訓練データと検証データ

インポートするファイル

- import numpy as np
- import matplotlib.pyplot as plt
- import seaborn as sns
- sns.set()
- from sklearn.model_selection import train_test_split
- from statsmodels.stats.outliners_influence import variance_inflation_factor

データの読み込み

pandasを使ってcsvファイルの読み込みを行う

```
raw_data = pd.read_csv('ファイル名')
```

読み込んだファイルの確認

```
raw_data
```

読み込んだファイルの最初の方を確認

```
raw_data.head()
```

読み込んだデータの整理

記述統計量を確認する。

```
raw_data.descrive(include='all')
```

検証に用いないデータの列の削除

```
data = raw_data.drop(columns=['列タイトル'])
```

欠損値のあるデータの確認

```
data.isnull().sum()
```

欠損値のあるデータの行の削除（元データの５％以内の量であれば問題ないと思われる。

```
data = data.dropna()
```

データの形状の確認

データが標準正規分布にしたがっていない場合の処理

distplotメソッドを使用してデータの形状を確認する
　　sns.distplot(data['列タイトル'])

quantileメソッドを使用してデータの外れ値を削除する
　　q = data['列タイトル'].quantile(0～1の間の数値)
　　data1 = data[data['列タイトル']<q]

data1をdata2と変数を更新していき、上記の内容をすべての列で行う

データのインデックスをつけなおして、data_cleanedに格納する
　　data_cleaned = data2.reset_index(drop=True)

データを確認する。
data_cleaned.descrive(include='all')

データの形状の確認

従属変数と独立変数の関係が線形でない場合の処理

scatterを使って散布図を確認する。

```
ax = plt.subplots(1, 3, sharey=True, figsize=(15,3))
ax.scatter(data_cleaned['独立変数名'],data_cleaned['従属変数名']
```

指数分布だった場合は対数変換をする。

```
log_data = np.log(data_cleaned['従属変数名'])
data_cleaned['log_data'] = log_data
```

多重共線性の確認

sklearnには多重共線性を確認するメソッドがないためstatsmodelライブラリを使用する。
from statsmodels.stats.outliners_influence import variance_inflation_factor

variables = data_cleaned[['独立変数名']]
vif = pd.DataFrame()
vif['features'] = variables.columns
vif['VIF'] = [variance_inflation_factor(variables.values, i) for i in range(variables.shape[1])]

多重共線性が強いと言われる基準はあいまいなので、ここには記述しない。
多重共線性が強いと思われる物については削除する。

```
data = raw_data.drop(columns=['列タイトル'])
```

データを二つに分ける

訓練データ(train)と検証データ(test)でデータに分割する。

```
train, test = train_test_split('分割したいデータ')
```

※1　デフォルトでは75:25で分割される。割合を変更したい場合は引数を指定する。

```
test_size = 割合
```

※2　データはランダムに分割される。ランダム要素を固定したい場合は次の引数を指定する。(365は例)

```
random_state = 365
```