# What is Siuba?

The siuba python library brings the power of R’s dplyr and the tidyverse to Python. Gain access to functions like:

select() – keep certain columns of data.

filter() – keep certain rows of data.

mutate() – create or modify an existing column of data.

summarize() – reduce one or more columns down to a single number.

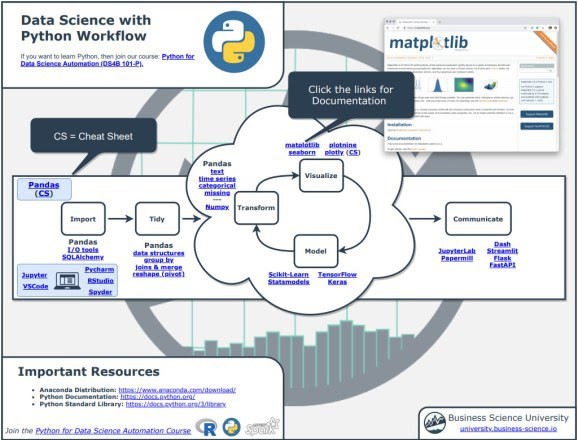
arrange() – reorder the rows of data.

# Before we get started, get the Python Cheat Sheet

Siuba is great for data wrangling in Python if you are coming from an R background. But, you might want to explore documentation for the entire Python Ecosystem.

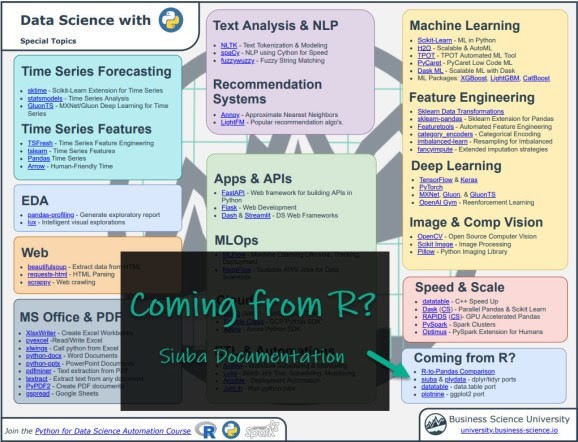
### Ultimate Python Cheat Sheet:

This gives you access to the entire Python Ecosystem at your fingertips via hyperlinked documenation and cheat sheets.



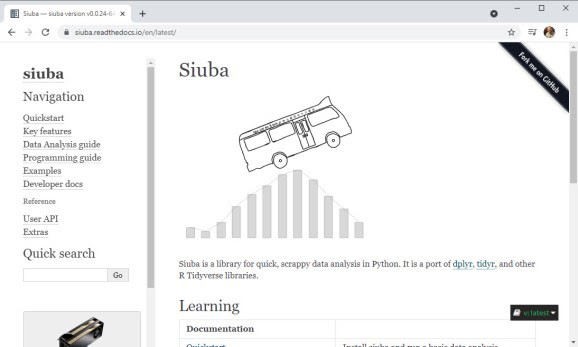
### Navigate to “Coming From R?” Section

Next, go to the section, “Coming from R?”. You can quickly get to the Siuba Documentation.



### Explore Siuba

You have access to the Siuba Documentation at your fingertips.



Onto the tutorial.

# How Siuba Works

From the *Siuba Documentation*, you can see that there are “verbs”, “siu expressions”, and the pipe (>>). We’ll test these out in our tutorial.

**concept example meaning**

a function that operates on a table, like a DataFrame or SQL

verb group\_by(...)

siu

expression \_.hp.mean()

mtcars >>

pipe

group\_by(...)

table

an expression created with siuba.\_, that represents actions you want to perform

a syntax that allows you to chain verbs with the >> operator

# Taking Siuba for a Test Spin

Let’s try out siuba’s data wrangling capabilities.

## Step 1: Load Libraries and Data

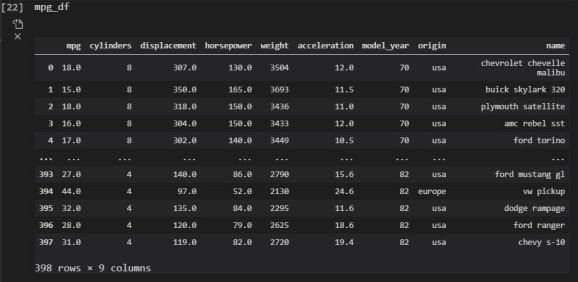
First, let’s load the libraries and data. From the libraries, we’ll import numpy and pandas along with:

\_: Needed to create “siu expressions”

dplyr.verbs: We’ll import group\_by(), summarize(), and mutate()



We’ll also load the mpg\_df data set.

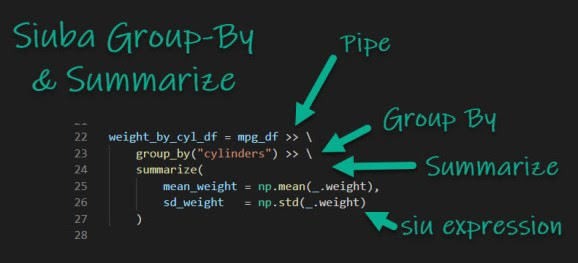


## Step 2: Group By and Summarize

Goal: Mean and Standard Deviation of weight by engine size

We’ll perform a basic operation: group\_by() and summarize() to get the mean and standard deviation of vehicle weight by engine size.

### Group-By Summarize Code



Let’s explain each operation in detail so you understand what’s going on.

**Pipe (>>):** This sends the output of a previous call (a Pandas DataFrame) as the first input of the next function.

**Group By (group\_by()):** This tells python that we want to perform an operation group- wise. We specify by the “cylinder” column.

**Summarize (summarize()):** This tells python that, for each group, we want to summarize the weight to return the mean and standard deviation.

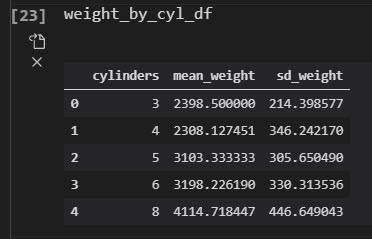
Inside the Summarize, we use **“siu expressions”**, which allow us to reference columns (e.g. \_.weight) while we perform the summarization.

We take advantage of numpy for our summarization, using np.mean() and

np.std() to calculate mean and standard deviation.

### Code Output

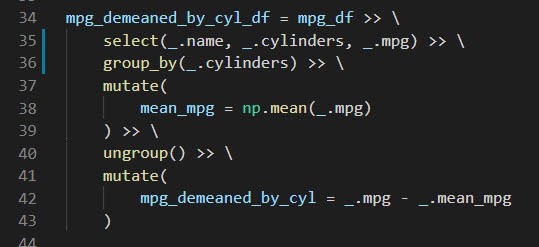
This returns the vehicle weight summarized by the engine size (number of cylinders).



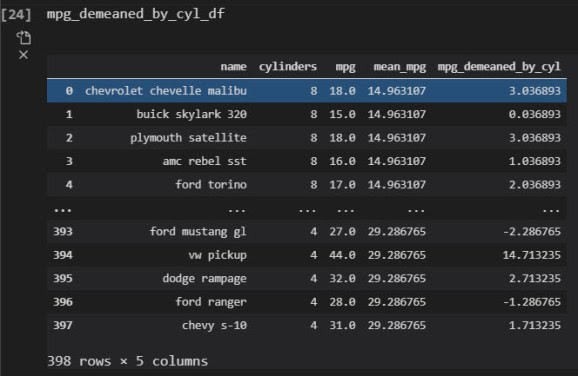
## Step 3: More Advanced Example (Group By and Mutate)

Goal: De-mean the mpg by average of each cylinder

We’ll go through a more complex example using a group\_by() and mutate(). See if you can figure out what’s going on here. Tip – Try reading it like a paragraph in a text.



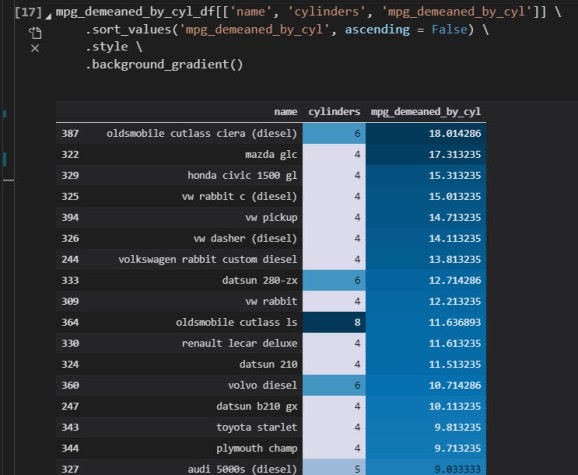
This returns the vehicle fuel efficiency (mpg) de-meaned (removing the average by cylinder class). So now we can compare how the fuel efficiency of each vehicle model compares to the average fuel efficiency within groups of the same engine size.



## Step 4: From Siuba To Pandas

Goal: Format the de-meaned MPG to highlight most fuel efficient vehicles by engine size.

Siuba is great. It returns a DataFrame, which means we can use Pandas. One thing you might want to do is format the background in the table to highlight if vehicle model’s fuel efficiency is above or below the average by engine size. We can do this with **Pandas Table Styles.**



# Summary

This was a short introduction to siuba, which brings dplyr to python. If you’re coming from R,

siuba is a great package to warm yourself up to Python.

With that said, you’re eventually going to want to learn pandas, the most widely used data wrangling tool in Python. Why?

Most Python Teams use Pandas

99% of data wrangling code is written in Pandas

So, it makes sense to eventually learn Pandas to help with communication and working on R/Python teams.