Guide de transformation

December 7, 2023

1 Guide de transformation de colonne irrégulière

Dans les fichiers comme “listings.csv”, vous pouvez trouver une colonne irrégulière appelée ‘name’ qui contient du texte décrivant le type de l’objet à louer, l’endroit où il se trouve, la note en termes d’étoiles, le nombre de chambres, le nombre de lits et le nombre de salles de bains. Ce format de texte n’est pas pratique pour l’analyse de données, il est donc utile de transformer ce texte en colonnes individuelles. Par exemple, au lieu de

“Rental unit in Zurich · 4.78 · 1 bedroom · 1 bed · 1 bath”

on peut avoir:

	place	stars	bedrooms	beds	bathrooms	Object_type
0	Zurich	4.78	1	1	1.0	Rental unit

In order to create these columns, you can use the transformation function below. In the example below, we use Python and Pandas. Firstly, we load data: data = pd.read_csv(“listings.csv”) than we apply the custom transformation prepared for you: result_df = extract_info(data)

Pour créer ces colonnes, vous pouvez utiliser la fonction de transformation ci-dessous. Dans l’exemple ci-dessous, nous utilisons Python et Pandas. Tout d’abord, nous chargeons les données : data = pd.read_csv(“listings.csv”) ensuite, nous appliquons la transformation personnalisée préparée pour vous : result_df = extract_info(data)

Vous pouvez enregistrer les données transformées dans un nouveau fichier et utiliser ce nouveau fichier dans Tableau. result_df.to_csv(output_csv_filename, index=False)

[18]:
```
import pandas as pd
import re
import numpy as np  # Import numpy to handle 'nan' values

def extract_info(df):
    # Define regular expressions for extracting information
    place_pattern = r'in (.*?) ·'
    stars_pattern = r' (.*?) ·'
    bedrooms_pattern = r'(\d+) bedroom'
```

```
    beds_pattern = r'(\d+) bed'
    bathrooms_pattern = r'(\d+) bath'

    # Extract information from 'name' column using regex
    df['place'] = df['name'].str.extract(place_pattern)
    df['stars'] = df['name'].str.extract(stars_pattern)

    # Replace 'NEW' with 'nan' in the 'stars' column
    df['stars'] = df['stars'].replace('New', np.nan).astype(float)

    df['bedrooms'] = df['name'].str.extract(bedrooms_pattern).astype(float)
    df['beds'] = df['name'].str.extract(beds_pattern).astype(float)
    df['bathrooms'] = df['name'].str.extract(bathrooms_pattern).astype(float)

    # Add a new column containing the beginning of the name
    df['Object_type'] = df['name'].str.extract(r'^(.*?)\s*in')

    # Drop the original 'name' column
    df.drop(columns=['name'], inplace=True)

    return df
```

[19]:
```
data = pd.read_csv("listings.csv")
#df = pd.DataFrame(data)
result_df = extract_info(data)
result_df
output_csv_filename = "Transformedlistings.csv"
result_df.to_csv(output_csv_filename, index=False)
```
