Digital plots of Modulus and Strength ratio for Rocks

Release 0.0

Grasselli's Geomechanics Group - University of Toronto

Dec 19, 2022

CONTENTS:

1 Introduction **1**
- 1.1 Deere-Miller - Modulus Ratio (MR) . . . 1
- 1.2 Tatone et al. - Strength Ratio (SR) . . . 5
- 1.3 Poisson's Ratio and Density Plots . . . 6
- 1.4 UCS Classification Systems . . . 8

2 pyrockmodulus **9**
- 2.1 pyrockmodulus package . . . 9

3 Indices and tables **17**

Python Module Index **19**

Index **21**

CHAPTER

ONE

INTRODUCTION

Makes it easier to classify the Deere-Miller - Modulus Ratio [MR] and Tatone et al. - Strength Ratio [SR].

For any suggestions, bugs or if you wish to contribute to the project => REPO

1.1 Deere-Miller - Modulus Ratio (MR)

Cite: Deere DU, Miller RP. Engineering Classification and Index Properties for Intact Rocks. Fort Belvoir, VA: Defense Technical Information Center; 1966.

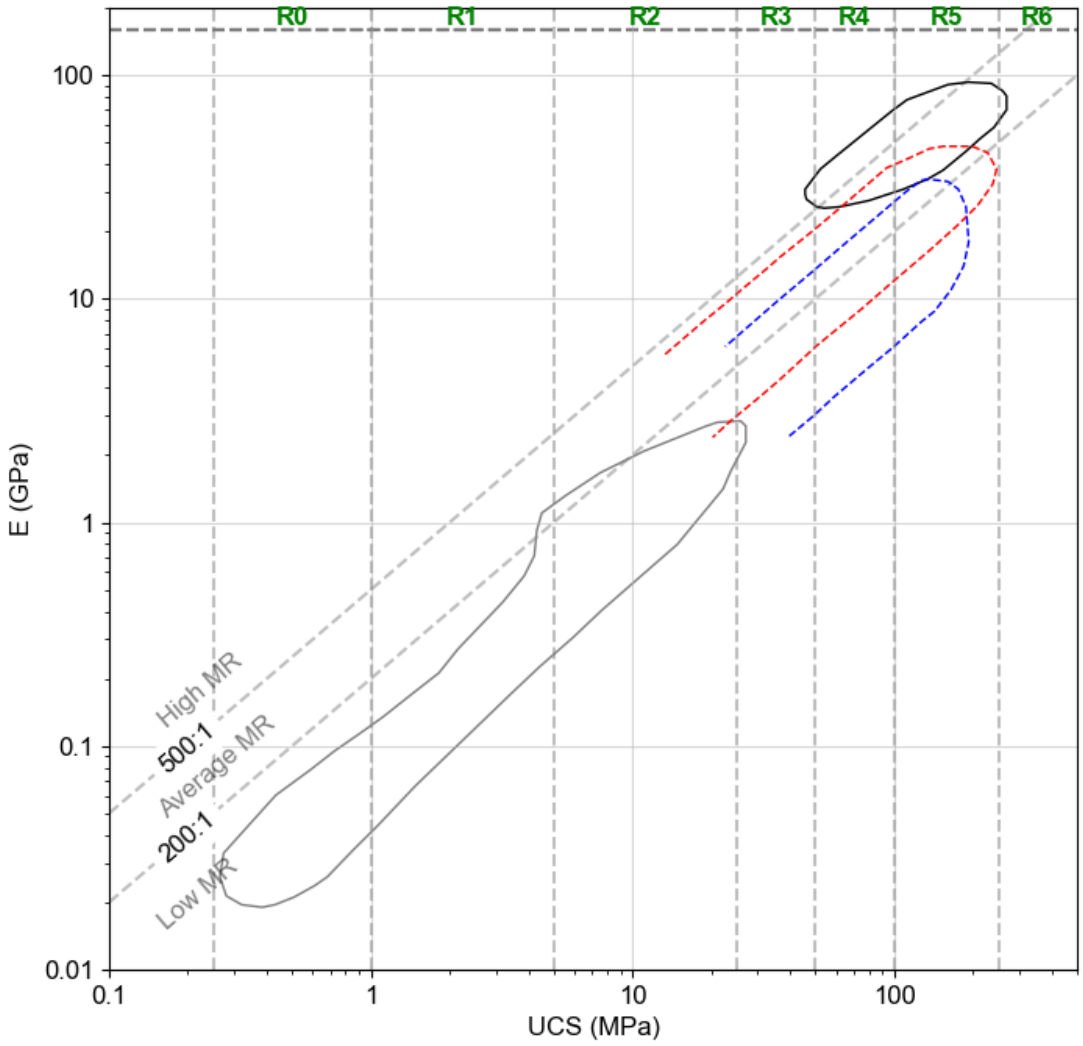
Loads the digitized Deere_Miller clusters and plots them based on the Major Rock Type *(i.e., Igneous / Metamorphic / Sedimentary)*.

- Plot all Major Rock Type in one graph.
- Plots them individually.

Deere-Miller [Modulus Ratio] Example

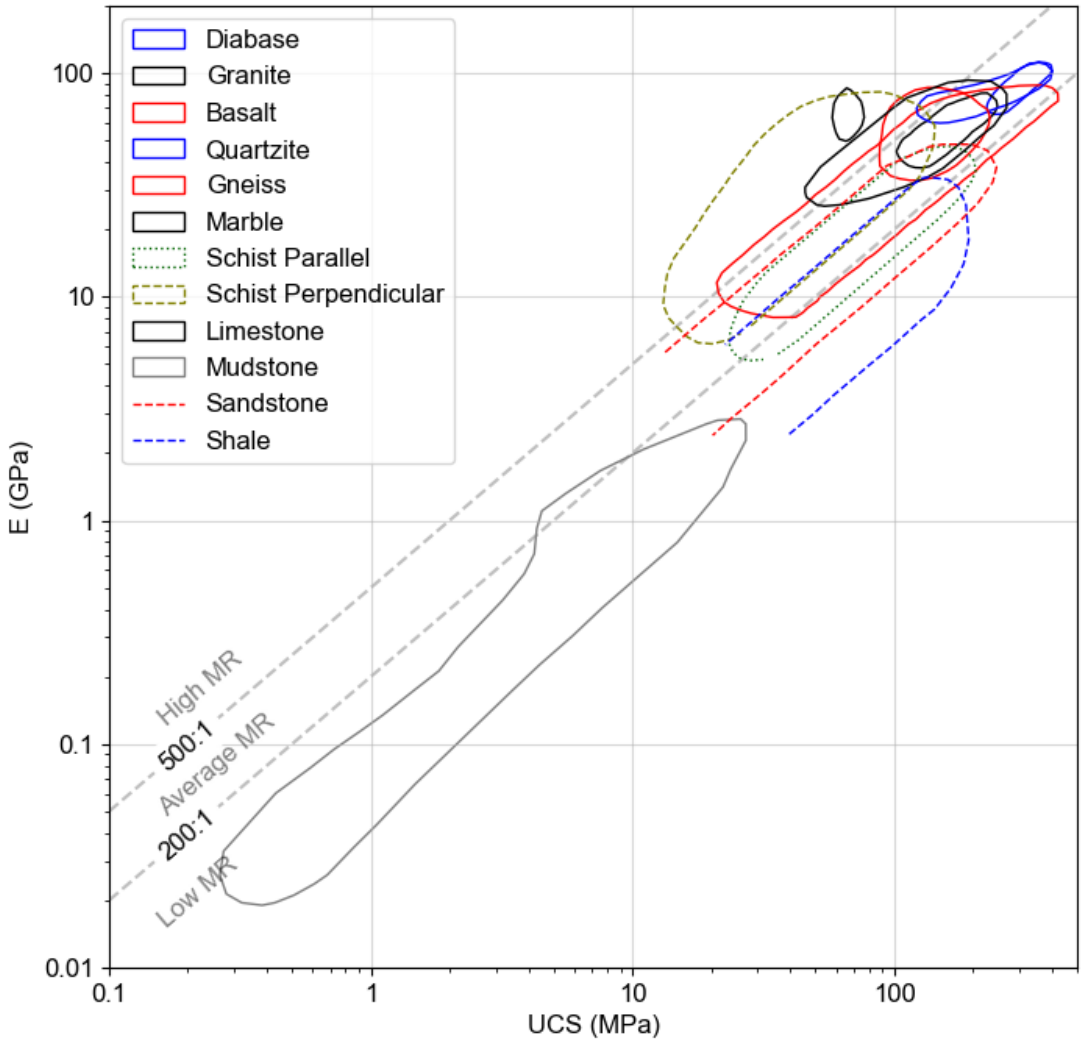
1. Plot the Modulus Ratio of just the Sedimentary clusters with the ISRM 1979 category classification.

```
import pyrockmodulus
import matplotlib.pyplot as plt

xx = pyrockmodulus.modulus_ratio()
xx.initial_processing(plot_all_clusters=False, rock_type_to_plot='Sedimentary', ucs_
↪class_type="ISRMCAT\n1979")
plt.ylabel("E (GPa)")
plt.xlabel("UCS (MPa)")
plt.show()
```



2. Plot the Modulus Ratio with all the categories without the classification. Legend enabled.

```
import pyrockmodulus
import matplotlib.pyplot as plt

xx = pyrockmodulus.modulus_ratio()
xx.initial_processing(plot_all_clusters=True)
plt.ylabel("E (GPa)")
plt.xlabel("UCS (MPa)")
plt.legend()
plt.show()
```



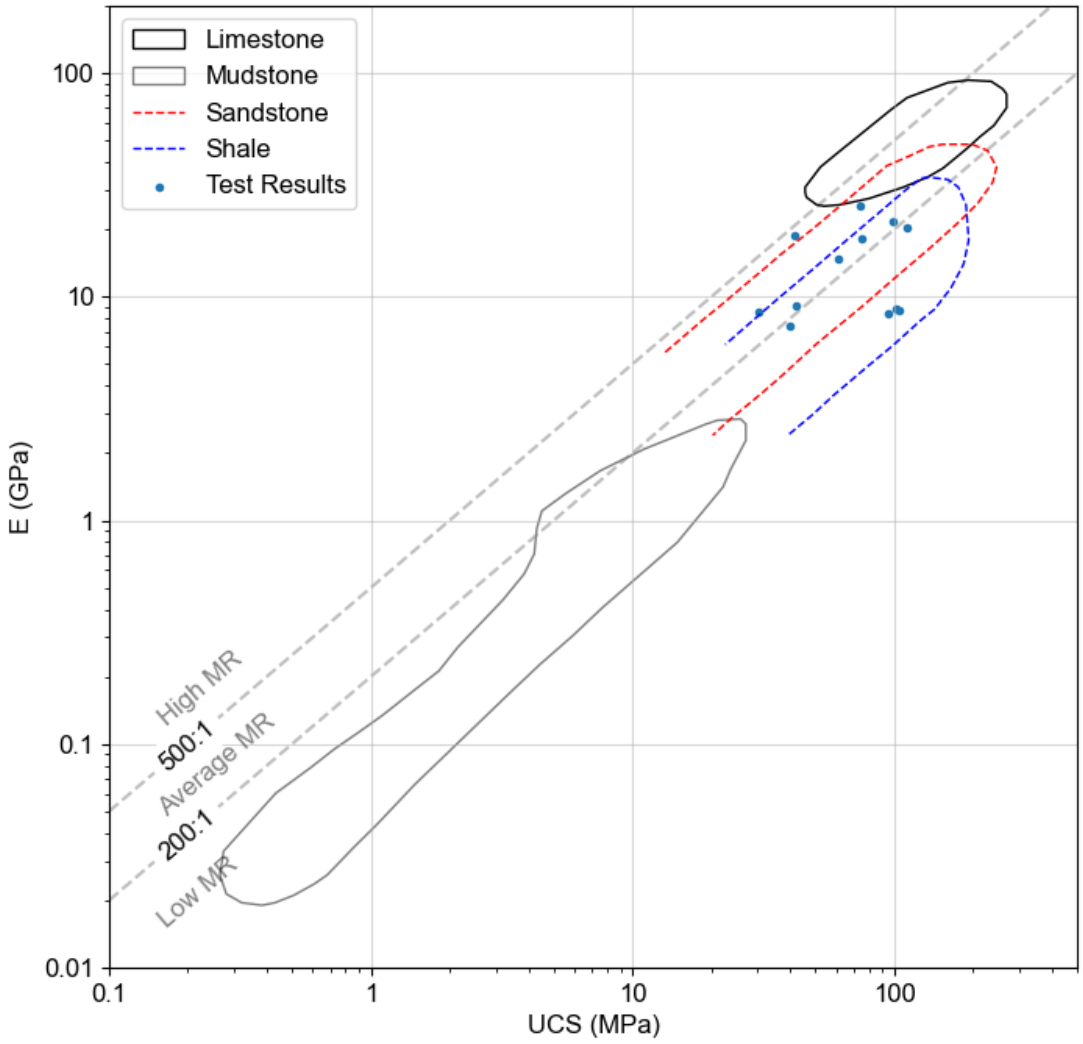
3. Plot the Modulus Ratio of just the Sedimentary clusters overlaid with data from tests.

```
import pyrockmodulus
import matplotlib.pyplot as plt
# Data Set
ucs_data = [75.33, 99.03, 111.69, 30.17, 73.76, 41.69, 42.09, 60.99, 39.65, 94.52, 104.6,
↪ 102.03]
E_data = [18.31, 21.85, 20.51, 8.62, 25.72, 18.68, 9.2, 14.67, 7.38, 8.48, 8.7, 8.82]
xx = pyrockmodulus.modulus_ratio()
plotting_axis = xx.initial_processing(rock_type_to_plot='Sedimentary')
# Plot the data on the Deere-Miller axis
plotting_axis.scatter(ucs_data, E_data, label='Test Results', marker='.')
plt.ylabel("E (GPa)")
plt.xlabel("UCS (MPa)")
```

(continues on next page)

(continued from previous page)

```
plt.legend()
plt.show()
```



1.2 Tatone et al. - Strength Ratio (SR)

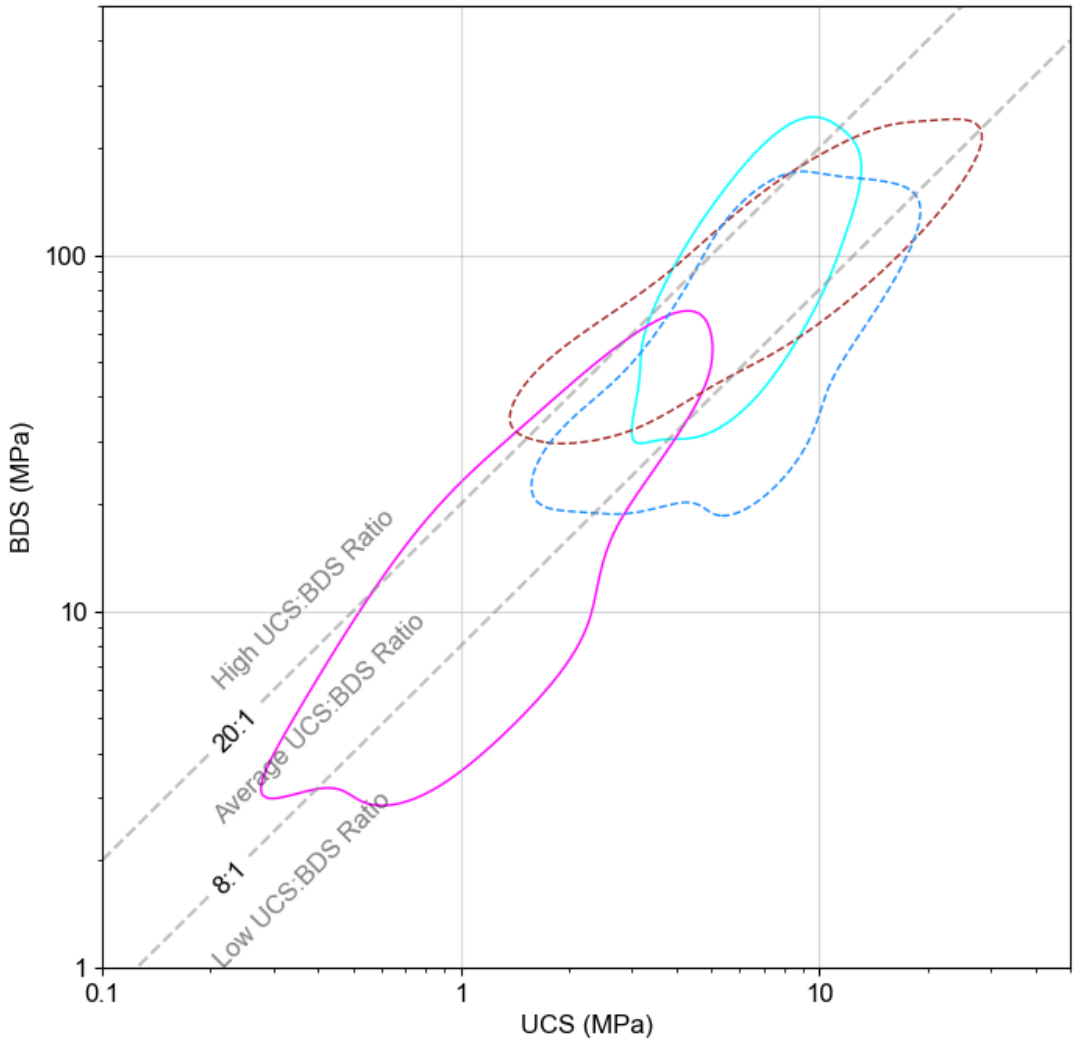
Tatone, B.S.A., Abdelaziz, A. & Grasselli, G. Novel Mechanical Classification Method of Rock Based on the Uniaxial Compressive Strength and Brazilian Disc Strength. Rock Mech Rock Eng 55, 2503–2507 (2022). https://doi.org/10.1007/s00603-021-02759-7

Loads the constructed Tatone et al. UCS:BDS clusters and plots them based on the Major Rock Type *(i.e., Igneous / Metamorphic / Sedimentary)*.

- Plot all Major Rock Type in one graph.
- Plots them individually.

The functionality is similar to that of the modulus ratio.

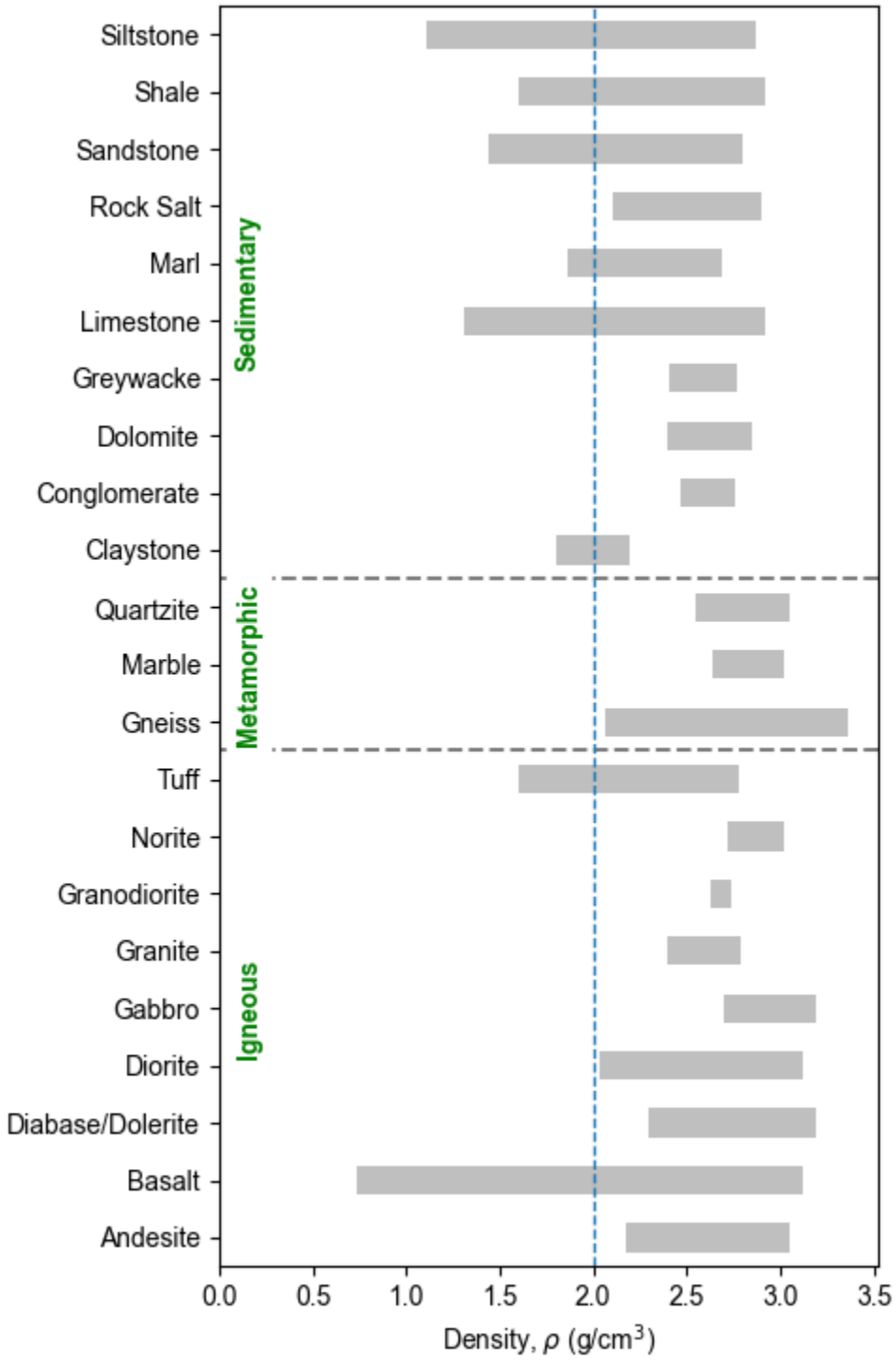
```
import pyrockmodulus
import matplotlib.pyplot as plt

xx = pyrockmodulus.strength_ratio()
xx.initial_processing(plot_all_clusters=False, rock_type_to_plot='Sedimentary')
plt.ylabel("BDS (MPa)")
plt.xlabel("UCS (MPa)")
plt.show()
```



1.3 Poisson's Ratio and Density Plots

Plot the most common ranges of density and poisson's ratio for rock. This data can then be overlaid with data from a specific source to show comparison.

```
import matplotlib.pyplot as plt
import pyrockmodulus
xx = pyrockmodulus.poisson_density()
df_data = xx.initial_processing()
ax1 = xx.plot_span_chart(df_data, ['Min_D', 'Max_D'], 'Density', r'$\rho$ g/cm$^{3}$')
ax1.axvline(2.0, lw=1, ls='--')
plt.show()
```


Siltstone
Shale
Sandstone
Rock Salt
Marl
Limestone
Greywacke
Dolomite
Conglomerate
Claystone
Quartzite
Marble
Gneiss
Tuff
Norite
Granodiorite
Granite
Gabbro
Diorite
Diabase/Dolerite
Basalt
Andesite
Sedimentary
Metamorphic
Igneous
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Density, ρ (g/cm³)

1.4 UCS Classification Systems

This file holds the dictionaries for the various UCS classification systems available. References for those systems are within the file. All values **must** be in **MPa**. Available classification systems *'ISRM\n1977', 'ISRMCAT\n1979', 'Bieniawski\n1974', 'Jennings\n1973', 'Broch & Franklin\n1972', 'Geological Society\n1970', 'Deere & Miller\n1966', 'Coates\n1964', 'Coates & Parsons\n1966', 'ISO 14689\n2017', 'Anon\n1977', 'Anon\n1979', 'Ramamurthy\n2004'*

UCS Classification System Examples

1. Display the limits and the classification system default in the script.

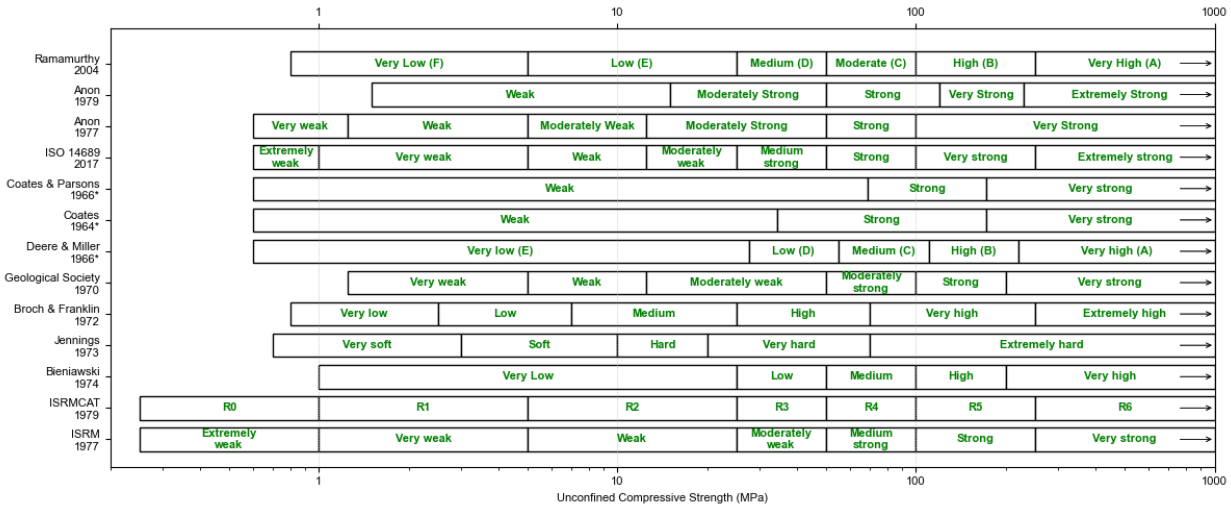
```
import pyrockmodulus.rock_variables as ucs_class
ucs_class.ucs_strength_criteria('ISRMCAT\n1979')
```

Output

```
(['R0', 'R1', 'R2', 'R3', 'R4', 'R5', 'R6'], [0.25, 1, 5, 25, 50, 100, 250, 1000])
```

2. A horizontal bar like plot to show the various uniaxial strength classification systems.

```
import pyrockmodulus.ucs_bar_chart_plot as ucs_classification_plot
import matplotlib.pyplot as plt

ucs_class = ucs_classification_plot.initial_processing()
plt.show()
```



CHAPTER

TWO

PYROCKMODULUS

2.1 pyrockmodulus package

2.1.1 Submodules

Deere-Miller - Modulus Ratio (MR)

class pyrockmodulus.pyrockmodulus.**modulus_ratio**

Bases: `object`

Based on the classification of Deere DU, Miller RP. Engineering Classification and Index Properties for Intact Rocks. Fort Belvoir, VA: Defense Technical Information Center; 1966. Data digitization courtesy of Rohatgi, Ankit. "WebPlotDigitizer." (2017).

ADVANCED: By assigning the *_rocktype_dictionary* variable, more control over the clusters being plotted is gained.

abline(*slope*, *intercept*, *dr_state*, *multiplier=1*, *ratio=''*, *ax=None*, *x_text_loc=0.15*)

Function to plot the slopped lines based on a slope and a y-intercept, basically mx+c. It is defined to form the Low/Avg/High MR ratio in the deere-miller classification plot.

Parameters

- **slope** (*float*) – the slope of the line
- **intercept** (*float*) – the intercept of the lube
- **dr_state** (*str*) – draw state to move between the line drawing and the placement/writing of the text. Options [Line, Text]
- **multiplier** (*int*) – in case of a need of a multiplier
- **ratio** (*str*) – text associated with the MR modulus
- **ax** (*matplotlib*) – Matplotlib Axis
- **x_text_loc** (*float*) – slope to write text

Returns

Return type

deere_miller_clusters(*ax*, *df_of_clusters_deere_miller*, *r_type=None*, *plot_all_clusters_bool=False*)

Load information needed to plot

Parameters

- **ax** (*matplotlib*) – Axis to plot on

- **df_of_clusters_deere_miller** (*dict*) – will plot defined cluster. Options Sedimentary, Igneous, Metamorphic.
- **r_type** (*str*) – Define the rock type to be plotted. plot_all_clusters_bool MUST be false.
- **plot_all_clusters_bool** (*bool*) – Plot all the clusters.

Returns

Return type

format_axis(*ax*, *state=''*, *major_axis_vline=True*)

Format log-log Axis

Parameters

- **ax** (*matplotlib*) – Axis to plot on
- **state** – state to enable to disable slopped lines
- **major_axis_vline** (*bool*) – Plot the major axis vlines

Returns

Return type

initial_processing(*rock_type_to_plot=None*, *plot_all_clusters=False*, *ucs_class_type=None*, *ax=None*)

Main function to plot the Modulus Ratio underlay

param rock_type_to_plot
Rock cluster type to plot.

type rock_type_to_plot
UCS Strength Criteria adopted. Options Sedimentary, Igneous, Metamorphic.

param ucs_class_type
UCS Strength Criteria adopted. Options 'ISRM

1977', 'ISRMCAT 1979', 'Bieniawski 1974', 'Jennings 1973', 'Broch & Franklin 1972', 'Geological Society 1970', 'Deere & Miller 1966', 'Coates 1964', 'Coates & Parsons 1966', 'ISO 14689 2017', 'Anon 1977', 'Anon 1979', 'Ramamurthy 2004'

type ucs_class_type
str

param ax
Axis to plot on

type ax
matplotlib

return
Axis

rtype
Matplotlib Axis

load_data(*df_deere_miller_data*)

Load the file that holds the digital deere_miller cluster points. This information will be used to plot the deere-miller clusters based on the user requirements.

Parameters
df_deere_miller_data (*str*) – file path to the location of the csv

Returns
dictionary containing the type of rock and the points that form its cluster.

Return type
dict

`plot_clusters`(*k*, *v*, *ax*, *df_of_clusters_deere_miller*)

Plot the clusters

Parameters

- **k** (*str*) – key
- **v** (*str*) – value
- **ax** (*matplotlib*) – Axis to plot on
- **df_of_clusters_deere_miller** (*dict*) – dictionary containing the type of rock and the points that form its cluster.

Returns

Return type

`plot_v_lines`(*vlines*, *ax*)

Plot lines and annotate the UCS Strength Criteria adopted

Parameters

- **vlines** (*list[float]*) – Locations of V Lines
- **ax** (*matplotlib*) – Axis to plot

Returns

Return type

Tatone et al. - Strength Ratio (SR)

`class` pyrockmodulus.pyrockmodulus.`strength_ratio`

Bases: `object`

Based on the classification of Tatone, B.S.A., Abdelaziz, A. & Grasselli, G. Novel Mechanical Classification Method of Rock Based on the Uniaxial Compressive Strength and Brazilian Disc Strength. Rock Mech Rock Eng 55, 2503–2507 (2022). https://doi.org/10.1007/s00603-021-02759-7 Data was built using a bivariant KDE # https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.gaussian_kde.html # https://towardsdatascience.com/simple-example-of-2d-density-plots-in-python-83b83b934f67

ADVANCED: By assigning the *_rocktype_dict* variable, more control over the clusters being plotted is gained.

`abline`(*slope*, *intercept*, *dr_state*, *multiplier=1*, *ratio=''*, *ax=None*)

Function to plot the slopped lines based on a slope and a y-intercept, basically mx+c. It is defined to form the Low/Avg/High MR ratio in the deere-miller classification plot.

Parameters

- **slope** (*float*) – the slope of the line
- **intercept** (*float*) – the intercept of the lube
- **dr_state** (*str*) – draw state to move between the line drawing and the placement/writing of the text. Options [Line, Text]

- **multiplier** (*int*) – in case of a need of a multiplier
- **ratio** (*str*) – text associated with the MR modulus
- **ax** (*matplotlib*) – Matplotlib Axis
- **x_text_loc** (*float*) – slope to write text

Returns

Return type

format_axis(*ax*, *state=''*, *major_axis_vline=True*)

Format log-log Axis

Parameters

- **ax** (*matplotlib*) – Axis to plot on
- **state** – state to enable to disable slopped lines
- **major_axis_vline** (*bool*) – Plot the major axis vlines

Returns

Return type

initial_processing(*rock_type_to_plot=None*, *plot_all_clusters=False*, *ucs_class_type=None*, *ax=None*)

Main function to plot the Modulus Ratio underlay

param rock_type_to_plot
Rock cluster type to plot.

type rock_type_to_plot
UCS Strength Criteria adopted. Options Sedimentary, Igneous, Metamorphic.

param ucs_class_type
UCS Strength Criteria adopted. Options 'ISRM

1977', 'ISRMCAT 1979', 'Bieniawski 1974', 'Jennings 1973', 'Broch & Franklin 1972', 'Geological Society 1970', 'Deere & Miller 1966', 'Coates 1964', 'Coates & Parsons 1966', 'ISO 14689 2017', 'Anon 1977', 'Anon 1979', 'Ramamurthy 2004'

type ucs_class_type
str

param ax
Axis to plot on

type ax
matplotlib

return
Axis

rtype
Matplotlib Axis

Poisson's Ratio and Density Plots

class pyrockmodulus.pyrockmodulus.**poisson_density**

Bases: object

Load Poisson Ratio and Density information

initial_processing()

Load the variables and initialise the dataframe.

Returns
DataFrame containing the Min/Max Poisson Ratio and the Min/Max Density divided by Rock Name nad ROck Group. The latter two impact the y-axis and the hbars and titles.

Return type
pandas.DataFrame

plot_span_chart(*df_to_plot*, *variable_span*, *variable_label*, *variable_units*, *ax=None*)

Plot a chart divided by the rock type and rock group.

Parameters

- **df_to_plot** (*pandas.DataFrame*) – Panda Dataframe to plot
- **variable_span** (*list[str, str]*) – Span (i.e., min and max values) passed as a list. Must be the Column Header name in the DataFrame!
- **variable_label** (*str*) – Variable Name. X axis label
- **variable_units** (*str*) – Variable Units. X axis label unit
- **ax** (*Matplolib*) – Matplotlib Axis to plot On

Returns
Matplotlib AxesSubplots

Return type
Matplotlib Axis

UCS Classification Systems

pyrockmodulus.rock_variables.**ucs_strength_criteria**(*type*)

Insert all UCS Strength Criterion Here. ## ALL VALUES ARE IN MPa # Name Format {Reference Name: [Name of Category]} # Value Format {Reference Name: [Boundaries Location]} <=> in MPa # converted_psi [Reference name that are converted from psi to MPa]

Parameters
type (*str*) – rock classification system to load

Returns

Return type

2.1.2 Supporting Modules

pyrockmodulus.formatting_codes

pyrockmodulus.formatting_codes.**bold_text**(*val*)

Returns text as bold

Parameters
val (*str*) – Text

Returns
Text as bold

Return type
str

pyrockmodulus.formatting_codes.**calc_timer_values**(*end_time*)

Function to calculate the time

Parameters
end_time (*float*) – Time (Difference in time in seconds)

Returns
Time in minutes and seconds

Return type
float

pyrockmodulus.formatting_codes.**docstring_creator**(*df*)

Write the example output for a docstring DataFrame

Parameters
df (*pandas.DataFrame*) – DataFrame to be read

Returns
prints the docstring and type for each element in the DataFrame

Return type
str

pyrockmodulus.formatting_codes.**green_text**(*val*)

Returns text as bold in green font color

Parameters
val (*str*) – Text

Returns
Text as bold in green font color

Return type
str

pyrockmodulus.formatting_codes.**print_progress**(*iteration*, *total*, *prefix=''*, *suffix=''*, *decimals=1*, *bar_length=50*)

Call in a loop to create terminal progress bar Adjusted bar length to 50, to display on small screen

Parameters

- **iteration** (*int*) – current iteration
- **total** (*int*) – total iteration

- **prefix** (*str*) – prefix string
- **suffix** (*str*) – suffix string
- **decimals** (*int*) – positive number of decimals in percent complete
- **bar_length** (*int*) – character length of bar

Returns
system output showing progress

Return type

pyrockmodulus.formatting_codes.**red_text**(*val*)

Returns text as bold in red font color

Parameters
val (*str*) – Text

Returns
Text as bold in red font color

Return type
str

pyrockmodulus.ucs_bar_chart_plot

pyrockmodulus.ucs_bar_chart_plot.**initial_processing**()

pyrockmodulus.ucs_bar_chart_plot.**my_path** = '/hdd/home/aly/Desktop/Dropbox/Python_Codes/digital_modulus_strength_ratio/pyrockmodulus'

Default MATPLOTLIB Fonts

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

p

pyrockmodulus.formatting_codes, 14
pyrockmodulus.pyrockmodulus, 13
pyrockmodulus.rock_variables, 13
pyrockmodulus.ucs_bar_chart_plot, 15

INDEX

A

abline() (*pyrockmodulus.pyrockmodulus.modulus_ratio method*), 9
abline() (*pyrockmodulus.pyrockmodulus.strength_ratio method*), 11

B

bold_text() (*in module pyrockmodulus.formatting_codes*), 14

C

calc_timer_values() (*in module pyrockmodulus.formatting_codes*), 14

D

deere_miller_clusters() (*pyrockmodulus.pyrockmodulus.modulus_ratio method*), 9
docstring_creator() (*in module pyrockmodulus.formatting_codes*), 14

F

format_axis() (*pyrockmodulus.pyrockmodulus.modulus_ratio method*), 10
format_axis() (*pyrockmodulus.pyrockmodulus.strength_ratio method*), 12

G

green_text() (*in module pyrockmodulus.formatting_codes*), 14

I

initial_processing() (*in module pyrockmodulus.ucs_bar_chart_plot*), 15
initial_processing() (*pyrockmodulus.pyrockmodulus.modulus_ratio method*), 10
initial_processing() (*pyrockmodulus.pyrockmodulus.poisson_density method*), 13
initial_processing() (*pyrockmodulus.pyrockmodulus.strength_ratio method*), 12

L

load_data() (*pyrockmodulus.pyrockmodulus.modulus_ratio method*), 10

M

module
    pyrockmodulus.formatting_codes, 14
    pyrockmodulus.pyrockmodulus, 9, 11, 13
    pyrockmodulus.rock_variables, 13
    pyrockmodulus.ucs_bar_chart_plot, 15
modulus_ratio (*class in pyrockmodulus.pyrockmodulus*), 9
my_path (*in module pyrockmodulus.ucs_bar_chart_plot*), 15

P

plot_clusters() (*pyrockmodulus.pyrockmodulus.modulus_ratio method*), 11
plot_span_chart() (*pyrockmodulus.pyrockmodulus.poisson_density method*), 13
plot_v_lines() (*pyrockmodulus.pyrockmodulus.modulus_ratio method*), 11
poisson_density (*class in pyrockmodulus.pyrockmodulus*), 13
print_progress() (*in module pyrockmodulus.formatting_codes*), 14
pyrockmodulus.formatting_codes
    module, 14
pyrockmodulus.pyrockmodulus
    module, 9, 11, 13
pyrockmodulus.rock_variables

module, 13
pyrockmodulus.ucs_bar_chart_plot
module, 15

R

red_text() (*in module pyrockmodulus.formatting_codes*), 15

S

strength_ratio (*class in pyrockmodulus.pyrockmodulus*), 11

U

ucs_strength_criteria() (*in module pyrockmodulus.rock_variables*), 13