**Lecture 1: What you need to start**

Hi there, **welcome** to this course of Sales data analysis in python. Here, we will go advanced in data analysis using graphical representations in python.

We hope you are already familiar with the language 'English' just. Yeah! Even if you are beginner you can start this course and will hopefully enjoy it!

**Lecture 2: What in this course**

We will:

* Clean our data
* Analyze it
* Extract some patterns in data
* Extract product giving good result economically
* Extract the type of audience interested in shopping

And much more! gear up!

**Lecture 3: Dataset we will be working with**

We will be using an Indian sales dataset. The dataset contain almost 10k rows of data with many columns like customer name, costumer id, product customer bought, amount he spent, number of orders he made and a few more. You can view and download that dataset from datasets giant (kaggle)  [here](https://www.kaggle.com/datasets/aman879788/diwali-sales-data/" \t "_blank)

**Lecture 4 is video lecture explaining the dataset**

**Lecture 5: importing dataset in Python**

First of all install the dataset from given link and import it in python using pandas module (a convenient way to work with datasets)

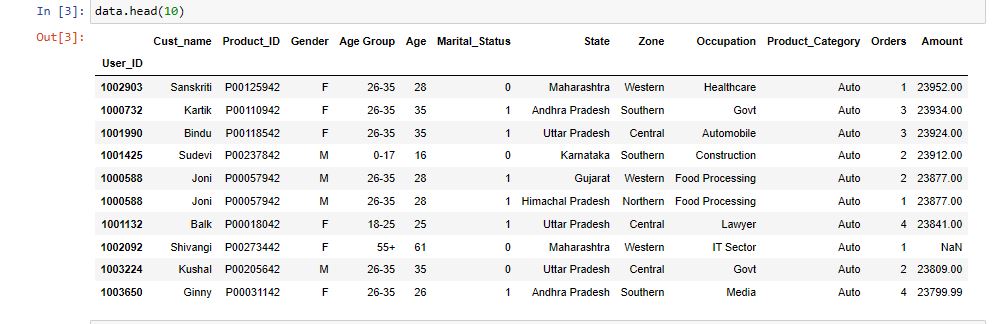
import pandas

data = pd.read\_csv('sales data.csv', index\_col=0, encoding='windows-1252')

* **First argument** is the path of dataset. Change the path according to your dataset locations.
* **Second argument is to change index** of data and make it first column of dataset. You can also skip this, it will have no impact. First column of dataset is customer id's. So instead of having index as 1,2,3.... I decided to have it as individual id's.
* Third argument is additional argument (**if you are having problem in loading dataset without this argument, use this one**. It has no impact on later workings)

After these commands you can view your data.

Dataset looks like this:

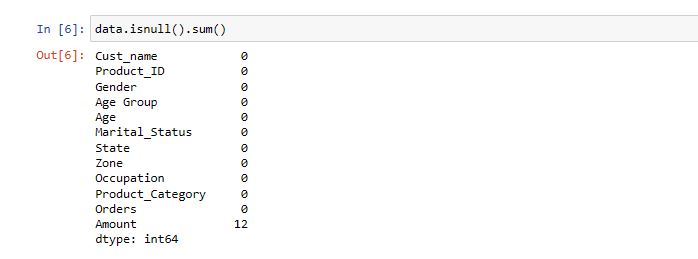


**Lecture 6 is video explanation of previous lecture**

**Lecture 7: Dealing with missing data**

You can use data.info() command to get some information about data.

Know try out **data.isnull().sum()** This command will tell you how many missing data do we have in our dataset.



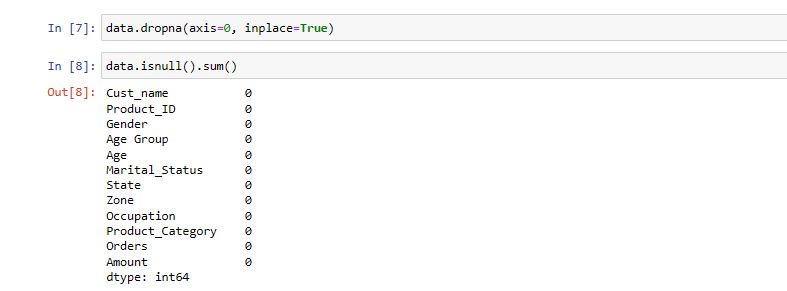
Here we see almost all data is good but there are missing values in 12 amount column of dataset. While plotting graphs these missing values can cause disturbance. So we will just delete these data points.

**data.dropna(axis=0, inplace=True)**

* This code will drop missing values in the rows. **axis =0** means to look for missing value in each row one by one. Any row with any missing value will be deleted.
* **inplace=True** means to do this step and update the current datasets.

Again when you will check out for missing values in dataset using **data.isnull().sum()**

you will then get a clean data.



**Lecture 8 is video explanation of upper lecture**

**Lecture 9: Plotting graphs, extracting data**

We will be using seaborn for analysis for this dataset.

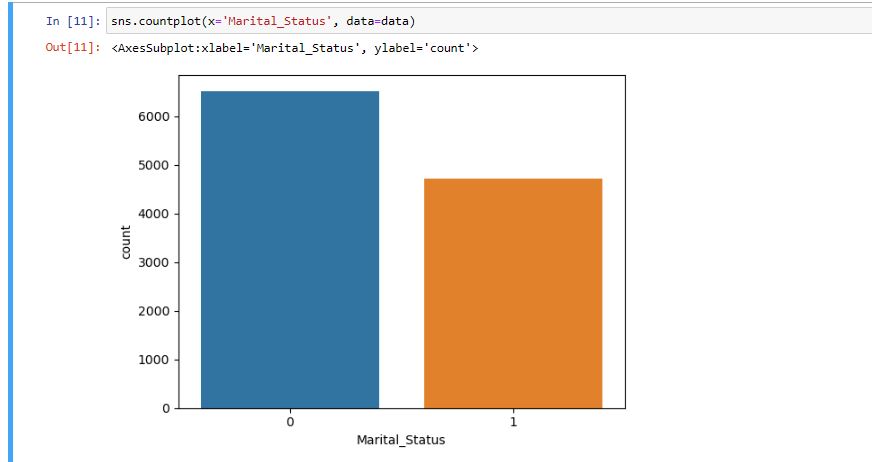
import seaborn as sns Use this command to import seaborn

Now lets plot some count plots!

seaborn takes two parameter, first is the column name and other is the whole data.

I have used data=data because we created a variable 'data'

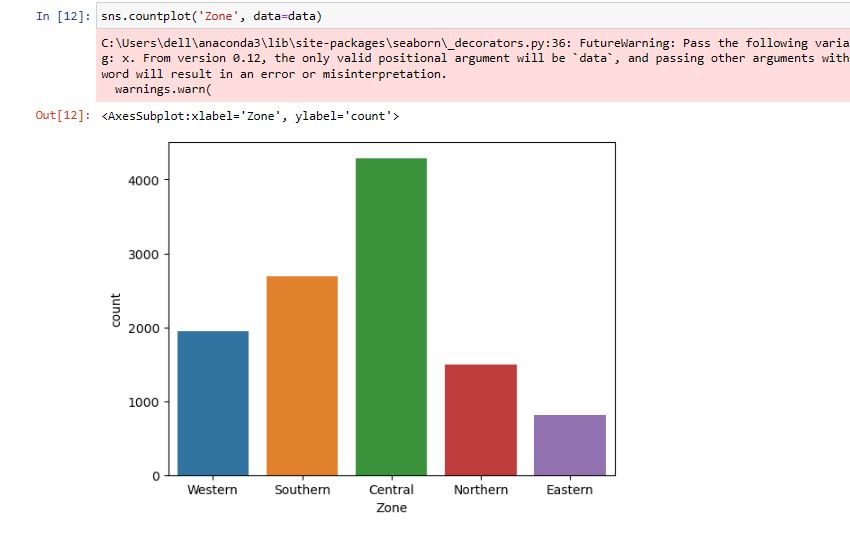
Lets see which customers tend to do more shopping...married or unmarried:



This is a binary data in which 0 represent unmarried and 1 represent married.. According to this dataset, we see that:

* **Unmarried peoples are doing more shopping**

Now lets see in which zone, there is more sales:



Graph shows that central zone has greatest sales rate...and eastern zone has least number of sales. Southern is also going average.

* **Central zone has most sales**
* **Eastern zone has minimum sales**

If you are a business, it can be good for you to know which category of product have most sales.. Lets do that.

plt.figure(figsize=(15,10))

plt.xticks(rotation=70)

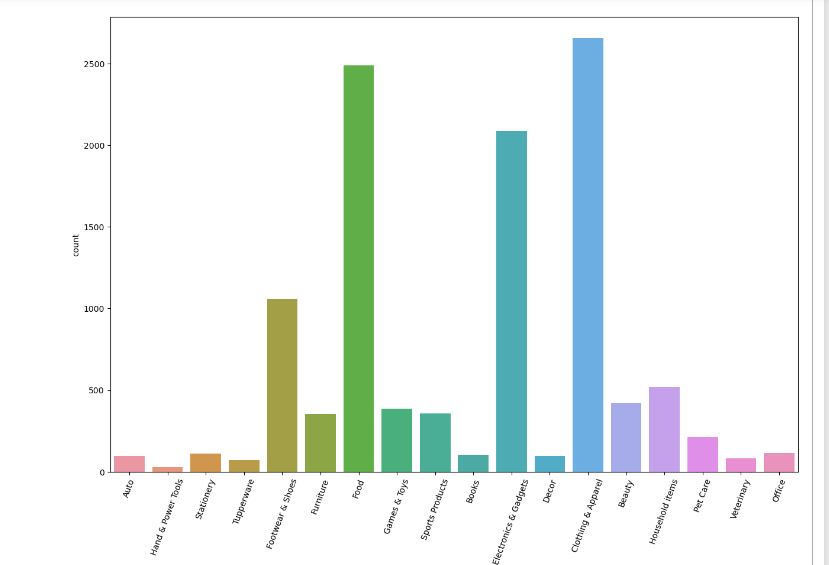
sns.countplot('Product\_Category', data=data)

First two lines are to make our graph size big, and rotate the names for better visualization. (you will have to import pyplot for this)

import matplotlib.pyplot as plt

Use this command for import

Output of upper code:



Greaat!! You see how whole market is based on clothing items and food products only! other products have not much orders compared to those two giants. By the way shoes and footwear is also good area to invest!

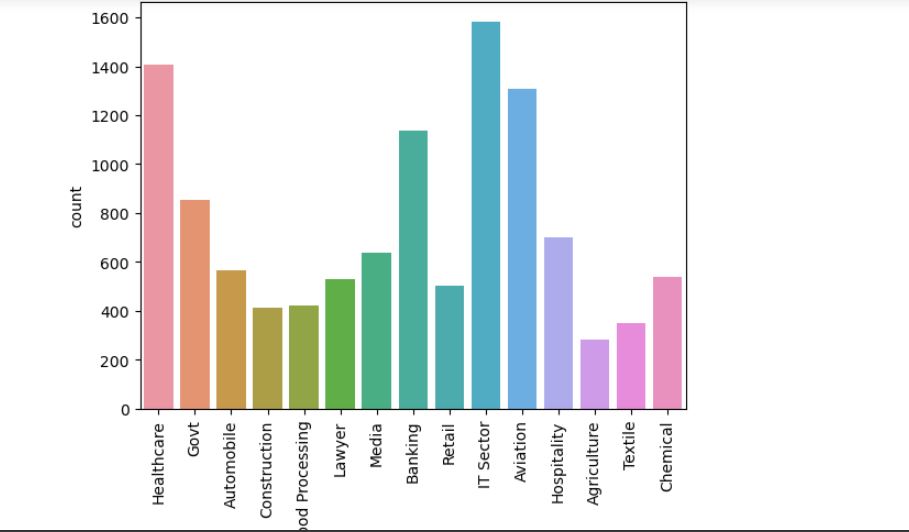
* **Most purchases: Food, Cloth items**
* **Least purchase: Books, power tools, Decor, office etc**

Lets have one last analysis on peoples who are purchasing most.

plt.figure(figsize=(15,10))

plt.xticks(rotation=70)

sns.countplot('Occupation', data=data)



**Lecture 10 and 11 are video explanation of upper lecture**

**Lecture 12: comparison graphs**

Now, we will be drawing comparison graphs to find relation between two of column of dataset. This will be something advanced and you will for sure enjoy it! I advice you to open your compiler along with and practice the code as well. You can practice same code or even a different on on this dataset or the data you are having.

**Lecture 13: Advanced Plotting, classifying sales on bases of Occupation and Gender both**

Now lets plot something advanced! You remember when we plotted the graph in which the data points was displayed in terms of occupation using this code:

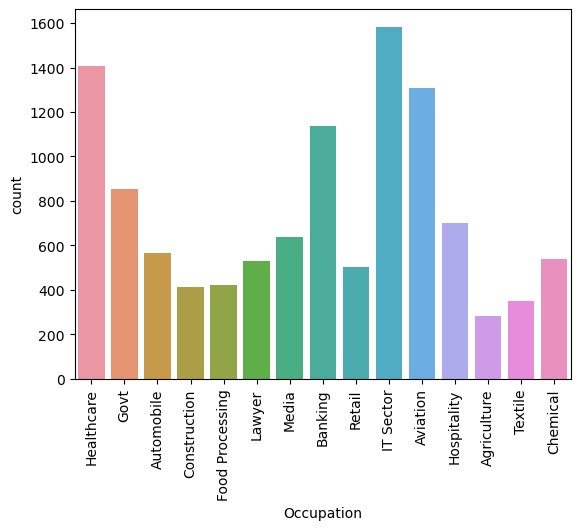
#plt.figure(figsize=(20,8))

plt.xticks(rotation=90)

sns.countplot('Occupation', data=data)

I have removed the first argument by commenting it out. That line of code play around with the size of graph. In addition, second line of code is to rotate the x-axis labels.

We have this graph from upper code:



Now I want to also know how many males and how many females in each of the sector. That will be robust!

For that we will add to add one additional argument to our code. that is called 'hue'.

Lets see this in action!

#plt.figure(figsize=(20,8))

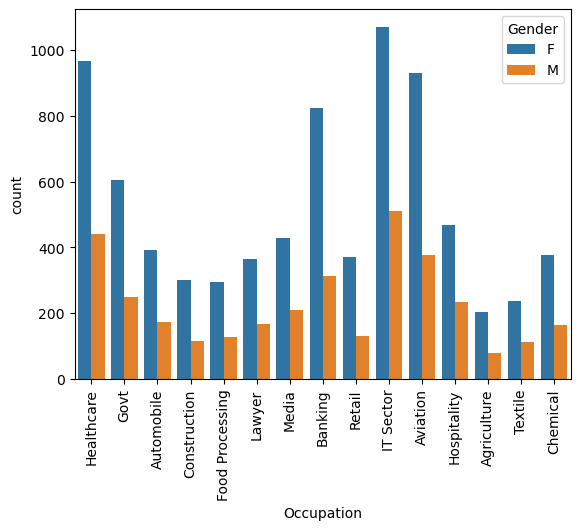
plt.xticks(rotation=90)

sns.countplot('Occupation', hue='Gender' , data=data)

If you want to change the figure size just remove the hashtag from that line.

Notice that in this code I have a new argument named **'hue'** and I have passed the column name Gender to it because I want to classify occupation on basis of gender.

This will return me this graph:



**Greaatt!!!** This was something advanced. There is so much in this graph.

There are two legends (at top right) in the the graph which are automatically generated to guide us whether which bar belongs to which category. As we see each blue bar is representing number of females, and each orange bar is representing males in our dataset. It is visible from the graph that in each sector or occupation females are doing shopping even more than the two time of males. Well, **women empowerment** maybe. But that is a amazing fact, If you want to start some sales program in India, try to target the women from IT sector. That will probably give you a boost. Lets make something even more interesting,

**Lecture 14 is video detail of upper lecture**

**Lecture 15: Classifying sales on the bases of category of product and gender**

Let us analyze product categories on the bases of gender. Which gender is more interested in which category (lets see it in details in a graph but most probably females will again be on the top)

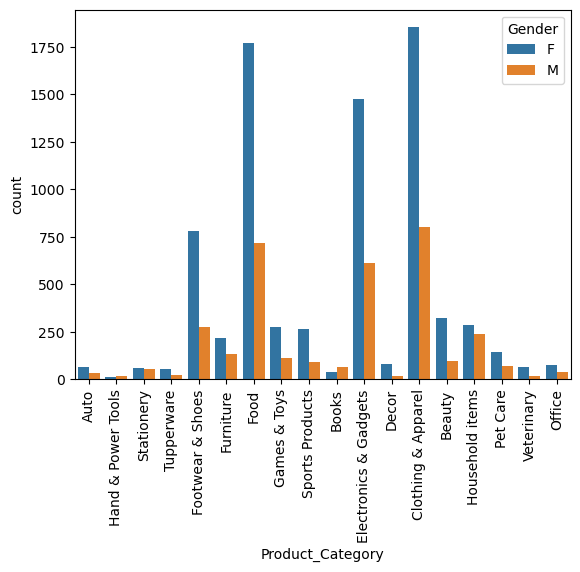
#plt.figure(figsize=(20,10))

plt.xticks(rotation=90)

sns.countplot('Product\_Category',hue='Gender' , data=data)

Using this code we will generate a graph.

Output of this code:



As expected, blue bar(specified for females) is way more higher than the males bar. As Exceptions are always there, have a look the category ‘Hand & Power tools’. There is very low rate of purchase in this category but having a focused look, It looks like men are purchasing more than women in this field.

Have a detailed look on this graph and if you are also working on your compiler, Write a program to classify sales on the bases of state they belong to, and whether they are married or unmarried.

**Lecture 16: analyzing sales with age group and occupation**

The next graph we will be plotting will be a little more advanced. We will look not into genders but in the age-groups and see which age group in more interested in sales. Let’s do this!

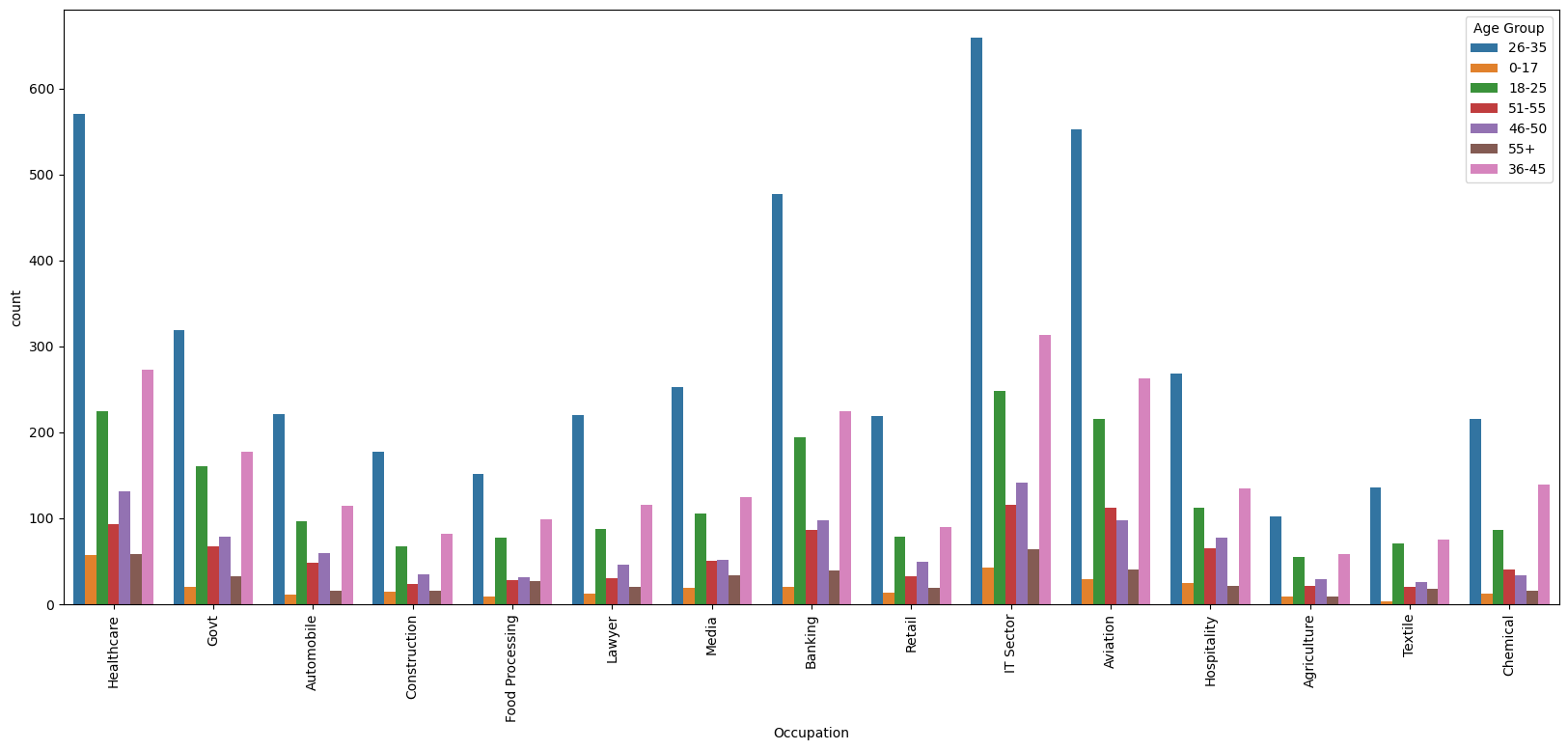
plt.figure(figsize=(20,8))

plt.xticks(rotation=90)

sns.countplot('Occupation', hue='Age Group' , data=data)

notice that **‘Age Group’** is a column on the basis of which we will classify occupations. Have a look at our raw data if getting any confusion.

Output of this code:



**Boom!! That’s colorful!!** Look at the legends on the top right corner which is specifying the number of age groups we have in our data. We are having total of seven age groups in our data. Hence the seven bars for each occupation.

**Each bar tells us how many peoples from this age-group (legend) and this occupation (labeled on x-axis) are purchasing things.** Look at the graph and legends carefully and try to extract something before reading further

.

.

It is visible in graph that:

* The blue bar representing age group 26-35 is more permanent in each the the occupation
* Orange bar representing 0-17 old youngsters are not purchasing things.
* 55+ peoples are not so much, but in comparing to other categories there bar is more prominent in Health care category(maybe doctors).
* Overall sales are based on the age group of 26-35 only. 35-40 age group is also active but not with that rate!
* You can extract more features the same way

**Adding Boundaries in our Graph:**

We just have to add one more argument to our code:

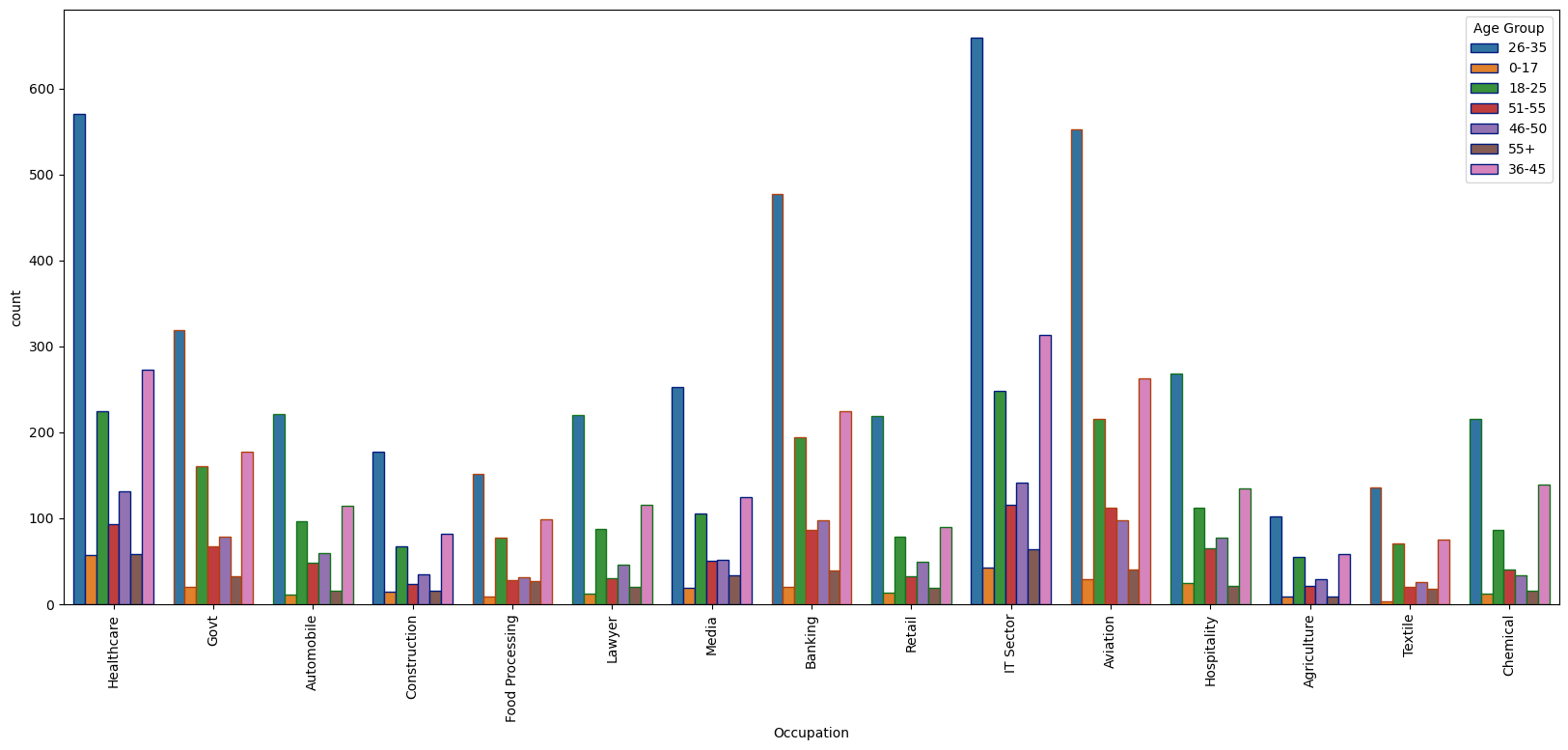
plt.figure(figsize=(20,8))

plt.xticks(rotation=90)

sns.countplot('Occupation', hue='Age Group' , data=data, edgecolor = sns.color\_palette('dark', 3))

I have added **edgecolor = sns.color\_palette('dark', 3)** this argument to add edge color from the category of dark colors from shade 3, You can play around with this.

Here we got this graph from upper updated code:



Compare both graphs and notice the difference how this graph is having borders and each bar is permanent.

Everything is same except for border. You can use this argument in any other graph of seaborn as well.

**Lecture 17 is video explanation of upper lecture**

**Lecture 18: you should..**

Go to your compiler, download the dataset and repeat the code, I know this can be boring to imply same codes, but as a beginner, this will give you a good base practice and you will see the code in real time.

Then try doing something different..... (check which gender do more participate in sales and much more)

Seaborn also have some robust comparison graphs. I will upload a course on that soon!

**Task for you:**Plot a graph which will explain which product category in being purchased by specific age group.

**Lecture 19: Congratulations**

Congratulations on completing this course! you are able to load your data, clean that, analyze, extract useful features, seeing them through seaborn in a more powerful way. I will try to make more course with videos based on the audience.

Thanks for being here! It was a beautiful journey with you.

For any question, review, advice, suggestion, advice contact me on linkdin [here](https://www.linkedin.com/in/harman-waheed-84004b255/)