```
#Linear Regression Libaries

import numpy as np #numerical python
import pandas as pd #data processing
import seaborn as sns #for regression plot
from statsmodels.formula.api import ols


df = pd.read_csv('/content/bert_pares.csv') #upload the data


#Check the dataset

df.head(5) #print the first 5 rows
```

	days	sales	pares_kawali	pares_flowers
0	1	9524.0	1008.0	609.0
1	2	19716.0	1041.0	529.0
2	3	6354.0	1625.0	281.0
3	4	18706.0	1143.0	1591.0
4	5	11462.0	1875.0	1629.0

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
#check the names of columns
df.columns
```

```
Index(['days', 'sales', 'pares_kawali', 'pares_flowers'], dtype='object')
```

```
df.dtypes
```

	0
days	int64
sales	float64
pares_kawali	float64
pares_flowers	float64

dtype: object

```
df.shape #(rows, columns)
```

```
(30, 4)
```

```
#Count ths missing values
df.isnull().sum()
```

	0
days	0
sales	3
pares_kawali	3
pares_flowers	1

dtype: int64

```
#Replace the missing values with mean

df['sales'] = df['sales'].fillna(df['sales'].mean())
df['pares_kawali'] = df['pares_kawali'].fillna(df['pares_kawali'].mean())
df['pares_flowers'] = df['pares_flowers'].fillna(df['pares_flowers'].mean())


#Check for the missing values

df.isnull().sum()
```

	0
days	0
sales	0
pares_kawali	0
pares_flowers	0

dtype: int64

```
#Compute for Summary Measures

df.describe()
```

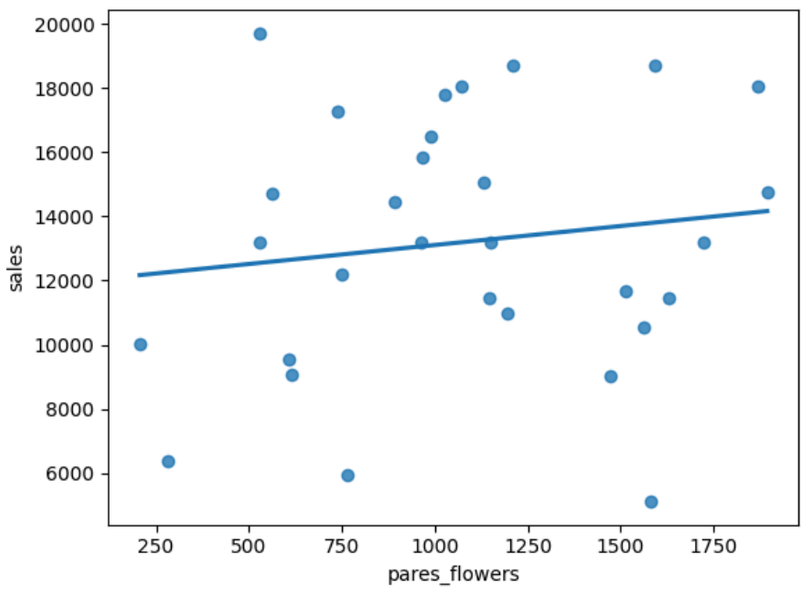
	days	sales	pares_kawali	pares_flowers
count	30.000000	30.000000	30.000000	30.000000
mean	15.500000	13193.037037	1201.777778	1072.034483
std	8.803408	3981.672145	509.811347	464.005535
min	1.000000	5124.000000	206.000000	206.000000
25%	8.250000	10662.500000	853.500000	741.250000
50%	15.500000	13193.037037	1201.777778	1049.017241
75%	22.750000	16333.000000	1599.750000	1503.500000
max	30.000000	19716.000000	1949.000000	1895.000000

```
#Compute for correlation coeffcient. Select the two variables the have the highest correlation (exclude days in the select
#your y is sales, select the appropriate x
df.corr()
```

	days	sales	pares_kawali	pares_flowers
days	1.000000	0.014172	0.200862	0.105899
sales	0.014172	1.000000	-0.151917	0.137831
pares_kawali	0.200862	-0.151917	1.000000	-0.003028
pares_flowers	0.105899	0.137831	-0.003028	1.000000

```
#generate a regression plot
sns.regplot(x="pares_flowers", y="sales", data=df, ci=None)
```

```
<Axes: xlabel='pares_flowers', ylabel='sales'>
```




```
#model: sales = intercept + slope * x

df_cleaned = df.fillna(df.mean())

X = df_cleaned[['pares_flowers']]
y = df_cleaned['sales']

model = LinearRegression()
model.fit(X, y)

intercept = model.intercept_
slope = model.coef_[0]
```

```
print(f"Model equation: sales = {intercept:.2f} + {slope:.2f} * pares_flowers")

#project the sales for the 32nd day using linear regression

pares_flowers_32 = df_cleaned['pares_flowers'].mean()
predicted_sales = model.predict([[pares_flowers_32]])

print(f"Predicted sales for the 32nd day: {predicted_sales[0]:.2f}")
```

```
Model equation: sales = 11925.10 + 1.18 * pares_flowers
Predicted sales for the 32nd day: 13193.04
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature n
  warnings.warn(
```

End of the Exam

Start coding or generate with AI.

```
# -*- coding: utf-8 -*-
"""CANTONES_FinalExam.ipynb

Automatically generated by Colab.

Original file is located at

https://colab.research.google.com/drive/1O8f13rxSpCH7IWhrYfgdiPop7Q1icSVb
"""

#Linear Regression Libaries

import numpy as np #numerical python
import pandas as pd #data processing
import seaborn as sns #for regression plot
from statsmodels.formula.api import ols

df = pd.read_csv('/content/bert_pares.csv') #upload the data

#Check the dataset

df.head(5) #print the first 5 rows

#check the names of columns
df.columns

df.dtypes

df.shape #(rows, columns)

#Count ths missing values
df.isnull().sum()

#Replace the missing values with mean

df['sales'] = df['sales'].fillna(df['sales'].mean())
df['pares_kawali'] = df['pares_kawali'].fillna(df['pares_kawali'].mean())
df['pares_flowers'] =
df['pares_flowers'].fillna(df['pares_flowers'].mean())

#Check for the missing values

df.isnull().sum()

#Compute for Summary Measures

df.describe()

#Compute for correlation coeffcient. Select the two variables the have
the highest correlation (exclude days in the selection)
#your y is sales, select the appropriate x
df.corr()

#generate a regression plot
```

```
sns.regplot(x="pares_flowers", y="sales", data=df, ci=None)

#model: sales = intercept + slope * x

df_cleaned = df.fillna(df.mean())

X = df_cleaned[['pares_flowers']]
y = df_cleaned['sales']

model = LinearRegression()
model.fit(X, y)

intercept = model.intercept_
slope = model.coef_[0]

print(f"Model equation: sales = {intercept:.2f} + {slope:.2f} *
pares_flowers")

#project the sales for the 32nd day using linear regression

pares_flowers_32 = df_cleaned['pares_flowers'].mean()
predicted_sales = model.predict([[pares_flowers_32]])

print(f"Predicted sales for the 32nd day: {predicted_sales[0]:.2f}")

"""##End of the Exam"""
```

{ "nbformat": 4, "nbformat_minor": 0, "metadata": { "colab": { "provenance": [] }, "kernelspec": { "name": "python3", "display_name": "Python 3" }, "language_info": { "name": "python" } }, "cells": [{ "cell_type": "code", "execution_count": 1, "metadata": { "id": "QuloJA5S9lYf" }, "outputs": [], "source": ["#Linear Regression Libaries\n", "\n", "import numpy as np #numerical python\n", "import pandas as pd #data processing\n", "import seaborn as sns #for regression plot\n", "from statsmodels.formula.api import ols"] }, { "cell_type": "code", "source": ["df = pd.read_csv('/content/bert_pares.csv') #upload the data"], "metadata": { "id": "H-LtD-dY-Jua" }, "execution_count": 2, "outputs": [] }, { "cell_type": "code", "source": ["#Check the dataset\n", "\n", "df.head(5) #print the first 5 rows"], "metadata": { "colab": { "base_uri": "https://localhost:8080/", "height": 206 }, "id": "yvkwBlwT-jgg", "outputId": "b8b4914d-d05d-41c6-c387-0df0f06f579f" }, "execution_count": 3, "outputs": [{ "output_type": "execute_result", "data": { "text/plain": [" days sales pares_kawali pares_flowers\n", "0 1 9524.0 1008.0 609.0\n", "1 2 19716.0 1041.0 529.0\n", "2 3 6354.0 1625.0 281.0\n", "3 4 18706.0 1143.0 1591.0\n", "4 5 11462.0 1875.0 1629.0"], "text/html": ["\n", "
\n", "
\n", "