Student No ;71701268
CNS account : s17126su
Name : 有働　峻介

#Python で進めました
#Python3 環境..JupyterLab

```
import pandas as pd
from urllib import request
from bs4 import BeautifulSoup
import numpy as np
#課題１
#下記のテーブル内容を取得し、「URL」を引数にとりテーブルをデータフレームとして出力する関数を作成して下さい。(コードを貼って下さい)
def scrR(url):
    html = request.urlopen(url)
    soup = BeautifulSoup(html, 'html.parser')

    td = soup.findAll("td")
    th = soup.findAll("th")

    th_s = []
    td_s = []

    for i in range(len(th)):
      th_s.append(th[i].string)
    for i in range(len(td)):
      td_s.append(td[i].string)

    td_s = np.reshape(td_s, (120, 45)) # ここまだ一般化できる余地あり

    return pd.DataFrame(td_s, columns = th_s)
```

```
df = scrR("http://web.sfc.keio.ac.jp/~sk/ynl-r-homework.html")

#課題2
#課題1で出力されたデータフレームを 「日時」「天気」「気温」「アトラクション名」「待ち時間」
#の５つのカラムで構成されるデータフレームに変換して下さい。(コードを貼って下さい)


df_2 = scrR("http://web.sfc.keio.ac.jp/~sk/ynl-r-homework.html")


df_changed = df_2[df_2.columns[3:45]]

#アトラクション名が入った配列を作成
atractions = []
for i in range(42):
    atr = [df_2.columns[i + 3]]*120
    atractions.extend(atr)
    #長さ120*42ね

#待ち時間が入った配列を作成
wait = []
for i in range(42):
    con = df_changed[df_changed.columns[i]]
    wait.extend(con)

#いらんとこ削除
for i in range(42):
    del df_2[df_2.columns[3]]
```

```
#atractions[], wait[]と合うようにdfの長さを調節
df_a = df_2

for i in range(41):
    df_2 = df_2.append(df_a)



#アトラクション名と待ち時間の配列をcancat()使って元のdfとくっつける
#元のdfじゃないやいらんとこ削除したやつや
df_2["アトラクション名"] = atractions
df_2["待ち時間"] = wait
#課題3
#「ジャングルクルーズ」「プーさんのハニーハント」「ミニーの家」の待ち時間のヒストグラムを作図して下さい。
#なお、平均と1SD値に補助線も引いて下さい。
#そもそも、、、度数・階級
#階級...今回は待ち時間

# なんか半透明に3つ並べてグラフを作れるらしいから余力あったらそれも試そう

# ４.ジャングルクルーズ
# １４.プーさんのハニーハント
# ２９.ミニーの家

import matplotlib.pyplot as plt

df_3 = scrR("http://web.sfc.keio.ac.jp/~sk/ynl-r-homework.html")

# 行/列指定でアトラクション別の待ち時間の値を引っ張ってくる
atr_jun = df_3["４.ジャングルクルーズ"].values
```

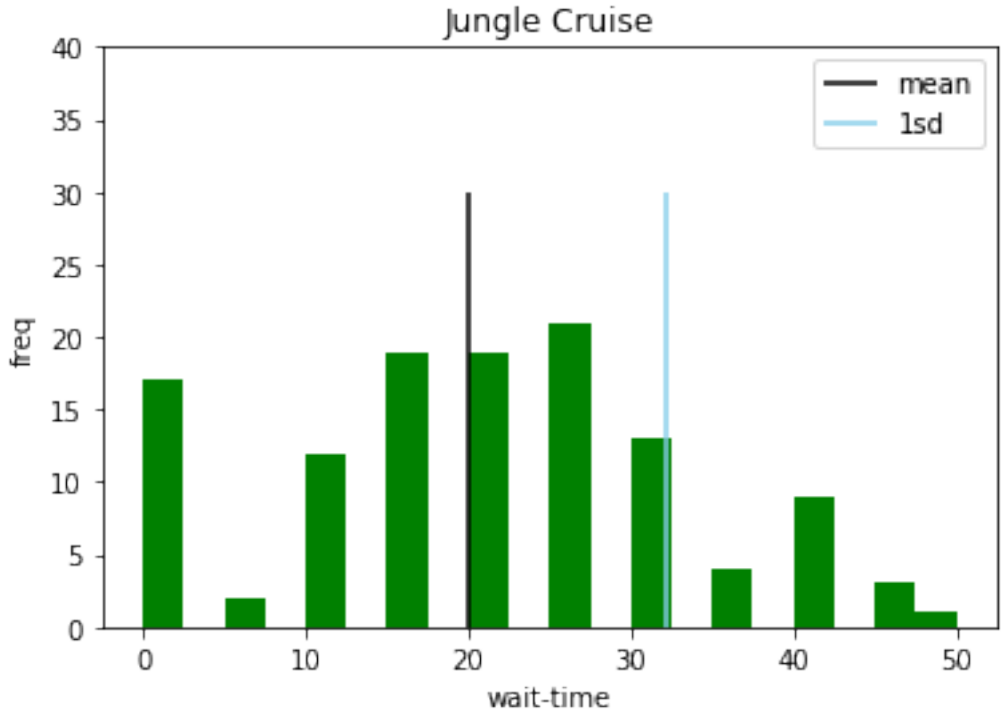
```
atr_pooh = df_3["１４.プーさんのハニーハント"].values
atr_minnie = df_3["２９.ミニーの家"].values

# ndarrayのままだとhist()との互換性がよろしくないのでlistに直します
atr_jun = atr_jun.tolist()
atr_pooh = atr_pooh.tolist()
atr_minnie = atr_minnie.tolist()



# よくわからない点...5分待ちがうまく表示されない。
# listの中が全部文字列だったから数値に直します
# 補助線引く観点でもこっちのがいいや
atr_jun = [0 if n =='-' else int(n) for n in atr_jun]
atr_pooh = [0 if n == '-' else int(n) for n in atr_pooh]
atr_minnie = [0 if n == '-' else int(n) for n in atr_minnie]

#平均値、1SD値をセット
jun_mean = np.mean(atr_jun)
pooh_mean = np.mean(atr_pooh)
minnie_mean = np.mean(atr_minnie)

jun_1sd = jun_mean + np.std(atr_jun)
pooh_1sd = pooh_mean + np.std(atr_pooh)
minnie_1sd = minnie_mean + np.std(atr_minnie)



# figure をセット
fig_jun = plt.figure().add_subplot(1, 1, 1)
fig_pooh = plt.figure().add_subplot(1, 1, 1)
fig_minnie = plt.figure().add_subplot(1, 1, 1)
```

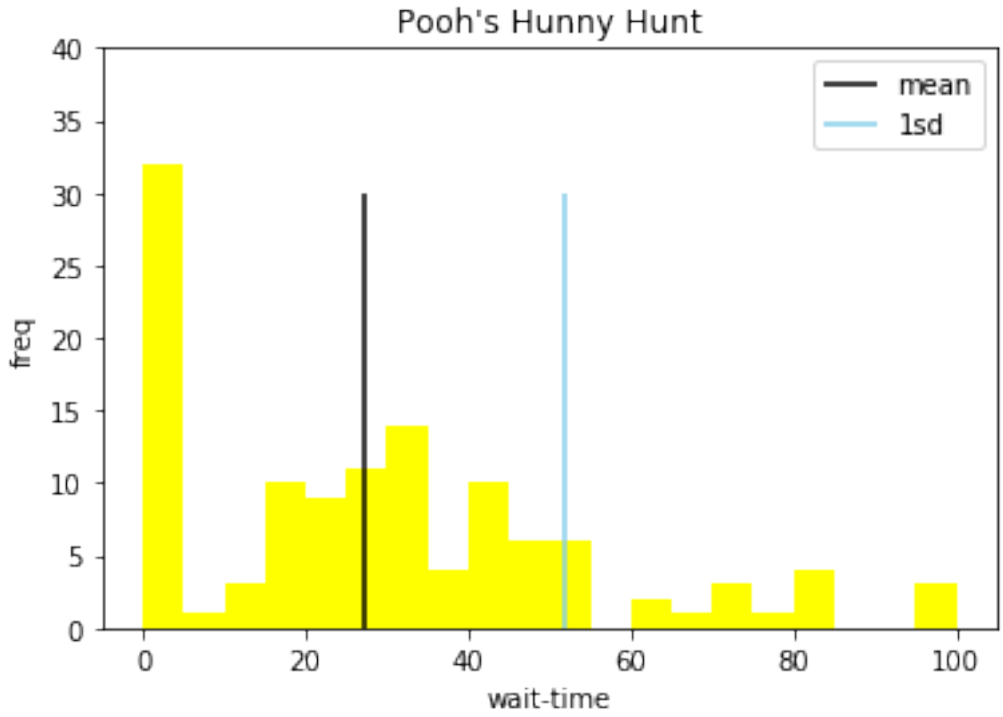
```
# ジャングルクルーズ
fig_jun.hist(atr_jun, bins=20, color="green")
fig_jun.set_title('Jungle Cruise')
fig_jun.set_xlabel('wait-time')
fig_jun.set_ylabel('freq')
fig_jun.set_ylim([0,40])
fig_jun.vlines(x=jun_mean,ymin=0, ymax=30, label="mean")
fig_jun.vlines(x=jun_1sd, ymin=0, ymax=30, color="skyblue",
label="1sd")
fig_jun.legend(loc='upper right')
```

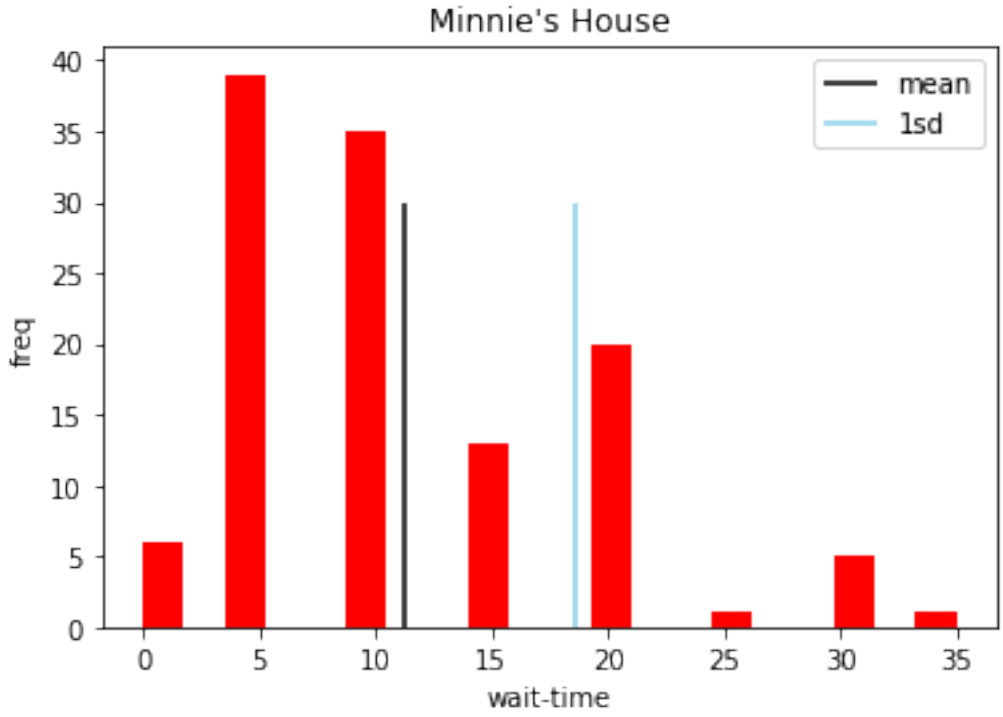
```
# プーさんのハニーハント
fig_pooh.hist(atr_pooh, bins=20, color="yellow")
fig_pooh.set_title("Pooh's Hunny Hunt")
fig_pooh.set_xlabel('wait-time')
fig_pooh.set_ylabel('freq')
fig_pooh.set_ylim([0, 40])
fig_pooh.vlines(x=pooh_mean,ymin=0, ymax=30, label="mean")
fig_pooh.vlines(x=pooh_1sd, ymin=0, ymax=30, color="skyblue",
label="1sd")
fig_pooh.legend(loc='upper right')
```

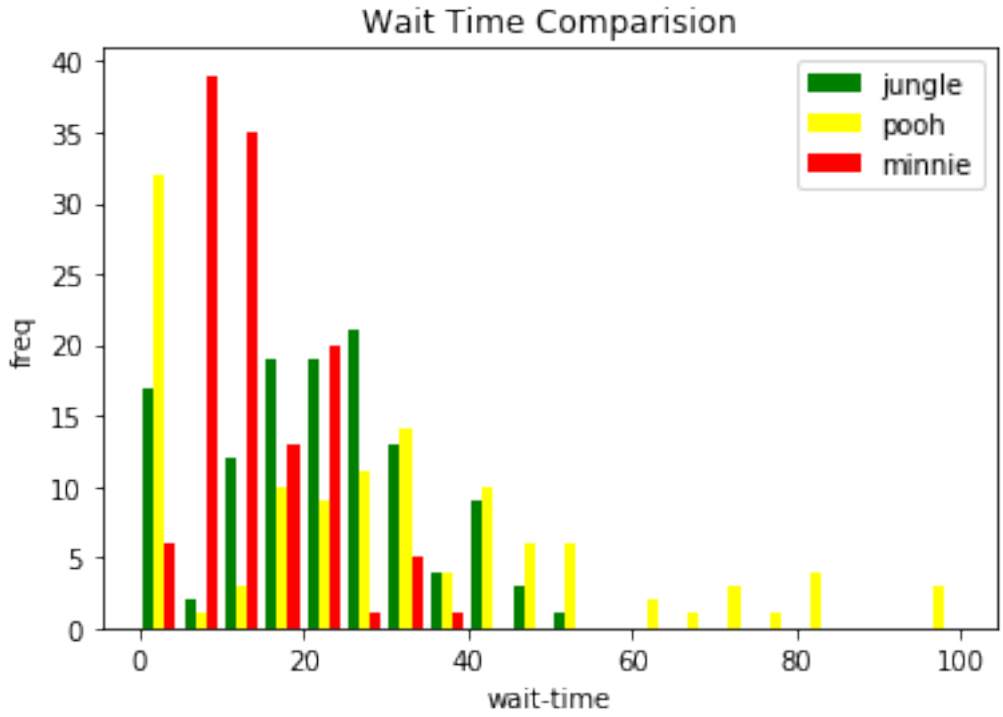
```
# ミニーの家　これが度数の最大値を取ったので比較しやすいように他もそれ
に合わせる（４０）
fig_minnie.hist(atr_minnie, bins=20, color="red")
fig_minnie.set_title("Minnie's House")
fig_minnie.set_xlabel('wait-time')
fig_minnie.set_ylabel('freq')
fig_minnie.vlines(x=minnie_mean,ymin=0, ymax=30, label="mean")
```

```
fig_minnie.vlines(x=minnie_1sd, ymin=0, ymax=30, color="skyblue",
label="1sd")
fig_minnie.legend(loc='upper right')

# 全部のやつ（いちばん直感的に見れるのかな）
fig_all = plt.figure().add_subplot(1, 1, 1)
fig_all.hist([atr_jun, atr_pooh, atr_minnie], bins=20, color=['green',
'yellow', 'red'], label=['jungle', 'pooh', 'minnie'])
fig_all.legend(loc='upper right')
fig_all.set_title("Wait Time Comparision")
fig_all.set_xlabel('wait-time')
fig_all.set_ylabel('freq')

# この後のために関数化したいなぁ
def graphize(atr_name):

    df = scrR("http://web.sfc.keio.ac.jp/~sk/ynl-r-homework.html")
    atr = df[atr_name].values
    atr = atr.tolist()
    atr = [0 if n =='-' else int(n) for n in atr]
    fig = plt.figure().add_subplot(1, 1, 1)
    fig.hist(atr, bins=20, color="green")
```


Jungle Cruise
mean
1sd
freq
wait-time
0
5
10
15
20
25
30
35
40
0
10
20
30
40
50


Pooh's Hunny Hunt
mean
1sd
freq
wait-time
0
5
10
15
20
25
30
35
40
0
20
40
60
80
100


Minnie's House
mean
1sd
freq
wait-time
0
5
10
15
20
25
30
35
40


Wait Time Comparision
jungle
pooh
minnie
freq
wait-time
0
20
40
60
80
100

#課題 4. 「ビッグサンダーマウンテン」の待ち時間を予測する重回帰モデルを作成し、5 月 26 日の待ち時間を予測して下さい。
#なお、予測に用いた説明変数も理由と共に明記して下さい。 5 月 26 日は気温：22.8、晴です。

```
from sklearn import linear_model



def multi_re(ob_tmp, ob_clm, atr_name):


    cls = linear_model.LinearRegression()

    df_4 = scrR("http://web.sfc.keio.ac.jp/~sk/ynl-r-homework.html")

    if ob_clm =="晴":
        ob_clm_int = 1
    elif ob_clm == "曇":
        ob_clm_int = 2
    elif ob_clm == "小":
        ob_clm_int = 3
    elif ob_clm == "雨":
        ob_clm_int = 4
    else:
        ob_clm_int = 0



    # 説明変数に気温と天気を使用
```

時間帯/日付（曜日）も説明変数に含むアルゴリズムの構築を試みたが、アイデア、技術力が追いつかず今回は断念。

配列（特にDataFrame, Series）についての深い考察、知識武装を行い再挑戦する見込み。

```
tmp = df_4.loc[:, ['気温']].as_matrix()
tmp[119] = 0

for n in range(len(tmp)):
    tmp[n] = float(tmp[n])



clm = df_4.loc[:, ["天気"]].as_matrix()
# 天気を数値化
for i in range(len(clm)):
    if clm[i] == "晴":
        clm[i] = 1
    elif clm[i] == "曇":
        clm[i] = 2
    elif clm[i] == "小":
        clm[i] = 3
    elif clm[i] == "雨":
        clm[i] = 4
    else:
        clm[i] = 0
```

```
    # 目的変数に待ち時間を使用
    atr = df_4[atr_name].as_matrix()
    atr = [0 if n =='-' else int(n) for n in atr]

    var = np.hstack([tmp, clm])

    cls.fit(var, atr)

    coef = cls.coef_
    intercept = cls.intercept_



    atr_wait = coef[0]*ob_tmp + coef[1]*ob_clm_int + intercept
    print("気温:" + str(ob_tmp) + " 天気:" + str(ob_clm)  + " " +
atr_name + "の予測待ち時間は" + str(atr_wait))
```

```
multi_re(ob_tmp=22.8, ob_clm="晴", atr_name="７.ビッグサンダー・マウンテン")
```

```
気温:22.8 天気:晴  ７.ビッグサンダー・マウンテンの予測待ち時間は
76.74172321039022
```

#課題5. 5/26 15:30に乗るべきアトラクションは何ですか？乗るべきでないアトラクションは何ですか？

```
df_5 = scrR("http://web.sfc.keio.ac.jp/~sk/ynl-r-homework.html")

# それぞれで分析、待ち時間を算出。その後最低のものを選択
```

```
for i in range(42):
    multi_re(ob_tmp=22.8, ob_clm="晴", atr_name=df_5.columns[i + 3])
```

#下記の解析結果より

#５.スイスファイミリーツリーハウス
#２８.ドナルドのボート
#３５.グランドサーキット・レースウェイ
#３８.エントランスグリ（ベビーカー、車椅子レンタル側）
#上記４アトラクションが最短で乗ることができる。故に乗るべきである。

#１２.スプラッシュ・マウンテン の待ち時間が最長で約 120 分。故に乗るべきでない。

気温:22.8 天気:晴 １.オムニバスの予測待ち時間は 1.5092550414067367
気温:22.8 天気:晴 ２.ウエスタンリバー鉄道の予測待ち時間は 24.010018501595063
気温:22.8 天気:晴 ３.カリブの海賊の予測待ち時間は 21.364106319163618
気温:22.8 天気:晴 ４.ジャングルクルーズの予測待ち時間は 20.012869403874497
気温:22.8 天気:晴 ５.スイスファイミリーツリーハウスの予測待ち時間は 0.0
気温:22.8 天気:晴 ６.魅惑のチキルーム：スティッチ・プレゼンツアロハ・エ・コモ・マイ！の予測待ち時間は 9.120071628974273
気温:22.8 天気:晴 ７.ビッグサンダー・マウンテンの予測待ち時間は 76.74172321039022
気温:22.8 天気:晴 ８.ウエスタンランド・シューティングギャラリーの予測待ち時間は 14.50200765265109
気温:22.8 天気:晴 ９.カントリーベア・シアターの予測待ち時間は 4.800329979343313

気温:22.8 天気:晴 １０.トムソーヤ島いかだの予測待ち時間は
4.806764047086098
気温:22.8 天気:晴 １１.蒸気船マークトウェイン号の予測待ち時間は
8.414250892224995
気温:22.8 天気:晴 １２.スプラッシュ・マウンテンの予測待ち時間は
120.19420461590511
気温:22.8 天気:晴 １３.ビーバーブラザーズのカヌー探険の予測待ち時間は
27.49143442746109
気温:22.8 天気:晴 １４.プーさんのハニーハントの予測待ち時間は
44.75774267344663
気温:22.8 天気:晴 １５.ホーンテッドマンションの予測待ち時間は
64.05533616187319
気温:22.8 天気:晴 １６.アリスのティーパーティーの予測待ち時間は
43.903970618753604
気温:22.8 天気:晴 １７.イッツ・ア・スモールワールドの予測待ち時間は
12.611765146728349
気温:22.8 天気:晴 １８.キャッスルカルーセルの予測待ち時間は
17.212195631601176
気温:22.8 天気:晴 １９.シンデレラのフェアリーテイル・ホールの予測待ち
時間は 24.762058121401296
気温:22.8 天気:晴 ２０.ピノキオの冒険旅行の予測待ち時間は
20.46554050942258
気温:22.8 天気:晴 ２１.ピーターパン空の旅の予測待ち時間は
38.40050292387835
気温:22.8 天気:晴 ２２.ミッキーのフィルハーマジックの予測待ち時間は
16.256340370538872
気温:22.8 天気:晴 ２３.白雪姫と七人のこびとの予測待ち時間は
24.688282799767464
気温:22.8 天気:晴 ２４.空飛ぶダンボの予測待ち時間は 40.24123353825321
気温:22.8 天気:晴 ２５.ガジェットのゴーコースターの予測待ち時間は
20.290737366279956

気温:22.8 天気:晴 ２６.グーフィーのペイント＆プレイハウスの予測待ち時間は 17.383993819255
気温:22.8 天気:晴 ２７.チップとデールのツリーハウスの予測待ち時間は 0.10656957362000927
気温:22.8 天気:晴 ２８.ドナルドのボートの予測待ち時間は 0.0
気温:22.8 天気:晴 ２９.ミニーの家の予測待ち時間は 15.85883163076532
気温:22.8 天気:晴 ３０.ロジャーラビットのカートゥーンスピンの予測待ち時間は 32.46964245990076
気温:22.8 天気:晴 ３１.スター・ツアーズ：ザ・アドベンチャーズ・コンティニューの予測待ち時間は 27.08201070169717
気温:22.8 天気:晴 ３２.スペース・マウンテンの予測待ち時間は 42.719729291178666
気温:22.8 天気:晴 ３３.バズライトイヤーのアストロブラスターの予測待ち時間は 59.53339154464419
気温:22.8 天気:晴 ３４.モンスターズ・インク“ライド＆ゴーシーク！”の予測待ち時間は 62.0923792078419
気温:22.8 天気:晴 ３５.グランドサーキット・レースウェイの予測待ち時間は 0.0
気温:22.8 天気:晴 ３６.スタージェットの予測待ち時間は 2.9373331379395786
気温:22.8 天気:晴 ３７.スティッチ・エンカウンター（キャプテンＥＯ・ミクロアドベンチャー！）の予測待ち時間は 24.448714248572983
気温:22.8 天気:晴 ３８.エントランスグリ（ベビーカー、車椅子レンタル側）の予測待ち時間は 0.0
気温:22.8 天気:晴 ３９.エントランスグリ（メインストリート・ハウス側）の予測待ち時間は 0.3381799255417409
気温:22.8 天気:晴 ４０.ウッドチャック・グリーティングトレイル（デイジー）の予測待ち時間は 41.37391010477071
気温:22.8 天気:晴 ４１.ウッドチャック・グリーティングトレイル（ドナルド）の予測待ち時間は 70.67796720555279
気温:22.8 天気:晴 ４２.ミッキーの家とミート・ミッキーの予測待ち時間は 69.04453021922617