datasetmed

June 11, 2025

0.1 We've been provided with data on cancer patients and are required to analyze it.

0.1.1 Dataset Description:

This dataset appears to contain information on **cancer patients**, with variables capturing **demographics**, **health background**, **lifestyle factors**, **treatment details**, and **outcomes**. Each row represents an individual pat.

0.1.2 Variables Overview:

Variable	Description
`id`	Unique identifier for each patient
`age`	Patient's age in years
`gender`	Gender of the patient (`Male` or `Female`)
`country`	Country of residence
`diagnosis_date`	Date when the patient was diagnosed with cancer
`cancer_stage`	Cancer stage at diagnosis (`Stage I`, `Stage III`, etc.)
`family_history`	Whether the patient has a family history of cancer (`Yes` or `No`)
`smoking_status`	Smoking behavior (`Passive Smoker`, `Former Smoker`, etc.)
`bmi`	Body Mass Index (BMI)
`cholesterol_level`	Measured cholesterol level
`hypertension`	Binary indicator for hypertension (`1` = Yes, `0` = No)
`asthma`	Binary indicator for asthma (`1` = Yes, `0` = No)
`cirrhosis`	Binary indicator for cirrhosis (`1` = Yes, `0` = No)
`other_cancer`	Binary indicator for presence of other types of cancer (`1` = Yes, `0` = No)
`treatment_type`	Type of cancer treatment (`Chemotherapy`, `Surgery`, `Combined`, etc.)
`end_treatment_date`	Date when treatment ended
`survived`	Survival outcome (`1` = Survived, `0` = Did not *ad this into a pandas DataFrame for analysis.

0.1.3 import Python libraries

[30]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import plotly.express as px
```

0.2 import dataset

[5]:

```
df = pd.read_csv(r'C:\Users\Administrator\Desktop\dataset_med.csv')
```

[6]:

```
df.head()
```

[6]:

```
   id   age  gender      country diagnosis_date cancer_stage family_history  \
0   1  64.0    Male       Sweden     2016-04-05      Stage I            Yes
1   2  50.0  Female  Netherlands     2023-04-20    Stage III            Yes
2   3  65.0  Female      Hungary     2023-04-05    Stage III            Yes
3   4  51.0  Female      Belgium     2016-02-05      Stage I             No
4   5  37.0    Male   Luxembourg     2023-11-29      Stage I             No

   smoking_status   bmi  cholesterol_level  hypertension  asthma  cirrhosis  \
0  Passive Smoker  29.4                199             0       0          1
1  Passive Smoker  41.2                280             1       1          0
2   Former Smoker  44.0                268             1       1          0
3  Passive Smoker  43.0                241             1       1          0
4  Passive Smoker  19.7                178             0       0          0

   other_cancer treatment_type end_treatment_date  survived
0             0   Chemotherapy         2017-09-10         0
1             0        Surgery         2024-06-17         1
2             0       Combined         2024-04-09         0
3             0   Chemotherapy         2017-04-23         0
4             0       Combined         2025-01-08         0
```

[7]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 890000 entries, 0 to 889999
Data columns (total 17 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   id                  890000 non-null  int64
 1   age                 890000 non-null  float64
 2   gender              890000 non-null  object
 3   country             890000 non-null  object
 4   diagnosis_date      890000 non-null  object
 5   cancer_stage        890000 non-null  object
 6   family_history      890000 non-null  object
 7   smoking_status      890000 non-null  object
 8   bmi                 890000 non-null  float64
 9   cholesterol_level   890000 non-null  int64
 10  hypertension        890000 non-null  int64
 11  asthma              890000 non-null  int64
 12  cirrhosis           890000 non-null  int64
 13  other_cancer        890000 non-null  int64
 14  treatment_type      890000 non-null  object
 15  end_treatment_date  890000 non-null  object
 16  survived            890000 non-null  int64
dtypes: float64(2), int64(7), object(8)
memory usage: 115.4+ MB
```

[8]:

```
duplicates = df.duplicated()
print (df[duplicates])
```

```
Empty DataFrame
Columns: [id, age, gender, country, diagnosis_date, cancer_stage,
family_history, smoking_status, bmi, cholesterol_level, hypertension, asthma,
cirrhosis, other_cancer, treatment_type, end_treatment_date, survived]
Index: []
```

1 Patient Demographics

1.1 We seek to answer the following questions:

1.1.1 1. What is the age distribution of diagnosed patients?

1.1.2 2. How does gender distribution vary across cancer stages?

1.1.3 3. Which countries have the highest number of cancer cases?

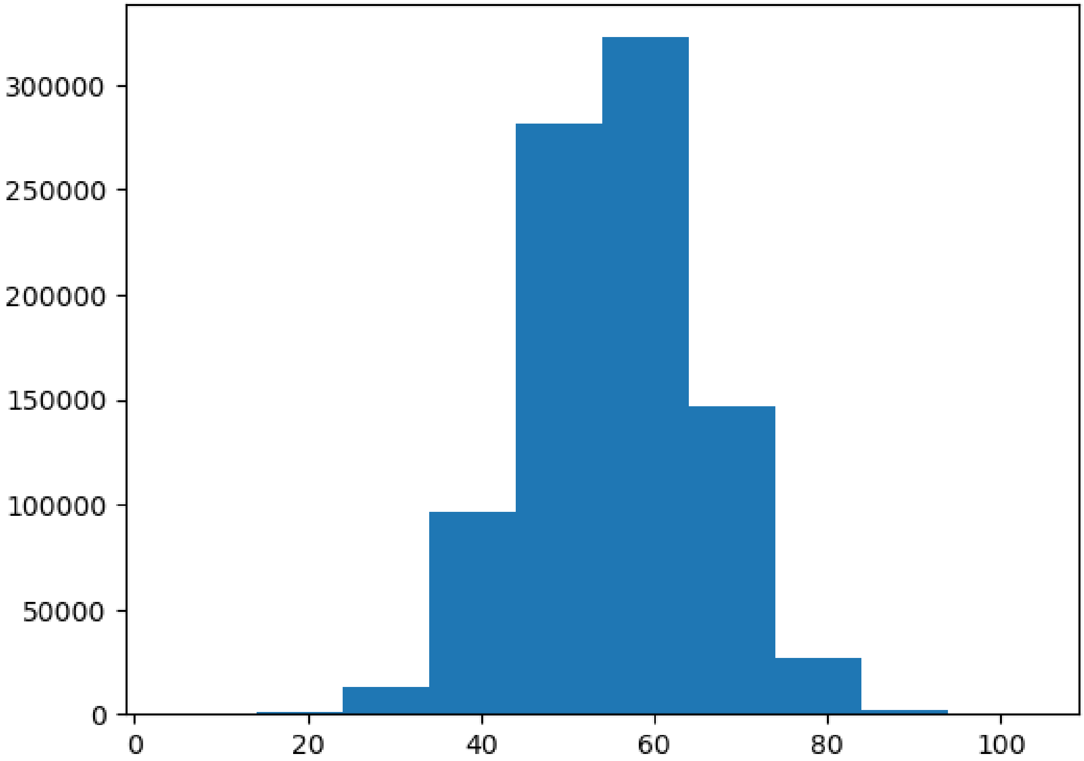
1.1.4 4. What is the average BMI by gender and age group?

[10]:

```
# we're using a histogram to see the age distribution of patients
plt.hist(df["age"])
```

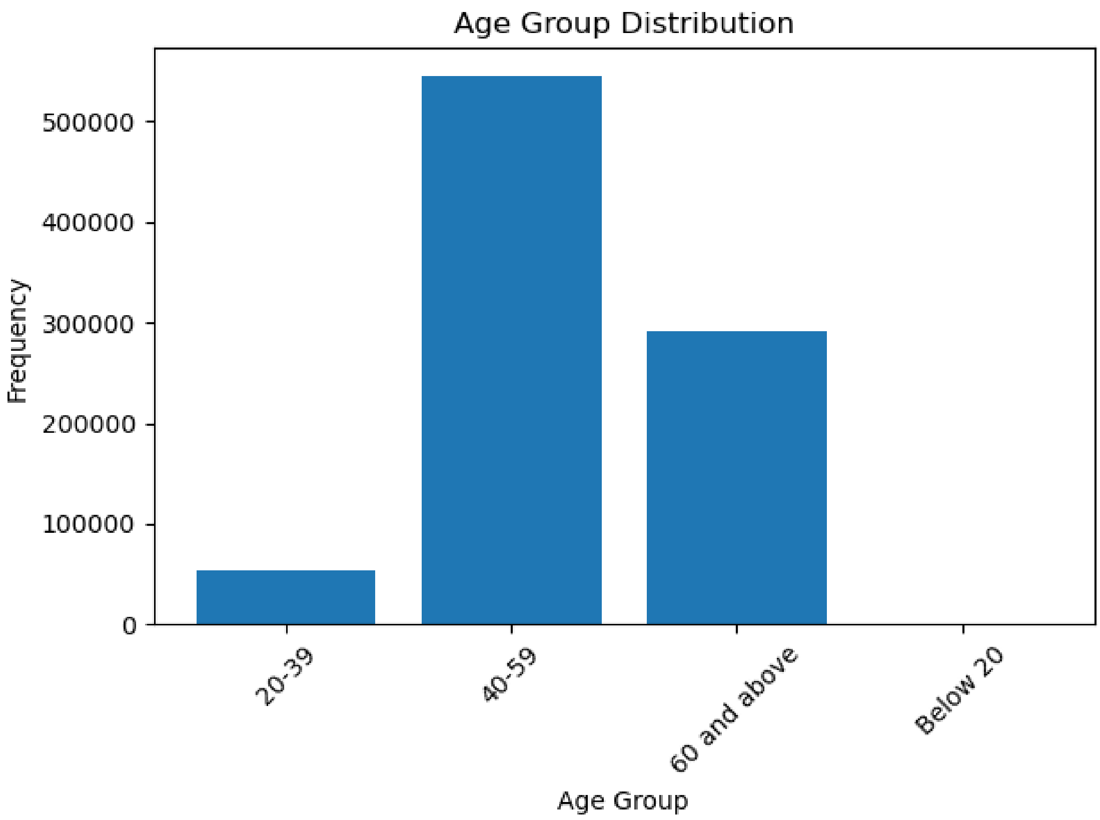
[10]:

```
(array([1.20000e+01, 7.30000e+02, 1.33220e+04, 9.64490e+04, 2.81231e+05,
        3.22483e+05, 1.47271e+05, 2.65550e+04, 1.90300e+03, 4.40000e+01]),
 array([  4.,  14.,  24.,  34.,  44.,  54.,  64.,  74.,  84.,  94., 104.]),
 <BarContainer object of 10 artists>)
```



```
[17]: df ['age_group']= np.where (df['age']<20, 'Below 20',
                        np.where (df['age']<40, '20-39',
                        np.where (df['age']<60, '40-59',
                        '60 and above')))
```

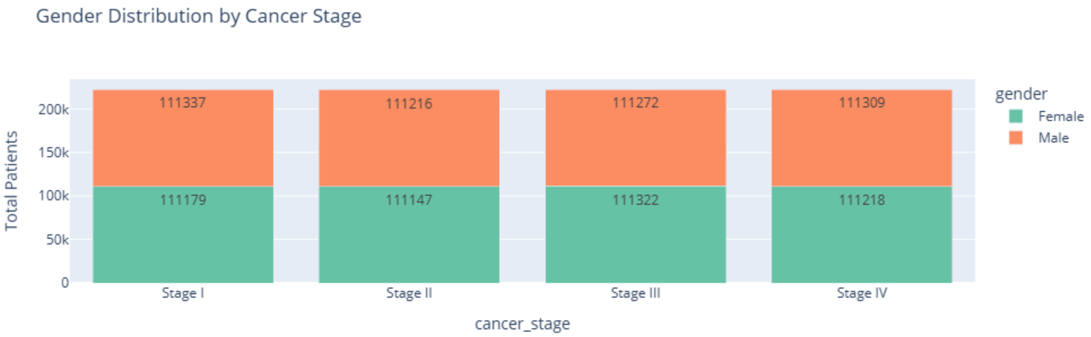
```
[19]: # Count the frequency of each age group
      age_counts = df['age_group'].value_counts().sort_index()

      # Plot
      plt.bar(age_counts.index, age_counts.values)
      plt.xlabel('Age Group')
      plt.ylabel('Frequency')
      plt.title('Age Group Distribution')
      plt.xticks(rotation=45)
      plt.tight_layout()
      plt.show()
```



```
[21]: # how gender distribution varies with cancer stages
gender_stage = df.groupby(['cancer_stage', 'gender']).size().
 ↪unstack(fill_value=0)
```

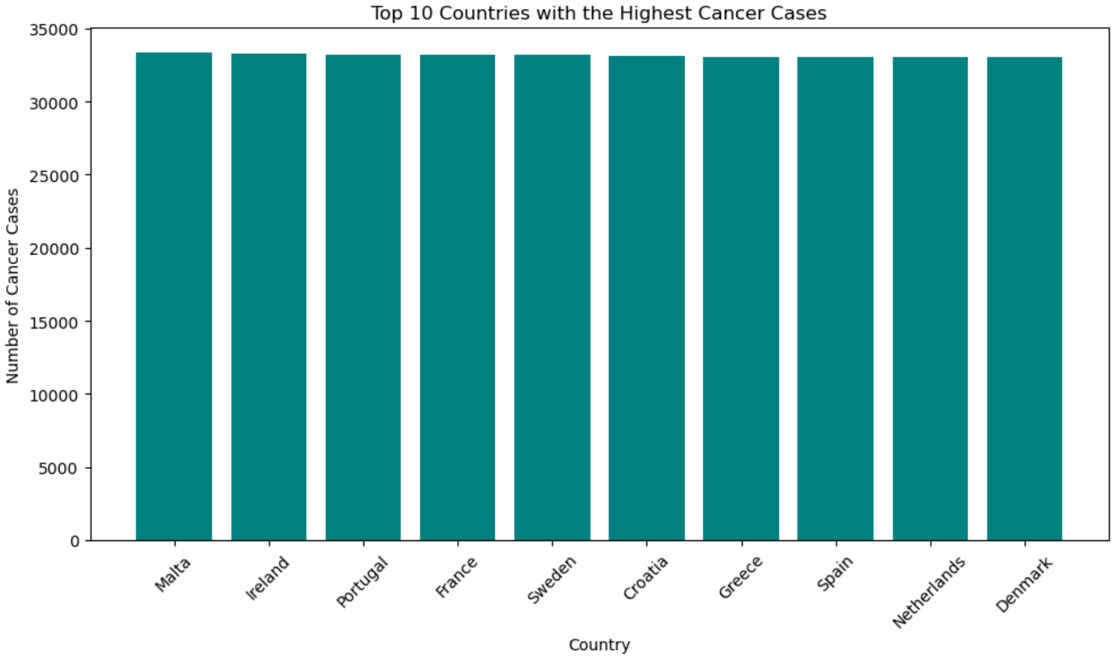
1.1.5 I opted for plotly.express because Matplotlib doesn't support hover interactivity

```
[32]: # Melt the DataFrame for Plotly
df_melted = gender_stage.reset_index().melt(id_vars='cancer_stage',␣
 ↪var_name='gender', value_name='count')

#id_vars='cancer_stage', Keep this column as it is
#var_name='gender',   New column name for what used to be column headers␣
 ↪(Female, Male)
#value_name='count'  New column for the values (e.g., 10, 5, etc.)

# Create interactive stacked bar chart
fig = px.bar(df_melted,
             x='cancer_stage',
             y='count',
             color='gender',
             text='count',
             title='Gender Distribution by Cancer Stage',
             labels={'count': 'Total Patients'},
             color_discrete_sequence=px.colors.qualitative.Set2)

fig.update_layout(barmode='stack')
fig.show()
```

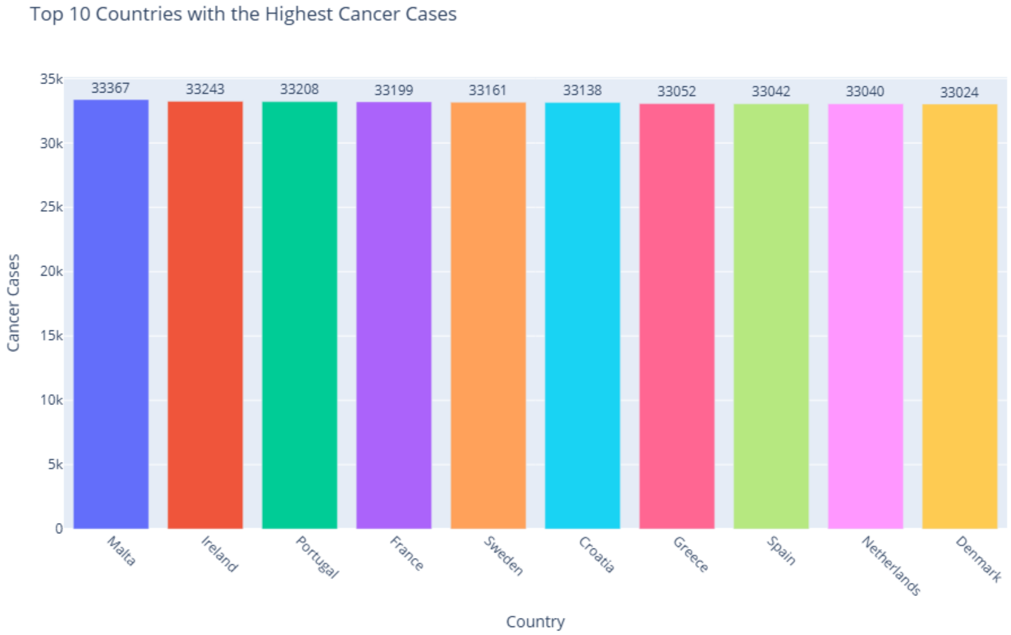


[34]:

```
top_10_countries = df['country'].value_counts().head(10)

# Plot
plt.figure(figsize=(10, 6))
plt.bar(top_10_countries.index, top_10_countries.values, color='teal')
plt.xlabel('Country')
plt.ylabel('Number of Cancer Cases')
plt.title('Top 10 Countries with the Highest Cancer Cases')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



```
[36]: # Get top 10 countries by cancer cases
      top_10_countries = df['country'].value_counts().head(10).reset_index()
      top_10_countries.columns = ['Country', 'Cancer Cases']

      # Create interactive bar chart
      fig = px.bar(top_10_countries,
                   x='Country',
                   y='Cancer Cases',
                   title='Top 10 Countries with the Highest Cancer Cases',
                   text='Cancer Cases',
                   color='Country')  # optional for color distinction

      # Improve layout and tooltip behavior
      fig.update_traces(textposition='outside')
      fig.update_layout(showlegend=False, xaxis_tickangle=45, height = 600 , width =␣
       ↪1000)

      fig.show()
```


Top 10 Countries with the Highest Cancer Cases
Cancer Cases
35k
30k
25k
20k
15k
10k
5k
0
33367
33243
33208
33199
33161
33138
33052
33042
33040
33024
Malta
Ireland
Portugal
France
Sweden
Croatia
Greece
Spain
Netherlands
Denmark
Country

[ ]: