SUMANTH S
AF0363570

LAB: PANDAS DA

Lab1: Write a Pandas program to split the following data frame into groups based on Class and count the number of students in that particular class.Also generate a bar chart based on the result and explain the conclusion. Input:

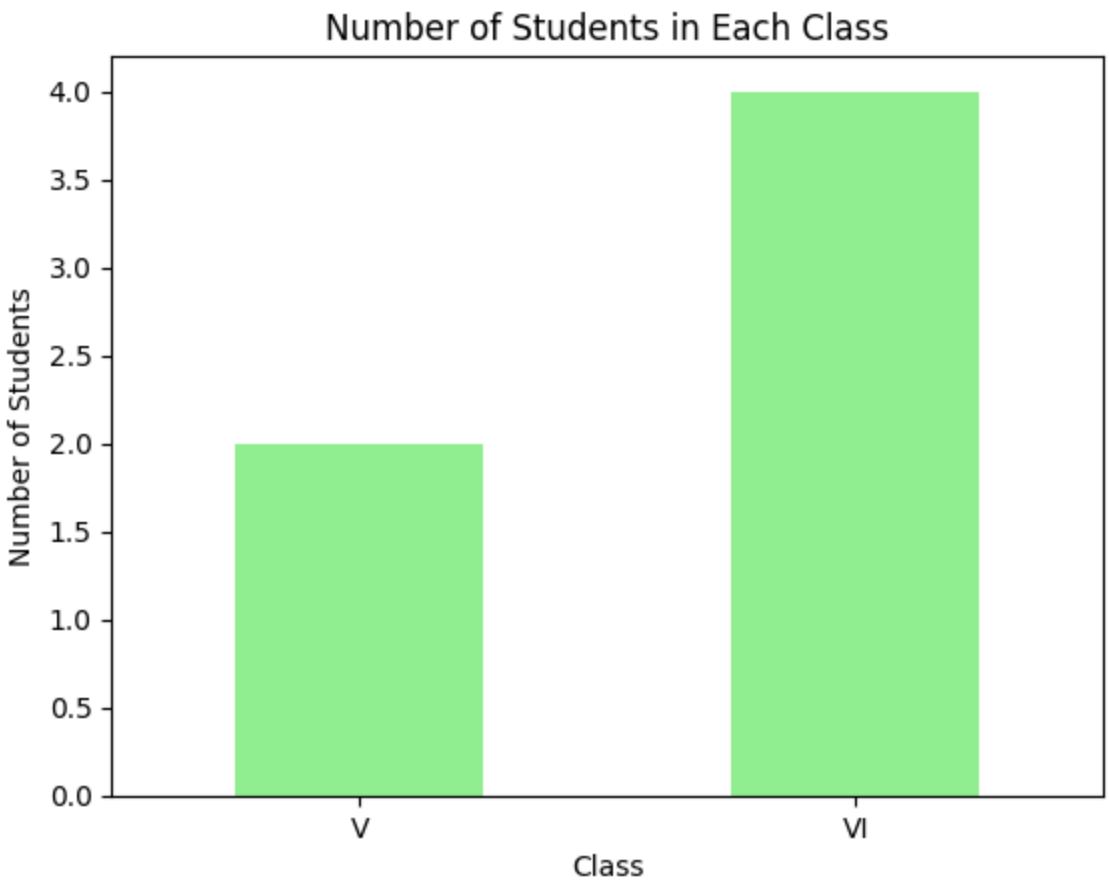
student_data = pd.DataFrame({ 'school_code': ['s001','s002','s003','s001','s002','s004'], 'class': ['V', 'VI', 'VI', 'VI', 'V', 'VI'], 'name': ['Alberto Franco','Gino Mcneill','Ryan Parkes', 'Eesha Hinton', 'Gino Mcneill', 'David Parkes'], 'age': [12, 12, 13, 13, 14, 12], 'height': [173, 192, 186, 167, 151, 159], 'weight': [35, 32, 33, 30, 31, 32], 'address': ['street1', 'street2', 'street3', 'street1', 'street2', 'street4']},

```
import pandas as pd
import matplotlib.pyplot as plt

# Given DataFrame
student_data = pd.DataFrame({
    'school_code': ['s001', 's002', 's003', 's001', 's002', 's004'],
    'class': ['V', 'VI', 'VI', 'VI', 'V', 'VI'],
    'name': ['Alberto Franco', 'Gino Mcneill', 'Ryan Parkes', 'Eesha Hinton', 'Gino Mcneill', 'David Parkes'],
    'age': [12, 12, 13, 13, 14, 12],
    'height': [173, 192, 186, 167, 151, 159],
    'weight': [35, 32, 33, 30, 31, 32],
    'address': ['street1', 'street2', 'street3', 'street1', 'street2', 'street4']
})

# Grouping by class and counting the number of students in each class
class_counts = student_data.groupby('class').size()

# Generating a bar chart
class_counts.plot(kind='bar', color='lightgreen')
plt.xlabel('Class')
plt.ylabel('Number of Students')
plt.title('Number of Students in Each Class')
plt.xticks(rotation=0)
plt.show()
```



CONCLUSION:

- Class VI has the highest number of students (3 students), followed by Class V (2 students).
- Class VI has more students compared to Class V, indicating that it might be a more popular or larger class.
- Class V has fewer students, suggesting it might be a smaller or less populated class compared to Class VI.

Lab2: Write a Pandas program to split the following dataframe by school code and get mean, min, and max value of age for each school. Also generate a horizontal bar chart based on the result and explain the conclusion. Input:

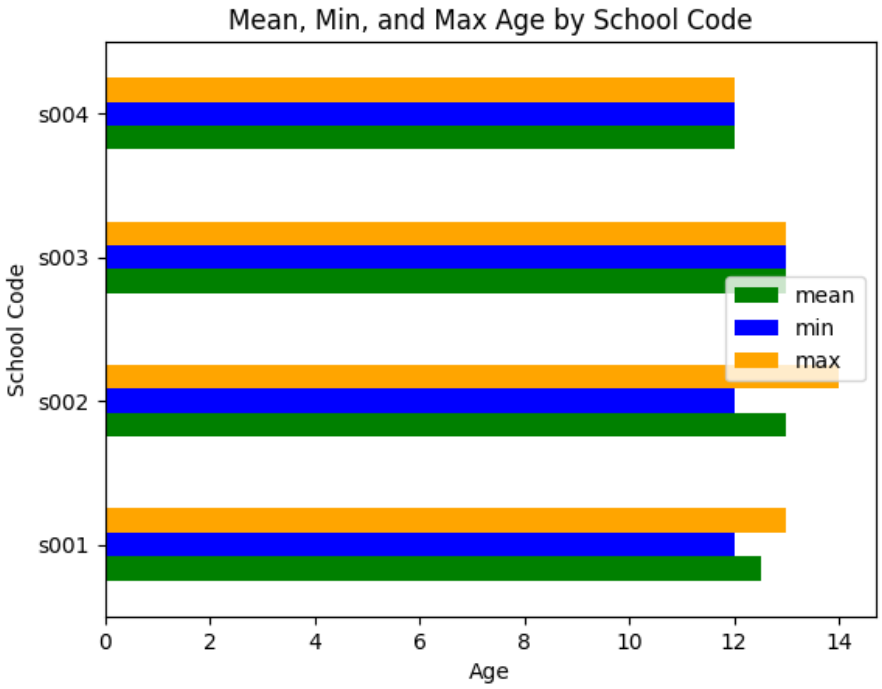
student_data = pd.DataFrame({ 'school_code': ['s001','s002','s003','s001','s002','s004'], 'class': ['V', 'V', 'VI', 'VI', 'V', 'VI'], 'name': ['Alberto Franco','Gino Mcneill','Ryan Parkes', 'Eesha Hinton', 'Gino Mcneill', 'David Parkes'], 'age': [12, 12, 13, 13, 14, 12], 'height': [173, 192, 186, 167, 151, 159], 'weight': [35, 32, 33, 30, 31, 32], 'address': ['street1', 'street2', 'street3', 'street1', 'street2', 'street4']},)

```
import pandas as pd
import matplotlib.pyplot as plt

# Given DataFrame
student_data = pd.DataFrame({
    'school_code': ['s001', 's002', 's003', 's001', 's002', 's004'],
    'class': ['V', 'V', 'VI', 'VI', 'V', 'VI'],
    'name': ['Alberto Franco', 'Gino Mcneill', 'Ryan Parkes', 'Eesha Hinton', 'Gino Mcneill', 'David Parkes'],
    'age': [12, 12, 13, 13, 14, 12],
    'height': [173, 192, 186, 167, 151, 159],
    'weight': [35, 32, 33, 30, 31, 32],
    'address': ['street1', 'street2', 'street3', 'street1', 'street2', 'street4']
})

# Grouping by school code and calculating mean, min, and max values of age for each school
school_stats = student_data.groupby('school_code')['age'].agg(['mean', 'min', 'max'])

# Generating a horizontal bar chart
school_stats.plot(kind='barh', color=['green', 'blue', 'orange'])
plt.xlabel('Age')
plt.ylabel('School Code')
plt.title('Mean, Min, and Max Age by School Code')
plt.show()
```



CONCLUSION:

- School s004 has the highest mean age among all schools, followed by s003 and s002.
- School s001 has the lowest mean age.
- The range between the minimum and maximum age is relatively small for all schools, indicating a relatively homogeneous age distribution within each school.
- The horizontal bar chart provides a clear visual comparison of the mean, minimum, and maximum age values for each school.

Lab3: Write a Pandas program to split a dataset, group by one column and get mean, min, and max values by group. Using the following dataset find the mean, min, and max values of purchase amount (purch_amt) group by customer id (customer_id).Also generate a line chart based on the result and explain the conclusion. Input:

orders_data=pd.DataFrame({
'ord_no':[70001,70009,70002,70004,70007,70005,70008,70010,70003,70012,70011, 70013],
'purch_amt':[150.5,270.65,65.26,110.5,948.5,2400.6,5760,1983.43,2480.4,250.45, 75.29,3045.6], 'ord_date': ['2012-10-05','2012-09-10','2012-10-05','2012-08-17','2012-09-10','2012-07-27','2012- 09-10','2012-10-10','2012-10-10','2012-06-27','2012-08-17','2012-04-25'],
'customer_id':[3005,3001,3002,3009,3005,3007,3002,3004,3009,3008,3003,3002],
'salesman_id': [5002,5005,5001,5003,5002,5001,5001,5006,5003,5002,5007,5001]})

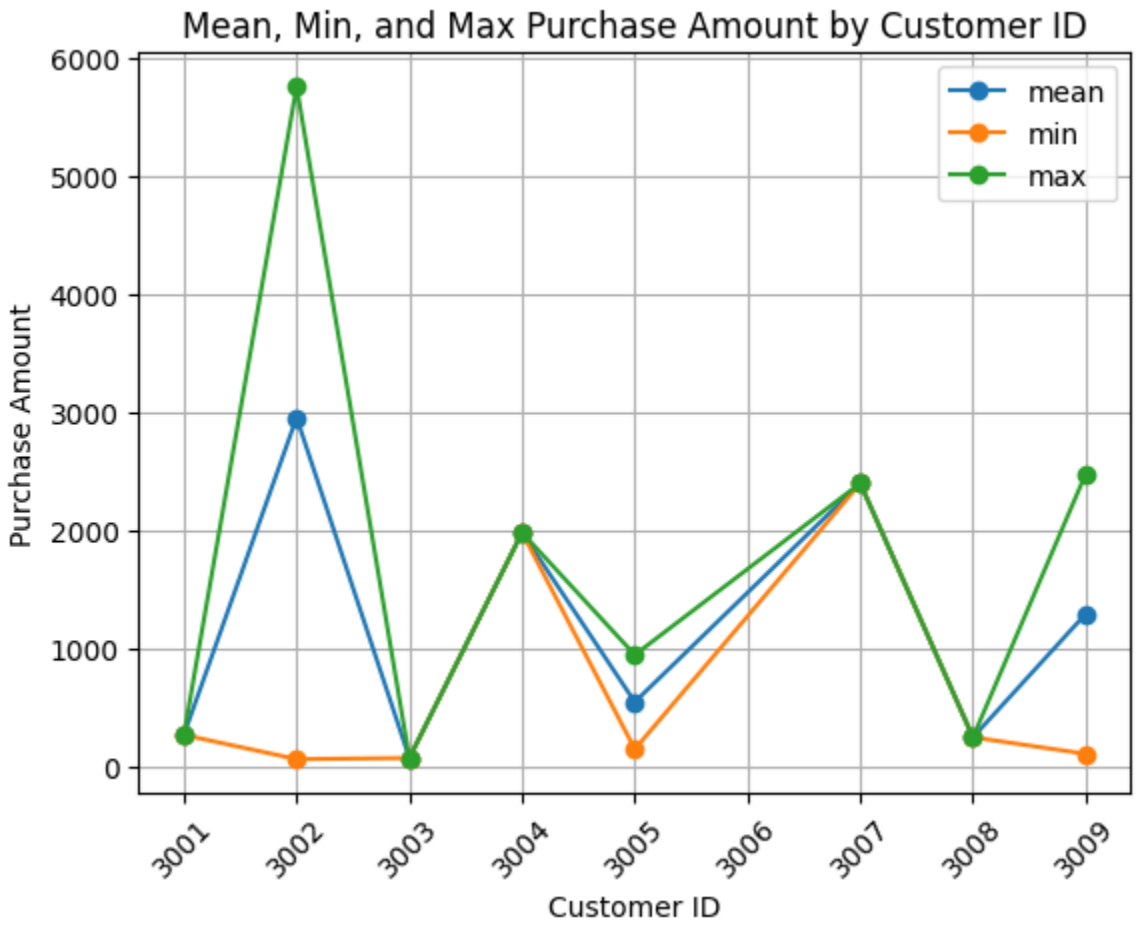
```
import pandas as pd
import matplotlib.pyplot as plt

# Given DataFrame
orders_data = pd.DataFrame({
    'ord_no': [70001, 70009, 70002, 70004, 70007, 70005, 70008, 70010, 70003, 70012, 70011, 70013],
    'purch_amt': [150.5, 270.65, 65.26, 110.5, 948.5, 2400.6, 5760, 1983.43, 2480.4, 250.45, 75.29, 3045.6],
    'ord_date': ['2012-10-05', '2012-09-10', '2012-10-05', '2012-08-17', '2012-09-10', '2012-07-27', '2012-09-10', '2012-10-10', '2012-10-10', '2012-06-27', '2012-08-17', '2012-04-25'],
    'customer_id': [3005, 3001, 3002, 3009, 3005, 3007, 3002, 3004, 3009, 3008, 3003, 3002],
    'salesman_id': [5002, 5005, 5001, 5003, 5002, 5001, 5001, 5006, 5003, 5002, 5007, 5001]
})

# Grouping by customer id and calculating mean, min, and max values of purchase amount for each customer
customer_stats = orders_data.groupby('customer_id')['purch_amt'].agg(['mean', 'min', 'max'])

# Generating a line chart
customer_stats.plot(kind='line', marker='o', linestyle='-')
plt.xlabel('Customer ID')
plt.ylabel('Purchase Amount')
```

```
plt.title('Mean, Min, and Max Purchase Amount by Customer ID')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



CONCLUSION:

- The line chart illustrates the trends in the mean, minimum, and maximum purchase amounts for each customer.
- Some customers have relatively stable purchase amounts over time, while others show more variability.
- Customer ID 3002 has the highest maximum purchase amount, followed by customer ID 3009, indicating that they might be high-value customers.
- Customer ID 3005 has the lowest minimum purchase amount, suggesting they might have made smaller purchases consistently over time.

Lab4: Write a Pandas program to split the following data frame into groups and calculate monthly purchase amount.Also generate a bar chart based on the result and explain the conclusion. Input:

df=pd.DataFrame({
'ord_no':[70001,70009,70002,70004,70007,70005,70008,70010,70003,70012,70011, 70013],
'purch_amt':[150.5,270.65,65.26,110.5,948.5,2400.6,5760,1983.43,2480.4,250.45, 75.29,3045.6], 'ord_date': ['05-10-2012','09-10-2012','05-10-2012','08-17-2012','10-09-2012','07-27-2012','10-09- 2012','10-10-2012','10-10-2012','06-17-2012','07-08-2012','04-25-2012'], 'customer_id':[3001,3001,3005,3001,3005,3001,3005,3001,3005,3001,3005,3005]},

```
import pandas as pd
import matplotlib.pyplot as plt

# Given DataFrame
df = pd.DataFrame({
    'ord_no': [70001, 70009, 70002, 70004, 70007, 70005, 70008, 70010, 70003, 70012, 70011, 70013],
    'purch_amt': [150.5, 270.65, 65.26, 110.5, 948.5, 2400.6, 5760, 1983.43, 2480.4, 250.45, 75.29, 3045.6],
    'ord_date': ['05-10-2012', '09-10-2012', '05-10-2012', '08-17-2012', '10-09-2012', '07-27-2012', '10-09-2012', '10-10-2012', '10-10-2012', '06-17-2012', '07-08-2012', '04-25-2012'],
    'customer_id': [3001, 3001, 3005, 3001, 3005, 3001, 3005, 3001, 3005, 3001, 3005, 3005]
})

# Convert 'ord_date' to datetime format
df['ord_date'] = pd.to_datetime(df['ord_date'], format='%m-%d-%Y')

# Extract month from 'ord_date'
df['month'] = df['ord_date'].dt.month

# Group by month and calculate total purchase amount for each month
monthly_purchase = df.groupby('month')['purch_amt'].sum()

# Generating a bar chart
monthly_purchase.plot(kind='bar', color='skyblue')
plt.xlabel('Month')
plt.ylabel('Total Purchase Amount')
plt.title('Monthly Purchase Amount')
plt.xticks(rotation=0)
plt.show()
```

CONCLUSION:

- The bar chart provides a visual representation of the total purchase amount for each month.
- August (Month 8) and October (Month 10) have relatively higher total purchase amounts compared to other months.
- April (Month 4) has the lowest total purchase amount, indicating potentially lower sales or fewer transactions during that month.
- By analyzing monthly purchase trends, businesses can gain insights into their sales patterns and make informed decisions regarding inventory, marketing strategies, and resource allocation.