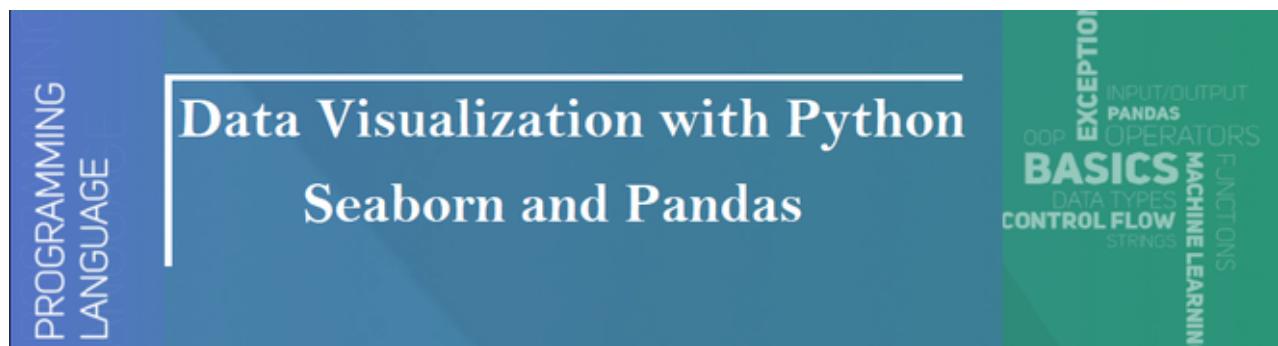




Data Visualization with Python Seaborn

Data Visualization is the presentation of data in pictorial format. It is extremely important for Data Analysis, primarily because of the fantastic ecosystem of data-centric Python packages. And it helps to understand the data, however, complex it is, the significance of data by summarizing and presenting a huge amount of data in a simple and easy-to-understand format and helps communicate information clearly and effectively.



Pandas and Seaborn is one of those packages and makes importing and analyzing data much easier. In this article, we will use Pandas and Seaborn to analyze data.

Pandas

Pandas offer tools for cleaning and process your data. It is the most popular Python library that is used for data analysis. In pandas, a data table is called a dataframe.



So, let's start with creating Pandas data frame:

Example 1:

Python3

```
# Python code demonstrate creating

import pandas as pd

# initialise data of lists.
data = {'Name':[ 'Mohe' , 'Karnal' , 'Yrik' , 'jack' ],
        'Age':[ 30 , 21 , 29 , 28 ]}

# Create DataFrame
df = pd.DataFrame( data )

# Print the output.
df
```



Output:

	Name	Age
0	Mohe	30
1	Karnal	21
2	Yrik	29
3	jack	28

Example 2: load the [CSV](#) data from the system and display it through pandas.

Python3

```
# import module
import pandas

# load the csv
data = pandas.read_csv("nba.csv")

# show first 5 column
data.head()
```

Output:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99.0	SF	25.0	6-6	235.0	Marquette	6796117.0
2	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28.0	SG	22.0	6-5	185.0	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0

Seaborn

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It is built on the top of [matplotlib](#) library and also closely integrated into the data structures from [pandas](#).

Installation

For python environment :

```
pip install seaborn
```

For conda environment :

```
conda install seaborn
```

Let's create Some basic plots using seaborn:

Python3

```

# Importing libraries
import numpy as np
import seaborn as sns

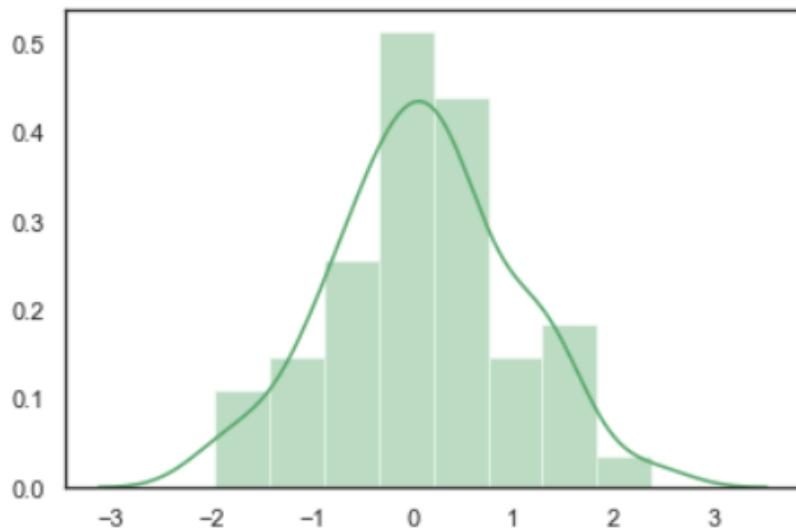
# Selecting style as white,
# dark, whitegrid, darkgrid
# or ticks
sns.set( style = "white" )

# Generate a random univariate
# dataset
rs = np.random.RandomState( 10 )
d = rs.normal( size = 50 )

# Plot a simple histogram and kde
# with binsize determined automatically
sns.distplot(d, kde = True, color = "g")

```

Output:



Seaborn: statistical data visualization

Seaborn helps to visualize the statistical relationships. To understand how variables in a dataset are related to one another and how that relationship is dependent on other variables, we perform statistical analysis. This Statistical analysis helps to visualize the trends and identify various patterns in the dataset.

These are the plots that will help to visualize:

- Line Plot
- Scatter Plot
- Box plot
- Point plot
- Count plot
- Violin plot

- Swarm plot
- Bar plot
- KDE Plot

Line plot:

Lineplot Is the most popular plot to draw a relationship between x and y with the possibility of several semantic groupings.

Syntax : `sns.lineplot(x=None, y=None)`

Parameters:

x, y: Input data variables; must be numeric. Can pass data directly or reference columns in data.

Let's visualize the data with a line plot and pandas:

Example 1:

Python3

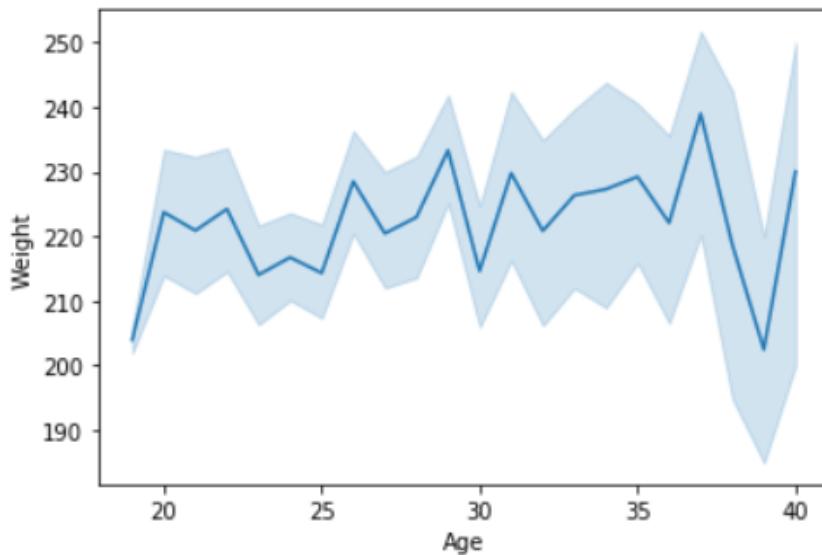
```
# import module
import seaborn as sns
import pandas

# loading csv
data = pandas.read_csv("nba.csv")

# plotting lineplot
sns.lineplot( data['Age'], data['Weight'])
```



Output:



Example 2: Use the hue parameter for plotting the graph.

Python3

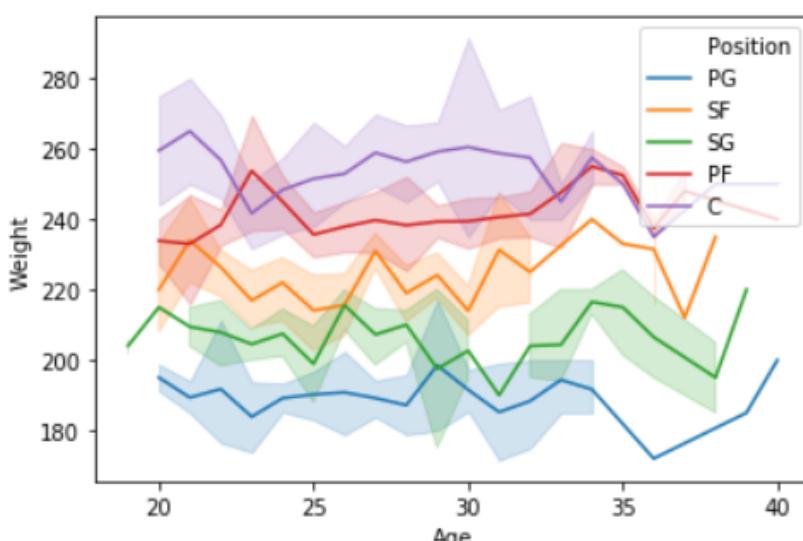
```
# import module
import seaborn as sns
import pandas

# read the csv data
data = pandas.read_csv("nba.csv")

# plot
sns.lineplot(data['Age'], data['Weight'], hue = data["Position"])
```



Output:



Scatter Plot:

Scatterplot Can be used with several semantic groupings which can help to understand well in a graph against continuous/categorical data. It can draw a two-dimensional graph.

Syntax: seaborn.scatterplot(x=None, y=None)

Parameters:

x, y: Input data variables that should be numeric.

Returns: This method returns the Axes object with the plot drawn onto it.

Let's visualize the data with a scatter plot and pandas:

Example 1:

Python3

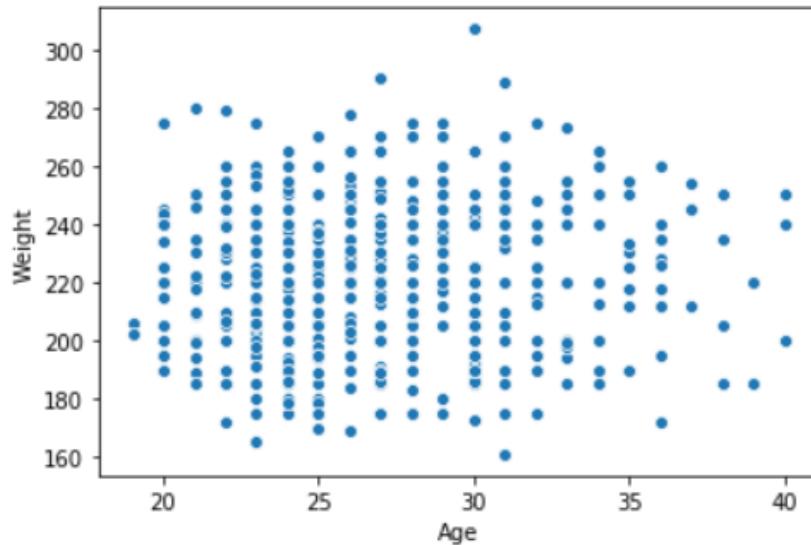
```
# import module
import seaborn
import pandas

# load csv
data = pandas.read_csv("nba.csv")

# plotting
seaborn.scatterplot(data['Age'], data['Weight'])
```



Output:



Example 2: Use the hue parameter for plotting the graph.

Python3

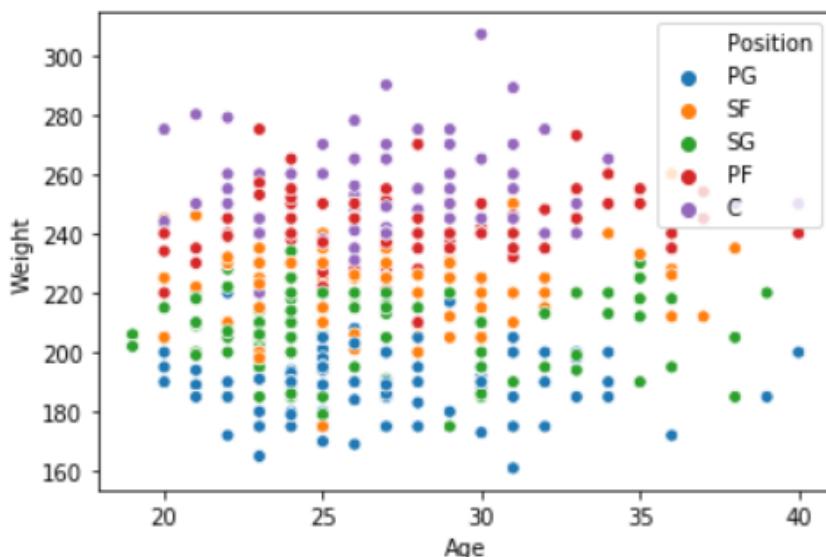
```

import seaborn
import pandas
data = pandas.read_csv("nba.csv")

seaborn.scatterplot( data['Age'], data['Weight'], hue =data["Position"])

```

Output:



Box plot:

A [box plot](#) (or box-and-whisker plot) is the visual representation of the depicting groups of numerical data through their quartiles against continuous/categorical data.

A box plot consists of 5 things.

- Minimum
- First Quartile or 25%
- Median (Second Quartile) or 50%
- Third Quartile or 75%
- Maximum

Syntax:

```
seaborn.boxplot(x=None, y=None, hue=None, data=None)
```

Parameters:

- **x, y, hue:** Inputs for plotting long-form data.
- **data:** Dataset for plotting. If x and y are absent, this is interpreted as wide-form.

Returns: It returns the Axes object with the plot drawn onto it.

Draw the box plot with Pandas:

Example 1:

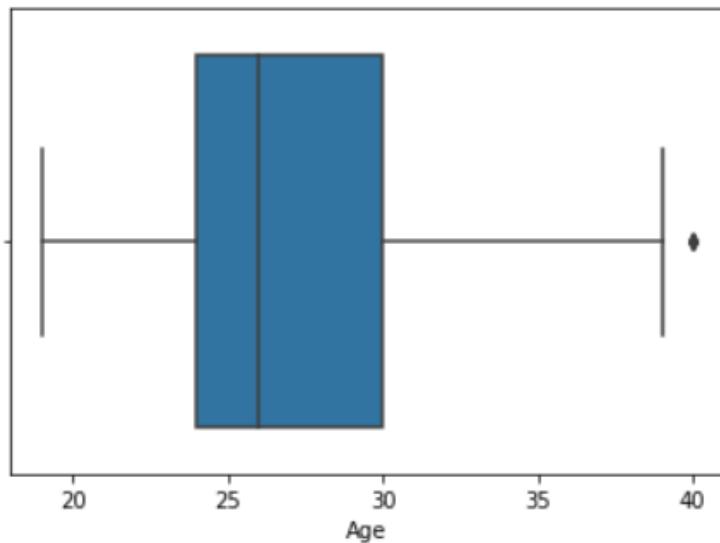
Python3

```
# import module
import seaborn as sns
import pandas

# read csv and plotting
data = pandas.read_csv( "nba.csv" )
sns.boxplot( data['Age'] )
```



Output:



Example 2:

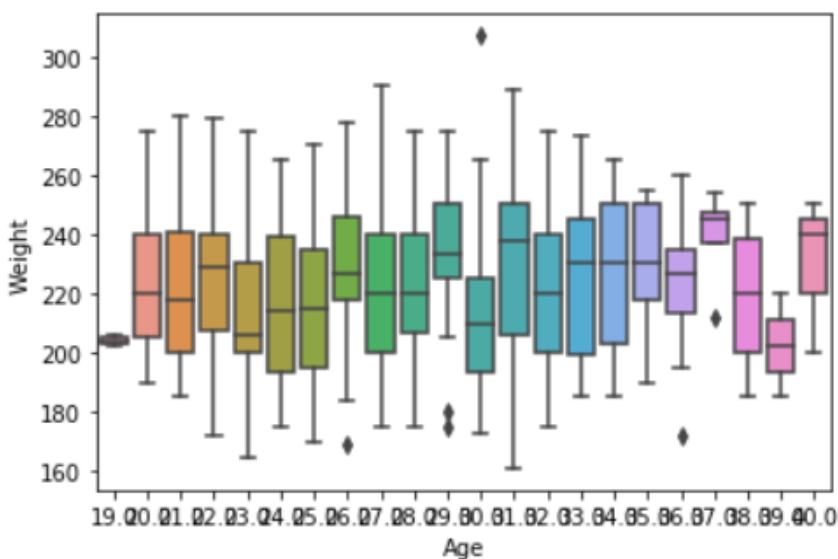
Python3

```
# import module
import seaborn as sns
import pandas

# read csv and plotting
data = pandas.read_csv( "nba.csv" )
sns.boxplot( data['Age'], data['Weight'] )
```



Output:



Violin Plot:

A violin plot is similar to a boxplot. It shows several quantitative data across one or more categorical variables such that those distributions can be compared.

Syntax: `seaborn.violinplot(x=None, y=None, hue=None, data=None)`

Parameters:

- **x, y, hue:** Inputs for plotting long-form data.
- **data:** Dataset for plotting.

Draw the violin plot with Pandas:

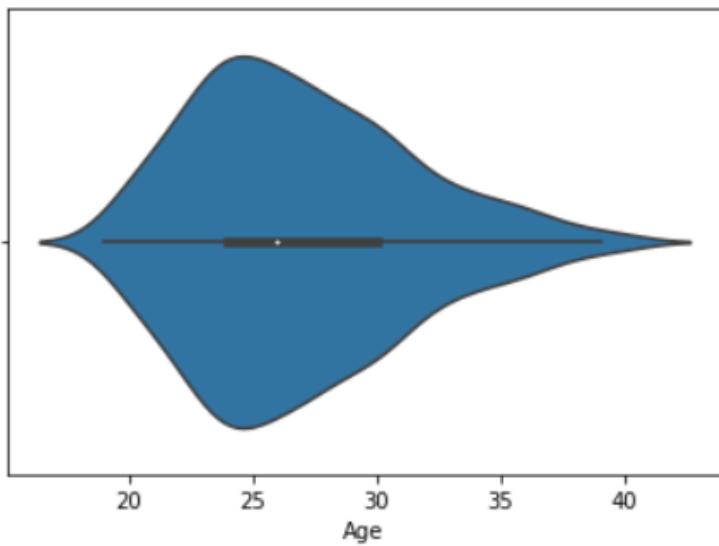
Example 1:

Python3

```
# import module
import seaborn as sns
import pandas

# read csv and plot
data = pandas.read_csv("nba.csv")
sns.violinplot(data['Age'])
```

Output:



Example 2:

Python3

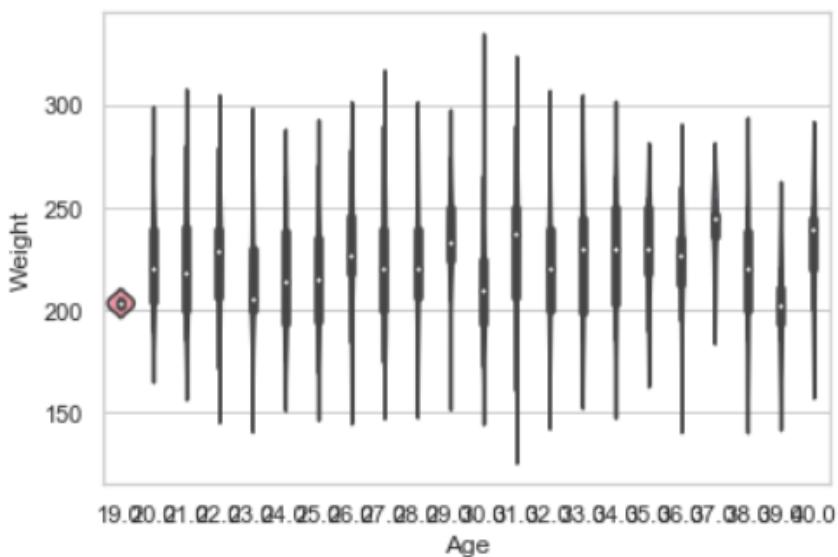
```
# import module
import seaborn

seaborn.set(style = 'whitegrid')

# read csv and plot
data = pandas.read_csv("nba.csv")
seaborn.violinplot(x ="Age", y ="Weight", data = data)
```



Output:



Swarm plot:

A swarm plot is similar to a strip plot. We can draw a swarm plot with non-overlapping points against categorical data.

Syntax: seaborn.swarmplot(x=None, y=None, hue=None, data=None)

Parameters:

- **x, y, hue:** Inputs for plotting long-form data.
- **data:** Dataset for plotting.

Draw the swarm plot with Pandas:

Example 1:

Python3

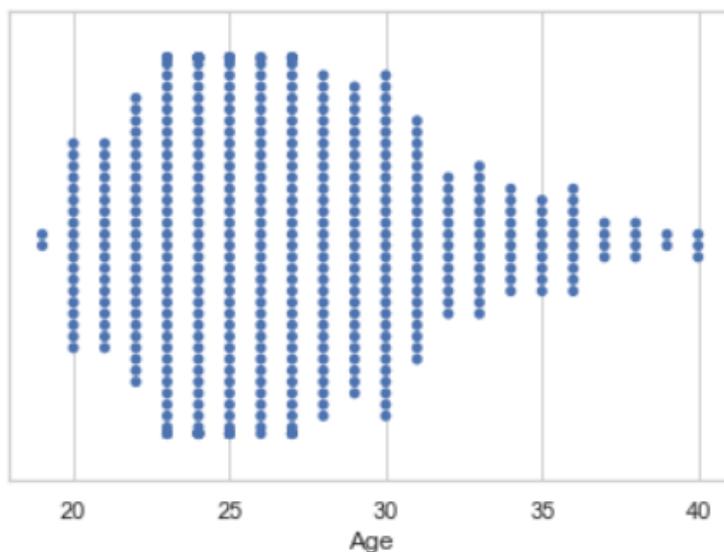
```
# import module
import seaborn

seaborn.set(style = 'whitegrid')

# read csv and plot
data = pandas.read_csv( "nba.csv" )
seaborn.swarmplot(x = data["Age"])
```



Output:



Example 2:

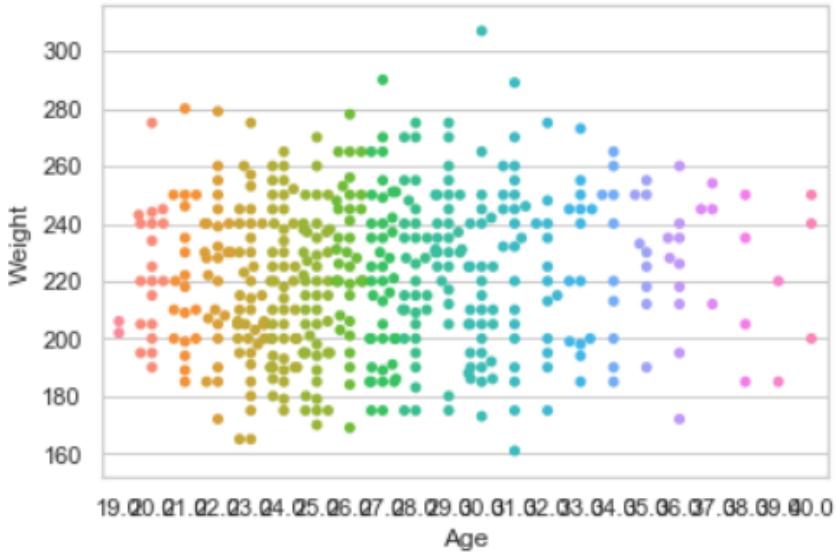
Python3

```
# import module
import seaborn
```

```
seaborn.set(style = 'whitegrid')

# read csv and plot
data = pandas.read_csv("nba.csv")
seaborn.swarmplot(x ="Age", y ="Weight", data = data)
```

Output:



Bar plot:

[Barplot](#) represents an estimate of central tendency for a numeric variable with the height of each rectangle and provides some indication of the uncertainty around that estimate using error bars.

Syntax : `seaborn.barplot(x=None, y=None, hue=None, data=None)`

Parameters :

- **x, y :** This parameter take names of variables in data or vector data, Inputs for plotting long-form data.
- **hue :** (optional) This parameter take column name for colour encoding.
- **data :** (optional) This parameter take DataFrame, array, or list of arrays, Dataset for plotting. If x and y are absent, this is interpreted as wide-form. Otherwise it is expected to be long-form.

Returns : Returns the Axes object with the plot drawn onto it.

Draw the bar plot with Pandas:

Example 1:

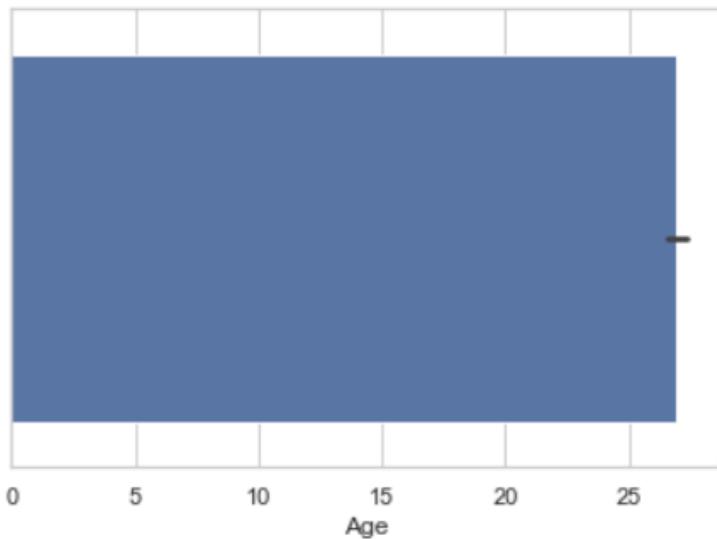
Python3

```
# import module
import seaborn

seaborn.set(style = 'whitegrid')

# read csv and plot
data = pandas.read_csv("nba.csv")
seaborn.barplot(x = data["Age"])
```

Output:



Example 2:

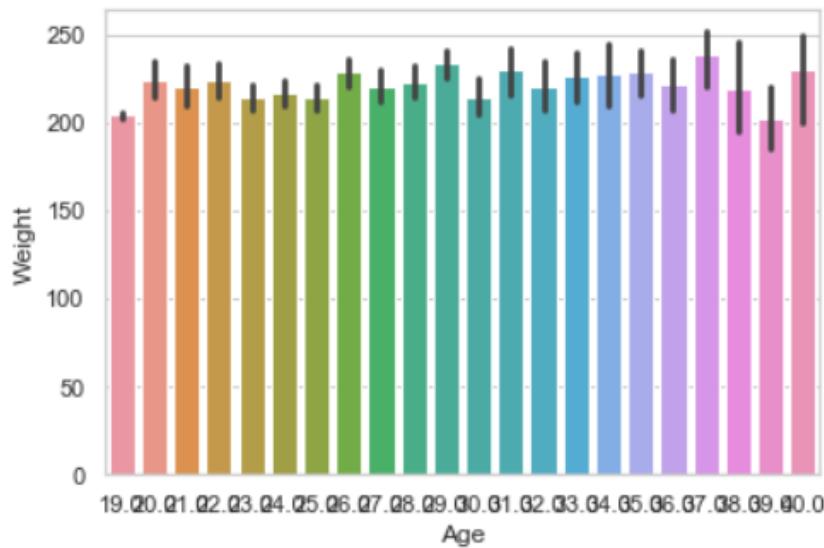
Python3

```
# import module
import seaborn

seaborn.set(style = 'whitegrid')

# read csv and plot
data = pandas.read_csv("nba.csv")
seaborn.barplot(x ="Age", y ="Weight", data = data)
```

Output:



Point plot:

Point plot used to show point estimates and confidence intervals using scatter plot glyphs. A point plot represents an estimate of central tendency for a numeric variable by the position of scatter plot points and provides some indication of the uncertainty around that estimate using error bars.

Syntax: `seaborn.pointplot(x=None, y=None, hue=None, data=None)`

Parameters:

- **x, y:** Inputs for plotting long-form data.
- **hue:** (optional) column name for color encoding.
- **data:** dataframe as a Dataset for plotting.

Return: The Axes object with the plot drawn onto it.

Draw the point plot with Pandas:

Example:

Python3

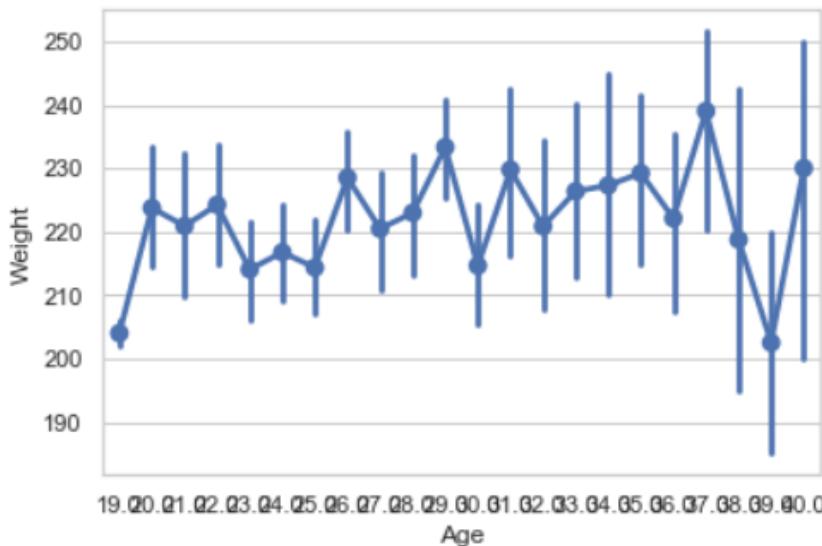
```
# import module
import seaborn

seaborn.set(style = 'whitegrid')

# read csv and plot
data = pandas.read_csv("nba.csv")
seaborn.pointplot(x = "Age", y = "Weight", data = data)
```



Output:



Count plot:

Count plot used to Show the counts of observations in each categorical bin using bars.

Syntax : `seaborn.countplot(x=None, y=None, hue=None, data=None)`

Parameters :

- **x, y:** This parameter take names of variables in data or vector data, optional, Inputs for plotting long-form data.
- **hue :** (optional) This parameter take column name for color encoding.
- **data :** (optional) This parameter take DataFrame, array, or list of arrays, Dataset for plotting. If x and y are absent, this is interpreted as wide-form. Otherwise, it is expected to be long-form.

Returns: Returns the Axes object with the plot drawn onto it.

Draw the count plot with Pandas:

Example:

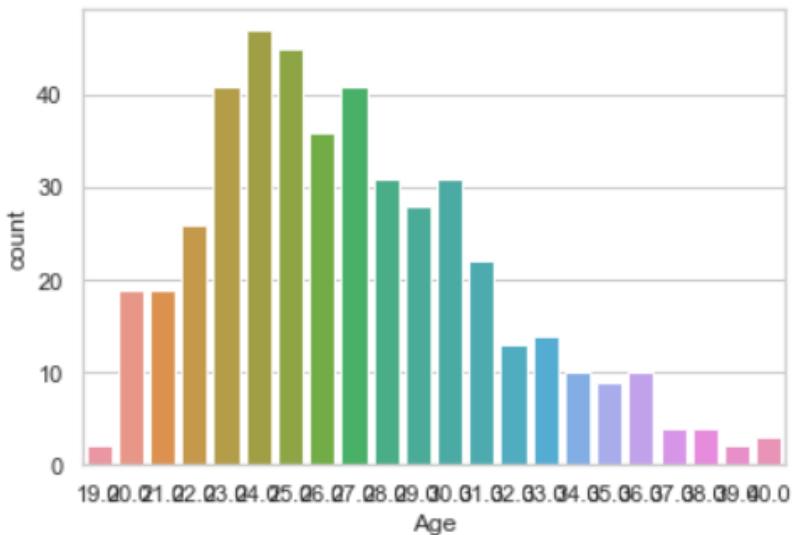
Python3

```
# import module
import seaborn

seaborn.set(style = 'whitegrid')
```

```
# read csv and plot  
data = pandas.read_csv("nba.csv")  
seaborn.countplot(data["Age"])
```

Output:



KDE Plot:

KDE Plot described as **Kernel Density Estimate** is used for visualizing the Probability Density of a continuous variable. It depicts the probability density at different values in a continuous variable. We can also plot a single graph for multiple samples which helps in more efficient data visualization.

Syntax: `seaborn.kdeplot(x=None, *, y=None, vertical=False, palette=None, **kwargs)`

Parameters:

x, y : vectors or keys in data

vertical : boolean (True or False)

data : pandas.DataFrame, numpy.ndarray, mapping, or sequence

Draw the KDE plot with Pandas:

Example 1:

Python3

```
# importing the required libraries  
from sklearn import datasets  
import pandas as pd
```

```

import seaborn as sns

# Setting up the Data Frame
iris = datasets.load_iris()

iris_df = pd.DataFrame(iris.data, columns=['Sepal_Length',
                                             'Sepal_Width', 'Petal_Length', 'Petal_Width'])

iris_df['Target'] = iris.target

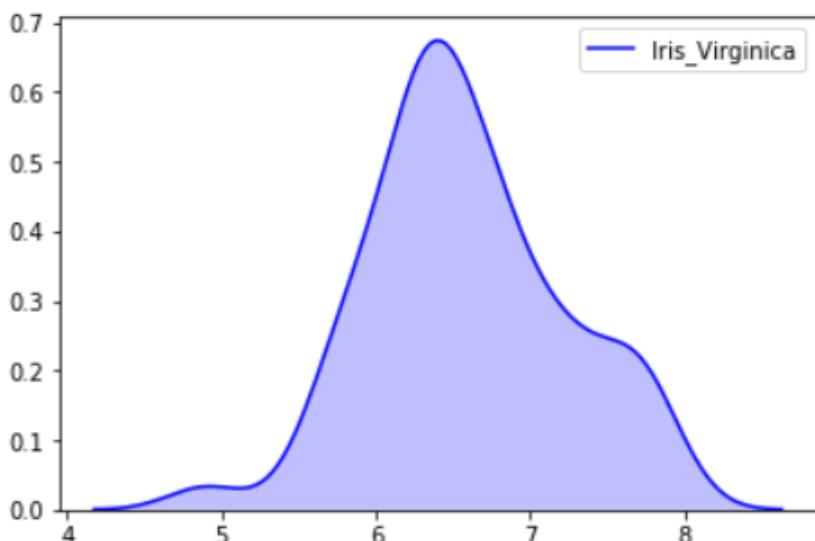
iris_df['Target'].replace([0], 'Iris_Setosa', inplace=True)
iris_df['Target'].replace([1], 'Iris_Vercicolor', inplace=True)
iris_df['Target'].replace([2], 'Iris_Virginica', inplace=True)

# Plotting the KDE Plot
sns.kdeplot(iris_df.loc[(iris_df['Target'] == 'Iris_Virginica'),
                         'Sepal_Length'], color = 'b', shade = True, Label ='Iris_Virginica')

```



Output:



Example 2:

Python3

```

# import module
import seaborn as sns
import pandas

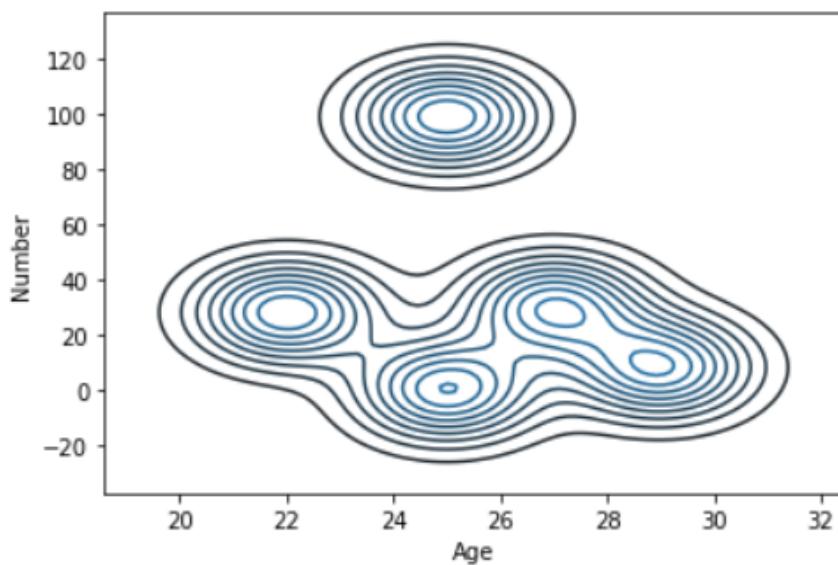
# read top 5 column
data = pandas.read_csv("nba.csv").head()

sns.kdeplot( data['Age'], data['Number'])

```



Output:



Bivariate and Univariate data using seaborn and pandas:

Before starting let's have a small intro of bivariate and univariate data:

Bivariate data: This type of data involves **two different variables**. The analysis of this type of data deals with causes and relationships and the analysis is done to find out the relationship between the two variables.

Univariate data: This type of data consists of **only one variable**. The analysis of univariate data is thus the simplest form of analysis since the information deals with only one quantity that changes. It does not deal with causes or relationships and the main purpose of the analysis is to describe the data and find patterns that exist within it.

Let's see an example of Bivariate data :

Example 1: Using the box plot.

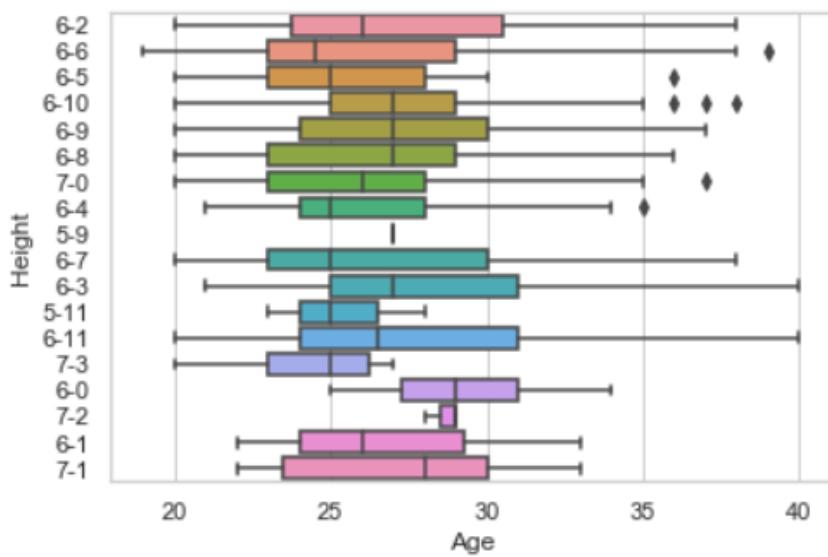
Python3

```
# import module
import seaborn as sns
import pandas

# read csv and plotting
data = pandas.read_csv( "nba.csv" )
sns.boxplot( data['Age'], data['Height'])
```



Output:



Example 2: using KDE plot.

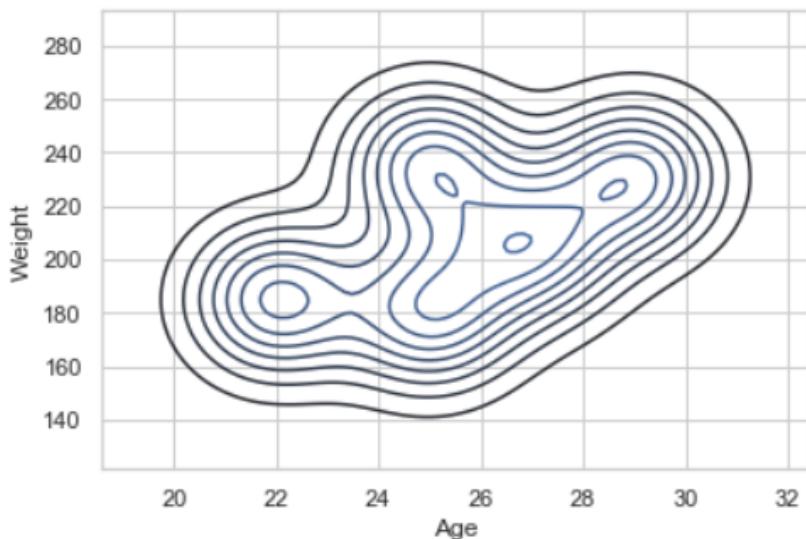
Python3

```
# import module
import seaborn as sns
import pandas

# read top 5 column
data = pandas.read_csv("nba.csv").head()

sns.kdeplot( data['Age'], data['Weight'])
```

Output:



Let's see an example of univariate data distribution:

Example: Using the dist plot

Python3

```
# import module
import seaborn as sns
```

```
import pandas
```

```
# read top 5 column
```

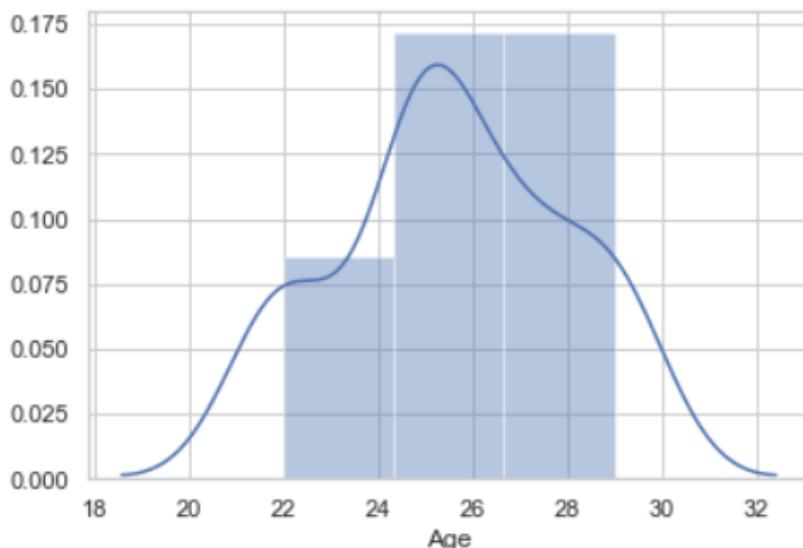
Read

Discuss

Courses

Practice

Output



Whether you're preparing for your first job interview or aiming to upskill in this ever-evolving tech landscape, [GeeksforGeeks Courses](#) are your key to success. We provide top-quality content at affordable prices, all geared towards accelerating your growth in a time-bound manner. Join the millions we've already empowered, and we're here to do the same for you. Don't miss out - [check it out now!](#)

Last Updated : 15 Jan, 2022

14

Previous

Next

Python - Data visualization tutorial

Python - Data visualization tutorial

Similar Reads

Data Visualization with Seaborn Line Plot

Box plot visualization with Pandas and Seaborn

Difference Between Data Science and Data Visualization

Conditional Data Visualization Using Google Data Studio

Data Visualisation in Python using Matplotlib and Seaborn

Data visualization with Pairplot Seaborn and Pandas

KDE Plot Visualization with Pandas and Seaborn

Why is Data Visualization so Important in Data Science?

Python | Titanic Data EDA using Seaborn

How to Make Boxplots with Data Points using Seaborn in Python?

Complete Tutorials

OpenAI Python API - Complete Guide

Python for Kids - Fun Tutorial to Learn Python Programming

Flask Tutorial

Pandas AI: The Generative AI Python Library

Data Analysis Tutorial

Article Contributed By :



kumar_satyam

kumar_satyam

Vote for difficulty

Current difficulty : [Hard](#)

Easy

Normal

Medium

Hard

Expert

Article Tags : Data Visualization , Python-pandas , Python-Seaborn ,
Technical Scripter 2020 , Data Science , Python , Technical Scripter

Practice Tags : python

[Improve Article](#)

[Report Issue](#)



A-143, 9th Floor, Sovereign Corporate Tower, Sector-136, Noida, Uttar Pradesh - 201305



Company	Explore	Languages	DSA	Data Science & ML	HTML & CSS
About Us	Job-A-Thon	Python	Data Structures		HTML
Legal	Hiring Challenge	Java	Algorithms	Data Science	CSS
Careers	Hack-A-Thon	C++	DSA for Beginners	With Python	Bootstrap
In Media	GfG Weekly Contest	PHP	Beginners	Data Science For Beginner	Tailwind CSS
Contact Us		GoLang	Basic DSA Problems	Machine Learning Tutorial	SASS
Advertise with us	Offline Classes (Delhi/NCR)	SQL	DSA Roadmap		LESS
GFG Corporate Solution	DSA in JAVA/C++	R Language	Top 100 DSA Interview Problems	Maths For Machine Learning	Web Design
Placement Training Program	Master System Design	Android Tutorial	DSA Roadmap by Sandeep Jain	Pandas Tutorial	

Apply for Mentor	GeeksforGeeks Videos	All Cheat Sheets	NumPy Tutorial Deep Learning Tutorial		
Computer Science	Python	DevOps	Competitive Programming	System Design	JavaScript
GATE CS Notes	Python Programming Examples	Git AWS	Top DS or Algo for CP	What is System Design	TypeScript ReactJS
Operating Systems	Django Tutorial	Docker Kubernetes	Top 50 Tree	Monolithic and Distributed SD	NextJS AngularJS
Computer Network	Python Projects Python Tkinter	Azure GCP	Top 50 Graph Top 50 Array	High Level Design or HLD	NodeJS Express.js
Database Management System	OpenCV Python Tutorial	DevOps Roadmap	Top 50 String Top 50 DP	Low Level Design or LLD	Lodash Web Browser
Software Engineering	Python Interview Question		Top 15 Websites for CP	Crack System Design Round	
Digital Logic Design				System Design Interview Questions	
Engineering Maths				Grokking Modern System Design	
NCERT Solutions	School Subjects	Commerce	Management & Finance	UPSC	SSC/BANKING
NCERT Solutions for Class 12	Mathematics Physics	Business Studies Indian	Management HR Managament	Geography Notes	SSC CGL Syllabus
NCERT Solution for Class 11	Chemistry	Economics	Income Tax	History Notes	SBI PO Syllabus
NCERT Solutions for Class 10	Biology	Macroeconomics	Finance	Science and Technology Notes	SBI Clerk Syllabus
NCERT Solutions for Class 9	Social Science English Grammar	Microeconomics Statistics for Economics	Economics	Economics Notes	IBPS PO Syllabus
NCERT Solutions for Class 8				Important Topics in Ethics	IBPS Clerk Syllabus
Complete Study Material				UPSC Previous Year Papers	SSC CGL Practice Papers
Colleges	Companies	Preparation Corner	Exams	More Tutorials	Write & Earn
	IT Companies		JEE Mains		Write an Article

Indian Colleges	Software Development Companies	Company Wise Preparation for Artificial Intelligence(AI) Companies	JEE Advanced GATE CS NEET UGC NET CAT	Software Testing Software Development Product Management SAP SEO Linux Excel	Improve an Article Pick Topics to Write Share your Experiences Internships
Admission & Campus Experiences					
Top Engineering Colleges	CyberSecurity Companies	Experienced Interviews Internship Interviews			
Top BCA Colleges	Service Based Companies	Competitive Programming			
Top MBA Colleges	Product Based Companies	Aptitude Preparation			
Top Architecture College	PSUs for CS Engineers	Puzzles			
Choose College For Graduation					

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved