Descriptive Statistic

Overview

- Mean
- Median
- Quartile
- Mode
- Variance and Standard Deviation
- Range

Mean

- **Mean** is the measure of central tendency that represents the average value of a set of numbers.
- Computed by *adding up all the values in the set* and *dividing the result* by the *total number of values*.
- For example, suppose we have a set of numbers {2, 4, 6, 8}.
  - The mean of this set can be calculated by adding up all the numbers and then dividing by the total number of values, which is 4:
  - (2 + 4 + 6 + 8) / 4 = 5
  - Therefore, the mean of the set {2, 4, 6, 8} is 5.

Median

- The median is a measure of central tendency that represents the middle value of a set of numbers when they are arranged in order from smallest to largest (or largest to smallest).
- To find the median of a set of numbers,
  - We first arrange the numbers in order from smallest to largest (or largest to smallest).
  - If the set has an odd number of values, then the median is the middle value.
  - If the set has an even number of values, then the median is the average of the two middle values.

Finding Median

- For example, suppose we have a set of numbers {3, 1, 6, 2, 7}.
  - First, we arrange the numbers in order:{1, 2, 3, 6, 7}
  - Since there are 5 values in this set, which is odd, the median is the middle value, which is 3.
- Now suppose we have another set of numbers {3, 1, 6, 2, 7, 5}.
  - Again, we arrange the numbers in order:{1, 2, 3, 5, 6, 7}
  - Since there are 6 values in this set, which is even, the median is the average of the two middle values, which are 3 and 5.
  - Therefore, the median of the set {3, 1, 6, 2, 7, 5} is (3 + 5) / 2 = 4.

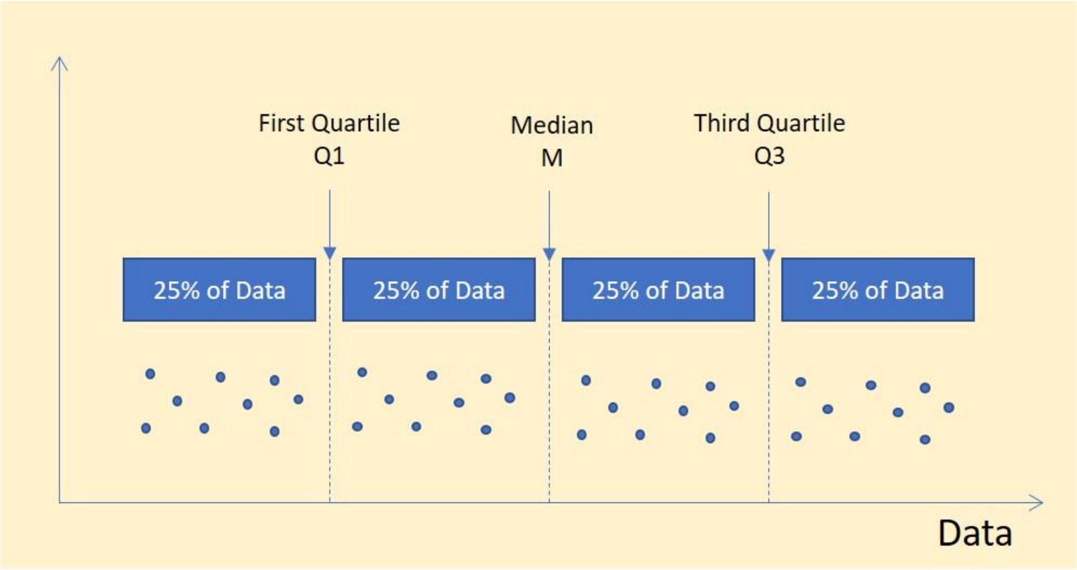
Quartiles

- Quartiles are statistical measures that divide a dataset into four equal parts,
- Used to understand the distribution and dispersion of numerical data.
- The three quartiles, denoted as Q1, Q2, and Q3, divide the data into four intervals.
- **First Quartile (Q1):** Q1 is the value that separates the lowest 25% of the data from the remaining 75%. It is also known as the lower quartile.
- **Second Quartile (Q2):** Q2 is the median of the data and represents the middle value that divides the dataset into two equal halves. It separates the lower 50% from the upper 50

Quartiles

- **Third Quartile (Q3):** Q3 is the value that separates the lowest 75% of the data from the remaining highest 25%. It is also known as the upper quartile.

- The **interquartile range (IQR)** is a measure of statistical dispersion, calculated as the difference between the third and first quartiles (Q3 - Q1).



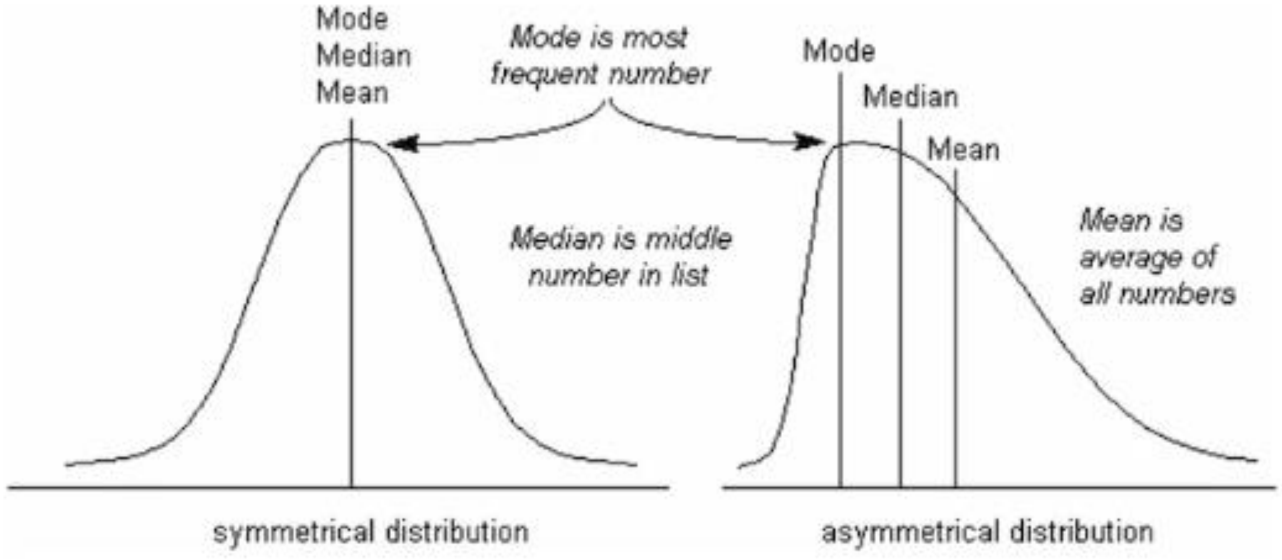
Mode

- The **mode** is a measure of central tendency that represents the most frequently occurring value in a set of numbers.
- To find the mode of a set of numbers,
    - We count the frequency of each value and
    - Identify the value with the highest frequency.
    - If two or more values have the same highest frequency, then the set has multiple modes.

Finding Mode

- Suppose we have a set of numbers {3, 1, 6, 2, 7, 3, 3, 2}.
  - We count the frequency of each value:
  - The value 1 occurs once
  - The value 2 occurs twice
  - The value 3 occurs three times
  - The value 6 occurs once
  - The value 7 occurs once
- The mode of the set of numbers is 3.

Mean, Median and Mode


Mode
Median
Mean
Mode is most frequent number
Mode
Median
Mean
Median is middle number in list
Mean is average of all numbers
symmetrical distribution
asymmetrical distribution

- For a symmetrical distribution of dataset, mean=median-mode.
- Otherwise,
  - Mode = Max occuring value
  - Mean = Average
  - Median = Middle-value

Variance and Standard Deviation

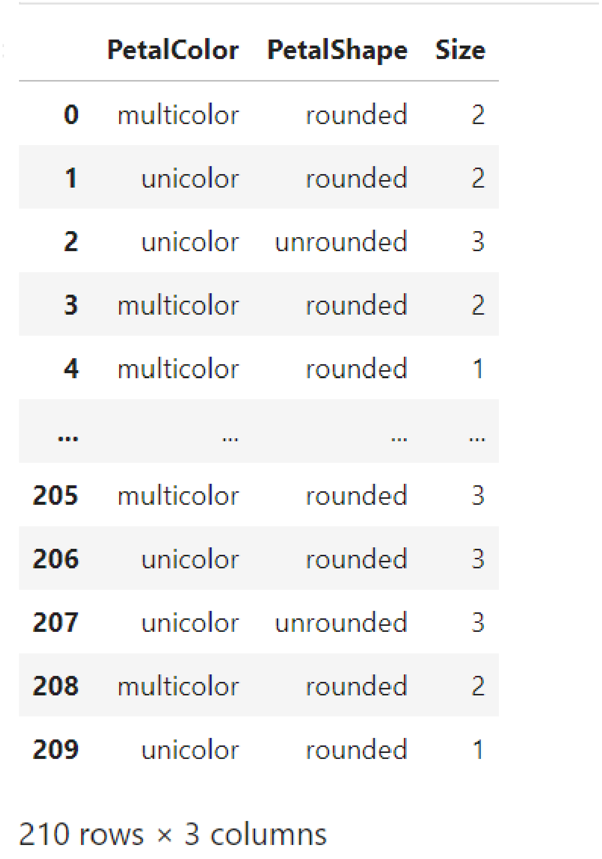
- Statistical measures that describe the spread or variability of a dataset.
- Variance is a measure of how far a set of numbers is spread out.
    - Calculated by taking the average of the squared differences from the mean of the data.
    - Variance measures how much the individual data points deviate from the average of the entire dataset.
- Standard deviation is the square root of the variance.
    - More commonly used measure than variance
    - Expressed in the same units as the original data.

Range

- Range is a statistical measure that represents the difference between the maximum and minimum values in a dataset.
    - It is the spread of the data values in a dataset.
    - Calculated by subtracting the smallest value from the largest value in the dataset.
    - If a dataset contains the values {1, 2, 3, 4, 5},
        - The range would be 5 (the largest value) *minus* 1 (the smallest value), which equals 4.
        - Therefore, the range of this dataset is 4.

Blockly Instructions

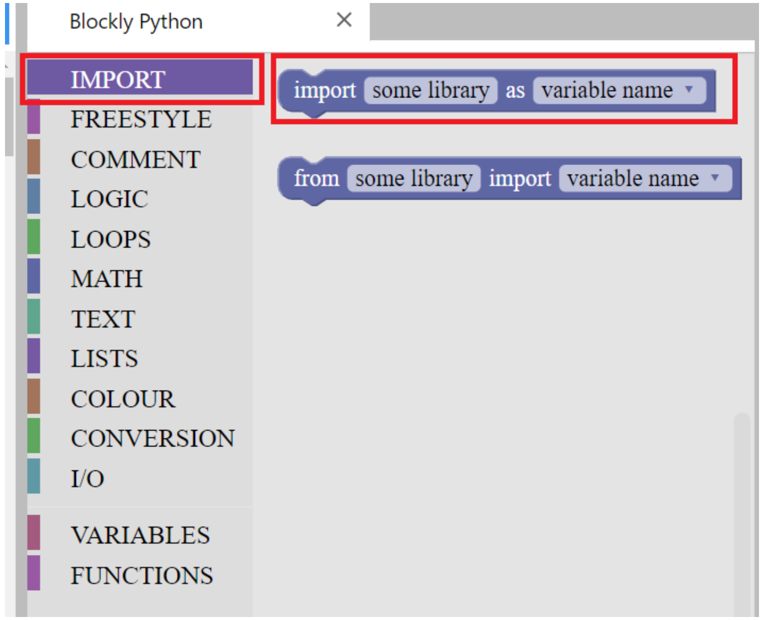
- We will now read the csv *flower-data-2020.csv*
- The csv contains information on petal color, petal shape and size.
- We will find the mean, median, mode of the petal size
- We will use pandas to find the mean median and mode of the petal size from the csv data.



	PetalColor	PetalShape	Size
0	multicolor	rounded	2
1	unicolor	rounded	2
2	unicolor	unrounded	3
3	multicolor	rounded	2
4	multicolor	rounded	1
...	...	...	...
205	multicolor	rounded	3
206	unicolor	rounded	3
207	unicolor	unrounded	3
208	multicolor	rounded	2
209	unicolor	rounded	1

210 rows × 3 columns

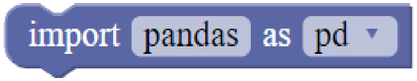
Blockly instruction: Importing pandas

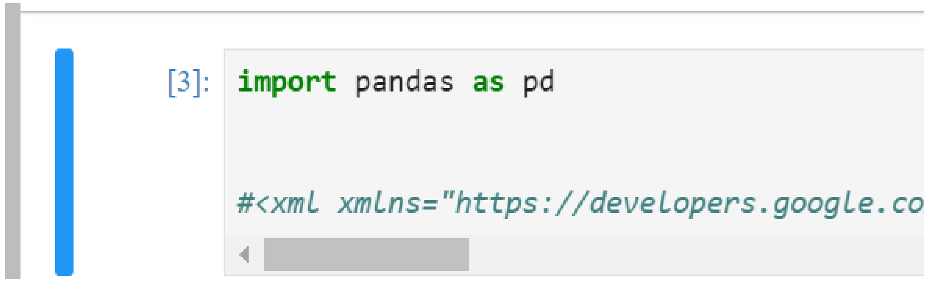


- We first import pandas as pd
- We will use pandas to read the csv and to compute the mean, median and mode

Step 1: Import the library pandas

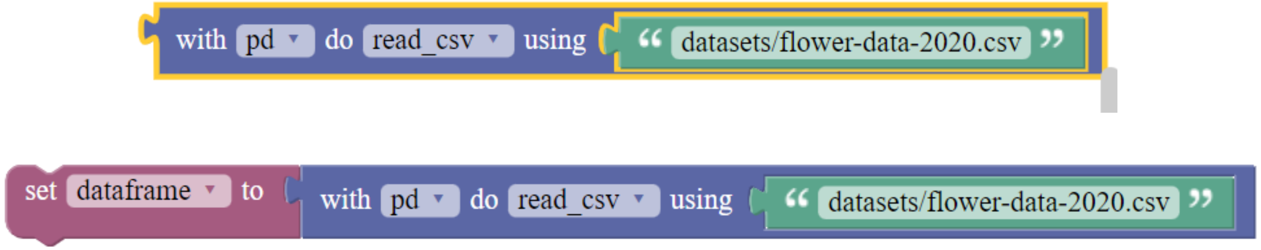
- Go to *import*
- Import pandas as pd

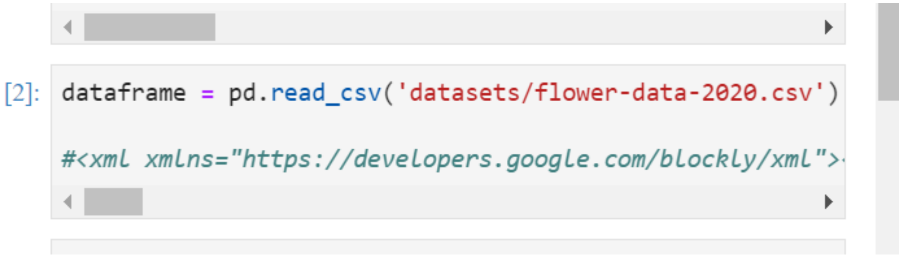




```
[3]: import pandas as pd


     #<xml xmlns="https://developers.google.co
```

Blockly Instructions : Read csv data into dataframe


with pd do read_csv using “ datasets/flower-data-2020.csv ”
set dataframe to with pd do read_csv using “ datasets/flower-data-2020.csv ”



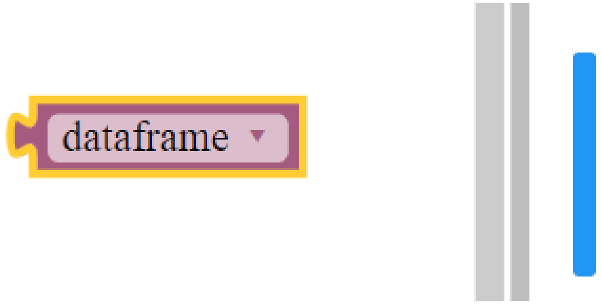
```
[2]: dataframe = pd.read_csv('datasets/flower-data-2020.csv')

     #<xml xmlns="https://developers.google.com/blockly/xml">
```

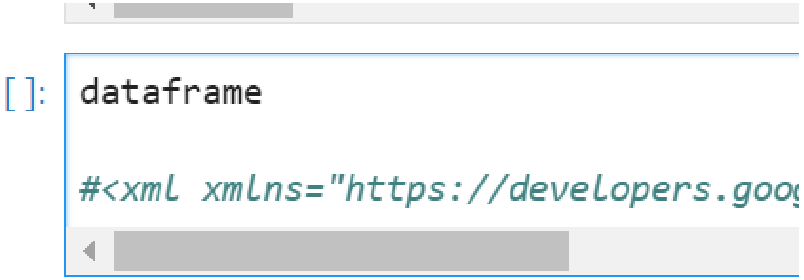
- Next, we read the csv into the variable dataframe

Step 2 : Read csv

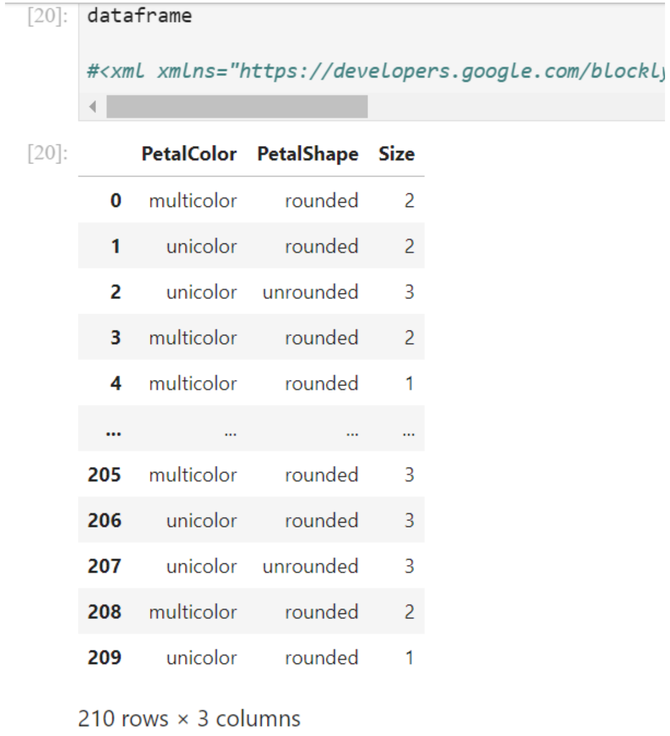
- Go to *variables*
- Select "*with pd ..*" option
- Select *read_csv* from the do options
- Go to *text* and select the empty text option write "datasets/*flower-data-2020 csv*"
- Put the text block inside the *"using …* " option
- Go to *variables* and create a new variable *dataframe*
- Set dataframe to the read csv value by merging the two blocks as shown in the figure

Blockly Instructions: Viewing the dataframe





```
[ ]: dataframe

     #<xml xmlns="https://developers.goog
```



```
[20]: dataframe

      #<xml xmlns="https://developers.google.com/blockly
```

[20]:

	PetalColor	PetalShape	Size
0	multicolor	rounded	2
1	unicolor	rounded	2
2	unicolor	unrounded	3
3	multicolor	rounded	2
4	multicolor	rounded	1
...	...	...	...
205	multicolor	rounded	3
206	unicolor	rounded	3
207	unicolor	unrounded	3
208	multicolor	rounded	2
209	unicolor	rounded	1

210 rows × 3 columns

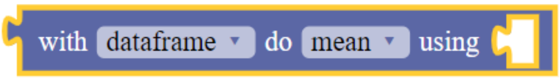
- We now, view the dataframe to know about the data in the csv

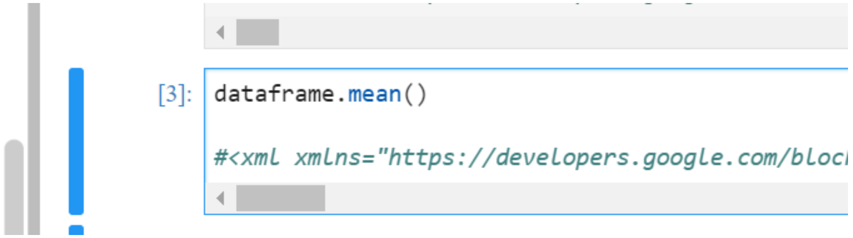
Step 3: View dataframe

- Go to *Variables*
- Select *dataframe*
- Run the cell to view the dataframe

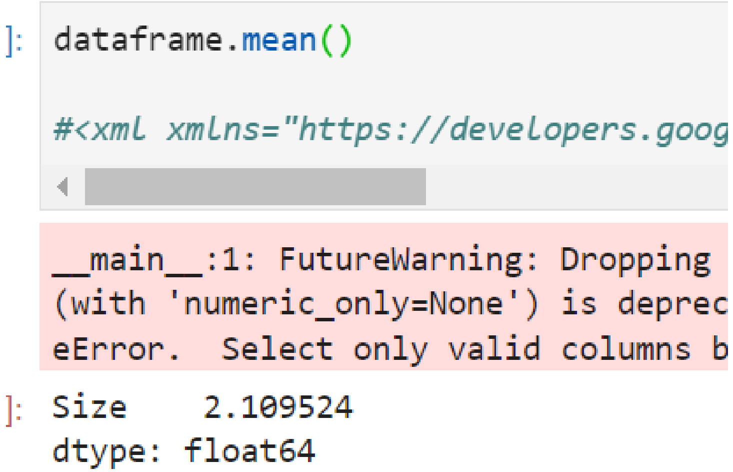
Blockly Instructions:

Calculating the mean of dataframe





```
[3]: dataframe.mean()

     #<xml xmlns="https://developers.google.com/bloc
```



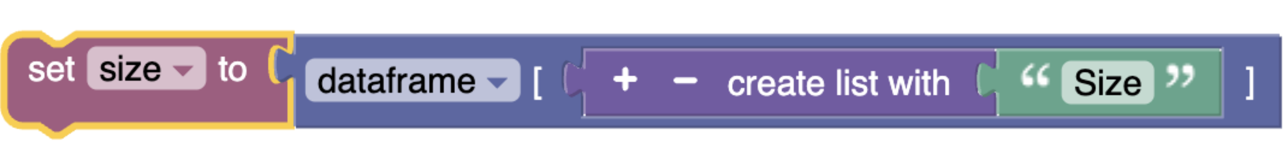
```
]: dataframe.mean()

   #<xml xmlns="https://developers.goog

   __main__:1: FutureWarning: Dropping
   (with 'numeric_only=None') is deprec
   eError.  Select only valid columns b

]: Size    2.109524
   dtype: float64
```

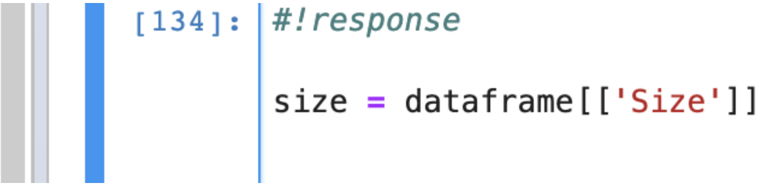
- We now compute the mean of the dataframe. Since dataframe represents the variable which reads the csv using pandas library, we can compute mean by using the mean method provided by pandas
- As only one column i.e the column size is numeric (non-textual), the average of the dataframe is the average for the column size.

Step 5: Get dataframe mean

- Go to *variables*
- Select "with dataframe...." Option
- On the do options select mean
- Run the cell to view the mean

Blockly Instructions:
Get a column in dataframe





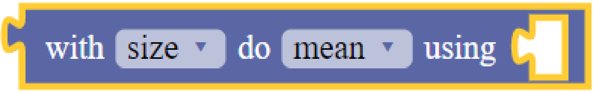
```
[134]: #!response

       size = dataframe[['Size']]
```

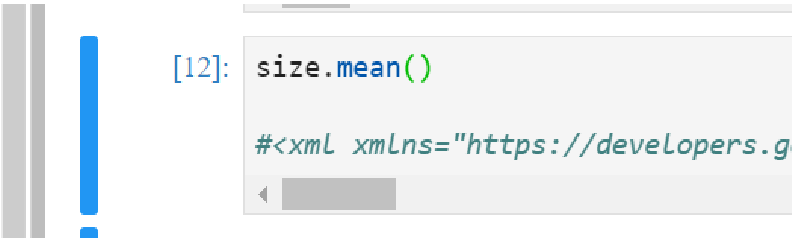
- Now, we store the value of the column **'Size'** of the dataframe as ***size*** variable, so that we can compute the mean, median and mode on this variable

Step 6: Declare variable size

- Go to *variables*
- *From LISTS, select {dictvariable}[..] and enter dataframe.*
- Create list with "Size" and insert inside dataframe[..]
- From variables, create a new variable *size*
- Set *size to the : dataframe[create list with "Size"]* block

Blockly Instructions





```
[12]: size.mean()

#<xml xmlns="https://developers.g
```

- Now we compute the mean of the variable size.
- The variable **size** represents only the column **Size** of the dataframe (the csv we've read)

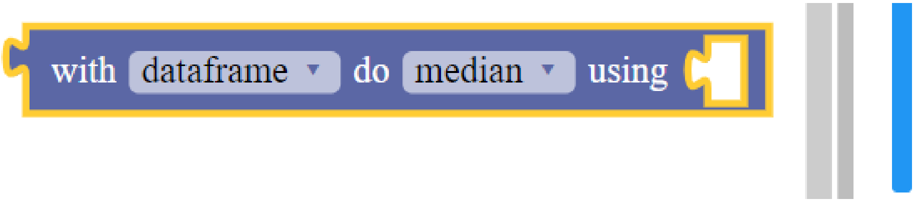
Step 7: Get Mean Size

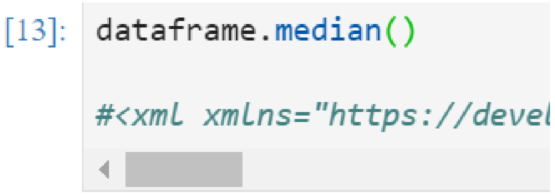
- Go to *variables*
- Select "*with size do...*"
- Select *mean* on the do options

Blockly Instructions:

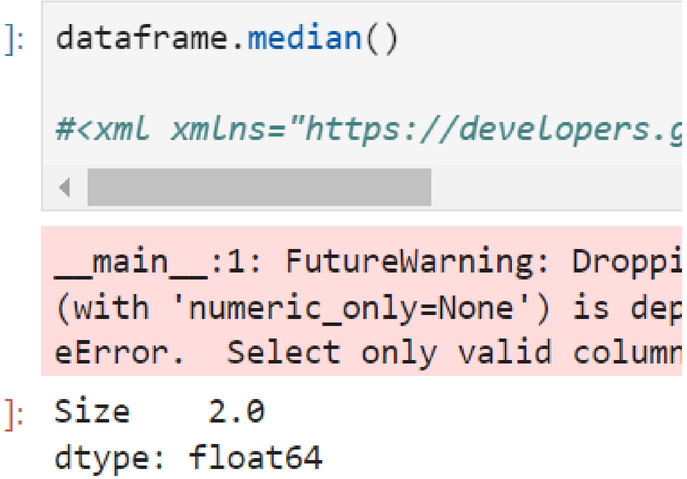
Calculating the median of dataframe

- Now, we compute the median of the dataframe





```
[13]: dataframe.median()

      #<xml xmlns="https://devel
```



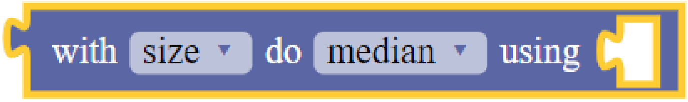
```
]: dataframe.median()

   #<xml xmlns="https://developers.g

   __main__:1: FutureWarning: Droppi
   (with 'numeric_only=None') is dep
   eError.  Select only valid column

]: Size    2.0
   dtype: float64
```

Step 8: Get dataframe Median

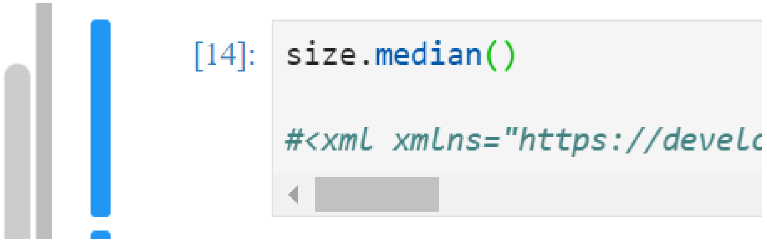
- Go to *variables*
- Select "*with dataframe....*" Option
- On the *do* options select *median*
- Run the cell to view the median

Blockly Instructions:

Calculating Median of column



with size ▾ do median ▾ using



```
[14]: size.median()

      #<xml xmlns="https://develo
```

- Compute the median of the variable *size*

Step 9: Get Median Size

- Go to *variables*
- Select "*with size do...*"
- Select *median* on the *do* options

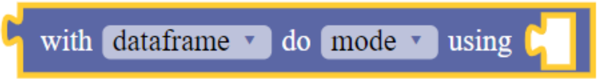
Blockly Instructions :

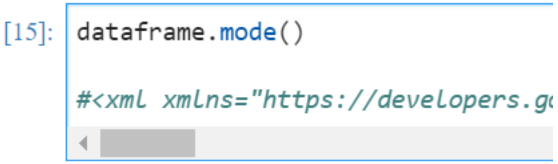
Calculating Mode of dataframe

- Now, we compute the mode of the dataframe

Step 10: Get dataframe Mode

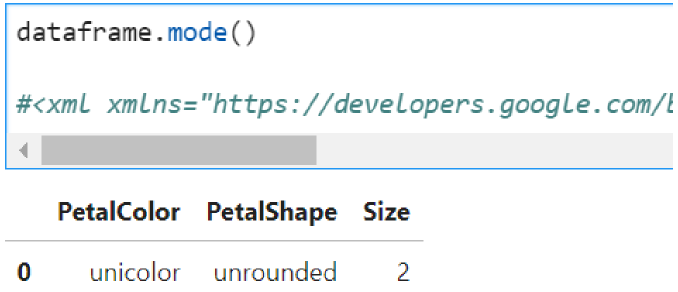
- Go to *variables*
- Select "*with dataframe....*" Option
- On the *do* options select *mode*





[15]:

```
dataframe.mode()

#<xml xmlns="https://developers.g
```

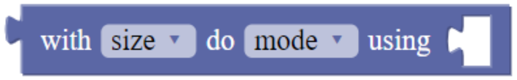


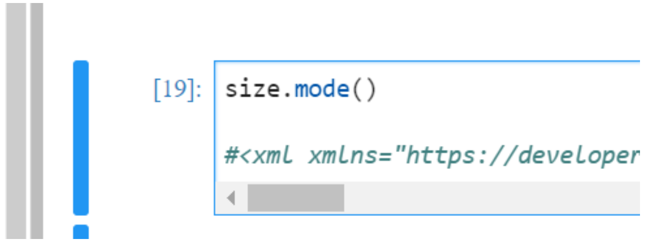
```
dataframe.mode()

#<xml xmlns="https://developers.google.com/
```

	PetalColor	PetalShape	Size
0	unicolor	unrounded	2

Blockly Instructions:

Calculating Mode of a column





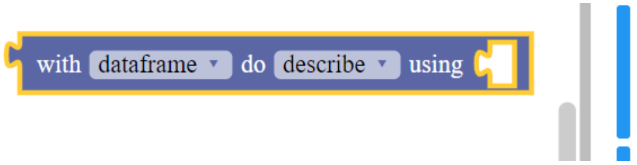
```
[19]: size.mode()

      #<xml xmlns="https://developer
```

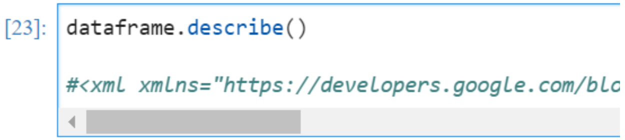
- Now, we compute the mode of the variable size

Step 11: Get Mode of Size

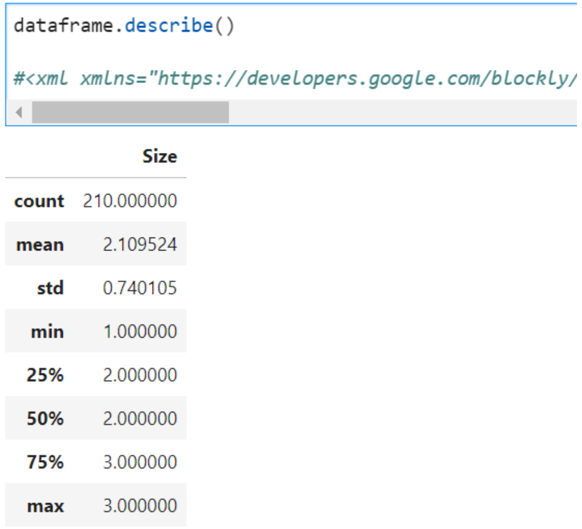
- Go to *variables*
- Select "*with size do...*"
- Select *mode* on the do options

Blockly Instructions:
Know the data spread




[23]:

```
dataframe.describe()

#<xml xmlns="https://developers.google.com/blc
```



```
dataframe.describe()

#<xml xmlns="https://developers.google.com/blockly/
```

	Size
count	210.000000
mean	2.109524
std	0.740105
min	1.000000
25%	2.000000
50%	2.000000
75%	3.000000
max	3.000000

- Now, we use the built-in function of the pandas library describe, which provides an insight on the descriptive statistics of the csv file
- Since we store the csv file, we've read using pandas in the dataframe variable, we can use the pandas utility directly using the variable dataframe

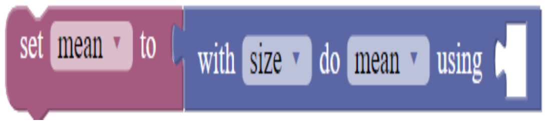
Step 12: Describe Dataframe

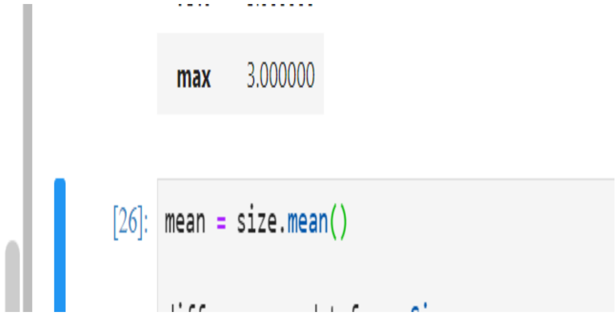
- Go to *variables*
- Select "*with size do...*"
- Select *describe* on the *do* options
- Run the cell to view the descriptive statistics

Blockly Instructions:

Calculate standard deviation

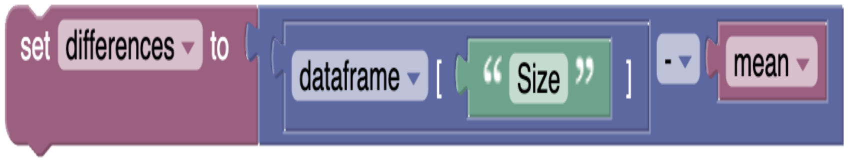
- Now, we compute the Standard Seviation (S.D) of the size of the petals, to compute the S.D we:
  - Calculate the mean.
  - Find the difference from the mean for each data point.
  - Fine the square of the differences.
  - Calculate the mean of the squared differences.
  - Take the square root to get the standard deviation.

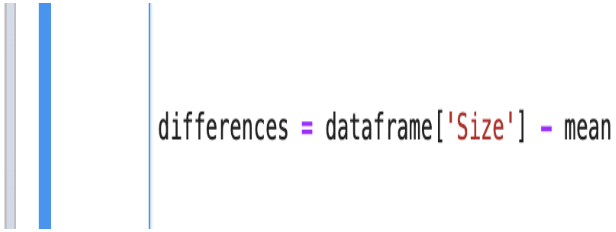




```
max    3.000000
```

```
[26]: mean = size.mean()
```



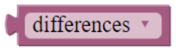


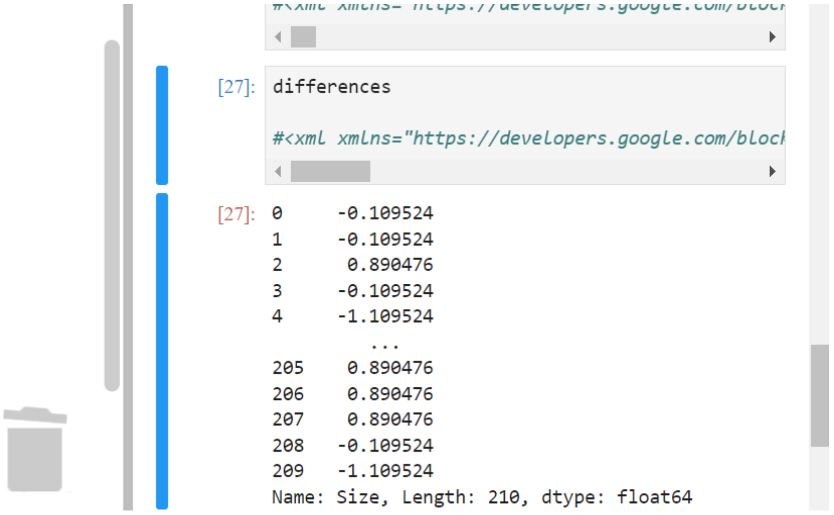
```
differences = dataframe['Size'] - mean
```

Step 13: Compute S.D

- **Substep: Compute the mean**
  - Go to *variables*
  - Create a new *variable* mean
  - Go to variables and select the option "*with size do..*"
  - On the *do* option select *mean*
  - Combine the block to set the value of *mean* as mean size
- **Substep: Compute the difference from mean for each data point**
  - Create a new variable *difference*
  - Got to variables and select the option "*from dataframe get ..*"
  - On the *get* option select *Size*
  - Combine the block to set the value of variable *difference* as the difference of dataframse.Size and mean

Blockly Instructions

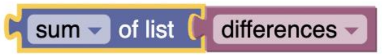


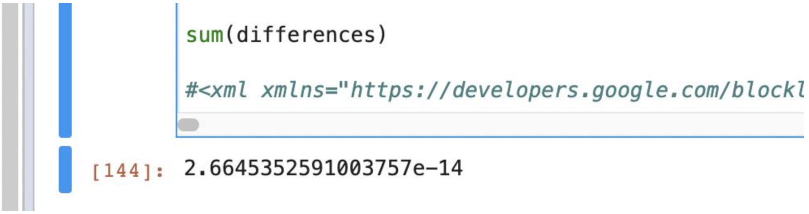


```
#<xml xmlns="https://developers.google.com/block
```

```
[27]: differences

      #<xml xmlns="https://developers.google.com/block
```

```
[27]: 0      -0.109524
      1      -0.109524
      2       0.890476
      3      -0.109524
      4      -1.109524
                ...
      205     0.890476
      206     0.890476
      207     0.890476
      208    -0.109524
      209    -1.109524
      Name: Size, Length: 210, dtype: float64
```





```
sum(differences)

#<xml xmlns="https://developers.google.com/blockl
```

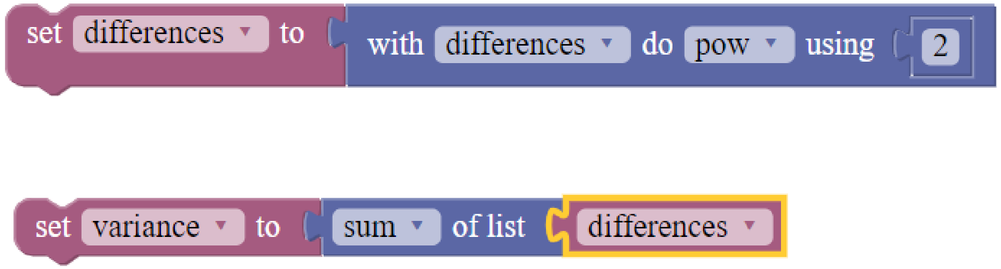
```
[144]: 2.6645352591003757e-14
```

Step 13 continued ….

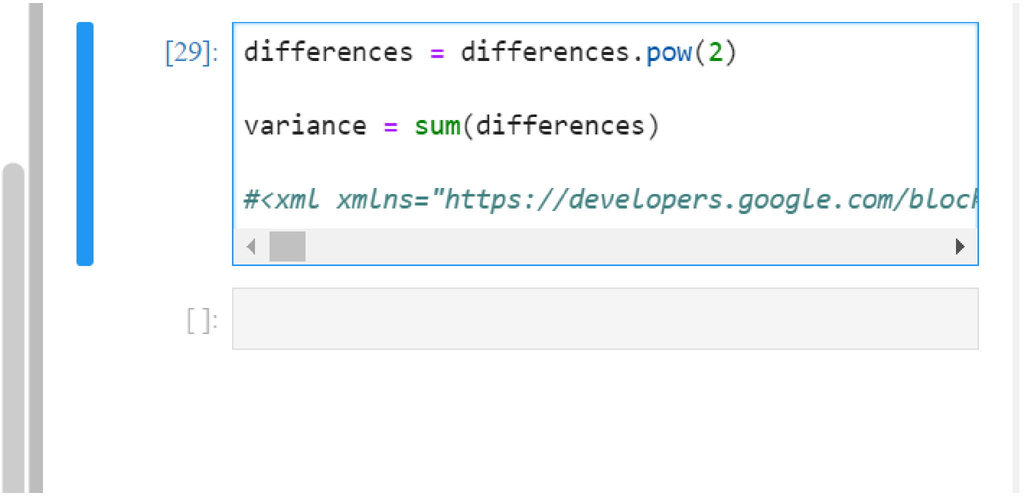
- **Substep: View Difference and get the sum of difference**
  - View the variable *difference*
  - From math select the "*sum of list*" option
  - On the *list* option put the variable *difference*
  - View the sum of difference

Blockly Instructions:

Calculating Variance



```
set differences to with differences do pow using 2

set variance to sum of list differences
```



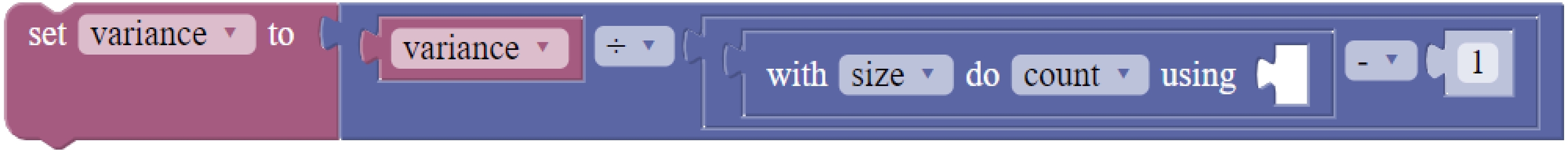
```
[29]: differences = differences.pow(2)

      variance = sum(differences)

      #<xml xmlns="https://developers.google.com/block
```

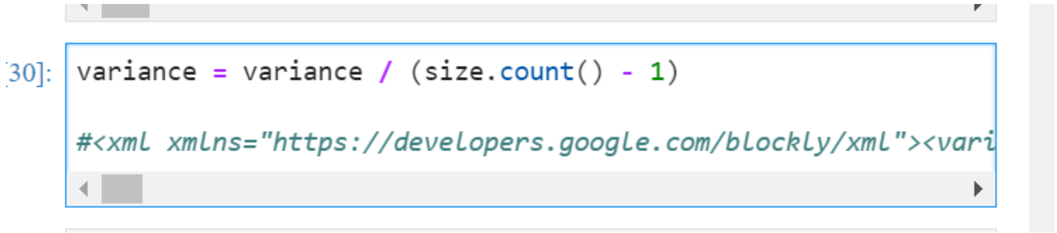
```
[ ]:
```

Step 13 continued…

- **Substep :Find the sum of squares of the difference**
  - On *Variable block*, select "*with difference do..*" option, on the *do* option select pow , on the *using* option **insert** a new *math* block. Set the value to 2.
  - Set the value of the variable difference to the value of the block by combining two blocks
  - Create a new variable *variance*, set it's value to sum of difference

Blockly Instructions





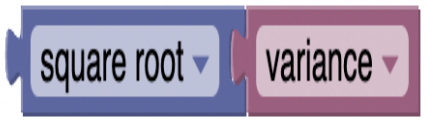
```
[30]: variance = variance / (size.count() - 1)

#<xml xmlns="https://developers.google.com/blockly/xml"><vari
```

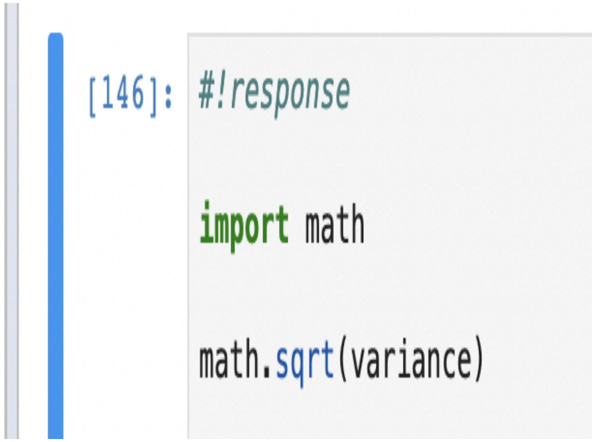
Step 13 continued ...

- **Substep: Divide Variance by size**
  - Select *math* option to *divide* .
  - Set variance on numerator of division.
  - On denominator, get the size of count from the *variable* block with "*get size using do..*" Option. Further use *math* block to substract 1 from the size
  - Set *variance* to the value of the block

Blockly Instructions:

Get standard deviation from variance





```
[146]: #!response

       import math

       math.sqrt(variance)
```

Step 13 continued….

- **Substep: Find the Standard Deviation**

  - To compute the square root, go to MATH block and select square root option.

  - From VARIABLES, select the variance block and connect with square root