

```
[2]: #matplotlib and seaborn visualization library which is used convert table data into graphs/charts
```

```
[4]: import matplotlib.pyplot as plt
import seaborn as sns
```

```
[6]: print(dir(plt),end=" ")
```

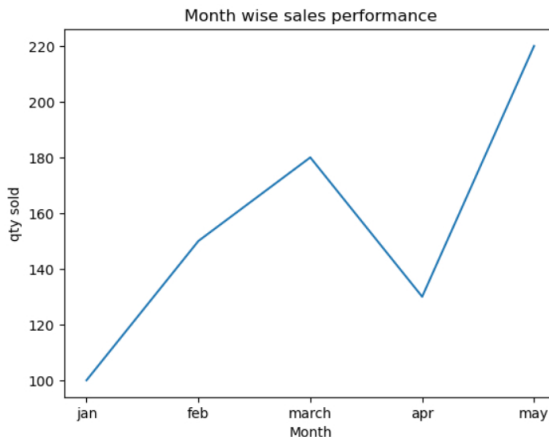
```
['AbstractContextManager', 'Annotation', 'Arrow', 'Artist', 'Autolocator', 'AxLine', 'Axes', 'BackendFilter', 'Button', 'Circle', 'Colormap', 'Enum', 'ExitStack', 'Figure', 'FigureBase', 'FigureCanvasBase', 'FigureManagerBase', 'FixedFormatter', 'FixedLocator', 'FormatStrFormatter', 'Formatter', 'FuncFormatter', 'GridSpec', 'IndexLocator', 'Line2D', 'LinearLocator', 'Locator', 'LogFormatter', 'LogFormatterExponent', 'LogFormatterMathtext', 'LogLocator', 'MaxNLocator', 'MouseButton', 'MultipleLocator', 'Normalize', 'NullFormatter', 'NullLocator', 'PolarAxes', 'Polygon', 'Rectangle', 'ScalarFormatter', 'Slider', 'Subplot', 'SubplotSpec', 'TYPE_CHECKING', 'Text', 'TickHelper', 'Widget', 'NO_PYPLLOT_NOTE', 'REPL_DISPLAYHOOK', 'ReplDisplayHook', 'annotations', 'builtins', 'cached', 'doc', 'file', 'loader', 'name', 'package', 'spec', 'add_pyplot_note', 'api', 'auto_draw_if_interactive', 'backend_mod', 'color_sequences', 'colormaps', 'copy_docstring_and_deprecators', 'docstring', 'draw_all_if_interactive', 'get_backend_mod', 'get_pyplot_commands', 'log', 'pylab_helpers', 'warn_if_gui_out_of_main_thread', 'acorn', 'angle_spectrum', 'annotate', 'annotations', 'arrow', 'autoscale', 'autumn', 'axes', 'axhline', 'axhspan', 'axis', 'axline', 'axvline', 'axvspan', 'backend_registry', 'bar', 'bar_label', 'barbs', 'barh', 'bone', 'box', 'boxplot', 'broken_barh', 'cast', 'cbook', 'cla', 'clabel', 'clf', 'clim', 'close', 'cm', 'cohere', 'color_sequences', 'colorbar', 'colormaps', 'connect', 'contour', 'contourf', 'cool', 'copper', 'csd', 'cycler', 'delaxes', 'disconnect', 'draw', 'draw_all', 'draw_if_interactive', 'ecdf', 'errorbar', 'eventplot', 'figaspect', 'figimage', 'figlegend', 'fignum_exists', 'figtext', 'figure', 'fill', 'fill_between', 'fill_betweenx', 'findobj', 'flag', 'functools', 'gca', 'gcf', 'gci', 'get', 'get_backend', 'get_cmap', 'get_current_fig_manager', 'get_figlabels', 'get_fignums', 'get_plot_commands', 'get_scale_names', 'get_tp', 'ginput', 'gray', 'grid', 'hexbin', 'hist', 'hist2d', 'hlines', 'hot', 'hsv', 'importlib', 'imread', 'imsave', 'imshow', 'inferno', 'inspect', 'install_repl_displayhook', 'interactive', 'ioff', 'ion', 'isinteractive', 'jet', 'legend', 'locator_params', 'logging', 'loglog', 'magma', 'magnitude_spectrum', 'margins', 'matplotlib', 'matshow', 'minorticks_off', 'minorticks_on', 'mlab', 'new_figure_manager', 'nipy_spectral', 'np', 'overload', 'pause', 'pcolor', 'pcolormesh', 'phase_spectrum', 'pie', 'pink', 'plasma', 'plot', 'plot_date', 'polar', 'prism', 'psd', 'quiver', 'quiverkey', 'rc', 'rcParams', 'rcParamsDefault', 'rcParamsOrig', 'rc_context', 'rcdefaults', 'rcsetup', 'rgrids', 'savefig', 'sca', 'scatter', 'sci', 'semilogx', 'semilogy', 'set_cmap', 'set_loglevel', 'setp', 'show', 'specgram', 'spring', 'spy', 'stackplot', 'stairs', 'stem', 'step', 'streamplot', 'style', 'subplot', 'subplot2grid', 'subplot_mosaic', 'subplot_tool', 'subplots', 'subplots_adjust', 'summer', 'suplot', 'switch_backend', 'sys', 'table', 'text', 'thetagrids', 'threading', 'tick_params', 'ticklabel_format', 'tight_layout', 'time', 'title', 'tricontour', 'tricontourf', 'tripcolor', 'tripplot', 'twinx', 'twiny', 'uninstall_repl_displayhook', 'violinplot', 'viridis', 'vlines', 'waitforbuttonpress', 'winter', 'xcorr', 'xkcd', 'xlabel', 'xlim', 'xscale', 'xticks', 'ylabel', 'ylim', 'yscale', 'yticks']
```

```
[8]: print(dir(sns),end=" ")
```

```
['FacetGrid', 'JointGrid', 'PairGrid', 'builtins', 'cached', 'doc', 'file', 'loader', 'name', 'package', 'path', 'spec', 'version', 'base', 'compat', 'core', 'docstrings', 'orig_rc_params', 'statistics', 'stats', 'algorithms', 'axes_style', 'axisgrid', 'barplot', 'blend_palette', 'boxenplot', 'boxplot', 'categorical', 'catplot', 'choose_colorbrewer_palette', 'choose_cubehelix_palette', 'choose_dark_palette', 'choose_diverging_palette', 'choose_light_palette', 'clustermap', 'cm', 'color_palette', 'colors', 'countplot', 'crayon_palette', 'crayons', 'cubehelix_palette', 'dark_palette', 'desaturate', 'despine', 'displot', 'distplot', 'distributions', 'diverging_palette', 'dogplot', 'ecdfplot', 'external', 'get_data_home', 'get_dataset_names', 'heatmap', 'histplot', 'hls_palette', 'husl_palette', 'jointplot', 'kdeplot', 'light_palette', 'lineplot', 'lmp', 'load_dataset', 'matrix', 'miscplot', 'move_legend', 'mpl', 'mpl_palette', 'pairplot', 'palettes', 'palplot', 'plotting_context', 'pointplot', 'rcmod', 'regplot', 'regression', 'relational', 'relplot', 'reset_defaults', 'reset_orig', 'residplot', 'rugplot', 'saturate', 'scatterplot', 'set', 'set_color_codes', 'set_context', 'set_hls_values', 'set_palette', 'set_style', 'set_theme', 'stripplot', 'swarmplot', 'utils', 'violinplot', 'widgets', 'xkcd_palette', 'xkcd_rgb']
```

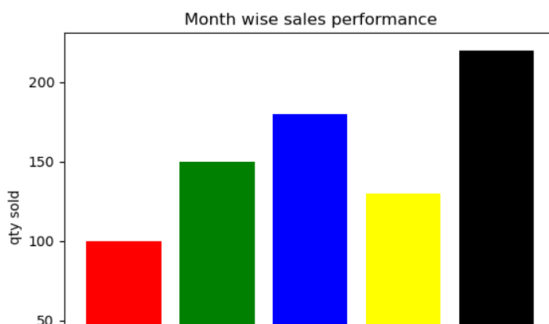
```
[20]: x=["jan","Feb","march","apr","may"]
y=[100,150,180,130,220]
```

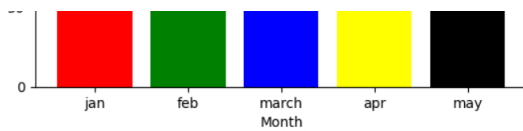
```
plt.plot(x,y)
plt.xlabel("Month")
plt.ylabel("qty sold")
plt.title("Month wise sales performance")
plt.show()
```



```
[26]: x=["jan","Feb","march","apr","may"]
y=[100,150,180,130,220]
c=["red","green","blue","yellow","black"]
```

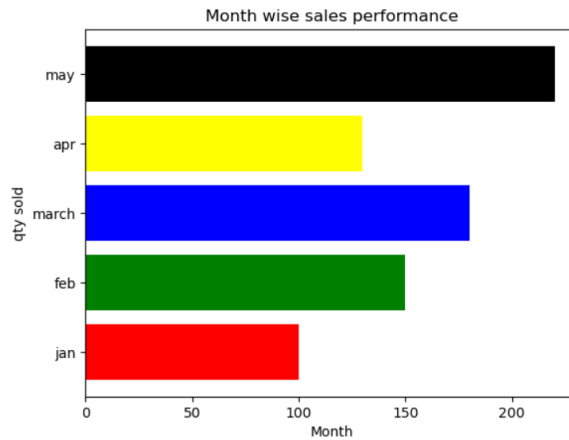
```
plt.bar(x,y,color=["red","green","blue","yellow","black"])
plt.xlabel("Month")
plt.ylabel("qty sold")
plt.title("Month wise sales performance")
plt.show()
```





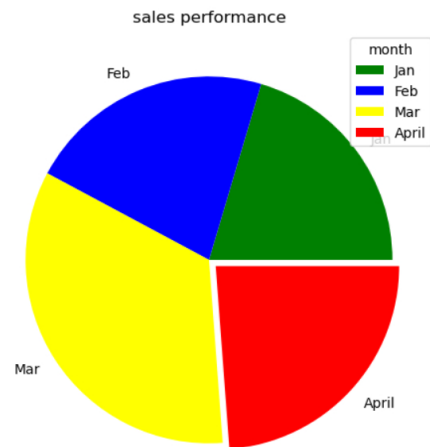
```
[28]: x=["jan", "feb", "march", "apr", "may"]
      y=[100,150,180,130,220]
      c=["red", "green", "blue", "yellow", "black"]

      plt.barh(x,y,color=["red", "green", "blue", "yellow", "black"])
      plt.xlabel("Month")
      plt.ylabel("qty sold")
      plt.title("Month wise sales performance")
      plt.show()
```



```
[44]: month = ["Jan", "Feb", "Mar", "April"]
      values= [150,160,250,175]

      plt.figure(figsize=(8,6))
      plt.pie(values,labels=month,startangle=0,explode=[0,0,0,0.05],shadow=False,colors=["green", "blue", "yellow", "red"])
      plt.legend(title="month")
      plt.title('sales performance')
      plt.show()
```

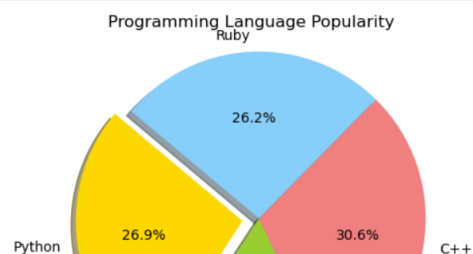


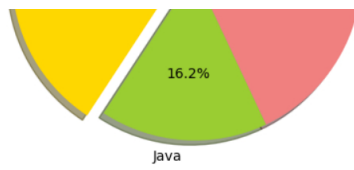
```
[46]: import matplotlib.pyplot as plt

      # Data to plot
      labels = ['Python', 'Java', 'C++', 'Ruby']
      sizes = [215, 130, 245, 210]
      colors = ['gold', 'yellowgreen', 'lightcoral', 'lightskyblue']
      explode = (0.1, 0, 0, 0) # explode the 1st slice

      # Plotting the pie chart
      plt.pie(sizes, explode=explode, labels=labels, colors=colors,
              autopct='%1.1f%%', shadow=True, startangle=140)

      # Equal aspect ratio ensures that pie is drawn as a circle.
      plt.axis('equal')
      plt.title('Programming Language Popularity')
      plt.show()
```





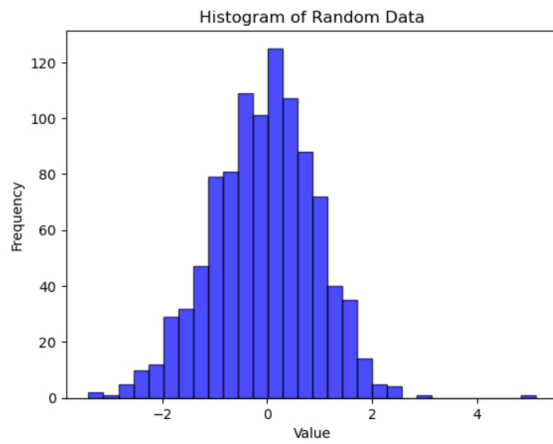
```
[48]: import matplotlib.pyplot as plt
import numpy as np

# Sample data
data = np.random.randn(1000) # Generate 1000 random numbers from a normal distribution

# Plotting the histogram
plt.hist(data, bins=30, color='blue', alpha=0.7, edgecolor='black')

# Adding titles and labels
plt.title('Histogram of Random Data')
plt.xlabel('Value')
plt.ylabel('Frequency')

# Show the plot
plt.show()
```



```
[ ]:
```