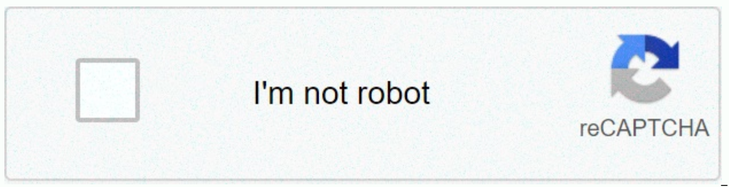

Continue

Dataframe object has no attribute' value_counts'

Hello, I'm in the task of manifesting regularly, . When I ran the code that led to the pass in the terminal of course, I got a strange problem in the local Jupyter laptop. In the course code, using a capture group that seems to extract the desired information: sample = r[(\w+)] tag_freq = titles.str.extract(pattern).value_counts() However, when I ran the above code in the Jupyter laptop, I received an attributeError property error: 'DataFrame' object does not have the 'value_counts' attribute. I checked the title in the cell below and it is a panda series, not a DataFrame! I also checked the values extracted by titles.str.extract(pattern) and found all the values to be NaN. The file I am using should be the same as the one used by the course since I downloaded it from the course terminal. Does anyone know why this happened? Thanks Xuehong 1 Like you made all jupyter laptop cells? Yes, I

did, many times. That's how I can check the title. Thanks Xuehong You have to share your files and laptop in order for someone to find out easier with your files. I think it's regex model. Shouldn't you get rid of parentheses? r\[(\w+)\] I received an error saying that there was no capture group when using r[(\w+)] without a way out. 1 Like April Thanks, I'm not sure how I missed the copy before [. I had it in jupyter laptop (see below). The problem is that the laptop confuses a panda series as a DataFrame in this particular cell. This is the code, sample = r[(\w+)] tag_freq = titles.str.extract(pattern).value_counts() This is the error message, AttributeError Traceback (last call) in 1 pattern = r[(\w+)] ----> 2 tag_freq = titles.str.extract(pattern).value_counts() C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in getattr(self, name) 5065 if self._info_axis._can_hold_identifiers_and_holds_name): 5066 return self[name] -> 5067 return object.getattribute(self, name) 5068 5069 def setattr(self, name, value): AttributeError: 'DataFrame' object has no attribute 'value_counts' pattern = r[(\w+)] No information has been given about the order in which specific cells are made in the Jupyter notebook. To avoid any other confusion, the code given below is given in a independent code box: This is the working code: enter the panda as pd hn = pd.read_csv(hacker_news.csv) title = hn [title] template = \[[(\w+)\] tag_freq = titles.str.extract(pattern).value_counts() Knowledge of pd. Series.str.extract: s= pd. Series(['a1', 'b2', 'c3']) b = s.str.extract(r'([ab])(\d)') >>> b 0 1 0 a 1 2 2 NaN NaN c = s.str.extract(r'([ab])') >>> b 0 0 a 1 b 2 NaN d = s.str.extract(r'([zZ])') sample >>> ValueError: does not contain shooting groups If there is a snapshot and expand = False, then pd. Series.str.extract returns a pd. If there is a capture group and expand = True (by default), then pd. Series.str.extract returns a pd. DataFrame object. If there is more than one group shooting, then pd. Series.str.extract returns a pd. DataFrame subjects Without group shooting, then ValueError: the model that contains no capture group is raised. Read more, you can read pd. Series.str.extract documentation. About your problems: xuehong.liu.pdx: AttributeError: 'DataFrame' object without 'value_counts' attribute Given from the problem, we only want a single group captured described by the model. And by default expand=False, pd. Series.str.extract(pattern) returns of the PD. Series category. You have to make sure of the following: the title is kind of pd. Series to use pd. Series.str.extract. model is kind of str. Your error is because titles.str.extract(pattern) is a pd. DataFrame object. There is no information given in your article above about what exactly the title represents. You must provide the code in the order in which it performs. Next problem: xuehong.liu.pdx: pattern = r[(\w+)] You must exit brackets [and]. That is, \[and \]. This not informs the pd function. The series.str.extract that [ ] is part of the regex statement. Otherwise, in your example code, the model is literally looking for an open parenthesis [and close]. Therefore, no patterns are found. When no model is found, ValueError: the sample contains no raised capture group. Format your code in your article: Use the 'three times back' marker to format a block of code that improves code readability, results in better quality discussion of your questions and creates a faster response time from the community. You can refer to this discourse article about markdown formatting a block of code. Hi Alvinctk, It's an error I made. I checked if I would get different errors if I modified the model and forgot to copy the original model when answering you the second question. I just copied your code paste in a jupyter cell and got the same error. I even added a statement to make sure the title is a pd series and that doesn't change anything. Not sure why my jupyter laptop acted so oddly. You can run the code in your jupyter laptop and see what can happen. This never happened to me. this is the code, enter the panda as pd hn = pd.read_csv(hacker_news.csv) title = hn [title] title = pd. Series (title) # note: giving this opinion out did not make any difference. pattern = [(\w+)] tag_freq = titles.str.extract(pattern).value_counts() This is the AttributeError Traceback error message for 6 7 patterns = [(\w+)] ----> 8 = titles.str.extract(pattern).value_counts() C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in getattr(self, name) 5065 if 5066 return self [name] -> 5067 back object.getattribute (self, name) 5068 5069 def setattr (self, name, value): AttributeError: 'DataFrame' object has no 'value_counts' attribute The same thing happened again after I copied the code from the guide to my laptop. It seems that this on;y occurs when str.extract() is used. title = template hn['title'] = r(\w+SQL) sql_flavors = titles.str.extract(pattern, flags=re. I) sql_flavors_freq = sql_flavors.value_counts() print(sql_flavors_freq) The above code leads to the error below, AttributeError Traceback (last call) in 1 pattern = r(\w+SQL) 2 sql_flavors = titles.str.extract(pattern, flags=re. I) ----> 3 sql_flavors_freq = sql_flavors.value_counts() 4 print(sql_flavors_freq) C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in getattr(self, name) 5066 if self._info_axis._can_hold_identifiers_and_holds_name: 5066 return self[name] -> 5067 return object.getattribute(self, name) 5068 5069 def setattr(self, name, value): AttributeError: 'DataFrame' object has no 'value_counts' attribute I learned a lot while looking at your problem! I downloaded the hacker news data set to work with it in Jupyter laptop and pasted in your code to see your error. Then I read the documentation for str.extract along with rereading alvinctk's article. It seems that expand=True is the default, so titles.str.extract(pattern, flags=re. I) is returns a DataFrame object and not a series. I added expand=False causing it to return a string. pattern = r(\w+SQL) sql_flavors = titles.str.extract(pattern, flags=re. I, expand=False) sql_flavors_freq = sql_flavors.value_counts() print(sql_flavors_freq) output: PostgreSQL 27 NoSQL 16 MySQL 12 nosql 1 MemSQL 1 mySql 1 CloudSQL 1 SparkSQL 1 Name: title, dtype: int64 1 Like I was stuck with the same problem as xuehong.liu.pdx as I'm using Jupyter on my computer. Thank you all for your help and solutions!!! Iris data set from Sklearn in Sklearn Bunch format: print(type(iris)) print(iris.keys()): <class 'sklearn.utils.bunch'=> dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names', 'file name']) So that's why you can access it like: x = iris.data y = iris.target But when you read the CSV file as DataFrame as mentioned by you : iris = pd.read_csv ('iris.csv', title =None).iloc[:,2:4] iris.head() output is: 2 3 0 petal_length petal_width 1 1.4 0.2 2 1.4 0.2 3 1.3 0.2 4 1.5 0.2 Here the column names are '1' and '2'. First of all you should read CSV files such as: df = pd.read_csv('iris.csv') you should not include the title = None as your csv file includes column names i.e. headers. So now what you can do is something like this: X = df.iloc [:, 3]] # Will give you columns 2 and 3 i.e. 'petal_length' and 'petal_width' y = df.iloc [:, 4] # Label column i.e. 'species' or if you want to use </class> </class> name then: X = df [['petal_length', 'petal_width']] y = df.iloc['species'] Additionally, if you want to convert labels from string to digital format use sklearn LabelEncoder from sklearn import pre-processed le = pre-processed. LabelEncoder() y = le.fit_transform(y) Watch 141 Star 6.3k Fork 958 You cannot perform that action at this time. You're signed in with a different tab or window. Reload to refresh your session. You're signed out in a different tab or window. Reload to refresh your session. We use optional third-party analytics cookies to understand how you GitHub.com and can build better products. Learn more. We use optional third-party analytics cookies to understand how you GitHub.com and can build better products. You can always update your choices by clicking Cookie Options at the bottom of the page. For more information, see our Privacy Policy. We use cookies necessary to perform essential website functions, e.g. they are used to log you in. Learn more Always Works We use analytical cookies to understand how you use our website so that we can make them better, for example, they are used to collect information about the pages you visit and the number of clicks you need to complete tasks. Learn more