

```
# -*- coding: utf-8 -*-
```

```
"""Untitled3.ipynb
```

Automatically generated by Colab.

Original file is located at

<https://colab.research.google.com/drive/1H8hpQD0HWW8oPLRUJYiH9bWmKnJlX9ww>

```
"""
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from google.colab import files
upload = files.upload()
```

```
df = pd.read_csv('uni_gym.csv')
print(df.head())
```

```
df.info()
```

```
df['1)- if there was a gym in the university how often would you visit ?
'].describe()
```

```
df.sort_values(by=['1)- if there was a gym in the university how often would you
visit ? '])
```

```
column_name = '1)- if there was a gym in the university how often would you visit ?
'
```

```
value_counts = df[column_name].value_counts()
```

```
plt.figure(figsize=(8, 8))
plt.pie(value_counts, labels=value_counts.index, autopct='%1.1f%%', startangle=90)
plt.title('Frequency of Gym Visits')
plt.axis('equal')
plt.show()
```

```
column_name = '2)- How long would your workout session be during the university day
?'
```

```
value_counts = df[column_name].value_counts()
```

```
plt.figure(figsize=(8, 8))
plt.pie(value_counts, labels=value_counts.index, autopct='%1.1f%%', startangle=90)
plt.title('workout session time')
plt.axis('equal')
plt.show()
```

```
column_name = '3)- what type of training would you do when you visit the gym ? '
```

```
value_counts = df[column_name].value_counts()
```

```
plt.figure(figsize=(8, 8))
plt.pie(value_counts, labels=value_counts.index, autopct='%1.1f%%', startangle=90)
plt.title('workout type')
plt.axis('equal')
plt.show()
```

```

column_name = '4)- what is your biggest challenge in maintaining a gym routine ? '
value_counts = df[column_name].value_counts()

plt.figure(figsize=(8, 8))
plt.pie(value_counts, labels=value_counts.index, autopct='%1.1f%%', startangle=90)
plt.title('biggest gym challenge')
plt.axis('equal')
plt.show()

column_name = '5)- how would you like the university gym trainer to measure your
progress ? '
value_counts = df[column_name].value_counts()

plt.figure(figsize=(8, 8))
plt.pie(value_counts, labels=value_counts.index, autopct='%1.1f%%', startangle=90)
plt.title('progress measuring')
plt.axis('equal')
plt.show()

column_name = '1)- if there was a gym in the university how often would you visit ? '
value_counts = df[column_name].value_counts()

plt.figure(figsize=(10, 6))
value_counts.plot(kind='bar')
plt.title('Frequency of Gym Visits')
plt.xlabel('How Often Would You Visit?')
plt.ylabel('Number of Students')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

column_name = '2)- How long would your workout session be during the university day
?'
value_counts = df[column_name].value_counts()

plt.figure(figsize=(10, 6))
value_counts.plot(kind='bar')
plt.title('Workout Session Duration')
plt.xlabel('Workout Session Length')
plt.ylabel('Number of Students')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

column_name = '3)- what type of training would you do when you visit the gym ? '
value_counts = df[column_name].value_counts()

plt.figure(figsize=(12, 6))
value_counts.plot(kind='bar')
plt.title('Types of Training at the Gym')
plt.xlabel('Training Type')
plt.ylabel('Number of Students')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

```

```

column_name = '4)- what is your biggest challenge in maintaining a gym routine ? '
value_counts = df[column_name].value_counts()

plt.figure(figsize=(12, 6))
value_counts.plot(kind='bar')
plt.title('Biggest Challenge in Maintaining a Gym Routine')
plt.xlabel('Challenge')
plt.ylabel('Number of Students')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

plt.figure(figsize=(10, 6))
sns.countplot(x=column_name, data=df)
plt.title(' progress measuring')
plt.xlabel('Frequency')
plt.ylabel('Number of Students')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

def describe_data(df, column_name):
    print(f"Function called with column: {column_name}") # Debugging line

    try:
        if column_name not in df.columns:
            raise KeyError(f"Error: Column '{column_name}' not found in the
DataFrame.")

        if not pd.api.types.is_numeric_dtype(df[column_name]):
            raise TypeError(f"Error: Cannot calculate statistics for non-numerical
column '{column_name}'.")

        desc = df[column_name].describe()
        print(f"\nDescriptive Statistics for '{column_name}':\n{desc}\n")
        print(f"Median: {df[column_name].median()}")
        print(f"Standard Deviation: {df[column_name].std()}")
        print(f"Minimum: {df[column_name].min()}")
        print(f"Maximum: {df[column_name].max()}")

    except KeyError as e:
        print(e)
    except TypeError as e:
        print(e)

```