コンピュータリテラシ発展 ～Pythonを学ぶ～

第11回：　計算やデータ　析をやってみよう

(shimizu@info.shonan-it.ac.jp)

今回の授業内容

今回の授業内容

-
-
-
-

前回の課題解説

前回の課題解説

-
-

解答

https://colab.research.google.com/drive/1BN8_-3cvnWOEKBMtSmVQ-gVyd1qjLbTL?usp=sharing

データの分析

データの準備

-
- Moodle **data_analysis.zip**
- Google Drive **data_analysis**
- **data_analysis**
  - customer.csv
  - item.csv
  - transaction_1.csv
  - transaction_2.csv

データの読み込み

目標：商品の購買データと顧客データを組み合わせ，購買分析を行います

- pandas ID 5
- `read_csv()` CSV pandas
- DataFrame
- DataFrame `head()` 5

```
import pandas as pd  # pandasモジュールをインポート

# CSVファイルのパスを指定
path = '/content/drive/MyDrive/???/data_analysis/customer.csv'

# 指定されたCSVファイルを読み込み，データフレームに格納
customer = pd.read_csv(path)

# データフレームの最初の5行を表示
customer.head()
```

データの読み込み

-
- pandas

データ形式	関数	解説
CSV	`read_csv()`	
excel	`read_excel()`	Excel .xls .xlsx
json	`read_json()`	JSON
html	`read_html()`	HTML

JSON wiki URL

データの読み込み

-
- pandas
- DataFrame

オプション	解説
`shape`	:15　6　　(15,6)
`columns`	
`dtypes`	

データの結合

- 2
  -
  -

データの結合

データを縦　向に結合

- `concat()`
  - transaction_1.csv　transaction_2.csv
  - 1
  - 2

データの結合

- `concat()`
  - `ignore_index` `True`
  - `ignore_index` `True`

```
path = '/content/drive/MyDrive/???/data_analysis/'

# 指定されたCSVファイルを読み込み、データフレームに格納
transaction_1 = pd.read_csv(path + 'transaction_1.csv')
transaction_2 = pd.read_csv(path + 'transaction_2.csv')

# 2つのデータフレームを結合し、新しいデータフレームに格納
# ignore_index=True は、結合後のデータフレームのインデックスを再設定することを示す
transaction = pd.concat([transaction_1, transaction_2], ignore_index=True)

transaction
```

データの結合

特定の　をキーにして横　向にデータを結合

- `merge()`
  - transaction　　ID
  - customer　　ID
  -　　ID

データの結合

- `merge()`
  - `on` ID
  - `concat()`

```
# transactionデータフレームとcustomerデータフレームを '顧客ID' 列でマージ
# '顧客ID' 列の値が一致する行を結合して、新しいデータフレームを作成
sales_data = pd.merge(transaction, customer, on='顧客ID')

# マージされたデータフレームの内容を表示
sales_data
```

データの集計

-
  -
  - ID
  - `merge` `sales_data`
  - `groupby()`
  - →
  - →
  - → `.sum()`

データの集計

-
  -
  - ID

```
# sales_data データフレームを '顧客ID' 列でグループ化し、
# '購買金額' と '購入数' 列の合計を計算
sales_per_customer = sales_data.groupby('顧客ID')[['購買金額', '購入数']].sum()

# 顧客ごとの合計購買金額と合計購入数を含むデータフレームの内容を表示
sales_per_customer
```

データの集計

- -
  - `groupby()` `GroupBy`
  - `GroupBy`

メソッド名	解説
`count()`	
`mean()`	
`sum()`	
`describe()`	,

データの集計

-
  -
  - `mean()`

```
# sales_data データフレームを '購買日' 列でグループ化し、
# '購買金額' 列の平均を計算
sales_per_day = sales_data.groupby('購買日').購買金額.mean()

# 日ごとの平均購買金額を含むデータフレームの内容を表示
sales_per_day
```

データの可視化

データを可視化するためのライブラリ

-
-
  - Matplotlib
    - Python
    -
  - seaborn
    - Matplotlib

データを可視化するためのライブラリ

-
- **Matplotlib　seaborn**　Colab
-

```
# matplotlibモジュールのpyplotサブモジュールをインポート（グラフ描画のためのツール）
from matplotlib import pyplot as plt

# seabornモジュールをインポート（データの可視化を行うための高レベルなインターフェース）
import seaborn as sns
```

データを可視化するためのライブラリ

-
-
  - Matplotlib
  - japanize_matplotlib

```
# pipコマンドを使って japanize_matplotlib パッケージをインストール
!pip install japanize_matplotlib

# japanize_matplotlib パッケージをインポート
import japanize_matplotlib
```

データを可視化するためのライブラリ

-
- sales_per_customer
- `barplot()`
- data DataFrame

```
# 顧客ごとの合計購買金額を棒グラフで表示
# x軸に顧客ID、y軸に購買金額を指定
ax = sns.barplot(x=sales_per_customer.index, y='購買金額', data=sales_per_customer)
```

データを可視化するためのライブラリ

-
- sales_per_day
- `lineplot()`
- data DataFrame

```
# 日ごとの平均購買金額を折れ線グラフで表示
# data引数にsales_per_dayデータフレームを指定
ax = sns.lineplot(data=sales_per_day)
```

データを可視化するためのライブラリ

-
  - x
  - Matplotlib
  - `figure()`　　`figsize`

```
# グラフのサイズを指定（幅18、高さ4）
plt.figure(figsize=(18, 4))

# 日ごとの平均購買金額を折れ線グラフで表示
# data引数にsales_per_dayデータフレームを指定
ax = sns.lineplot(data=sales_per_day)
```

データを可視化するためのライブラリ

-
  - `set_title()`
  - `set()` `xlabel` `ylabel` x y

```
# グラフのサイズを指定（幅18、高さ4）
plt.figure(figsize=(18, 4))

# 日ごとの平均購買金額を折れ線グラフで表示
# data引数にsales_per_dayデータフレームを指定
ax = sns.lineplot(data=sales_per_day)

# グラフのタイトルを設定
ax.set_title('購買日ごとの平均売上')

# x軸とy軸のラベルを設定
ax.set(xlabel='購買日', ylabel='平均売上（円）')
```

課題

課題

- Moodle SCfCL-11-prac.ipynb Colab
- File > Download > Download .ipynb .ipynb
- **.ipynbファイル** Moodle
- **7月4日(木) 20時まで**