Python Chilla Data Cleaning Notebook

> **Ali Nawaz\ Artificial Intelligence Engineer at NUST\ Education : Master in Software Engineering**

In [ ]:
```
import plotly.express as px
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('D:/Python ka Chilla/python_chilla/data/cleaned_chilla_data.csv')
df.head(2)
```

Out[ ]:

	sex	location	age_limit	qaulification	subject	purpose	employment	blood	SIM_company	si
0	Male	Pakistan	36-40	Masters	Natural Sciences	to boost my skill set	Unemplyed	B+	U-fone	Prepa
1	Male	Pakistan	26-30	Bachelors	IT	to boost my skill set	Student	B+	U-fone	Prepa

2 rows × 23 columns

Data Cleaning and Analyzing

In [ ]:
```
# # rename_col_name
# df.rename(columns={'Qualification_completed': 'Qaulification', 'field_of_study': 'Sub
# 'Purpose_for_chilla': 'purpose','What are you?': 'Employment','Blood group ':'Blood',
# 'Your favorite programming language?':'Programming_language','Marital Status?':'Marit
# 'Where do you live?':'living_place','Research/Working experience (Float/Int) years':'
# 'Your Weight in kg? (float)':'Weight','Height in cm? Freelancer- (Float)':'Height','H
# 'Light kitni der band hti hy? int':'Loadsheeding'}, inplace = True)
```

In [ ]:
```
# df = df.replace({'Age' : { 36-40 : 38, 26-30 : 28, 31-35 : 33, 21-25 : 23, 16-20 : 16
# df['Age'] = df['Age'].str.replace('36-40','38') other way to change
# df = df.replace({'marital_status' : { 'Yes' : 1, 'No' : 0}})
# df.housing.map(dict(yes=1, no=0))

df['experience'] = df['experience'].astype(float)#.apply(pd.to_numeric)
# df['experience'] = pd.to_numeric(df['experience'], downcast='float')
df['age'] = df['age'].astype(float)
df['weight'] = df['weight'].astype(float)
df['height'] = df['height'].astype(float)
df['coding_duration'] = df['coding_duration'].astype(float)
df['loadsheeding'] = df['loadsheeding'].astype(float)
# df.drop('age_limit', axis=1, inplace=True)
df.to_csv("D:/Python ka Chilla/python_chilla/data/cleaned_chilla_data.csv", index=False
```

In [ ]:
```
df.head(5)
```

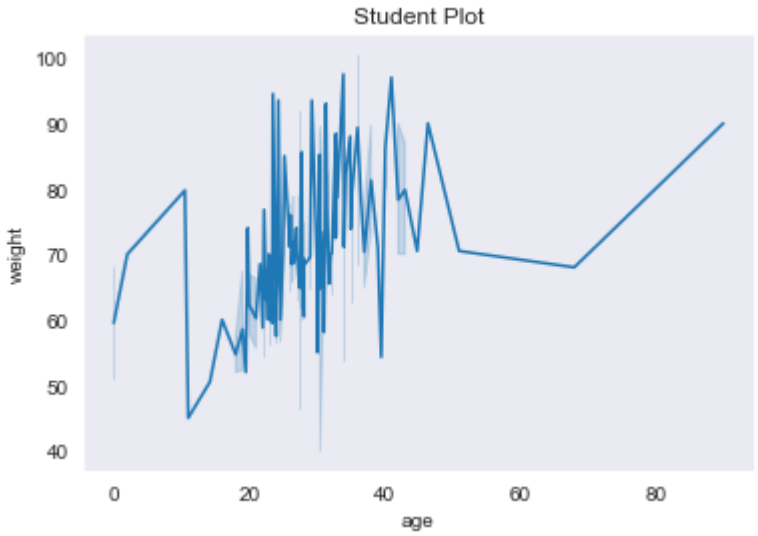
Out[ ]:

	sex	location	age_limit	qaulification	subject	purpose	employment	blood	SIM_company
0	Male	Pakistan	36-40	Masters	Natural Sciences	to boost my skill set	Unemplyed	B+	U-fone
1	Male	Pakistan	26-30	Bachelors	IT	to boost my skill set	Student	B+	U-fone
2	Male	Pakistan	31-35	Masters	Enginnering	Switch my field of study	Employed	B+	Zong
3	Female	Pakistan	31-35	Masters	IT	to boost my skill set	Employed	O+	U-fone
4	Female	Pakistan	26-30	Masters	Enginnering	to boost my skill set	Student	A-	Mobilink

5 rows × 23 columns

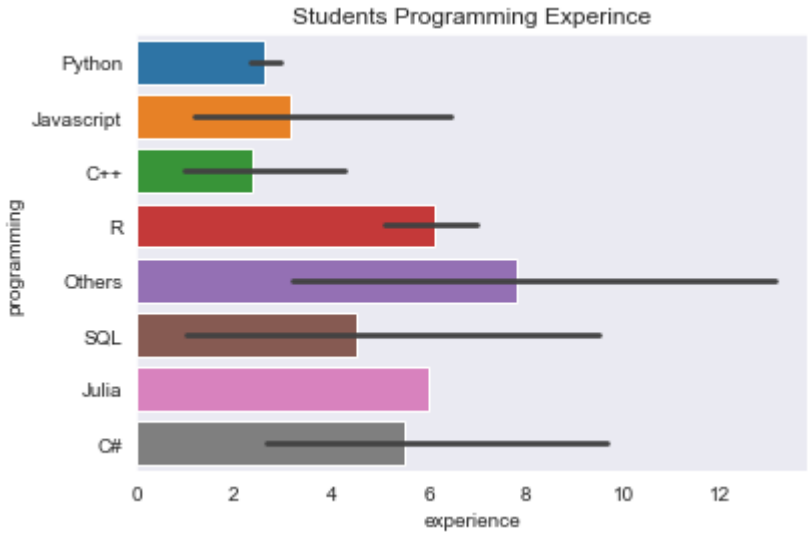
In [ ]:
```
fig = px.ecdf(df, x="coding_duration", color="sex")
fig.show()
```

In [ ]:
```
sns.lineplot(x='age', y = "weight", data=df)
plt.title("Student Plot")
plt.show()
```



In [ ]:
```
sns.barplot(x='experience', y = "programming", data=df, saturation=0.8)
sns.set_style('dark')
```

```
plt.title("Students Programming Experince")
plt.show()
```



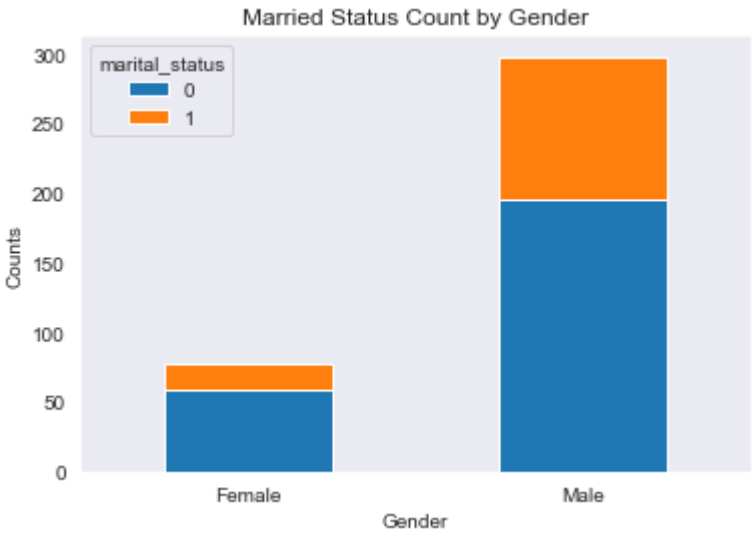
In [ ]:

```
dff = df[['sex', 'marital_status']]

# create a pivot table
dfp = dff.pivot_table(index='sex', columns=['marital_status'], aggfunc=len)

# plot the dataframe
dfp.plot(kind='bar', stacked=True, ylabel='Counts', xlabel='Gender',
         title='Married Status Count by Gender', rot=0)
```

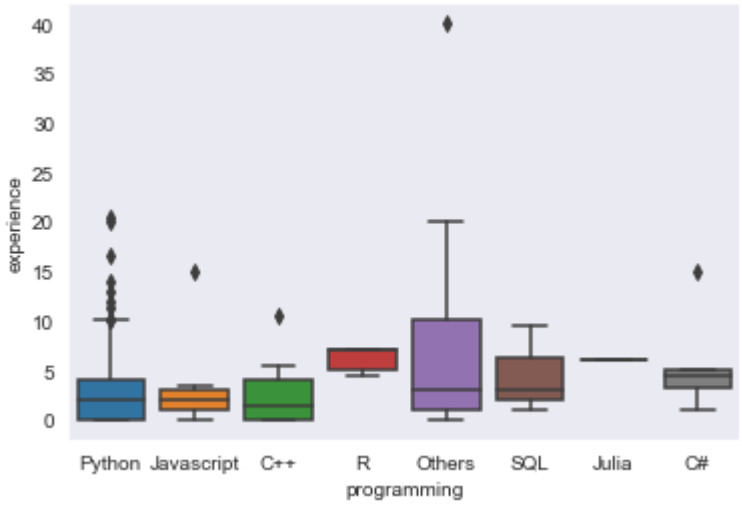
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x1a8242737f0>



In [ ]:

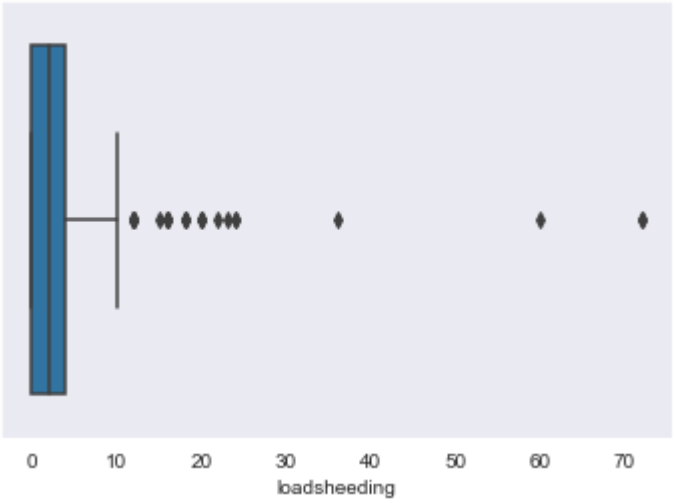
```
sns.boxplot(x='programming', y = "experience", data=df)
```

Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x1a821dbde48>



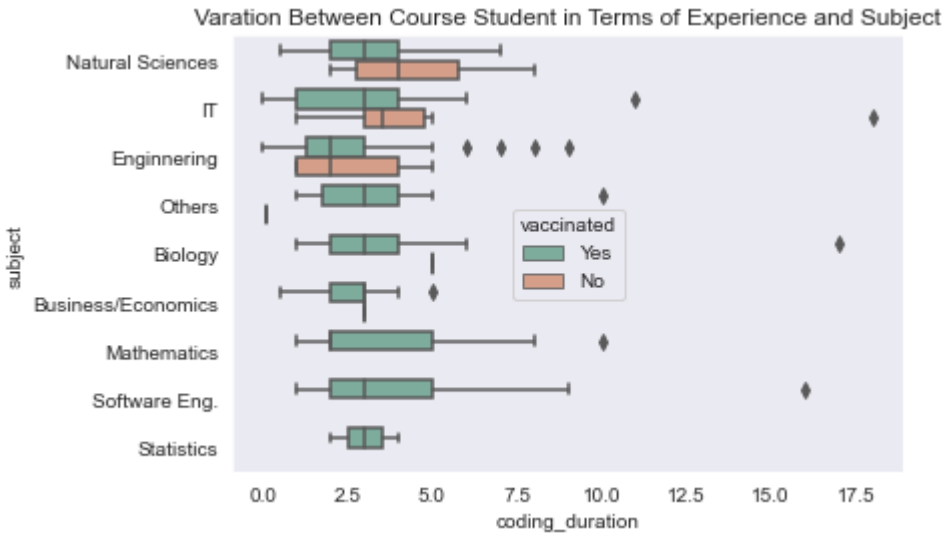
In [ ]:
```
sns.boxplot(x=df['loadsheeding'])
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a821cee0f0>
```



In [ ]:
```
sns.boxplot(x='coding_duration', y = "subject", data=df, hue='vaccinated', palette= 'Se
plt.title("Varation Between Course Student in Terms of Experience and Subject")
```

Out[ ]:
```
Text(0.5, 1.0, 'Varation Between Course Student in Terms of Experience and Subject')
```



In [ ]:
```
fig = px.scatter(df, x="experience", y="coding_duration", color="living_place", margina
           marginal_x="box", trendline="ols", template="simple_white")
fig.show()
```

In [ ]:
```
# df = df.query("weight == 178.0").query("living_place == 'Urban'")
# df.loc[df['experience'] < 2.0, 'programming'] = 'employment' # Represent only large c
fig = px.pie(df, values='experience', names='programming', title='Experience in Program
fig.show()
```

In [ ]:
```
fig = px.sunburst(df, path=['employment', 'qaulification'], values='coding_duration',
                  color='experience', hover_data=['location'])
fig.show()
```

In [ ]:
```
fig = px.violin(df, y="experience", x="vaccinated", color="sex", box=True, points="all"
fig.show()
```

In [ ]:
```
fig = px.scatter(df, x="weight", y="height", color="SIM_company")
fig.show()
```

In [ ]:
```
fig = px.bar(df, x="subject", y="experience", color="pc",
             pattern_shape="pc", pattern_shape_sequence=[".", "x", "+"])
fig.show()
```

In [ ]:
```
fig = px.parallel_categories(df, color="age", color_continuous_scale=px.colors.sequenti
fig.show()
```

In [ ]:
```
fig = px.bar_polar(df, r="age_limit", theta="subject", color="age_limit", template="plo
            color_discrete_sequence= px.colors.sequential.Plasma_r)
fig.show()
```

In [ ]:
```
fig = px.line(df, x='experience', y='age', color='subject', markers=True)
fig.show()
```

In [ ]:
```
fig = px.scatter_3d(df, x='age', y='experience', z='coding_duration',
              color='subject')
fig.show()
```

In [ ]:
```

```