今日の内容

1. データ分析に必要な統計学の基礎を学ぶ
2. 比較して2変数の関係を考える
3. データに基づいて判断を下すための手法を学ぶ
4. ビジネスにおける予想と分析結果の報告

59

第2章の内容

1. クロス集計の軸設定と見方
2. 散布図と相関の調べ方
3. 相関係数と因果関係の違い
4. 時系列データの見方、分析方法

60

1.クロス集計の軸設定と見方

クロス集計とは

数量データの関係性を見る際は相関係数が利用できるが，カテゴリカルデータの関係性を見る際には**クロス集計**を用いるのが便利。分割表とも言う場合もある。
単純にはカテゴリごとの度数を記録した表，ただし，2つ以上の変数を対象とし，その組み合わせで度数を求める。

縦軸：表側（＝原因）どのような視点から調査結果を見たいか ※分析軸
横軸：表頭（＝結果）どのような結果を見たいか

61

1.クロス集計の軸設定と見方

CSVファイルのマウント

```
import numpy as np
import pandas as pd

from google.colab import files

uploaded = files.upload()
```

62

1.クロス集計の軸設定と見方

CSVファイルのマウント

```
import pandas as pd
import io

df = pd.read_csv('sample_cross.csv')
print(df)
```

```
    広告    購入 性別 年齢
0    B しなかった 男性 31
1    B しなかった 女性 28
2    A しなかった 女性 25
3    A しなかった 男性 31
4    B しなかった 男性 33
.. ..   ... .. ..
988  B しなかった 女性 27
989  B しなかった 男性 25
990  B しなかった 男性 32
991  B しなかった 女性 32
992  B しなかった 女性 28

[993 rows x 4 columns]
```

63

1.クロス集計の軸設定と見方

CSVファイルのマウント

```
n=len(df)
print(n)
df.head()
```

993

	広告	購入	性別	年齢
0	B	しなかった	男性	31
1	B	しなかった	女性	28
2	A	しなかった	女性	25
3	A	しなかった	男性	31
4	B	しなかった	男性	33

64

1.クロス集計の軸設定と見方

CSVファイルのマウント

```
ad_cross=pd.crosstab(df['広告'], df['性別'])
ad_cross
```

性別	女性	男性
広告		
A	200	199
B	297	297

65

1.クロス集計の軸設定と見方

CSVファイルのマウント

```
ad_cross=pd.crosstab(df['購入'], df['広告'])
ad_cross
```

広告	A	B
購入		
した	41	68
しなかった	358	526

66

1.クロス集計の軸設定と見方

CSVファイルのマウント

```
ad_cross=pd.crosstab(df['広告'], df['性別'], normalize=True)
ad_cross
```

性別	女性	男性
広告		
A	0.201410	0.200403
B	0.299094	0.299094

pandas.crosstab()
第一引数：indexに結果の行見出し
第二引数：columsに結果の列見出し
引数normalizeで全体・行ごと・列ごとに規格化（正規化）

pandas.DataFrame を返す

67

2.散布図と相関の調べ方

散布図とは

2つの要素からなる1組のデータが得られたときに，2つの要素の間にある関係（相関関係）を見るためのグラフ。因果関係（どちらかが原因となって，もう一方が起こる）を示すものではない。QC7つ道具の一つ。

例1）数学の点数と英語の点数：数学の点数が高い生徒は英語の点数も高い傾向がある（反対も含める）。
例2）高齢者の運動量と体力：運動量が増えると体力も増える（〃）。

68

2.散布図と相関の調べ方

相関分析

2つの要素（2変数）間の関係を数値で表現する分析方法。「相関」とは2つ以上の変数があるときに，「**どれぐらい類似しているのか**」という「類似度」を意味する。２つの変量の強弱を数値化したものを「相関係数」という。「類似度」の強さを「-1から1」までの範囲で表現される。正（または負）の相関関係が強いほど1（または-1）に近く，相関関係が弱いほど0に近くなる。

69

2.散布図と相関の調べ方

CSVファイルのマウント

```
from google.colab import files

uploaded = files.upload()
```

```
import pandas as pd
import io

df = pd.read_csv(io.BytesIO(uploaded['sample_covid19.csv']))
df
```

70

2.散布図と相関の調べ方

CSVファイルのマウント

```
[1] from google.colab import files

    uploaded = files.upload()

    ファイル選択 sample_covid19.csv
    • sample_covid19.csv(text/csv) - 42033 bytes, last modified: 2023/12/17 - 100% done
    Saving sample_covid19.csv to sample_covid19.csv

    import pandas as pd
    import io

    df = pd.read_csv(io.BytesIO(uploaded['sample_covid19.csv']))
    df
```

	Date	ALL	Hokkaido	Aomori	Iwate	Miyagi	Akita	Yamagata	Fukushima	Ibaraki	...	Ehime	Kochi	Fukuoka	Saga	Nagasaki	Kumamoto	Oi
0	2021/1/1	3249	98	10	4	30	3	3	12	42	...	6	6	158	1	20	30	
1	2021/1/2	3055	77	4	2	4	0	1	13	20	...	6	7	124	5	28	26	
2	2021/1/3	3136	68	10	3	20	3	6	14	52	...	7	11	104	3	30	22	
3	2021/1/4	3333	93	10	0	18	3	5	25	32	...	9	2	128	23	24	34	
4	2021/1/5	4936	79	7	6	37	8	5	25	67	...	25	7	187	10	55	63	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
268	2021/9/26	2146	54	14	2	16	0	1	9	47	...	10	0	88	4	8	14	
269	2021/9/27	1159	19	5	1	6	1	1	3	31	...	2	3	46	12	9	10	
270	2021/9/28	1743	26	25	0	12	6	1	3	25	...	10	4	43	8	5	17	
271	2021/9/29	1980	45	23	1	21	0	1	4	44	...	7	7	44	8	7	19	
272	2021/9/30	1571	26	23	0	14	2	1	5	20	...	14	3	37	6	6	15	

273 rows × 49 columns

71

2.散布図と相関の調べ方

CSVファイルのマウント

```
# 変数dfに格納したデータの何番目を
    取得するか指定
df.iloc[:,9]

df.iloc[:,14]
```

```
df['Ibaraki']

df['Tokyo']
```

```
df.iloc[:,9]
0    42
1    20
2    52
3    32
4    67
    ..
268  47
269  31
270  25
271  44
272  20
Name: Ibaraki, Length: 273, dtype: int64
```

```
df.iloc[:,14]
0     793
1     829
2     826
3     905
4    1315
     ...
268   302
269   155
270   250
271   268
272   219
Name: Tokyo, Length: 273, dtype: int64
```

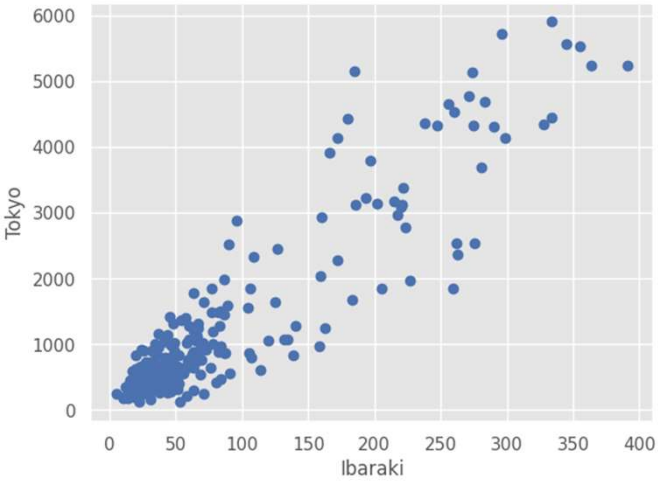
72

2.散布図と相関の調べ方

プロット（東京都と茨城県の新規感染者数）

```
%matplotlib inline
# グラフツール matplotlibをインポート
import matplotlib.pyplot as plt
# カラーリングするツールをインポート
import warnings

#xlabel、ylabelはX軸、Y軸を指定する
plt.style.use('ggplot')
plt.plot(df.iloc[:,9],
df.iloc[:,14],'bo')
plt.xlabel('Ibaraki')
plt.ylabel('Tokyo')
plt.show()
```



73

2.散布図と相関の調べ方

プロット（東京都と茨城県の新規感染者数）：相関係数

```
y_colum='Tokyo'

# 全体の相関係数を計算し、corrの
matrix上に表示させる
corr_matrix = df.corr()
# 変数y_corrにy_colum(つまりTokyo)
との相関係数を格納
y_corr = corr_matrix[y_colum]
y_corr
```

```
ALL          0.933407
Hokkaido     0.562570
Aomori       0.597597
Iwate        0.720486
Miyagi       0.694157
Akita        0.658737
Yamagata     0.599073
Fukushima    0.857339
Ibaraki      0.919476
Tochigi      0.916396
Gunma        0.889860
Saitama      0.959624
Chiba        0.911269
Tokyo        1.000000
Kanagawa     0.932625
Niigata      0.853796
Toyama       0.835855
Ishikawa     0.793950
Fukui        0.736283
Yamanashi    0.875588
Nagano       0.794002
Gifu         0.681034
Shizuoka     0.848566
Aichi        0.634805
Mie          0.738161
Shiga        0.838877
Kyoto        0.852544
Osaka        0.757124
Hyogo        0.740625
Nara         0.711916
Wakayama     0.751228
Tottori      0.797398
```

```
Ishikawa     0.793950
Fukui        0.736283
Yamanashi    0.875588
Nagano       0.794002
Gifu         0.681034
Shizuoka     0.848566
Aichi        0.634805
Mie          0.738161
Shiga        0.838877
Kyoto        0.852544
Osaka        0.757124
Hyogo        0.740625
Nara         0.711916
Wakayama     0.751228
Tottori      0.797398
Shimane      0.708764
Okayama      0.744315
Hiroshima    0.665415
Yamaguchi    0.670897
Tokushima    0.490078
Kagawa       0.824499
Ehime        0.733994
Kochi        0.660480
Fukuoka      0.882209
Saga         0.786857
Nagasaki     0.789520
Kumamoto     0.856704
Oita         0.722250
Miyazaki     0.761319
Kagoshima    0.828517
Okinawa      0.893809
Name: Tokyo, dtype: float64
```

74

2.散布図と相関の調べ方

プロット（東京都と茨城県の新規感染者数）：相関係数

```
y_colum='Tokyo'

corr_matrix = df.corr()['Ibaraki']
y_corr = corr_matrix[y_colum]
y_corr
```

```
0.9194759486275046
```

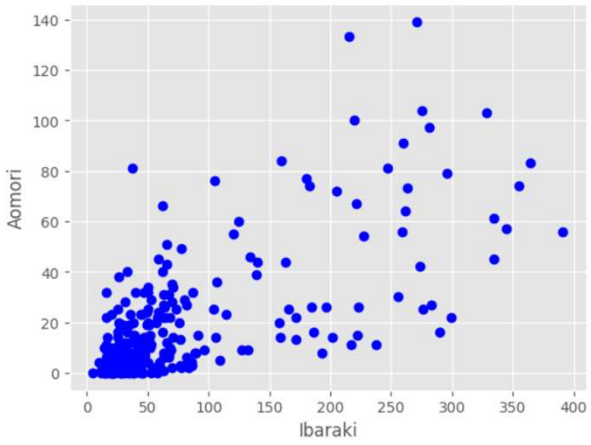
75

2.散布図と相関の調べ方

プロット（青森県と茨城県の新規感染者数）

```
plt.style.use('ggplot')
plt.plot(df.iloc[:,9],
df.iloc[:,3],'bo')
plt.xlabel('Ibaraki')
plt.ylabel('Aomori')
plt.show()
```

```
0.6844695489050351
```



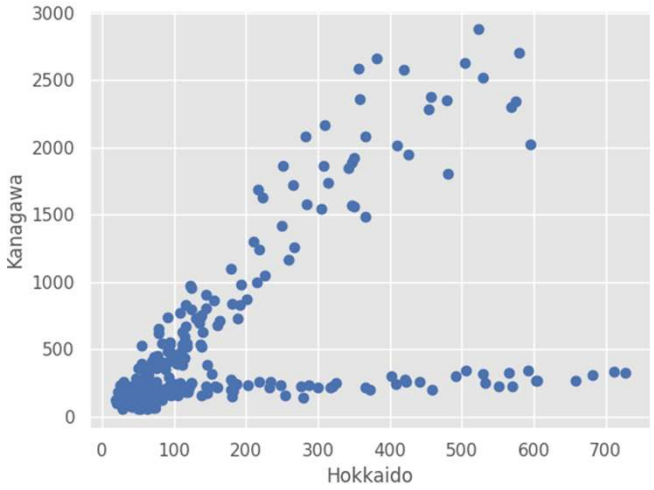
76

2.散布図と相関の調べ方

プロット（神奈川県と北海道の新規感染者数）

```
plt.style.use('ggplot')
plt.plot(df.iloc[:,2],
df.iloc[:,15],'bo')
plt.xlabel('Hokkaido')
plt.ylabel('Kanagawa')
plt.show()
```

```
0.5503568815983438
```

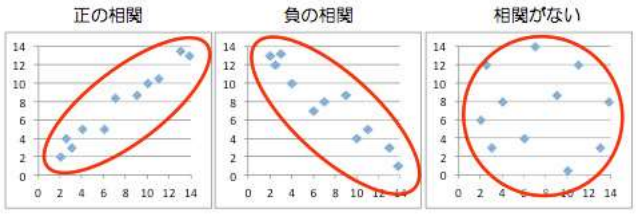


77

3.相関係数と因果関係の違い　※セクション3の紹介

相関係数

２つの変量の強弱を数値化したものを「相関係数」という。
「類似度」の強さを「-1から1」までの範囲で表現される。
正（または負）の相関関係が強いほど1（または-1）に近く，
相関関係が弱いほど0に近くなる。



なるほど統計学園：https://www.stat.go.jp/naruhodo/10_tokucho/hukusu.html

78

2.散布図と相関の調べ方- iris dataset -

irisデータセット

```
import pandas as pd
import seaborn as sns
sns.set()
iris = sns.load_dataset('iris')
titanic = sns.load_dataset('titanic')
```

79

2.散布図と相関の調べ方- iris dataset -

データセットの利用

```
print(iris.head())
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

80

2.散布図と相関の調べ方- iris dataset -

データセットの利用

```
print(titanic.head())
```

```
   survived  pclass     sex   age  sibsp  parch     fare embarked  class
0         0       3    male  22.0      1      0   7.2500        S  Third
1         1       1  female  38.0      1      0  71.2833        C  First
2         1       3  female  26.0      0      0   7.9250        S  Third
3         1       1  female  35.0      1      0  53.1000        S  First
4         0       3    male  35.0      0      0   8.0500        S  Third

     who  adult_male deck  embark_town alive  alone
0    man        True  NaN  Southampton    no  False
1  woman       False    C    Cherbourg   yes  False
2  woman       False  NaN  Southampton   yes   True
3  woman       False    C  Southampton   yes  False
4    man        True  NaN  Southampton    no   True
```

81

2.散布図と相関の調べ方- iris dataset -

データセットの利用

```
print(iris.head())
print(iris.info())
print(iris.shape)
print(iris.ndim)
print(iris.columns)
```

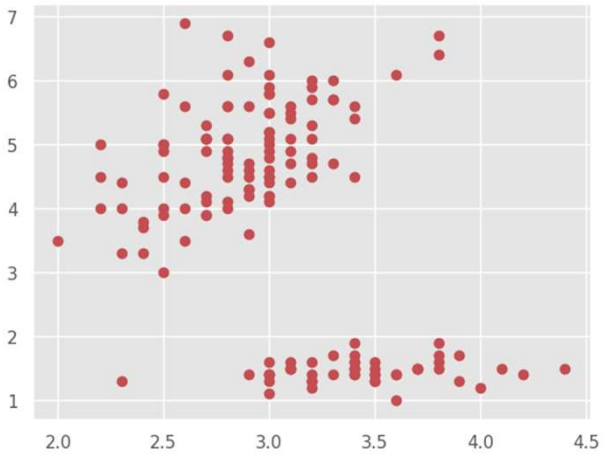
```
   sepal_length  sepal_width  petal_length  petal_width species
0           5.1          3.5           1.4          0.2  setosa
1           4.9          3.0           1.4          0.2  setosa
2           4.7          3.2           1.3          0.2  setosa
3           4.6          3.1           1.5          0.2  setosa
4           5.0          3.6           1.4          0.2  setosa
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
None
(150, 5)
2
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

82

2.散布図と相関の調べ方- iris dataset -

データセットの利用

```
%matplotlib inline
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
plt.style.use('ggplot')
plt.plot(iris.iloc[:,1], iris.iloc[:,2],
'ro')
plt.show()
```



83

2.散布図と相関の調べ方- iris dataset -

データセットの利用

```
iris.corr()
```

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.117570	0.871754	0.817941
sepal_width	-0.117570	1.000000	-0.428440	-0.366126
petal_length	0.871754	-0.428440	1.000000	0.962865
petal_width	0.817941	-0.366126	0.962865	1.000000

84

2.散布図と相関の調べ方- iris dataset -

データセットの利用

```
y_colum='sepal_width'

corr_matrix = iris.corr()
y_corr = corr_matrix[y_colum]
y_corr
```

```
sepal_length  -0.117570
sepal_width    1.000000
petal_length  -0.428440
petal_width   -0.366126
Name: sepal_width, dtype: float64
```

85

2.散布図と相関の調べ方- iris dataset -

データセットの利用

```
print(iris.describe())
```

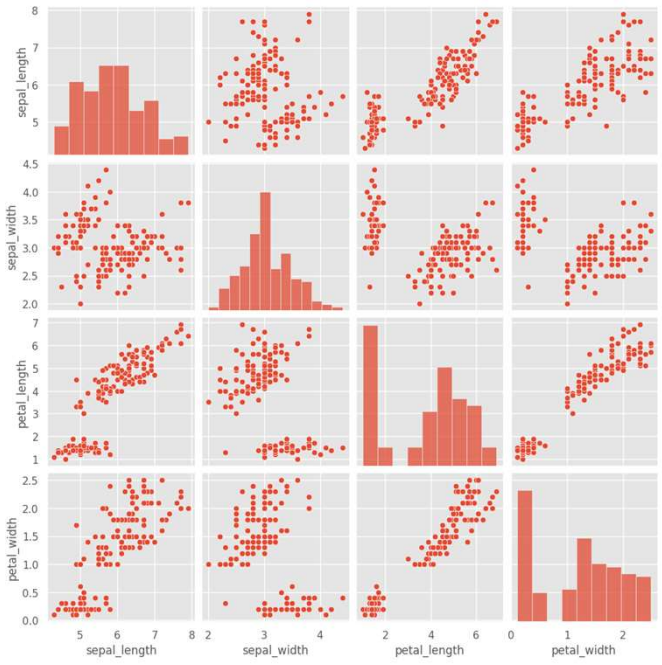
```
       sepal_length  sepal_width  petal_length  petal_width
count    150.000000   150.000000    150.000000   150.000000
mean       5.843333     3.057333      3.758000     1.199333
std        0.828066     0.435866      1.765298     0.762238
min        4.300000     2.000000      1.000000     0.100000
25%        5.100000     2.800000      1.600000     0.300000
50%        5.800000     3.000000      4.350000     1.300000
75%        6.400000     3.300000      5.100000     1.800000
max        7.900000     4.400000      6.900000     2.500000
```

86

2.散布図と相関の調べ方 - iris dataset -

```
pg=sns.pairplot(iris)
plt.show(pg)
```

87

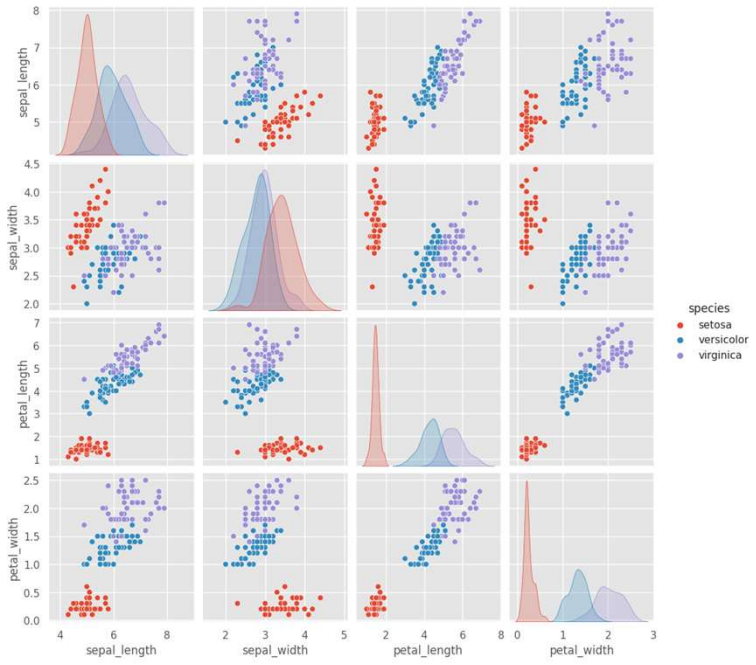


88

2.散布図と相関の調べ方 - iris dataset -

```
pg=sns.pairplot(iris,hue='species')
plt.show(pg)
```

89

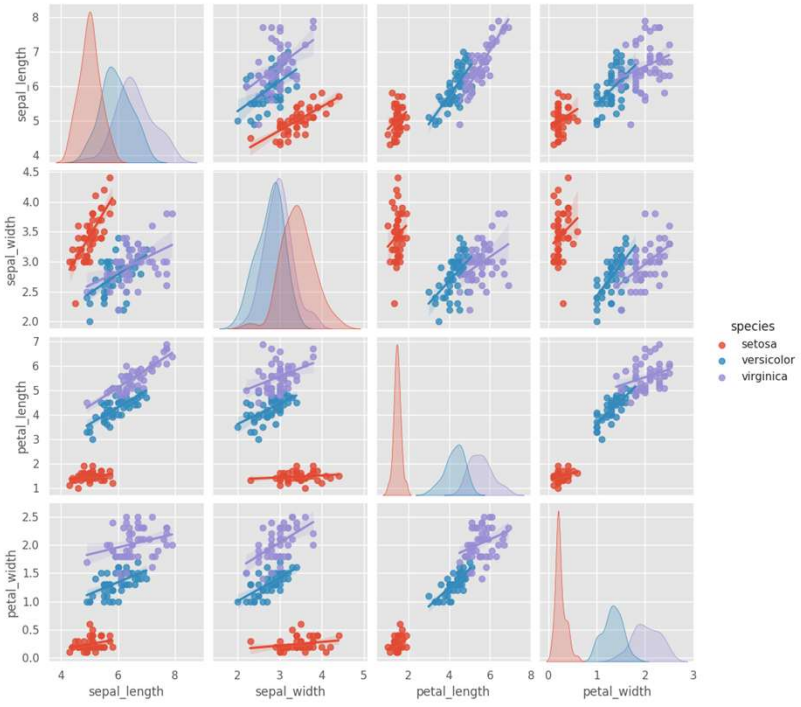


90

2.散布図と相関の調べ方 - iris dataset -

```
pg=sns.pairplot(iris,hue='species', kind='reg')
plt.show(pg)
```

91

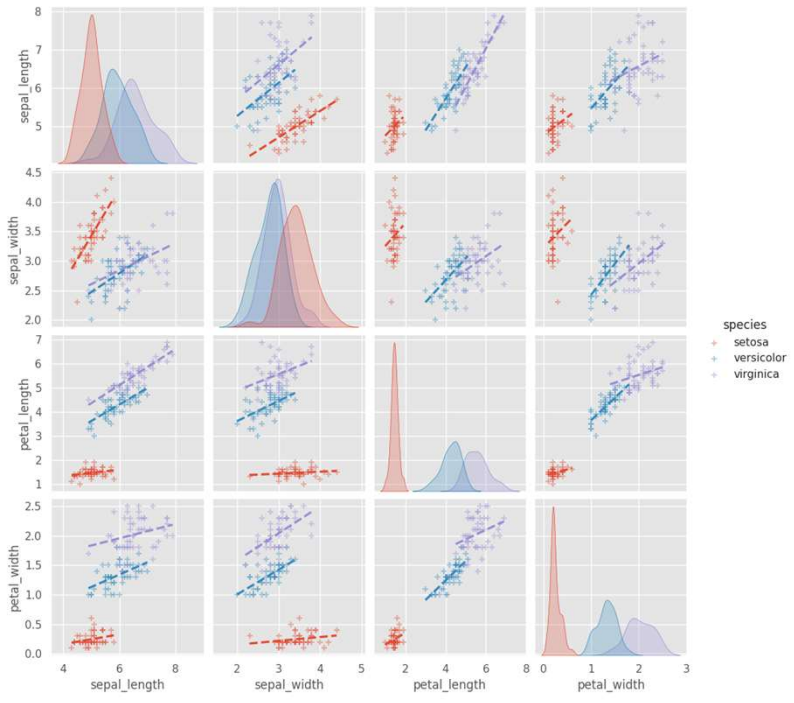


92

2.散布図と相関の調べ方 - iris dataset -

```
pg=sns.pairplot(iris,hue='species', kind='reg' ,plot_kws={'ci':
None,'marker': '+','scatter_kws': {'alpha': 0.4},'line_kws':
{'linestyle': '--'}})
plt.show(pg)
```

93

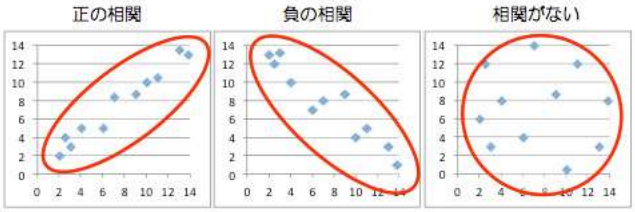


94

3.相関係数と因果関係の違い

相関係数

２つの変量の強弱を数値化したものを「相関係数」という。「類似度」の強さが「-1から1」までの範囲で表現される。正（または負）の相関関係が強いほど1（または-1）に近く，相関関係が弱いほど0に近くなる。



なるほど統計学園：https://www.stat.go.jp/naruhodo/10_tokucho/hukusu.html

95

3.相関係数と因果関係の違い

因果関係

「原因とそれによって生じる結果との関係」（広辞苑、第6版）を**因果関係**という。要因とアウトカムの間において、関連はみられるが要因が結果を導く関係（真の因果関係）になっていないこともあるため、その判断には注意が必要である。

（一般社団法人日本疫学会：https://jeaweb.jp/glossary/glossary015.html）

相関関係：AとBの事柄になんらかの関連性があるもの
因果関係：Aを原因としてBが変動すること

96

4.時系列データの見方、分析方法

時系列データ

時間的な順序をともないながら観測されるデータのこと。
ある一定の時間間隔で観測されたデータや、イベントが発生した時刻・頻度などが含まれる。

時系列解析：時系列データに潜む傾向や特徴を把握したり，時系列データの将来の値を予測したりする際に有効な技術

例）株価データ，天気予報の気温や降水確率などの気象データ，
人口統計データ，センサーデータ，販売数データ，など

97

4.時系列データの見方、分析方法

時系列データがもつ情報

- 長期（傾向）変動（トレンド）
  時系列の長期的傾向、時間の経過とともに増加・減少する傾向
- 循環変動（サイクル）
  傾向変動より短期的で，周期的に繰り返される変動
- 季節変動（シーズナル）
  （通常）1年を周期とする規則的な変動
- 不規則変動（ノイズ）
  上記の変動で説明できないような，短期的かつ不規則な変動

98

4.時系列データの見方、分析方法

CSVファイルのマウント

```
from google.colab import files

uploaded = files.upload()
```

```
import pandas as pd
import io
                                        ※Date列を日付データとして解析する
df = pd.read_csv(io.BytesIO(uploaded['sample_covid19.csv']) ,
parse_dates=['Date'])
df
```

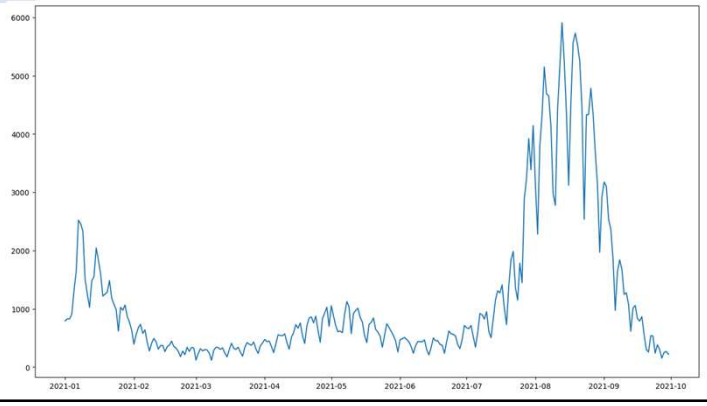
99

4.時系列データの見方、分析方法

折れ線グラフで表現

```
import matplotlib.pyplot as plt

plt.plot(df['Date'], df['Tokyo'])
```

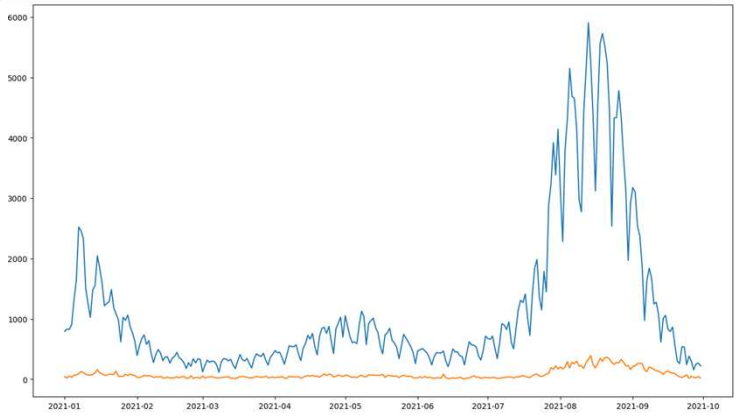


100

4.時系列データの見方、分析方法

折れ線グラフで表現

```
plt.plot(df['Date'], df['Tokyo'])
# 要素を追加すると、棒グラフが重なって表示される
plt.plot(df['Date'], df['Ibaraki'])
```



101

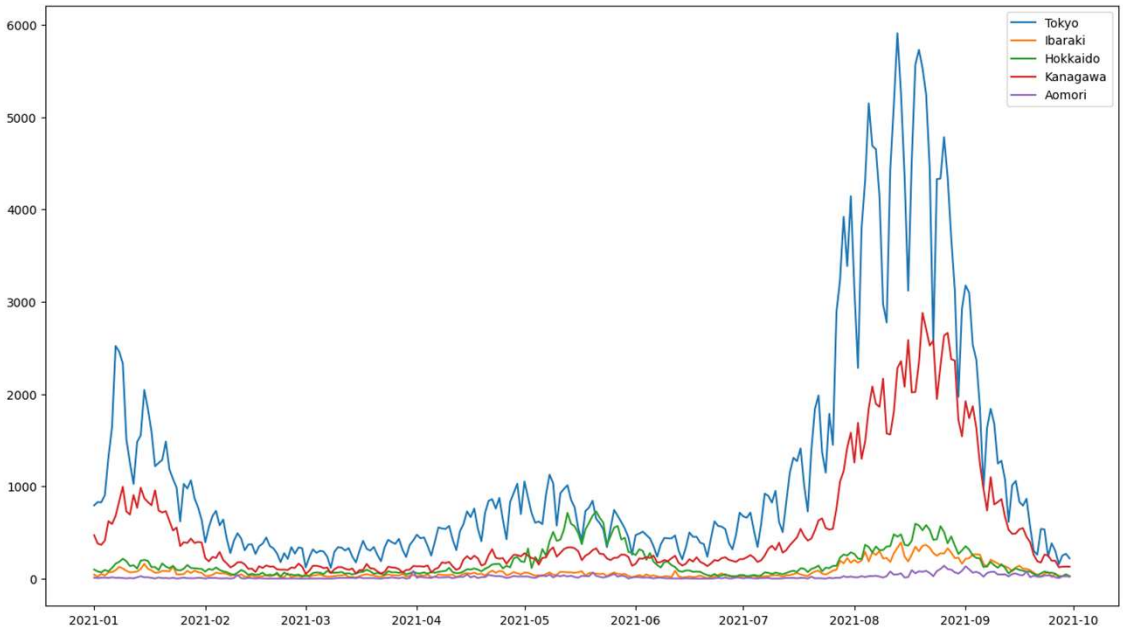
4.時系列データの見方、分析方法

折れ線グラフで表現

```
plt.plot(df['Date'], df['Tokyo'], label='Tokyo')
plt.plot(df['Date'], df['Ibaraki'], label='Ibaraki')
plt.plot(df['Date'], df['Hokkaido'], label='Hokkaido')
plt.plot(df['Date'], df['Kanagawa'], label='Kanagawa')
plt.plot(df['Date'], df['Aomori'], label='Aomori')
plt.legend()
```

※要素にラベルを作成し，凡例に反映させる

102



103

今日の内容

1 データ分析に必要な統計学の基礎を学ぶ

2 比較して2変数の関係を考える

3 **データに基づいて判断を下すための手法を学ぶ**

4 ビジネスにおける予想と分析結果の報告

104