

```
In [98]: import pandas as pd
import random, string
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme(style="whitegrid")
sns.set(font_scale=1.8)
```

```
In [99]: Columns = ['Full Name', 'Number', 'Email', 'Country', 'Province', 'City', 'District', 'Year', 'Quarter',
'Month', 'Week of Year', 'Day', 'hour', 'Age', 'Gender', 'Language', 'IsNew', 'TimesInWeek',
'TimesInMonth', 'Pages/Session', 'AvgTime/Page', 'Device', 'TrafficSource', 'Browser', 'OS',
'Payment Method', 'ShipVia', 'Discount %', 'NumberOfItems', 'TotalPrice']
```

```
In [101]: Browser = ['Google Chrome', 'Mozilla Firefox', 'Opera', 'Safari', 'Internet Explorer', 'Slimjet', 'Maxthon', 'SlimBrowser',
'Netscape', 'UC Browser', ]
PaymentMethod = ['Credit/Debit card', 'Prepaid card payments', 'Bank transfers', 'E-Wallets', 'Cash', 'Mobile payments',
'Cryptocurrencies', 'Cash on delivery']
OS = ["Android", "Windows", "iOS", "Macintosh", "Linux", "Amazon Fire OS", "SmartTV", "Chrome OS"]
Device = ['Mobile', 'Computer', 'Laptop', 'Tablet', 'TV']
Source = ["Facebook", "Google", "Gmail", "WhatsApp", "Yahoo", "Youtube", "Instagram", "Bing"]
Country = ['USA', 'Russia', 'Pakistan', 'India', 'Afghanistan', 'China', 'Canada', 'UK', 'Germany', 'Bangladesh', 'Sudan',
'Australia', 'Turkey', 'France', 'Spain', 'Italy', 'Poland']
Province = ['Sindh', 'KPK', 'Balochistan', 'Punjab', 'AB', 'AC', 'AD', 'AE', 'AF', 'AG', 'AH', 'AK', 'AI', 'AL', 'AM']
District = ['Badin', 'Dadu', 'Ghotki', 'Hyderabad', 'Jacobabad', 'Jamshoro', 'Karachi Central', 'Karachi East', 'Karachi South',
'Karachi West', 'Kashmore', 'Khairpur', 'Korangi', 'Larkana', 'Malir', 'Matiani', 'Mirpur Khas', 'Naushahro Feroze',
'Qambar Shahdadt', 'Sanghar', 'Shaheed Benazir Abad', 'Shikarpur', 'Sujawal', 'Sukkur', 'Tando Allahyar',
'Tando Muhammad Khan', 'Tharparkar', 'Thatta', 'Umerkot[20]', 'Attock', 'Bahawalnagar', 'Bahawalpur', 'Bhakar',
'Chakwal', 'Chiniot', 'Dera Ghazi Khan', 'Faisalabad', 'Gujranwala', 'Gujrat', 'Hafizabad', 'Jhang', 'Jhelum',
'Kasur', 'Khanewal', 'Khushab', 'Lahore', 'Layyah', 'Lodhran', 'Mandi Bahauddin', 'Mianwali', 'Multan',
'Muzaffargarh', 'Narowal', 'Nankana Sahib[5]', 'Okara', 'Pakpattan', 'Rahim Yar Khan', 'Rajapur', 'Rawalpindi',
'Sahiwal', 'Sargodha', 'Sheikhupura', 'Sialkot', 'Toba Tek Singh', 'Vehari', 'Abbottabad', 'Bajaur', 'Bannu',
'Battagram', 'Buner', 'Charsadda', 'Chitral', 'Dera Ismail Khan', 'Hangu', 'Haripur', 'Karak', 'Khyber', 'Kohat',
'Kurram', 'Lakki Marwat', 'Lower Dir', 'Lower Kohistan', 'Malakand', 'Mansehra', 'Mardan', 'Mohmand',
'North Waziristan', 'Nowshera', 'Orakzai', 'Peshawar', 'Shangla', 'South Waziristan', 'Swabi', 'Swat', 'Tank',
'Torghar', 'Upper Dir', 'Upper Kohistan', 'Awaran', 'Barkhan', 'Chagai', 'Dera Bugti', 'Gwadar', 'Harnai[17]',
'Jafarabad', 'Jhal Magsi', 'Kachhi', 'Kalat', 'Kech', 'Kharan', 'Khuzdar', 'Killa Abdullah', 'Killa Saifullah',
'Kohlu', 'Lasbela', 'Lehri', 'Loralai', 'Mastung', 'Musahel', 'Nasirabad', 'Nushki[18]', 'Panjgur', 'Pishin',
'Quetta', 'Sherani', 'Sibi', 'Sohbatpur', 'Washuk', 'Zhob', 'Ziarat', 'Duki']
City = ['Mumbai', 'Delhi', 'Bangalore', 'Hyderabad', 'Ahmedabad', 'Chennai', 'Kolkata', 'Surat', 'Pune', 'Jaipur', 'Lucknow',
'Kanpur', 'Nagpur', 'Indore', 'Thane', 'Bhopal', 'Visakhapatnam[4]', 'Pimpri-Chinchwad', 'Patna', 'Vadodara',
'Ghaziabad', 'Ludhiana', 'Agra', 'Nashik', 'Ranchi', 'Faridabad', 'Meerut', 'Rajkot', 'Kalyan-Dombivli', 'Vasai-Virar',
'Varanasi', 'Srinagar', 'Aurangabad', 'Dhanbad', 'Amritsar', 'Navi Mumbai', 'Allahabad', 'Howrah', 'Gwalior',
'Jabalpur', 'Coimbatore', 'Vijayawada', 'Jodhpur', 'Madurai', 'Raipur', 'Kota[6]', 'Chandigarh', 'Guwahati', 'Solapur',
'Hubli-Dharwad', 'Karachi', 'Lahore', 'Faisalabad', 'Rawalpindi', 'Gujranwala', 'Peshawar', 'Multan', 'Hyderabad',
'Islamabad', 'Quetta', 'Bahawalpur', 'Sargodha', 'Sialkot', 'Sukkur', 'Larkana', 'Sheikhupura', 'Rahim Yar Khan',
'Jhang', 'Dera Ghazi Khan', 'Gujrat', 'Sahiwal', 'Wah Cantonment', 'Mardan', 'Kasur', 'Okara', 'Mingora', 'Nawabshah',
'Chiniot', 'Kotri', 'Kamoke', 'Hafizabad', 'Sadiqabad', 'Mirpur Khas', 'Burewala', 'Kohat', 'Khanewal',
'Dera Ismail Khan', 'Turbat', 'Muzaffargarh', 'Abbottabad', 'Mandi Bahauddin', 'Shikarpur', 'Jacobabad', 'Jhelum',
'Khanpur', 'Khairpur', 'Khuzdar', 'Pakpattan', 'Hub', 'Daska', 'Gojra', 'Dadu', 'Muridke', 'Bahawalnagar', 'Samundri',
'Tando Allahyar', 'Tando Adam', 'Jaranwala', 'Chishtian', 'Muzaffarabad', 'Attock', 'Vehari', 'Kot Abdul Malik',
'Ferozwala', 'Chakwal', 'Gujranwala Cantonment', 'Kamalia', 'Umerkot', 'Ahmedpur East', 'Kot Addu', 'Wazirabad',
'Mansehra', 'Layyah', 'Mirpur', 'Swabi', 'Chaman', 'Taxila', 'Nowshera', 'Khushab', 'Shahdadt', 'Mianwali', 'Kabal',
'Lodhran', 'Hasilpur', 'Charsadda', 'Bhakar', 'Badin', 'Arif Wala', 'Ghotki', 'Sambrial', 'Jatoi', 'Haroonabad',
'Daharki', 'Narowal', 'Tando Muhammad Khan', 'Kamber Ali Khan', 'Mirpur Mathelo', 'Kandhkot', 'Bhalwal',
'New York City[d]', 'Los Angeles', 'Chicago', 'Houston[3]', 'Phoenix', 'Philadelphia[e]', 'San Antonio', 'San Diego',
'Dallas', 'San Jose', 'Austin', 'Jacksonville[f]', 'Fort Worth', 'Columbus', 'Charlotte', 'San Francisco[g]',
'Indianapolis[h]', 'Seattle', 'Denver[i]', 'Washington[j]', 'Boston', 'El Paso', 'Nashville[k]', 'Detroit',
'Oklahoma City', 'Portland', 'Las Vegas', 'Memphis', 'Louisville[l]', 'Baltimore[m]', 'Milwaukee', 'Albuquerque',
'Tucson', 'Fresno', 'Mesa', 'Sacramento', 'Atlanta', 'Kansas City', 'Colorado Springs', 'Omaha', 'Raleigh', 'Miami',
'Long Beach', 'Virginia Beach[m]', 'Oakland', 'Minneapolis', 'Tulsa', 'Tampa', 'Arlington', 'New Orleans']
Gender = ['Male', 'Female']
Language = ['Mandarin Chinese', 'Spanish', 'English', 'Hindi (sanskritised Hindustani)[9]', 'Bengali', 'Portuguese', 'Russian',
'Japanese', 'Western Punjabi[10]', 'Marathi', 'Telugu', 'Wu Chinese', 'Turkish', 'Korean', 'French', 'German',
'Vietnamese', 'Tamil', 'Yue Chinese', 'Urdu (persianised Hindustani)[9]', 'Javanese', 'Italian', 'Egyptian Arabic',
'Gujarati', 'Iranian Persian', 'Bhojpuri', 'Min Nan Chinese', 'Hakka Chinese', 'Jin Chinese', 'Hausa', 'Kannada',
'Indonesian (Indonesian Malay)', 'Polish', 'Yoruba', 'Xiang Chinese', 'Malayalam', 'Odia', 'Maithili', 'Burmese',
'Eastern Punjabi[10]', 'Sunda', 'Sudanese Arabic', 'Algerian Arabic', 'Moroccan Arabic', 'Ukrainian', 'Igbo',
'Northern Uzbek', 'Sindhi', 'North Levantine Arabic', 'Romanian', 'Tagalog', 'Dutch', 'Sa'idi Arabic',
'Gan Chinese', 'Amharic', 'Northern Pashto', 'Magahi', 'Thai', 'Saraiki', 'Khmer', 'Chhattisgarhi', 'Somali',
'Malaysian (Malaysian Malay)', 'Cebuano', 'Nepali', 'Mesopotamian Arabic', 'Assamese', 'Sinhalese',
'Northern Kurdish', 'Hejazi Arabic', 'Nigerian Fulfulde', 'Bavarian', 'South Azerbaijani', 'Greek', 'Chittagonian',
'Kazakh', 'Deccan', 'Hungarian', 'Kinyarwanda', 'Zulu', 'South Levantine Arabic', 'Tunisian Arabic',
'Sanaani Spoken Arabic', 'Min Bei Chinese', 'Southern Pashto', 'Rundi', 'Czech', 'Ta'izzi-Adeni Arabic', 'Uyghur',
'Min Dong Chinese', 'Sylheti']
ShipVia = ['2-day shipping', 'Same-day delivery', 'Overnight shipping', 'Expedited shipping', 'International shipping',
'Freight shipping', 'Calculating shipping costs', 'Dimensional weight']
Day = ['Mon', 'Tue', 'Wed', 'Thur', 'Fri', 'Sat', 'Sun']
```

```
In [102]: dictionary = dict()
```

```
In [103]: N = 1000000
dictionary['Full Name'] = ['User'+str(i) for i in range(N)]
dictionary['Number'] = [random.randint(1000000000, 9999999999) for i in range(N)]
dictionary['Email'] = ['.'.join(random.sample('abcdefghijklmnopqrstuvwxyz', 5))+ '@gmail.com' for i in range(N)]
dictionary['Country'] = random.choices(Country, k = N)
dictionary['Province'] = random.choices(Province, k = N)
dictionary['District'] = random.choices(District, k = N)
dictionary['City'] = random.choices(City, k = N)
dictionary['Year'] = [random.randint(2000, 2020) for i in range(N)]
dictionary['Quarter'] = [random.randint(1, 4) for i in range(N)]
dictionary['Month'] = [random.randint(1, 12) for i in range(N)]
dictionary['WeekOfYear'] = [random.randint(1, 52) for i in range(N)]
dictionary['Day'] = random.choices(Day, k = N)
dictionary['Hour'] = [random.randint(1, 24) for i in range(N)]
dictionary['Age'] = [random.randint(15, 60) for i in range(N)]
dictionary['Gender'] = random.choices(Gender, k = N)
dictionary['Language'] = random.choices(Language, k = N)
dictionary['IsNew'] = random.choices([0,1], k = N)
dictionary['TimesInWeek'] = [random.randint(1, 10) for i in range(N)]
dictionary['TimesInMonth'] = [random.randint(1, 25) for i in range(N)]
dictionary['Pages/Session'] = [random.randint(1, 15) for i in range(N)]
dictionary['AvgTime/Session'] = [random.randint(1, 5) for i in range(N)]
dictionary['Device'] = random.choices(Device, k = N)
dictionary['TrafficSource'] = random.choices(Source, k = N)
dictionary['Browser'] = random.choices(Browser, k = N)
dictionary['OS'] = random.choices(OS, k = N)
dictionary['PaymentMethod'] = random.choices(PaymentMethod, k = N)
dictionary['ShippedVia'] = random.choices(ShipVia, k = N)
dictionary['Discount %'] = [random.randint(1, 10) for i in range(N)]
dictionary['NoOfItems'] = [random.randint(1, 20) for i in range(N)]
dictionary['TotalPrice'] = [random.randint(1000, 20000) for i in range(N)]
f = lambda x : 1 if x in ['Sat', 'Sun'] else 0
dictionary['Weekend'] = [f(i) for i in range(N)]
```

```
In [104]: df = pd.DataFrame(dictionary)
```

```
In [105]: df.to_csv('DW_Final_Project.csv', index = False)
```

Average Total Sale Per Country

```
In [108]: X = df.groupby('Country')['TotalPrice'].mean().index
Y = df.groupby('Country')['TotalPrice'].mean().values

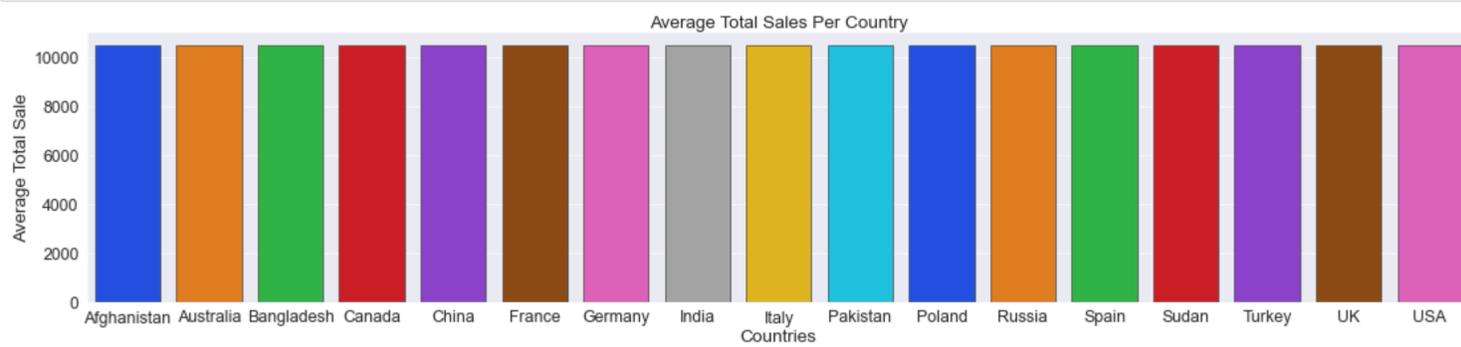
plt.figure(figsize = (30, 6), dpi = 45)

plot = sns.barplot(x = X, y = Y, palette="bright", linewidth=1, edgecolor=".2")

plt.xlabel('Countries')
plt.ylabel('Average Total Sale')

plt.title('Average Total Sales Per Country')

plt.show()
figure = plot.get_figure()
figure.savefig('Average Total Sale Per Country.jpeg')
```



Daywise sales distribution

```
In [109]: X = df.groupby('Day')['TotalPrice'].sum().index
Y = df.groupby('Day')['TotalPrice'].sum().values

plt.figure(figsize = (16, 6), dpi = 70)

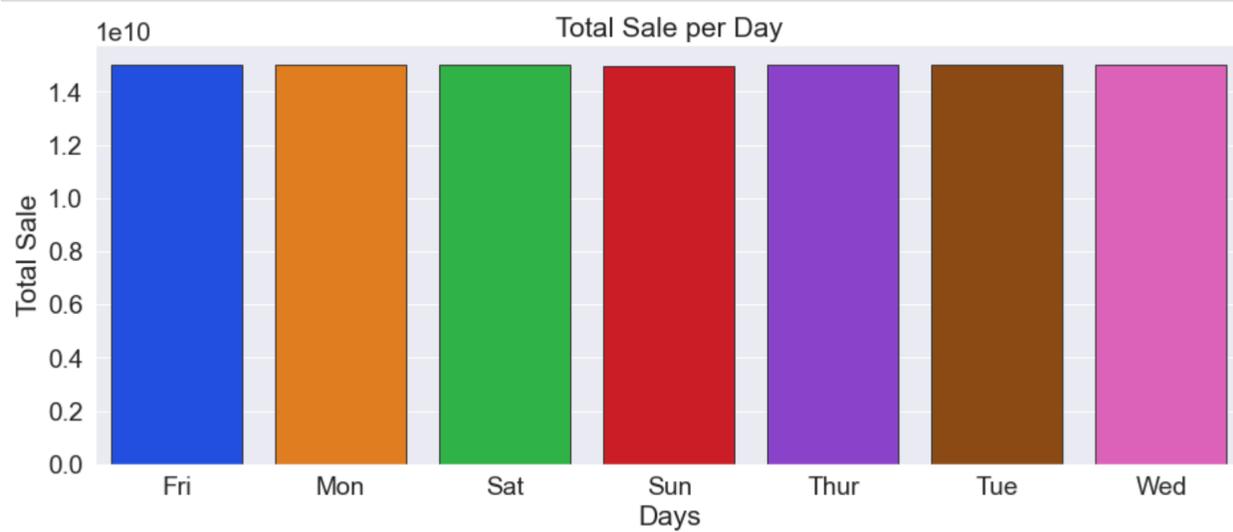
sns.barplot(x = X, y = Y, palette="bright", linewidth=1, edgecolor=".2")

plt.xlabel('Days')
plt.ylabel('Total Sale')

plt.title('Total Sale per Day')

plt.show()

figure = plot.get_figure()
figure.savefig('Daywise sales distribution.jpeg')
```



New Customers vs Returning customers

```
In [110]: X = df['IsNew'].value_counts().index
Y = df['IsNew'].value_counts().values

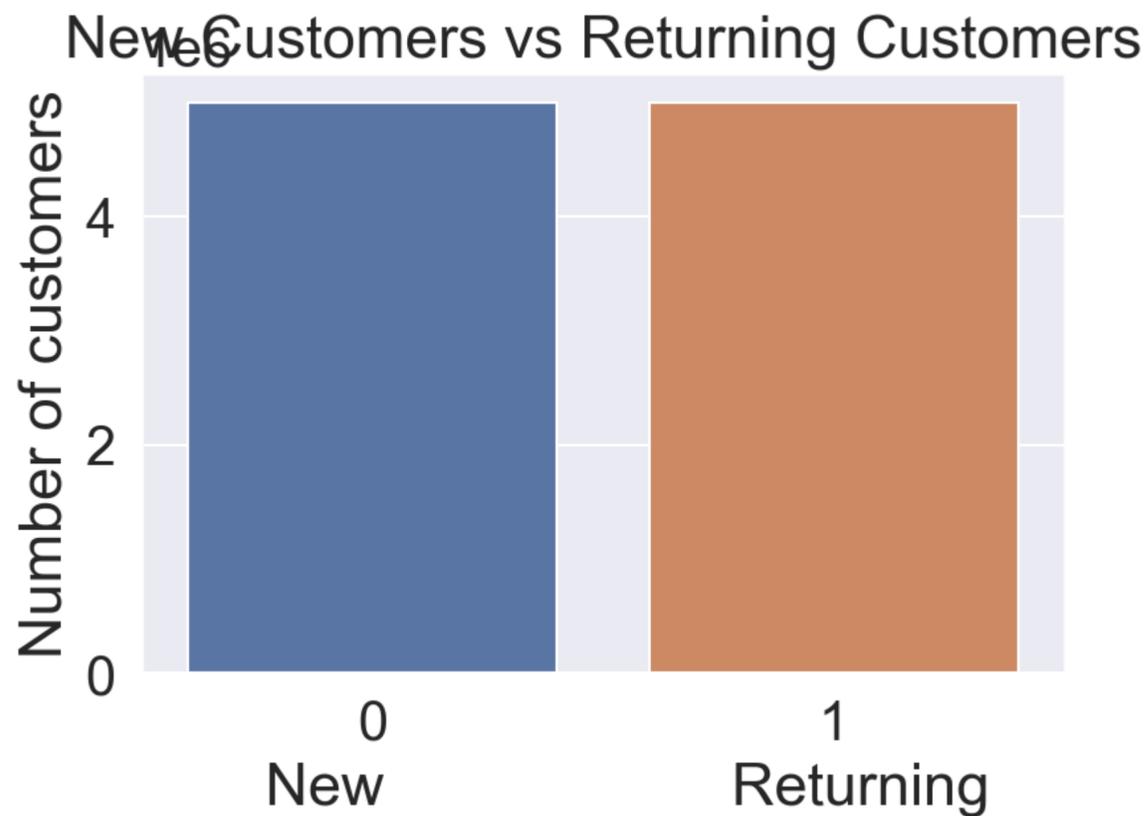
plt.figure(dpi = 150)

sns.barplot(x = X, y = Y)

plt.xlabel(' New          Returning')
plt.ylabel('Number of customers')

plt.title('New Customers vs Returning Customers')
plt.show()

figure = plot.get_figure()
figure.savefig('New Customers vs Returning customers.jpeg')
```



Average Sale per Year

```
In [111]: X = df.groupby('Year')['TotalPrice'].mean().index
Y = df.groupby('Year')['TotalPrice'].mean().values

plt.figure(figsize = (20, 6), dpi = 60)

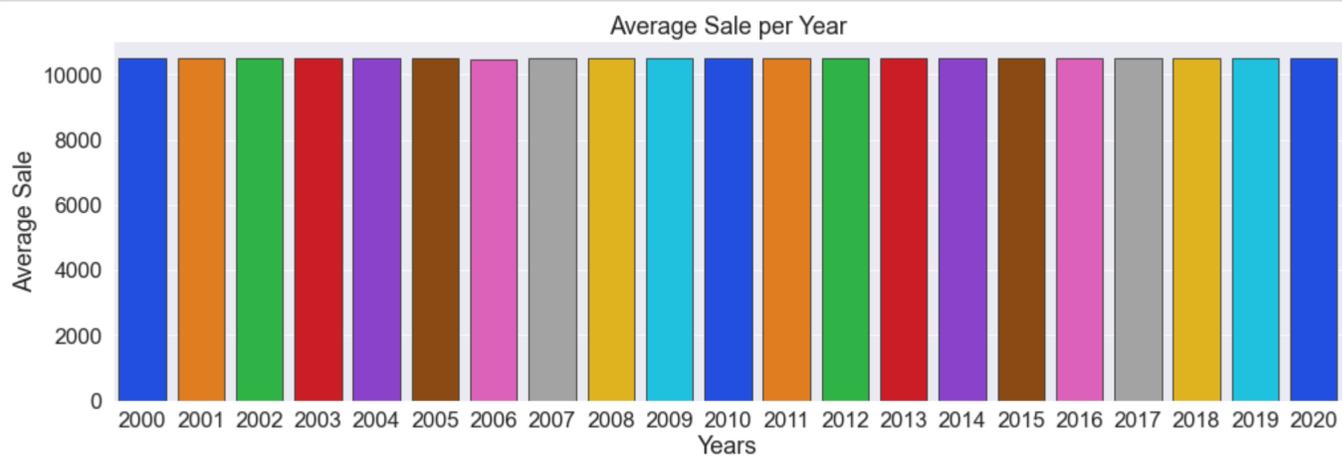
sns.barplot(x = X, y = Y, palette="bright", linewidth=1, edgecolor=".2")

plt.xlabel('Years')
plt.ylabel('Average Sale')

plt.title('Average Sale per Year')

plt.show()

figure = plot.get_figure()
figure.savefig('Average Sale per Year.jpeg')
```



Average Number of Items sold per Month

```
In [112]: X = df.groupby('Month')['TotalPrice'].mean().index
Y = df.groupby('Month')['TotalPrice'].mean().values

plt.figure(figsize = (16, 6), dpi = 70)

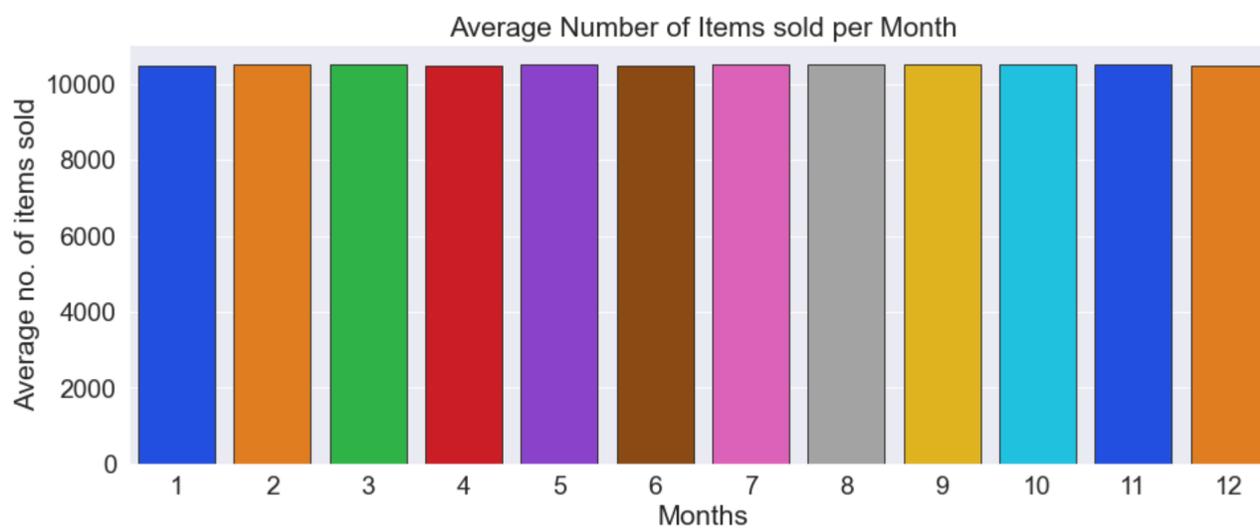
sns.barplot(x = X, y = Y, palette="bright", linewidth=1, edgecolor=".2")

plt.xlabel('Months')
plt.ylabel('Average no. of items sold')

plt.title('Average Number of Items sold per Month')

plt.show()

figure = plot.get_figure()
figure.savefig('Average Number of Items sold per Month.jpeg')
```



Average Time spent per Session by Operating systems

```
In [113]: X = df.groupby('OS')['AvgTime/Session'].mean().index
Y = df.groupby('OS')['AvgTime/Session'].mean().values

plt.figure(figsize = (16, 6), dpi = 70)

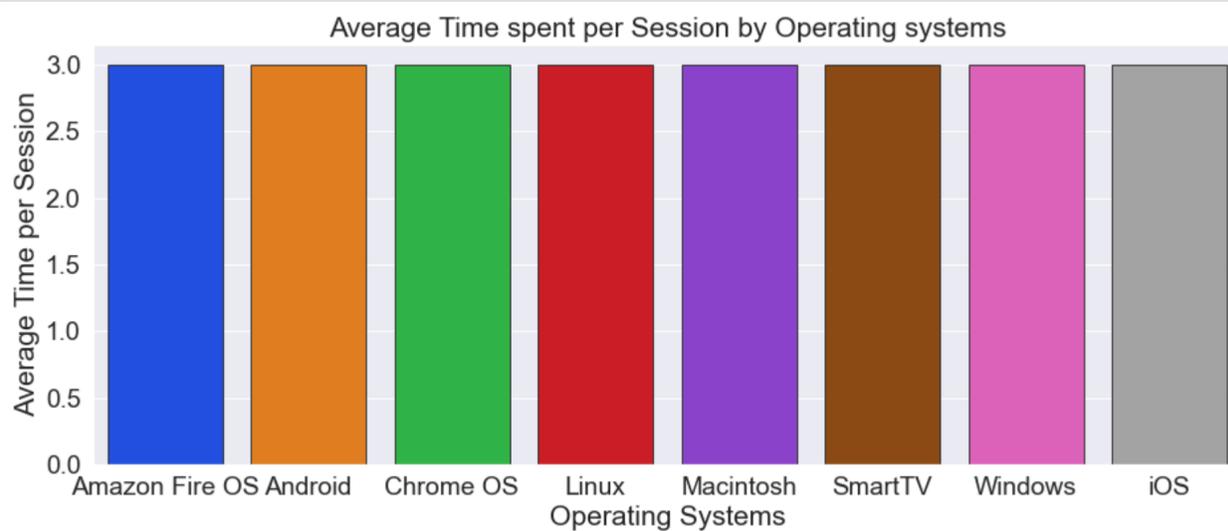
sns.barplot(x = X, y = Y, palette="bright", linewidth=1, edgecolor=".2")

plt.xlabel('Operating Systems')
plt.ylabel('Average Time per Session')

plt.title('Average Time spent per Session by Operating systems')

plt.show()

figure = plot.get_figure()
figure.savefig('Average Time spent per Session by Operating systems.jpeg')
```



Average Pages viewed per Session by browsers

```
In [114]: X = df.groupby('Browser')['Pages/Session'].mean().index
Y = df.groupby('Browser')['Pages/Session'].mean().values

plt.figure(figsize = (35, 6), dpi = 35)

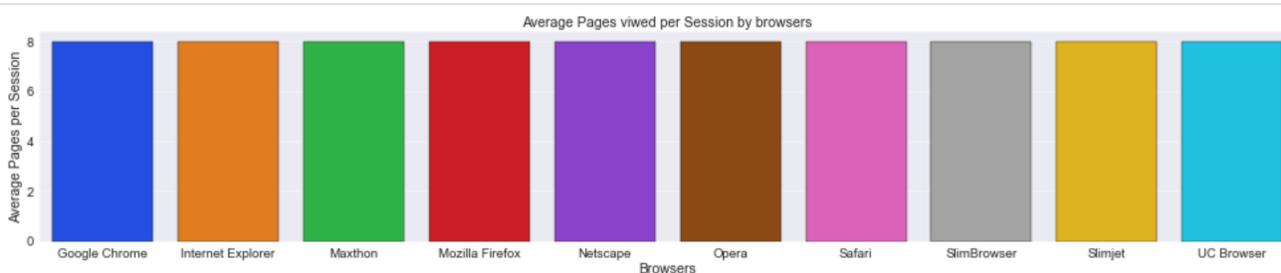
sns.barplot(x = X, y = Y, palette="bright", linewidth=1, edgecolor=".2")

plt.xlabel('Browsers')
plt.ylabel('Average Pages per Session')

plt.title('Average Pages viewed per Session by browsers')

plt.show()

figure = plot.get_figure()
figure.savefig('Average Pages viewed per Session by browsers.jpeg')
```



Average number of website visits by the user per device

```
In [115]: X = df.groupby('Device')['TimesInWeek'].mean().index
Y = df.groupby('Device')['TimesInWeek'].mean().values

plt.figure(figsize = (18, 6), dpi = 60)

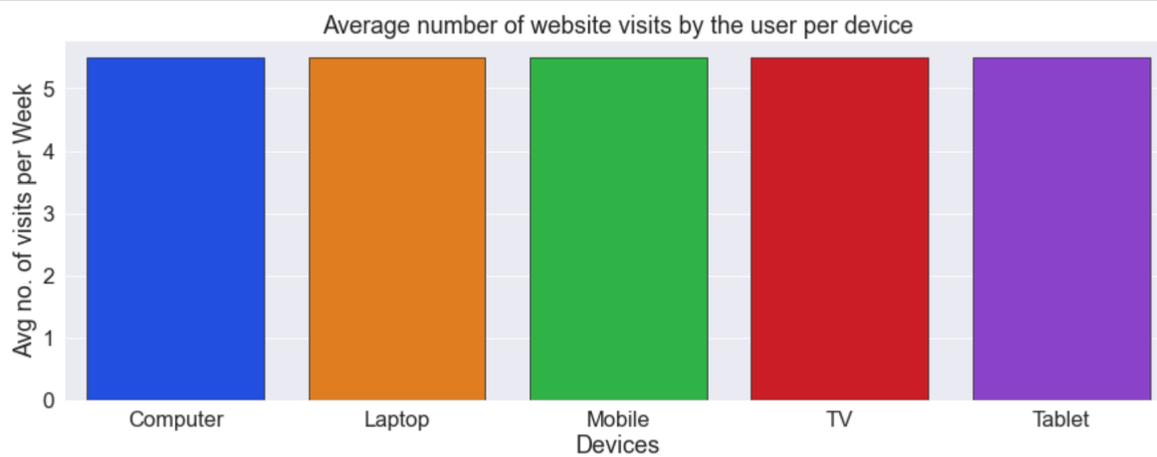
sns.barplot(x = X, y = Y, palette="bright", linewidth=1, edgecolor=".2")

plt.xlabel('Devices')
plt.ylabel('Avg no. of visits per Week')

plt.title('Average number of website visits by the user per device')

plt.show()

figure = plot.get_figure()
figure.savefig('Average number of website visits by the user per device.jpeg')
```



Average number of website visits by the user per Month per device

```
In [116]: X = df.groupby('TrafficSource')['TimesInMonth'].mean().index
Y = df.groupby('TrafficSource')['TimesInMonth'].mean().values

plt.figure(figsize = (27, 6), dpi = 45)

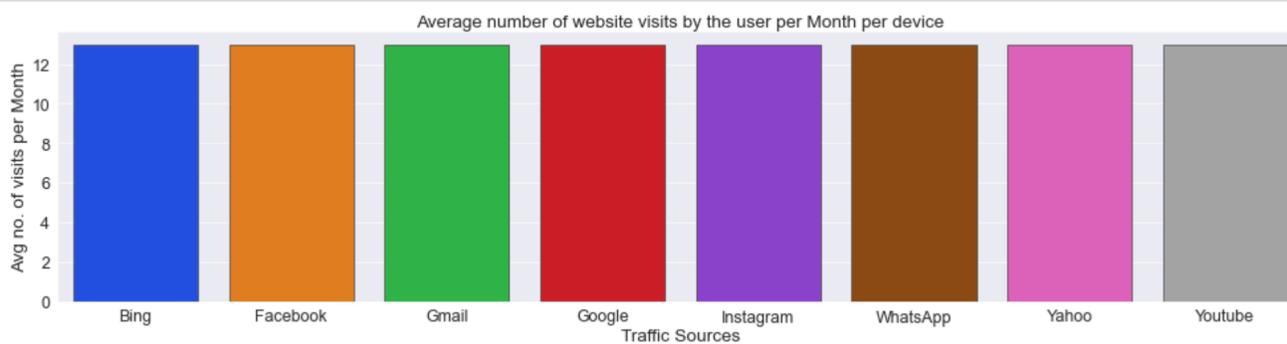
sns.barplot(x = X, y = Y, palette="bright", linewidth=1, edgecolor=".2")

plt.xlabel('Traffic Sources')
plt.ylabel('Avg no. of visits per Month')

plt.title('Average number of website visits by the user per Month per device')

plt.show()

figure = plot.get_figure()
figure.savefig('Average number of website visits by the user per Month per device.jpeg')
```



Numbers of new and returning customers based on their gender

```
In [117]: plt.figure(figsize = (18, 8), dpi = 45)

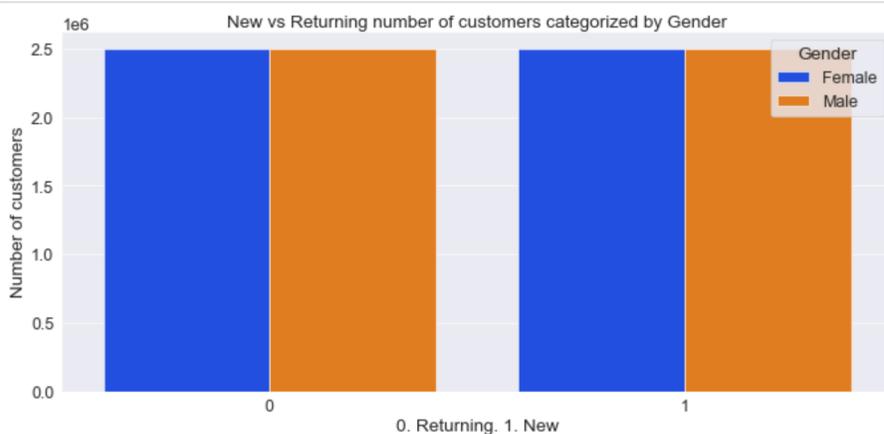
sns.countplot(data = df, x = 'IsNew', hue = 'Gender', palette="bright")

plt.xlabel('0. Returning. 1. New')
plt.ylabel('Number of customers')

plt.title('New vs Returning number of customers categorized by Gender')

plt.show()

figure = plot.get_figure()
figure.savefig('Numbers of new and returning customers based on their gender.jpeg')
```



Traffic by Traffic sources categorized by Devices

```
In [118]: plt.figure(figsize = (20, 8), dpi = 60)

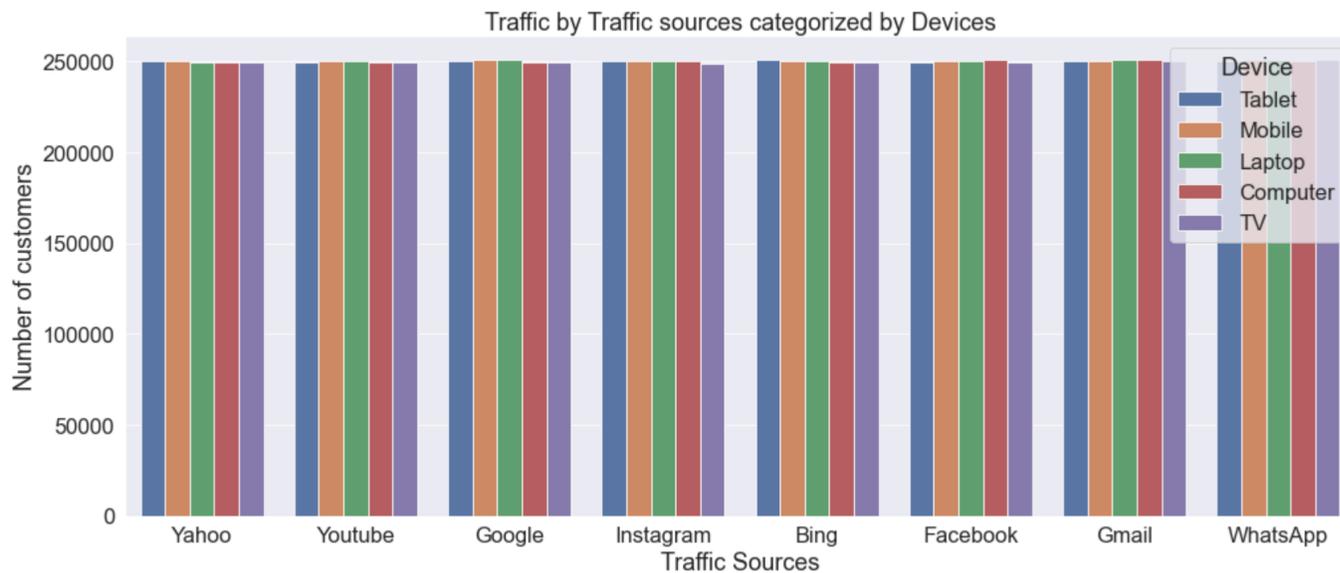
sns.countplot(data = df, x = 'TrafficSource', hue = 'Device')

plt.xlabel('Traffic Sources')
plt.ylabel('Number of customers')

plt.title('Traffic by Traffic sources categorized by Devices')

plt.show()

figure = plot.get_figure()
figure.savefig('Traffic by Traffic sources categorized by Devices.jpeg')
```



No. of Cutomers in every month of every year

```
In [119]: plt.figure(figsize = (20, 8), dpi = 80)

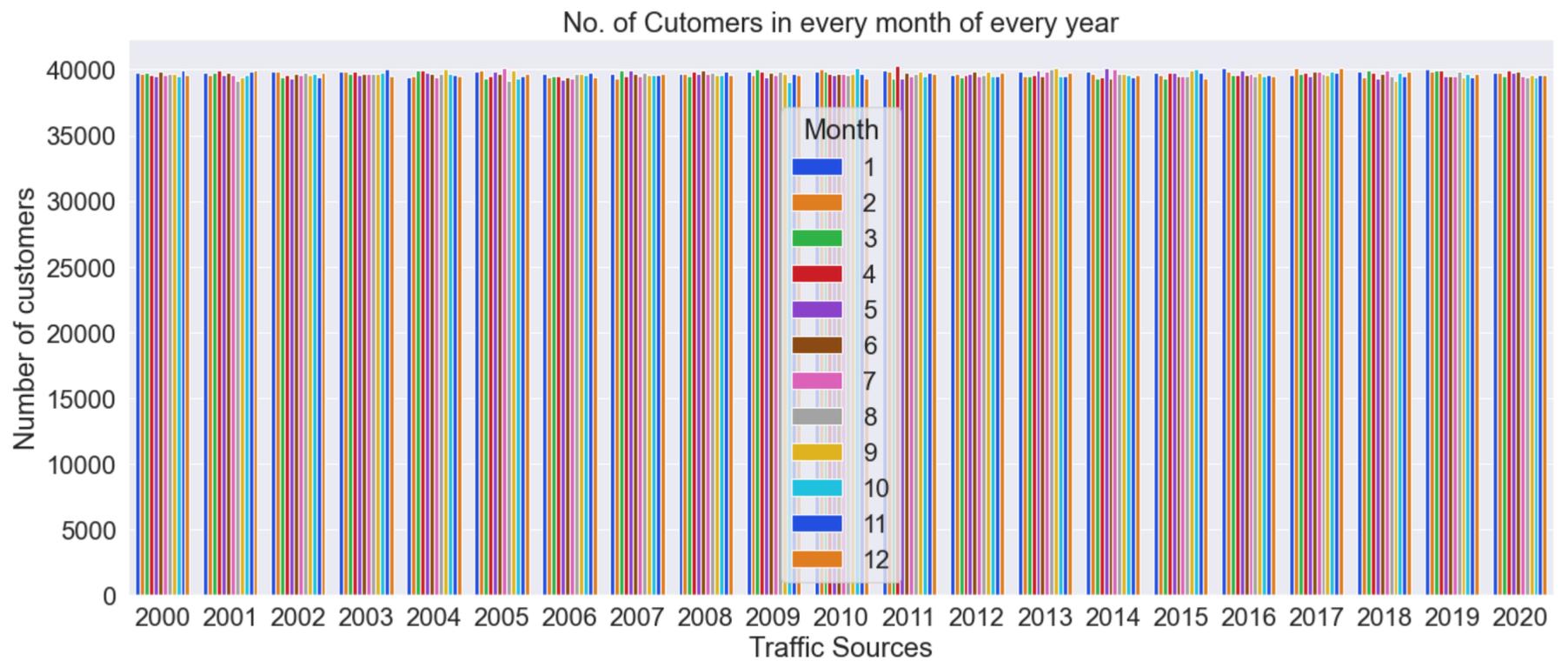
sns.countplot(data = df, x = 'Year', hue = 'Month', palette="bright")

plt.xlabel('Traffic Sources')
plt.ylabel('Number of customers')

plt.title('No. of Cutomers in every month of every year')

plt.show()

figure = plot.get_figure()
figure.savefig('No. of Cutomers in every month of every year.jpeg')
```



Queries

Q1: What is the average discount given to Female Indian customers in last 5 years?

```
In [120]: df[(df['Country'] == 'India') & (df['Gender'] == 'Female') & (df['Year'] > 2015)][ 'Discount %'].mean()
```

```
Out[120]: 5.511597424061093
```

Q2: What is the average number of items bought by Pakistani customers from Sindh in summers and paid in Cash?

```
In [121]: df[(df['Country']=='Pakistan') & ((df['Month']==6) | (df['Month']==7)) & (df['PaymentMethod']=='Cash')][ 'NoOfItems'].mean()
```

```
Out[121]: 10.53390239974231
```

Q3: How many new users visited website from Safari Web Browser via mobile?

```
In [122]: df[(df['IsNew'] == 1) & (df['Browser'] == 'Safari') & (df['Device'] == 'Mobile')].shape[0]
```

```
Out[122]: 100141
```

Q4: Which Traffic Source brought maximum traffic to the website?

```
In [123]: df['TrafficSource'].value_counts().index[df['TrafficSource'].value_counts().argmax()]
```

```
Out[123]: 'Gmail'
```

Q5: How many Indian male users visited the website in 2020 on the weekend from Safari Browser?

```
In [124]: df[(df['Country'] == 'India') & (df['Gender'] == 'Male') & (df['Year'] == 2020) & (df['Weekend'] == 1) & (df['Browser'] == 'Safari')].shape[0]
```

```
Out[124]: 399
```

Q6 What is the average page/session ratio of new UK users from Android mobile in 2018?

```
In [125]: df[(df['IsNew'] == 1) & (df['Country'] == 'UK') & (df['OS'] == 'Android') & (df['Device'] == 'Mobile') & (df['Year'] == 2018)][['Pages/Session']].mean()
```

```
Out[125]: 7.805405405405406
```

Q7: What is the average discount availed by Sindh customers who chose Credit/Debit card payment method from Whatsapp traffic source?

```
In [126]: df[(df['Country'] == 'Pakistan') & (df['Province'] == 'Sindh') & (df['PaymentMethod'] == 'Credit/Debit card') & (df['TrafficSource'] == 'WhatsApp)][['Discount %']].mean()
```

```
Out[126]: 5.391162029459902
```

Q8: What is the average price order By Sindh customers who visited the website at least 10 times a month in 2020?

```
In [127]: df[(df['Language'] == 'Sindhi') & (df['TimesInMonth'] >= 10) & (df['Year'] == 2020)][['TotalPrice']].mean()
```

```
Out[127]: 10283.884487450861
```

Q9: How many items were shipped via Same-day delivery to Tharparkar in 2015?

```
In [128]: df[(df['ShippedVia'] == 'Same-day delivery') & (df['District'] == 'Tharparkar') & (df['Year'] == 2015)][['NoOfItems']].sum()
```

```
Out[128]: 4910
```

Q10 What is the average age of users who joined website through mobile from Karachi, Sindh, Pakistan?

```
In [129]: df[(df['Country'] == 'Pakistan') & (df['Province'] == 'Sindh') & (df['City'] == 'Karachi') & (df['Device'] == 'Mobile)][['Age']].mean()
```

```
Out[129]: 38.348837209302324
```

```
In [ ]:
```