


vivainsights Python library: CHEAT SHEET

Full documentation site:
<https://microsoft.github.io/vivainsights-py/>



Basics

 [vivainsights](#) is a Python library that offers a set of tools and functions for analysing and visualising data from [Microsoft Viva Insights](#)

You can install or update the package from PyPI by running:
`pip install vivainsights`

 The latest development version and documentation can be found on our GitHub repository: <https://github.com/microsoft/vivainsights-py/>

Example – loading and running the library:

```
import vivainsights as vi
pq_df = vi.import_query("path/pq_df.csv")
vi.create_bar(pq_df, metric = 'Email_hours')
```





Data import / export

Use our handy functions below which are optimized for best practice forgetting data *in* and *out* of Python

- **import_query()**
Import CSV queries faster and pre-formatted for **vivainsights** functions (instead of `read.csv()`)
- **export()**
Copy a data frame to clipboard, or write as a CSV, or a ggplot object as PNG or SVG





Inbuilt datasets

Explore **vivainsights** by using inbuilt demo datasets

-  **load_pq_data()**
Loads a sample Person Query data to the environment
-  **load_mt_data()**
Loads a sample Standard meeting query data to the environment
-  **load_g2g_data()**
Loads a sample group-to-group query data to the environment
-  **p2p_data_sim()**
Person to Person query / edge list based on the graph






Data validation

Validate and understand your data prior to starting a piece of analysis

-  **identify_datefreq()**
Identify date frequency based on a series of dates
-  **identify_holidayweeks()**
Identify likely holiday weeks (for the entire pop) where collaboration hours lie far outside the mean
-  **identify_inactiveweeks()**
Identify likely person-weeks where collaboration hours lie far outside the mean relative to the population average
-  **identify_nkw()**
Identify likely non-knowledge workers where average person collaboration hours lie far outside the mean



Flexible analysis

Flexible analysis functions are versatile, allowing you to pass any metric as a string parameter, e.g., `metric = 'Email_hours'`

-  **create_bar()**
Returns a bar plot showing the average of a selected metric by default. This function creates a bar chart directly from the aggregated / summarized data
-  **create_boxplot()**
Analyzes a selected metric and returns a box plot by default
-  **create_inc()**
Returns a heatmap for the generated incidence analysis
-  **create_line()**
By default, returns a line chart for the defined metric
-  **create_sankey()**
Create a 'networkD3' style sankey chart based on a long count table with two variables


Network analysis

Analyze edge list datasets (e.g., Person-to-person, Group-to-group) from Viva Insights





-  **network_g2g()**
Pass a data frame containing a group-to-group query and return a network plot
-  **network_p2p()**
Pass a data frame containing a person-to-person query and return a network visualization

Exploratory analysis

Explore the data and surface initial hypotheses

-  **create_rank()**
Returns a plot by default, with an option to return a table with all groups (across multiple HR attributes) ranked by the specified metric

Other analysis

-  **create_IV()**
Specify an outcome variable and return IV outputs
-  **p_test()**
Specify an outcome variable and return p-test outputs. All numeric variables in the dataset are used as predictor variables
-  **compute_gini()**
Compute the Gini coefficient, a measure of statistical dispersion to represent inequality
-  **create_lorenz()**
This function computes the Gini coefficient and plots the Lorenz curve based on a selected metric from a Person Query data frame

