Australian National University

COMP3430 / COMP8430
Data wrangling

Lecture 9: Data pre-processing using Rattle and Python

(Lecturer: Thilina Ranbaduge)

Lecture outline

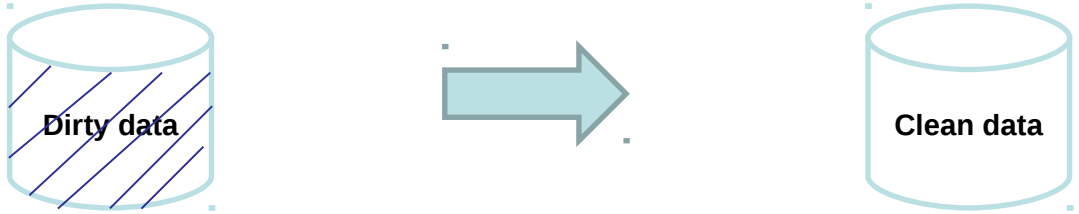
- Data pre-processing revisited
- Data pre-processing tools

- Data pre-processing using Rattle
- Data pre-processing using Python

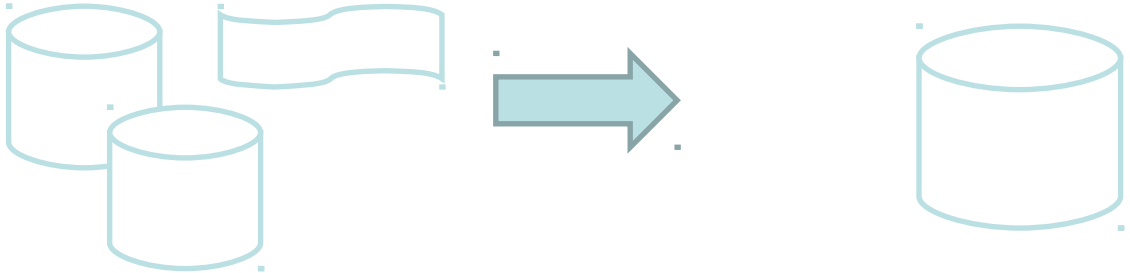
- Summary

Data pre-processing revisited

Data cleaning



Data integration



Data transformation

-1	27	100	57	63

-0.01	0.27	1.0	0.57	0.63

Data reduction

A1	A2	…..	A126
R1			
R2			
…… R800			

A1	A2	…..	A100
R1			
R2			
…… R100			

Data pre-processing tools

- Various tools available:
  - **OpenRefine** - Open source Google code project for working with messy data (http://openrefine.org/)
  - **Drake** – Open source text-based data workflow tool where steps are defined along with their inputs and outputs (https://github.com/Factual/drake)
  - **Data cleaner** – Profiling, duplicate detection, and cleansing commercial software (http://datacleaner.org/)
  - **WinPure cleaning tool** – powerful commercial tool (http://www.winpure.com/article-datacleaningtool.html)
  - **Rattle** – Open source, built on R for cleaning data
  - **Python** and **Pandas** – Open source, allows efficient data cleaning

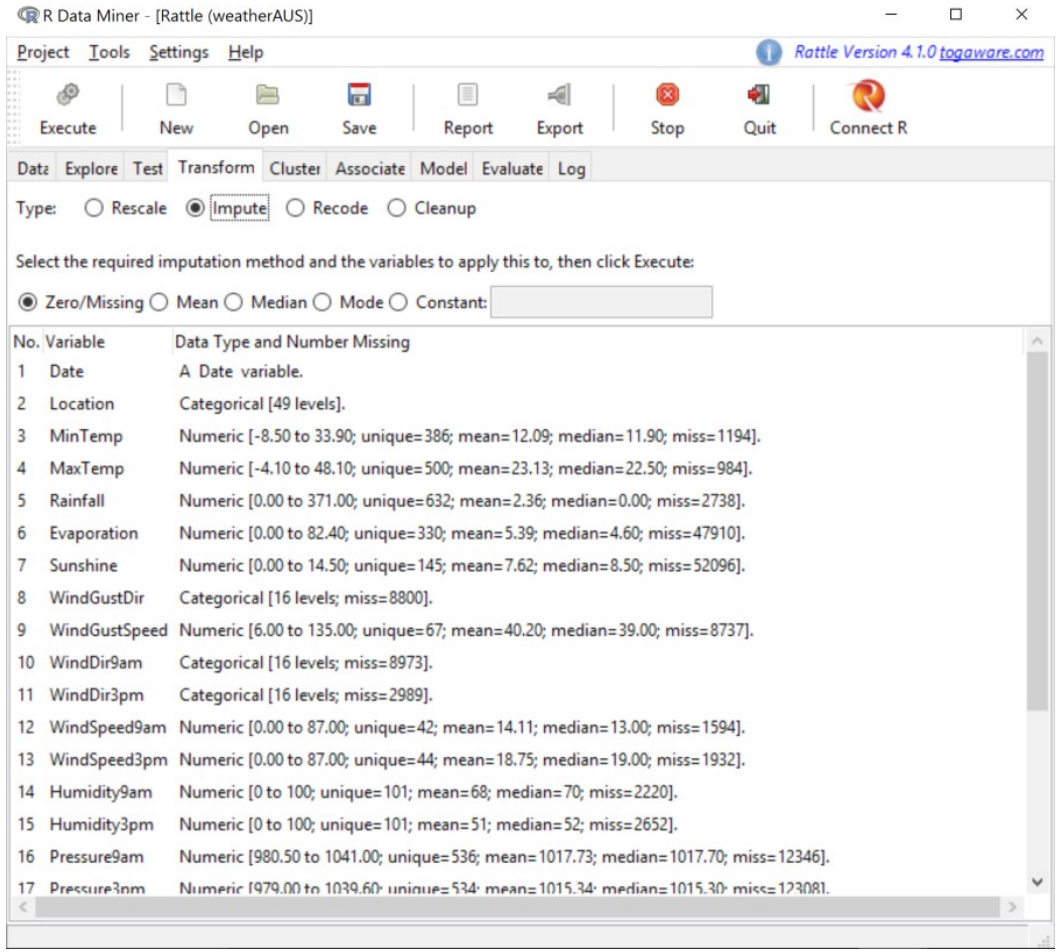
Data pre-processing with Rattle

- R is a powerful language for performing data wrangling, analysis and mining
- Rattle provides a GUI for such tasks
- The typical workflow is:
  - Loading dataset
  - Exploring dataset
  - **Transforming and cleaning dataset**
  - Building models
  - Evaluating models
  - Exporting models for deployment

The last three steps are related to data mining and will be covered in COMP3425 or COMP8410

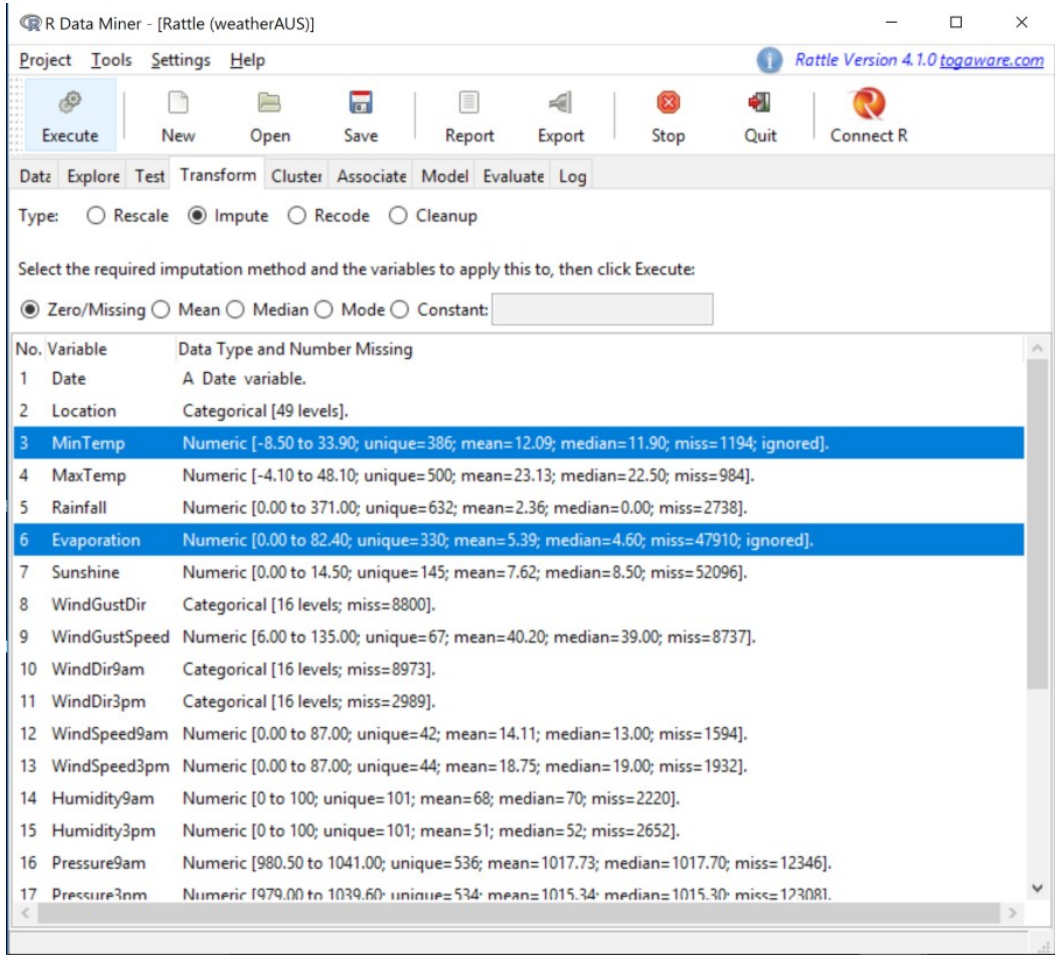
Handling missing values in Rattle (1)

- Load Rattle weather dataset
- Transform tab -> Impute
- Several options:
  - Zero/Missing
  - Mean
  - Median
  - Mode
  - Constant value



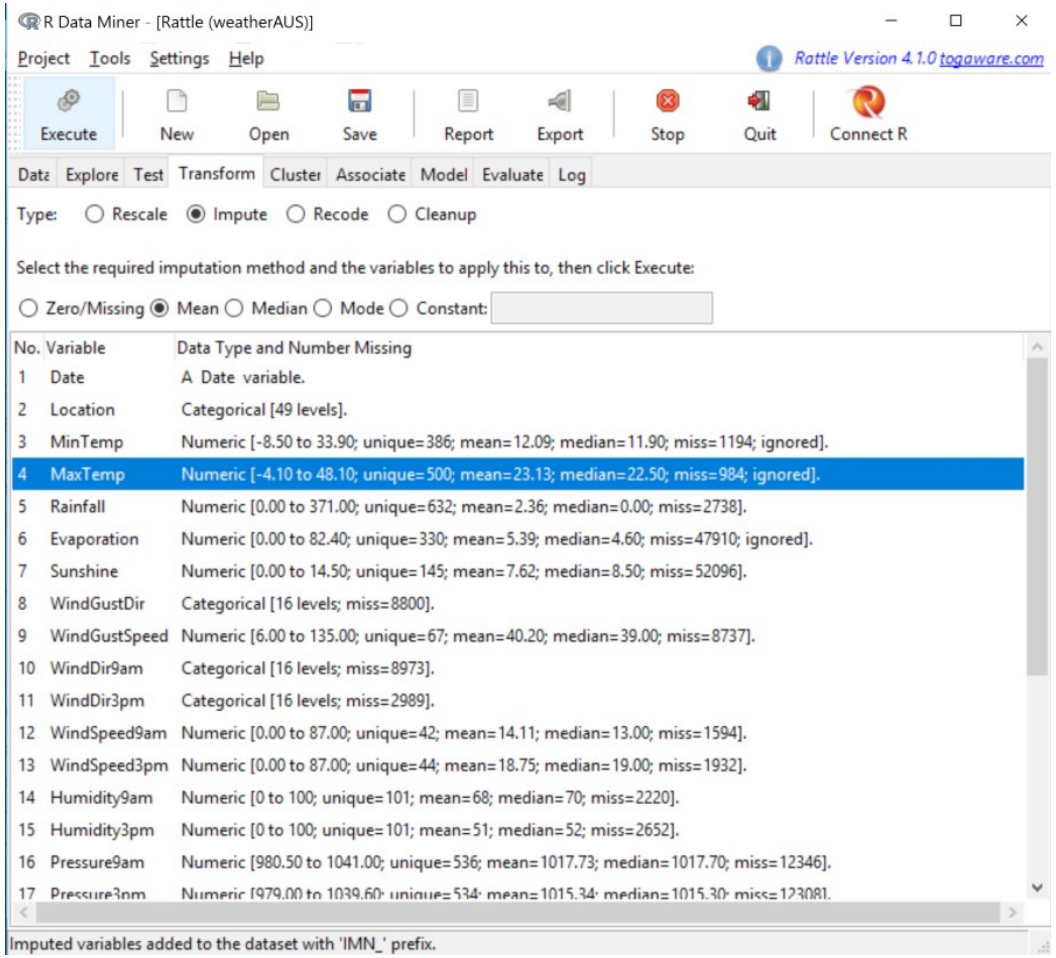
Handling missing values in Rattle (2)

- Zero/Missing value imputation
  - The simplest imputation
  - Replaces all missing values with a single value
  - Numerical variable – 0
  - Categorical variable – ‘Missing’



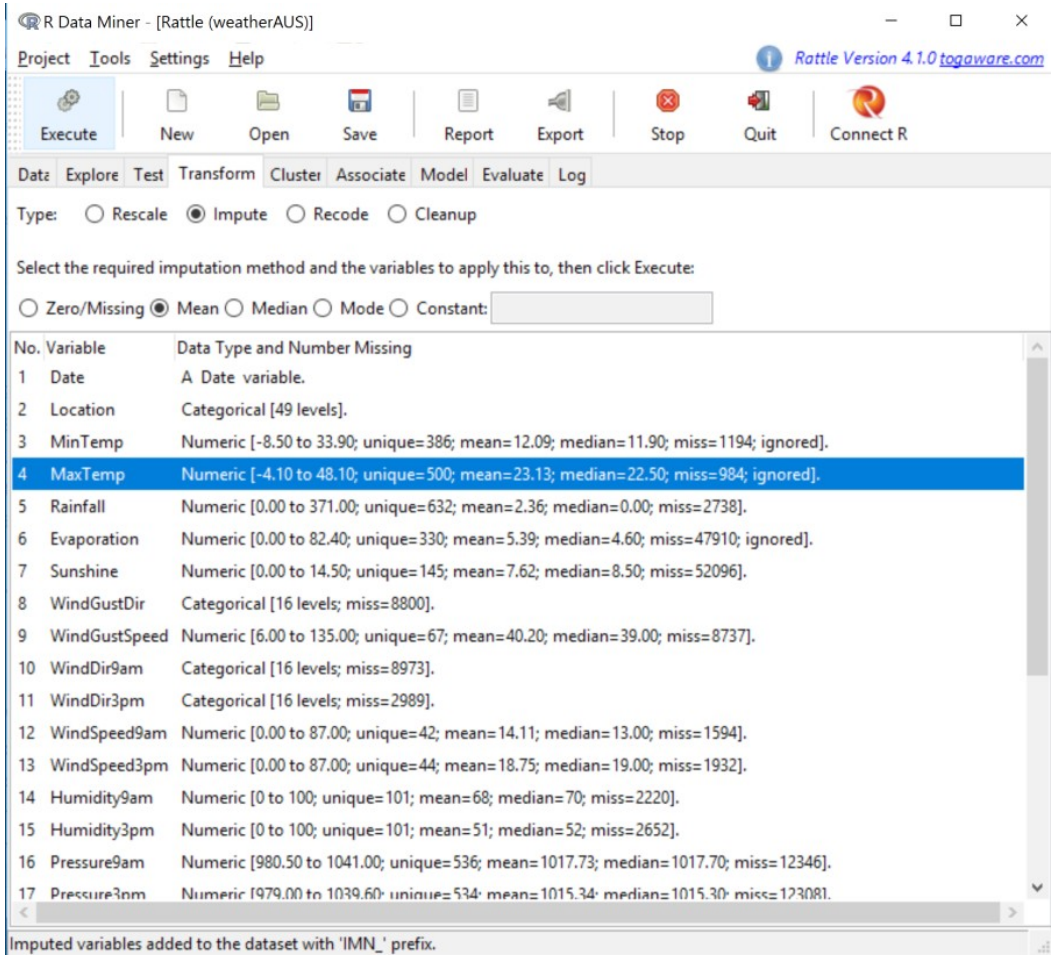
Handling missing values in Rattle (3)

- Mean / median / mode value imputation
  - Use some ‘central’ value of the variable
  - Numerical variable with normal distribution – Mean
  - Numerical variable with skewed distribution – Median
  - Categorical variable - Mode



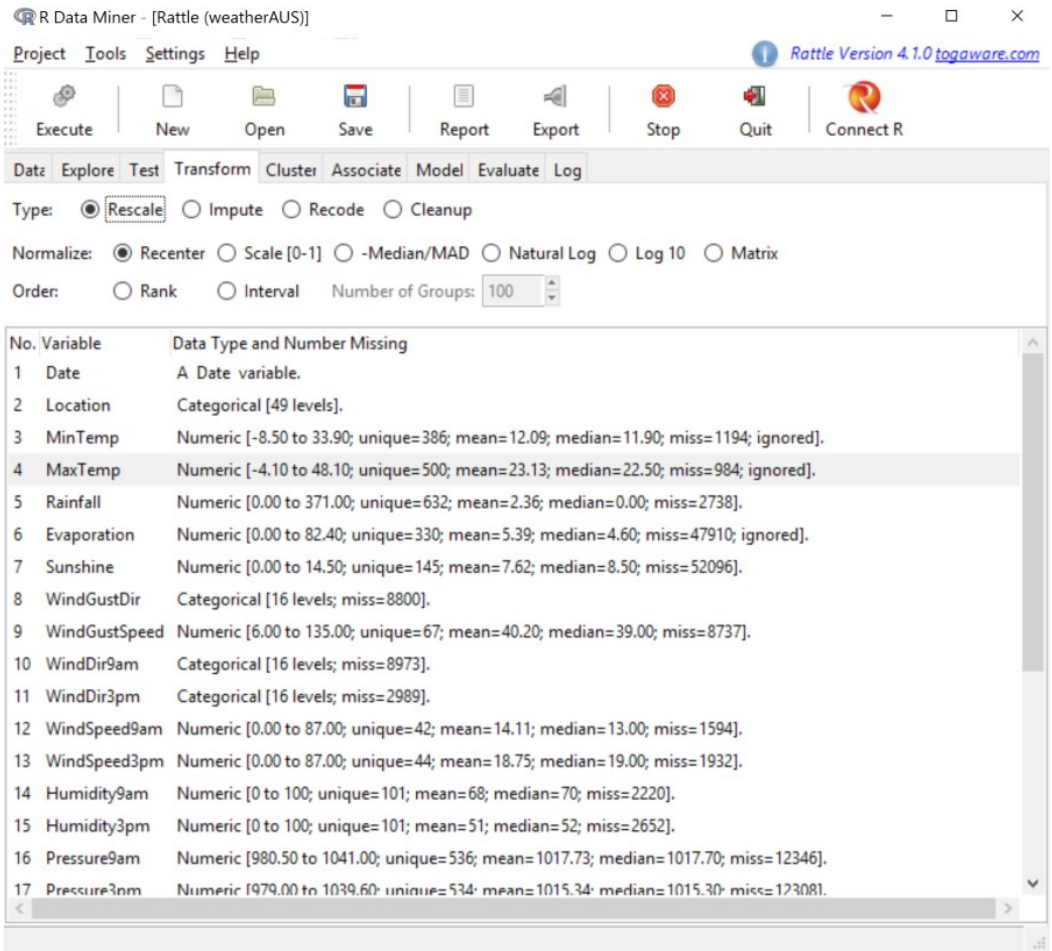
Handling missing values in Rattle (4)

- Allows using a constant value for imputation
  - Define own default value to be imputed
  - Integer/real number for numerical variable
  - Special marker for categorical variable



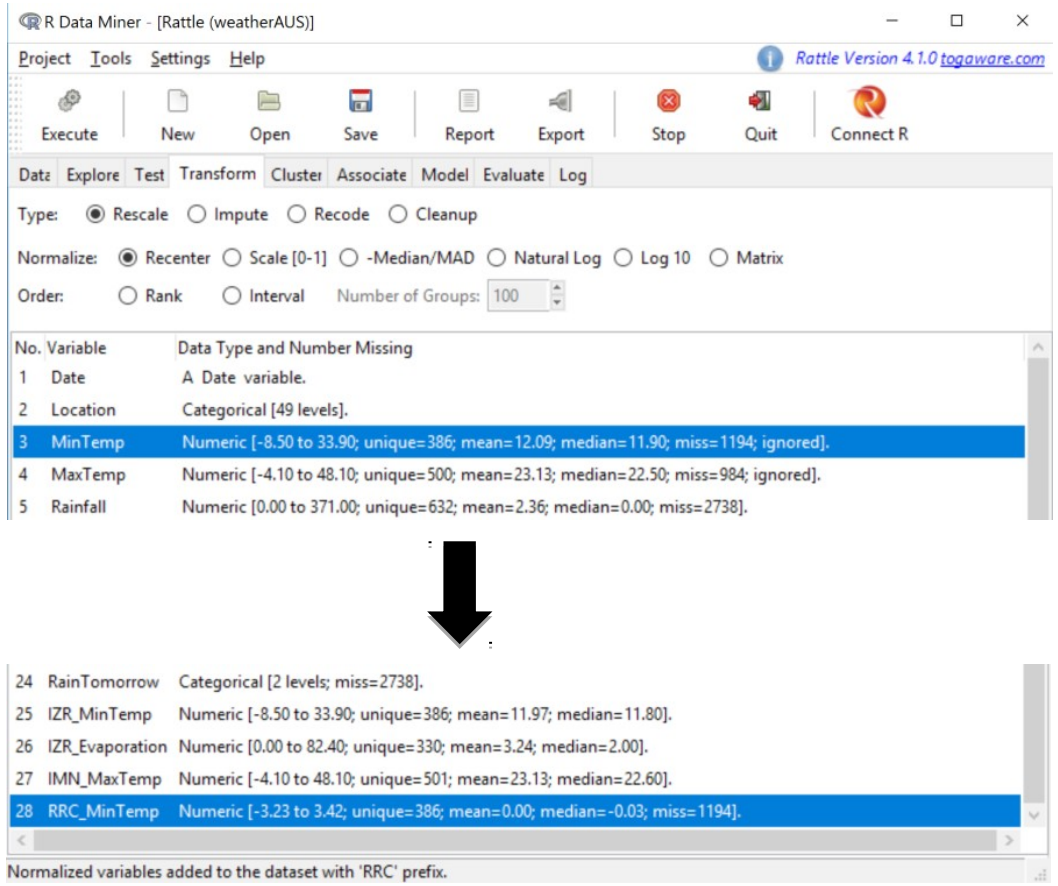
Data transformation in Rattle (1)

- Transform tab -> Rescale
  - Recentering to be around 0
  - Rescaling to be in [0-1]
  - Robust rescaling around zero using the median
  - Applying logarithm
  - Multiple variables with one divisor (matrix)
  - Ranking
  - Rescaling by group (interval)



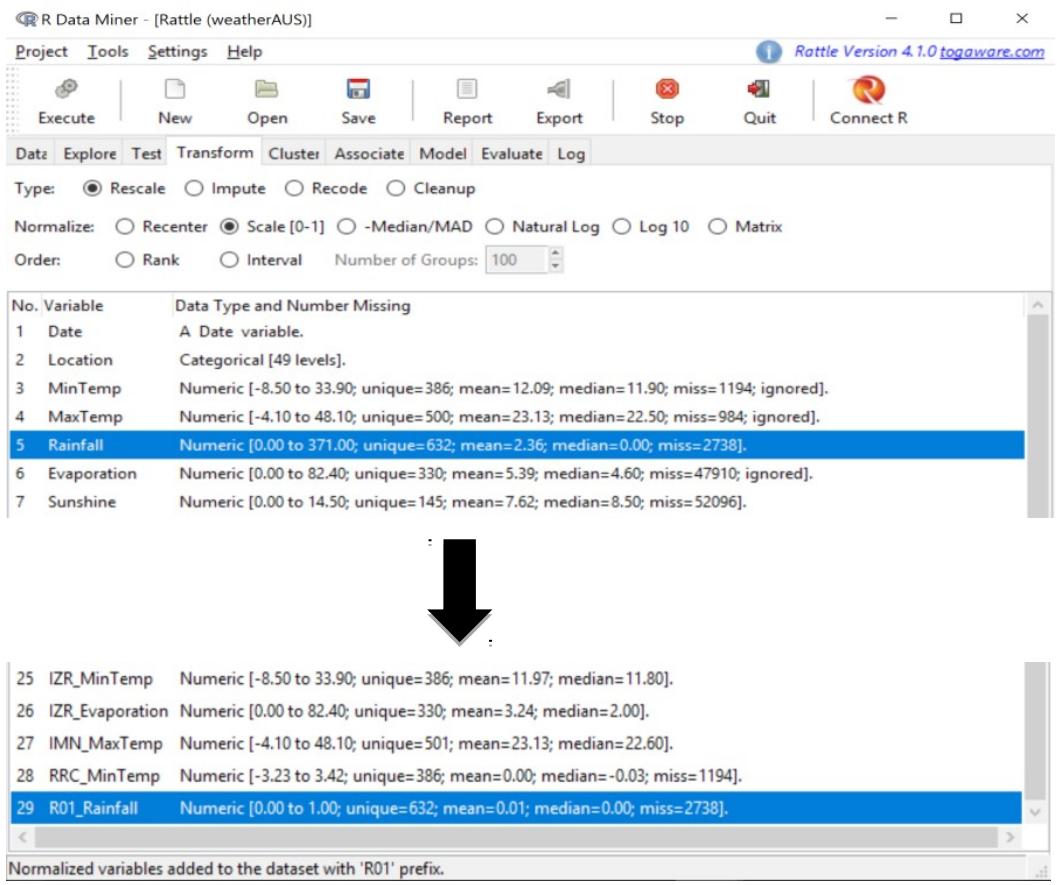
Data transformation in Rattle (2)

- Recentering
  - Common normalisation – recentres and rescales data
  - Subtracts the mean value from each value of a variable (to recentre the variable)
  - Divides by the standard deviation (to rescale)



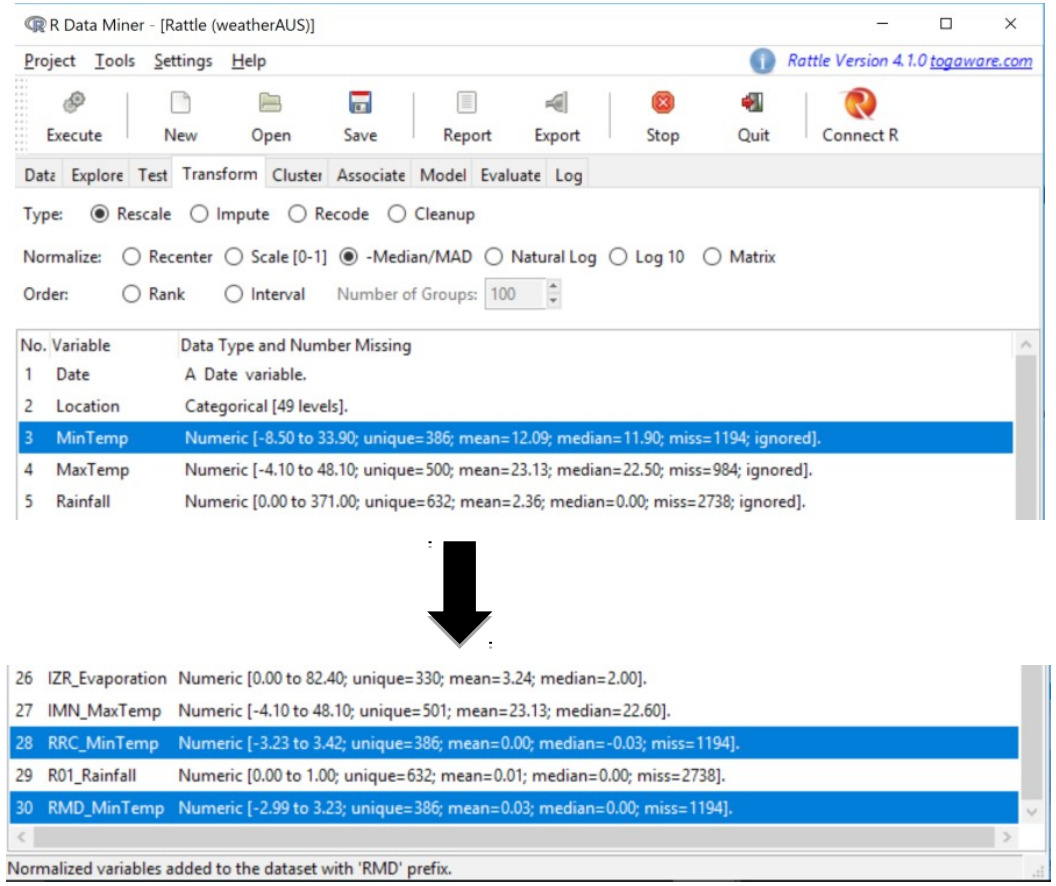
Data transformation in Rattle (3)

- Scaling [0-1]
  - Rescaling to be in [0-1]
  - Subtracts the minimum value from each value of a variable
  - Divides by the difference between maximum and minimum values



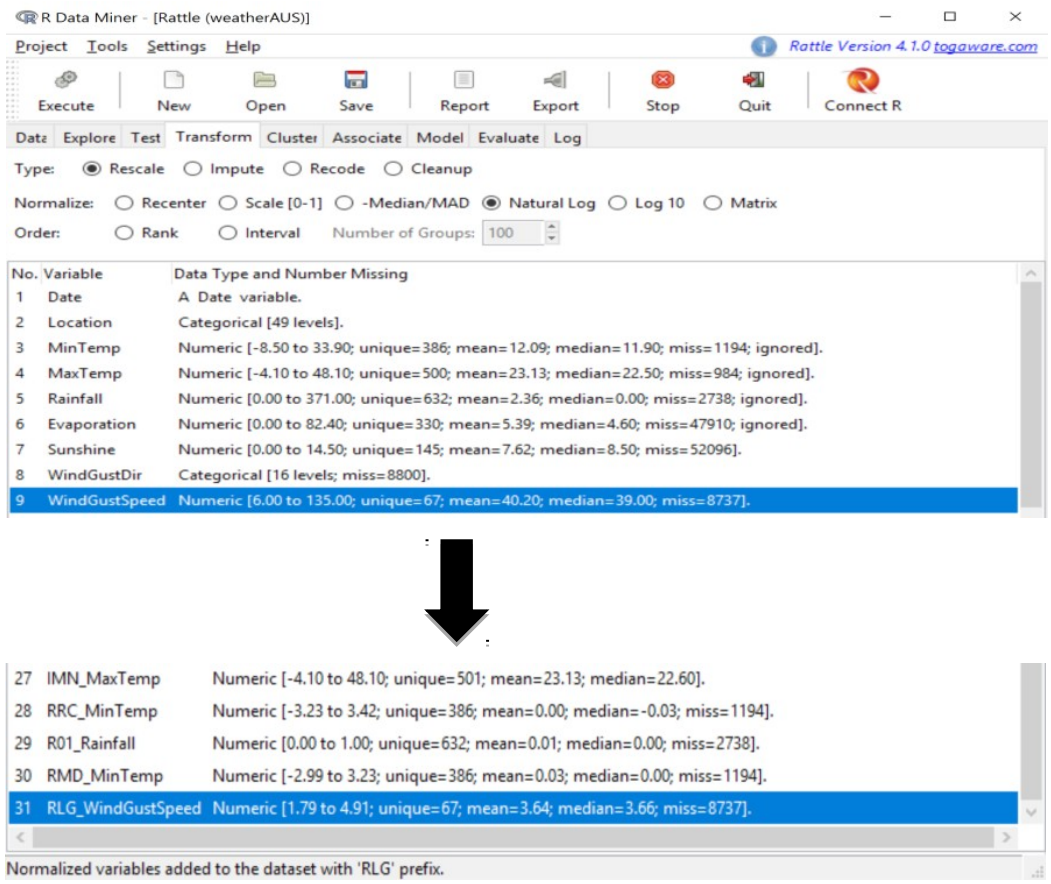
Data transformation in Rattle (4)

- Robust rescaling
  - Robust version of recentering option
  - Subtracts the ***median*** value from each value of a variable (to recentre the variable)
  - Divides by the ***median absolute deviation*** (MAD to rescale)



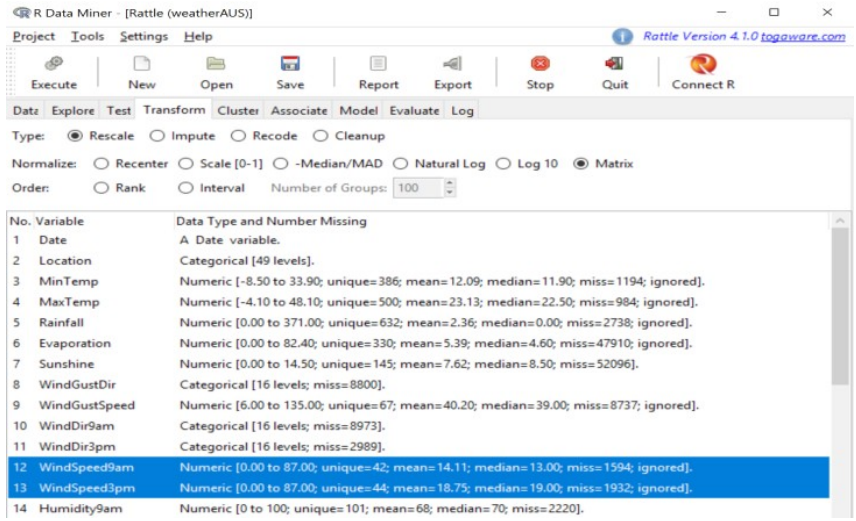
Data transformation in Rattle (5)

- Logarithm transformation
  - Variables with skewed distribution (such as income)
  - Logarithm (as well as natural logarithm) effectively reduces the spread of values
  - Base 10 logarithm: $10,000 -> 4, $100,000 -> 5, $1,000,000-> 6



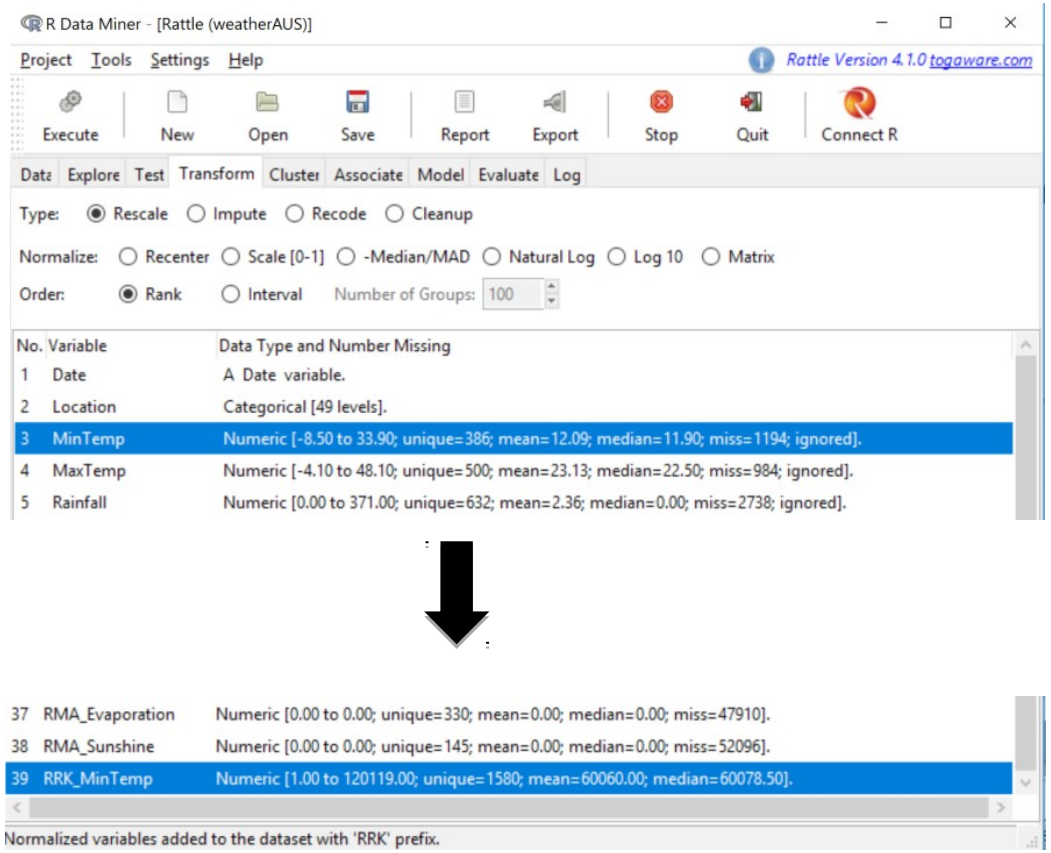
Data transformation in Rattle (6)

- Matrix
  - Transforming data using multiple variables
  - Calculates the sum of all values of multiple variables as matrix total
  - Divides each value of a variable by the matrix total



Data transformation in Rattle (7)

- Ranking
  - Not the actual values, but the relative position within the distribution of values
  - A list of integers (ranks)
  - E.g. [100,50,17,78,20,5,50,6] → [8, 5, 3, 7, 4, 1,5 ,2]



Data transformation using Python

- Several Python packages available for data cleaning, profiling, and analysis
- Most important ones:
  - **Pandas**: provides easy-to-use data structures and data analysis tools
  - **Numpy** and **Scipy**: fundamental packages for scientific computing
  - **Sklearn**: Library for machine learning in Python
  - **Matplotlib**: For generating plots and visualisation

Loading a data set using Python

- Importing libraries

  import pandas as pd

  import numpy as np

  import matplotlib.pyplot as plt

- Reading the dataset in a dataframe using Pandas

  df = pd.read_csv("weather.csv")

Handling missing values in Python (1)

- Checking the number of nulls/NaNs (not-a-number) in the data sets

  df.apply(lambda x: sum(x.isnull()),axis=0)

- Prints number of null values in each variable

- Note: missing values may not always be NaNs.
  - For example: Unknown, 0, -1

Handling missing values in Python (2)

- Deletion

  df.dropna(how='any')

- Mean/median/mode imputation

  df['MinTemp'].fillna(df['MinTemp'].mean(), inplace=True)

  df['MinTemp'].fillna(df['MinTemp'].median(), inplace=True)

  df['WindDir9am'].fillna(df['WindDir9am'].mode(), inplace=True)

Data transformation in Python (1)

- Recentering and rescaling

```
mean_val = df['WindGustSpeed'].mean()
std_val = df['WindGustSpeed'].std()
WindGustSpeedRct = []
for val in df['WindGustSpeed']:
    WindGustSpeedRct.append((val – mean_val) / std_val)
df['WindGustSpeedRct'] = WindGustSpeedRct
```

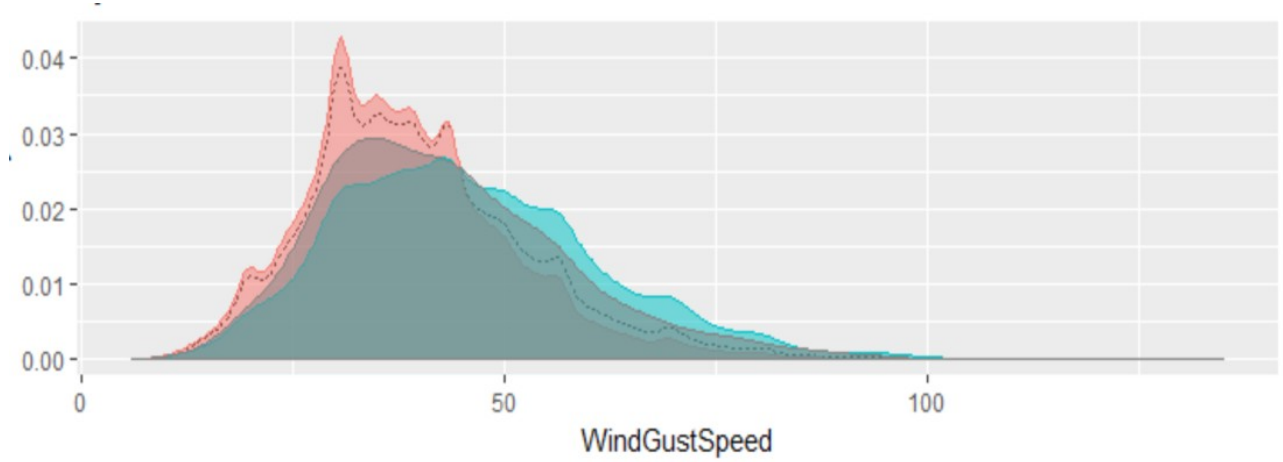
Data transformation in Python (2)

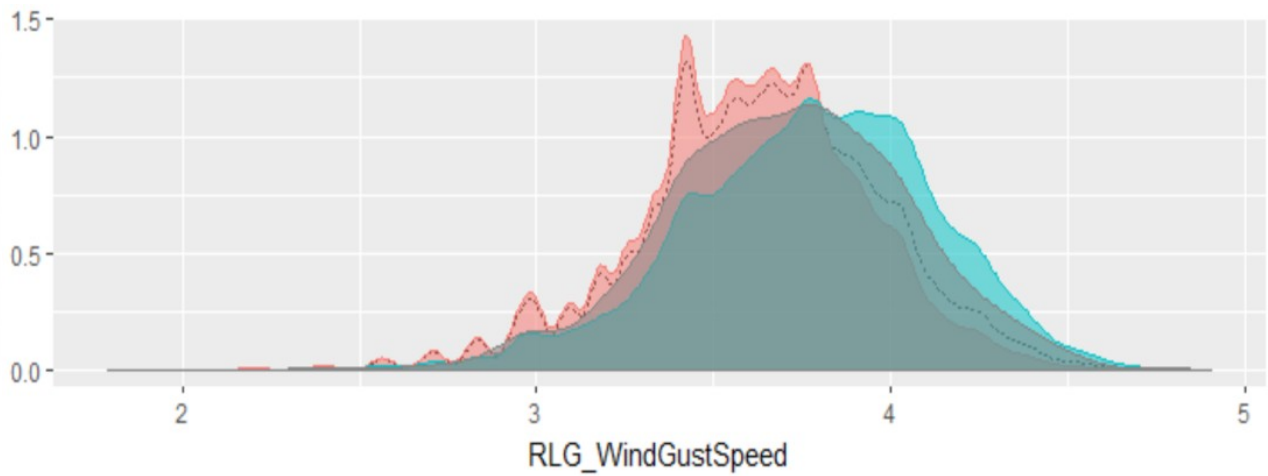
- Logarithm transformation

df['WindGustSpeed'].hist(bins=20)

df['WindGustSpeedLog']=np.log(df['WindGustSpeed'])

df['WindGustSpeedLog'].hist(bins=20)





Summary

- Several data pre-processing tools (open source and commercial) available for efficient data science applications
- Python and Rattle are two such open source tools that are becoming increasingly popular among the data scientists

- Future directions are required towards tools with full life-cycle of data science and interactive design