wlj0zrujl

February 18, 2025

[1]:
```
import pandas as pd
import numpy as np
```

[2]:
```
df=pd.read_csv('salary.csv')
```

[3]:
```
df
```

[3]:
```
       Age  Gender    Education Level              Job Title  \
0     32.0    Male         Bachelor's      Software Engineer
1     28.0  Female           Master's           Data Analyst
2     45.0    Male                PhD         Senior Manager
3     36.0  Female         Bachelor's        Sales Associate
4     52.0    Male           Master's               Director
…     …     …                      …                      …
6699  49.0  Female                PhD  Director of Marketing
6700  32.0    Male        High School        Sales Associate
6701  30.0  Female  Bachelor's Degree      Financial Manager
6702  46.0    Male    Master's Degree      Marketing Manager
6703  26.0  Female        High School        Sales Executive

      Years of Experience    Salary
0                     5.0   90000.0
1                     3.0   65000.0
2                    15.0  150000.0
3                     7.0   60000.0
4                    20.0  200000.0
…                       …       …
6699                 20.0  200000.0
6700                  3.0   50000.0
6701                  4.0   55000.0
6702                 14.0  140000.0
6703                  1.0   35000.0

[6704 rows x 6 columns]
```

[4]:
```
df.mean()
```

```
C:\Users\Welcome\AppData\Local\Temp\ipykernel_11636\972437606.py:1:
```

```
FutureWarning: The default value of numeric_only in DataFrame.mean is
deprecated. In a future version, it will default to False. In addition,
specifying 'numeric_only=None' is deprecated. Select only valid columns or
specify the value of numeric_only to silence this warning.
  df.mean()
```

[4]:
```
Age                        33.620859
Years of Experience         8.094687
Salary                 115326.964771
dtype: float64
```

[5]:
```
df.loc[:,'Age'].mean()
```

[5]:
```
33.62085944494181
```

[6]:
```
df.mean(axis=1)[0:4]
```

```
C:\Users\Welcome\AppData\Local\Temp\ipykernel_11636\850889490.py:1:
FutureWarning: Dropping of nuisance columns in DataFrame reductions (with
'numeric_only=None') is deprecated; in a future version this will raise
TypeError.  Select only valid columns before calling the reduction.
  df.mean(axis=1)[0:4]
```

[6]:
```
0    30012.333333
1    21677.000000
2    50020.000000
3    20014.333333
dtype: float64
```

[7]:
```
df.median()
```

```
C:\Users\Welcome\AppData\Local\Temp\ipykernel_11636\530051474.py:1:
FutureWarning: The default value of numeric_only in DataFrame.median is
deprecated. In a future version, it will default to False. In addition,
specifying 'numeric_only=None' is deprecated. Select only valid columns or
specify the value of numeric_only to silence this warning.
  df.median()
```

[7]:
```
Age                        32.0
Years of Experience         7.0
Salary                 115000.0
dtype: float64
```

[8]:
```
df.loc[:,'Age'].median()
```

[8]:
```
32.0
```

[9]:
```
df.median(axis=1)[0:4]
```

```
C:\Users\Welcome\AppData\Local\Temp\ipykernel_11636\381455229.py:1:
FutureWarning: Dropping of nuisance columns in DataFrame reductions (with
'numeric_only=None') is deprecated; in a future version this will raise
TypeError.  Select only valid columns before calling the reduction.
  df.median(axis=1)[0:4]
```

[9]:
```
0    32.0
1    28.0
2    45.0
3    36.0
dtype: float64
```

[10]:
```
df.mode()
```

[10]:
```
    Age Gender    Education Level          Job Title  Years of Experience  \
0  27.0   Male  Bachelor's Degree  Software Engineer                  2.0

     Salary
0  140000.0
```

[11]:
```
df.loc[:,'Age'].mode()
```

[11]:
```
0    27.0
Name: Age, dtype: float64
```

[12]:
```
df.min()
```

```
C:\Users\Welcome\AppData\Local\Temp\ipykernel_11636\3962516015.py:1:
FutureWarning: The default value of numeric_only in DataFrame.min is deprecated.
In a future version, it will default to False. In addition, specifying
'numeric_only=None' is deprecated. Select only valid columns or specify the
value of numeric_only to silence this warning.
  df.min()
```

[12]:
```
Age                    21.0
Years of Experience     0.0
Salary                350.0
dtype: float64
```

[13]:
```
df.loc[:,'Age'].min(skipna = False)
```

[13]:
```
nan
```

[14]:
```
df.max()
```

```
C:\Users\Welcome\AppData\Local\Temp\ipykernel_11636\1299571182.py:1:
FutureWarning: The default value of numeric_only in DataFrame.max is deprecated.
In a future version, it will default to False. In addition, specifying
```

```
'numeric_only=None' is deprecated. Select only valid columns or specify the
value of numeric_only to silence this warning.
  df.max()
```

[14]:
```
Age                        62.0
Years of Experience        34.0
Salary                 250000.0
dtype: float64
```

[15]:
```
df.loc[:,'Age'].max(skipna = False)
```

[15]:
```
nan
```

[16]:
```
df.std()
```

```
C:\Users\Welcome\AppData\Local\Temp\ipykernel_11636\3390915376.py:1:
FutureWarning: The default value of numeric_only in DataFrame.std is deprecated.
In a future version, it will default to False. In addition, specifying
'numeric_only=None' is deprecated. Select only valid columns or specify the
value of numeric_only to silence this warning.
  df.std()
```

[16]:
```
Age                        7.614633
Years of Experience        6.059003
Salary                 52786.183911
dtype: float64
```

[17]:
```
df.loc[:,'Age'].std()
```

[17]:
```
7.614632626251171
```

[18]:
```
df.std(axis=1)[0:4]
```

```
C:\Users\Welcome\AppData\Local\Temp\ipykernel_11636\3966588610.py:1:
FutureWarning: Dropping of nuisance columns in DataFrame reductions (with
'numeric_only=None') is deprecated; in a future version this will raise
TypeError.  Select only valid columns before calling the reduction.
  df.std(axis=1)[0:4]
```

[18]:
```
0    51950.845001
1    37518.820650
2    86585.221170
3    34628.606156
dtype: float64
```

[20]:
```
df.groupby(['Gender'])['Age'].mean()
```

[20]:
```
Gender
Female    32.624088
Male      34.415895
Other     39.571429
Name: Age, dtype: float64
```

[21]:
```
from sklearn import preprocessing
enc = preprocessing.OneHotEncoder()
enc_df = pd.DataFrame(enc.fit_transform(df[['Gender']]).toarray())
enc_df
```

[21]:

	0	1	2	3
0	0.0	1.0	0.0	0.0
1	1.0	0.0	0.0	0.0
2	0.0	1.0	0.0	0.0
3	1.0	0.0	0.0	0.0
4	0.0	1.0	0.0	0.0
…	…	…	…	…
6699	1.0	0.0	0.0	0.0
6700	0.0	1.0	0.0	0.0
6701	1.0	0.0	0.0	0.0
6702	0.0	1.0	0.0	0.0
6703	1.0	0.0	0.0	0.0

[6704 rows x 4 columns]

[25]:
```
df_u = df.rename(columns={'Salary)': 'Income'}, inplace=False)  # Fix the␣
↪parenthesis
print(df_u.groupby('Gender')['Salary'].mean())  # Fix the grouping and indexing
```

```
Gender
Female    107888.998672
Male      121389.870915
Other     125869.857143
Name: Salary, dtype: float64
```

[28]:
```
df_encode = df_u.join(enc_df)
print(df_encode)  # Use the correct variable name
```

```
       Age  Gender    Education Level              Job Title  \
0     32.0    Male         Bachelor's      Software Engineer
1     28.0  Female           Master's           Data Analyst
2     45.0    Male                PhD         Senior Manager
3     36.0  Female         Bachelor's        Sales Associate
4     52.0    Male           Master's               Director
…      …      …                 …                      …
6699  49.0  Female                PhD  Director of Marketing
6700  32.0    Male        High School        Sales Associate
```

```
6701  30.0  Female  Bachelor's Degree      Financial Manager
6702  46.0    Male    Master's Degree      Marketing Manager
6703  26.0  Female        High School        Sales Executive

      Years of Experience    Salary    0    1    2    3
0                     5.0   90000.0  0.0  1.0  0.0  0.0
1                     3.0   65000.0  1.0  0.0  0.0  0.0
2                    15.0  150000.0  0.0  1.0  0.0  0.0
3                     7.0   60000.0  1.0  0.0  0.0  0.0
4                    20.0  200000.0  0.0  1.0  0.0  0.0
…                     …        …   …   …   …   …
6699                 20.0  200000.0  1.0  0.0  0.0  0.0
6700                  3.0   50000.0  0.0  1.0  0.0  0.0
6701                  4.0   55000.0  1.0  0.0  0.0  0.0
6702                 14.0  140000.0  0.0  1.0  0.0  0.0
6703                  1.0   35000.0  1.0  0.0  0.0  0.0

[6704 rows x 10 columns]
```

[29]:
```python
import pandas as pd

# Calculate skewness for numerical columns
skewness = df_encode.skew()

print("Skewness of numerical columns:")
print(skewness)
```

```
Skewness of numerical columns:
Age                    0.905596
Years of Experience    0.981188
Salary                 0.057344
0                      0.202749
1                     -0.193060
2                     21.819080
3                     57.883500
dtype: float64

C:\Users\Welcome\AppData\Local\Temp\ipykernel_11636\3033343048.py:4:
FutureWarning: The default value of numeric_only in DataFrame.skew is
deprecated. In a future version, it will default to False. In addition,
specifying 'numeric_only=None' is deprecated. Select only valid columns or
specify the value of numeric_only to silence this warning.
  skewness = df_encode.skew()
```

[30]:
```python
import numpy as np
from scipy import stats
```

[31]:
```python
z = np.abs(stats.zscore(df['Salary']))
```

[32]:
```
print(z)
```

```
0      NaN
1      NaN
2      NaN
3      NaN
4      NaN
       ..
6699   NaN
6700   NaN
6701   NaN
6702   NaN
6703   NaN
Name: Salary, Length: 6704, dtype: float64
```

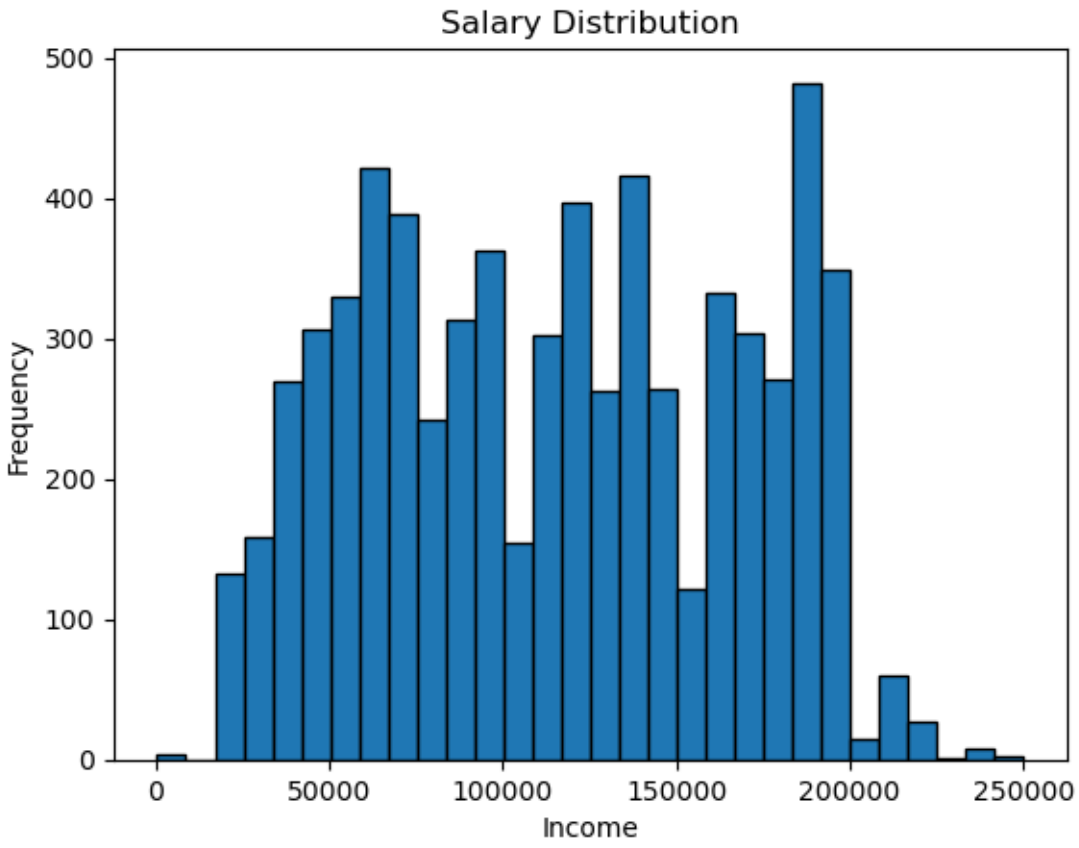
[33]:
```
import matplotlib.pyplot as plt
new_df['Salary'].plot(kind = 'hist')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[33], line 2
      1 import matplotlib.pyplot as plt
----> 2 new_df['Salary'].plot(kind = 'hist')

NameError: name 'new_df' is not defined
```

[35]:
```
import matplotlib.pyplot as plt

df_encode['Salary'].plot(kind='hist', bins=30, edgecolor='black')  # Use the␣
 ↪correct DataFrame name
plt.xlabel('Income')
plt.ylabel('Frequency')
plt.title('Salary Distribution')
plt.show()
```

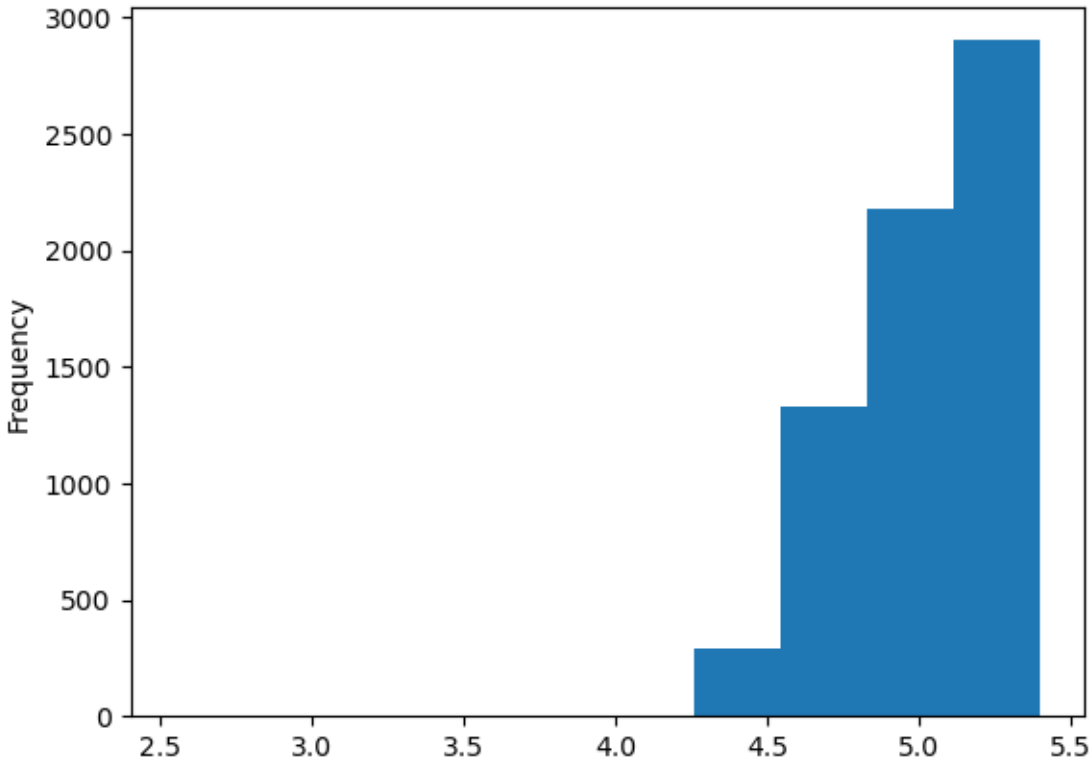


[37]:
```
import numpy as np

df['log_math'] = np.log10(df['Salary'])  # Added the missing closing parenthesis
```

[38]:
```
df['log_math'].plot(kind = 'hist')
```

[38]:
```
<Axes: ylabel='Frequency'>
```


3000
2500
2000
1500
1000
500
0
Frequency
2.5
3.0
3.5
4.0
4.5
5.0
5.5

[ ]: