

## KODLASAM PYTHON PROJESİ: NOBEL PRİZE VERİ SETİNİN İNCELENMESİ

```
In [1]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean_squared_error, r2_score

In [3]: df1= pd.read_csv(r'C:\ProgramData\anaconda3\Lib\site-packages\pandas\io\parsers\python_proje\asset-v1_Kodlasam+MS01+2023_06+type6

In [4]: df1= df1.drop_duplicates()
df1
```

Veri setinin manipülasyonunda ve görselleştirilmesinde kullanacağım kütüphaneleri ve veri setini yükledim.

### VERİ MANİPÜLASYONU

```
In [ ]: 
```

nobel barış ödülünü alan ülke için nobel aldığı yıl ölüm yılından sonra ise doğduğu ülke olarak organizasyon ülkesi olarak doldurulabilir

organizasyon ismi doğduğu ülke olabilir

```
In [7]: df1.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 911 entries, 0 to 910
Data columns (total 18 columns):
 #   Column              Non-Null Count  Dtype
---  --
 0   year                911 non-null   int64
 1   category            911 non-null   object
 2   prize               911 non-null   object
 3   motivation          823 non-null   object
 4   prize_share         911 non-null   object
 5   laureate_id         911 non-null   int64
 6   laureate_type       911 non-null   object
 7   full_name           911 non-null   object
 8   birth_date          883 non-null   object
 9   birth_city          883 non-null   object
10  birth_country       885 non-null   object
11  sex                 885 non-null   object
12  organization_name    665 non-null   object
13  organization_city    667 non-null   object
14  organization_country 667 non-null   object
15  death_date          593 non-null   object
16  death_city          576 non-null   object
17  death_country       582 non-null   object
dtypes: int64(2), object(16)
memory usage: 128.2+ KB
```

Veriyi kontrol ettiğimizde boş olan satırların olduğunu görüyoruz.

```
In [8]: df1.isnull().values.any()
```

```
Out[8]: True
```

```
In [9]: df1.isnull().sum()
```

```
Out[9]: year                0
category                  0
prize                     0
motivation               88
prize_share              0
laureate_id              0
laureate_type            0
full_name                0
birth_date              28
birth_city              28
birth_country           26
sex                     26
organization_name       246
organization_city       244
organization_country    244
death_date             318
death_city             335
death_country          329
dtype: int64
```

```
In [11]: df1['motivation'] = df1['motivation'].fillna(df1['prize'])
df1['motivation']
```

```
Out[11]: 0      "in recognition of the extraordinary services ..."
1      "in special recognition of his poetic composi..."
2      "for his work on serum therapy, especially its..."
3      The Nobel Peace Prize 1901
4      The Nobel Peace Prize 1901
...
906    "for his discoveries of mechanisms for autophagy"
907    "for his resolute efforts to bring the country..."
908    "for theoretical discoveries of topological ph..."
909    "for theoretical discoveries of topological ph..."
910    "for theoretical discoveries of topological ph..."
Name: motivation, Length: 911, dtype: object
```

```
In [12]: df1.isnull().sum()
```

```
Out[12]: year                0
category                    0
prize                       0
motivation                  0
prize_share                 0
laureate_id                 0
laureate_type               0
full_name                   0
birth_date                  28
birth_city                  28
birth_country               26
sex                         26
organization_name           246
organization_city           244
organization_country        244
death_date                  318
death_city                  335
death_country               329
dtype: int64
```

'motivation' ve 'prize' kategorisi benzer ifadeleri içerdiği için boş olan hücreleri 'Prize' kategorisinden yararlanarak dolduruyoruz. Çünkü veri setinin doğruluğunu etkilemiyor.

'Organization\_name' kategorisinde eksik olan hücreleri kontrol ettiğimizde boş olan hücreler olduğunu görüyoruz. 'organization\_name' sütununun boşluklarını Nobel ödülünü almadan önce vefat eden kişilerin ülkelerine atfedildiğini öğrendiğim için bir döngü oluşturarak boş hücreleri 'birth\_country' ile doldurdum. 'organization\_name', 'organization\_city' ve 'organization\_country' den bağımsız olması düşük olduğu için de 'organization\_name' de boş olan

hücreleri ülke ve şehir isimleri ile doldurdum. Veriyi kırpma verilerin doğruluğunu düşüreceği için bu bölümleri birbirleriyle ilişki bilgilerle doldurmak daha mantıklı olacağına karar verdim.

```
In [131]: df1['death_date'] = pd.to_datetime(df1['death_date'], errors='coerce')
for i, row in df1.iterrows():
    if pd.isna(row['death_date']) or (row['death_date'].year < row['year']):
        df1.at[i, 'organization_name'] = df1.at[i, 'birth_country']
```

'birth\_country' sütunundaki boşlukları 'organization\_country' bilgileriyle dolduruyoruz.

```
In [31]: # "peace" kategorisine sahip ödül sahiplerini filtrele
df1_peace = df1[df1['category'] == 'peace']

# Nobel Barış Ödülü alan kişilerin birth_country sütunundaki NaN veya null değerlerini organization_country ile doldur
df1_peace['organization_country'].fillna(df1_peace['birth_country'], inplace=True)
```

'Literature' ve 'peace' kategorileri genellikle tüm dünyaya atfedilen ödüller olduğu için bu kategorilerde boşluklar çok fazla. Bu nedenle 'birth\_country' ve 'organization\_country' bilgilerini kullanarak boş olan hücreleri doldurdum.

```
In [133]: # "Literature" kategorisine sahip ödül sahiplerini filtrele
df1_Literature = df1[df1['category'] == 'Literature']

# fillna işleminden önce ilgili sütunun değerlerini göster
print("Before fillna:")
print(df1_Literature[['organization_country', 'birth_country']])

# Nobel Edebiyat Ödülü alan kişilerin birth_country sütunundaki NaN veya null değerlerini organization_country ile doldur
df1_Literature['organization_country'].fillna(df1_Literature['birth_country'], inplace=True)

# fillna işleminden sonra ilgili sütunun değerlerini göster
print("\nAfter fillna:")
df1_Literature[['organization_country', 'birth_country']]
```

```
In [125]: # birth_country ve death_country sütunlarının eşleştiği kişilerin birth_city bilgilerini kullanarak
#death_city sütunundaki NaN veya null değerlerini doldur
mask = df1['birth_country'] == df1['death_country']
df1.loc[mask, 'death_city'] = df1.loc[mask, 'birth_city']
df1

In [193]: # Peace ve Literature kategorisindeki boş olan organization_name, organization_city ve organization_country sütunlarını doldur
category_names = ['Peace', 'Literature']
fill_values = {
    'organization_name': 'Nobel Foundation',
    'organization_city': 'Oslo',
    'organization_country': 'world'
}

# Bu kategorilere ait satırları seç
mask = df1['category'].isin(category_names)

# Seçilen satırların belirli sütunlarındaki boş değerleri doldur
df1.loc[mask, 'organization_name'] = df1.loc[mask, 'organization_name'].fillna(fill_values['organization_name'])
df1.loc[mask, 'organization_city'] = df1.loc[mask, 'organization_city'].fillna(fill_values['organization_city'])
df1.loc[mask, 'organization_country'] = df1.loc[mask, 'organization_country'].fillna(fill_values['organization_country'])

# Sonuçları kontrol etme
print(df1[df1['category'].isin(category_names)][['category', 'organization_name', 'organization_city', 'organization_country']])
```

Doğum tarihlerinde boşlukları ise her 20 yıl için ortalama olarak doldurdum.

```
In [141]: # Doğum yıllarını elde etme ve gruplama
df1['birth_year'] = df1['birth_date'].dt.year
birth_year_bins = list(range(1900, 2021, 20)) # Her 20 yılda bir grupta
df1['birth_year_group'] = pd.cut(df1['birth_year'], bins=birth_year_bins, right=False)

# Her grup için doğum yılının ortalamasını hesaplama
group_means = df1.groupby('birth_year_group')['birth_year'].mean()

# NaN değerleri grup ortalamaları ile doldurma fonksiyonu
def fill_birth_year(row):
    if pd.isna(row['birth_year']):
        group = row['birth_year_group']
        return group_means[group]
    return row['birth_year']

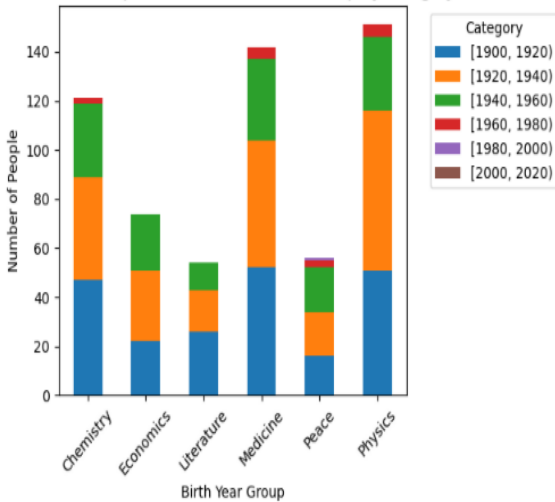
df1['birth_year'] = df1.apply(fill_birth_year, axis=1)

# Doldurulmuş doğum yıllarını tekrar datetime formatına çevirme
df1['birth_date'] = pd.to_datetime(df1['birth_year'], format='%Y')

# Sonuçları kontrol etme
print(df1[['birth_date', 'birth_year', 'birth_year_group']])
```

Bu verileri bir de grafik üzerinde görselleştirdim.

Number of People Born in Each 20-Year Group by Category



```
In [142]: # birth_date sütununu datetime formatına çevir
df1['birth_date'] = pd.to_datetime(df1['birth_date'])

# Her bir kategori için doğum yılı gruplaması yap
#birth_year = df1['birth_date'].dt.year
#birth_year_bins = list(range(1900, 2021, 20)) # Her 20 yılda bir grupta
#birth_year_group = pd.cut(df1['birth_year'], bins=birth_year_bins, right=False)

# Her bir kategori ve doğum yılı grubu için kişi sayısını hesaplama
grouped = df1.groupby(['category', 'birth_year_group']).size().unstack(fill_value=0)

# Grafiği oluştur
plt.figure(figsize=(12, 6))
grouped.plot(kind='bar', stacked=True)
plt.xlabel('Birth Year Group')
plt.ylabel('Number of People')
plt.title('Number of People Born in Each 20-Year Group by Category')
plt.xticks(rotation=45)
plt.legend(title='Category', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

## Proje Soruları

### 1) Nobel Ödüllerini en çok kazanan ilk on ülkeyi bulunuz.

```
In [194]: # organization_country kırılımında ülkeleri gruplandırma ve her ülkenin kazandığı ödül sayısını sayma
top_countries = df1.groupby('organization_country')['prize'].count().reset_index()

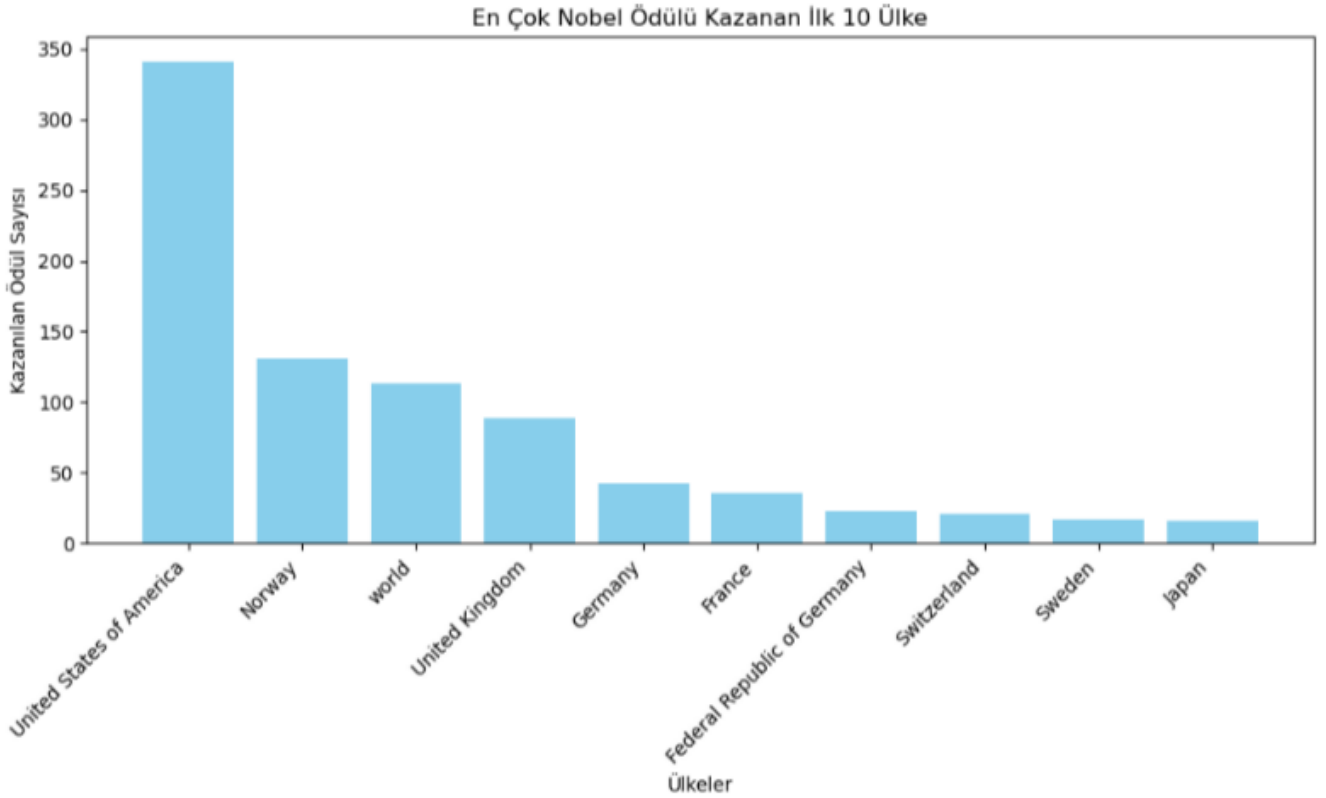
# Toplam ödül sayısına göre ilk 10 ülkeyi seçme
top_10_countries = top_countries.sort_values(by='prize', ascending=False).head(10)

print("En çok Nobel Ödülü kazanan ilk 10 ülke:")
print(top_10_countries)
```

```
In [197]: # Grafik için verileri hazırlama
top_10_countries = top_countries.sort_values(by='prize', ascending=False).head(10)

# Grafik oluşturma
plt.figure(figsize=(10, 6))
plt.bar(top_10_countries['organization_country'], top_10_countries['prize'], color='skyblue')
plt.xlabel('Ülkeler')
plt.ylabel('Kazanılan Ödül Sayısı')
plt.title('En çok Nobel Ödülü Kazanan İlk 10 Ülke')
plt.xticks(rotation=45, ha='right') # X eksen etiketlerini sağa döndürme
plt.tight_layout() # Grafik boyutunu ayarlama

# Grafiği gösterme
plt.show()
```



## 2) Nobel Ödüllerini kazanan ilk kadınları listeleyiniz.

```
In [198]: # Kategorilere göre gruplama yapalım
grouped_gender = df1.groupby('sex')
print(grouped_gender.size())
```

```
sex
Female    49
Male     836
dtype: int64
```

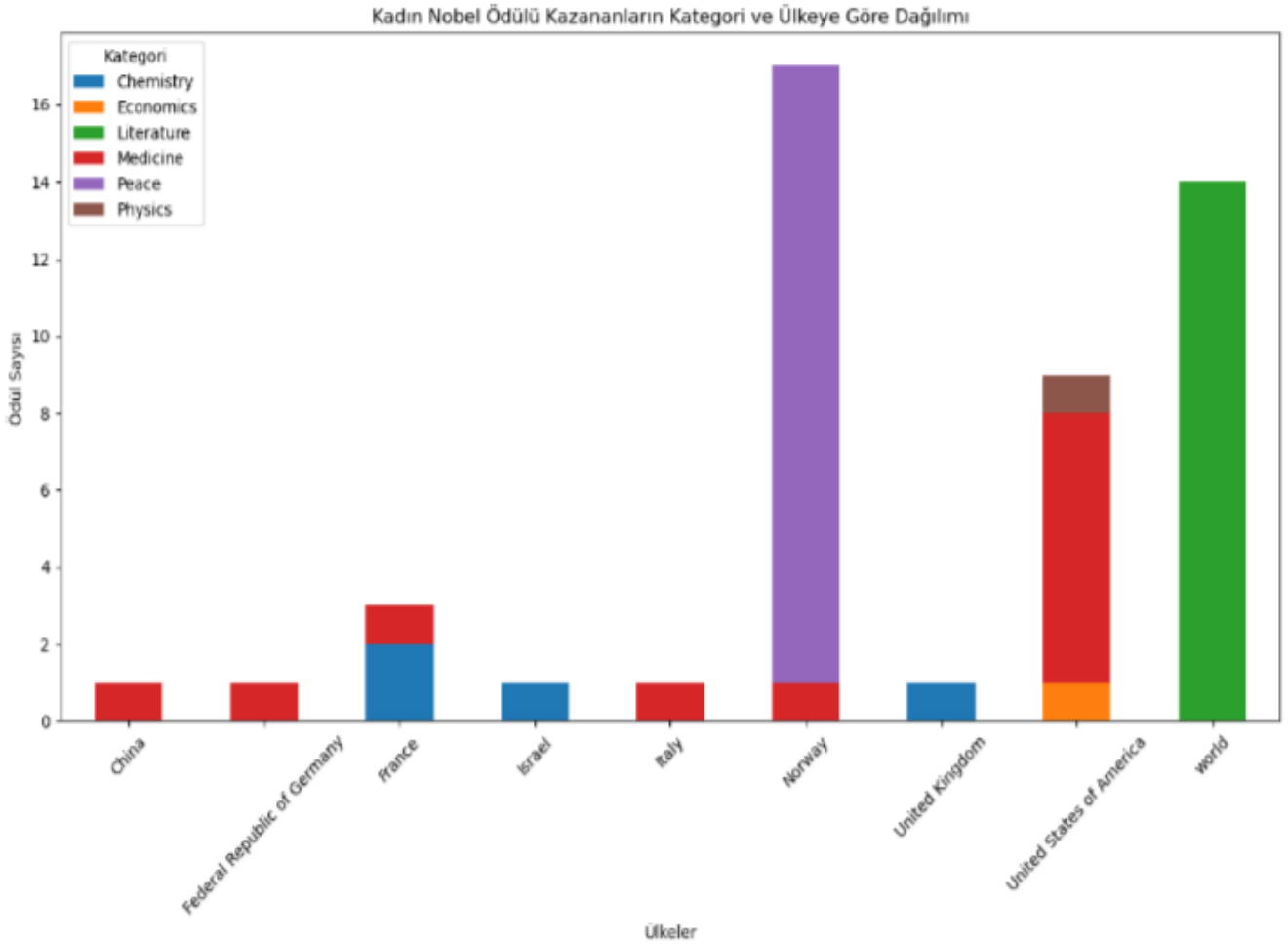
```
In [201]: # 'category' sütununa göre gruplama
grouped = df1.groupby('category')
```

```
# Her grupta 'sex' sütununda 'Female' olanları filtreleme ve ardından 'year' sütununa göre sıralama
filtered_and_sorted = grouped.apply(lambda x: x[x['sex'] == 'Female'].sort_values(by='year').head(3))
filtered_and_sorted
```

```
In [202]: # 'category' sütununa göre gruplama ve 'sex' sütununda 'Female' olanları filtreleme
grouped = df1[df1['sex'] == 'Female'].groupby(['category', 'organization_country']).size().reset_index(name='count')
```

```
# Pivot table oluşturma
pivot_table = grouped.pivot(index='organization_country', columns='category', values='count').fillna(0)
```

```
# Grafik oluşturma
pivot_table.plot(kind='bar', stacked=True, figsize=(12, 8))
plt.title('Kadın Nobel Ödülü Kazananların Kategori ve Ülkeye Göre Dağılımı')
plt.xlabel('Ülkeler')
plt.ylabel('Ödül Sayısı')
plt.legend(title='Kategori')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



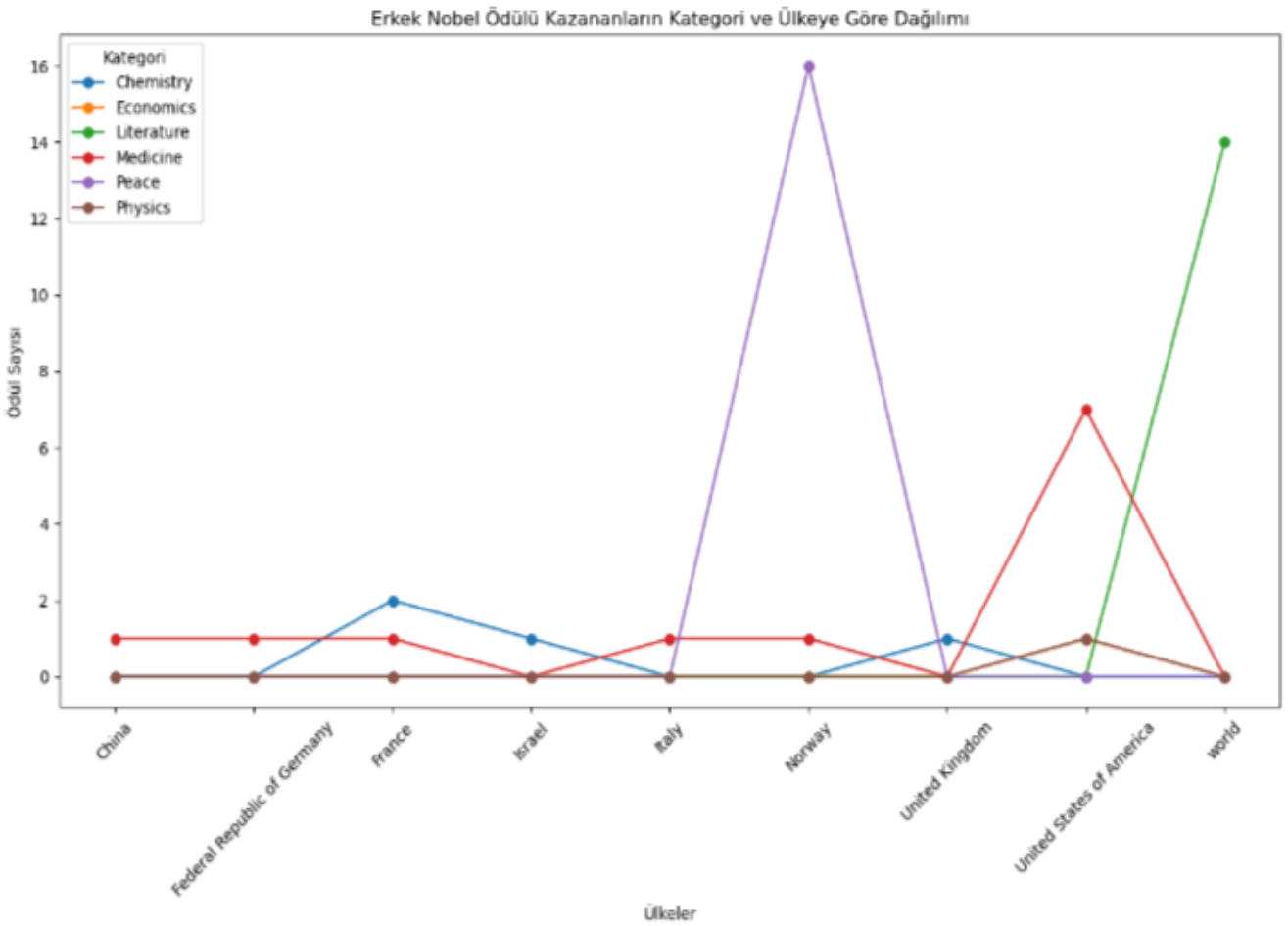
### 3) Nobel Ödüllerini kazanan ilk erkekleri listeleyiniz.

```
In [203]: # 'category' sütununa göre gruptama
grouped = df1.groupby('category')

# Her grupta 'sex' sütununda 'Male' olanları filtreleme ve ardından 'year' sütununa göre sıralama
filtered_and_sorted_m = grouped.apply(lambda x: x[x['sex'] == 'Male'].sort_values(by='year'))
filtered_and_sorted_m
```

```
In [205]: # Grafik oluşturma
plt.figure(figsize=(12, 8))
for column in pivot_table.columns:
    plt.plot(pivot_table.index, pivot_table[column], marker='o', label=column)

plt.title('Erkek Nobel Ödülü Kazananların Kategori ve Ülkeye Göre Dağılımı')
plt.xlabel('Ülkeler')
plt.ylabel('Ödül Sayısı')
plt.legend(title='Kategori')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



**NOT:** "Norway" sütunundaki "peace" kategorisinin yüksek çıkmasının sebebi null değerleri barış ödülü veren ülkeyi ve oraganizasyonun adıyla doldurmandan kaynaklanıyor.



#### 4) Nobel ödülünü en çok kazanan ülkenin hangi yıldan itibaren hakimiyet sağladığını görselleştirip bu hakimiyette rol oynayan şeyler nelerdir? İlgörülerinizi paylaşır mısınız?

```
In [206]: # 'organization_country' sütununa göre grupta  
grouped_o_c = df1.groupby('organization_country')  
  
# Her grupta 'year' sütununa göre sıralama ve ardından organizasyon ülkesi bilgilerini içeren DataFrame'leri birleştirme  
filtered_and_sorted_o_c = grouped_o_c.apply(lambda x: x.sort_values(by='year')).reset_index(drop=True)  
  
# Sonucu yazdırma  
result = filtered_and_sorted_o_c[['organization_country', 'year', 'prize']]  
result
```

```
In [207]: # 'organization_country' sütununa göre grupta  
grouped_o_c = df1.groupby('organization_country')  
  
# Her grupta 'year' sütununa göre sıralama ve ardından organizasyon ülkesi bilgilerini içeren DataFrame'leri birleştirme  
filtered_and_sorted_o_c = grouped_o_c.apply(lambda x: x.sort_values(by='year')).reset_index(drop=True)  
  
# Saçılım grafiği oluşturma  
plt.figure(figsize=(14, 8))  
sns.scatterplot(data=filtered_and_sorted_o_c, x='year', y='organization_country', hue='organization_country', palette='viridis',  
plt.title('Ülkelerin Yıllara Göre Nobel Ödülü Kazanma Dağılımı')  
plt.xlabel('Yıl')  
plt.ylabel('Ülkeler')  
plt.legend(title='Ülkeler', bbox_to_anchor=(1.05, 1), loc='upper left')  
plt.xticks(rotation=45)  
plt.tight_layout()  
plt.show()
```



1940 ve 1942 yılları arasında II. Dünya Savaşı nedeniyle Nobel Ödülleri verilememiştir. Grafikte bu yılların olduğu kısım boş olarak görünüyor. Bunun dışında Norveç için kullandığım "organization\_country" null değerleri doldurma kodundan dolayı Norveçte çok fazla nobel ödülü verisi görüntülenmiş. ancak bunların dışında özellikle USA'nın 1940'lı yıllardan sonra iyi bir çıkış yakaladığını söyleyebiliriz. Germany'nin ise 1940 tan önce diğer ülkelerden daha iyi durumda olduğu gözlemleniyor. United Kingdom'ın ise 1940-1980 yılları arasında diğer zamanlara göre oldukça iyi durumda olduğunu söyleyebiliriz. Germany'nin durumuna bakıldığında 2. Dünya savaşından dolayı böyle sonuç çıktığını söyleyebiliriz. USA için ise ABD Senatosu Versailles Barış Antlaşması'nı onaylamayarak bağımsız bir dış politika izlemeyi tercih etti. ABD 1920'lerde hızla gelişti. Ancak bu gelişme ABD tarihindeki en büyük ekonomik bunalım olan 1929 Dünya Ekonomik Bunalımına yol açtı. Bu ekonomik sebeplerin USA'nın nobel ödülü sayısını etkilediğini söyleyebiliriz. Nüfus, eğitimin kalitesi, bilimsel çalışmaların desteklenmesi gibi faktörlerin de nobel ödülü sayısını etkilediğini söyleyebiliriz.

**Not:** World olarak belirtilen veri organization\_country hücresi boş olan peace ve literature kategorileridir.

5) Nobel Ödülü kazananların cinsiyetlerini, yaşlarını, ödül kategorisi ve yılları kullanarak görselleştiriniz.(Her bir ödül kategorisi için ayrı grafik gösteriniz) Çıkan sonuçlara göre görseli yorumlayınız.

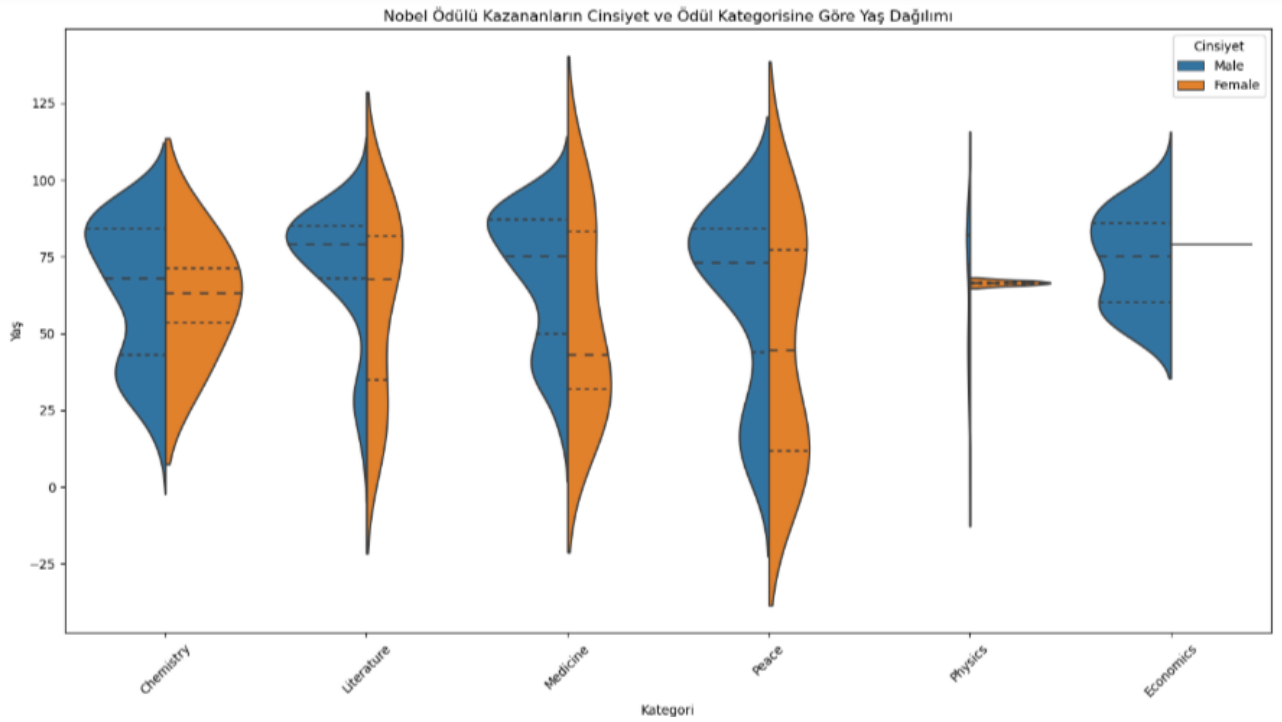
In [ ]:

```
In [208]: # Tarih sütunlarını datetime formatına çevirme
df1['birth_date'] = pd.to_datetime(df1['birth_date'], errors='coerce')
df1['death_date'] = pd.to_datetime(df1['death_date'], errors='coerce')

# Yaş hesaplama
df1['age'] = (df1['death_date'] - df1['birth_date']).dt.days // 365.25

# Geçerli yaş değerlerine sahip verileri filtreleme (yaş > 0)
df1_age_filtered = df1[(df1['age'] > 0)]

# Görselleştirme
plt.figure(figsize=(14, 8))
sns.violinplot(x='category', y='age', hue='sex', data=df1_age_filtered, split=True, inner='quart')
plt.title('Nobel Ödülü Kazananların Cinsiyet ve Ödül Kategorisine Göre Yaş Dağılımı')
plt.xlabel('Kategori')
plt.ylabel('Yaş')
plt.legend(title='Cinsiyet')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



60-90 yaş aralığında nobel ödülü alan erkeklerin daha fazla olduğunu, kadınlarda ise erkeklere göre nobel ödülü sayısı az olduğundan grafikte neredeyse her yaş grubundan katılımcı olduğu söylenebilir. bunun dışında kadınlarda kimya alanında 45-75 yaş aralığında diğer kategorilere göre daha fazla ödül alan olduğunu söyleyebiliriz. Sonuçların bu şekilde çıkmasındaki temel sebeplerden birisi kadınların eğitim-öğretime aktif olarak yeni yeni kabul ediliyor olmaları diyebiliriz. Uzun yıllar kadınlar, üniversite okusalar dahi çalışma hayatında da akedemide de kendilerine yer bulamamışlardır. Erkekler de ise her kategoride farklı yaş gruplarında ödül alan sayısının fazla olduğunu gözlemliyoruz. Özellikle 60-90 yaş grubu verileri Erkek katılımcıların akedemide kendilerine daha fazla yer bulduklarını gösteriyor. Bunların dışında 50 yaş grubu erkeklerin bütün kategorilerde azaldığı gözlemleniyor. Bunun sebebinin ise Dünya savaşları olduğunu söyleyebiliriz.

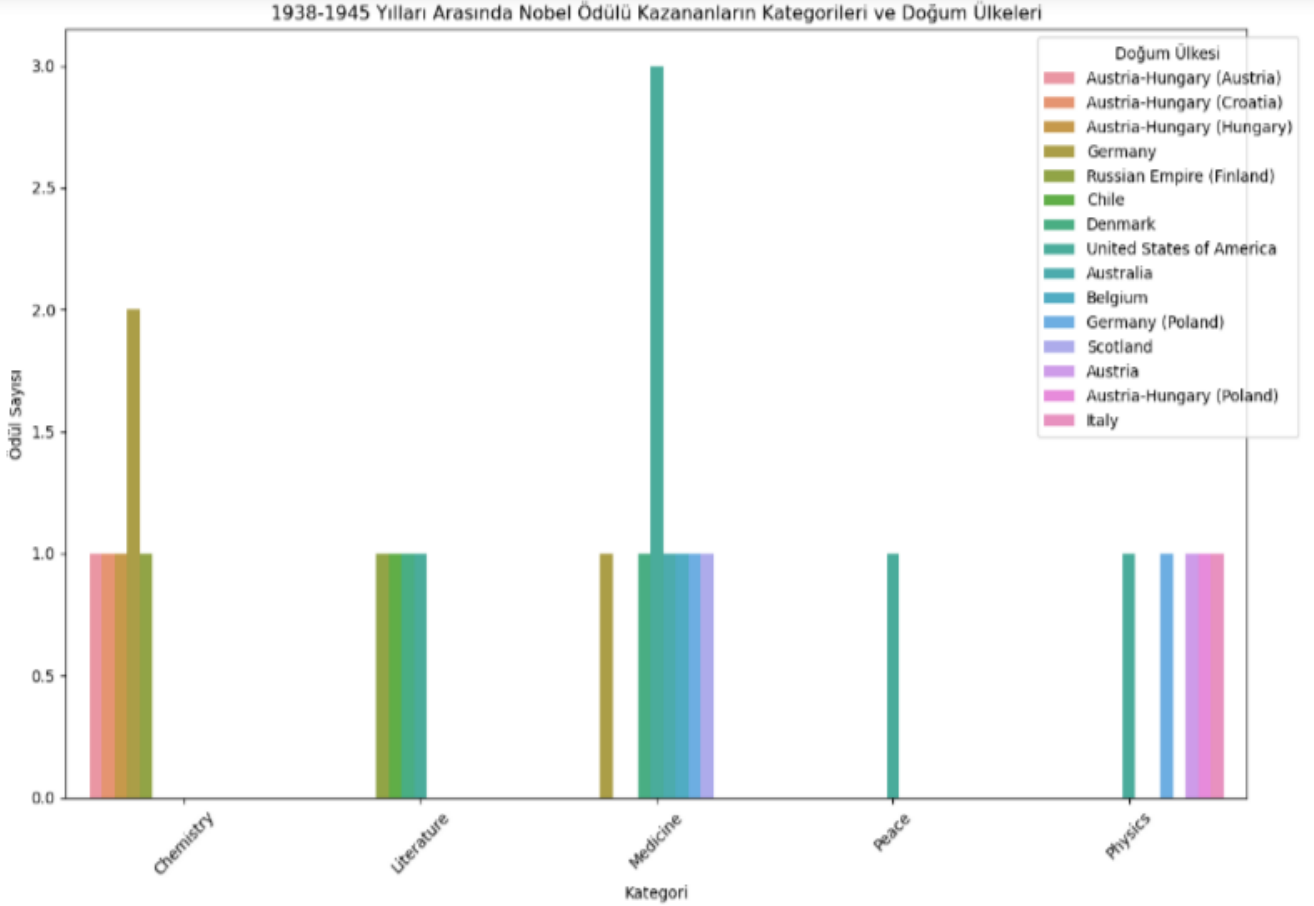


###### 6) 1938-1945 yılı arasında Nobel Ödülü kazananların kategorilerini ve ülkelerini görselleştirip yorumlayınız.

```
In [220]: # Yıl sütununu datetime olarak ayarlayın (eğer değilse)
df1['year'] = pd.to_datetime(df1['year'], format='%Y')
# 1938-1945 yılları arasında Nobel Ödülü kazananları filtreleyin
filtered_df = df1[(df1['year'].dt.year >= 1938) & (df1['year'].dt.year <= 1945)]

# Kategorileri ve doğum ülkelerini grupta
grouped = filtered_df.groupby(['category', 'birth_country']).size().reset_index(name='count')

# Grafik oluşturma
plt.figure(figsize=(12, 8))
sns.barplot(data=grouped, x='category', y='count', hue='birth_country')
plt.title('1938-1945 Yılları Arasında Nobel Ödülü Kazananların Kategorileri ve Doğum Ülkeleri')
plt.xlabel('Kategori')
plt.ylabel('Ödül Sayısı')
plt.xticks(rotation=45)
plt.legend(title='Doğum Ülkesi', bbox_to_anchor=(1.05, 1), loc='upper right')
plt.tight_layout()
plt.show()
```



Yukarıdaki grafikte sağ taraftaki ülkelerin renkleri yukarıdan aşağıya doğru kontrol edilerek bir renk sıralaması yapılmıştır. Yıl aralığı 2. Dünya Savaşı yılları olduğu için birth\_country ye göre sıraladım. çünkü özellikle Almanya gibi savaşta büyük yara almış ülkelerde insanlar başka yerlere göç ederek kendi ülkelerinde almaları gereken ödülü başka bir yerde almış olabilirler. Kimya, Edebiyat, Medicine, fizik kategorilerinde nobel ödülü sayılarının fazla olduğunu görüyoruz. Sayıların bu şekilde çıkmasında yine savaşın etkili olduğu düşüncesindeyim. Fizik nobel ödülleri incelendiğinde özellikle atom üzerinde yapılan çalışmalar silahlanma noktasında ülkeleri bir adım ileri taşıdığını söyleyebiliriz. Barış ödülünün tek olması da bu durumu özetliyor gibi.

7) 1947-1991 yılları arasında Nobel Ödülü kazananların kategorilerini ve ülkelerini görselleştirip yorumlayınız.(Her kategori için ayrı bir grafik olması istenmektedir)

```
In [222]: # Yıl sütununu datetime olarak ayarlayın (eğer değilse)
df1['year'] = pd.to_datetime(df1['year'], format='%Y')

# 1947-1991 yılları arasında Nobel Ödülü kazananları filtreleyin
filtered_df = df1[(df1['year'].dt.year >= 1947) & (df1['year'].dt.year <= 1991)]

# Kategorilere göre veri çerçeveleri oluşturun
categories = filtered_df['category'].unique()

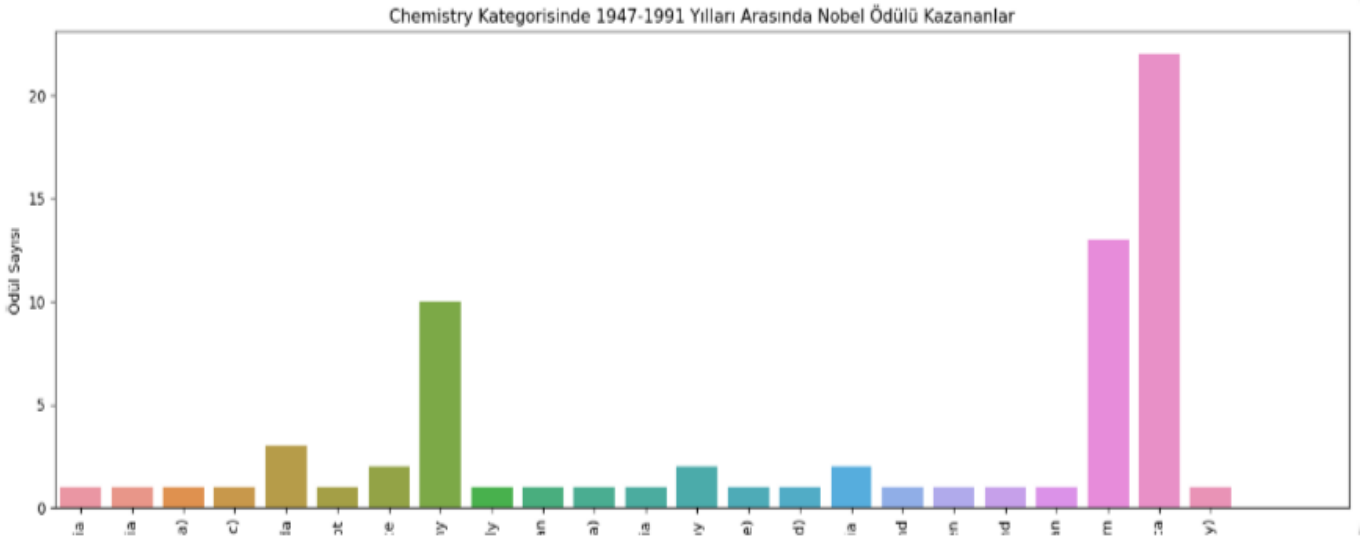
# Her kategori için ayrı grafikler oluşturma
for category in categories:
    category_df = filtered_df[filtered_df['category'] == category]
    grouped = category_df.groupby(['category', 'birth_country']).size().reset_index(name='count')

    plt.figure(figsize=(14, 8))
    sns.barplot(data=grouped, x='birth_country', y='count', dodge=True)

    # Sütunlar arasında boşluk eklemek için xlim kullanımı
    country_count = len(grouped['birth_country'].unique())
    plt.xlim(-0.5, country_count - 0.5 + (country_count - 1) * 0.1) # Her sütun arasına %10 boşluk ekle

    plt.title(f'{category} Kategorisinde 1947-1991 Yılları Arasında Nobel Ödülü Kazananlar')
    plt.xlabel('Doğum Ülkesi')
    plt.ylabel('Ödül Sayısı')
    plt.xticks(rotation=90)
    plt.tight_layout()
    plt.show()
```

Örnek Grafik



En fazla nobel ödülü alan ülkelerin genellikle aynı olduğu gözlemleniyor. Bunun sebebi ise öncelikle eğitime verilen önem, ekonomi, dünya savaşları, nüfus yoğunluğu olarak yorumlanabilir. Bu yıllar arasında toplumlar sosyal olarak da büyük değişimler geçirmişler. Dolayısıyla bu değişime uyum sağlayabilen ülkeler diğer alanlarda da gelişim gösteriyorlar. Özellikle Amerika'nın kültürel anlamdaki çeşitliliğinin çatışmayı tetiklemesi beklenirken savaşlardan neredeyse hiç etkilenmiyor. Bu durum da Amerika'nın her anlamda grafiklere göre lider olmasında etkili olmuştur diyebiliriz.

8)Kimya, Edebiyat, Barış, Fizik ve Tıp kategorilerindeki 2000 sonrasındaki kişilerin ülkelerini, yaşlarını görselleştirin.(Her bir Kategori için ayrı görselleştirme yapılması istenmektedir) Veriyi yorumlayınız.

```
In [245]: # Kategorilere göre gruplama yapalım
grouped_category = df1.groupby('category')
print(grouped_category.size())

category
Chemistry      175
Economics       78
Literature     113
Medicine       211
Peace          130
Physics        204
dtype: int64

In [249]: # Yıl sütununu datetime olarak ayarlayın (eğer değilse)
df1['year'] = pd.to_datetime(df1['year'], format='%Y')

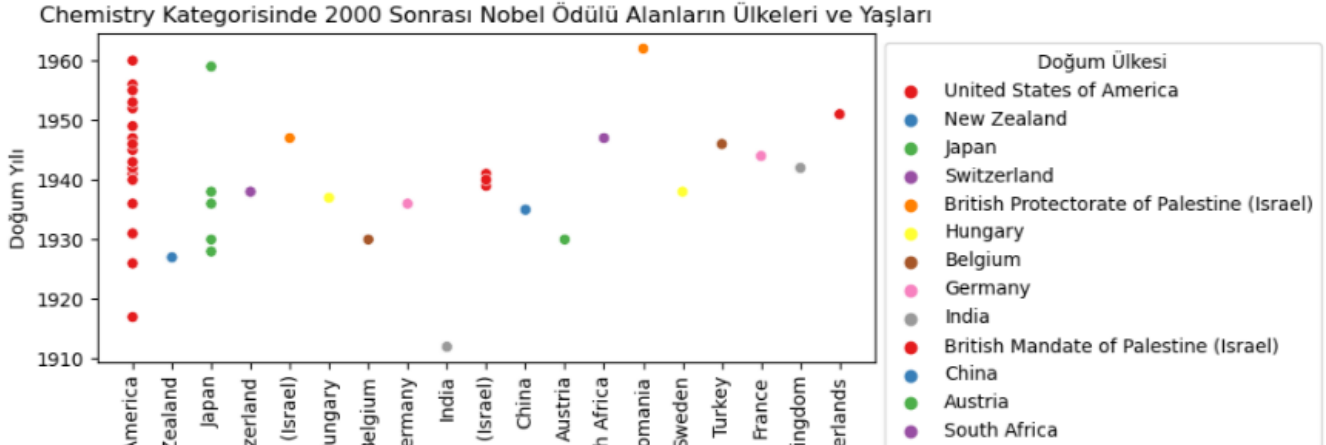
# Belirtilen kategorilerdeki ve 2000 sonrasındaki kişileri filtreleme
categories = ['Chemistry', 'Literature', 'Peace', 'Physics', 'Medicine']
filtered_df = df1[(df1['category'].isin(categories)) & (df1['year'].dt.year >= 2000)]

# Her kategori için ayrı görselleştirme yapma
for category in categories:
    category_df = filtered_df[filtered_df['category'] == category]

    plt.figure(figsize=(10, 6))
    sns.scatterplot(data=category_df, x='birth_country', y='birth_year', hue='birth_country', legend='full', palette='Set1')

    plt.title(f'{category} Kategorisinde 2000 Sonrası Nobel Ödülü Alanların Ülkeleri ve Yaşları')
    plt.xlabel('Doğum Ülkesi')
    plt.ylabel('Doğum Yılı')
    plt.xticks(rotation=90)
    plt.legend(title='Doğum Ülkesi', bbox_to_anchor=(1, 1), loc='upper left')
    plt.tight_layout()
    plt.show()
```

Örnek Grafik



Fizik kategorisinde diğerlerinden farklı olarak Japonya'yı da ön planda görüyoruz. Bunun sebebi ise 1945 yılında Hiroşima'ya atılan atom bombasının etkilerinin minimuma inmiş olması olabilir. Böyle bir olay bütün ülkeyi her yönden etkileyeceği için bir çöküş döneminden sonra oldukça iyi bir ivme kazandığını söyleyebiliriz.