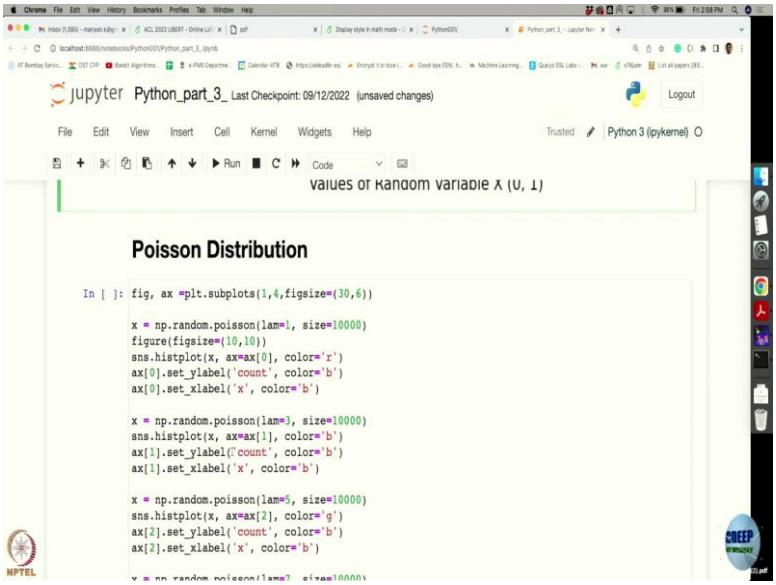
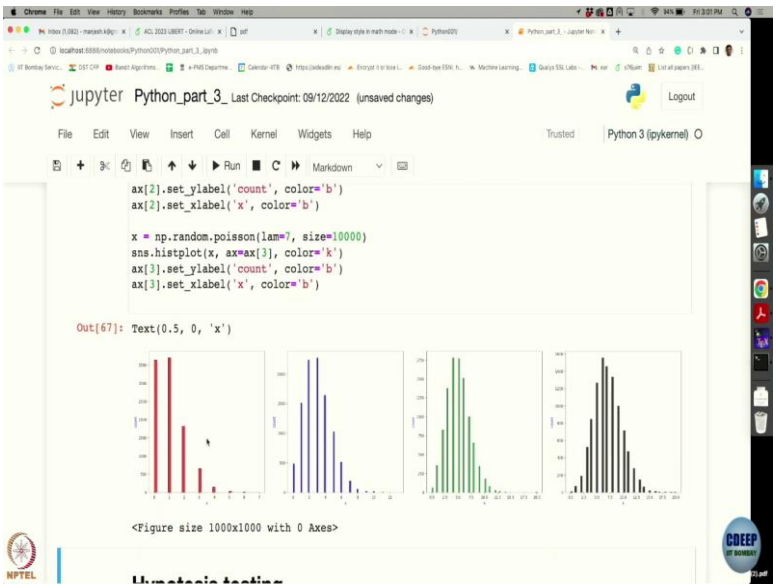
Engineering Statistics
Professor Manjesh Hanawal
Department of Industrial Engineering and Operational Research
Indian Institute of Technology, Bombay
Lecture 60
Generating Random Samples using Python,
Hypothesis Testing using Python (Contd.)

(Refer Slide Time: 0:15)





Next similarly, we can also do, Poisson distribution. But in the Poisson distribution, we will say like how to actually one figure we can use to index multiple figures, or like in one figure we can plot multiple figures as subplots. So, here I am creating subplots. And now I specifying 1 comma 4. That means what I am trying, asking Python to do is I want 4 fingers in 1 row. It is like a 1 cross 4 matrix, and 4 figures I want which are at placed adjacent to each other. And the command to get the Poisson sample generated is you just need to invoke

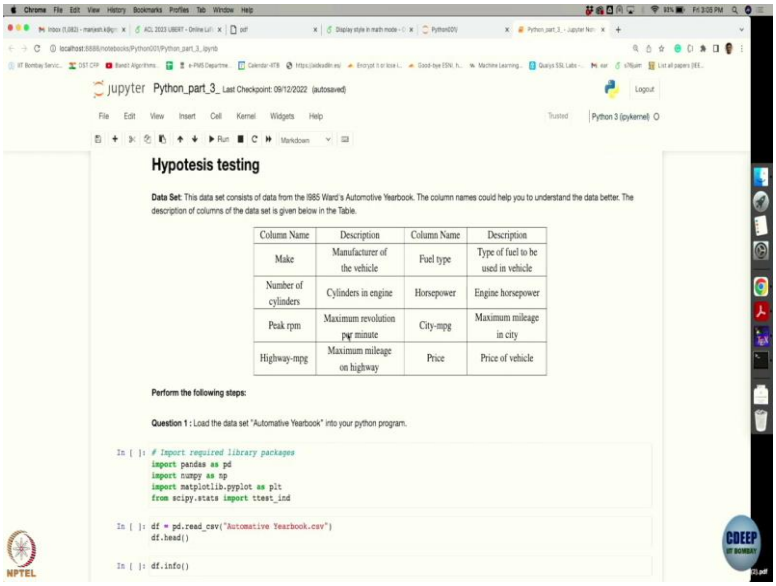
this function random.poisson by passing the lambda value, and how many samples you want, it will generate that many samples.

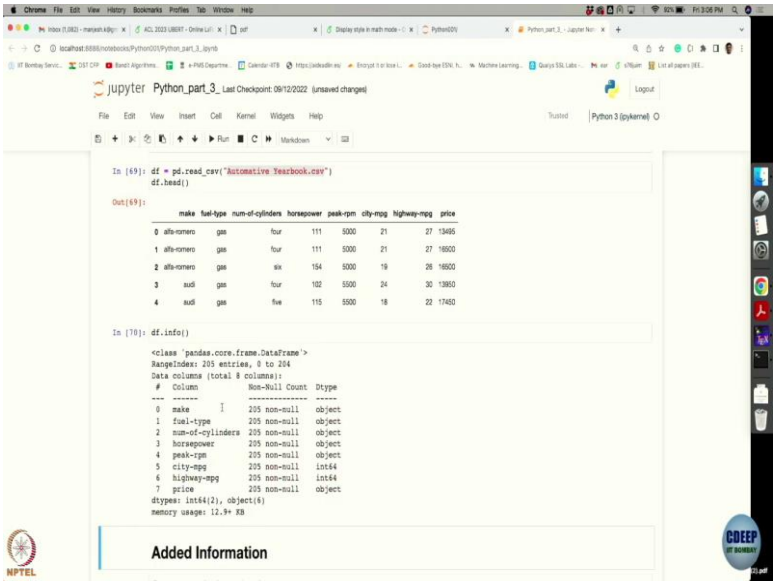
Now, I am going to decide what should be the size of my figure, and that figure size is mentioned here. And now I am going to draw the histogram of this data sample at the first element in my array of 1 cross 4, like I have declared to get 4 figures. And now the first figure I want to put my first figure in the first position. And for that, I am going to use this x equals ax 0, and then tell, plot the histogram of x to this function histplot. Let us first get that. So, and now you can set the label here, y label as count and your x label or x. And this one is the figure that corresponds to it.

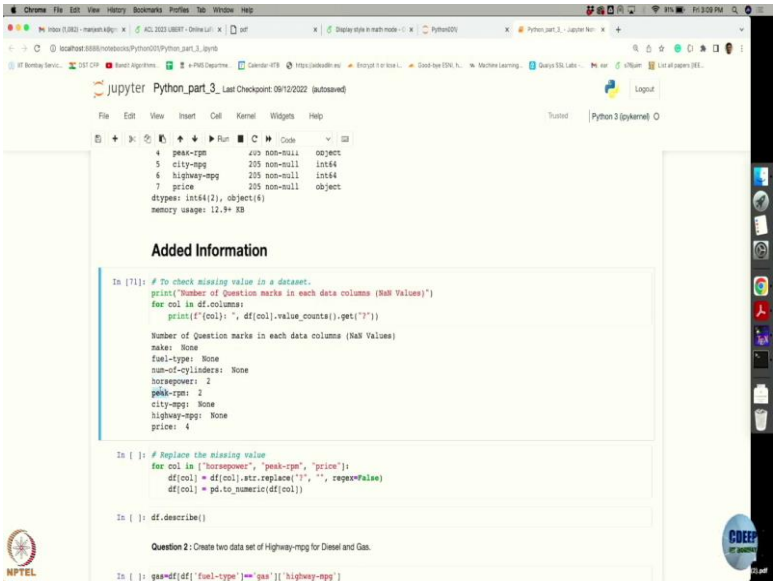
And similarly, if you want to generate data according to another lambda parameters, and if you see its histogram, but you want it to be the second element in your array, you have to just use set x equals to ax 1, then you pass the argument to the histplot. Similarly, you can do, you can generate third and fourth plot. In that case, the only thing you want to do is like, whichever data generator you generate, and then specify that handler number of that figure, to a histplot to generate this plot. So, here, we wanted 4 figures placed adjacent to each other. And that is what we got here.

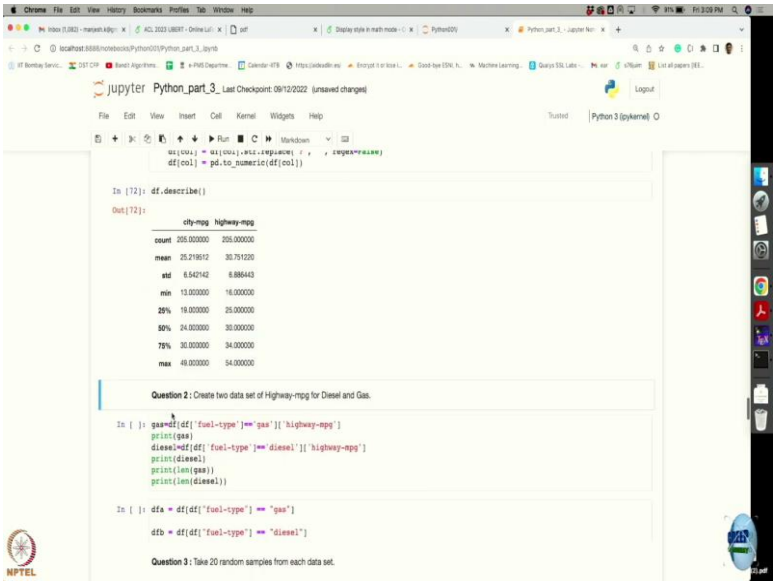
So, this is about basic distributions functions that are available in Python. I just talked about basic normal distribution, exponential distribution, gamma and beta distributions, so you can play with other distributions. Usually, this Python libraries are very well documented. So, just go and play around it.

(Refer Slide Time: 3:20)




jupyter Python_part_3_ Last Checkpoint: 09/12/2022 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (pykernel)
Run
Markdown
In [69]: df = pd.read_csv("Automative Yearbook.csv")
df.head()
Out[69]:
make fuel-type num-of-cylinders horsepower peak-rpm city-mpg highway-mpg price
0 alfa-romero gas four 111 5000 21 27 13495
1 alfa-romero gas four 111 5000 21 27 16500
2 alfa-romero gas six 154 5000 19 26 16500
3 audi gas four 102 5500 24 30 13950
4 audi gas five 115 5500 18 22 17450
In [70]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 8 columns):
Column Non-Null Count Dtype
0 make 205 non-null object
1 fuel-type 205 non-null object
2 num-of-cylinders 205 non-null object
3 horsepower 205 non-null object
4 peak-rpm 205 non-null object
5 city-mpg 205 non-null int64
6 highway-mpg 205 non-null int64
7 price 205 non-null object
dtypes: int64(2), object(6)
memory usage: 12.9+ KB
Added Information


jupyter Python_part_3_ Last Checkpoint: 09/12/2022 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (pykernel)
Run
Code
4 peak-rpm 205 non-null object
5 city-mpg 205 non-null int64
6 highway-mpg 205 non-null int64
7 price 205 non-null object
dtypes: int64(2), object(6)
memory usage: 12.9+ KB
Added Information
In [71]: # To check missing value in a dataset.
print("Number of Question marks in each data columns (NaN Values)")
for col in df.columns:
print(f"{col}: ", df[col].value_counts().get("?"))
Number of Question marks in each data columns (NaN Values)
make: None
fuel-type: None
num-of-cylinders: None
horsepower: 2
peak-rpm: 2
city-mpg: None
highway-mpg: None
price: 4
In [ ]: # Replace the missing value
for col in ["horsepower", "peak-rpm", "price"]:
df[col] = df[col].str.replace("?", "", regex=False)
df[col] = pd.to_numeric(df[col])
In [ ]: df.describe()
Question 2 : Create two data set of Highway-mpg for Diesel and Gas.
In [ ]: gas=df[df['fuel-type']=='gas']['highway-mpg']


jupyter Python_part_3_ Last Checkpoint: 09/12/2022 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (pykernel)
Run
Markdown
df[col] = pd.to_numeric(df[col])
In [72]: df.describe()
Out[72]:
city-mpg highway-mpg
count 205.000000 205.000000
mean 25.219512 30.751220
std 6.542142 6.886443
min 13.000000 16.000000
25% 19.000000 25.000000
50% 24.000000 30.000000
75% 30.000000 34.000000
max 49.000000 54.000000
Question 2 : Create two data set of Highway-mpg for Diesel and Gas.
In [ ]: gas=df[df['fuel-type']=='gas']['highway-mpg']
print(gas)
diesel=df[df['fuel-type']=='diesel']['highway-mpg']
print(diesel)
print(len(gas))
print(len(diesel))
In [ ]: dfa = df[df["fuel-type"] == "gas"]
dfb = df[df["fuel-type"] == "diesel"]
Question 3 : Take 20 random samples from each data set.

Now, we will briefly touch upon how to apply or how to use Python in some of our other tests we did, we will going to talk about hypothesis testing here. If you recall, in hypothesis testing, we wanted to check whether a null hypothesis is true or not. And for that, we come up with a decision criteria. And based on the decision, we accept or reject the null hypothesis. Here I am interested in comparing performance of vehicles, when they use diesel or petrol. So, I think this petrol I do not know they use simply the word gas here. So, maybe we will just say diesel and gas here.

So, this data set so, for to understand this we are going to use a data set which is 1985 Ward's Automotive Yearbook, the column names here specifies what is the data type it is, for example, what make, so, the column are like a descriptions like for example, a column could be make which provides the manufacturer of the vehicle, and the column number of cylinder cues explanation about how number, how many cylinders are there in the engine, and the column peak RPM use what is the maximum revolution per minute like that. So, in this we are going to compare whether the performance of diesel and gas vehicles are going to be the same or not. And we are going to do that using this data here.

So, to play with data, we are going to use this Pandas and use NumPy to see some visualisation we have use matplotlib. And also, to some statistics, we use this scipy stats package and from that we use this ttest particularly. So, let us execute this and then let us read the file. And as we discussed yesterday when I read this file, I will see the first 5 rows with description about the headers and the data in it up to 5 rows, and this info will give me a brief summary of all the columns I have, there are looks like there are about 7 columns here sorry 8 here and they have no null values here.

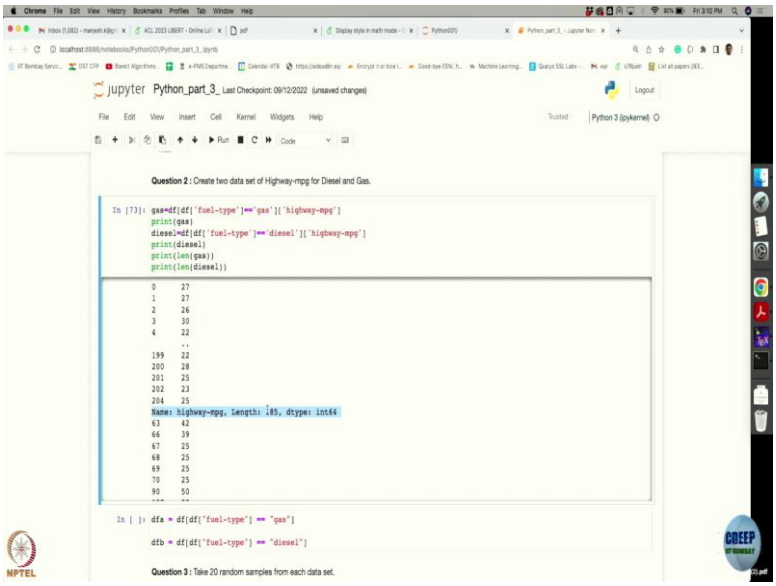
So, they have all the about the discount 2 naught 5. So, there are 8 columns. And that is saying there are 2 naught 5 entries in here which are indexed from 0 to 2 naught 4. And all these columns have some values, that is why this is all of it is showing 2 naught 5 as a null, non-null values. Now, if you want to check there is any missing values in it.

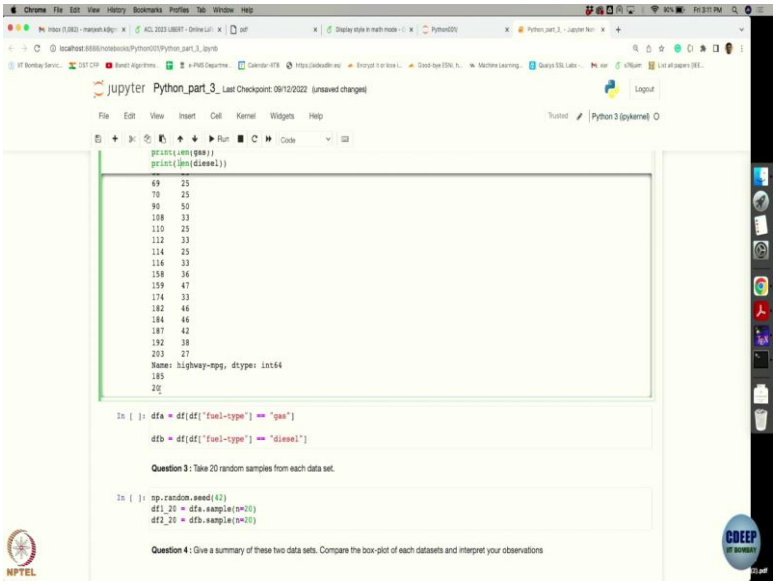
Like for example, if there is no missing value, either that place would have been left empty or there would I am going to put a question mark. And since it is saying here do not find null values, maybe there is a question mark if there is data missing there. So, let us see this is, what I am trying to do here is I am trying to find in columns checking in each columns, if there is any question mark there.

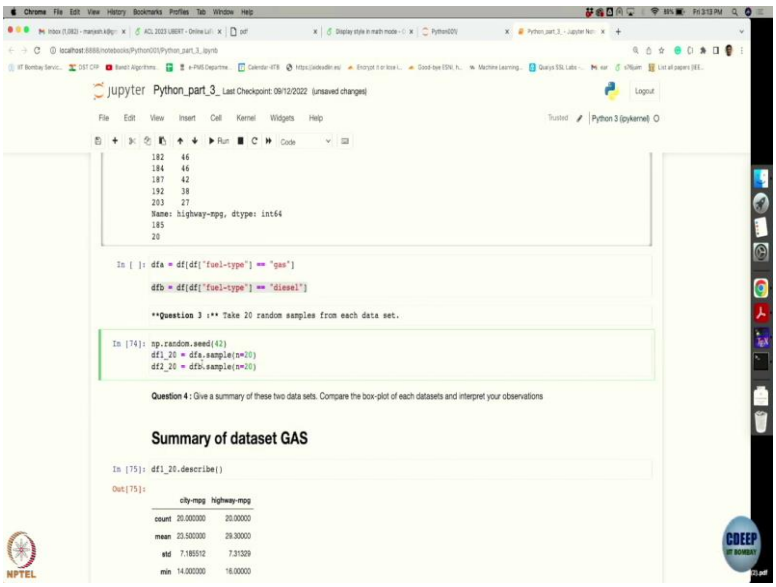
So, for that, I am we define this variable column and taking it through all possible columns, and under each column I am checking if any of this value is a question marks. To check that we have used this function value count, and telling what is the value we are looking into. And this will give us how many times question mark is appearing in columns and that is printed here. So, notice that this looks like few lines of code, but this could be very compactly written like this in one line. So, in make column there are none, which are question mark, fuel type also there is none, and in horsepower there are 2 big question mark like that.

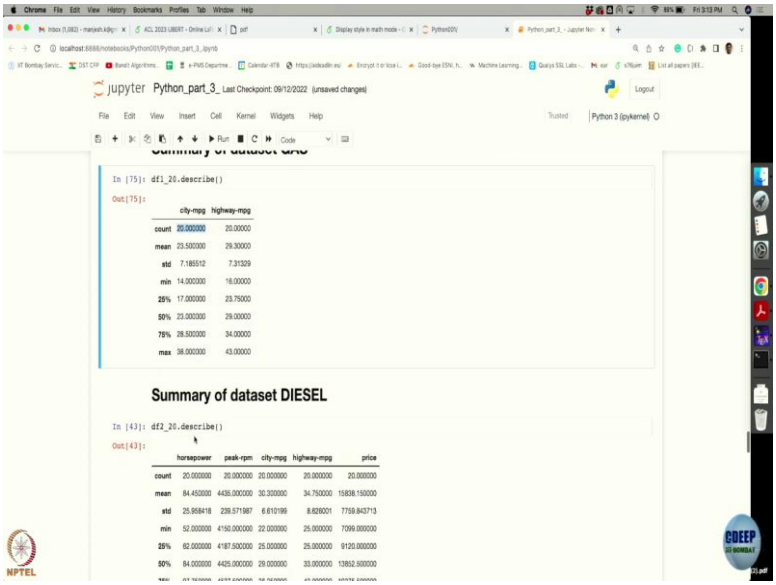
Now, you will want to remove these missing values. So, 2 missing values, first we identified the columns where the values are missing those are horsepower, peak time, and price. So, we have put the values, replace the question mark by false here. And now if you again can describe, and now everything is value here now we can count the statistics like mean, standard deviations, and all.

(Refer Slide Time: 10:15)




jupyter Python_part_3_ Last Checkpoint: 09/12/2022 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3 (ipykernel)
print(len(gas))
print(len(diesel))
69 25
70 25
90 50
108 33
110 25
112 33
114 25
116 33
158 36
159 47
174 33
182 46
184 46
187 42
192 38
203 27
Name: highway-mpg, dtype: int64
185
20
In [ ]: dfa = df[df["fuel-type"] == "gas"]
dfb = df[df["fuel-type"] == "diesel"]
Question 3 : Take 20 random samples from each data set.
In [ ]: np.random.seed(42)
df1_20 = dfa.sample(n=20)
df2_20 = dfb.sample(n=20)
Question 4 : Give a summary of these two data sets. Compare the box-plot of each datasets and interpret your observations
NPTEL
CDEEP IIT BOMBAY


jupyter Python_part_3_ Last Checkpoint: 09/12/2022 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3 (ipykernel)
182 46
184 46
187 42
192 38
203 27
Name: highway-mpg, dtype: int64
185
20
In [ ]: dfa = df[df["fuel-type"] == "gas"]
dfb = df[df["fuel-type"] == "diesel"]
Question 3 : Take 20 random samples from each data set.
In [74]: np.random.seed(42)
df1_20 = dfa.sample(n=20)
df2_20 = dfb.sample(n=20)
Question 4 : Give a summary of these two data sets. Compare the box-plot of each datasets and interpret your observations
Summary of dataset GAS
In [75]: df1_20.describe()
Out[75]:
city-mpg highway-mpg
count 20.000000 20.00000
mean 23.500000 29.30000
std 7.185512 7.31329
min 14.000000 16.00000
NPTEL
CDEEP IIT BOMBAY


jupyter Python_part_3_ Last Checkpoint: 09/12/2022 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3 (ipykernel)
In [75]: df1_20.describe()
Out[75]:
city-mpg highway-mpg
count 20.000000 20.00000
mean 23.500000 29.30000
std 7.185512 7.31329
min 14.000000 16.00000
25% 17.000000 23.75000
50% 23.000000 29.00000
75% 28.500000 34.00000
max 38.000000 43.00000
Summary of dataset DIESEL
In [43]: df2_20.describe()
Out[43]:
horsepower peak-rpm city-mpg highway-mpg price
count 20.000000 20.000000 20.000000 20.000000 20.000000
mean 84.450000 4435.000000 30.300000 34.750000 15838.150000
std 25.968418 239.571987 6.610199 8.826001 7759.843713
min 52.000000 4150.000000 22.000000 25.000000 7099.000000
25% 62.000000 4187.500000 25.000000 25.000000 9120.000000
50% 84.000000 4425.000000 29.000000 33.000000 13852.500000
NPTEL
CDEEP IIT BOMBAY

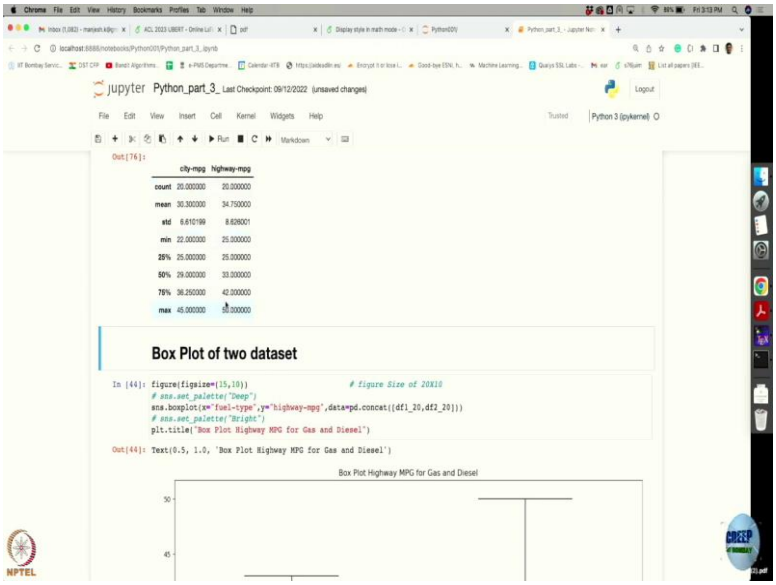
Now, what we want to do is we want to now look into their mileage. So, we want to look into their highway mileage. So, and I want to look into the vehicles which are gas and diesel type here. I have so I will look into all the columns which were all the vehicles which are running on gas and list their highway mileage. So, mpg stands for miles per gallon. And then again for fuel the vehicles which are running on diesel I again list their mileage, highway mileage and let us see, when I do this, I am going to get this, low the first maybe I am printing too much here, first I printed this gas.

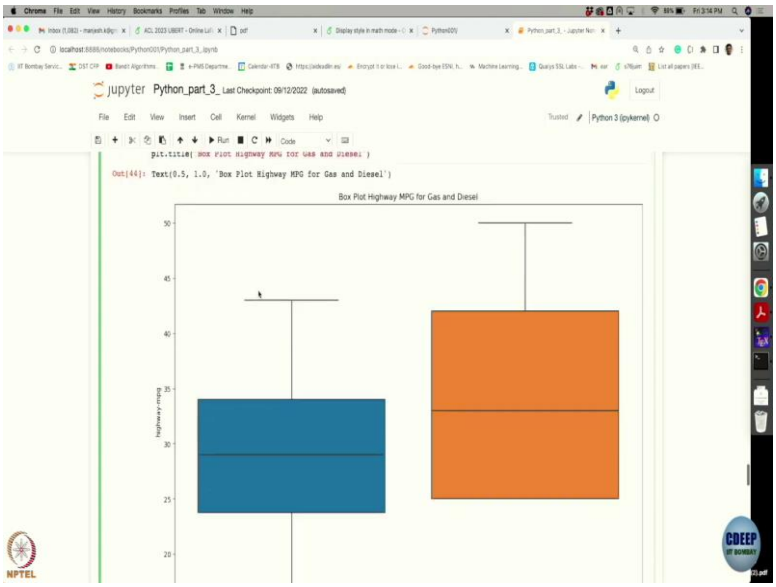
So, there looks like a gas there are about 185 element vehicles which are running on gas and the second part corresponding to this print which is showing me that there are about, this just did not show me the length here, but when I look into the length of the diesel, it is showing me that is 20. So, here I have this diesel vehicles, sorry the vehicles running on gas are about 185, and the one which are running on diesel are about 20 here.

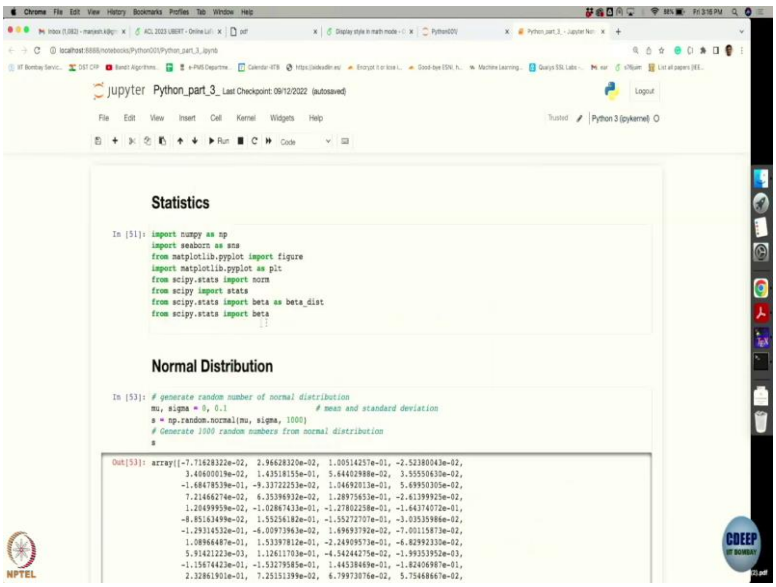
Now, I stored them separately in a new data frame, which the gas is stored in a data frame dfa, and the diesel one showed in dfp here. See, notice that storing getting extracting the column information is very easy in Python, all you have to do is first look into all the data where fuel type base gas. And when you pass it to only those indexes to give a data frame, which is storing all that data, it will only return those data's, and not others.

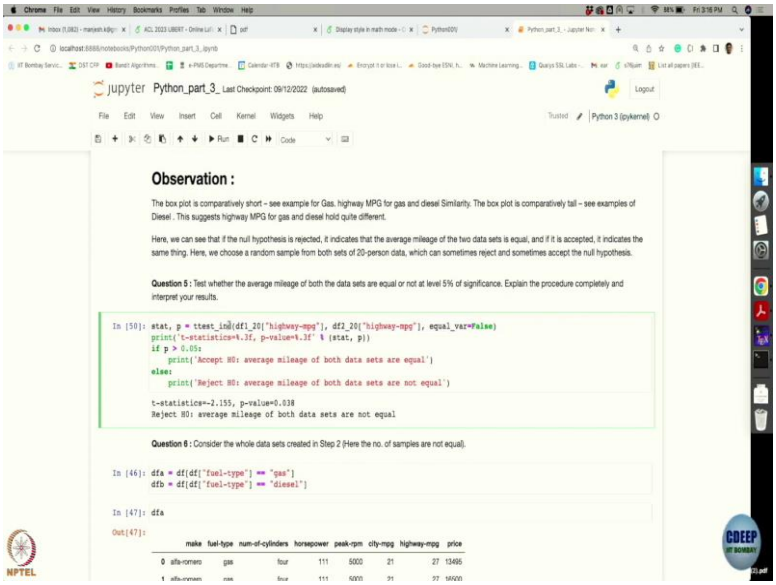
And similarly, here for the diesel case. Now, what I want to do is I want to take 20 sample from each one of these samples and use it to do my hypothesis testing whether diesel and gas have the similar mileage. Now, you will see that when I do the describe both before that, to achieve that 20, getting 20 samples, I am going to use a rand function, first the random function is seeded, so that you get the same value of this randomness, and in the first I will get 20 samples from the dfa, and another 20 samples from dfp. Now, when I look into the summary of this df1 20, you will see that there are 20 elements in that and we are seeing this mean, and median, and highway also has similar values populated here, this is you can also see.

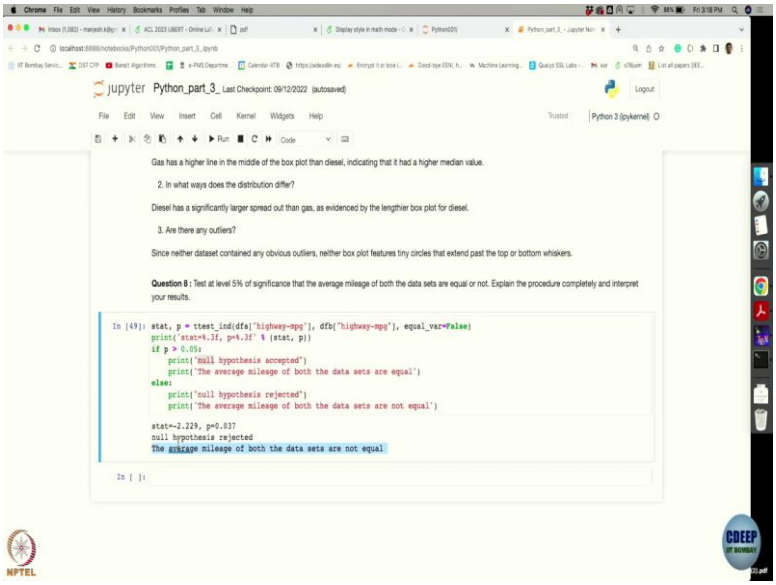
(Refer Slide Time: 13:58)











Now, often these are like just like statistic count, maybe one can want to use it pictorially for that one can use the box plot here, the box plot to do this, we are going to use box plot function in this sns library, and the plotting, for that we are going to use this command so x is simply going to be the fuel type. We have to just say x is the string is fuel type, y is highway mpg, so highway mpg here, and x is fuel type here. And the data we are going to give you is this concatenated value of df1 and df2.

Now, when I plot this, you see this 1, 2, 3, 4, 5 end up us. The bottom one here corresponding to the lowest value that is the minimum value, which is 22 here, and the middle one, so, the second one here is going to correspond to the 25th quartile of the data and the middle one is the 50th quartile, which is also the median, and this is like a 75th quartile, and the last one here is going to represent the maximum value.

So, now we have named this figure as box plot highway mpg for gas and diesel. Now, based on this you see that if you compare this just take the mean, it gives a sense that diesel has a better efficiency, has better mileage. Now, we want to test this. Now, our hypothesis is we want to test the hypothesis our average mileage or both of them are same, and my alternate hypothesis is they are not.

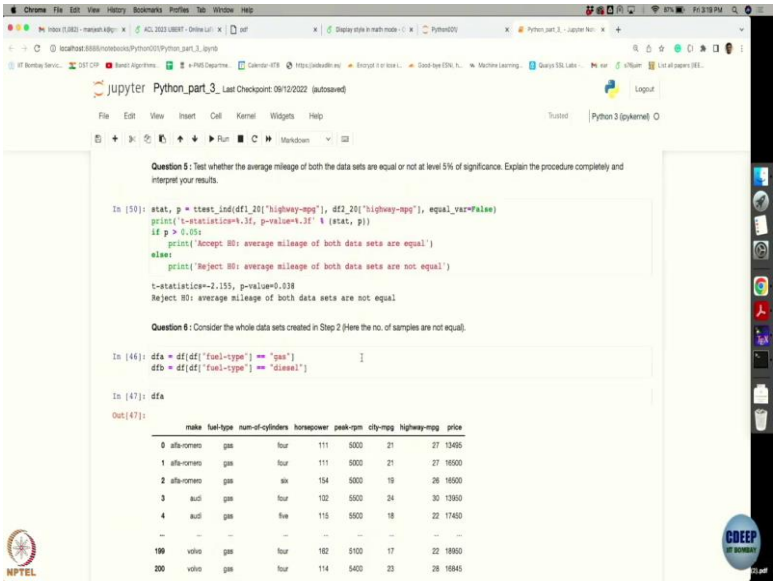
Now, I want to test it, and I want to use the t test for it. So, if you recall in hypothesis testing, the accept or reject by comparing to a threshold. And now we want to first compute the statistics and the p value of that statistics. So, to compute the t states, I am going to use this t test ind, which is the t test we had already defined a t test, we did we define t test here before, no.

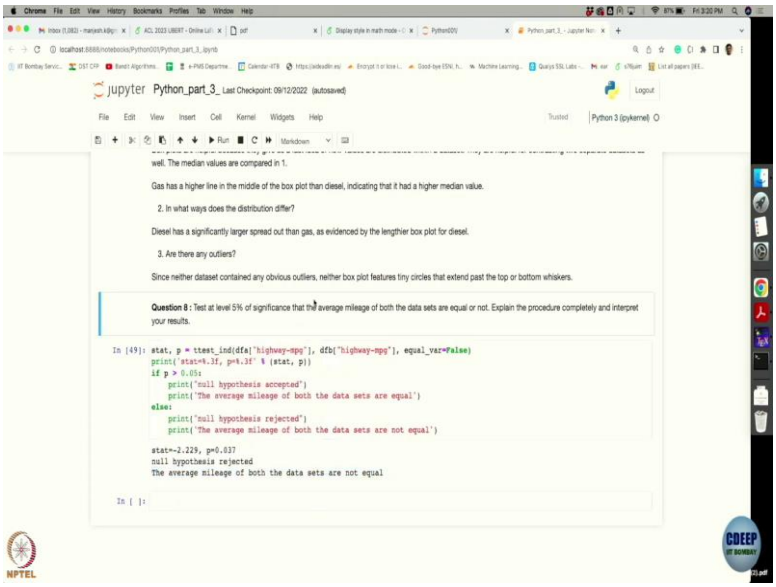
So, this t test is a function which takes the 2 data sets that we want to compare, and here equal variance tells whether I should be taking the variance of these 2 tests to be the same or not, here we have set it false. So, this will not take this data to be the same variance. And recall that when we are talking t tests, it is assumed that the data is following a quotient distribution.

Now, this function is going to return me 2 value stat and p, stat will be the value of the statistic that we compute and p is its probability that statistics is going to be larger than some number if I by default, that is like a set the significance level is set to be .05. So, we see that now p we compare against the significance level 0.05, if it is true, we say that null hypothesis accepted. And otherwise, we are going to say that it is rejected.

Now, it so happens that in this example, p value turns out to be minus 2.229 and whereas the p value is 0.037. And clearly 0.307 is going to be smaller than 0.05. So, we are going to see that the null hypothesis is going to be rejected, that is the performance of these 2 are not the same. Like this, you can do here I just showed you that, let me see why is this happening at some more things here.

(Refer Slide Time: 18:47)

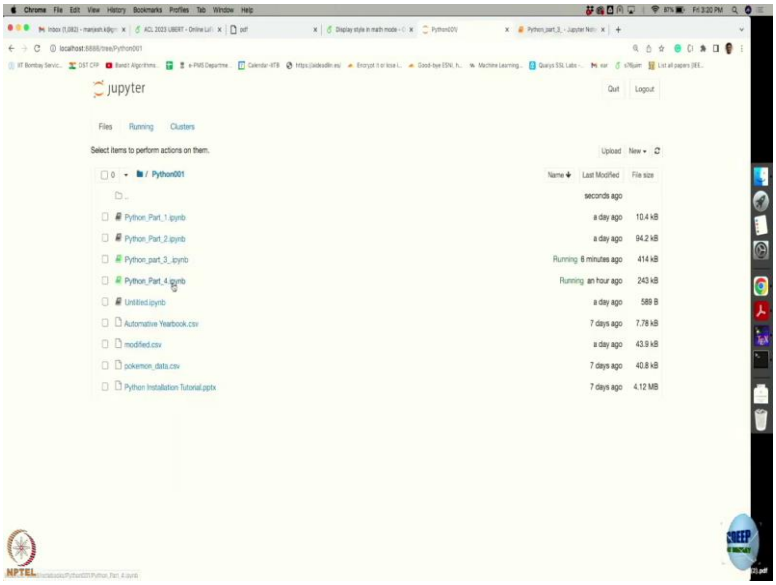


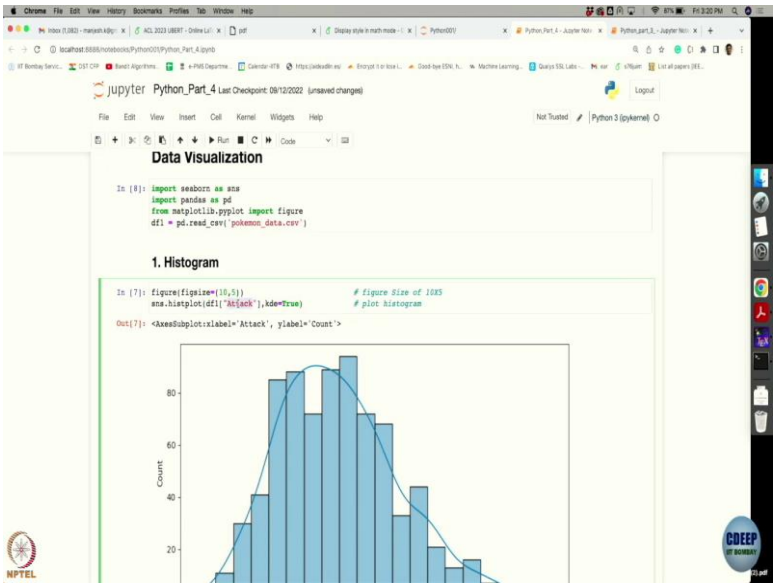


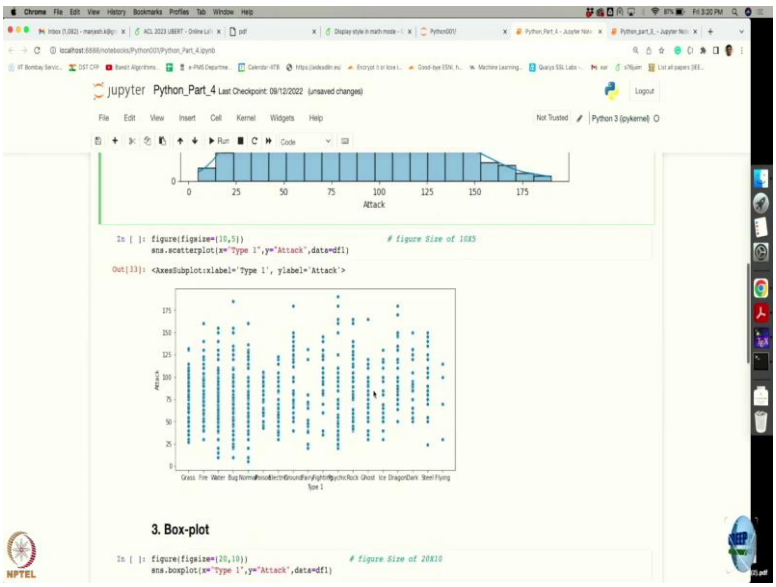
So, here based on this, you can, here with the 20 random samples generated, you conclude at that, yes, this your null hypothesis that both are the same performances is rejected. Now, you can do this again. But this time, instead of not necessarily taking 20 samples for both, maybe you can take all the possible samples available for each of these and compare.

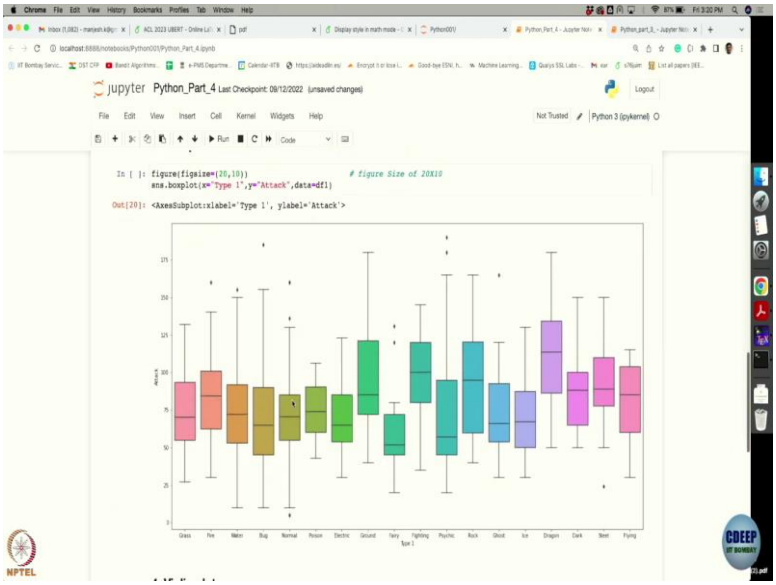
And you can do the same process here. I am not going to go into the details, but again, to get a sense here, you can first put the blocks plot. And again, here when I take all the data also like looks like again, diesel is performing better. And then you can try to test this and when you do this test, at significance level .05 again, the null hypothesis is rejected here. So, you see that like before we applying this hypothesis test, sometimes the visualisation gives more idea about how the data looks like.

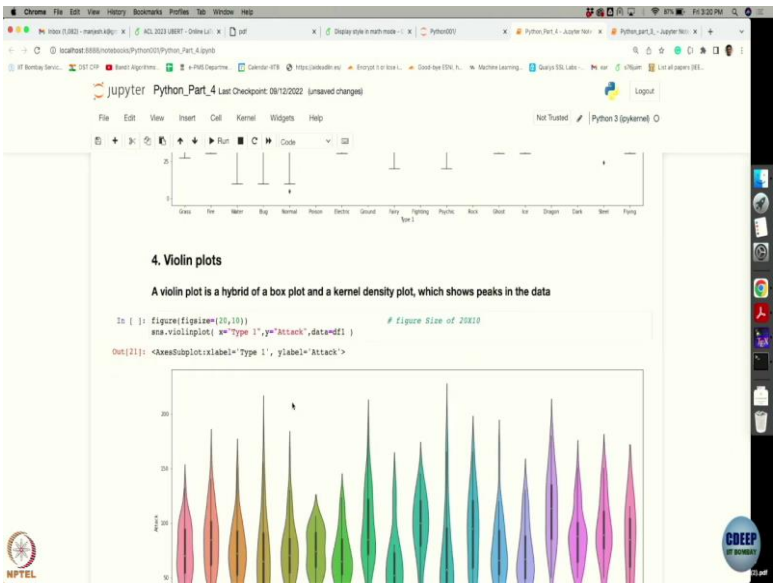
(Refer Slide Time: 19:51)











So, I want to quickly spend one a minute on this various visualisation tools. So, that is why data visualisation is important and this seaborn tool provides some pretty good functions to do that. So, one function which we already used is histogram, and when you have a data, you can see that like, for example, earlier I do this Pokémon data. In the Pokémon data, let us say I want to see how this attack column looks like I can generate the histogram like this.

And also, there are other plots called scatter plots, box plot we just saw and there is another things violin plots, these are other good visualisation techniques. I encourage all of you to play with them. So, with that, we will stop. But this Python is a very comes with a very rich libraries. So, you can readily use it to do good statistical analysis. I hope whatever we discuss is going to be helpful to you.