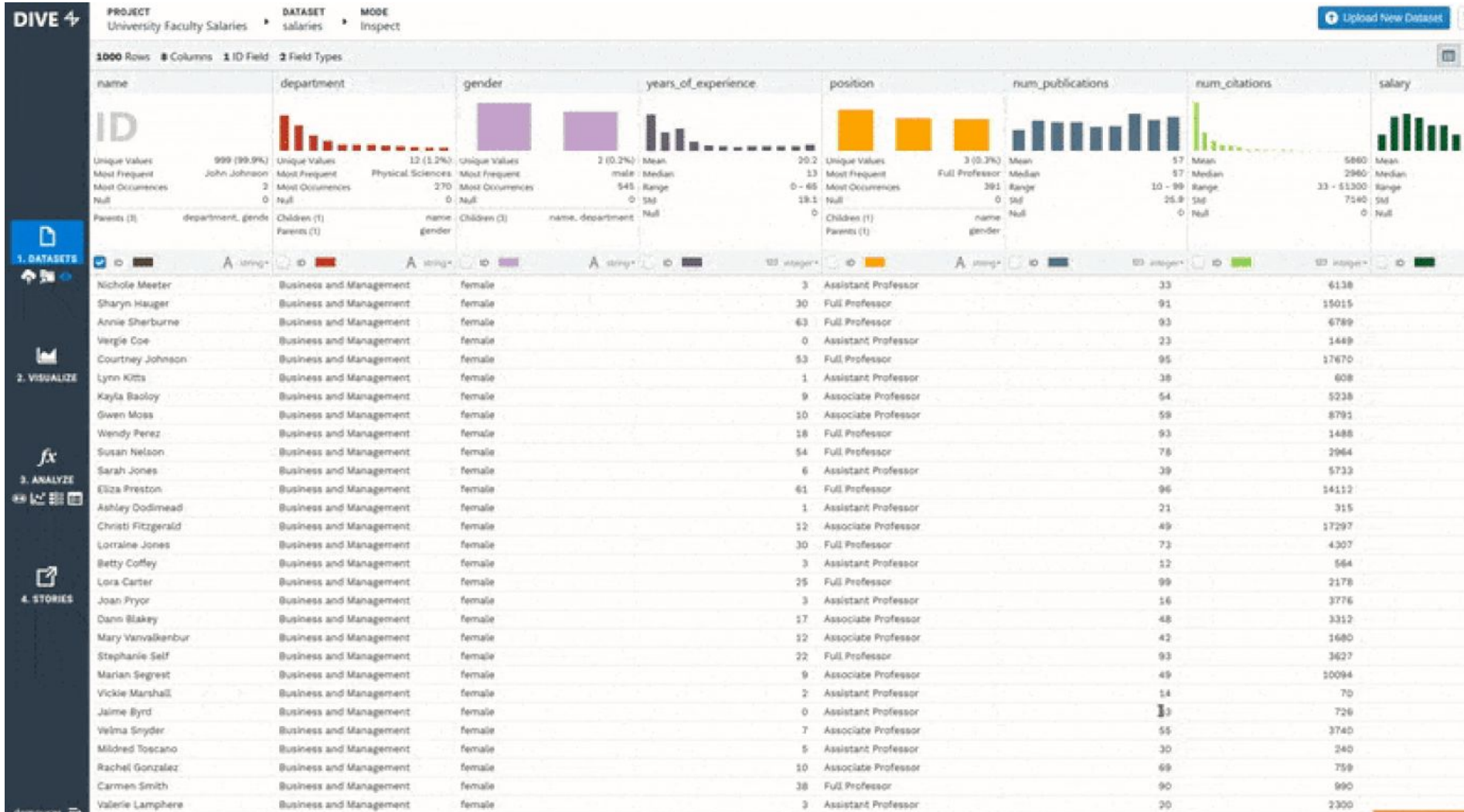
What can you expect from autoEDA? An example of DIVE

‹ Research

In this Project:

DIVE is a web-based data exploration system that lets non-technical users create stories from their data without writing code. DIVE combines semantic data ingestion, recommendation-based visualization and analysis, and dynamic story sharing into a unified workflow.

https://www.media.mit.edu/projects/dive/overview/



Nichole Meeter	Business and Management	female	3	Assistant Professor	33	6138
Sharyn Hauger	Business and Management	female	30	Full Professor	91	15015
Annie Sherburne	Business and Management	female	63	Full Professor	93	6789
Vergie Coe	Business and Management	female	0	Assistant Professor	23	1449
Courtney Johnson	Business and Management	female	53	Full Professor	95	17670
Lynn Kitts	Business and Management	female	1	Assistant Professor	38	608
Kayla Baoloy	Business and Management	female	9	Associate Professor	54	5238
Gwen Moss	Business and Management	female	10	Associate Professor	59	8791
Wendy Perez	Business and Management	female	18	Full Professor	93	1488
Susan Nelson	Business and Management	female	54	Full Professor	78	2964
Sarah Jones	Business and Management	female	6	Assistant Professor	39	5733
Eliza Preston	Business and Management	female	61	Full Professor	96	14112
Ashley Dodimead	Business and Management	female	1	Assistant Professor	21	315
Christi Fitzgerald	Business and Management	female	12	Associate Professor	49	17297
Lorraine Jones	Business and Management	female	30	Full Professor	73	4307
Betty Coffey	Business and Management	female	3	Assistant Professor	12	564
Lora Carter	Business and Management	female	25	Full Professor	99	2178
Joan Pryor	Business and Management	female	3	Assistant Professor	16	3776
Dann Blakey	Business and Management	female	17	Associate Professor	48	3312
Mary Vanvalkenbur	Business and Management	female	12	Associate Professor	42	1680
Stephanie Self	Business and Management	female	22	Full Professor	93	3627
Marian Segrest	Business and Management	female	9	Associate Professor	49	10094
Vickie Marshall	Business and Management	female	2	Assistant Professor	14	70
Jaime Byrd	Business and Management	female	0	Assistant Professor	13	726
Velma Snyder	Business and Management	female	7	Associate Professor	55	3740
Mildred Toscano	Business and Management	female	5	Assistant Professor	30	240
Rachel Gonzalez	Business and Management	female	10	Associate Professor	69	759
Carmen Smith	Business and Management	female	38	Full Professor	90	990
Valerie Lamphere	Business and Management	female	3	Assistant Professor	20	2300


DIVE
PROJECT
University Faculty Salaries
DATASET
salaries
MODE
Explore
Visualizations of salary
Individual Matches (1)
Distribution of salary
Expanded Matches (9)
salary vs. years_of_experience
salary vs. num_publications
salary vs. num_citations
Distribution of salary by department
Distribution of salary by gender
Distribution of salary by position
Sum of salary by department
Sum of salary by gender
Sum of salary by position
1. DATASETS
2. VISUALIZE
3. ANALYZE
4. STORIES
demouser
RECOMMENDATION OPTIONS
RECOMMENDATION MODE
Regular
Expanded
SORT BY
VARIABLE SELECTION
CATEGORICAL
position
gender
department
QUANTITATIVE
salary
num_citations
num_publications
years_of_experience
Give Feedback

DIVE

PROJECT University Faculty Salaries ▸ DATASET salaries ▸ MODE Regression

Save

1. DATASETS
2. VISUALIZE
3. ANALYZE
4. STORIES

demouser

Explaining salary in terms of gender and num_citations

Variables	(1)	(2)	(3)
gender: male	-9906.499*		-9119.431***
	(4115.877)		(1829.294)
num_citations		8.132***	8.128***
		(0.128)	(0.126)
Constant	159636.444***	106622.368***	111617.63***
	(3038.362)	(1192.239)	(1548.702)
R^2	0.00478	0.799	0.803
Model Properties	–	–	–

* p < 0.05 ** p < 0.01 *** p < 0.001

Summary

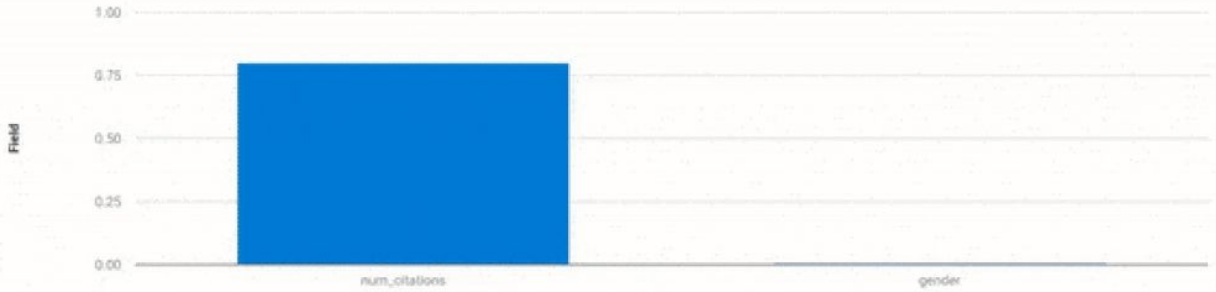
This table displays the results of a linear regression explaining the dependent variable salary with combinations of the independent variables gender, and num_citations.

For each variable, the regression coefficient is the first value, significance is represented by number of asterisks, and standard error by the number in parentheses.

The $\overline{R^2}$, the amount of variance explained by the independent variables, varies from 0.803 in equation (3) to 0.005 in equation (1).

Contribution to R^2, determined by comparing models without a variable to the full model with all variables, is highest for num_citations and lowest for variable gender.

Contribution to R^2



REGRESSION OPTIONS

Recommend Model

RECOMMENDATION TYPE: Forward Selection on R^2

TABLE LAYOUT MODE: Leave One Out

REGRESSION TYPE: Linear

VARIABLE SELECTION

DEPENDENT VARIABLE (Y): salary

EXPLANATORY FACTORS (X)

CATEGORICAL

position
gender
department

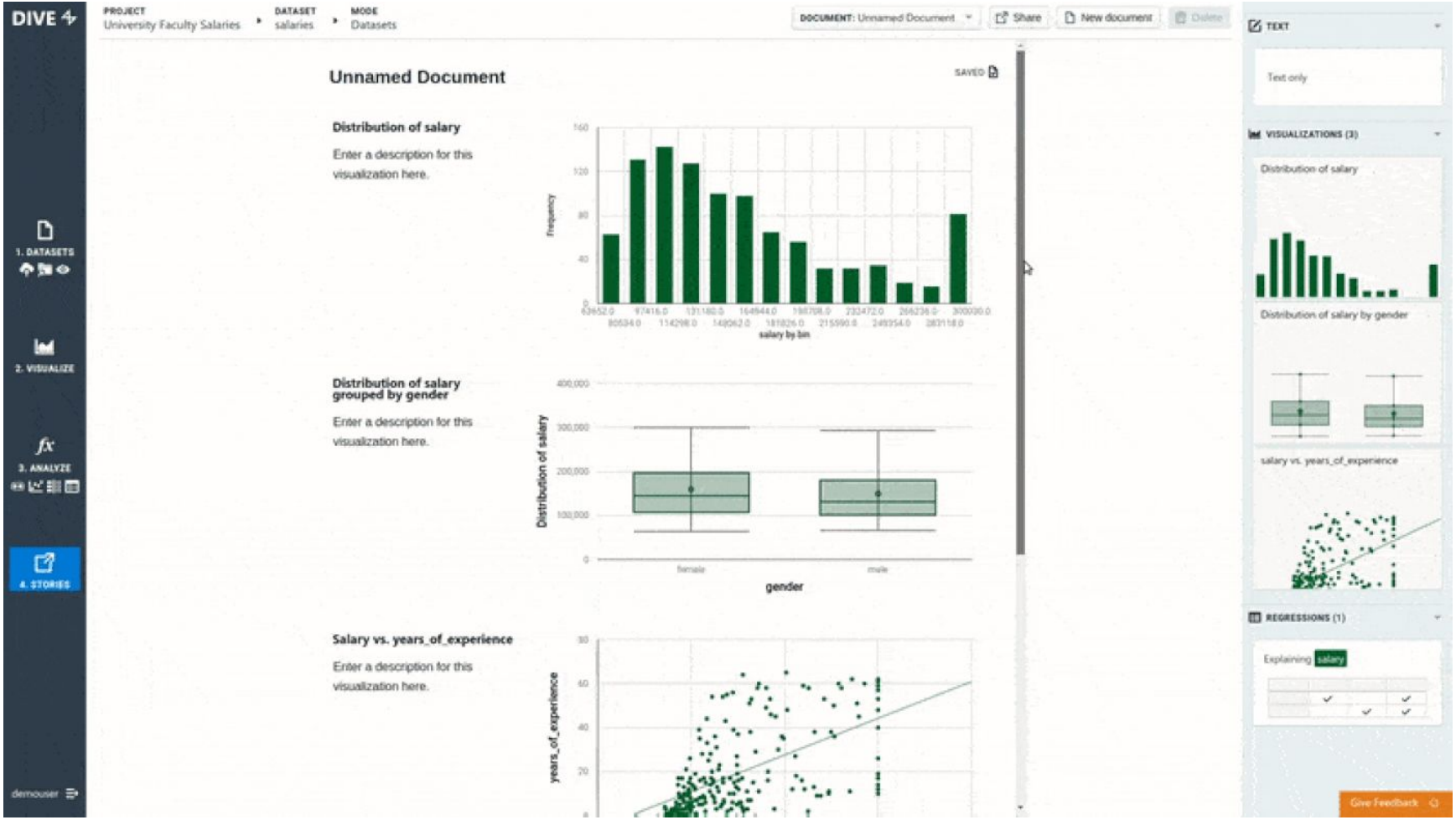
QUANTITATIVE

salary
num_citations
num_publications
years_of_experience

INTERACTION TERMS

FILTERS

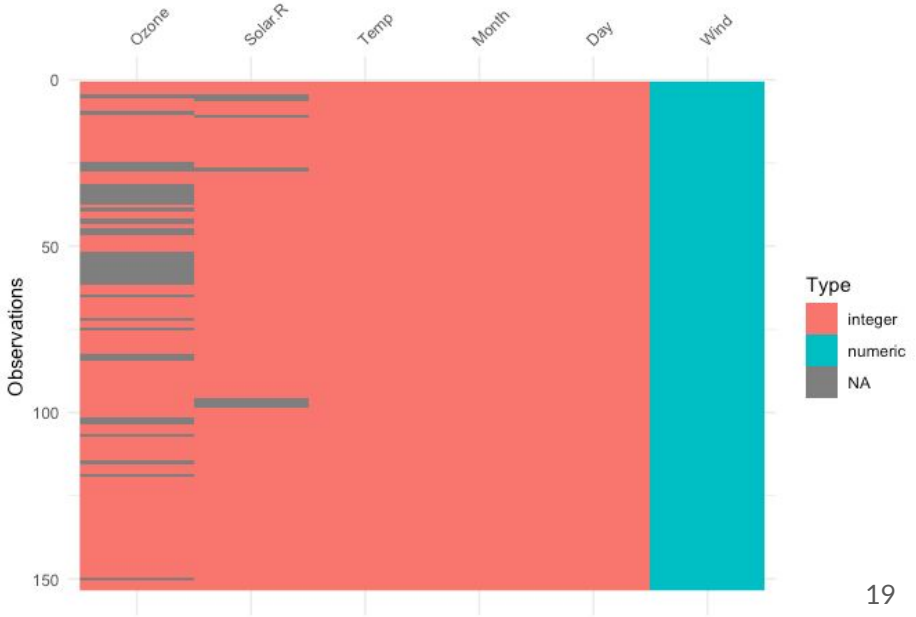
Give Feedback


DIVE
PROJECT
University Faculty Salaries
DATASET
salaries
MODE
Datasets
DOCUMENT: Unnamed Document
Share
New document
Delete
1. DATASETS
2. VISUALIZE
3. ANALYZE
4. STORIES
demouser
Unnamed Document
SAVED
Distribution of salary
Enter a description for this visualization here.
Frequency
salary by bin
Distribution of salary grouped by gender
Enter a description for this visualization here.
Distribution of salary
gender
female
male
Salary vs. years_of_experience
Enter a description for this visualization here.
years_of_experience
TEXT
Text only
VISUALIZATIONS (3)
Distribution of salary
Distribution of salary by gender
salary vs. years_of_experience
REGRESSIONS (1)
Explaining
salary
Give Feedback

What can you expect from autoEDA packages?

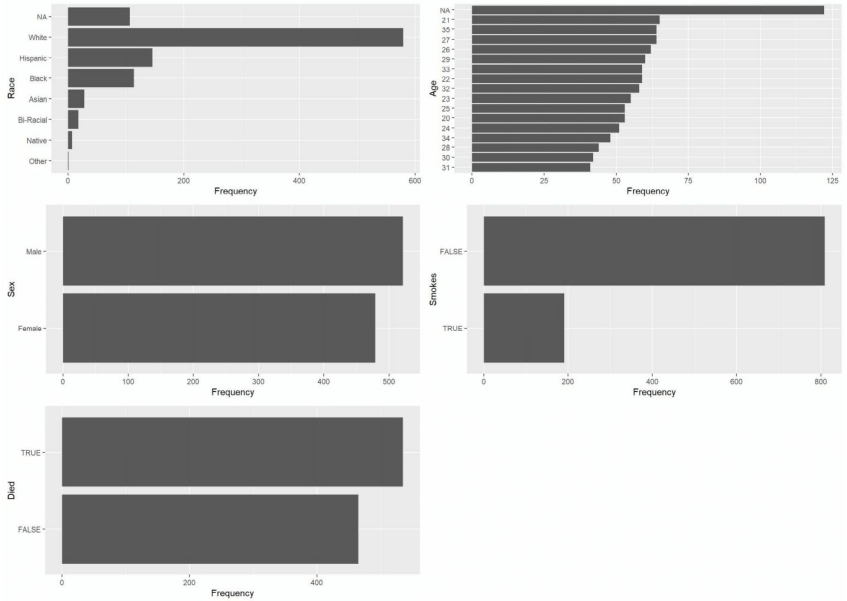
Whole data summaries

- Missing value count
- Quantiles
- Scale & location
- Variable types
- Dataset size
- Datasets comparisons



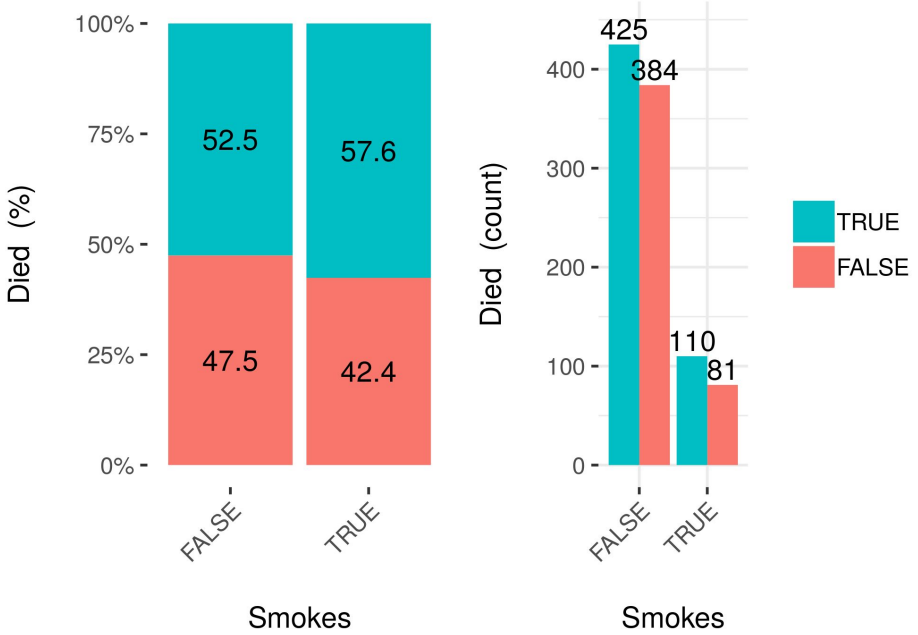
Univariate distributions

- Descriptive statistics
- Histograms
- Bar plots
- QQ plots



Bivariate relationships

- Scatter plots
- Grouped barplots
- Boxplots
- Contingency tables
- Correlation matrix



More: transformations & multivariate analysis

Transforming variables:

- skewness
- outliers
- arbirtrary transformations

Multivariate tools:

- regression models
- PCA
- visualization (Parallel Coordinate Plots)

Reporting & interactivity

Typical approach:

All functions are run automatically and outputs are saved to pdf

Interactive approach:

The package assists in exploration - user chooses variables to explore interactively

Top packages to look into

Part 1

Data report overview

The dataset examined has the following dimensions:

Feature	Result
Number of observations	1000
Number of variables	9

Checks performed

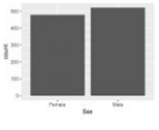
The following variable checks were performed, depending on the data type of each variable:

	character	factor	labelled	haven labelled	numeric	integer	logical	Date
Identify miscoded missing values	×	×	×	×	×	×		×
Identify prefixed and suffixed whitespace	×	×	×	×				
Identify levels with < 6 obs.	×	×	×	×				
Identify case issues	×	×	×	×				
Identify misclassified numeric or integer variables	×	×	×	×				
Identify outliers					×	×		×

Please note that all numerical values in the following have been rounded to 2 decimals.

Sex

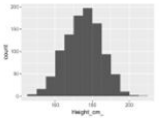
Feature	Result
Variable type	factor
Number of missing obs.	0 (0 %)
Number of unique values	2
Mode	"Male"
Reference category	Male



Height_cm_

Feature	Result
Variable type	numeric
Number of missing obs.	0 (0 %)
Number of unique values	365
Median	175.3
1st and 3rd quartiles	168.2; 182.03
Min. and max.	146.3; 207.2

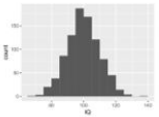
- Note that the following possible outlier values were detected: "201.1", "207.2".



IQ

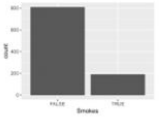
Feature	Result
Variable type	numeric
Number of missing obs.	102 (10.2 %)
Number of unique values	57
Median	100
1st and 3rd quartiles	93; 107
Min. and max.	68; 137

- Note that the following possible outlier values were detected: "68", "129", "137".



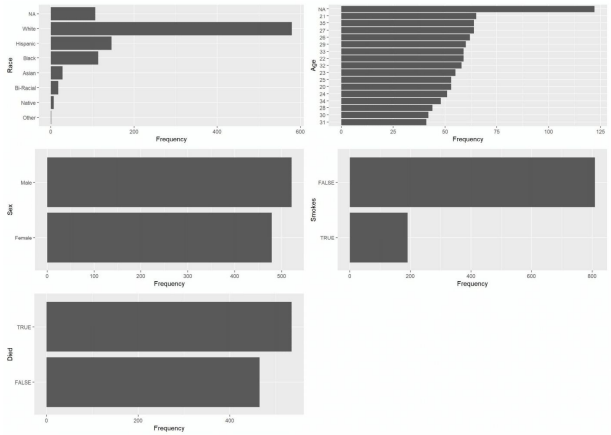
Smokes

Feature	Result
Variable type	logical
Number of missing obs.	0 (0 %)
Number of unique values	2
Mode	"FALSE"

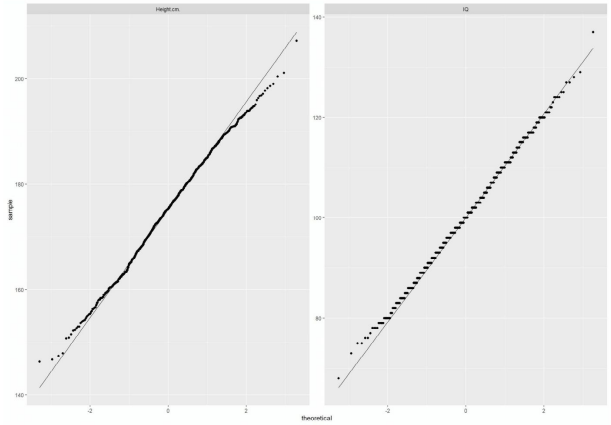


dataMaid

Example report:
https://bit.ly/2mMAlQJ



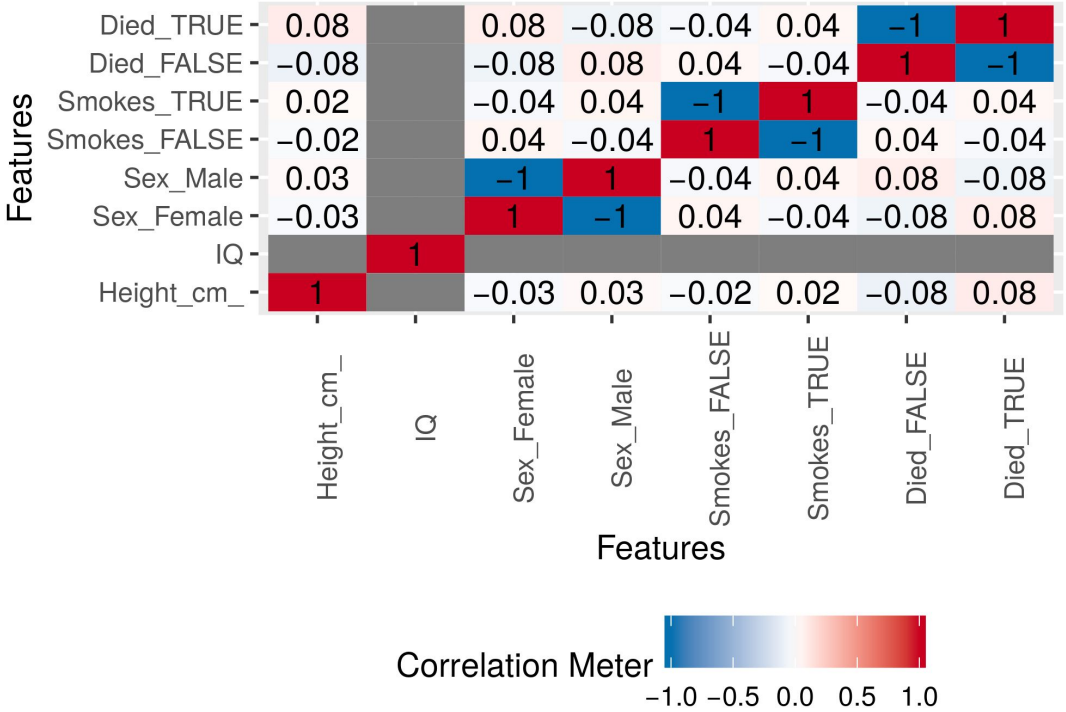
QQ Plot



QQ Plot (by Died)

DataExplorer

Example report: https://bit.ly/2mJLiT4



Exploratory Data Analysis Report

2019-03-15

Exploratory Data analysis (EDA)

Analyzing the data sets to summarize their main characteristics of variables, often with visual graphs, without using a statistical model.

1. Overview of the data

Understanding the dimensions of the dataset, variable names, overall missing summary and data types of each variables

```
# Overiview of the data
ExpData(data=data,type=1)
# Structure of the data
ExpData(data=data,type=2)
```

Overview of the data

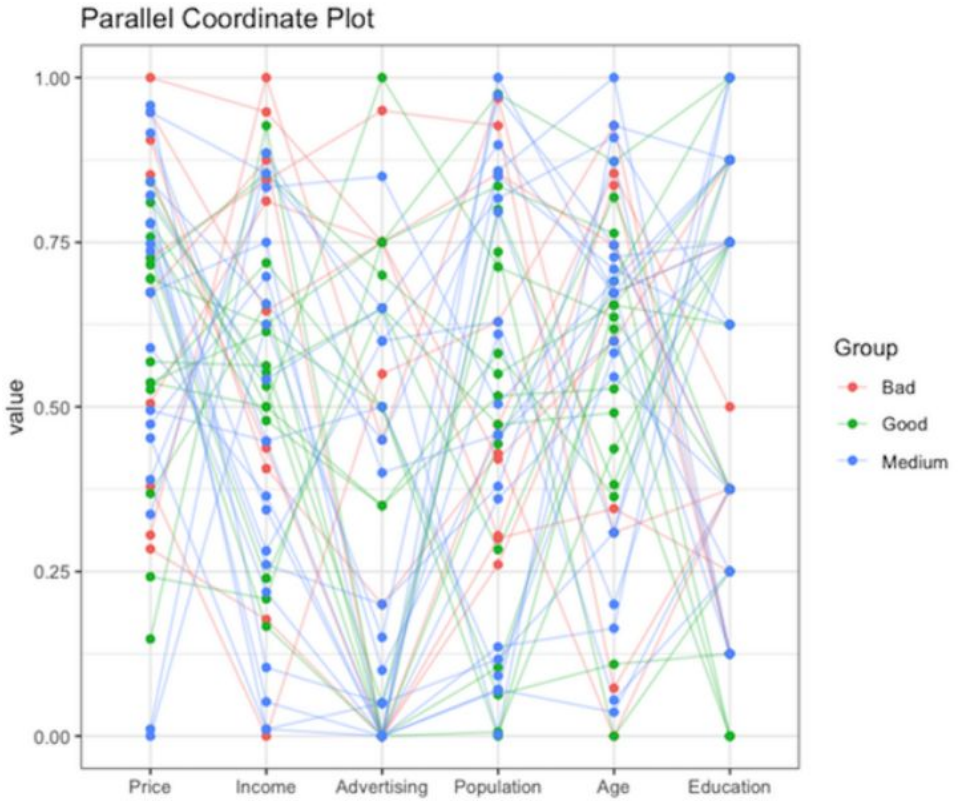
Descriptions <fctr>	Obs <fctr>
Sample size (Nrow)	1000
No. of Variables (Ncol)	9
No. of Numeric Variables	2
No. of Factor Variables	3
No. of Text Variables	2
No. of Logical Variables	2
No. of Date Variables	0
No. of Zero variance Variables (Uniform)	0
%. of Variables having complete cases	55.56% (5)
%. of Variables having <50% missing cases	44.44% (4)

1-10 of 12 rows Previous 1 2 Next

Structure of the data

SmartEDA

Example report: https://bit.ly/2mxvQJY



IQ

normality test : Shapiro-Wilk normality test
statistic : 0.99821, p-value : 0.47445

type	skewness	kurtosis
original	0.0753	2.9651
log transformation	-0.2225	3.0516
sqrt transformation	-0.0725	2.9698

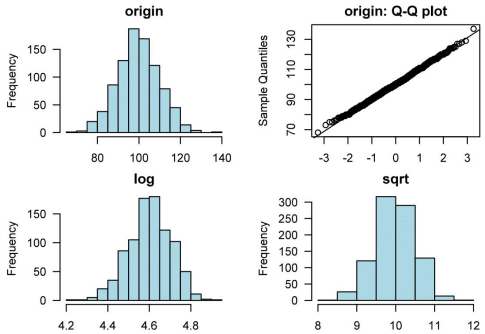


Figure 2.2: IQ

dlookr

Example report: https://bit.ly/2mKcFfZ

Chapter 1

Introduction

The EDA Report provides exploratory data analysis information on objects that inherit data.frame and data.frame.

1.1 Information of Dataset

The dataset that generated the EDA Report is an 'data.frame' object. It consists of 1,000 observations and 9 variables.

1.2 Information of Variables

Table 1.1: Information of Variables

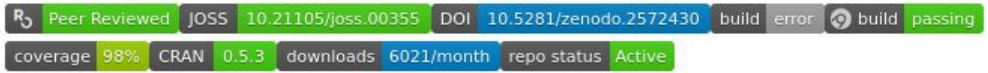
variables	types	missing_count	missing_percent	unique_count	unique_rate
ID	character	0	0.0	1000	1.000
Race	factor	107	10.7	8	0.008
Age	character	122	12.2	17	0.017
Sex	factor	0	0.0	2	0.002
Height(cm)	numeric	0	0.0	365	0.365
IQ	numeric	102	10.2	58	0.058
Smokes	logical	0	0.0	2	0.002
Income	factor	100	10.0	901	0.901
Died	logical	0	0.0	2	0.002

The target variable of the data is 'Died', and the data type of the variable is logical.

1.3 About EDA Report

EDA reports provide information and visualization results that support the EDA process. In particular, it provides a variety of information to understand the relationship between the target variable and the rest of the variables of interest.

visdat



How to install

visdat is available on CRAN

```
install.packages("visdat")
```

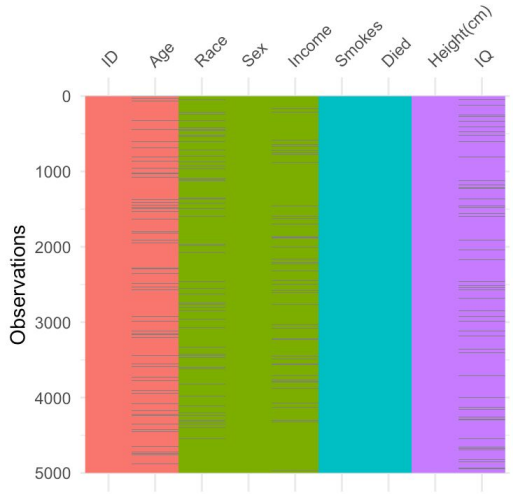
If you would like to use the development version, install from github with:

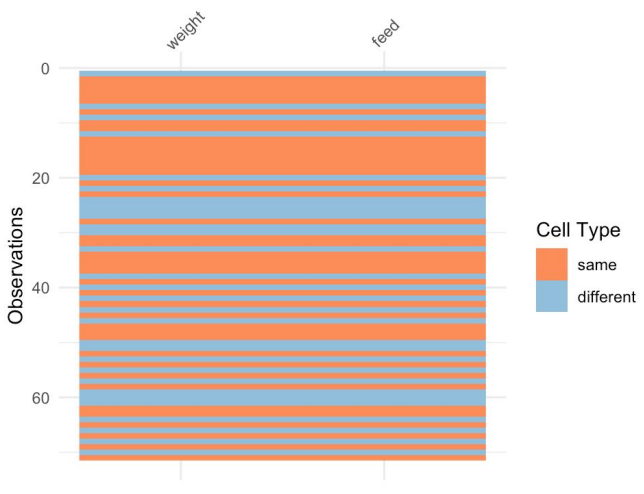
```
# install.packages("devtools")
devtools::install_github("ropensci/visdat")
```

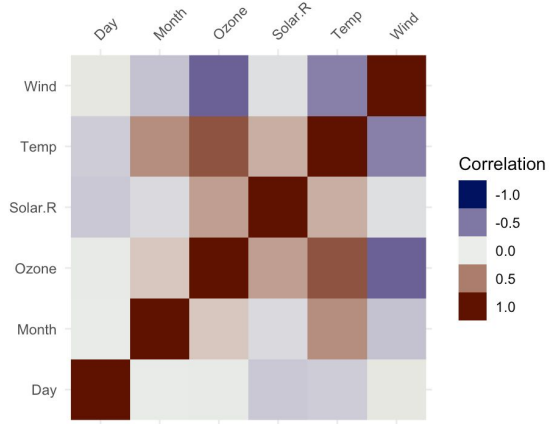
What does visdat do?

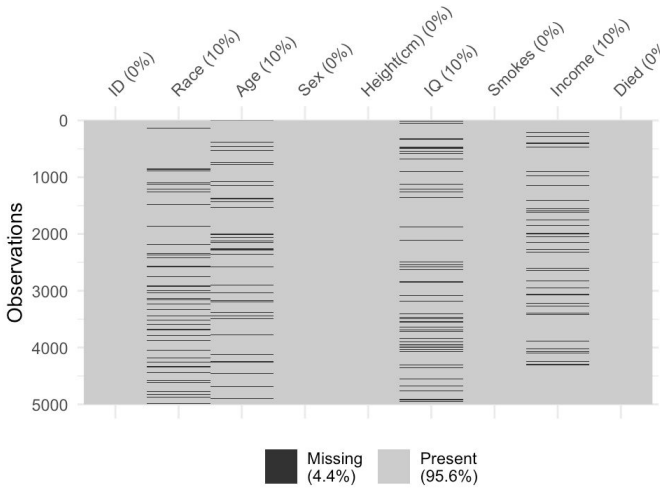
Initially inspired by `csv-fingerprint`, `vis_dat` helps you visualise a dataframe and "get a look at the data" by displaying the variable classes in a dataframe as a plot with `vis_dat`, and getting a brief look into missing data patterns using `vis_miss`.

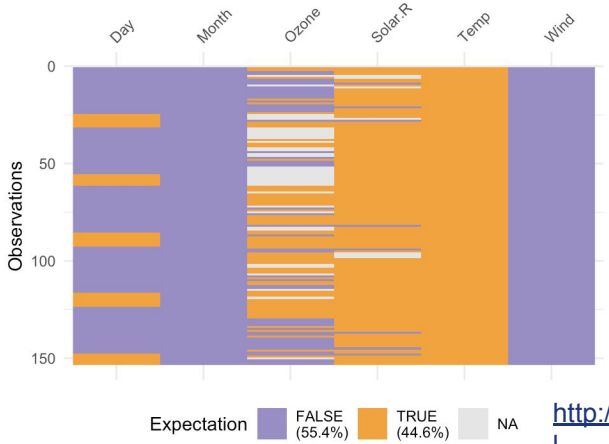
`visdat` has 6 functions:

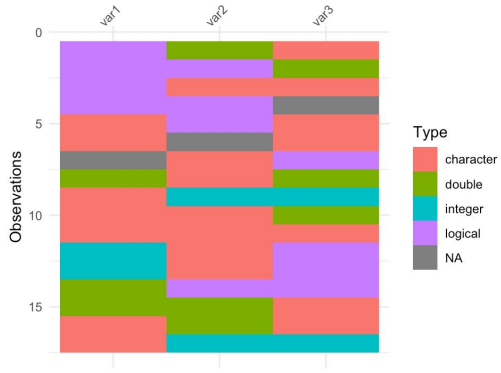












http://visdat.njtierney.com/articles/using_visdat.html

dataMaid

dataMaid is an R package for documenting and creating reports on data cleanliness.

Installation

This github page contains the *development version* of dataMaid. For the latest stable version download the package from CRAN directly using

```
install.packages("dataMaid")
```

To install the development version of dataMaid run the following commands from within R (requires that the `devtools` package is already installed)

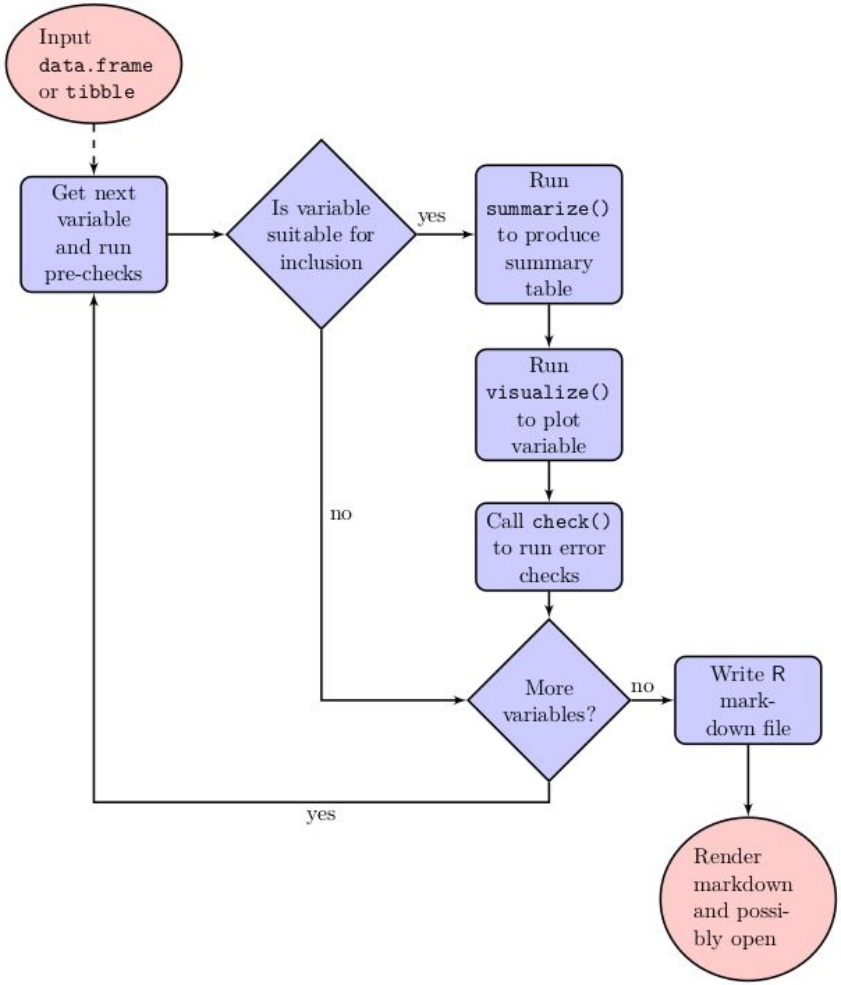
```
devtools::install_github("ekstroem/dataMaid")
```

Package overview

A super simple way to get started is to load the package and use the `makeDataReport()` function on a data frame (if you try to generate several reports for the same data, then it may be necessary to add the `replace=TRUE` argument to overwrite the existing report).

```
library("dataMaid")
data(trees)
makeDataReport(trees)
```

https://www.jstatsoft.org/article/view/v090i06



`cleaner` : Fast and Easy Data Cleaning

(Previously called `clean`, but renamed to `cleaner` upon CRAN request)

Website of this package: https://msberends.github.io/cleaner

CRAN 1.2.0

The R package for **cleaning and checking data columns** in a fast and easy way. Relying on very few dependencies, it provides **smart guessing**, but with user options to override anything if needed.

It also provides two new data types that are not available in base R: `currency` and `percentage`.

Contents:

Honorable mentions

Group factor

None

Select a factor for subsetting specific analyses to.

Outlier treatment

- (•) No treatment
- ( ) Winsorization 1%/99%
- ( ) Winsorization 5%/95%
- ( ) Truncation 1%/99%
- ( ) Truncation 5%/95%

By group factor

None

Indicate whether you want no outlier treatment or whether you want outliers to be winsorized to the given percentile or truncated if they exceed the given percentile. Give a by group if you want outlier treatment to be done independently by group.

Bar Chart

Select factor to display

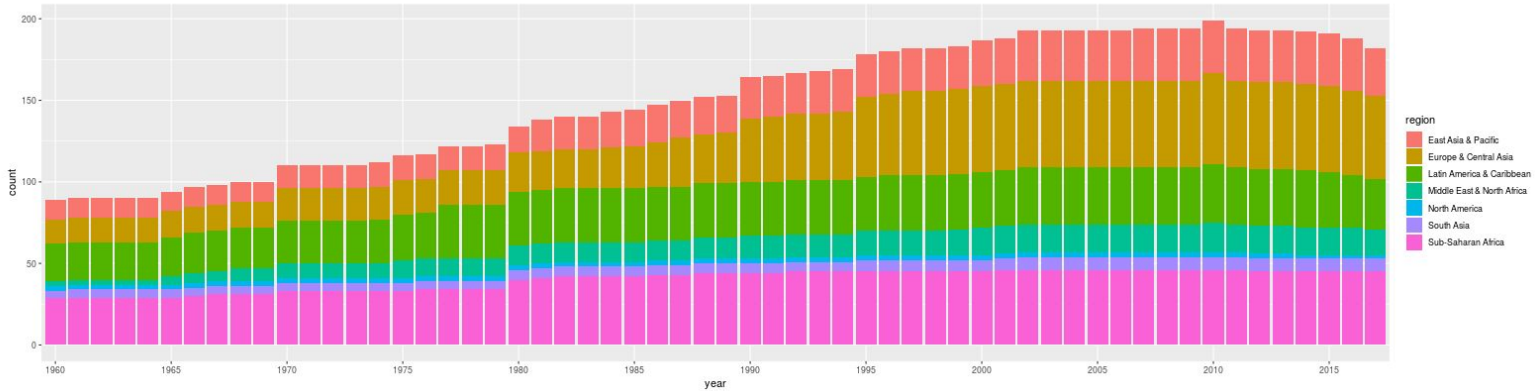
year

Select additional factor to display

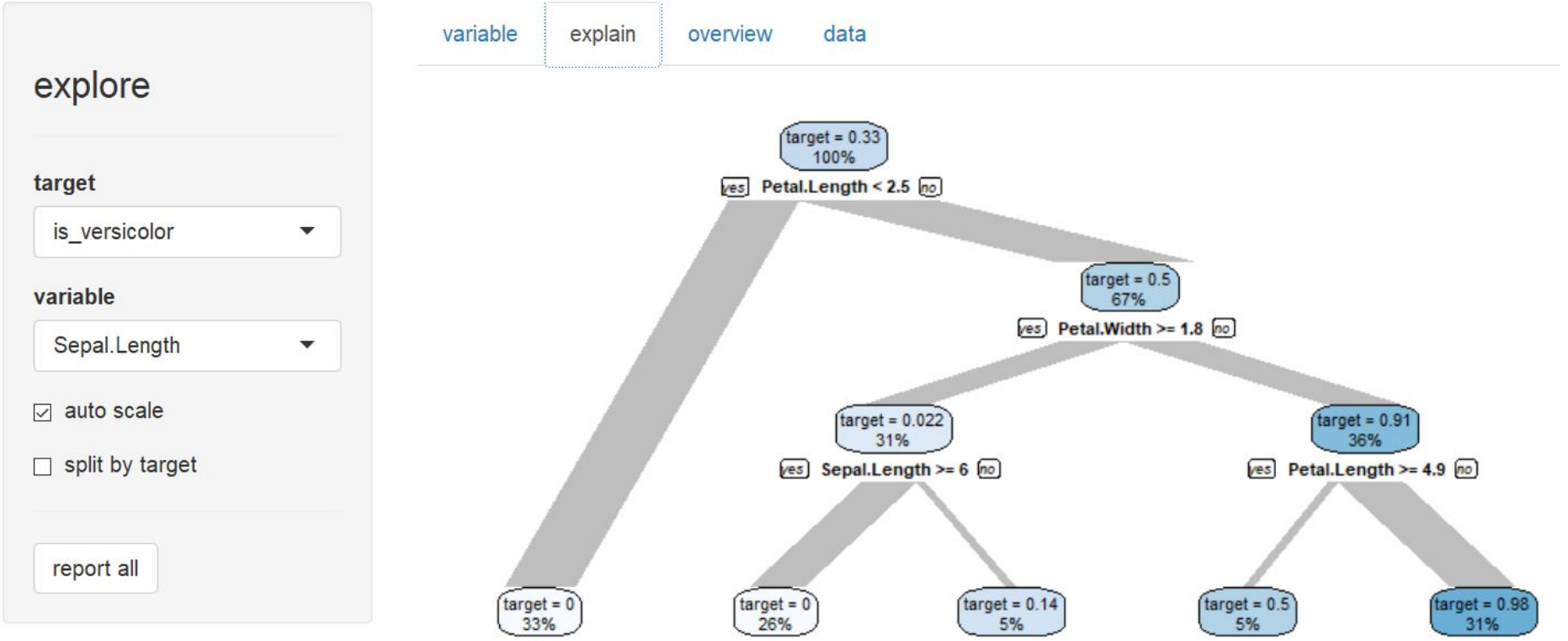
region

[ ] Relative display

Check if you want to see the additional factor relative to the first factor.



ExPanDaR


explore
target
is_versicolor
variable
Sepal.Length
auto scale
split by target
report all
variable
explain
overview
data
target = 0.33
100%
yes
Petal.Length < 2.5
no
target = 0.5
67%
yes
Petal.Width >= 1.8
no
target = 0.022
31%
yes
Sepal.Length >= 6
no
target = 0.91
36%
yes
Petal.Length >= 4.9
no
target = 0
33%
target = 0
26%
target = 0.14
5%
target = 0.5
5%
target = 0.98
31%

explore

The idea is: you can take a `data.frame` of messy real world data and easily, faithfully, reliably, and repeatably prepare it for machine learning using documented methods using `vtreat`. Incorporating `vtreat` into your machine learning workflow lets you quickly work with very diverse structured data.

In all cases (classification, regression, unsupervised, and multinomial classification) the intent is that `vtreat` transforms are essentially one liners.

The preparation commands are organized as follows:

- **Regression**: `R` regression example, `Python` regression example.
- **Classification**: `R` classification example, `Python` classification example.
- **Unsupervised tasks**: `R` unsupervised example, `Python` unsupervised example.
- **Multinomial classification**: `R` multinomial classification example, `Python` multinomial classification example.

In all cases: variable preperation is intended to be a "one liner."

These current revisions of the examples are designed to be small, yet complete. So as a set they have some overlap, but the user can rely mostly on a single example for a single task type.

For more detail please see here: arXiv:1611.09477 stat.AP (the documentation describes the `R` version, however all of the examples can be found worked in `Python` here).

`vtreat` is available as an `R` package, and also as a `Python` / `Pandas` package.

Build a transform appropriate for regression problems.

Now that we have the data, we want to treat it prior to modeling: we want training data where all the input variables are numeric and have no missing values or `NaN`s.

First create the data treatment transform object, in this case a treatment for a regression problem.

```
transform_design = vtreat::mkCrossFrameNExperiment(
    dframe = d,                              # data to learn transform from
    varlist = setdiff(colnames(d), c('y')),  # columns to transform
    outcomename = 'y'                        # outcome variable
)
```

```
## [1] "vtreat 1.5.1 start initial treatment design Tue Jan 14 09:53:06 2020"
## [1] " start cross frame work Tue Jan 14 09:53:07 2020"
## [1] " vtreat::mkCrossFrameNExperiment done Tue Jan 14 09:53:07 2020"
```

```
transform <- transform_design$treatments
d_prepared <- transform_design$crossFrame
score_frame <- transform$scoreFrame
score_frame$recommended <- score_frame$varMoves & (score_frame$sig < 1/nrow(score_frame))
```

Note that for the training data `d`: `transform_design$crossFrame` is **not** the same as `prepare(transform, d)`; the second call can lead to nested model bias in some situations, and is **not** recommended. For other, later data, not seen during transform design `transform.preprare(o)` is an appropriate step.

`vtreat` version `1.5.1` and newer issue a warning if you call the incorrect transform pattern on your original training data:

```
d_prepared_wrong <- prepare(transform, d)
```

```
## Warning in prepare.treatmentplan(transform, d): possibly called prepare() on
## same data frame as designTreatments*()/mkCrossFrame*Experiment(), this can lead
## to over-fit. To avoid this, please use mkCrossFrame*Experiment$crossFrame.
```

The prepared data should be clean: completely numeric, with no missing values.

```r
# Table report for a linear model
lm(Sepal.Length ~ Petal.Length + Species, data=iris) %>%
  report() %>%
  to_table()
```

```
## Parameter         | Coefficient |         95% CI |      p | Coefficient (std.) |  Fit
## ------------------------------------------------------------------------------------
## (Intercept)       |        3.68 |   [3.47, 3.89] | < .001 |               1.50 |
## Petal.Length      |        0.90 |   [0.78, 1.03] | < .001 |               1.93 |
## Speciesversicolor |       -1.60 | [-1.98, -1.22] | < .001 |              -1.93 |
## Speciesvirginica  |       -2.12 | [-2.66, -1.58] | < .001 |              -2.56 |
##                   |             |                |        |                    |
## R2                |             |                |        |                    | 0.84
## R2 (adj.)         |             |                |        |                    | 0.83
```

Finally, you can also find more details using `to_fulltext()`:

```r
# Full report for a Bayesian logistic mixed model with effect sizes
library(rstanarm)

stan_glmer(vs ~ mpg + (1|cyl), data=mtcars, family="binomial") %>%
  report(standardize="full", effsize="cohen1988") %>%
  to_fulltext()
```

```
## We fitted a Bayesian logistic mixed model (estimated using MCMC sampling with 4 chains of 2000
## iterations and a warmup of 1000) to predict vs with mpg (formula = vs ~ mpg). The model included
## cyl as random effects (formula = ~1 | cyl). Priors over parameters were set as normal (mean = 0.6
## SD = 0.41) distributions. The Region of Practical Equivalence (ROPE) percentage was defined as th
## proportion of the posterior distribution within the [-0.18, 0.18] range. The 89% Credible Interva
## (CIs) were based on Highest Density Intervals (HDI). Parameters were scaled by the mean and the S
## of the response variable. Effect sizes were labelled following Cohen's (1988) recommendations.
##
## The model's explanatory power is substantial (R2's median = 0.57, 89% CI [0.42, 0.69] Within this
## model, the explanatory power related to the fixed effects alone (marginal R2's median) is of 0.27
## (89% CI [0.00, 0.48]). The model's intercept, corresponding to vs = 0, mpg = 0 and cyl = 0, is at
## -5.16 (89% CI [-12.09, 2.10], 1.55% in ROPE, std. median = 0.00). Within this model:
##
##   - The effect of mpg has a probability of 86.00% of being positive and can be considered as medi
## and not significant (median = 0.23, 89% CI [-0.11, 0.56], 38.10% in ROPE, std. median = 1.41). Th
## algorithm successfuly converged (Rhat = 1.001) and the estimates can be considered as stable (ESS
## 1276).
```

https://easystats.github.io/report

Bonus: Python libraries

build passing | codecov 88% | release v2.4.0 | python 3.5 | 3.6 | 3.7 | code style black

Generates profile reports from a pandas `DataFrame`. The pandas `df.describe()` function is great but a little basic for serious exploratory data analysis. `pandas_profiling` extends the pandas DataFrame with `df.profile_report()` for quick data analysis.

For each column the following statistics - if relevant for the column type - are presented in an interactive HTML report:

- **Type inference**: detect the types of columns in a dataframe.
- **Essentials**: type, unique values, missing values
- **Quantile statistics** like minimum value, Q1, median, Q3, maximum, range, interquartile range
- **Descriptive statistics** like mean, mode, standard deviation, sum, median absolute deviation, coefficient of variation, kurtosis, skewness
- **Most frequent values**
- **Histogram**
- **Correlations** highlighting of highly correlated variables, Spearman, Pearson and Kendall matrices
- **Missing values** matrix, count, heatmap and dendrogram of missing values
- **Text analysis** learn about categories (Uppercase, Space), scripts (Latin, Cyrillic) and blocks (ASCII) of text data.

lens

DOI 10.5281/zenodo.2593336

`lens` is a library for exploring data in Pandas DataFrames. It computes single column summary statistics and estimates the correlation between columns. We wrote `lens` when we realised that the initial steps of acquiring a new data set were almost formulaic: What data type is in this column? How many null values are there? Which columns are correlated? What's the distribution of this value? `lens` calculates all this for you.

See the documentation for more details.

Installation

`lens` can be installed from PyPI with `pip`:

```
pip install lens
```

Dora

Exploratory data analysis toolkit for Python.

Contents

Summary

Dora is a Python library designed to automate the painful parts of exploratory data analysis.

The library contains convenience functions for data cleaning, feature selection & extraction, visualization, partitioning data for model validation, and versioning transformations of data.

The library uses and is intended to be a helpful addition to common Python data analysis tools such as pandas, scikit-learn, and matplotlib.

Data Versioning

```
# save a version of your data
>>> dora.data
   A   B  C      D  useless_feature
0  1   2  0   left               1
1  4 NaN  1  right               1
2  7   8  2   left               1
>>> dora.snapshot('initial_data')

# keep track of changes to data
>>> dora.remove_feature('useless_feature')
>>> dora.extract_ordinal_feature('D')
>>> dora.impute_missing_values()
>>> dora.scale_input_values()
>>> dora.data
   A         B         C    D=left   D=right
0  1 -1.224745 -1.224745  0.707107 -0.707107
1  4  0.000000  0.000000 -1.414214  1.414214
2  7  1.224745  1.224745  0.707107 -0.707107

>>> dora.logs
["self.remove_feature('useless_feature')", "self.extract_ordinal_feature('D')", 'self.impute_missing_value

# use a previous version of the data
>>> dora.snapshot('transform1')
>>> dora.use_snapshot('initial_data')
>>> dora.data
   A   B  C      D  useless_feature
0  1   2  0   left               1
1  4 NaN  1  right               1
2  7   8  2   left               1
>>> dora.logs
[]

# switch back to your transformation
>>> dora.use_snapshot('transform1')
>>> dora.data
   A         B         C    D=left   D=right
0  1 -1.224745 -1.224745  0.707107 -0.707107
1  4  0.000000  0.000000 -1.414214  1.414214
2  7  1.224745  1.224745  0.707107 -0.707107
>>> dora.logs
["self.remove_feature('useless_feature')", "self.extract_ordinal_feature('D')", 'self.impute_missing_value
```

More

- https://journal.r-project.org/archive/2019/RJ-2019-033/
- https://www.researchgate.net/publication/332014513_The_Landscape_of_R_Packages_for_Automated_Exploratory_Data_Analysis
- https://github.com/mstaniak/autoEDA-resources
- https://mstaniak.github.io

Contact:

- github.com/mstaniak
- mtst@mstaniak.pl