```
import pandas as pd
```

```
df = pd.read_csv('social_ads.csv')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   segment        200 non-null    int64  
 1   google         200 non-null    float64
 2   facebook       200 non-null    float64
 3   instagram      200 non-null    float64
 4   sales          200 non-null    float64
 5   size           200 non-null    object 
 6   is_large       200 non-null    int64  
 7   area           200 non-null    object 
 8   area_suburban  200 non-null    int64  
 9   area_urban     200 non-null    int64  
dtypes: float64(4), int64(4), object(2)
memory usage: 15.8+ KB
```

```
df.head(10)
```

	segment	google	facebook	instagram	sales	size	is_large	area	area_suburban	area_urban
0	1	230.1	37.8	69.2	22.1	large	1	rural	0	0
1	2	44.5	39.3	45.1	10.4	small	0	urban	0	1
2	3	17.2	45.9	69.3	9.3	small	0	rural	0	0
3	4	151.5	41.3	58.5	18.5	small	0	urban	0	1
4	5	180.8	10.8	58.4	12.9	large	1	suburban	1	0
5	6	8.7	48.9	75.0	7.2	large	1	suburban	1	0
6	7	57.5	32.8	23.5	11.8	large	1	suburban	1	0
7	8	120.2	19.6	11.6	13.2	large	1	urban	0	1
8	9	8.6	2.1	1.0	4.8	large	1	rural	0	0
9	10	199.8	2.6	21.2	10.6	large	1	suburban	1	0

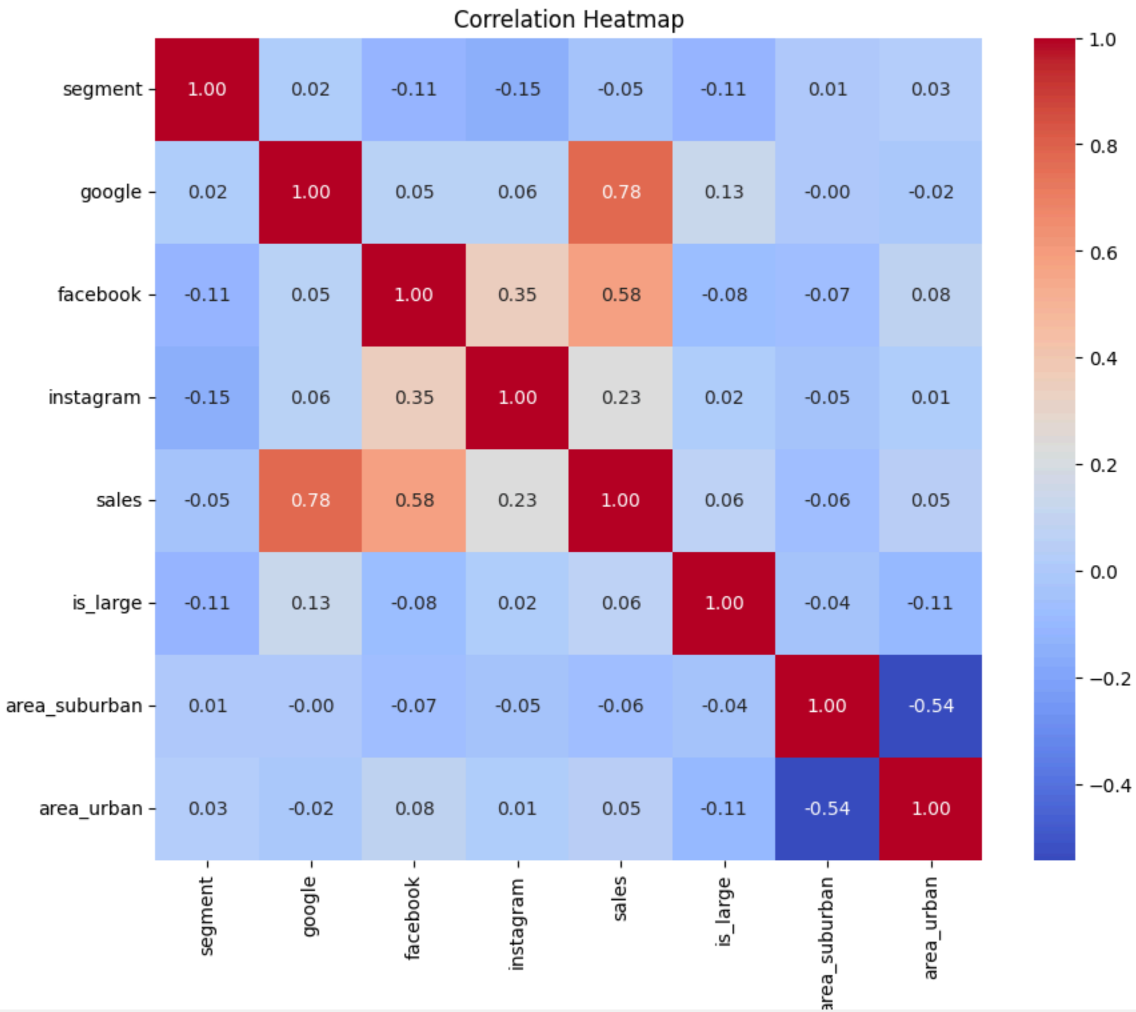
```
df.describe()
```

	segment	google	facebook	instagram	sales	is_large	area_suburban	area_urban
count	200.000000	200.000000	200.000000	200.000000	200.000000	200.000000	200.000000	200.000000
mean	100.500000	147.042500	23.264000	30.554000	14.022500	0.540000	0.390000	0.315000
std	57.879185	85.854236	14.846809	21.778621	5.217457	0.499648	0.488974	0.465682
min	1.000000	0.700000	0.000000	0.300000	1.600000	0.000000	0.000000	0.000000
25%	50.750000	74.375000	9.975000	12.750000	10.375000	0.000000	0.000000	0.000000
50%	100.500000	149.750000	22.900000	25.750000	12.900000	1.000000	0.000000	0.000000
75%	150.250000	218.825000	36.525000	45.100000	17.400000	1.000000	1.000000	1.000000
max	200.000000	296.400000	49.600000	114.000000	27.000000	1.000000	1.000000	1.000000

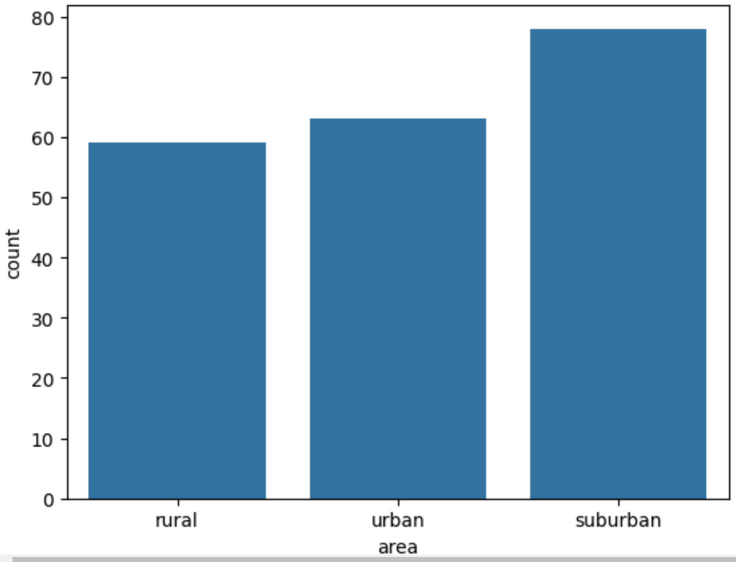
```
corr = df.corr(numeric_only=True)
corr
```

```
                segment    google   facebook  instagram     sales   is_large  area_suburban  area_urban
segment        1.000000  0.017715  -0.110680  -0.154944  -0.051616  -0.110514       0.006747    0.027686
google         0.017715  1.000000   0.054809   0.056648   0.782224   0.131097      -0.004299   -0.023388
facebook      -0.110680  0.054809   1.000000   0.354104   0.576223  -0.078994      -0.068522    0.078618
instagram     -0.154944  0.056648   0.354104   1.000000   0.228299   0.016979      -0.046722    0.012733
sales         -0.051616  0.782224   0.576223   0.228299   1.000000   0.061626      -0.062942    0.052911
is_large      -0.110514  0.131097  -0.078994   0.016979   0.061626   1.000000      -0.043605   -0.108417
area_suburban  0.006747 -0.004299  -0.068522  -0.046722  -0.062942  -0.043605       1.000000   -0.542222
area_urban     0.027686 -0.023388   0.078618   0.012733   0.052911  -0.108417      -0.542222    1.000000
```

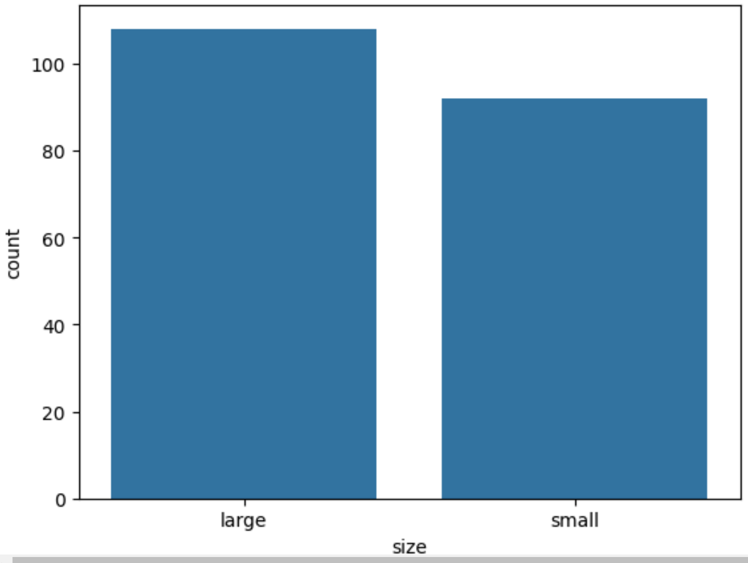
```
# prompt: Using dataframe corr: draw a heatmap

import matplotlib.pyplot as plt
import seaborn as sns

# Create a heatmap
plt.figure(figsize=(10, 8))  # Adjust figure size as needed
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
```



```
ax = sns.countplot(data=df, x="area")
```

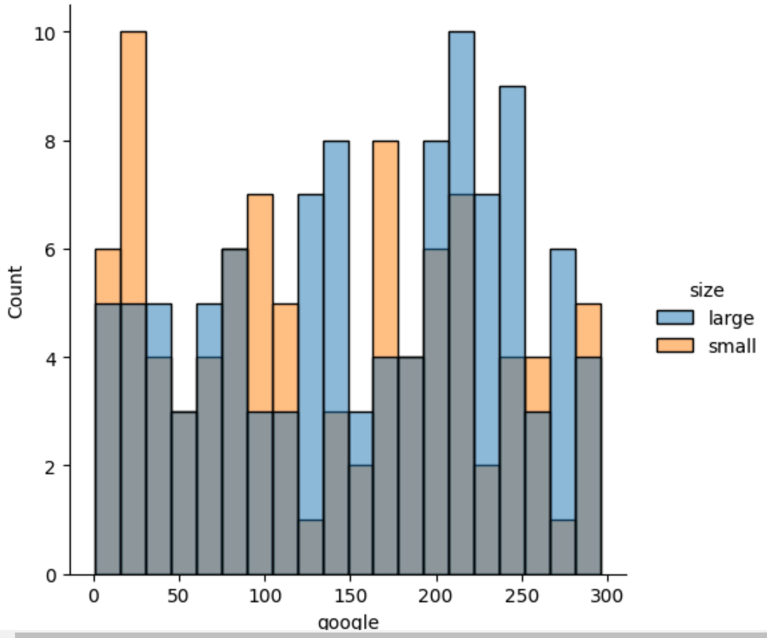


```
ax = sns.countplot(data=df, x=″size″)
```

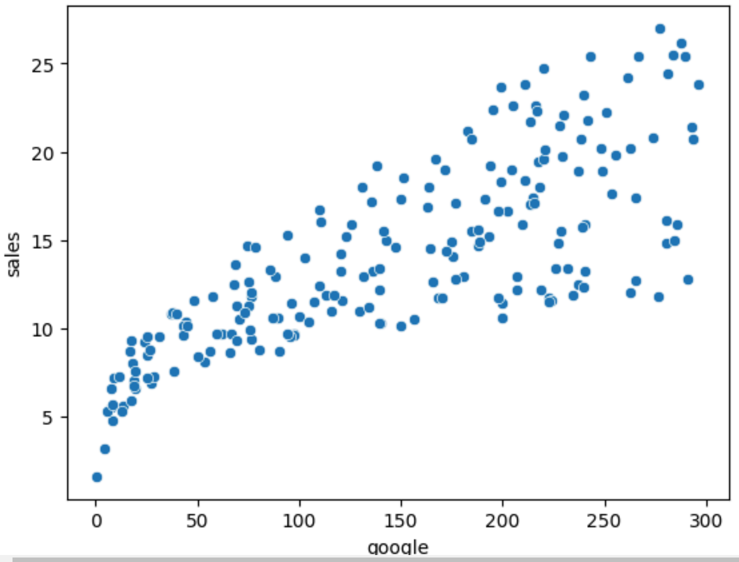


```
sns.displot(data=df, x=″google″, bins=20,hue=″size″)
```

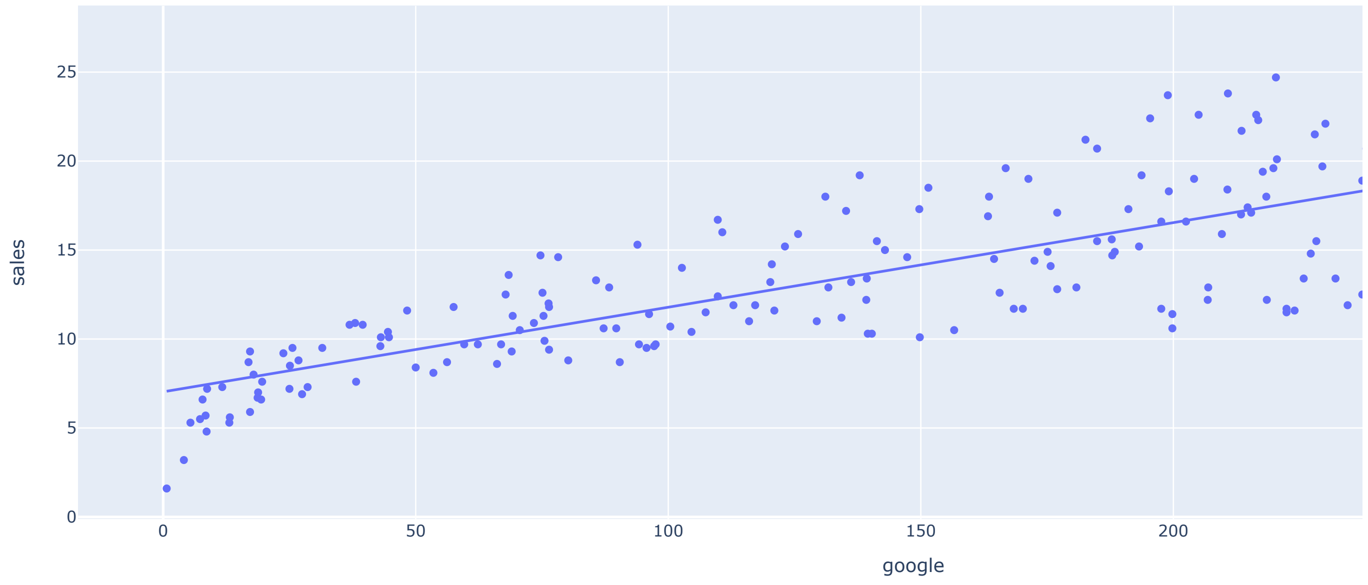
```
<seaborn.axisgrid.FacetGrid at 0x7bc22360b970>
```



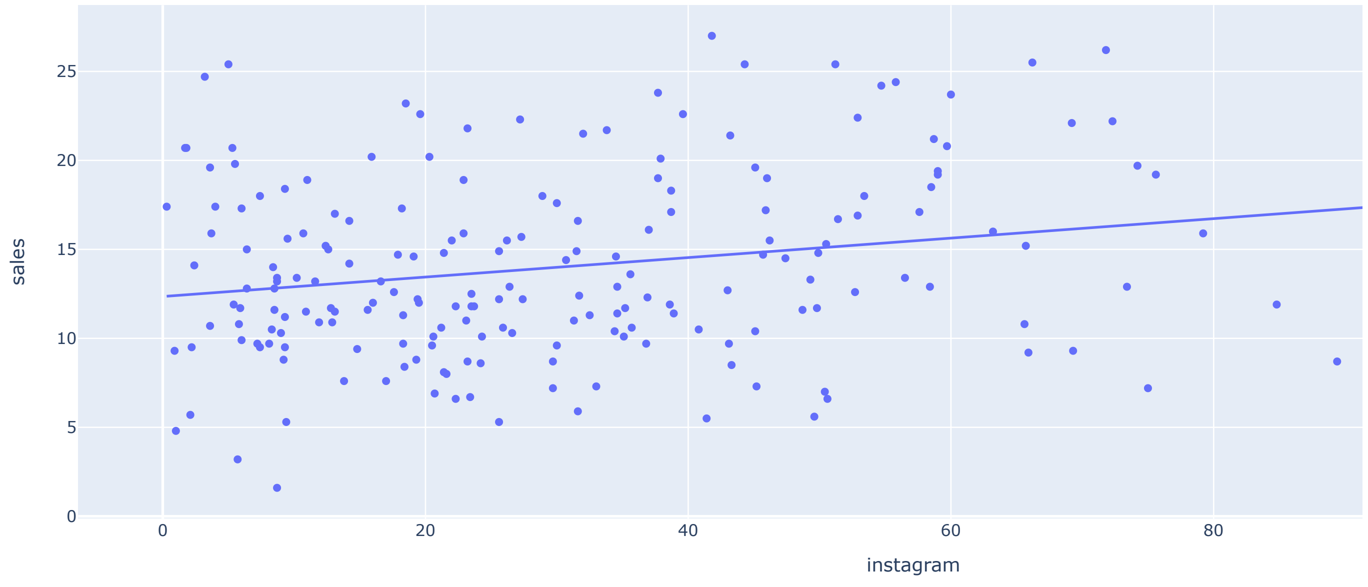
```
ax = sns.scatterplot(data=df, x="google", y="sales")
```



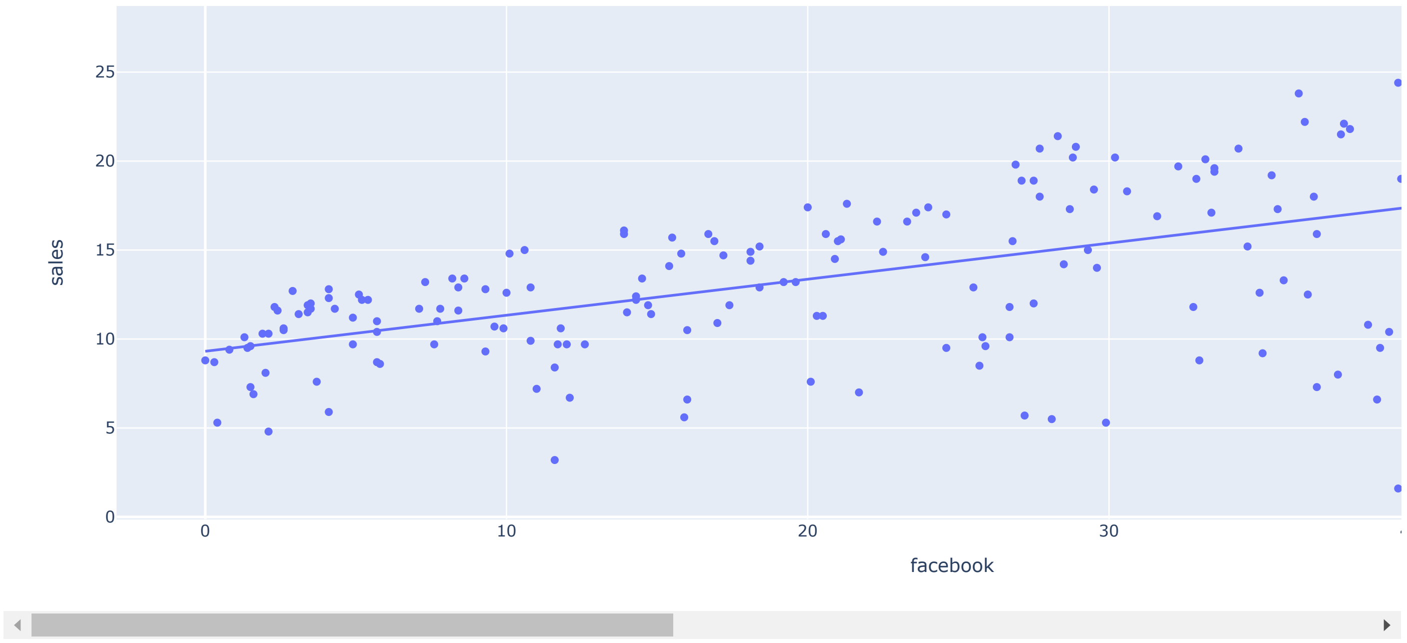
```
# prompt: create a interactive scatter plot

import plotly.express as px

# Assuming 'df' is your DataFrame
fig = px.scatter(df, x="google", y="sales", trendline="ols")
fig.show()
```



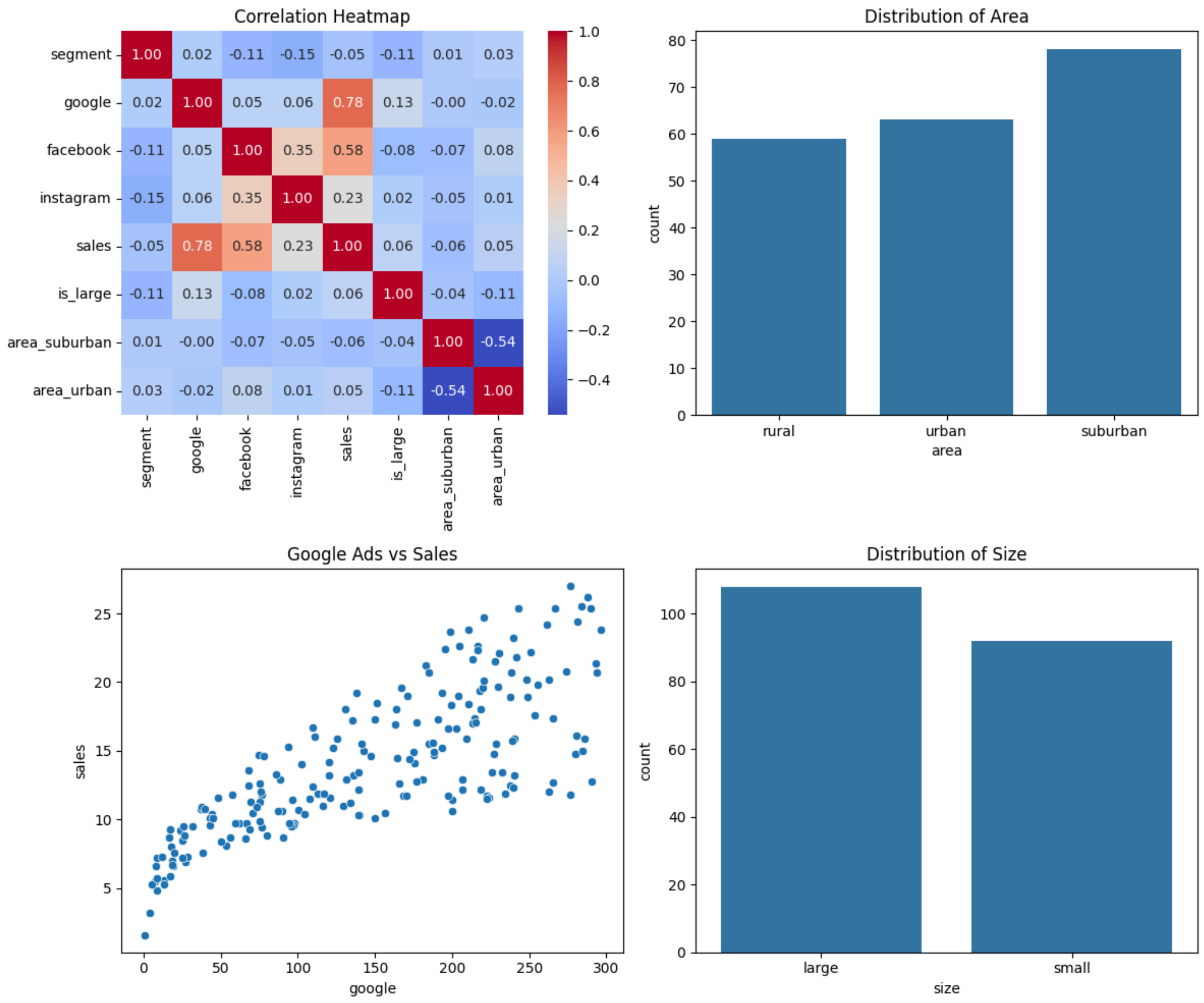
```
# prompt: create a interactive scatter plot

import plotly.express as px

# Assuming ’df’ is your DataFrame
fig = px.scatter(df, x=″instagram″, y=″sales″, trendline=″ols″)
fig.show()
```



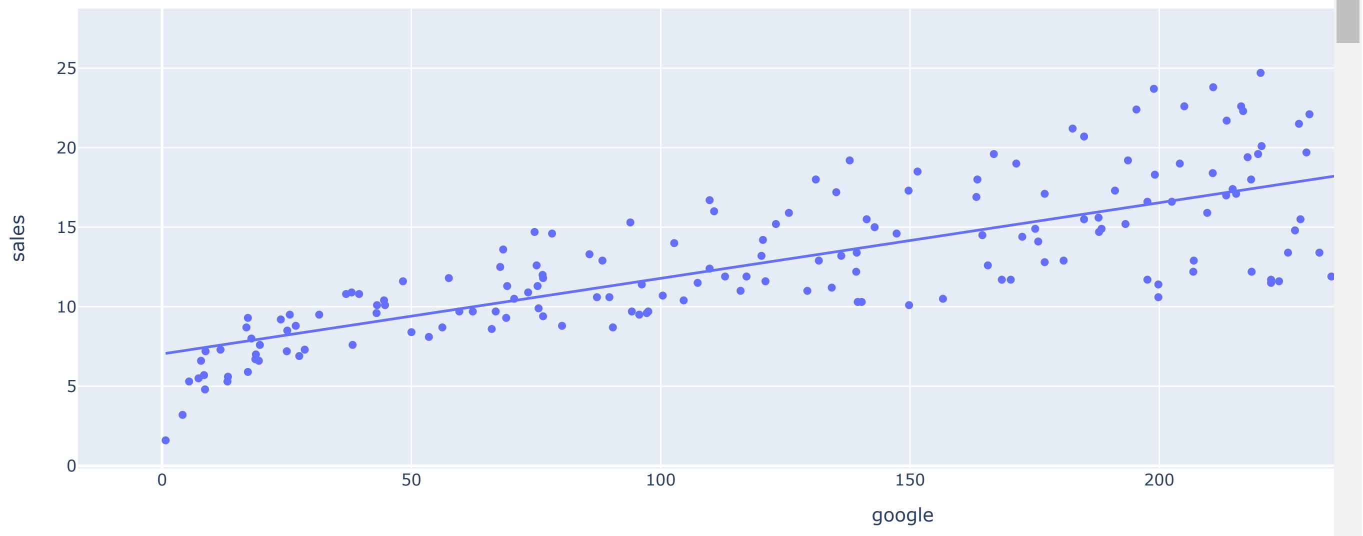
```
# prompt: create a interactive scatter plot

import plotly.express as px

# Assuming ’df’ is your DataFrame
fig = px.scatter(df, x=″facebook″, y=″sales″, trendline=″ols″)
fig.show()
```



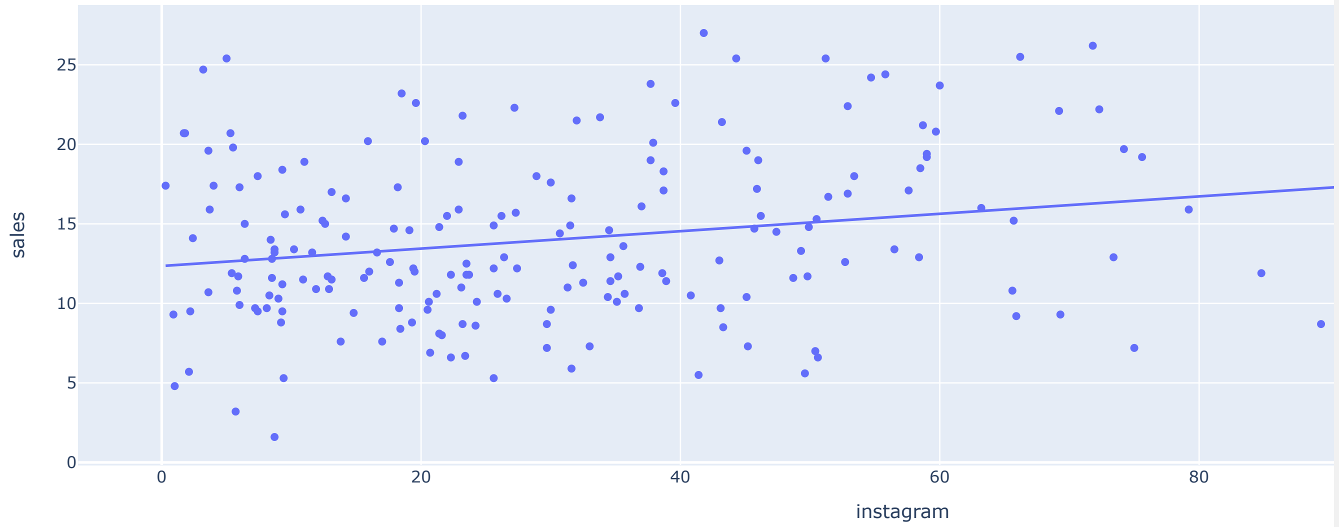
```
# prompt: put the graph together

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

# Assuming ’df’ is already loaded

# Create subplots
fig, axes = plt.subplots(2, 2, figsize=(12, 10))

# Heatmap
sns.heatmap(corr, annot=True, cmap=’coolwarm’, fmt=″.2f″, ax=axes[0, 0])
axes[0, 0].set_title(’Correlation Heatmap’)

# Countplot for ’area’
sns.countplot(data=df, x=″area″, ax=axes[0, 1])
axes[0, 1].set_title(’Distribution of Area’)

# Scatterplot for ’google’ vs ’sales’ with trendline
sns.scatterplot(data=df, x=″google″, y=″sales″, ax=axes[1, 0])
axes[1, 0].set_title(’Google Ads vs Sales’)

# Countplot for ’size’
sns.countplot(data=df, x=″size″, ax=axes[1, 1])
axes[1, 1].set_title(’Distribution of Size’)

# Adjust layout
plt.tight_layout()
plt.show()

# Interactive plots using Plotly Express (shown separately)
fig1 = px.scatter(df, x=″google″, y=″sales″, trendline=″ols″, title=’Google Ads vs Sales (Interactive)’)
fig1.show()

fig2 = px.scatter(df, x=″instagram″, y=″sales″, trendline=″ols″, title=’Instagram Ads vs Sales (Interactive)’)
fig2.show()

fig3 = px.scatter(df, x=″facebook″, y=″sales″, trendline=″ols″, title=’Facebook Ads vs Sales (Interactive)’)
fig3.show()
```



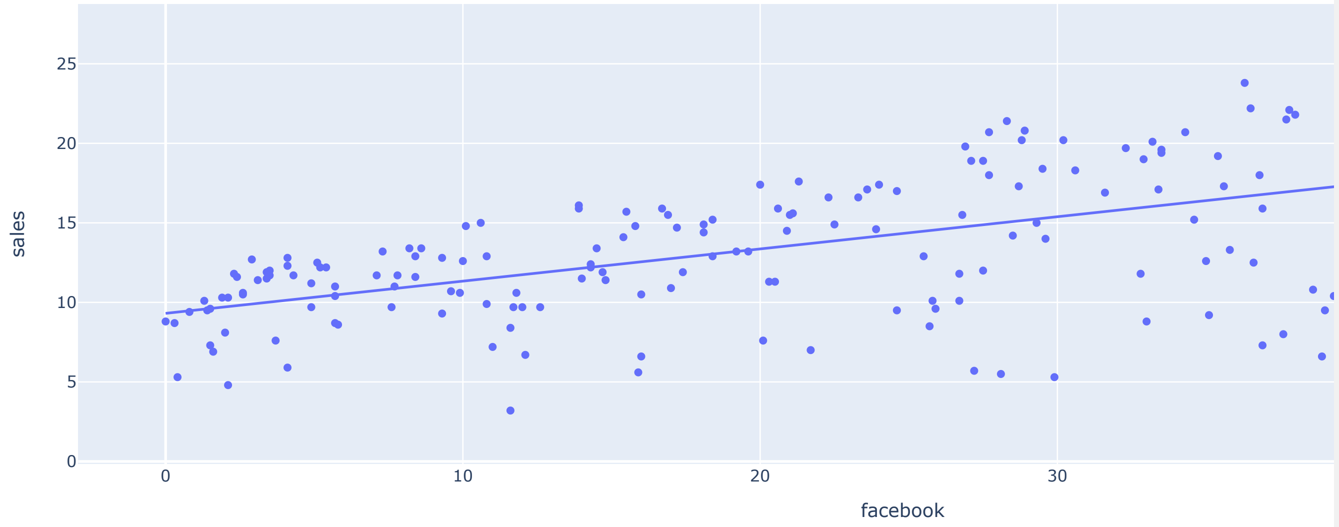
Google Ads vs Sales (Interactive)



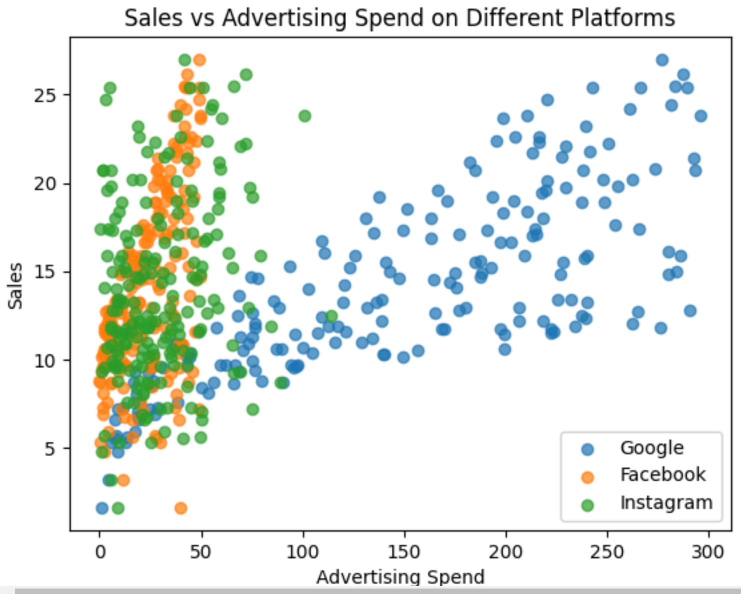
Instagram Ads vs Sales (Interactive)



Facebook Ads vs Sales (Interactive)



```
# prompt: scatter plot 'google','facebook','instaram' in one figure

# Create a figure and axes
fig, ax = plt.subplots()

# Scatter plots for each platform
ax.scatter(df['google'], df['sales'], label='Google', alpha=0.7)
ax.scatter(df['facebook'], df['sales'], label='Facebook', alpha=0.7)
ax.scatter(df['instagram'], df['sales'], label='Instagram', alpha=0.7)

# Set labels and title
ax.set_xlabel('Advertising Spend')
ax.set_ylabel('Sales')
ax.set_title('Sales vs Advertising Spend on Different Platforms')

# Add a legend
ax.legend()

# Display the plot
plt.show()
```



```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Assuming 'df' is your DataFrame
X = df[['google', 'facebook','instagram','is_large']]  # Select features
y = df['sales']  # Target variable


# Create and fit the model
model = LinearRegression()
model.fit(X,y)
model.score(X,y)
```

```
0.8972400790092475
```