Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

**\_01\_**

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| 1 | You work at a public library that wants to digitize its book catalog and membership management. Your task is to create a Library Management System using Google Colab to allow librarians to add/edit book records and manage member information. |
| 2 | You are a data analyst at a real estate company. You have been given a dataset of housing sale prices in different regions over the past 5 years. Load the data into Pandas and preprocess it by handling missing values and formatting columns. |
| 3 | You work for an e-commerce company and have been given a dataset with information on customer orders over the past year. Load the data into Pandas, analyze it using methods like .info(), .describe(), Which products have the highest/lowest sales? Which customer segments spend the most |
| 4 | You are a data analyst working for an automobile company. You have been provided with the Vega dataset which contains details on different vehicle models like price, engine size, horsepower, dimensions etc. |

Submitted On:

**Date: \_\_\_\_\_18/02/2024\_\_\_\_\_**

1. You work at a public library that wants to digitize its book catalog and membership management. Your task is to create a Library Management System using Google Colab to allow librarians to add/edit book records and manage member information.

import pandas as pd

import ipywidgets as widgets

books\_df = pd.DataFrame(columns=['Title', 'Author', 'Genre', 'Year'])

def add\_book(title, author, genre, year):

    global books\_df

    books\_df = books\_df.append({'Title': title, 'Author': author, 'Genre': genre, 'Year': year}, ignore\_index=True)

def edit\_book(index, title, author, genre, year):

    global books\_df

    books\_df.loc[index] = {'Title': title, 'Author': author, 'Genre': genre, 'Year': year}

def delete\_book(index):

    global books\_df

    books\_df.drop(index, inplace=True)

title\_text = widgets.Text(description='Title:', style={'description\_width': 'initial'})

author\_text = widgets.Text(description='Author:', style={'description\_width': 'initial'})

genre\_text = widgets.Text(description='Genre:', style={'description\_width': 'initial'})

year\_text = widgets.Text(description='Year:', style={'description\_width': 'initial'})

add\_book\_button = widgets.Button(description='Add Book')

edit\_book\_button = widgets.Button(description='Edit Book')

delete\_book\_button = widgets.Button(description='Delete Book')

def handle\_add\_book(button):

    add\_book(title\_text.value, author\_text.value, genre\_text.value, year\_text.value)

    print('Book record added successfully!')

def handle\_edit\_book(button):

    # Get the selected index from the dropdown

    index = dropdown\_index.value

    edit\_book(index, title\_text.value, author\_text.value, genre\_text.value, year\_text.value)

    print('Book record edited successfully!')

def handle\_delete\_book(button):

    # Get the selected index from the dropdown

    index = dropdown\_index.value

    delete\_book(index)

    print('Book record deleted successfully!')

add\_book\_button.on\_click(handle\_add\_book)

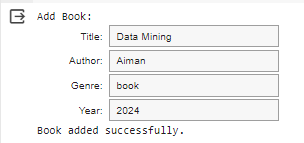
edit\_book\_button.on\_click(handle\_edit\_book)

delete\_book\_button.on\_click(handle\_delete\_book)

dropdown\_index = widgets.Dropdown(options=books\_df.index, description='Select Book:', style={'description\_width': 'initial'})

book\_widgets = widgets.VBox([title\_text, author\_text, genre\_text, year\_text, add\_book\_button, edit\_book\_button, delete\_book\_button, dropdown\_index])

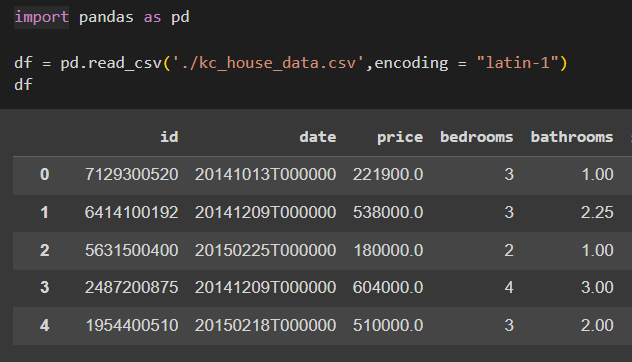
book\_widgets

 A screenshot of a computer

Description automatically generated

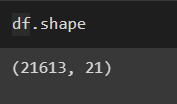
2 - You are a data analyst at a real estate company. You have been given a dataset of housing sale prices in different regions over the past 5 years. Load the data into Pandas and preprocess it by handling missing values and formatting columns.

* Read csv Info()

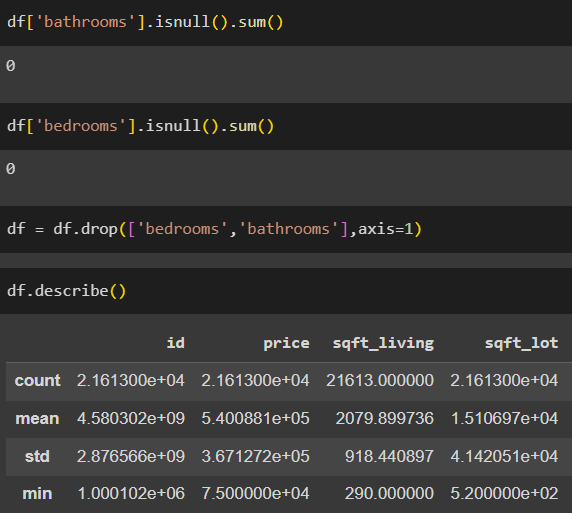
 A screenshot of a computer

Description automatically generated

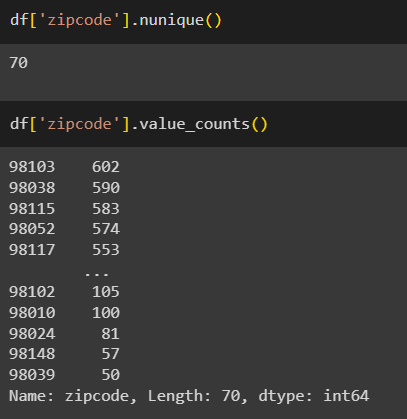
Shape



Describe after dropping some columns

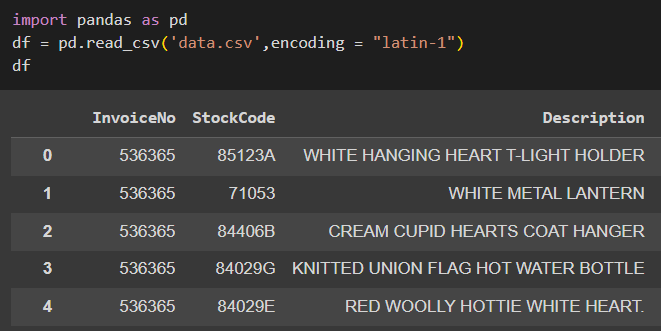


Finding Unique and repeated values



1. You work for an e-commerce company and have been given a dataset with information on customer orders over the past year. Load the data into Pandas, analyze it using methods like .info(), .describe(), Which products have the highest/lowest sales? Which customer segments spend the most?

* Read Csv



* Finding and removing Null values

A screenshot of a computer program

Description automatically generated A screenshot of a computer program

Description automatically generated

* Finnding Duplicates

A screenshot of a computer

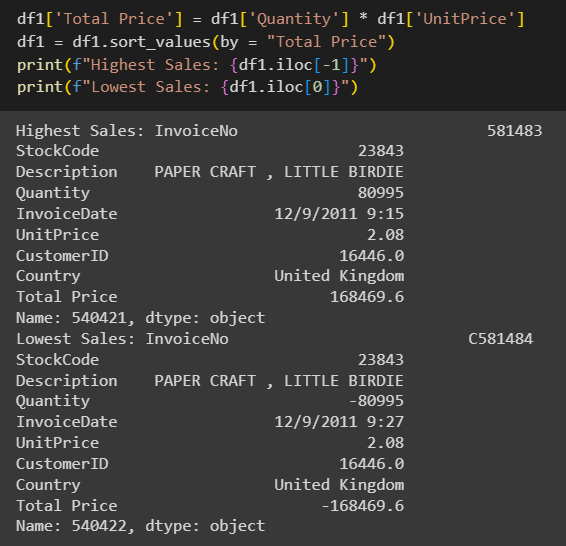
Description automatically generated

Dropping Duplicates

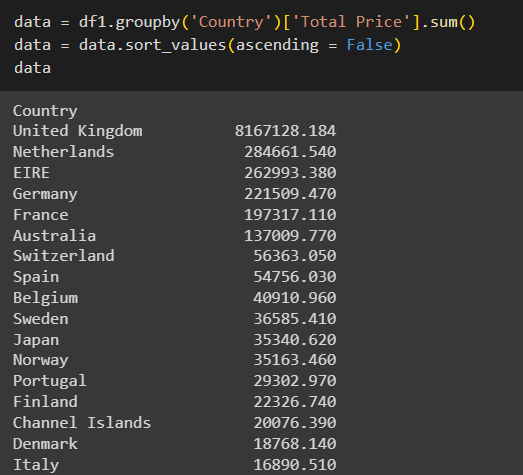
A screenshot of a computer

Description automatically generated

Finding Highest and lowest sales



Finding Segmentation wrt Country



1. You are a data analyst working for an automobile company. import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from vega\_datasets import data

import altair as alt

vega\_df = data.cars()

vega\_df.columns

vega\_df.head()

plt.figure(figsize=(4,4))

plt.scatter(vega\_df['Displacement'],vega\_df['Acceleration'],alpha = 0.5) #alpha = blurry effect

plt.title("Scatter Plot Displacement vs Acceleration")

plt.ylabel('Displacement')

plt.xlabel('Acceleration')

plt.show()

plt.figure(figsize = (4,4))

sns.histplot(vega\_df['Horsepower'],kde = True) #kde = line

plt.title("Hist Plot Horsepower")

plt.ylabel('Frequency')

plt.xlabel('Horsepower')

plt.show()

origin\_stats = vega\_df.groupby('Origin').agg({'Weight\_in\_lbs':['mean','median','std']}).reset\_index()

origin\_stats.columns = ['Origin','Mean Weight','Median Weight','Weight Std']

alt.Chart(origin\_stats).mark\_bar().encode(

    x = 'Origin',

    y = 'Mean Weight',

    color = alt.Color('Origin',legend = None),

    tooltip = ['Origin','Mean Weight','Median Weight','Weight Std']

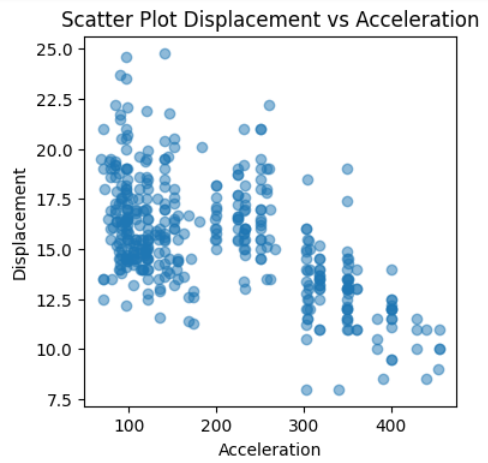
).properties(

    height = 300,

    width = 400,

    title = "Altair Plot of Groupby wrt Weight"

).interactive()

 A graph of a number of blue lines

Description automatically generated with medium confidence

A graph of a number of people

Description automatically generated with medium confidence