In [ ]:
```
# import libraries
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
```

In [ ]:
```
# import your data
df=pd.read_csv("multiple linear regression.csv")
df.head()
```

Out[ ]:

	age	distance	YearsExperience	Salary
0	31.1	77.75	1.1	39343
1	31.3	78.25	1.3	46205
2	31.5	78.75	1.5	37731
3	32.0	80.00	2.0	43525
4	32.2	80.50	2.2	39891

In [ ]:
```
df.isnull().sum()
```

Out[ ]:
```
age                0
distance           0
YearsExperience    0
Salary             0
dtype: int64
```

In [ ]:
```
x= df[['age','distance','YearsExperience']]
y= df['Salary']
```

In [ ]:
```
x.head()
```

Out[ ]:

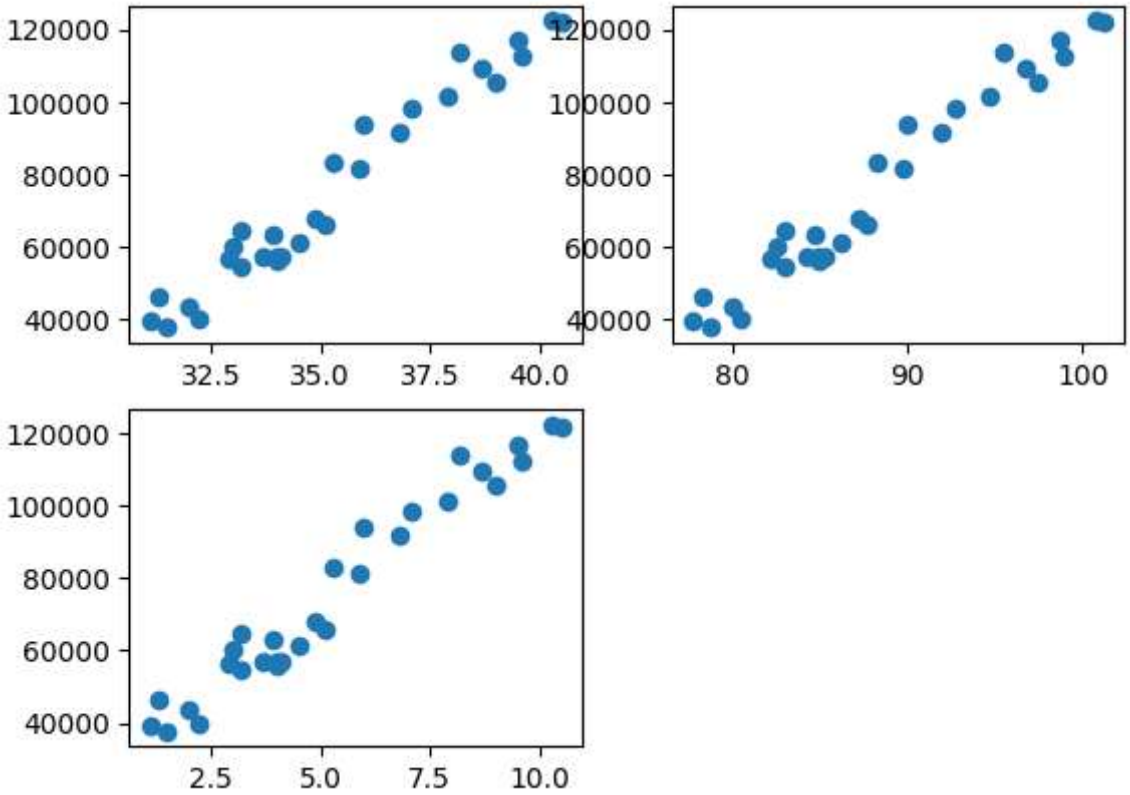
	age	distance	YearsExperience
0	31.1	77.75	1.1
1	31.3	78.25	1.3
2	31.5	78.75	1.5
3	32.0	80.00	2.0
4	32.2	80.50	2.2

In [ ]:
```
y.head()
```

Out[ ]:
```
0    39343
1    46205
2    37731
3    43525
4    39891
Name: Salary, dtype: int64
```

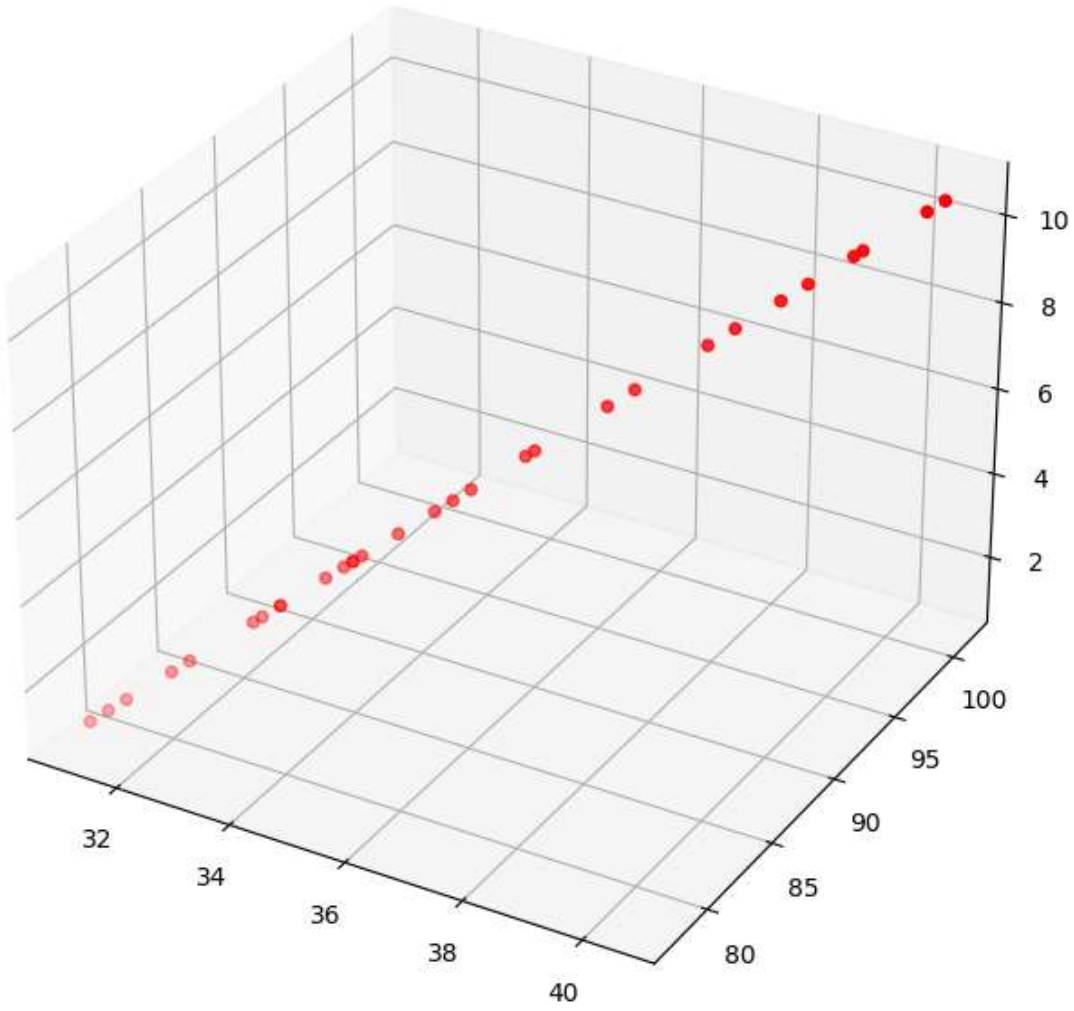
In [ ]:
```
plt.subplot(2,2,1)
plt.scatter(x['age'],y)
plt.subplot(2,2,2)
plt.scatter(x['distance'],y)
plt.subplot(2,2,3,)
plt.scatter(x['YearsExperience'],y)
```

Out[ ]:
```
<matplotlib.collections.PathCollection at 0x1d646ee1990>
```



In [ ]:
```
plt.style.use('default')

fig = plt.figure(figsize=(16, 8))

ax = fig.add_subplot(111, projection='3d')
x1=(x['age'])
x2=(x['distance'])
x3=(x['YearsExperience'])
ax.scatter(x1,x2,x3, c='r', marker='o')
```

Out[ ]:
```
<mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x1d64778f7c0>
```



In [ ]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test=train_test_split(x, y, test_size=0.2, random_stat
```

In [ ]:
```
from sklearn.linear_model import LinearRegression
model=LinearRegression().fit(x_train, y_train)
```

In [ ]:
```
print('score of test data=', model.score(x_test, y_test))
print('score of train data=', model.score(x_train, y_train))
```

```
score of test data= 0.988401541985491
score of train data= 0.9409532368371482
```

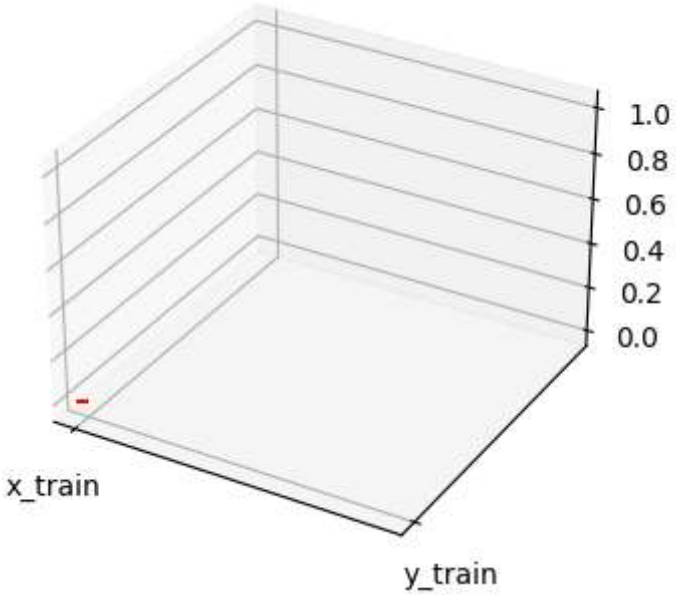
In [ ]:
```
reg=LinearRegression().fit(x_test, y_test)
print('Test score=', reg.score(x_test, y_test))
print('train score=', reg.score(x_train, y_train))
```

```
Test score= 0.9934433748762496
train score= 0.930766846826246
```

In [ ]:
```
plt.style.use('default')

fig = plt.figure(figsize=(12, 4))
```

```
ax = fig.add_subplot(111, projection='3d')
x1=(['age'])
x2=(x['distance'])
x3=(x['YearsExperience'])
ax.scatter('x_train','y_train', c='r', marker=0)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_7424/3742707303.py in <module>
      7 # x1=['x_train']
      8 # y1=['y_train']
----> 9 ax.scatter('x_train','y_train', c='r', marker=0)

~\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\__init__.py
in inner(ax, data, *args, **kwargs)
   1410     def inner(ax, *args, data=None, **kwargs):
   1411         if data is None:
-> 1412             return func(ax, *map(sanitize_sequence, args), **kwargs)
   1413
   1414         bound = new_sig.bind(ax, *args, **kwargs)

~\AppData\Local\Programs\Python\Python310\lib\site-packages\mpl_toolkits\mplot3d\a
xes3d.py in scatter(self, xs, ys, zs, zdir, s, c, depthshade, *args, **kwargs)
   2394             self.set_zmargin(0.05)
   2395
-> 2396         self.auto_scale_xyz(xs, ys, zs, had_data)
   2397
   2398         return patches

~\AppData\Local\Programs\Python\Python310\lib\site-packages\mpl_toolkits\mplot3d\a
xes3d.py in auto_scale_xyz(self, X, Y, Z, had_data)
    650         # minimum sized rectangular volume holds the data.
    651         if np.shape(X) == np.shape(Y):
--> 652             self.xy_dataLim.update_from_data_xy(
    653                 np.column_stack([np.ravel(X), np.ravel(Y)]), not had_data)
    654         else:

~\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\transforms.
py in update_from_data_xy(self, xy, ignore, updatex, updatey)
    965             return
    966
--> 967         path = Path(xy)
    968         self.update_from_path(path, ignore=ignore,
    969                               updatex=updatex, updatey=updatey)

~\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\path.py in
__init__(self, vertices, codes, _interpolation_steps, closed, readonly)
    127             and codes as read-only arrays.
    128         """
--> 129         vertices = _to_unmasked_float_array(vertices)
    130         _api.check_shape((None, 2), vertices=vertices)
    131

~\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\cbook\__ini
t__.py in _to_unmasked_float_array(x)
   1296         return np.ma.asarray(x, float).filled(np.nan)
   1297     else:
-> 1298         return np.asarray(x, float)
   1299
   1300

ValueError: could not convert string to float: 'x_train'
```



In [ ]:

```
model.predict([[31.3,77.5,1.1]])
```

```
C:\Users\Haier\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\b
ase.py:450: UserWarning: X does not have valid feature names, but LinearRegression
was fitted with feature names
  warnings.warn(
```

Out[ ]:

```
array([-2.42122788e+15])
```

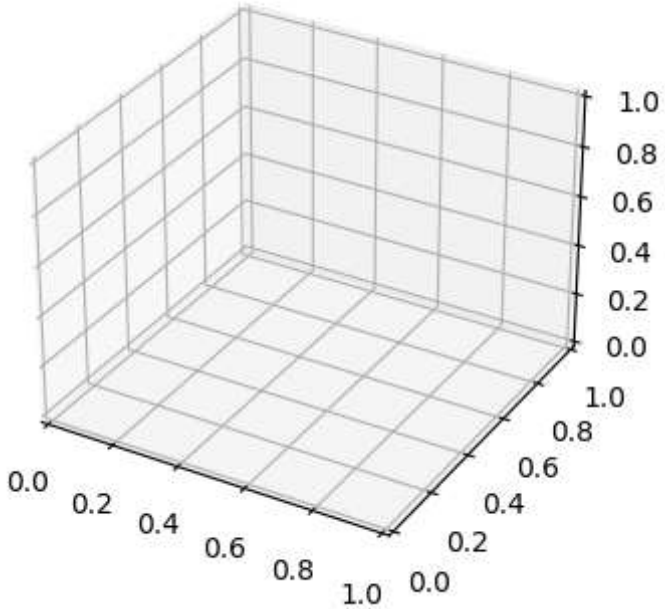
In [ ]:

In [ ]:

```
# an attempt to draw a single prdicted value (failed)
model.predict([[31.3,77.5,1.1]])
plt.style.use('default')

fig = plt.figure(figsize=(12, 4))

ax1 = fig.add_subplot(111, projection='3d')
ax.scatter(model.predict([[31.3,77.5,1.1]]), c='r', marker='o')
```

```
C:\Users\Haier\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\b
ase.py:450: UserWarning: X does not have valid feature names, but LinearRegression
was fitted with feature names
  warnings.warn(
C:\Users\Haier\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\b
ase.py:450: UserWarning: X does not have valid feature names, but LinearRegression
was fitted with feature names
  warnings.warn(
```

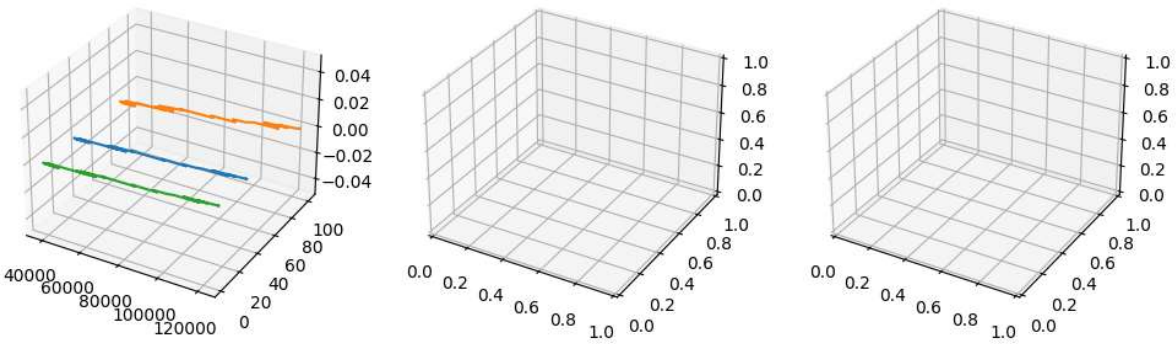
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_7424/3962961087.py in <module>
      6
      7 ax1 = fig.add_subplot(111, projection='3d')
----> 8 ax.scatter(model.predict([[31.3,77.5,1.1]]), c='r', marker='o')

~\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\__init__.py
in inner(ax, data, *args, **kwargs)
   1410     def inner(ax, *args, data=None, **kwargs):
   1411         if data is None:
-> 1412             return func(ax, *map(sanitize_sequence, args), **kwargs)
   1413
   1414         bound = new_sig.bind(ax, *args, **kwargs)

TypeError: Axes3D.scatter() missing 1 required positional argument: 'ys'
```



In [ ]:

```
plt.style.use('default')

fig = plt.figure(figsize=(12, 4))

ax1 = fig.add_subplot(131, projection='3d')
ax2 = fig.add_subplot(132, projection='3d')
ax3 = fig.add_subplot(133, projection='3d')

axes = [ax1, ax2, ax3]
for ax in axes:
    ax.plot(x, y)
    ax.scatter(x_train, y_train,facecolor=(0,0,0,0), s=20, edgecolor='#70b3f0')
    ax.set_xlabel('Porosity (%)', fontsize=12)
    ax.set_ylabel('VR', fontsize=12)
    ax.set_zlabel('Gas Prod. (Mcf/day)', fontsize=12)
    ax.locator_params(nbins=4, axis='x')
    ax.locator_params(nbins=5, axis='x')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_7424/2536680817.py in <module>
     10 for ax in axes:
     11     ax.plot(x, y)
---> 12     ax.scatter(x_train, y_train,facecolor=(0,0,0,0), s=20, edgecolor='#70b
3f0')
     13
     14

~\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\__init__.py
in inner(ax, data, *args, **kwargs)
   1410     def inner(ax, *args, data=None, **kwargs):
   1411         if data is None:
-> 1412             return func(ax, *map(sanitize_sequence, args), **kwargs)
   1413
   1414         bound = new_sig.bind(ax, *args, **kwargs)

~\AppData\Local\Programs\Python\Python310\lib\site-packages\mpl_toolkits\mplot3d\a
xes3d.py in scatter(self, xs, ys, zs, zdir, s, c, depthshade, *args, **kwargs)
   2377         zs_orig = zs
   2378
-> 2379         xs, ys, zs = np.broadcast_arrays(
   2380             *[np.ravel(np.ma.filled(t, np.nan)) for t in [xs, ys, zs]])
   2381         s = np.ma.ravel(s)  # This doesn't have to match x, y in size.

~\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\core\overrides.p
y in broadcast_arrays(*args, **kwargs)

~\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\lib\stride_trick
s.py in broadcast_arrays(subok, *args)
    537     args = [np.array(_m, copy=False, subok=subok) for _m in args]
    538
--> 539     shape = _broadcast_shape(*args)
    540
    541     if all(array.shape == shape for array in args):

~\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\lib\stride_trick
s.py in _broadcast_shape(*args)
    419     # use the old-iterator because np.nditer does not handle size 0 arrays
    420     # consistently
--> 421     b = np.broadcast(*args[:32])
    422     # unfortunately, it cannot handle 32 or more arguments directly
    423     for pos in range(32, len(args), 31):

ValueError: shape mismatch: objects cannot be broadcast to a single shape.  Mismat
ch is between arg 0 with shape (24,) and arg 1 with shape (72,).
```

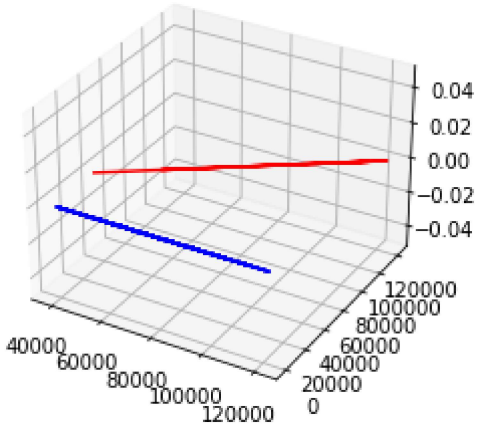


In [ ]:

```
import matplotlib.pyplot
fig = matplotlib.pyplot.figure()
ax  = fig.add_subplot(111, projection = '3d')
```

```
ax.plot(x_train, y_train, color = 'b')
ax.plot(x_test, x_test, color = 'r')

ax.scatter(x_train, x_train, y, color = 'g', marker = "o")
ax.scatter(x_test, y_test, y, color = 'y', marker = "^")

matplotlib.pyplot.show()
```

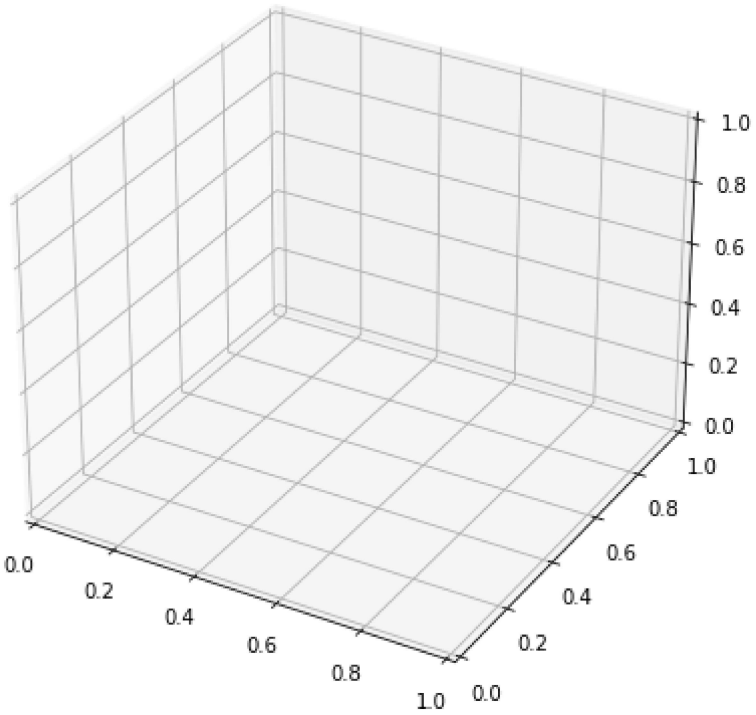
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_7424/39341545.py in <module>
      6 ax.plot(x_test, x_test, color = 'r')
      7
----> 8 ax.scatter(x_train, x_train, y, color = 'g', marker = "o")
      9 ax.scatter(x_test, y_test, y, color = 'y', marker = "^")
     10

~\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\__init__.py
in inner(ax, data, *args, **kwargs)
   1410     def inner(ax, *args, data=None, **kwargs):
   1411         if data is None:
-> 1412             return func(ax, *map(sanitize_sequence, args), **kwargs)
   1413
   1414         bound = new_sig.bind(ax, *args, **kwargs)

~\AppData\Local\Programs\Python\Python310\lib\site-packages\mpl_toolkits\mplot3d\a
xes3d.py in scatter(self, xs, ys, zs, zdir, s, c, depthshade, *args, **kwargs)
   2377         zs_orig = zs
   2378
-> 2379         xs, ys, zs = np.broadcast_arrays(
   2380             *[np.ravel(np.ma.filled(t, np.nan)) for t in [xs, ys, zs]])
   2381         s = np.ma.ravel(s)  # This doesn't have to match x, y in size.

~\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\core\overrides.p
y in broadcast_arrays(*args, **kwargs)

~\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\lib\stride_trick
s.py in broadcast_arrays(subok, *args)
    537     args = [np.array(_m, copy=False, subok=subok) for _m in args]
    538
--> 539     shape = _broadcast_shape(*args)
    540
    541     if all(array.shape == shape for array in args):

~\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\lib\stride_trick
s.py in _broadcast_shape(*args)
    419     # use the old-iterator because np.nditer does not handle size 0 arrays
    420     # consistently
--> 421     b = np.broadcast(*args[:32])
    422     # unfortunately, it cannot handle 32 or more arguments directly
    423     for pos in range(32, len(args), 31):

ValueError: shape mismatch: objects cannot be broadcast to a single shape.  Mismat
ch is between arg 0 with shape (24,) and arg 2 with shape (90,).
```



In [ ]:

```
fig = plt.figure(figsize = (10, 7))
ax = plt.axes(projection ="3d")
ax.scatter3D(x_train, y_train, color = "green")
plt.title("simple 3D scatter plot")
plt.show()
```

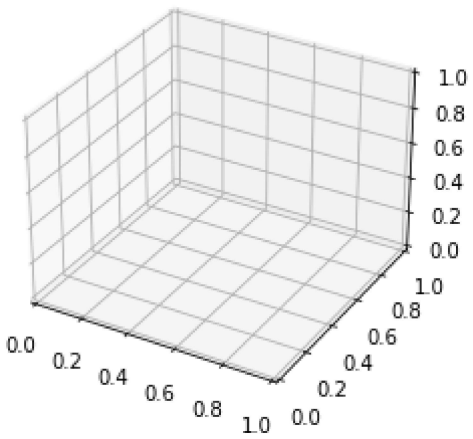
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_7424/3052281377.py in <module>
      1 fig = plt.figure(figsize = (10, 7))
      2 ax = plt.axes(projection ="3d")
----> 3 ax.scatter3D(x_train, y_train, color = "green")
      4 plt.title("simple 3D scatter plot")
      5 plt.show()

~\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\__init__.py
in inner(ax, data, *args, **kwargs)
   1410     def inner(ax, *args, data=None, **kwargs):
   1411         if data is None:
-> 1412             return func(ax, *map(sanitize_sequence, args), **kwargs)
   1413
   1414         bound = new_sig.bind(ax, *args, **kwargs)

~\AppData\Local\Programs\Python\Python310\lib\site-packages\mpl_toolkits\mplot3d\a
xes3d.py in scatter(self, xs, ys, zs, zdir, s, c, depthshade, *args, **kwargs)
   2377         zs_orig = zs
   2378
-> 2379         xs, ys, zs = np.broadcast_arrays(
   2380             *[np.ravel(np.ma.filled(t, np.nan)) for t in [xs, ys, zs]])
   2381         s = np.ma.ravel(s)  # This doesn't have to match x, y in size.

~\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\core\overrides.p
y in broadcast_arrays(*args, **kwargs)

~\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\lib\stride_trick
s.py in broadcast_arrays(subok, *args)
    537     args = [np.array(_m, copy=False, subok=subok) for _m in args]
    538
--> 539     shape = _broadcast_shape(*args)
    540
    541     if all(array.shape == shape for array in args):

~\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\lib\stride_trick
s.py in _broadcast_shape(*args)
    419     # use the old-iterator because np.nditer does not handle size 0 arrays
    420     # consistently
--> 421     b = np.broadcast(*args[:32])
    422     # unfortunately, it cannot handle 32 or more arguments directly
    423     for pos in range(32, len(args), 31):

ValueError: shape mismatch: objects cannot be broadcast to a single shape.  Mismat
ch is between arg 0 with shape (72,) and arg 1 with shape (24,).
```



In [ ]:

```
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
# ax.scatter(X['foo'], X['bar'], Y, label='data')

# for x0, x1, yt, yp in zip(x_test['foo'], x_test['bar'], y_test, y_train):
#     ax.plot([x0, x0], [x1, x1], [yt, yp], color='red')

ax.scatter(x_train, y_train, x_test, color='red', marker='s', label='prediction')

ax.set_xlabel('X0')
ax.set_ylabel('X1')
ax.set_zlabel('y')
ax.legend()
fig.show()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_13616/662887485.py in <module>
      6 #     ax.plot([x0, x0], [x1, x1], [yt, yp], color='red')
      7
----> 8 ax.scatter(x_train, y_train, y_train, color='red', marker='s', label='pred
iction')
      9
     10 ax.set_xlabel('X0')

~\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\__init__.py
in inner(ax, data, *args, **kwargs)
   1410     def inner(ax, *args, data=None, **kwargs):
   1411         if data is None:
-> 1412             return func(ax, *map(sanitize_sequence, args), **kwargs)
   1413
   1414         bound = new_sig.bind(ax, *args, **kwargs)

~\AppData\Local\Programs\Python\Python310\lib\site-packages\mpl_toolkits\mplot3d\a
xes3d.py in scatter(self, xs, ys, zs, zdir, s, c, depthshade, *args, **kwargs)
   2377         zs_orig = zs
   2378
-> 2379         xs, ys, zs = np.broadcast_arrays(
   2380             *[np.ravel(np.ma.filled(t, np.nan)) for t in [xs, ys, zs]])
   2381         s = np.ma.ravel(s)  # This doesn't have to match x, y in size.

~\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\core\overrides.p
y in broadcast_arrays(*args, **kwargs)

~\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\lib\stride_trick
s.py in broadcast_arrays(subok, *args)
    537     args = [np.array(_m, copy=False, subok=subok) for _m in args]
    538
--> 539     shape = _broadcast_shape(*args)
    540
    541     if all(array.shape == shape for array in args):

~\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\lib\stride_trick
s.py in _broadcast_shape(*args)
    419     # use the old-iterator because np.nditer does not handle size 0 arrays
    420     # consistently
--> 421     b = np.broadcast(*args[:32])
    422     # unfortunately, it cannot handle 32 or more arguments directly
    423     for pos in range(32, len(args), 31):

ValueError: shape mismatch: objects cannot be broadcast to a single shape.  Mismat
ch is between arg 0 with shape (72,) and arg 1 with shape (24,).
```

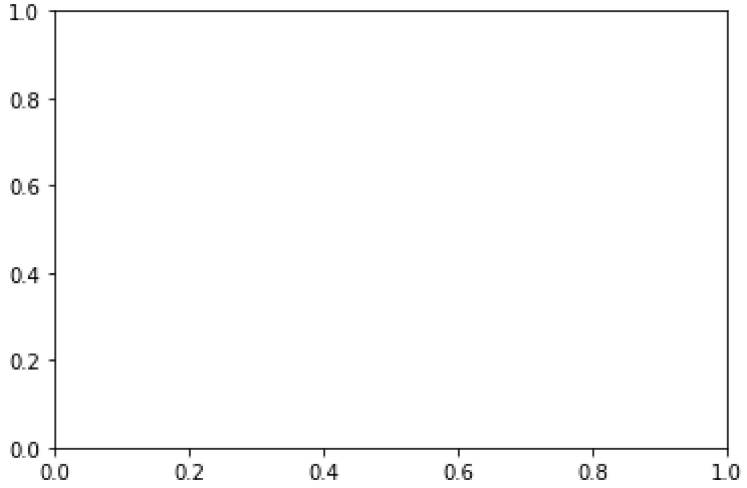


- import seaborn as sns didn't work
- sns.regplot(x=y_train, y=y_test, ci=None, color="b")

In [ ]:

```
import matplotlib.pyplot as plt
plt.scatter(x_train, y_train)
plt.plot(x_train, model.predict(x_train), color="green")
plt.xlabel("tajurba")
plt.ylabel("tankhwa")
plt.title("train plot")
plt.show()
```

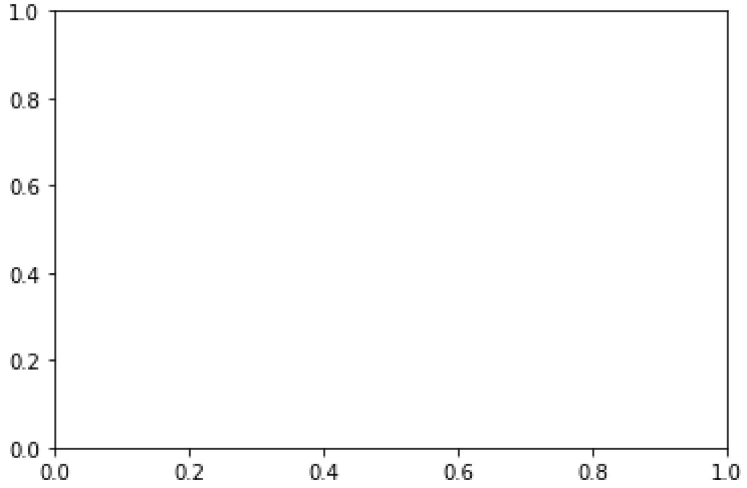
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_13616/3707649035.py in <module>
      1 import matplotlib.pyplot as plt
----> 2 plt.scatter(x_train, y_train)
      3 plt.plot(x_train, model.predict(x_train), color="green")
      4 plt.xlabel("tajurba")
      5 plt.ylabel("tankhwa")

~\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\pyplot.py i
n scatter(x, y, s, c, marker, cmap, norm, vmin, vmax, alpha, linewidths, edgecolor
s, plotnonfinite, data, **kwargs)
   2805         vmin=None, vmax=None, alpha=None, linewidths=None, *,
   2806         edgecolors=None, plotnonfinite=False, data=None, **kwargs):
-> 2807     __ret = gca().scatter(
   2808         x, y, s=s, c=c, marker=marker, cmap=cmap, norm=norm,
   2809         vmin=vmin, vmax=vmax, alpha=alpha, linewidths=linewidths,

~\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\__init__.py
in inner(ax, data, *args, **kwargs)
   1410     def inner(ax, *args, data=None, **kwargs):
   1411         if data is None:
-> 1412             return func(ax, *map(sanitize_sequence, args), **kwargs)
   1413
   1414         bound = new_sig.bind(ax, *args, **kwargs)

~\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\axes\_axes.
py in scatter(self, x, y, s, c, marker, cmap, norm, vmin, vmax, alpha, linewidths,
edgecolors, plotnonfinite, **kwargs)
   4367         y = np.ma.ravel(y)
   4368         if x.size != y.size:
-> 4369             raise ValueError("x and y must be the same size")
   4370
   4371         if s is None:

ValueError: x and y must be the same size
```



In [ ]:

```
plt.scatter(x_test, y_test)
plt.plot(x_test, model.predict(x_test), color="green")
plt.xlabel("linear")
plt.ylabel("tankhwa")
plt.title("train plot")
plt.show()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_13616/710065700.py in <module>
----> 1 plt.scatter(x_test, y_test)
      2 plt.plot(x_test, model.predict(x_test), color="green")
      3 plt.xlabel("linear")
      4 plt.ylabel("tankhwa")
      5 plt.title("train plot")

~\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\pyplot.py i
n scatter(x, y, s, c, marker, cmap, norm, vmin, vmax, alpha, linewidths, edgecolor
s, plotnonfinite, data, **kwargs)
   2805         vmin=None, vmax=None, alpha=None, linewidths=None, *,
   2806         edgecolors=None, plotnonfinite=False, data=None, **kwargs):
-> 2807     __ret = gca().scatter(
   2808         x, y, s=s, c=c, marker=marker, cmap=cmap, norm=norm,
   2809         vmin=vmin, vmax=vmax, alpha=alpha, linewidths=linewidths,

~\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\__init__.py
in inner(ax, data, *args, **kwargs)
   1410     def inner(ax, *args, data=None, **kwargs):
   1411         if data is None:
-> 1412             return func(ax, *map(sanitize_sequence, args), **kwargs)
   1413
   1414         bound = new_sig.bind(ax, *args, **kwargs)

~\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\axes\_axes.
py in scatter(self, x, y, s, c, marker, cmap, norm, vmin, vmax, alpha, linewidths,
edgecolors, plotnonfinite, **kwargs)
   4367         y = np.ma.ravel(y)
   4368         if x.size != y.size:
-> 4369             raise ValueError("x and y must be the same size")
   4370
   4371         if s is None:

ValueError: x and y must be the same size
```



In [ ]: