YieldGroupAnalysis

February 26, 2021

1 Importing the Libraries and Data

[2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
Q1=pd.read_csv(r"C:\Users\Yakov\Downloads\FirstQOf2020YieldGroupAnalysis.csv")
Q2=pd.read_csv(r"C:\Users\Yakov\Downloads\2QOf2020YieldGroupAnalysis.csv")
Q3=pd.read_csv(r"C:\Users\Yakov\Downloads\3QOf2020YieldGroupAnalysis.csv")
Q4=pd.read_csv(r"C:\Users\Yakov\Downloads\4QOf2020YieldGroupAnalysis.csv")
```

2 Data Cleaning

[3]:
```
# Renaming some incorrectly downloaded column names
Q1=Q1.rename(columns={"Yield group estimated revenue (â¸¬)": "Yield group␣
 ↪estimated revenue (€)", "Yield group estimated CPM (â¸¬)": "Yield group␣
 ↪estimated CPM (€)","Mediation third-party eCPM (â¸¬)":"Mediation third-party␣
 ↪eCPM (€)",})
Q3=Q3.rename(columns={"Yield group estimated revenue (â¸¬)": "Yield group␣
 ↪estimated revenue (€)", "Yield group estimated CPM (â¸¬)": "Yield group␣
 ↪estimated CPM (€)","Mediation third-party eCPM (â¸¬)":"Mediation third-party␣
 ↪eCPM (€)",})
```

3 Visualization

[296]:
```
# Making a dataset consisted of all 4 quarters
df=pd.concat([Q1,Q2,Q3,Q4])
df=df.drop(columns="Unnamed: 14")
```

[81]:
```
df.head(5)
```

[81]:

	Yield group	Country	Demand channel	\
0	2nd Level impressions	Hungary	Ad Exchange	
1	2nd Level impressions	Italy	Open Bidding	
2	2nd Level impressions	Kazakhstan	Ad Exchange	
3	2nd Level impressions	Turkey	Ad Exchange	

```
4  AN | EN AMP Pages (RON)  Afghanistan    Ad Exchange

  Creative size (delivered) Month and year Device category  \
0                   728x90         Jan-20         Desktop
1                   728x90         Jan-20         Desktop
2                   728x90         Jan-20         Desktop
3                   728x90         Jan-20         Desktop
4                  300x250         Jan-20      Smartphone

                         Browser       Yield partner type  \
0                 Google Chrome  DoubleClick Ad Exchange
1                 Google Chrome             Open bidding
2                 Firefox Other  DoubleClick Ad Exchange
3  Microsoft Internet Explorer 11  DoubleClick Ad Exchange
4                 Google Chrome  DoubleClick Ad Exchange

   Yield group impressions  Yield group estimated revenue (€)  \
0                      2.0                                0.00
1                      1.0                                0.00
2                      1.0                                0.00
3                      1.0                                0.00
4                     94.0                                0.01

   Yield group estimated CPM (€)  Mediation fill rate  Mediation passbacks  \
0                           0.05                 0.0                  0.0
1                           0.27                 0.0                  0.0
2                           0.01                 0.0                  0.0
3                           0.04                 0.0                  0.0
4                           0.15                 0.0                  0.0

   Mediation third-party eCPM (€)
0                             0.0
1                             0.0
2                             0.0
3                             0.0
4                             0.0
```

[298]:
```python
#df["Month and year"] = pd.to_datetime(df["Month and year"], format='%b-%y')
import datetime
arr=['']*len(df)
j=0
for i in df.loc[:,"Month and year"]:
    #df["Month and year"]=dt.strptime(df["Month and year"],"%y-%m-%d").dt.
 ↪strftime('%b')
    i=str(i).replace(" 00:00:00",'')
    #print(datetime.datetime.strptime(str(i),"%Y-%m-%d").strftime("%b"))
    arr[j]=datetime.datetime.strptime(str(i),"%b-%y").strftime("%b")
```

```
    #i.to_pydatetime()
    #print(type(i))
    #datetime.fromtimestamp(i).strftime("%A, %B %d, %Y %I:%M:%S")
    j=j+1

#for i in df.loc[:,"Month and year"]:
 #   print(type(str(i)))

df["Month"]=arr
df.head(10)
```

[298]:

	Yield group	Country	Demand channel	Creative size (delivered)	Month and year	Device category	Browser	Yield partner type
0	2nd Level impressions	Hungary	Ad Exchange	728x90	Jan-20	Desktop	Google Chrome	DoubleClick Ad Exchange
1	2nd Level impressions	Italy	Open Bidding	728x90	Jan-20	Desktop	Google Chrome	Open bidding
2	2nd Level impressions	Kazakhstan	Ad Exchange	728x90	Jan-20	Desktop	Firefox Other	DoubleClick Ad Exchange
3	2nd Level impressions	Turkey	Ad Exchange	728x90	Jan-20	Desktop	Microsoft Internet Explorer 11	DoubleClick Ad Exchange
4	AN | EN AMP Pages (RON)	Afghanistan	Ad Exchange	300x250	Jan-20	Smartphone	Google Chrome	DoubleClick Ad Exchange
5	AN | EN AMP Pages (RON)	Afghanistan	Ad Exchange	300x250	Jan-20	Smartphone	Safari (iPhone/iPod)	DoubleClick Ad Exchange
6	AN | EN AMP Pages (RON)	Afghanistan	Ad Exchange	300x250	Feb-20	Desktop	Google Chrome	DoubleClick Ad Exchange
7	AN | EN AMP Pages (RON)	Afghanistan	Ad Exchange	300x250	Feb-20	Smartphone	Google Chrome	DoubleClick Ad Exchange
8	AN | EN AMP Pages (RON)	Afghanistan	Ad Exchange	300x250	Feb-20	Smartphone	Safari (iPhone/iPod)	DoubleClick Ad Exchange
9	AN | EN AMP Pages (RON)	Afghanistan	Ad Exchange	300x250	Feb-20	Tablet	Google Chrome	DoubleClick Ad Exchange

```
   Yield group impressions  Yield group estimated revenue (€)  \
0                      2.0                               0.00
1                      1.0                               0.00
2                      1.0                               0.00
3                      1.0                               0.00
4                     94.0                               0.01
5                     34.0                               0.00
6                      4.0                               0.00
7                    260.0                               0.11
8                     21.0                               0.01
9                     22.0                               0.00

   Yield group estimated CPM (€)  Mediation fill rate  Mediation passbacks  \
0                           0.05                  0.0                  0.0
1                           0.27                  0.0                  0.0
2                           0.01                  0.0                  0.0
3                           0.04                  0.0                  0.0
4                           0.15                  0.0                  0.0
5                           0.13                  0.0                  0.0
6                           0.55                  0.0                  0.0
7                           0.42                  0.0                  0.0
8                           0.27                  0.0                  0.0
9                           0.22                  0.0                  0.0

   Mediation third-party eCPM (€) Month
0                             0.0   Jan
1                             0.0   Jan
2                             0.0   Jan
3                             0.0   Jan
4                             0.0   Jan
5                             0.0   Jan
6                             0.0   Feb
7                             0.0   Feb
8                             0.0   Feb
9                             0.0   Feb
```

[299]:
```
# Droping the Month and year column as we already have a nicely formatted month␣
↪column
df=df.drop(columns=["Month and year"])
```

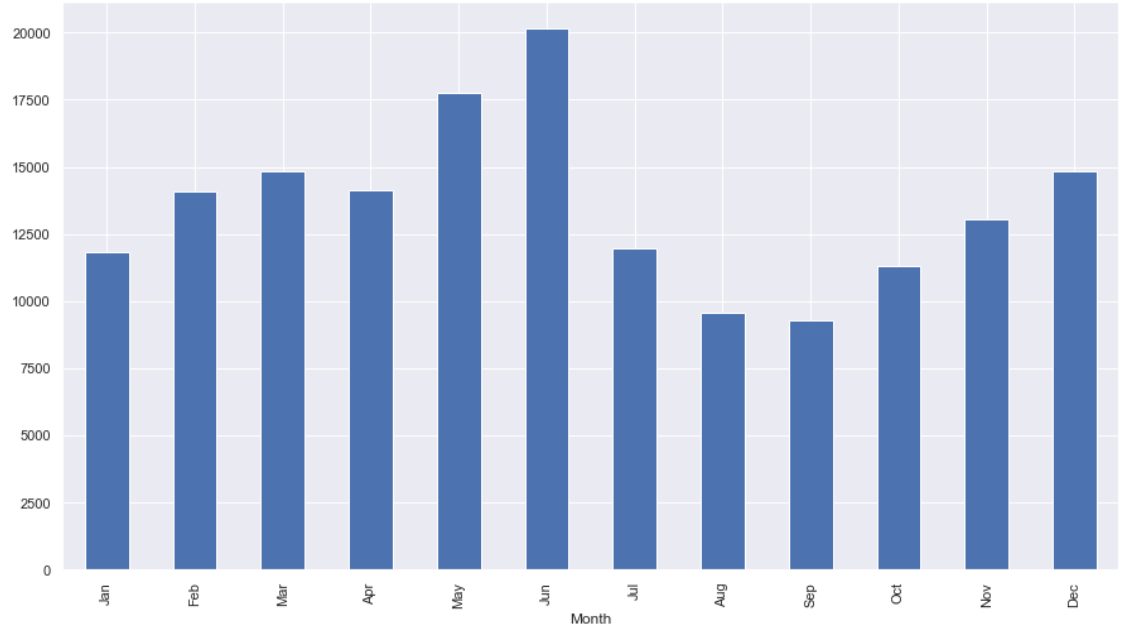
3.0.1 Plotting revenue

[313]:
```
# Preparing the dataset for plotting the revenue
df_time_revenue=pd.DataFrame(df.groupby(["Month"]).agg({"Yield group estimated␣
↪revenue (€)":['sum']}))
df_time_revenue=df_time_revenue.reset_index()
df_time_revenue.head(5)
```

[313]:

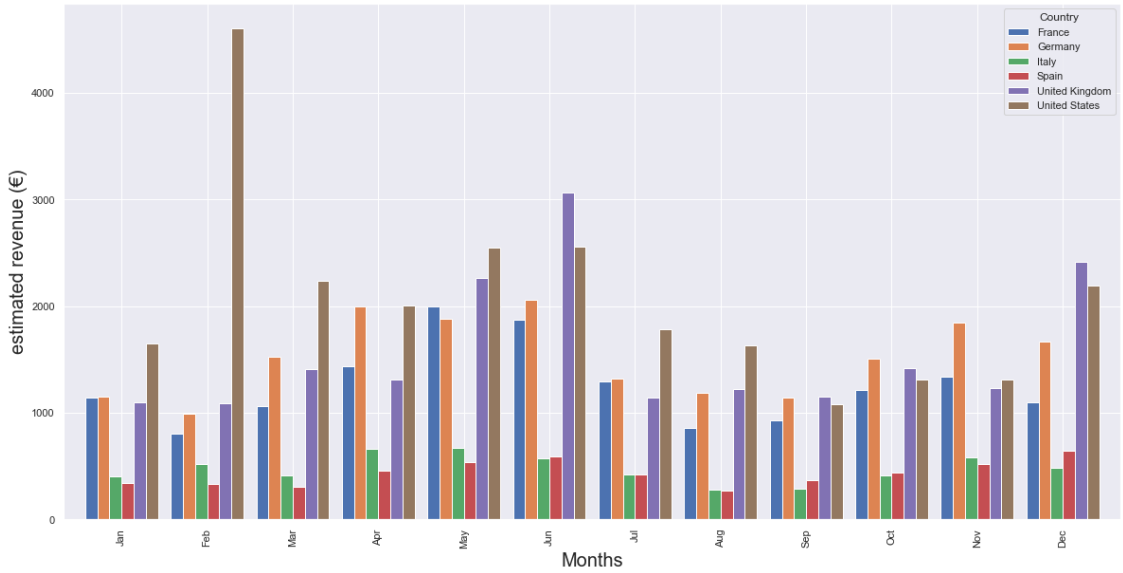
	Month	Yield group estimated revenue (€)
		sum
0	Apr	14122.03
1	Aug	9580.19
2	Dec	14849.36
3	Feb	14068.93
4	Jan	11847.84

[314]:
```
# Plotting bar chart
field = "Month"
month_order = ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep",␣
 ↪"Oct", "Nov", "Dec"]
ax = df_time_revenue.set_index(field).loc[month_order].plot(kind="bar",␣
 ↪legend=False)
```



[317]:
```
from matplotlib import pyplot as plt
# Plotting the revenue by core countries
df_core_revenue=pd.DataFrame(df.groupby(["Month", "Country"]).agg({"Yield group␣
 ↪estimated revenue (€)":['sum']}))
df_core_revenue=df_core_revenue.reset_index()
df_core_revenue=df_core_revenue[(df_core_revenue["Country"]).isin(["Italy",␣
 ↪"Spain", "Germany", "United Kingdom", "United States", "France"])]
df_core_revenue.columns = df_core_revenue.columns.map(lambda x: x[0])
#df_core_revenue=df_core_revenue.pivot("Month", "Country","Yield group␣
 ↪estimated revenue (€)")
#df_core_revenue = df_core_revenue.set_index("Month").loc[month_order]
```

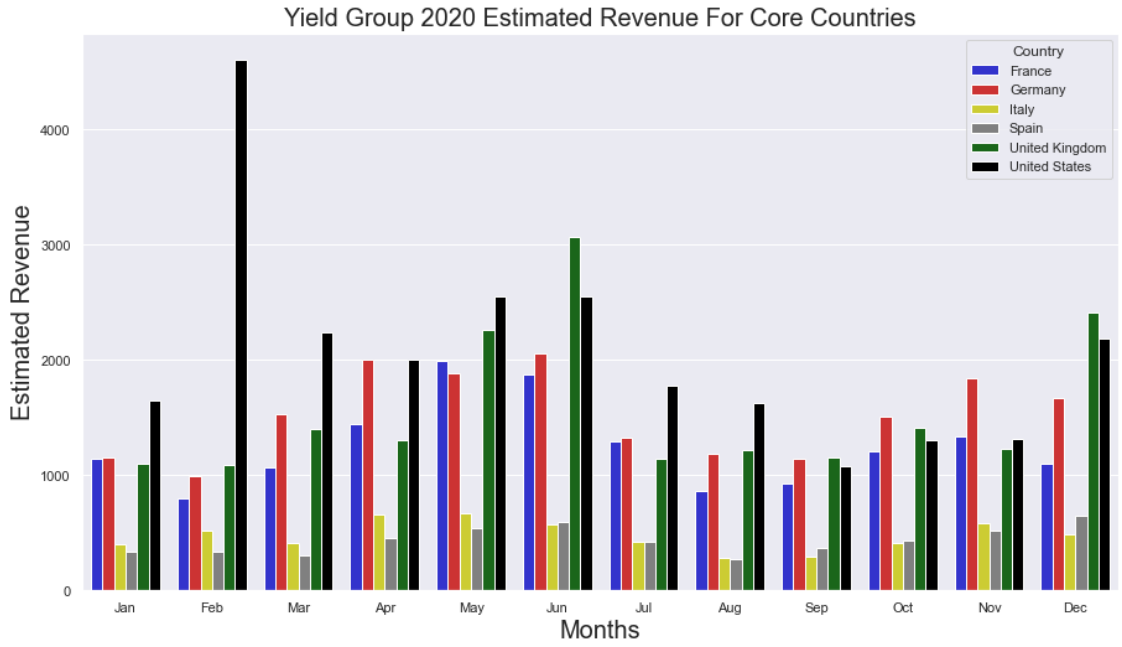
```
#df_core_revenue=df_core_revenue.pivot( "Country","Yield group estimated␣
 ↪revenue (€)")
df_core_revenue["Month"] = pd.Categorical(
        df_core_revenue["Month"], categories = month_order,ordered=True
            )
df_core_revenue.sort_values("Month",inplace=True)
df_pivoted = df_core_revenue.pivot("Month", "Country","Yield group estimated␣
 ↪revenue (€)")
#fig , axs = plt.subplots()
#plt.bar(df_pivoted)
df_pivoted.plot(kind="bar", figsize=(20,10),width=0.85)
plt.ylabel("estimated revenue (€)", fontsize=20)
plt.xlabel("Months", fontsize=20)


plt.show()
#df_core_revenue.plot(kind="bar")

#x = df_core_revenue.plot(rot=0, color={"Spain": "red", "France":"blue",␣
 ↪"United Kingdom": "green", "United States": "purple", "Germany" :"black",␣
 ↪"Italy": "yellow"})
#lt.show
#set_index("Month").loc[month_order].
```



[318]:
```
sns.set(rc={'figure.figsize':(15,8.27)})
plt.figure()
```

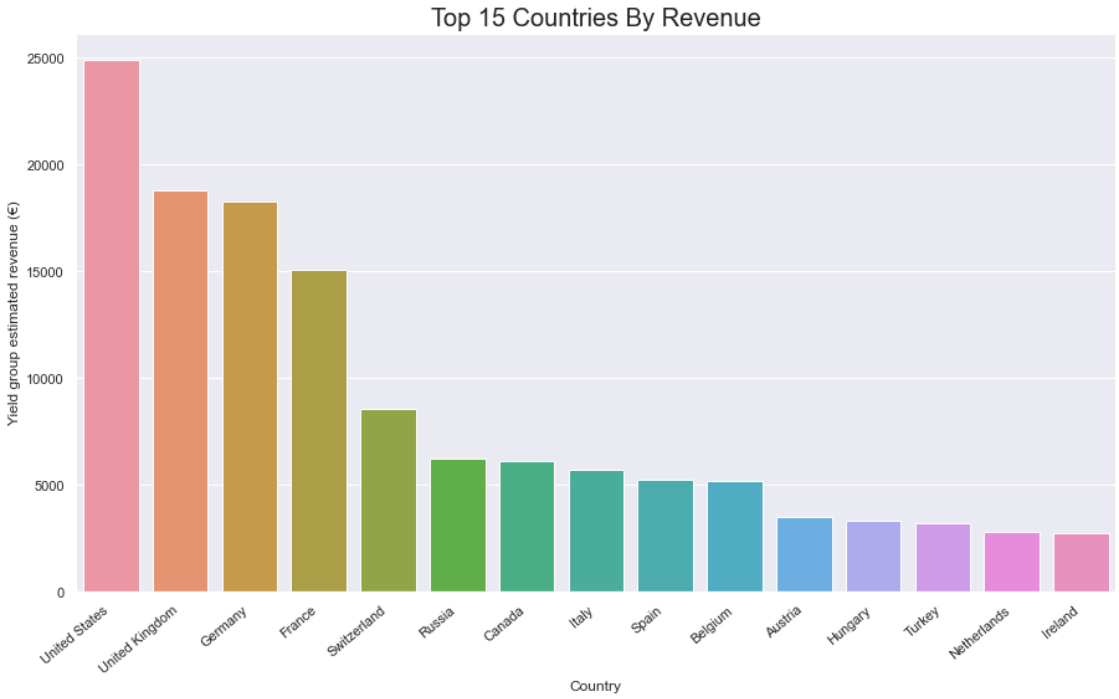
```
ax = sns.barplot(data=df_core_revenue, x="Month", y="Yield group estimated␣
 ↪revenue (€)", hue="Country",palette=['blue', 'red', 'yellow',␣
 ↪'grey',"green","black"], saturation=0.6)
ax.set_title("Yield Group 2020 Estimated Revenue For Core Countries",␣
 ↪fontsize=20)
ax.set_ylabel("Estimated Revenue", fontsize="20")
ax.set_xlabel("Months", fontsize="20")
```

[318]: Text(0.5, 0, 'Months')



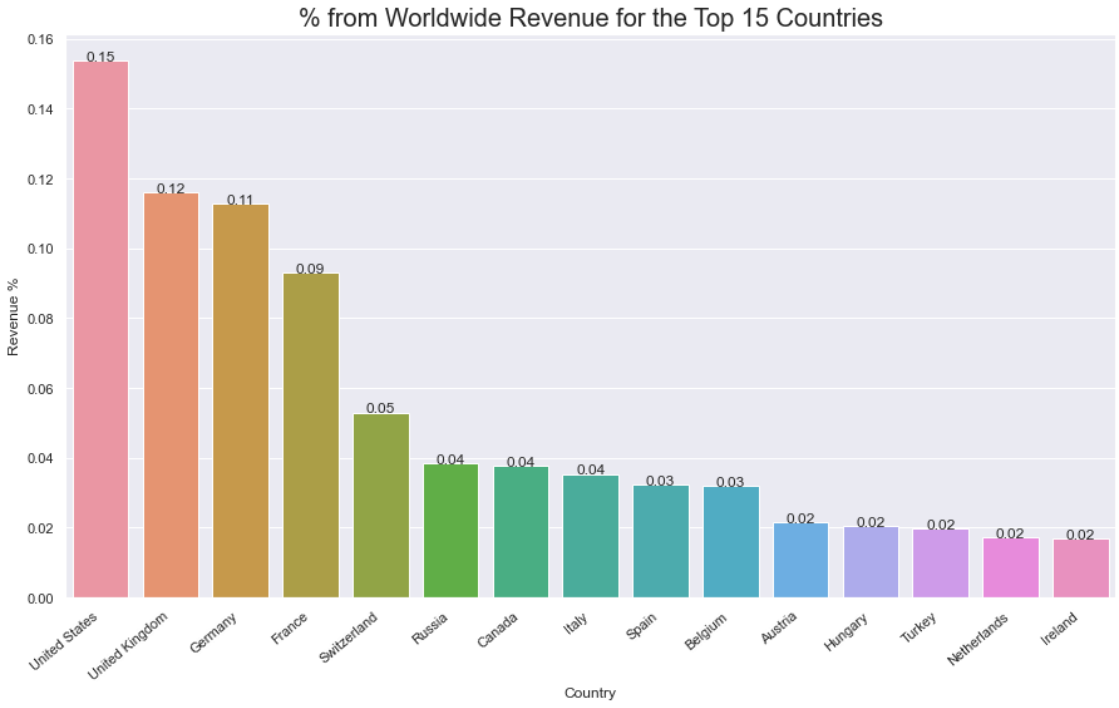
[240]:
```
# Plotting the distribution of whole year revenue
yearly_revenue = pd.DataFrame(df.groupby(["Country"]).agg({"Yield group␣
 ↪estimated revenue (€)":['sum']}))
yearly_revenue = yearly_revenue.reset_index()
yearly_revenue.columns = yearly_revenue.columns.map(lambda x: x[0])
yearly_revenue = yearly_revenue.sort_values(by=["Yield group estimated revenue␣
 ↪(€)"], ascending=False)
yearly_revenue=yearly_revenue[:15]
#df_core_revenue=df_core_revenue[(df_core_revenue["Country"]).isin(["Italy",␣
 ↪"Spain", "Germany", "United Kingdom", "United States", "France"])]
#yearly_revenue=yearly_revenue[(yearly_revenue["Country"]).isin(["Italy",␣
 ↪"Spain", "Germany", "United Kingdom", "United States", "France"])]
ax = sns.barplot(data=yearly_revenue.sort_values(by=["Yield group estimated␣
 ↪revenue (€)"], ascending=False), x="Country", y="Yield group estimated␣
 ↪revenue (€)")
```

```
ax.set_xticklabels(ax.get_xticklabels(), rotation=40, ha="right")
ax.set_title("Top 15 Countries By Revenue", fontsize=20)
plt.show()
```



[263]:
```
yearly_revenue = pd.DataFrame(df.groupby(["Country"]).agg({"Yield group␣
 ↪estimated revenue (€)":['sum']}))
yearly_revenue = yearly_revenue.reset_index()
yearly_revenue.columns = yearly_revenue.columns.map(lambda x: x[0])
yearly_revenue["Revenue %"]=yearly_revenue["Yield group estimated revenue (€)"]/
 ↪yearly_revenue["Yield group estimated revenue (€)"].sum()
yearly_revenue = yearly_revenue.sort_values(by = ["Revenue %"], ascending =␣
 ↪False)
yearly_revenue = yearly_revenue[:15]


# Function to show values above the bars in the bar plot
def show_values_on_bars(axs):
    def _show_on_single_plot(ax):
        for p in ax.patches:
            _x = p.get_x() + p.get_width() / 2
            _y = p.get_y() + p.get_height()
            value = '{:.2f}'.format(p.get_height())
            ax.text(_x, _y, value, ha="center")
```

```
    if isinstance(axs, np.ndarray):
        for idx, ax in np.ndenumerate(axs):
            _show_on_single_plot(ax)
    else:
        _show_on_single_plot(axs)


ax = sns.barplot(data=yearly_revenue, x="Country", y="Revenue %")
ax.set_xticklabels(ax.get_xticklabels(), rotation=40, ha="right")
ax.set_title("% from Worldwide Revenue for the Top 15 Countries", fontsize=20)
show_values_on_bars(ax)
```



We can see that the percentage of the revenue is low if we take the countries individually. Let's see how EU, UK and US perform compared to the rest of the world.

[273]:
```
yearly_revenue = pd.DataFrame(df.groupby(["Country"]).agg({"Yield group␣
↪estimated revenue (€)":['sum']}))
yearly_revenue = yearly_revenue.reset_index()
yearly_revenue.columns = yearly_revenue.columns.map(lambda x: x[0])
yearly_revenueEU=yearly_revenue
yearly_revenueEU = yearly_revenue[(yearly_revenueEU["Country"]).
↪isin(['Austria', 'Belgium', 'France', 'Germany', 'Hungary', 'Ireland',␣
↪'Netherlands',
'Portugal', 'United Kingdom', 'United States', 'Bulgaria', 'Croatia', 'Cyprus',
'Czechia', 'Denmark', 'Estonia', 'Finland', 'Greece', 'Italy', 'Latvia',
```

```
 'Lithuania', 'Luxembourg', 'Malta', 'Poland', 'Romania', 'Slovakia',␣
 ↪'Slovenia',
 'Spain', 'Sweden'])]

yearly_revenueROW = yearly_revenue
yearly_revenueROW = yearly_revenue[~(yearly_revenue["Country"]).
 ↪isin(['Austria', 'Belgium', 'France', 'Germany', 'Hungary', 'Ireland',␣
 ↪'Netherlands',
 'Portugal', 'United Kingdom', 'United States', 'Bulgaria', 'Croatia', 'Cyprus',
 'Czechia', 'Denmark', 'Estonia', 'Finland', 'Greece', 'Italy', 'Latvia',
 'Lithuania', 'Luxembourg', 'Malta', 'Poland', 'Romania', 'Slovakia',␣
 ↪'Slovenia',
 'Spain', 'Sweden'])]

print(yearly_revenueEU)
yearly_revenueROW
```

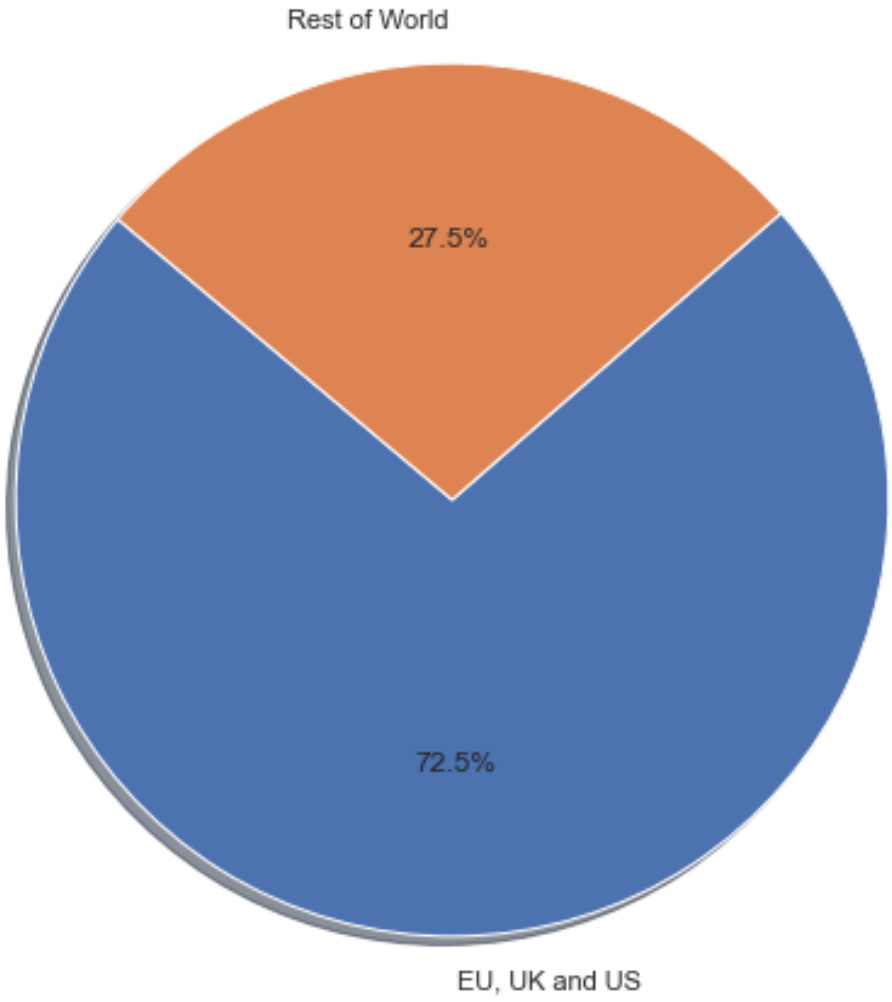
	Country	Yield group estimated revenue (€)
12	Austria	3474.07
18	Belgium	5166.34
30	Bulgaria	206.31
48	Croatia	84.46
50	Cyprus	466.62
51	Czechia	397.52
53	Denmark	744.37
62	Estonia	241.41
69	Finland	562.20
70	France	15055.35
75	Germany	18276.23
78	Greece	1885.21
91	Hungary	3296.88
96	Ireland	2737.65
98	Italy	5717.68
110	Latvia	381.36
116	Lithuania	197.37
117	Luxembourg	412.38
124	Malta	293.71
142	Netherlands	2791.03
162	Poland	779.02
163	Portugal	2431.20
168	Romania	1174.22
186	Slovakia	169.74
187	Slovenia	104.84
192	Spain	5239.29
196	Sweden	1496.02
217	United Kingdom	18799.71
218	United States	24902.27

[273]:

	Country	Yield group estimated revenue (€)
0	Afghanistan	6.33
1	Albania	46.33
2	Algeria	442.73
3	American Samoa	0.05
4	Andorra	5.47
..	…	…
224	Vietnam	44.56
225	Western Sahara	1.14
226	Yemen	5.69
227	Zambia	49.90
228	Zimbabwe	25.39

[200 rows x 2 columns]

[274]:
```
sumEU = yearly_revenueEU["Yield group estimated revenue (€)"].sum()
sumROW = yearly_revenueROW["Yield group estimated revenue (€)"].sum()
labels = ["EU, UK and US", "Rest of World"]
sizes = [sumEU, sumROW]

plt.pie(sizes, labels=labels,
        autopct='%1.1f%%', shadow=True, startangle=140)
plt.show()
```

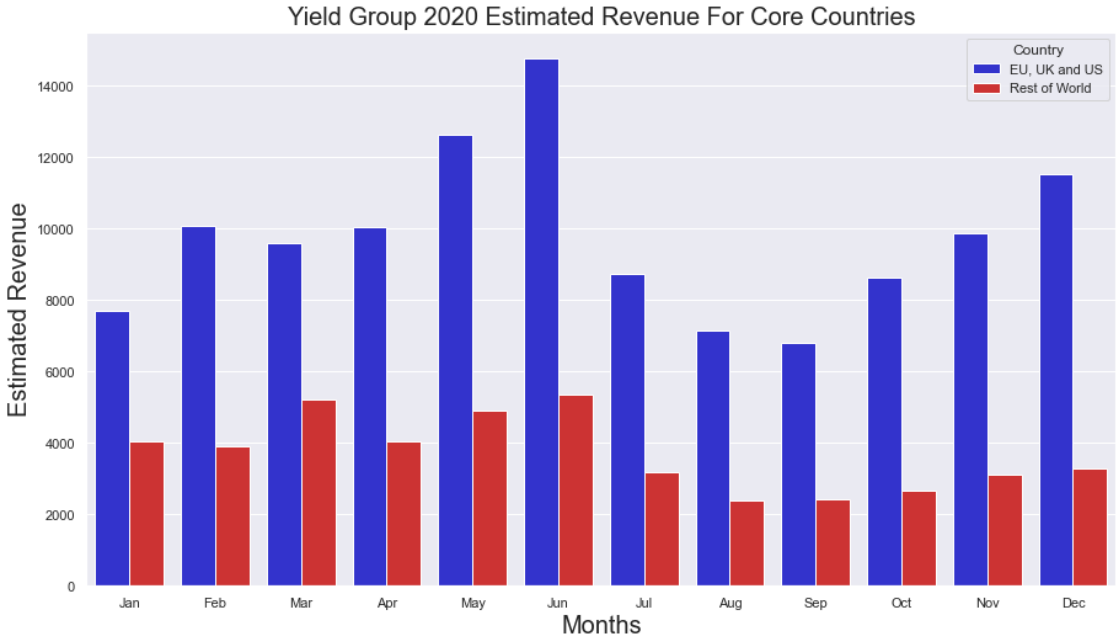


Let's see the monthly figures for EU, UK and US compared to the rest of the world

[329]:
```
df_core_revenue=pd.DataFrame(df.groupby(["Month", "Country"]).agg({"Yield group␣
↪estimated revenue (€)":['sum']}))
df_core_revenue=df_core_revenue.reset_index()
df_core_revenue.columns = df_core_revenue.columns.map(lambda x: x[0])

df_core_revenue.loc[df_core_revenue["Country"].isin(['Austria', 'Belgium',␣
↪'France', 'Germany', 'Hungary', 'Ireland', 'Netherlands',
'Portugal', 'United Kingdom', 'United States', 'Bulgaria', 'Croatia', 'Cyprus',
'Czechia', 'Denmark', 'Estonia', 'Finland', 'Greece', 'Italy', 'Latvia',
'Lithuania', 'Luxembourg', 'Malta', 'Poland', 'Romania', 'Slovakia',␣
↪'Slovenia',
```

```
 'Spain', 'Sweden']),"Country"] = "EU, UK and US"
df_core_revenue.loc[df_core_revenue["Country"] != "EU, UK and US", "Country"]=␣
 ↪"Rest of World"
```

[295]:
```
df_core_revenue = pd.DataFrame(df_core_revenue.groupby(["Month", "Country"]).
 ↪agg({"Yield group estimated revenue (€)":['sum']}))
df_core_revenue = df_core_revenue.reset_index()
df_core_revenue.columns = df_core_revenue.columns.map(lambda x: x[0])

sns.set(rc={'figure.figsize':(15,8.27)})
plt.figure()

ax = sns.barplot(data=df_core_revenue, x="Month", y="Yield group estimated␣
 ↪revenue (€)", hue="Country",palette=['blue', 'red'], saturation=0.6)
ax.set_title("Yield Group 2020 Estimated Revenue For Core Countries",␣
 ↪fontsize=20)
ax.set_ylabel("Estimated Revenue", fontsize="20")
ax.set_xlabel("Months", fontsize="20")
plt.show()
```

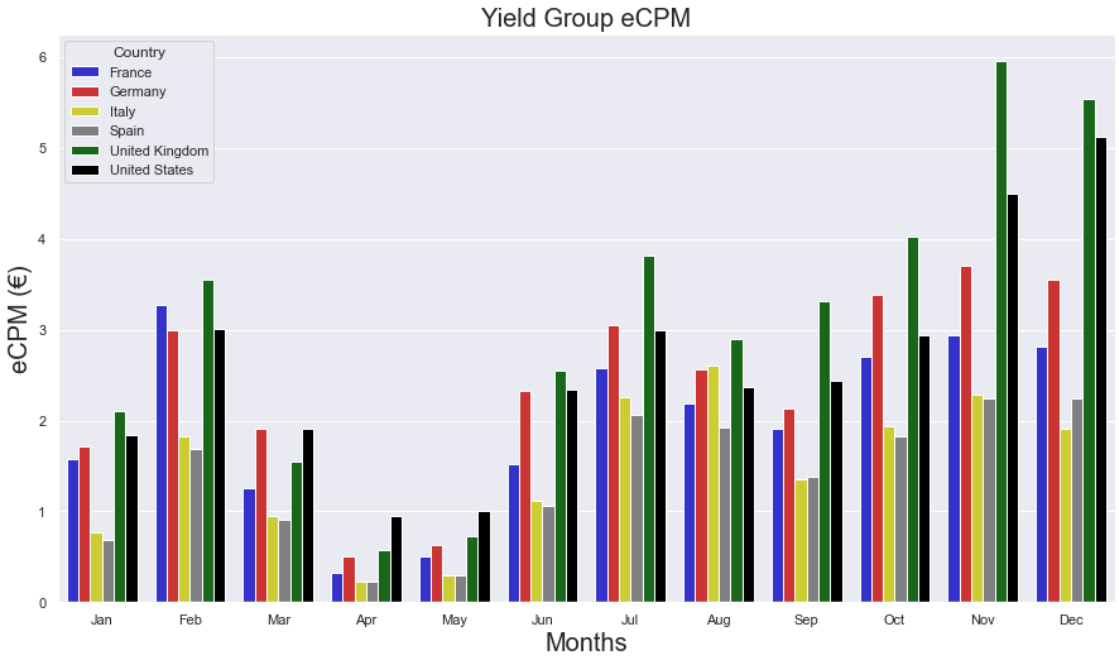


3.1 Visualizing eCPM

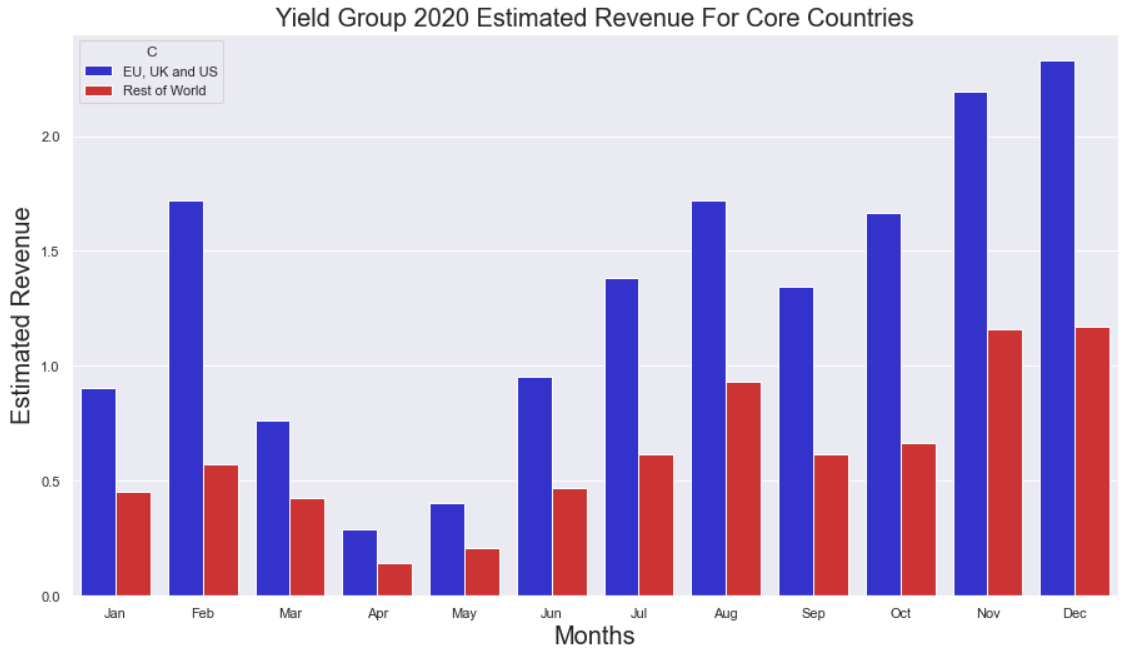
[339]:
```
import warnings
warnings.filterwarnings('ignore')
```

```
df_ecpm = pd.DataFrame(df.groupby(["Month", "Country"]).agg({"Yield group␣
 ↪estimated CPM (€)":['mean']}))
df_ecpm = df_ecpm.reset_index()
df_ecpm.columns = df_ecpm.columns.map(lambda x: x[0])

df_ecpm_core = df_ecpm[(df_ecpm["Country"]).isin(["United States", "Spain",␣
 ↪"France", "United Kingdom", "Italy", "Germany"])]

df_ecpm_core["Month"] = pd.Categorical(
        df_ecpm_core["Month"], categories = month_order,ordered=True
            )
df_ecpm_core.sort_values("Month", inplace=True)


sns.set(rc={'figure.figsize':(15,8.27)})
plt.figure()

ax = sns.barplot(data=df_ecpm_core, x="Month", y="Yield group estimated CPM␣
 ↪(€)", hue="Country",palette=['blue', 'red', 'yellow',␣
 ↪'grey',"green","black"], saturation=0.6)
ax.set_title("Yield Group eCPM", fontsize=20)
ax.set_ylabel("eCPM (€)", fontsize="20")
ax.set_xlabel("Months", fontsize="20")
```

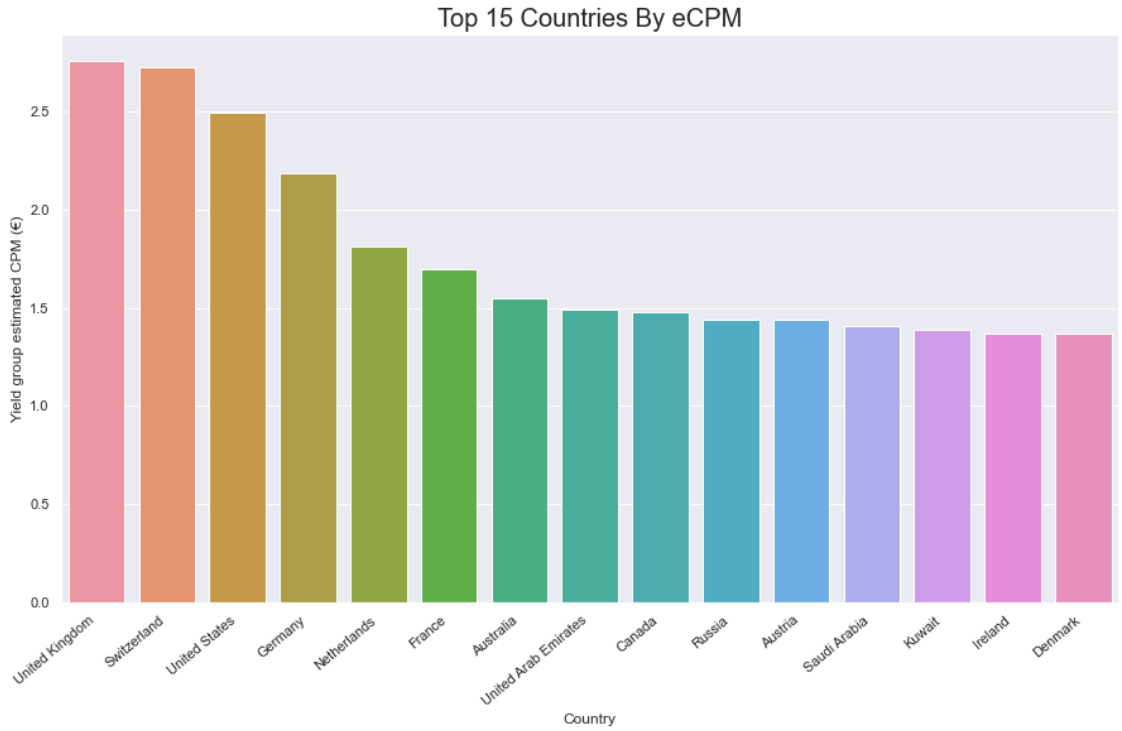
[339]: Text(0.5, 0, 'Months')



[345]:
```
#df_ecpm.loc[df_ecpm["Country"].isin(['Austria', 'Belgium', 'France',␣
 ↪'Germany', 'Hungary', 'Ireland', 'Netherlands',
 #'Portugal', 'United Kingdom', 'United States', 'Bulgaria', 'Croatia',␣
 ↪'Cyprus',
 #'Czechia', 'Denmark', 'Estonia', 'Finland', 'Greece', 'Italy', 'Latvia',
 #'Lithuania', 'Luxembourg', 'Malta', 'Poland', 'Romania', 'Slovakia',␣
 ↪'Slovenia',
 #'Spain', 'Sweden']),"Country"] = "EU, UK and US"
#df_ecpm.loc[df_ecpm["Country"] != "EU, UK and US", "Country"]= "Rest of World"

#df_ecpm = pd.DataFrame(df_ecpm.groupby(["Month","Country"]).mean())
#df_ecpm = df_ecpm.reset_index()
#df_ecpm.columns = df_ecpm.columns.map(lambda x : x[0])

sns.set(rc={'figure.figsize':(15,8.27)})
plt.figure()

ax = sns.barplot(data=df_ecpm, x="M", y="Y", hue="C",palette=['blue', 'red'],␣
 ↪saturation=0.6)
ax.set_title("Yield Group 2020 Estimated Revenue For Core Countries",␣
 ↪fontsize=20)
ax.set_ylabel("Estimated eCPM", fontsize="20")
ax.set_xlabel("Months", fontsize="20")
plt.show()
```



[352]:
```
import warnings
warnings.filterwarnings('ignore')
df_ecpm = pd.DataFrame(df.groupby(["Country"]).agg({"Yield group estimated CPM␣
↪(€)":['mean']}))
df_ecpm = df_ecpm.reset_index()
df_ecpm.columns = df_ecpm.columns.map(lambda x: x[0])
df_ecpm = df_ecpm.sort_values(by=["Yield group estimated CPM (€)"],␣
↪ascending=False)
df_ecpm = df_ecpm[:15]

ax = sns.barplot(data=df_ecpm, x="Country", y="Yield group estimated CPM (€)")
ax.set_xticklabels(ax.get_xticklabels(), rotation=40, ha="right")
ax.set_title("Top 15 Countries By eCPM", fontsize=20)
plt.show()
```

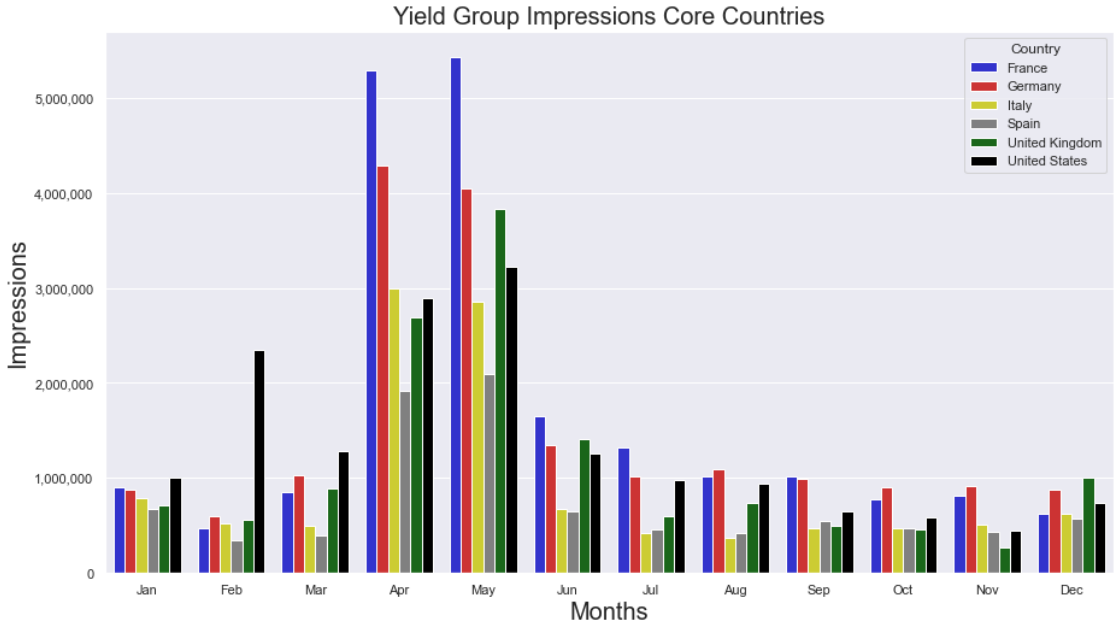


We can clearly see that the eCPM has dropped in March to May due to the demand decline.

3.2 Visualizing impressions

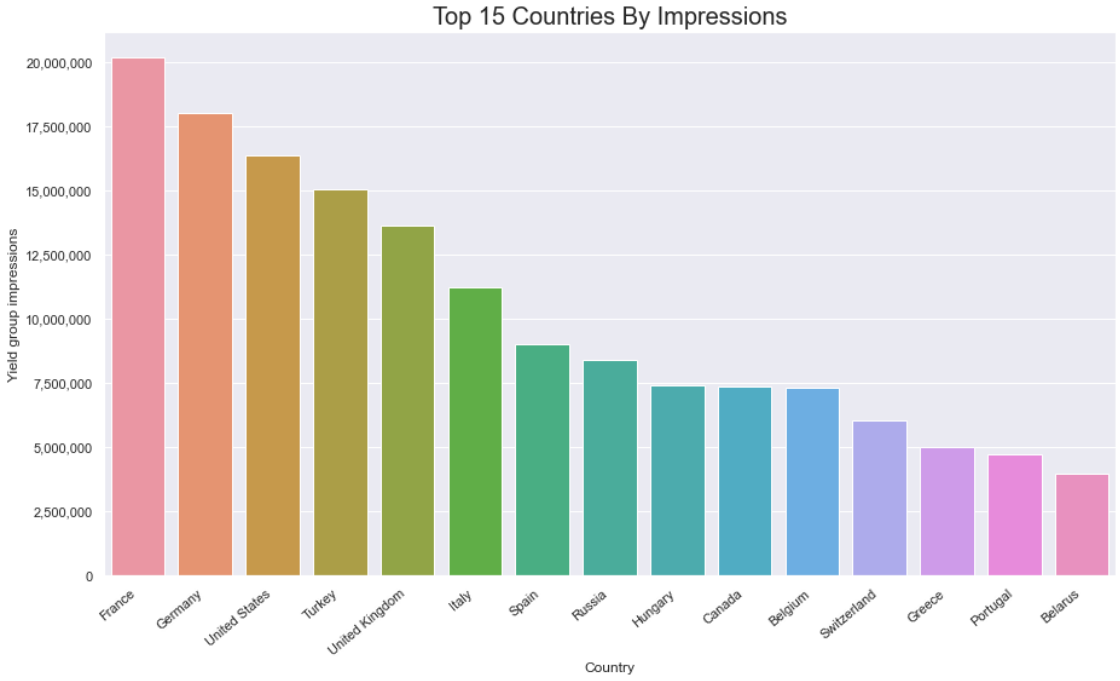
[391]:
```
df_impressions=pd.DataFrame(df.groupby(["Month", "Country"]).agg({"Yield group␣
↪impressions":['sum']}))
df_impressions=df_impressions.reset_index()
df_impressions.columns = df_impressions.columns.map(lambda x: x[0])
```

```
#df_impressions = df_impressions.sort_values(by=["Yield group impressions"],␣
 ↪ascending=False)
#df_impressions = df_impressions[:15]
df_impressions_core = df_impressions[(df_impressions["Country"]).isin(["United␣
 ↪States", "Spain", "France", "United Kingdom", "Italy", "Germany"])]


plt.figure()
ax = sns.barplot(data=df_impressions_core, x="Month", y="Yield group␣
 ↪impressions", hue="Country",palette=['blue', 'red', 'yellow',␣
 ↪'grey',"green","black"], saturation=0.6)
ax.set_title("Yield Group Impressions Core Countries", fontsize=20)
ax.set_ylabel("Impressions", fontsize="20")
ax.set_xlabel("Months", fontsize="20")
ylabels = [f'{int(x):,}' for x in ax.get_yticks()]
ax.set_yticklabels(ylabels)
plt.show()
```



[389]:
```
import matplotlib.ticker as ticker
df_impressions=pd.DataFrame(df.groupby(["Country"]).agg({"Yield group␣
 ↪impressions":['sum']}))
df_impressions=df_impressions.reset_index()
df_impressions.columns = df_impressions.columns.map(lambda x: x[0])
df_impressions = df_impressions.sort_values(by=["Yield group impressions"],␣
 ↪ascending=False)
df_impressions = df_impressions[:15]
```

```
fig, ax = plt.subplots(figsize=(15, 8.27))
ax = sns.barplot(data=df_impressions, x="Country", y="Yield group impressions")
ax.set_xticklabels(ax.get_xticklabels(), rotation=40, ha="right")
ax.set_title("Top 15 Countries By Impressions", fontsize=20)
ylabels = [f'{int(x):,}' for x in ax.get_yticks()]
ax.set_yticklabels(ylabels)
plt.show()
```



[406]:
```
df_impressions=pd.DataFrame(df.groupby(["Month", "Country"]).agg({"Yield group␣
↪impressions":['sum']}))
df_impressions=df_impressions.reset_index()
df_impressions.columns = df_impressions.columns.map(lambda x: x[0])

df_impressions.loc[df_impressions["Country"].isin(['Austria', 'Belgium',␣
↪'France', 'Germany', 'Hungary', 'Ireland', 'Netherlands',
'Portugal', 'United Kingdom', 'United States', 'Bulgaria', 'Croatia', 'Cyprus',
'Czechia', 'Denmark', 'Estonia', 'Finland', 'Greece', 'Italy', 'Latvia',
'Lithuania', 'Luxembourg', 'Malta', 'Poland', 'Romania', 'Slovakia',␣
↪'Slovenia',
'Spain', 'Sweden']),"Country"] = "EU, UK and US"
df_impressions.loc[df_impressions["Country"] != "EU, UK and US", "Country"]=␣
↪"Rest of World"
```

```
df_impressions=pd.DataFrame(df_impressions.groupby(["Month","Country"]).
 ↪agg({"Yield group impressions":['sum']}))
df_impressions=df_impressions.reset_index()
df_impressions.columns = df_impressions.columns.map(lambda x: x[0])

ax = sns.barplot(data=df_impressions, x="Month", y="Yield group impressions",␣
 ↪hue="Country",palette=['blue', 'red'], saturation=0.6)
ax.set_title("Yield Group 2020 Impressions EU, UK and US compared to ROW",␣
 ↪fontsize=20)
ax.set_ylabel("Impressions", fontsize="20")
ax.set_xlabel("Months", fontsize="20")
plt.show()
```