|  |  |
| --- | --- |
| Name Of The Student | Himanshu |
| Internship Project Topic | TCS iON RIO-210: Build a Classification Model for Drug Trials Dataset |
| Name of the Organization | TCS iON |
| Name of the Industry Mentor | Himdweep Walia |
| Name of the Institute | Amity University |

|  |  |  |
| --- | --- | --- |
| Date | Day # | Hours Spent |
| 01-06-2024 | Day-39 | 6 Hours |
| Activities done during the day:  **Project Hands-on – Visualization the data By different type of charts**  **Link of the google drive google Colab file :-**  <https://colab.research.google.com/drive/1VQRq0l6oc9Uj4cOOqiuhkfS1JmpKr3fU?usp=sharing>    **Check and create a chart of the age range of the male and female members**  age\_ranges = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]  # Categorize the ages into the defined ranges  data['Age Range'] = pd.cut(data['AGE'], bins=age\_ranges)  # Group the data by age range and gender, and count the number of occurrences  grouped\_data = data.groupby(['Age Range', 'GENDER']).size().unstack()  # Set the figure size  plt.figure(figsize=(10, 6))  # Create a stacked bar plot  ax = grouped\_data.plot(kind='bar', stacked=True)  # Add labels and title  plt.xlabel('Age Range')  plt.ylabel('Count')  plt.title('Number of Males and Females in Each Age Range')  # Display the plot  plt.show() | | |
| **Check and create the chart of number of male, females in the data in age range**  ata['Age Range'] = pd.cut(data['AGE'], bins=age\_ranges).astype(str)  # Group the data by age range and gender, and count the number of occurrences  grouped\_data = data.groupby(['Age Range', 'GENDER']).size().unstack()  # Create a stacked bar plot using Plotly  fig = go.Figure()  # Add bars for males and females in each age range  for col in grouped\_data.columns:      fig.add\_trace(go.Bar(x=grouped\_data.index, y=grouped\_data[col], name=col,                           text=grouped\_data[col], textposition='auto'))  # Update the layout of the figure  fig.update\_layout(barmode='stack', xaxis\_title='Age Range', yaxis\_title='Count',                    title='Number of Males and Females in Each Age Range')  # Display the plot  fig.show() | | |

side\_effect\_counts = data['SIDE EFFECTS'].value\_counts()

plt.bar(side\_effect\_counts.index, side\_effect\_counts.values)

# Add labels and title

plt.xlabel('Side Effect')

plt.ylabel('Count')

plt.title('Number of Side Effects')

# Rotate x-axis labels if needed

plt.xticks(rotation='vertical')

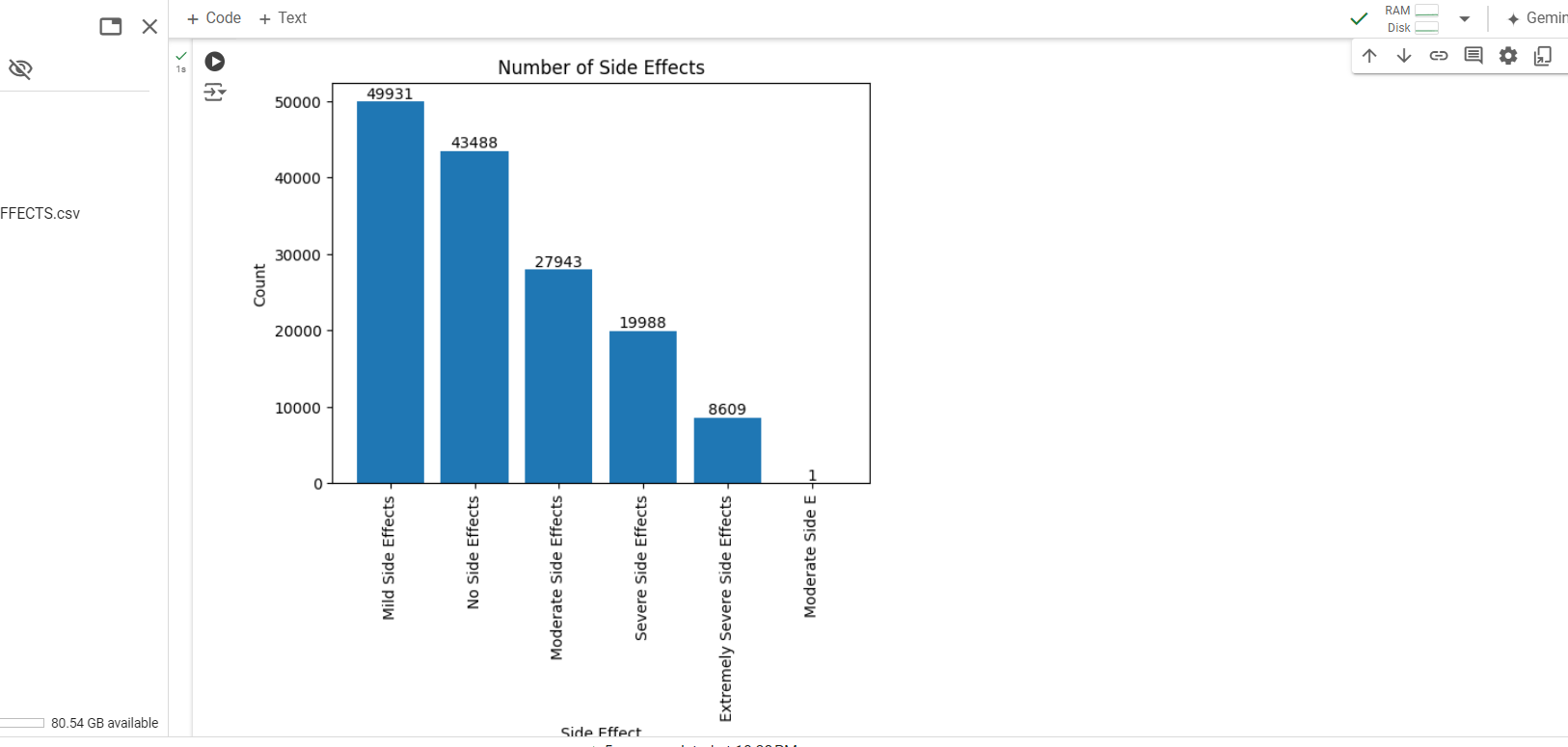
# Add number labels on top of each bar

for i, v in enumerate(side\_effect\_counts.values):

    plt.text(i, v, str(v), ha='center', va='bottom')

# Display the plot

plt.show()



race\_counts = data['RACE'].value\_counts()

fig = go.Figure(data=[go.Bar(x=race\_counts.index, y=race\_counts.values,

                            text=race\_counts.values, textposition='auto')])

# Add labels and title

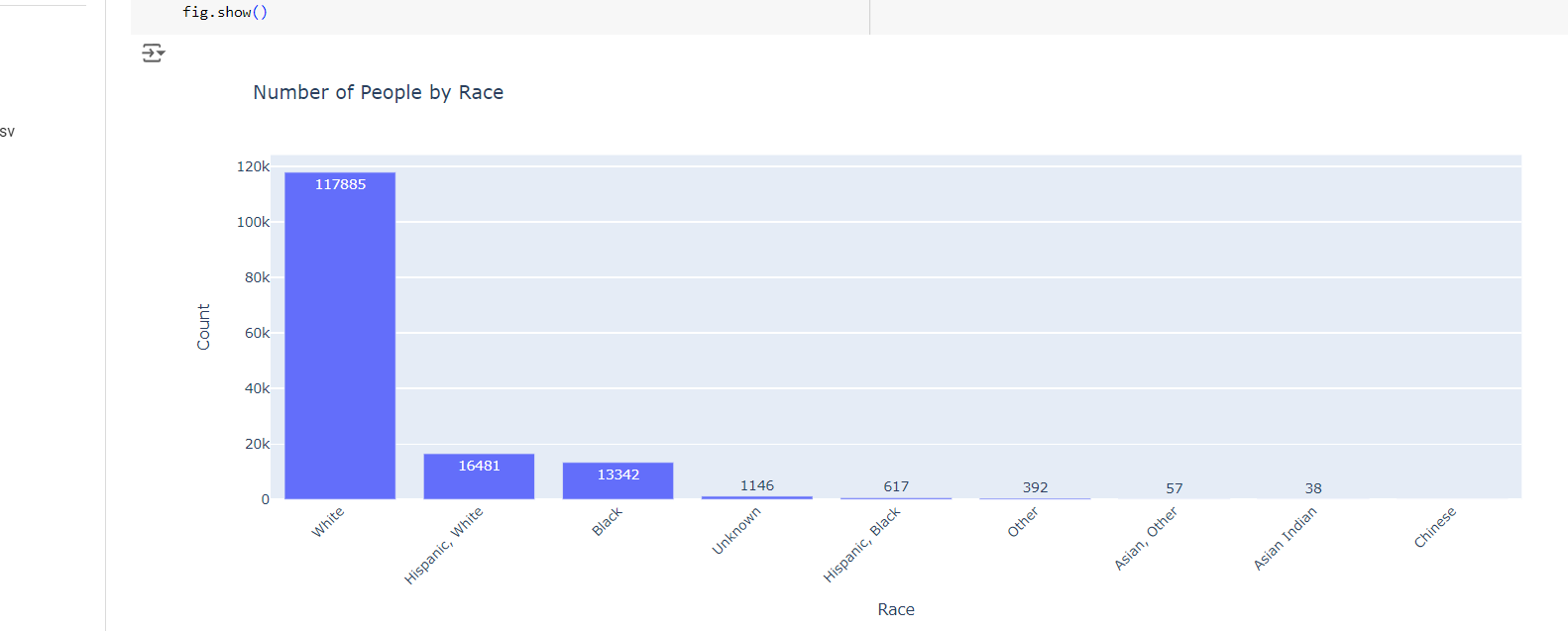
fig.update\_layout(xaxis\_title='Race', yaxis\_title='Count', title='Number of People by Race')

# Rotate x-axis labels if needed

fig.update\_layout(xaxis\_tickangle=-45)

# Display the plot

fig.show()



# Define the side effect categories

side\_effect\_categories = ['Mild Side Effects', 'No Side Effects', 'Moderate Side Effects', 'Severe Side Effects', 'Extremely Severe Side Effects']

# Iterate over each side effect category

for category in side\_effect\_categories:

    # Filter the data for the specific side effect category

    filtered\_data = data[data['SIDE EFFECTS'] == category]

    # Count the occurrences of each drug name

    drug\_counts = filtered\_data['DRUG NAME'].value\_counts()

    # Sort the drug names based on the counts

    sorted\_drugs = drug\_counts.sort\_values(ascending=False)

    # Select the top 20 drug names

    top\_20\_drugs = sorted\_drugs.head(20)

    # Create a bar plot to display the top 20 drug names

    plt.figure(figsize=(15, 6))

    plt.bar(top\_20\_drugs.index, top\_20\_drugs.values)

    # Add labels and title

    plt.xlabel('Drug Name')

    plt.ylabel('Count')

    plt.title(f'Top 20 Drug Names with {category}')

    # Rotate x-axis labels if needed

    plt.xticks(rotation='vertical')

    # Add count values on top of each bar

    for i, value in enumerate(top\_20\_drugs.values):

        plt.text(i, value, str(value), ha='center', va='bottom')

    # Display the plot

    plt.show()

