Project Name - Optimizing Social Media Ad Campaign Performance for Audience Engagement and ROI

Data Wrangling

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

```
# Load the files
kag = pd.read_csv("/content/drive/MyDrive/Kag /KAG.csv")
```

```
kag
```

	ad_id	xyz_campaign_id	fb_campaign_id	age	gender	interest	Impressions	Clicks	Spent	Total_Conversion	Approved_Conver
0	708746	916	103916	30-34	M	15	7350	1	1.430000	2	
1	708749	916	103917	30-34	M	16	17861	2	1.820000	2	
2	708771	916	103920	30-34	M	20	693	0	0.000000	1	
3	708815	916	103928	30-34	M	28	4259	1	1.250000	1	
4	708818	916	103928	30-34	M	28	4133	1	1.290000	1	
...	...	...	...	...	...	...	...	...	...	...	...
1138	1314410	1178	179977	45-49	F	109	1129773	252	358.189997	13	
1139	1314411	1178	179978	45-49	F	110	637549	120	173.880003	3	

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

```
kag.isnull().sum()
```

	0
ad_id	0
xyz_campaign_id	0
fb_campaign_id	0
age	0
gender	0
interest	0
Impressions	0
Clicks	0
Spent	0
Total_Conversion	0
Approved_Conversion	0

dtype: int64

```
# Change the datatypes
kag.dtypes
```

	0
ad_id	int64
xyz_campaign_id	int64
fb_campaign_id	int64
age	object
gender	object
interest	int64
Impressions	int64
Clicks	int64
Spent	float64
Total_Conversion	int64
Approved_Conversion	int64

dtype: object

```
# Checking For Duplicates
kag.duplicated().sum()
```

```
np.int64(0)
```

```
# Feature Engineering

kag["Age_Gender"] = kag["age"].astype(str) + "_" + kag["gender"].astype(str)
```

```
kag.isnull().sum()
```

	0
ad_id	0
xyz_campaign_id	0
fb_campaign_id	0
age	0
gender	0
interest	0
Impressions	0
Clicks	0
Spent	0
Total_Conversion	0
Approved_Conversion	0
Age_Gender	0

dtype: int64

```
kag.nunique()
```

	0
ad_id	1143
xyz_campaign_id	3
fb_campaign_id	691
age	4
gender	2
interest	40
Impressions	1130
Clicks	183
Spent	869
Total_Conversion	32
Approved_Conversion	16
Age_Gender	8

dtype: int64

```
kag.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1143 entries, 0 to 1142
Data columns (total 12 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   ad_id                1143 non-null   int64  
 1   xyz_campaign_id      1143 non-null   int64  
 2   fb_campaign_id       1143 non-null   int64  
 3   age                  1143 non-null   object 
 4   gender               1143 non-null   object 
 5   interest             1143 non-null   int64  
 6   Impressions          1143 non-null   int64  
 7   Clicks               1143 non-null   int64  
 8   Spent                1143 non-null   float64
 9   Total_Conversion     1143 non-null   int64  
 10  Approved_Conversion  1143 non-null   int64  
 11  Age_Gender           1143 non-null   object 
dtypes: float64(1), int64(8), object(3)
memory usage: 107.3+ KB
```

```
kag.describe()
```

	ad_id	xyz_campaign_id	fb_campaign_id	interest	Impressions	Clicks	Spent	Total_Conversion	Approved_Conve
count	1.143000e+03	1143.000000	1143.000000	1143.000000	1.143000e+03	1143.000000	1143.000000	1143.000000	1143.0
mean	9.872611e+05	1067.382327	133783.989501	32.766404	1.867321e+05	33.390201	51.360656	2.855643	0.9
std	1.939928e+05	121.629393	20500.308622	26.952131	3.127622e+05	56.892438	86.908418	4.483593	1.7
min	7.087460e+05	916.000000	103916.000000	2.000000	8.700000e+01	0.000000	0.000000	0.000000	0.0
25%	7.776325e+05	936.000000	115716.000000	16.000000	6.503500e+03	1.000000	1.480000	1.000000	0.0
50%	1.121185e+06	1178.000000	144549.000000	25.000000	5.150900e+04	8.000000	12.370000	1.000000	1.0
75%	1.121804e+06	1178.000000	144657.500000	31.000000	2.217690e+05	37.500000	60.025000	3.000000	1.0
max	1.314415e+06	1178.000000	179982.000000	114.000000	3.052003e+06	421.000000	639.949998	60.000000	21.0

```
kag.shape
```

```
(1143, 12)
```

```
kag
```

	ad_id	xyz_campaign_id	fb_campaign_id	age	gender	interest	Impressions	Clicks	Spent	Total_Conversion	Approved_Conver
0	708746	916	103916	30-34	M	15	7350	1	1.430000	2	
1	708749	916	103917	30-34	M	16	17861	2	1.820000	2	
2	708771	916	103920	30-34	M	20	693	0	0.000000	1	
3	708815	916	103928	30-34	M	28	4259	1	1.250000	1	
4	708818	916	103928	30-34	M	28	4133	1	1.290000	1	
...	...	...	...	...	...	...	...	...	...	...	
1138	1314410	1178	179977	45-49	F	109	1129773	252	358.189997	13	
1139	1314411	1178	179978	45-49	F	110	637549	120	173.880003	3	

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

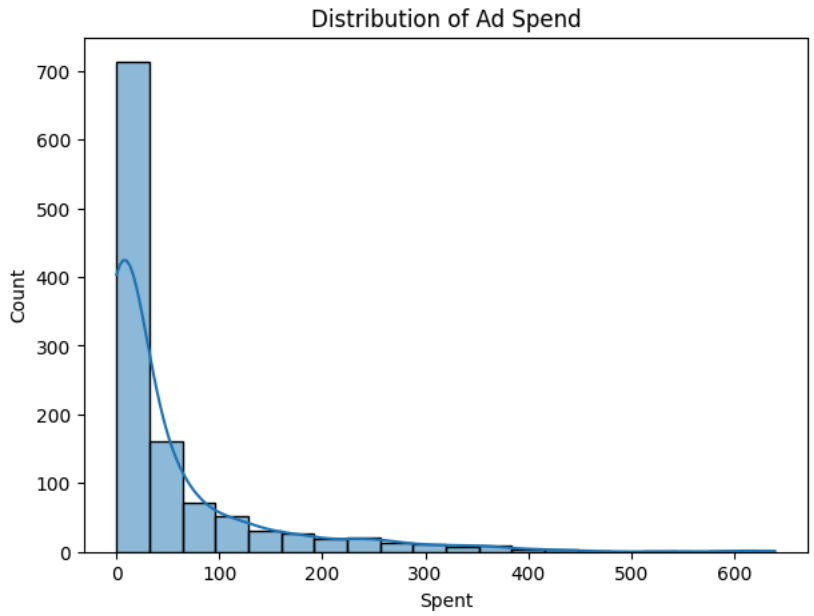
```
a1 = kag.groupby("interest")[["Clicks","Impressions"]].sum().reset_index()
a1["CTR"] = a1["Clicks"] / a1["Impressions"]
a1
```

	interest	Clicks	Impressions	CTR
0	2	311	1727646	0.000180
1	7	410	2612839	0.000157
2	10	3317	17989844	0.000184
3	15	1609	10745856	0.000150
4	16	5144	31809524	0.000162
5	18	1524	8646488	0.000176
6	19	1188	6083217	0.000195
7	20	1234	6899907	0.000179
8	21	512	2833321	0.000181
9	22	717	3965401	0.000181
10	23	375	1836368	0.000204
11	24	419	2256874	0.000186
12	25	1066	5251719	0.000203
13	26	1113	4868639	0.000229
14	27	3409	16352527	0.000208
15	28	2025	10959830	0.000185
16	29	3315	18768653	0.000177
17	30	389	2190783	0.000178
18	31	195	1075312	0.000181
19	32	1138	6455261	0.000176
20	36	128	922928	0.000139
21	63	1675	8365640	0.000200
22	64	989	5085460	0.000194
23	65	372	1737547	0.000214
24	66	138	893407	0.000154
25	100	395	2023690	0.000195
26	101	524	2960453	0.000177
27	102	150	1160953	0.000129
28	103	333	1921053	0.000173
29	104	265	1412110	0.000188
30	105	453	2656351	0.000171
31	106	332	1592431	0.000208
32	107	639	4482111	0.000143
33	108	402	2763404	0.000145
34	109	572	2980365	0.000192
35	110	365	2434719	0.000150
36	111	260	1490896	0.000174
37	112	339	2324572	0.000146
38	113	233	1830565	0.000127
39	114	191	1066164	0.000179

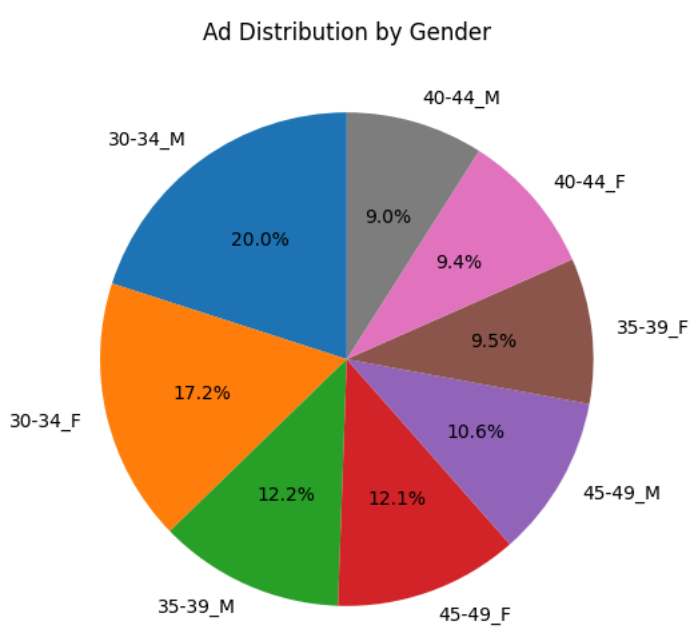
Next steps: Generate code with a1 View recommended plots New interactive sheet

Data Visualization

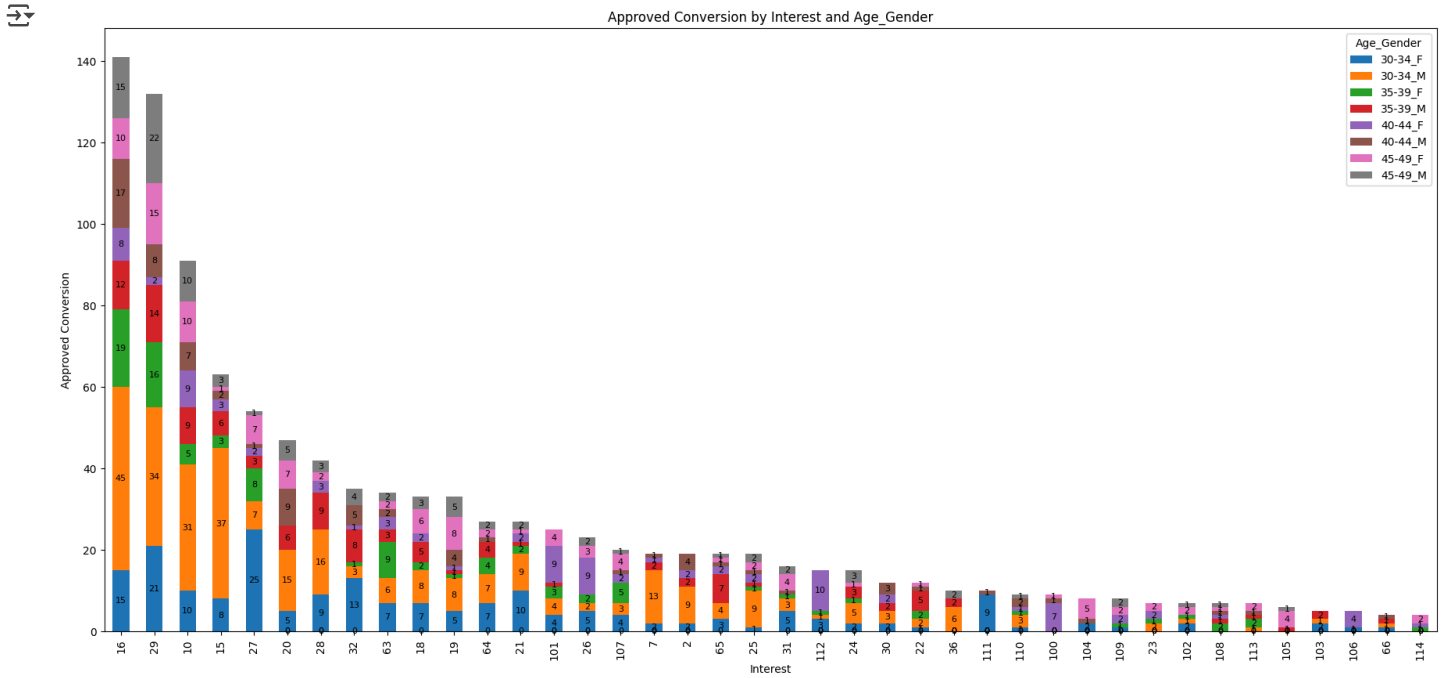
```
plt.figure(figsize=(7,5))
sns.histplot(kag["Spent"], bins=20, kde=True)
plt.title("Distribution of Ad Spend")
plt.show()
```



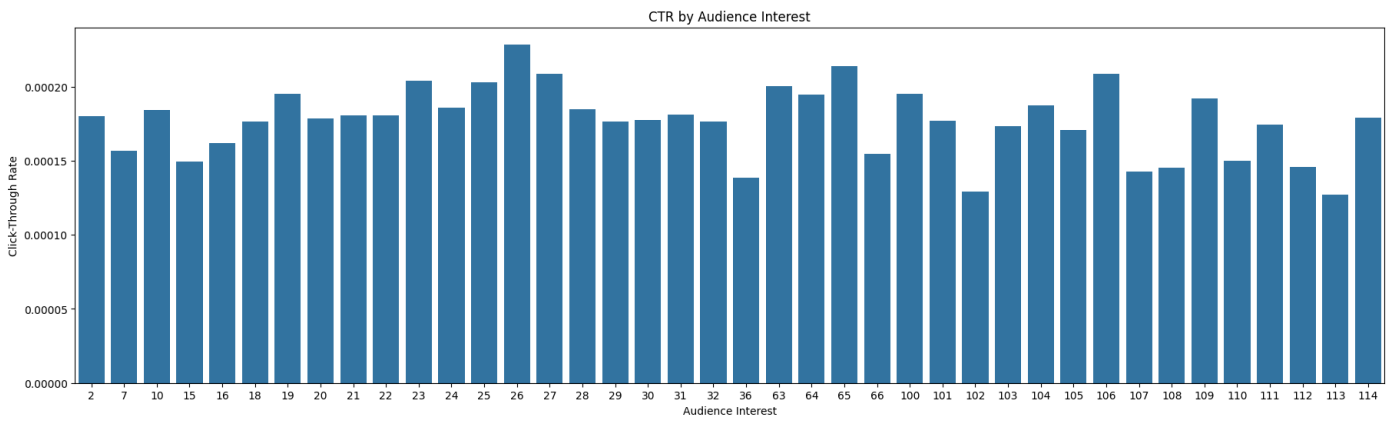

```
data1 = kag["Age_Gender"].value_counts()
plt.figure(figsize=(6,6))
plt.pie(data1, labels = data1.index, autopct="%1.1f%%", startangle=90)
plt.title("Ad Distribution by Gender")
plt.show()
```



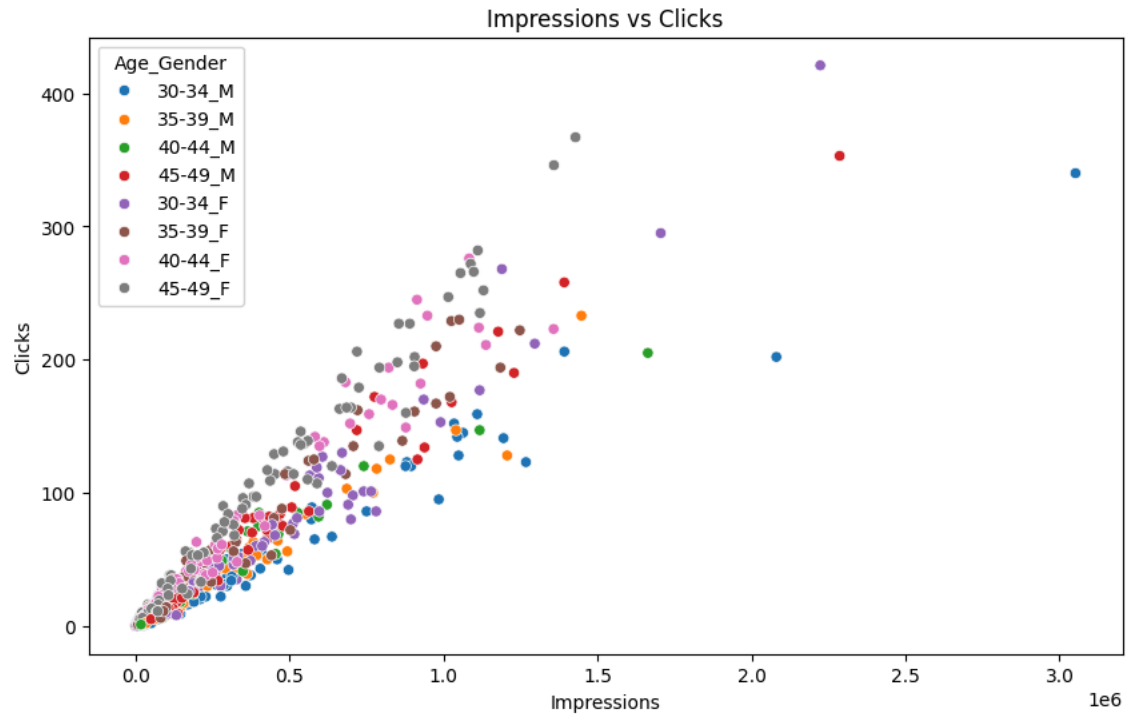

```
p5 = kag.groupby(["interest", "Age_Gender"])["Approved_Conversion"].sum().unstack(fill_value=0)
p5 = p5.loc[p5.sum(axis=1).sort_values(ascending=False).index]

ax = p5.plot(kind="bar", stacked=True, figsize=(22, 10))
for container in ax.containers:
    ax.bar_label(container, label_type="center", fontsize=8)

plt.title("Approved Conversion by Interest and Age_Gender")
plt.xlabel("Interest")
plt.ylabel("Approved Conversion")
plt.show()
```



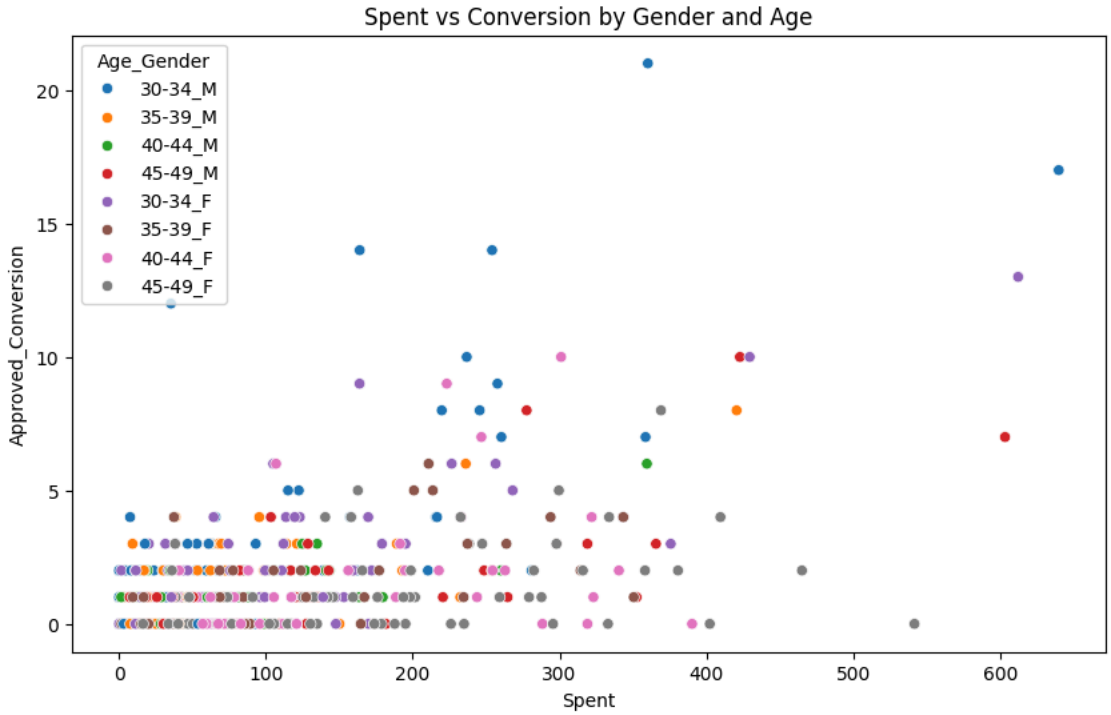

```
plt.figure(figsize=(22,6))
sns.barplot( data = a1, x="interest", y="CTR")

plt.title("CTR by Audience Interest")
plt.xlabel("Audience Interest")
plt.ylabel("Click-Through Rate")
plt.show()
```



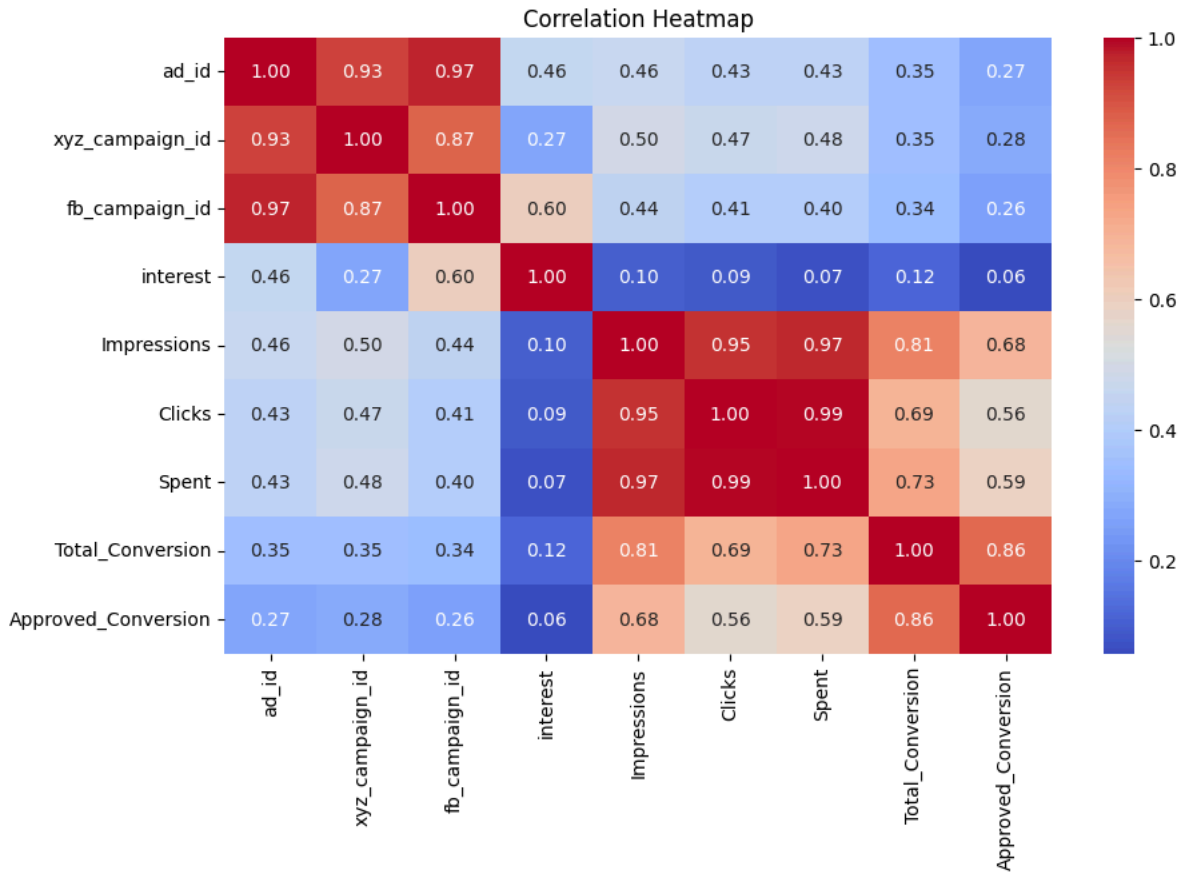

```
plt.figure(figsize=(10,6))
sns.scatterplot(data=kag, x="Impressions", y="Clicks", hue="Age_Gender")
plt.title("Impressions vs Clicks")
plt.show()
```




```
plt.figure(figsize=(10,6))
sns.scatterplot(data=kag, x="Spent", y="Approved_Conversion", hue="Age_Gender")
plt.title("Spent vs Conversion by Gender and Age")
plt.show()
```




```
plt.figure(figsize=(10,6))
corr = kag.corr(numeric_only=True)
sns.heatmap(corr, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```



Tasks

1) Which social media platforms drive the highest conversion rates and lowest CPA?

```
d1 = kag.groupby("xyz_campaign_id")[["Impressions","Clicks" ,"Spent","Approved_Conversion"]].sum().reset_index()
d1
```

	xyz_campaign_id	Impressions	Clicks	Spent	Approved_Conversion
0	916	482925	113	149.710001	24
1	936	8128187	1984	2893.369999	183
2	1178	204823716	36068	55662.149959	872

Next steps: Generate code with d1 | View recommended plots | New interactive sheet

```
d1["CTR"] = (d1["Clicks"] / d1["Impressions"])*100
d1["Conversion_Rate"] = np.where(d1["Clicks"] > 0, (d1["Approved_Conversion"] / d1["Clicks"])*100, 0)
d1["CPA"] = np.where(d1["Approved_Conversion"] > 0, d1["Spent"] / d1["Approved_Conversion"], 0)
d1 = d1.set_index("xyz_campaign_id")
d1
```

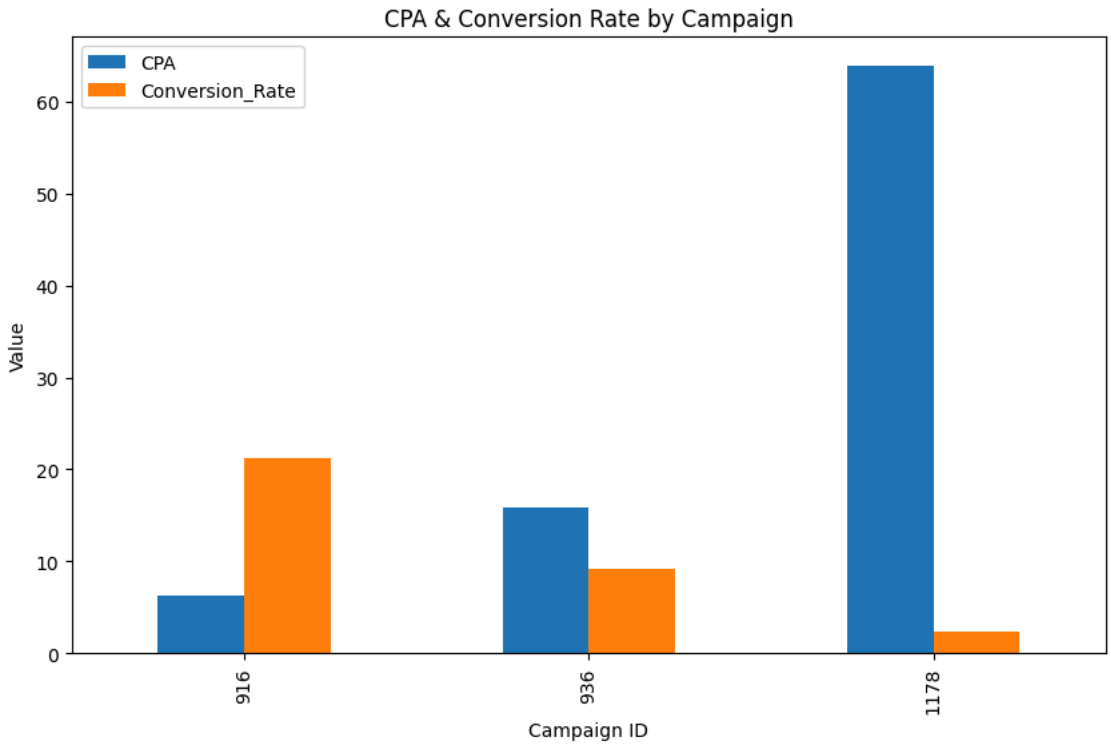
xyz_campaign_id	Impressions	Clicks	Spent	Approved_Conversion	CTR	Conversion_Rate	CPA
916	482925	113	149.710001	24	0.023399	21.238938	6.237917
936	8128187	1984	2893.369999	183	0.024409	9.223790	15.810765
1178	204823716	36068	55662.149959	872	0.017609	2.417656	63.832741

Next steps: Generate code with d1 | View recommended plots | New interactive sheet

```
print(d1["Conversion_Rate"].idxmax())
print(d1["CPA"].idxmin())
```

```
916
916
```

```
d1[["CPA", "Conversion_Rate"]].plot(kind="bar",figsize=(10, 6))
plt.title("CPA & Conversion Rate by Campaign")
plt.xlabel("Campaign ID")
plt.ylabel("Value")
plt.show()
```



2) How do audience demographics (age, gender, location) influence campaign effectiveness?

```
d2 = kag.groupby("Age_Gender")[["Impressions","Clicks" ,"Spent","Approved_Conversion"]].sum().reset_index()
d2
```

	Age_Gender	Impressions	Clicks	Spent	Approved_Conversion
0	30-34_F	31571576	5099	7611.479995	195
1	30-34_M	36421443	4384	7640.919991	299
2	35-39_F	21439505	4161	6061.349992	95
3	35-39_M	20665139	2933	5051.080003	112
4	40-44_F	23396175	5177	7396.579984	93
5	40-44_M	16208132	2559	4193.149997	77
6	45-49_F	38455591	9441	13433.209993	112
7	45-49_M	25277267	4411	7317.460004	96

Next steps: Generate code with d2 | View recommended plots | New interactive sheet

```
d2["CTR"] = (d2["Clicks"] / d2["Impressions"])*100
d2["Conversion_Rate"] = np.where(d2["Clicks"] > 0, (d2["Approved_Conversion"] / d2["Clicks"])*100, 0)
d2["CPA"] = np.where(d2["Approved_Conversion"] > 0, d2["Spent"] / d2["Approved_Conversion"], 0)
d2 = d2.set_index("Age_Gender")
d2 = d2.sort_values(by = "CPA", ascending = False)
d2
```

Age_Gender	Impressions	Clicks	Spent	Approved_Conversion	CTR	Conversion_Rate	CPA
45-49_F	38455591	9441	13433.209993	112	0.024550	1.186315	119.939375
40-44_F	23396175	5177	7396.579984	93	0.022128	1.796407	79.533118
45-49_M	25277267	4411	7317.460004	96	0.017450	2.176377	76.223542
35-39_F	21439505	4161	6061.349992	95	0.019408	2.283105	63.803684
40-44_M	16208132	2559	4193.149997	77	0.015788	3.008988	54.456493
35-39_M	20665139	2933	5051.080003	112	0.014193	3.818616	45.098929
30-34_F	31571576	5099	7611.479995	195	0.016151	3.824279	39.033231
30-34_M	36421443	4384	7640.919991	299	0.012037	6.820255	25.554916

Next steps: Generate code with d2 View recommended plots New interactive sheet

```
heatmap = d2.style.background_gradient(cmap="YlGnBu")
heatmap
```

Age_Gender	Impressions	Clicks	Spent	Approved_Conversion	CTR	Conversion_Rate	CPA
45-49_F	38455591	9441	13433.209993	112	0.024550	1.186315	119.939375
40-44_F	23396175	5177	7396.579984	93	0.022128	1.796407	79.533118
45-49_M	25277267	4411	7317.460004	96	0.017450	2.176377	76.223542
35-39_F	21439505	4161	6061.349992	95	0.019408	2.283105	63.803684
40-44_M	16208132	2559	4193.149997	77	0.015788	3.008988	54.456493
35-39_M	20665139	2933	5051.080003	112	0.014193	3.818616	45.098929
30-34_F	31571576	5099	7611.479995	195	0.016151	3.824279	39.033231
30-34_M	36421443	4384	7640.919991	299	0.012037	6.820255	25.554916

3) Which ad creatives or campaign types generate the greatest engagement (CTR) and conversions?

```
d3 = kag.groupby("ad_id")[["Impressions","Clicks" ,"Spent","Approved_Conversion"]].sum().reset_index()
d3
```

	ad_id	Impressions	Clicks	Spent	Approved_Conversion
0	708746	7350	1	1.430000	1
1	708749	17861	2	1.820000	0
2	708771	693	0	0.000000	0
3	708815	4259	1	1.250000	0
4	708818	4133	1	1.290000	1
...	...	...	...	...	...
1138	1314410	1129773	252	358.189997	2
1139	1314411	637549	120	173.880003	0
1140	1314412	151531	28	40.289999	0
1141	1314414	790253	135	198.710001	2
1142	1314415	513161	114	165.609999	2

1143 rows × 5 columns

Next steps: Generate code with d3 View recommended plots New interactive sheet

```
d3["CTR"] = (d3["Clicks"] / d3["Impressions"])*100
d3["Conversion_Rate"] = np.where(d3["Clicks"] > 0, (d3["Approved_Conversion"] / d3["Clicks"])*100, 0)
d3["CPA"] = np.where(d3["Approved_Conversion"] > 0, d3["Spent"] / d3["Approved_Conversion"], 0)
```

```
d3 = d3.set_index("ad_id").sort_values(by="CTR", ascending=False)
d3
```

ad_id	Impressions	Clicks	Spent	Approved_Conversion	CTR	Conversion_Rate	CPA
738637	944	1	1.42	0	0.105932	0.000000	0.00
950224	2367	2	2.84	1	0.084495	50.000000	2.84
951779	3277	2	2.68	0	0.061031	0.000000	0.00
951202	5307	3	4.29	1	0.056529	33.333333	4.29
950537	1884	1	1.41	0	0.053079	0.000000	0.00
...	...	...	...	...	...	...	...
734313	790	0	0.00	1	0.000000	0.000000	0.00
734314	962	0	0.00	0	0.000000	0.000000	0.00
708771	693	0	0.00	0	0.000000	0.000000	0.00
708979	1224	0	0.00	0	0.000000	0.000000	0.00
708820	1915	0	0.00	1	0.000000	0.000000	0.00

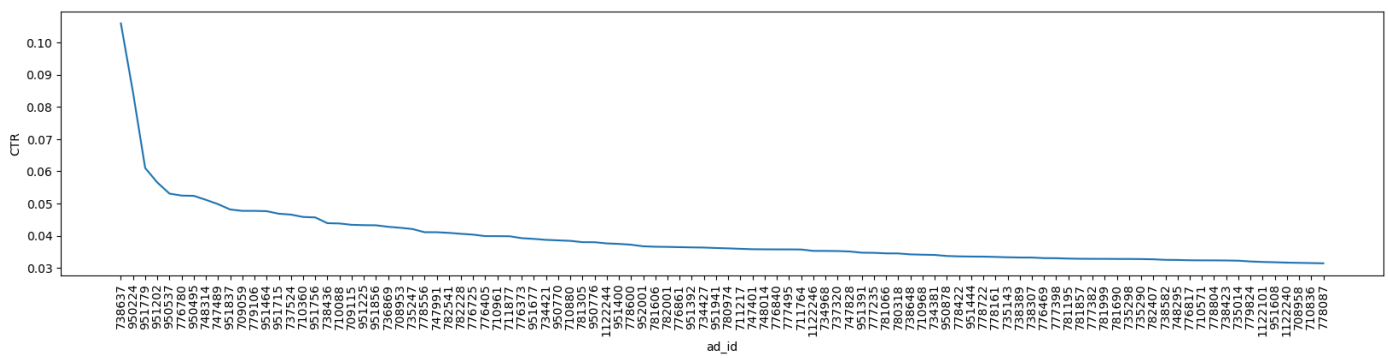
1143 rows × 7 columns

Next steps: Generate code with d3 | View recommended plots | New interactive sheet

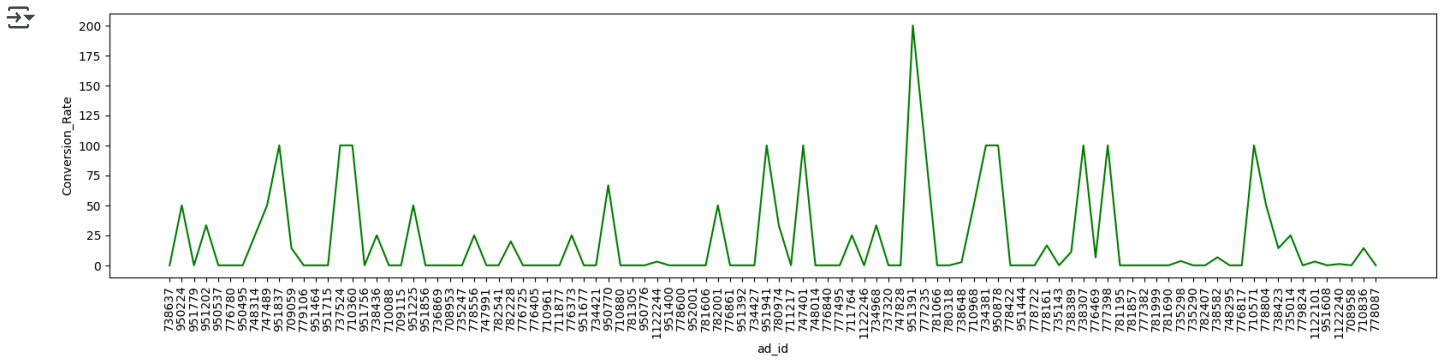
```
print(d3["CTR"].idxmax())
```

```
738637
```

```
d3 = d3.reset_index().head(100)
d3["ad_id"] = d3["ad_id"].astype(str)
plt.figure(figsize=(20,4))
sns.lineplot(data = d3, x="ad_id", y="CTR")
plt.xticks(rotation = 90)
plt.show()
```




```
plt.figure(figsize=(20,4))
sns.lineplot(data = d3, x="ad_id", y="Conversion_Rate", color ="green")
plt.xticks(rotation = 90)
plt.show()
```



4) What are the time-based performance trends - are there optimal days or times for ad spend?

```
# As Date and Time information is not given, so we assume "xyz_campaign_id", "fb_campaign_id"  as proxies for performance trends.

d4 = kag.groupby(["xyz_campaign_id", "fb_campaign_id"])[["Impressions","Clicks" ,"Spent","Approved_Conversion"]].sum().reset_index()
d4
```

	xyz_campaign_id	fb_campaign_id	Impressions	Clicks	Spent	Approved_Conversion
0	916	103916	7350	1	1.430000	1
1	916	103917	17861	2	1.820000	0
2	916	103920	693	0	0.000000	0
3	916	103928	8392	2	2.540000	1
4	916	103929	1915	0	0.000000	1
...	...	...	...	...	...	...
686	1178	179977	1129773	252	358.189997	2
687	1178	179978	637549	120	173.880003	0
688	1178	179979	151531	28	40.289999	0
689	1178	179981	790253	135	198.710001	2
690	1178	179982	513161	114	165.609999	2

691 rows × 6 columns

Next steps: Generate code with d4 | View recommended plots | New interactive sheet

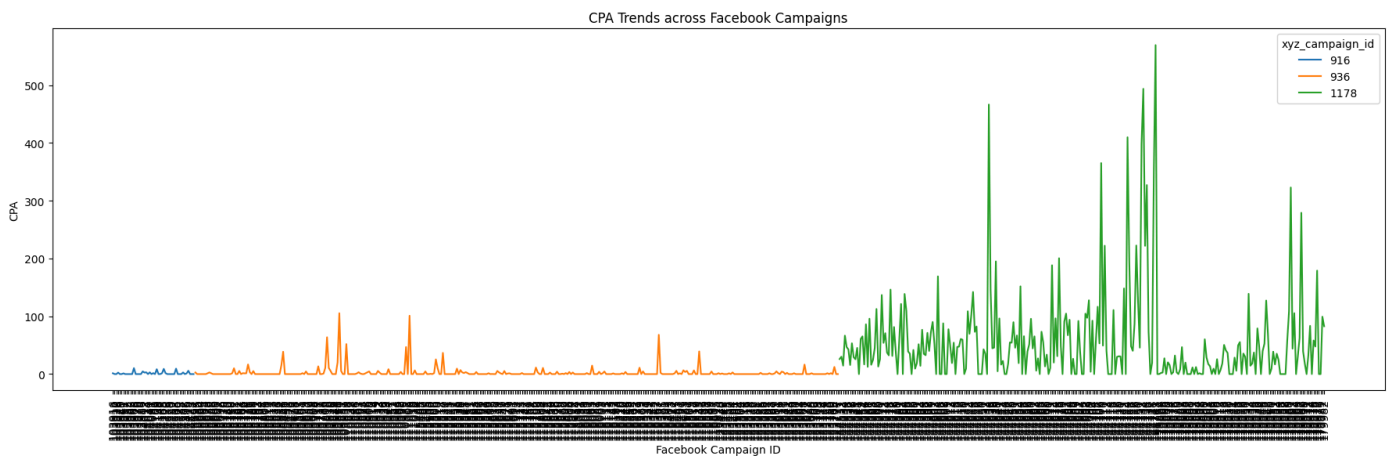
```
d4["CTR"] = (d4["Clicks"] / d4["Impressions"])*100
d4["Conversion_Rate"] = np.where(d4["Clicks"] > 0, (d4["Approved_Conversion"] / d4["Clicks"])*100, 0)
d4["CPA"] = np.where(d4["Approved_Conversion"] > 0, d4["Spent"] / d4["Approved_Conversion"], 0)
d4 = d4.set_index("xyz_campaign_id")
d4
```

xyz_campaign_id	fb_campaign_id	Impressions	Clicks	Spent	Approved_Conversion	CTR	Conversion_Rate	CPA
916	103916	7350	1	1.430000	1	0.013605	100.000000	1.430000
916	103917	17861	2	1.820000	0	0.011198	0.000000	0.000000
916	103920	693	0	0.000000	0	0.000000	0.000000	0.000000
916	103928	8392	2	2.540000	1	0.023832	50.000000	2.540000
916	103929	1915	0	0.000000	1	0.000000	0.000000	0.000000
...	...	...	...	...	...	...	...	...
1178	179977	1129773	252	358.189997	2	0.022305	0.793651	179.094999
1178	179978	637549	120	173.880003	0	0.018822	0.000000	0.000000
1178	179979	151531	28	40.289999	0	0.018478	0.000000	0.000000
1178	179981	790253	135	198.710001	2	0.017083	1.481481	99.355000
1178	179982	513161	114	165.609999	2	0.022215	1.754386	82.804999

691 rows × 8 columns

Next steps: Generate code with d4 | View recommended plots | New interactive sheet

```
d4 = d4.reset_index()
d4["fb_campaign_id"] = d4["fb_campaign_id"].astype(str)
plt.figure(figsize=(22,6))
sns.lineplot(data=d4, x="fb_campaign_id", y="CPA", hue="xyz_campaign_id", palette = "tab10")
plt.xticks(ticks=d4["fb_campaign_id"], rotation=90)
plt.title("CPA Trends across Facebook Campaigns")
plt.xlabel("Facebook Campaign ID")
plt.ylabel("CPA")
plt.show()
```



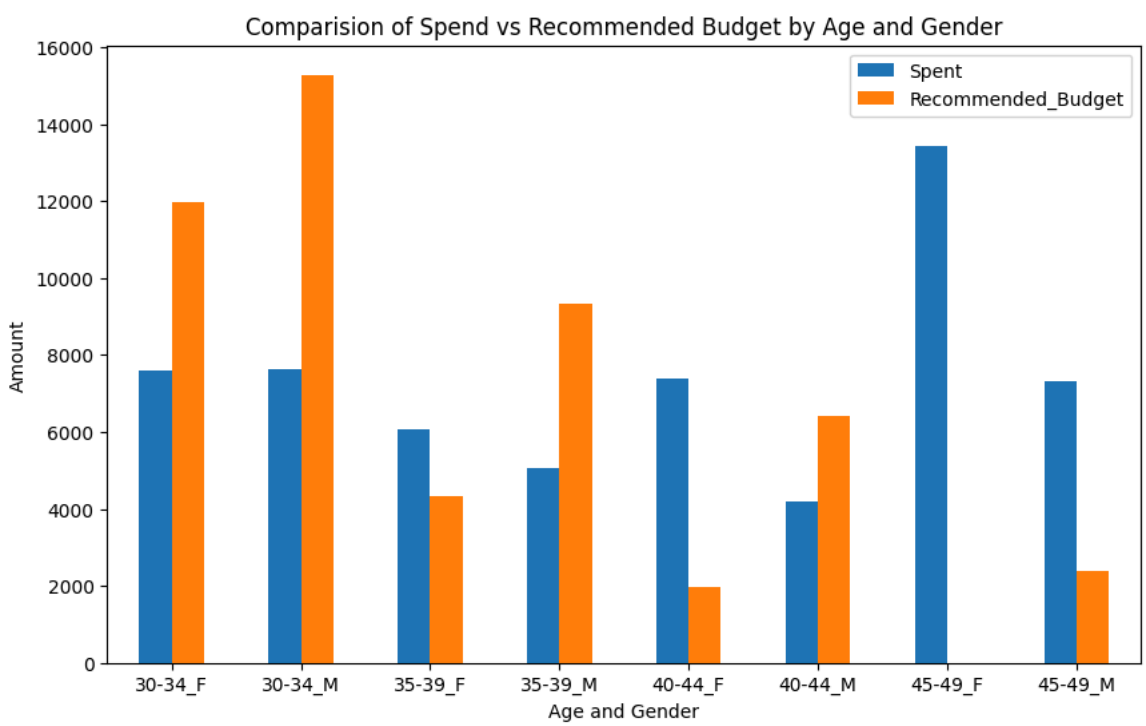
5) How should future budgets be reallocated to maximize ROI across channels and segments?

```
# We assumed each approved conversion generates a fixed revenue of 100 units. This assumption allows us to calculate ROI in a standardized w

d5a = kag.groupby(["Age_Gender"]).agg({"Spent" : "sum","Approved_Conversion" : "sum" }).reset_index()
```

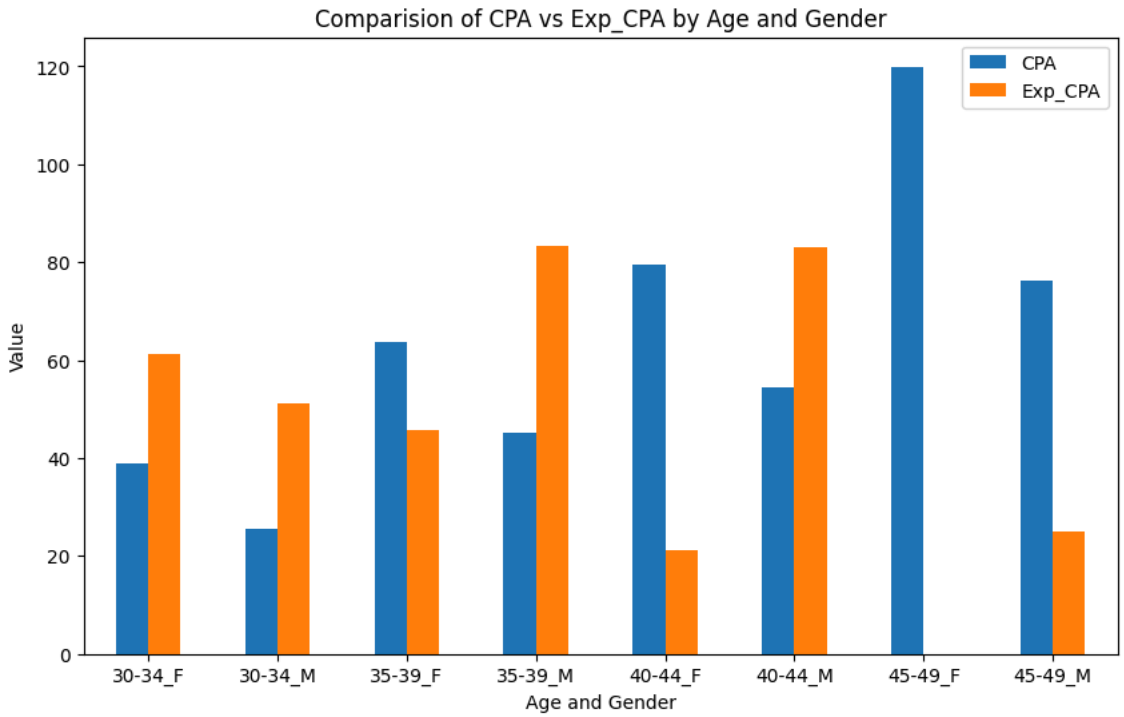
```
d5a["CPA"] = np.where(d5a["Approved_Conversion"] > 0, d5a["Spent"] / d5a["Approved_Conversion"], 0)
d5a["CPS"] = np.where(d5a["Spent"] > 0, d5a["Approved_Conversion"]/d5a["Spent"], 0)
d5a["ROI"] = np.where(d5a["Spent"] > 0,(d5a["Approved_Conversion"]*100 - d5a["Spent"]) / d5a["Spent"],0)
d5a["ROI_Positive"] = d5a["ROI"].clip(lower=0)

Total_spend = d5a["Spent"].sum()

d5a["Recommended_Budget"] = (d5a["ROI_Positive"] / d5a["ROI_Positive"].sum()) * Total_spend
d5a["Recommended_Budget"] = d5a.apply(lambda row: min(row["Recommended_Budget"], row["Spent"]*2), axis=1) # Here is maximum budget spent lim
d5a["Exp_CPA"] = np.where(d5a["Approved_Conversion"] > 0, d5a["Recommended_Budget"] / d5a["Approved_Conversion"], 0)
d5a["Exp_ROI"] = np.where(d5a["Recommended_Budget"] > 0,(d5a["Approved_Conversion"]*100 - d5a["Recommended_Budget"]) / d5a["Recommended_Budg
heatmap1 = d5a.style.background_gradient(cmap="YlGnBu")
heatmap1
```

	Age_Gender	Spent	Approved_Conversion	CPA	CPS	ROI	ROI_Positive	Recommended_Budget	Exp_CPA	Exp_ROI
0	30-34_F	7611.479995	195	39.033231	0.025619	1.561920	1.561920	11962.049214	61.343842	0.630155
1	30-34_M	7640.919991	299	25.554916	0.039131	2.913141	2.913141	15281.839983	51.109633	0.956571
2	35-39_F	6061.349992	95	63.803684	0.015673	0.567308	0.567308	4344.757239	45.734287	1.186543
3	35-39_M	5051.080003	112	45.098929	0.022173	1.217348	1.217348	9323.124795	83.242186	0.201314
4	40-44_F	7396.579984	93	79.533118	0.012573	0.257338	0.257338	1970.836407	21.191789	3.718809
5	40-44_M	4193.149997	77	54.456493	0.018363	0.836328	0.836328	6405.067255	83.182692	0.202173
6	45-49_F	13433.209993	112	119.939375	0.008338	-0.166245	0.000000	0.000000	0.000000	0.000000
7	45-49_M	7317.460004	96	76.223542	0.013119	0.311931	0.311931	2388.938344	24.884774	3.018521

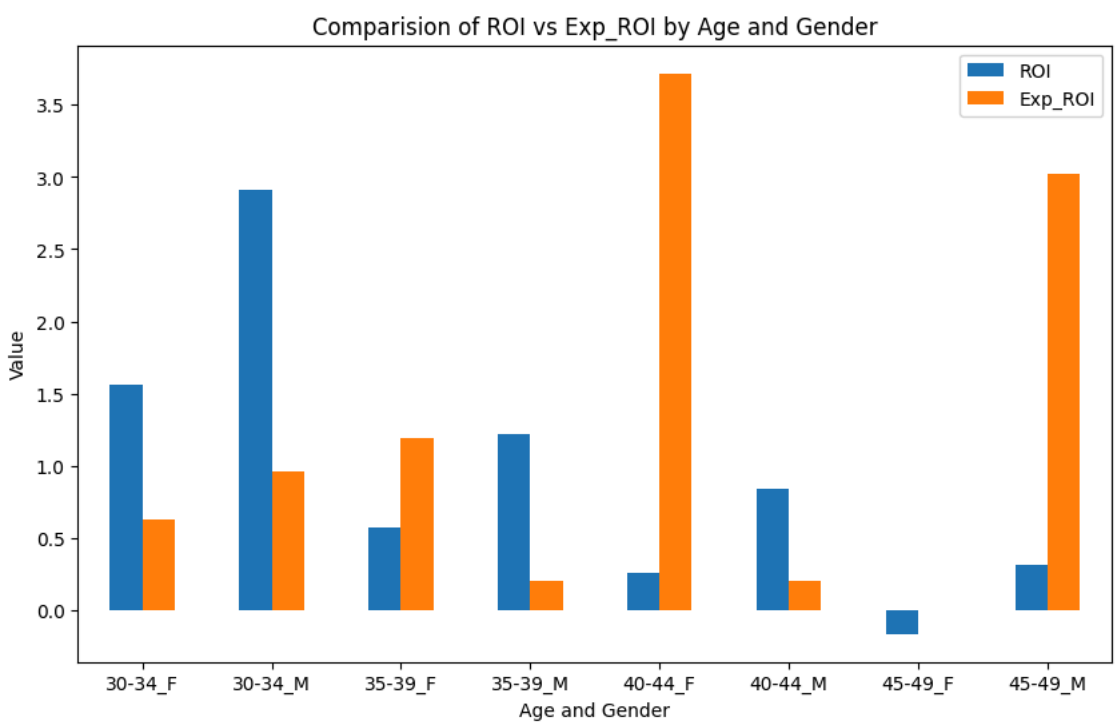
```
d5a.set_index("Age_Gender")[["Spent", "Recommended_Budget"]].plot(kind="bar", figsize=(10,6))
plt.title("Comparision of Spend vs Recommended Budget by Age and Gender")
plt.ylabel("Amount")
plt.xlabel("Age and Gender")
plt.xticks(rotation = 0)
plt.show()
```




```
d5a.set_index("Age_Gender")[["CPA", "Exp_CPA"]].plot(kind="bar", figsize=(10,6))
plt.title("Comparision of CPA vs Exp_CPA by Age and Gender")
plt.ylabel("Value")
plt.xlabel("Age and Gender")
plt.xticks(rotation = 0)
plt.show()
```




```
d5a.set_index("Age_Gender")[["ROI", "Exp_ROI"]].plot(kind="bar", figsize=(10,6))
plt.title("Comparision of ROI vs Exp_ROI by Age and Gender")
plt.ylabel("Value")
plt.xlabel("Age and Gender")
plt.xticks(rotation = 0)
plt.show()
```

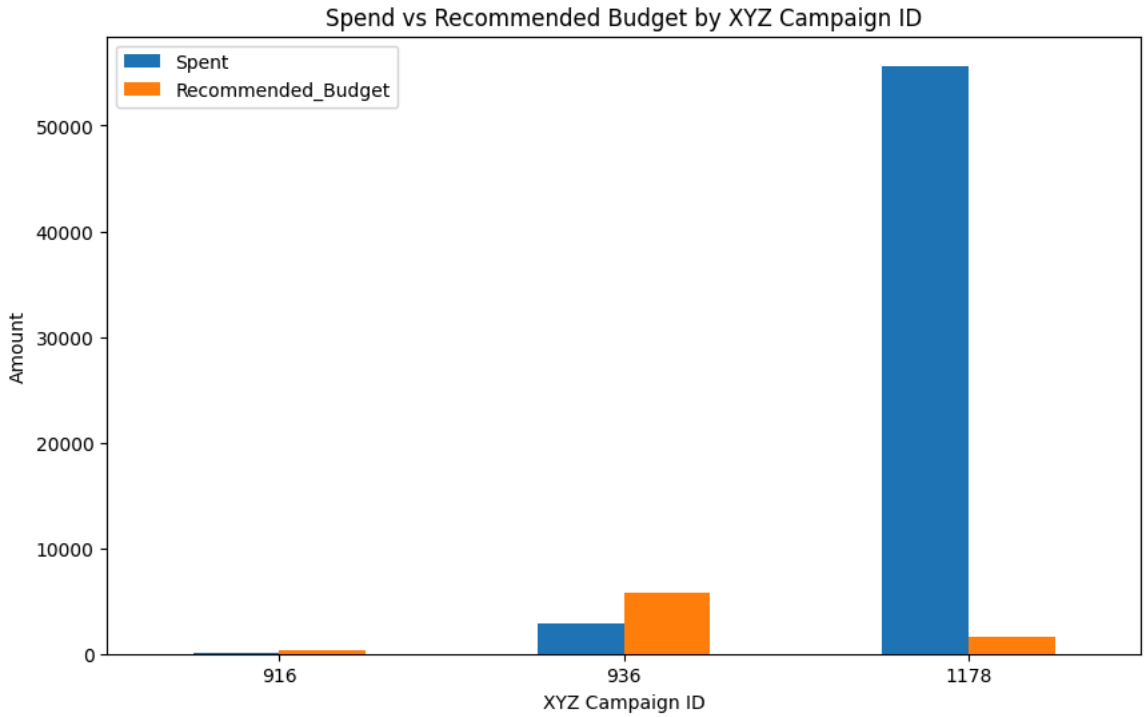



```
d5b = kag.groupby(["xyz_campaign_id"]).agg({"Spent" : "sum","Approved_Conversion" : "sum"}).reset_index()

d5b["CPA"] = np.where(d5b["Approved_Conversion"] > 0, d5b["Spent"] / d5b["Approved_Conversion"], 0)
d5b["CPS"] = np.where(d5b["Spent"] > 0, d5b["Approved_Conversion"]/d5b["Spent"], 0)
d5b["ROI"] = np.where(d5b["Spent"] > 0,(d5b["Approved_Conversion"]*100 - d5b["Spent"]) / d5b["Spent"],0)
d5b["ROI_Positive"] = d5b["ROI"].clip(lower=0)

Total_spend = d5b["Spent"].sum()
```

```
d5b["Recommended_Budget"] = (d5b["ROI_Positive"] / d5b["ROI_Positive"].sum()) * Total_spend
d5b["Recommended_Budget"] = d5b.apply(lambda row: min(row["Recommended_Budget"], row["Spent"]*2), axis=1)  # Here is maximum budget spent li
d5b["Exp_CPA"] = np.where(d5b["Approved_Conversion"] > 0, d5b["Recommended_Budget"] / d5b["Approved_Conversion"], 0)
d5b["Exp_ROI"] = np.where(d5b["Recommended_Budget"] > 0,(d5b["Approved_Conversion"]*100 - d5b["Recommended_Budget"]) / d5b["Recommended_Budg
heatmap2 = d5b.style.background_gradient(cmap="YlGnBu")
heatmap2
```

	xyz_campaign_id	Spent	Approved_Conversion	CPA	CPS	ROI	ROI_Positive	Recommended_Budget	Exp_CPA	Exp_R
0	916	149.710001	24	6.237917	0.160310	15.030993	15.030993	299.420001	12.475833	7.0154
1	936	2893.369999	183	15.810765	0.063248	5.324805	5.324805	5786.739998	31.621530	2.1624
2	1178	55662.149959	872	63.832741	0.015666	0.566594	0.566594	1589.781964	1.823144	53.8502

```
d5b.set_index("xyz_campaign_id")[["Spent", "Recommended_Budget"]].plot(kind="bar", figsize=(10,6))
plt.title("Spend vs Recommended Budget by XYZ Campaign ID")
plt.ylabel("Amount")
plt.xlabel("XYZ Campaign ID")
plt.xticks(rotation = 0)
plt.show()
```




```
d5b.set_index("xyz_campaign_id")[["CPA","Exp_CPA","ROI","Exp_ROI"]].plot(kind="bar", figsize=(10,6))
plt.title("Spend vs Recommended Budget by XYZ Campaign ID")
plt.ylabel("Amount")
plt.xlabel("XYZ Campaign ID")
plt.xticks(rotation = 0)
plt.show()
```