Data Visualisation - Graded Questions

`Note` - *This stub file doesn't contain the conceptual questions asked on the platform*

I) Marks Analysis

In the **'Marks.csv'** file, you can find the scores obtained by 200 students in 4 subjects of a standardised test. The different columns - `Score A`, `Score B`, `Score C` and `Score D` indicate the score obtained by a particular student in the respective subjects A, B, C and D.

Load the dataset to your notebook and answer the following questions

In [2]:
```
#Load the necessary Libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from collections import Counter
from datetime import datetime
```

In [3]:
```
#Load the dataset
df1 = pd.read_csv("Marks.csv")
```

In [4]:
```
df1.head()
```

Out[4]:

	Score A	Score B	Score C	Score D
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	12.0
3	151.5	41.3	58.5	16.5
4	180.8	10.8	58.4	17.9

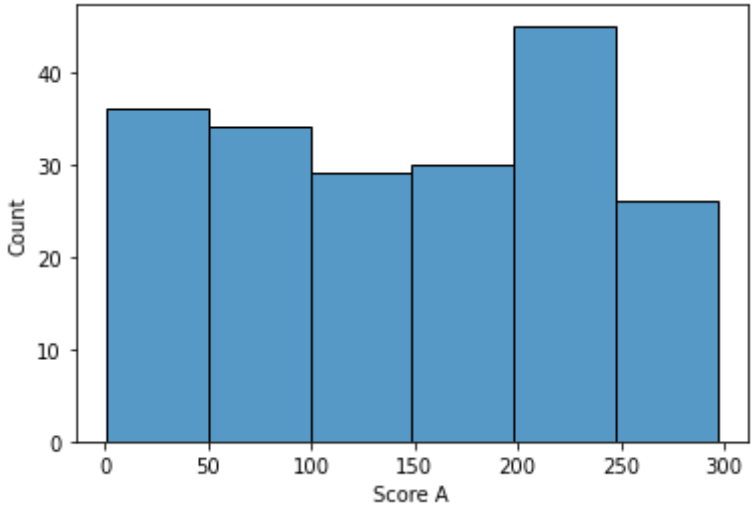
Q1) Load the dataset and plot a histogram for the `Score A` column by keeping the `number of bins to 6`. Which bin range among the following has the highest frequency?

(`Note` - *The bin ranges mentioned in the options are approximate values for the bin ranges that you'll actually get when you plot the histogram*)

a)0-50
b)50-100
c)150-200
d)200-250

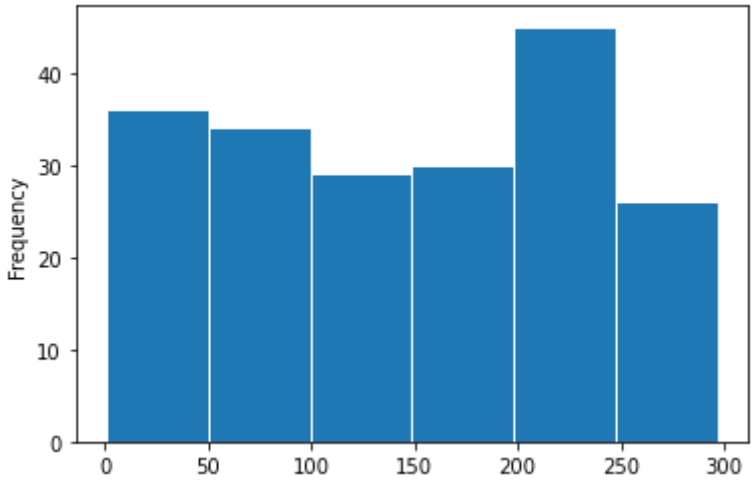
Answer: D

In [9]:
```
sns.histplot(data=df1["Score A"],bins=6)
plt.show()
```



In [11]:
```
df1["Score A"].plot.hist(edgecolor="white",bins=6)
```

Out[11]:
```
<AxesSubplot:ylabel='Frequency'>
```



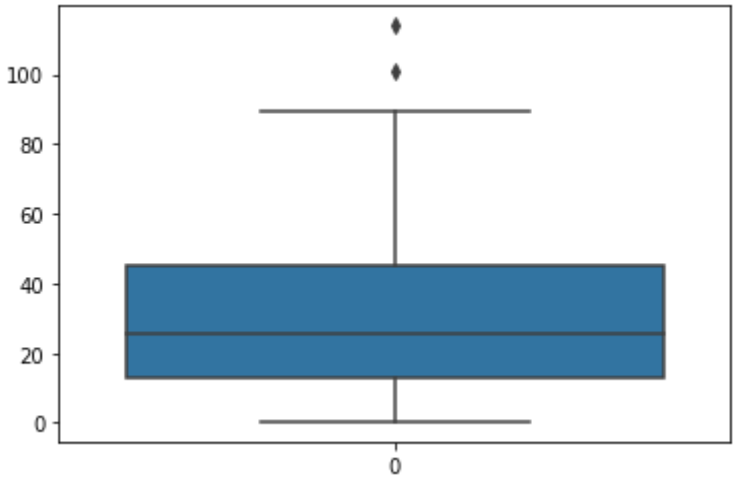
Q2) Plot a box plot for the column `Score C` and choose the correct option.

```
A - The 25th percentile lies between 20 and 40
B - The 75th percentile lies between 40 and 60
C - The 25th percentile lies between 0 and 20
D - Both B and C
```

Answer: D

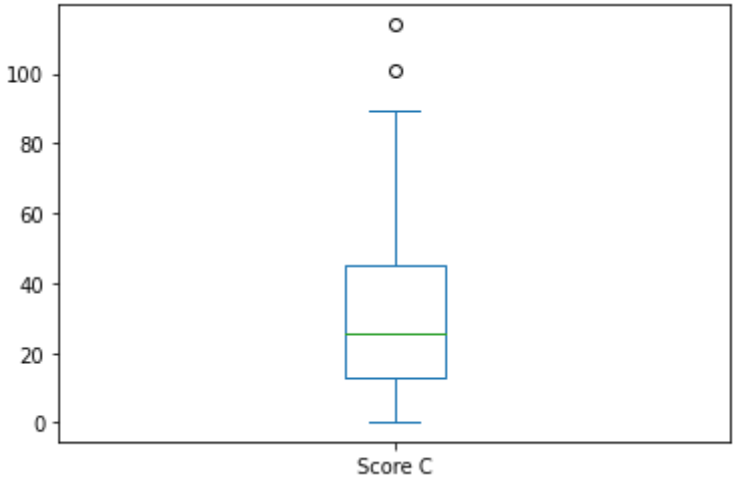
In [17]:

```
sns.boxplot(data=df1["Score C"])
plt.show()
```



In [19]:

```
df1["Score C"].plot.box()
plt.show()
```



II) Superstore Data

In the `superstore.csv` file, you have the details of orders purchased in an American online retail store. Load the dataset, observe and analyse the different columns and answer the following questions.

In [20]:
```
#Load the dataset
df2 = pd.read_csv("superstore.csv")
```

In [21]:
```
df2.head()
```

Out[21]:

	Order ID	Ship Mode	Segment	Region	Product ID	Sales	Quantity	Discount	Profit
0	CA-2016-152156	Second Class	Consumer	South	FUR-BO-10001798	261.9600	2	0%	41.9136
1	CA-2016-152156	Second Class	Consumer	South	FUR-CH-10000454	731.9400	3	0%	219.5820
2	CA-2016-138688	Second Class	Corporate	West	OFF-LA-10000240	14.6200	2	0%	6.8714
3	US-2015-108966	Standard Class	Consumer	South	FUR-TA-10000577	957.5775	5	0.45%	-383.0310
4	US-2015-108966	Standard Class	Consumer	South	OFF-ST-10000760	22.3680	2	0.20%	2.5164

In [29]:
```
df2.Discount = df2.Discount.replace({"%":""},regex=True)
df2.Discount = df2.Discount.astype(float)
df2.Discount
```

Out[29]:
```
0       0.00
1       0.00
2       0.00
3       0.45
4       0.20
        ...
9989    0.20
9990    0.00
9991    0.20
9992    0.00
9993    0.00
Name: Discount, Length: 9994, dtype: float64
```

In [36]:
```
df2.Profit
```

Out[36]:
```
0        41.9136
1       219.5820
2         6.8714
```

```
3      -383.0310
4         2.5164
             ...
9989      4.1028
9990     15.6332
9991     19.3932
9992     13.3200
9993     72.9480
Name: Profit, Length: 9994, dtype: float64
```

In [37]:
```
df2[df2.isna().any(axis=1)]
```

Out[37]:

Order ID	Ship Mode	Segment	Region	Product ID	Sales	Quantity	Discount	Profit

1. Jointplot

The following Jointplot was plotted using seaborn for 'Sales' vs 'Profit' from the dataset loaded to the 'df2' dataframe.

Answer:

df3 = df2[(df2.Profit < 0) & (df2.Sales < 15000)]

sns.jointplot('Sales', 'Profit', df3)

plt.show()

2. Data Visualisation

You can observe that the column Ship Mode has 4 categories - First Class, Second Class, Same Day and Standard Class, and the Segment has 3 - Consumer, Home Office, Corporate.

Now let's say you want to visualise how the average Profit varies across every Segment - Ship Mode combination (like Consumer - First Class, Corporate - Standard Class and so on). Which of the following visualisations can be utilised for the same?

Answer:

Heat map

3. Which of the following codes would help you create a bar plot in seaborn that compares the average Sales across the different Segment types?

Answer:

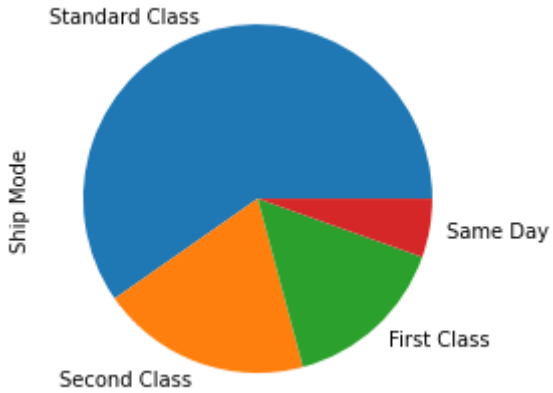
Both A and B

Q4) Plot a pie-chart to find the Ship Mode through which most of the orders are being delivered.

```
a)Standard Class
b)First Class
c)Second Class
d)Same Day
```

Answer: a) Standard Class

In [43]:
```
df2["Ship Mode"].value_counts().plot.pie()
plt.show()
```



Q5) Plot a bar chart comparing the average `Discount` across all the `Regions` and report back the `Region` getting the highest average discount

Note - You need to clean the `Discount` column first

a)Central
b)South
c)West
d)East

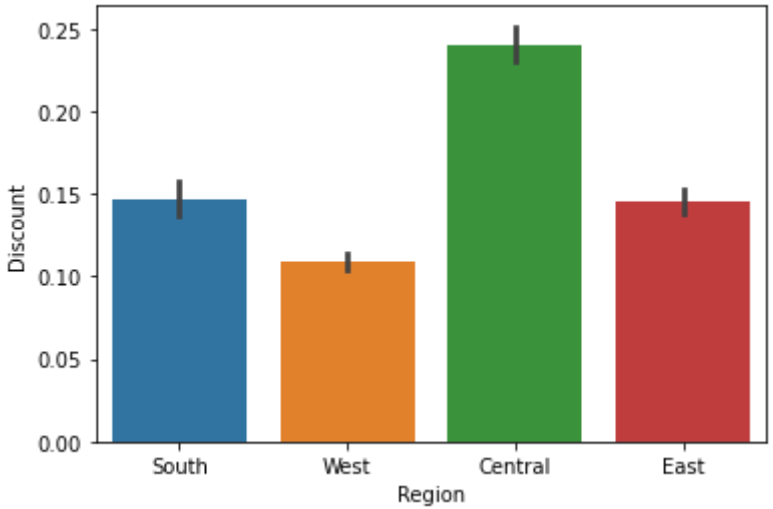
Answer: a) Central

In [50]:
```
df2.groupby("Region")["Discount"].mean()
```

Out[50]:
```
Region
Central    0.240353
East       0.145365
South      0.147253
West       0.109335
Name: Discount, dtype: float64
```

In [49]:
```
sns.barplot(data=df2,x="Region",y="Discount")
```

Out[49]:
```
<AxesSubplot:xlabel='Region', ylabel='Discount'>
```



In [51]:
```
help(sns.jointplot)
```

```
Help on function jointplot in module seaborn.axisgrid:

jointplot(*, x=None, y=None, data=None, kind='scatter', color=None, height=6, ratio=5, space=0.2, dropna=False, xlim=None, ylim=No
ne, marginal_ticks=False, joint_kws=None, marginal_kws=None, hue=None, palette=None, hue_order=None, hue_norm=None, **kwargs)
    Draw a plot of two variables with bivariate and univariate graphs.

    This function provides a convenient interface to the :class:`JointGrid`
    class, with several canned plot kinds. This is intended to be a fairly
    lightweight wrapper; if you need more flexibility, you should use
    :class:`JointGrid` directly.

    Parameters
    ----------
    x, y : vectors or keys in ``data``
        Variables that specify positions on the x and y axes.
    data : :class:`pandas.DataFrame`, :class:`numpy.ndarray`, mapping, or sequence
        Input data structure. Either a long-form collection of vectors that can be
        assigned to named variables or a wide-form dataset that will be internally
        reshaped.
    kind : { "scatter" | "kde" | "hist" | "hex" | "reg" | "resid" }
        Kind of plot to draw. See the examples for references to the underlying functions.
    color : :mod:`matplotlib color <matplotlib.colors>`
        Single color specification for when hue mapping is not used. Otherwise, the
        plot will try to hook into the matplotlib property cycle.
    height : numeric
        Size of the figure (it will be square).
    ratio : numeric
        Ratio of joint axes height to marginal axes height.
    space : numeric
        Space between the joint and marginal axes
    dropna : bool
        If True, remove observations that are missing from ``x`` and ``y``.
    {x, y}lim : pairs of numbers
        Axis limits to set before plotting.
    marginal_ticks : bool
        If False, suppress ticks on the count/density axis of the marginal plots.
    {joint, marginal}_kws : dicts
        Additional keyword arguments for the plot components.
    hue : vector or key in ``data``
        Semantic variable that is mapped to determine the color of plot elements.
        Semantic variable that is mapped to determine the color of plot elements.
    palette : string, list, dict, or :class:`matplotlib.colors.Colormap`
        Method for choosing the colors to use when mapping the ``hue`` semantic.
        String values are passed to :func:`color_palette`. List or dict values
```

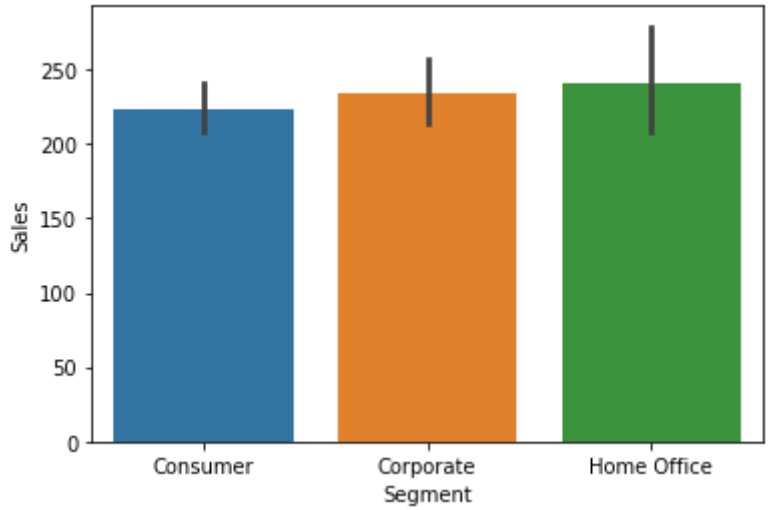
```
    imply categorical mapping, while a colormap object implies numeric mapping.
hue_order : vector of strings
    Specify the order of processing and plotting for categorical levels of the
    ``hue`` semantic.
hue_norm : tuple or :class:`matplotlib.colors.Normalize`
    Either a pair of values that set the normalization range in data units
    or an object that will map from data units into a [0, 1] interval. Usage
    implies numeric mapping.
kwargs
    Additional keyword arguments are passed to the function used to
    draw the plot on the joint Axes, superseding items in the
    ``joint_kws`` dictionary.

Returns
-------
:class:`JointGrid`
    An object managing multiple subplots that correspond to joint and marginal axes
    for plotting a bivariate relationship or distribution.

See Also
--------
JointGrid : Set up a figure with joint and marginal views on bivariate data.
PairGrid : Set up a figure with joint and marginal views on multiple variables.
jointplot : Draw multiple bivariate plots with univariate marginal distributions.

Examples
--------

.. include:: ../docstrings/jointplot.rst
```

In [60]:

```
sns.barplot(data = df2, x = 'Segment', y = 'Sales', estimator = np.mean)
```

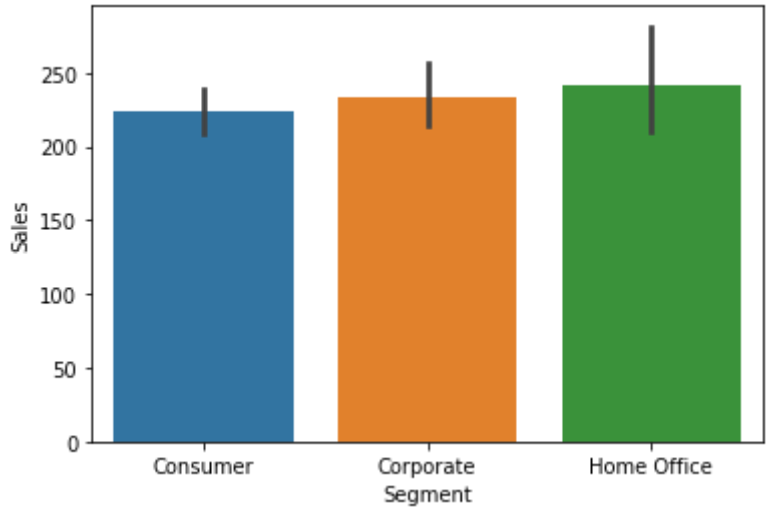
Out[60]:

```
<AxesSubplot:xlabel='Segment', ylabel='Sales'>
```



In [61]:
```
sns.barplot(data = df2, x = 'Segment', y = 'Sales')
```

Out[61]:
```
<AxesSubplot:xlabel='Segment', ylabel='Sales'>
```



In [ ]: