Data Wrangling

Prerequisites:

It is assumed that you will install and take a look at the following packages in python before heading to activities:

- numpy (http://www.numpy.org/)
- (http://flask.pocoo.org/) pandas (https://pandas.pydata.org/)

All of the following activities will be based on the *Books dataset* (Download the csv file (https://github.com/mysilver/COMP9321-Data-Services/blob/master/Week3_Data_Cleansing/Books.csv)). This dataset includes 20 records of Books (each of which has 15 properties) borrowed from here (https://github.com/realpython/python-data-cleaning/blob/master/Datasets/BL-Flickr-Images-Book.csv) .

Activity-1: Dropping Unwanted Columns

Description : Usually, a given task only depends on a few columns of data, and the rest of columns can be dropped to save memory. In this Activity, you need to get rid of unwanted columns in the above dataset:

```
'Edition Statement', 'Corporate Author', 'Corporate Contributors', 'Former owner', 'Eng
```

Steps :

1. Load the dataset into a dataframe
2. Print the columns of the dataset, and print the dataset to be familiar with the data
3. Calculate and print the number of nan (not a number) in each column (https://stackoverflow.com/questions/26266362/how-to-count-the-nan-values-in-a-column-in-pandas-dataframe)
4. Drop the columns of dataframe by the above-mentioned black list (https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.drop.html)
5. Print the columns of the dataset to make sure the dataframe includes only desired columns

(https://github.com/mysilver/COMP9321-Data-Services/blob/master/Week3_Data_Cleansing/activity_1.py)

Activity-2: Cleaning Columns

Description : As you can also see here, some of the columns are not of uniformed format. For example: the "Place of Publication" column contains entries such as: ("London", "London]" , "London; Virtue & Yorston") which all refer to the same place. In this activity, you are supposed to clear and uniform the entries for two

columns: "Place of Publication" and "Date of Publication".

Steps :

1. Load the dataset into a dataframe
2. Replace the cell value of "Place of Publication" with "London" if it contains "London", and replace all '-' characters with space (http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.apply.html)
3. Keep the first 4 digit number in "Date of Publication" (https://pandas.pydata.org/pandas-docs/stable/generated/pandas.Series.str.extract.html)
4. Convert "Date of Publication" cells to numbers (https://pandas.pydata.org/pandas-docs/version/0.23/generated/pandas.to_numeric.html)
5. Replace NaN with 0 for the cells of "Date of Publication" (https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.fillna.html)
6. Print the dataframe to see if the changes have been applied properly

(https://github.com/mysilver/COMP9321-Data-Services/blob/master/Week3_Data_Cleansing/activity_2.py)

Activity-3: Filtering Rows

Description : Likewise it is sometime desirable to filter the rows of the dataset. In this Activity, you are supposed to query the dataframe and filter rows.

Steps :

1. Load the dataset into a dataframe
2. Apply the cleansing methods discussed in Activity-2 to the dataframe
3. Replace the spaces in the column names with the underline character ('_') (https://stackoverflow.com/questions/13757090/pandas-column-access-w-column-names-containing-spaces/30514678)
   Because panda's query method does not work well with column names which contains white spaces
4. Filter the rows and only keep books which are published in "London" after 1866. (https://pandas.pydata.org/pandas-docs/version/0.22/generated/pandas.DataFrame.query.html)
5. Print the dataframe and validate the result

(https://github.com/mysilver/COMP9321-Data-Services/blob/master/Week3_Data_Cleansing/activity_3.py)

Activity-4: Merging Two Dataframes

Description : There are times that you need to merge/join two or more datasets to get information you need. In this activity, using a new toy dataset, you will find publication count by country (*how many books are published in each country)* . Since the given dataset dose not provide the country of publication, we use City.csv (https://github.com/mysilver/COMP9321-Data-Services/blob/master/Week3_Data_Cleansing/City.csv) to map cities to countries.

Steps :

1. Load the Books dataset into a dataframe
2. Apply the cleansing methods discussed in Activity-2 to the dataframe
3. Replace the spaces in the column names with the underline character ('_')
4. Load the City dataset into a dataframe
5. Merge two datasets based on the name of city (https://stackoverflow.com/questions/25888207/pandas-join-dataframes-on-field-with-different-names/25888471#25888471)
6. Group by the resultant dataframe based on the country column (https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.groupby.html)
   While grouping by, to keep the name of countries set as_index=False (https://stackoverflow.com/questions/41236370/what-is-as-index-in-groupby-in-pandas)
   Use count() (https://pandas.pydata.org/pandas-docs/stable/generated/pandas.core.groupby.GroupBy.count.html) to calculate the number of publications by country.
7. Print the dataframe

(https://github.com/mysilver/COMP9321-Data-Services/blob/master/Week3_Data_Cleansing/activity_4.py)

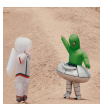
Resource created 2 months ago (Monday 14 February 2022, 01:09:39 PM), last modified 2 months ago (Saturday 26 February 2022, 02:10:35 PM).

Comments

(/COMP9321/22T1/forums/search?forum_choice=resource/73143)

(/COMP9321/22T1/forums/resource/73143)

Add a comment



Wanting Zhou (/users/z5347036) 2 months ago (Thu Mar 10 2022 20:41:00 GMT+0800 (China Standard Time))

Hi, for activity 4. When I used 'groupby' , why the output did not show the dataframe?

```
df3 = pd.merge(df, df2, left_on = ['Place_of_Publication'], right_on = ['City'])
print(df3.groupby('Country'))

if __name__ == "__main__"
```

```
new
D:\python\python.exe C:/Users/zwt10/Desktop/COMP9321-Data-Services-master/Week3_Data_Cleansing/new.py
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x0000022AA5138C40>
```

Reply

Gordon Chen (/users/z5161163) 2 months ago (Fri Mar 11 2022 00:26:03 GMT+0800 (China Standard Time))

It's another object type called DataFrameGroupBy. Have a look at this page:

https://stackoverflow.com/questions/22691010/how-t... (https://stackoverflow.com/questions/22691010/how-to-print-a-groupby-object)

Reply

Wanting Zhou (/users/z5347036) 2 months ago (Fri Mar 11 2022 09:02:19 GMT+0800 (China Standard Time))

It helps! Thank you!

Reply

Reinier De Leon (/users/z5257456) 2 months ago (Sun Mar 06 2022 21:35:59 GMT+0800 (China Standard Time))

Hi, for activity 2, what is the purpose of storing "Date of Publication" column as integers/numbers? Would it make a difference if they were stored as strings? I seem to be getting segmentation faults whenever I try to store multiple types of data in the same dataframe and I managed to stop the error if all the data was stored as 1 data type.

Reply

Yifan He (/users/z5173587) 2 months ago (Mon Feb 28 2022 17:12:44 GMT+0800 (China Standard Time)), last modified 2 months ago (Mon Feb 28 2022 17:13:21 GMT+0800 (China Standard Time))

```
def rep2(x):
    if x != x:
        return x
    return str(x)[0:4]

df = pd.read_csv('Books.csv')
df['Place of Publication'] = df['Place of Publication'].apply(rep)
print(df['Place of Publication'])
df['Date of Publication'] = df['Date of Publication'].apply(rep2)
print(df['Date of Publication'])
```

I tried an alternative way of solving task 2, part 3. It is correct, but if I change the line x != x to **np.isnan(x)** , I would get an error. But in theory, these two methods should be equivalent in many cases. I'm confused why in this situation, we cannot use np.isnan(x)?

Thanks.

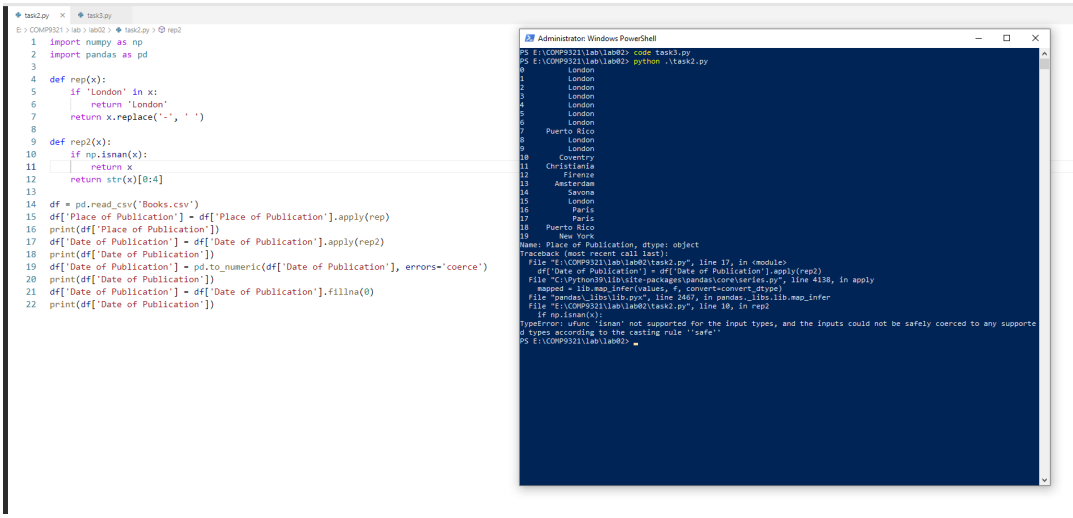
Reply

Mohammadali Yaghoubzadehfard (/users/z5138589) 2 months ago (Mon Feb 28 2022 17:28:24 GMT+0800 (China Standard Time))

Hi Yifan,
It is strange, can you share your error?
if x!=x is never true;
What I can guess is that there are some rows with null/empty/invalid values; and when whey try to convert it to string and get the first 4 charachters error happens; please double check

Reply

Yifan He (/users/z5173587) 2 months ago (Mon Feb 28 2022 17:46:30 GMT+0800 (China Standard Time))



Hi, x!= x would work, because NaN != NaN, but np.isnan(x) would get the error.

Reply

Matthew Notarangelo (/users/z5116928) 2 months ago (Mon Feb 28 2022 22:50:38 GMT+0800 (China Standard Time)), last modified 2 months ago (Mon Feb 28 2022 22:51:20 GMT+0800 (China Standard Time))

Hi Yifan, that's an interesting way of approaching the problem. I tried some solutions and found that pd.isnull(x) will work for this particular scenario. See the image below

```
def rep2(x):
    if pd.isnull(x):
        return x
    return str(x)[0:4]

df['Date of Publication'] = df['Date of Publication'].apply(rep2)
print(df['Date of Publication'])
```

The numpy documentation says that np.isnan() takes in an input array, which is probably why you were getting a type error. Let me know if pd.isnull(x) works

Reply

Yifan He (/users/z5173587) 2 months ago (Tue Mar 01 2022 02:35:35 GMT+0800 (China Standard Time))

Thanks, it works. I think pd.isna(x) also works, because pd.isnull and pd.isna are equivalent.

Reply

Zheng Fu (/users/z5285691) 2 months ago (Wed Mar 09 2022 11:07:37 GMT+0800 (China Standard Time))

somebody refer this source code in pandas, which indicate they do the same thing

https://github.com/pandas-dev/pandas/blob/537b65cb... (https://github.com/pandas-dev/pandas/blob/537b65cb0fd2aa318e089c5e38f09e81d1a3fe35/pandas/

core/dtypes/missing.py#L109)

Reply