

6. 分析結果(中区分全体)

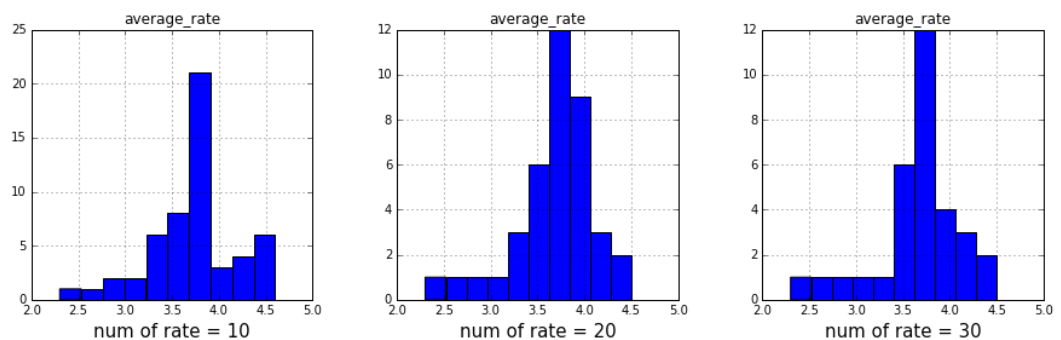
```
In [2]: # coding: shift-jis
%matplotlib inline
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data_amazon = pd.read_csv('api_data_smart_watches.csv')
```

```
In [3]: data_amazon.columns
```

```
Out[3]: Index([u'asin', u'date', u'manufacturer', u'price', u'rank', u'average_rate',
              u'num_of_rate', u'review_url'],
              dtype='object')
```

```
In [4]: fig = plt.figure(figsize=(15,4))
ax1 = fig.add_subplot(1,3,1)
ax2 = fig.add_subplot(1,3,2)
ax3 = fig.add_subplot(1,3,3)
data_amazon[data_amazon['num_of_rate'] >= 10].hist('average_rate',ax=ax1,bins=10)
data_amazon[data_amazon['num_of_rate'] >= 20].hist('average_rate',ax=ax2,bins=10)
data_amazon[data_amazon['num_of_rate'] >= 30].hist('average_rate',ax=ax3,bins=10)
ax1.set_xlabel("num of rate = 10",fontsize=15)
ax2.set_xlabel("num of rate = 20",fontsize=15)
ax3.set_xlabel("num of rate = 30",fontsize=15)
plt.show()
```



```
In [5]: data_10 = data_amazon[data_amazon[u'num_of_rate'] >= 10]
#succeed
suc = data_10[data_10['average_rate']>=4.0]
print('succeeded companies are %n%s') %suc[u'manufacturer'].unique()
#normal
normal = data_10[(data_10['average_rate'] >= 3.5) & (data_10['average_rate'] <= 3.8)]
print('%nnormal companies are %n%s') %str(normal[u'manufacturer'].unique()).decode('string-esca
scape')
#fale
fale = data_10[data_10['average_rate'] < 3.5]
print('%nfalse companies are %n%s') %str(fale[u'manufacturer'].unique()).decode('string-esca
pe')
```

succeeded companies are
['EloBeth' 'OEM' 'Pebble' 'ASUSTek' 'Samsung' 'by Galaxy']

normal companies are
['null' 'OZONE' 'Sony' 'Pebble' 'Withings'
'ソニー' 'Motorola' 'Shenzhen Wave' 'HUAWEI'
'Leesentec(リーセンテック)']

false companies are
['Shenzhen Wave' 'STK'
'ネット販売専門'
'Eagle Eye' 'MisSmart'
'Sleep tracker(スリープトラッカー)'
nan 'Sony' 'Sincere Inc.' 'Withings' 'Samsung']

```
In [6]: company_suc = [u'サムスン',u'ASUSTEK COMPUTER INC.'] #[u'サムスン',u'ASUSTEK COMPUTER INC.',
u'ベブルテクノロジー']
company_nor = [u'モトローラ',u'ソニー'] #[u'モトローラ',u'ソニー',u'ファウウェイ']

data_pat = pd.read_csv('/home/kyohei/wearable/data/01.文献リスト_特許.csv',encoding='SHIFT-J
IS')
categories_small = pd.read_csv('/home/kyohei/wearable/data/T_小区分.csv',encoding='SHIFT-JIS
')
categories_midlle = pd.read_csv('/home/kyohei/wearable/data/T_中区分.txt',encoding='SHIFT-JI
S')

data_suc_pat = data_pat[data_pat[u"筆頭出願人名"].isin(company_suc)]
data_fal_pat = data_pat[data_pat[u"筆頭出願人名"].isin(company_nor)]

data_suc_pat['succeed'] = 1
data_fal_pat['succeed'] = 0

data_all = pd.concat([data_suc_pat,data_fal_pat],axis=0)
data_all_watch = data_all[data_all[u'B0302']==1]
```

/home/kyohei/anaconda2/lib/python2.7/site-packages/IPython/core/interactiveshell.py:2723: Dt
ypeWarning: Columns (16,17) have mixed types. Specify dtype option on import or set low_memo
ry=False.

interactivity=interactivity, compiler=compiler, result=result)
/home/kyohei/anaconda2/lib/python2.7/site-packages/ipykernel/__main__.py:11: SettingWithCopy
Warning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: [http://pandas.pydata.org/pandas-docs/stable/indexing.h
tml#indexing-view-versus-copy](http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)

/home/kyohei/anaconda2/lib/python2.7/site-packages/ipykernel/__main__.py:12: SettingWithCopy
Warning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: [http://pandas.pydata.org/pandas-docs/stable/indexing.h
tml#indexing-view-versus-copy](http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)

```
In [7]: print("成功企業の内訳\n%s"%data_all_watch[data_all_watch['succeed']==1][u'筆頭出願人名'].value_counts())
print("\n非成功企業の内訳\n%s"%data_all_watch[data_all_watch['succeed']==0][u'筆頭出願人名'].value_counts())
```

```
成功企業の内訳
サムスン                134
ASUSTEK COMPUTER INC.    5
Name: 筆頭出願人名, dtype: int64
```

```
非成功企業の内訳
ソニー                 36
モトローラ             9
Name: 筆頭出願人名, dtype: int64
```

```
In [8]: middle_class = categories_midlle[u'記号'].values
df = []
keys = []
data_all_cal = data_all.iloc[:,19:]
for value in middle_class:
    category = categories_small[categories_small[u'記号'].str.startswith(value)]
    data_middle = data_all_watch[category[u'記号']].apply(lambda x: (np.sum(x)) / (np.sum(x)
) if np.sum(x) > 0 else np.sum(x),axis=1)
    df.append(data_middle)
    keys.append(value)
df_middle = pd.concat(df,keys=keys,axis=1)
```

```
In [9]: data_all_cal = pd.concat([data_all_watch.iloc[:,19:],df_middle,data_all_watch.iloc[:,1:]],axis=1)
```

```
In [10]: data_suc_cal = data_all_cal[data_all_cal['succeed']==1].iloc[:,19:-1].sum() / float(data_all_cal[data_all_cal['succeed']==1].shape[0])
data_fal_cal = data_all_cal[data_all_cal['succeed']==0].iloc[:,19:-1].sum() / float(data_all_cal[data_all_cal['succeed']==0].shape[0])

compare_category = pd.concat([data_suc_cal,data_fal_cal],axis=1)
compare_category.rename(columns={0:"succeed",1:"normal"},inplace=True)
compare_category["diff"] = compare_category["succeed"] - compare_category["normal"]
compare_category["diff_abs"] = abs(compare_category["succeed"] - compare_category["normal"])

categories_midlle = categories_midlle.set_index(u"記号",drop=True)
compare_category = pd.concat([categories_midlle,compare_category],axis=1)
```

```
In [11]: #data_all = data_all.fillna(0)

data_all_test = data_all_cal.iloc[:,19:]
data_all_test.loc[:,((data_all_cal.sum() > 0) & (data_all_cal.sum() < data_all_cal.shape[0]))]

from scipy.stats import chi2_contingency
data_all_test.columns[:-1]
pvalues = []
columns = data_all_test.columns[:-1]
for row in columns:
    table = pd.crosstab(data_all_test['succeed'],data_all_test[row])
    chisq_value,pvalue,df,expected = chi2_contingency(table,correction=False) #補正無し:correction=False
    pvalues.append([row,pvalue,chisq_value,df])

pvalues_df = pd.DataFrame(np.array(pvalues)[:,1:],index=np.array(pvalues)[:,0],columns=['pvalue'])
compare_category = pd.concat([compare_category,pvalues_df],axis=1)
compare_category['pvalue'] = compare_category['pvalue'].astype(float)
```

6.1 プラス域(成功企業の方が比率が高い)の注目技術

- 電源装置
- 全体設計(デザイン)

6.2 マイナス域(非成功企業の方が比率が高い)の注目技術

- ヘルスケア用途
- 訪問地
- 通信方式
- センサ(気候、バイタル、ポジション)
- 情報(オケーショナル、気候、場所・周辺、バイタル)

6.3 考察

小区分毎で差のあった区分の所属する中区分で同様に差があるという結果になった。成功企業では、電源装置や全体設計(デザイン)で出願比率が高く、ウェアラブルのハード的側面に対して注力していると考えられる。他方、非成功企業ではヘルスケア用途、訪問地というように、利用場面を想定していることが伺える。

```
In [12]: compare_category.sort_values('diff', ascending=False).head(20).style.bar(subset=['succeed', 'normal', 'diff_abs'], color='#d65f5f')
```

Out[12]:

	中区分	succeed	normal	diff	diff_abs	pvalue
B99	種別 非限定・その他	0.402878	0.0444444	0.358433	0.358433	6.85551e-06
D04	入力 装置	0.654676	0.355556	0.299121	0.299121	0.000407243
D05	出力 装置	0.748201	0.488889	0.259313	0.259313	0.00113704
D02	電源 装置	0.165468	0	0.165468	0.165468	0.00353246
D01	全体 設計 (デザ イン)	0.251799	0.111111	0.140687	0.140687	0.0467356
B06	装身 型	0.201439	0.0666667	0.134772	0.134772	0.0357812
D06	通信 装置	0.302158	0.2	0.102158	0.102158	0.183283
B01	HMD・ スマー トグラ ス型	0.402878	0.311111	0.0917666	0.0917666	0.270435
I99	入出 力情 報非 限定・ その他	0.129496	0.0444444	0.085052	0.085052	0.111116
F09	ラジ エー ション (音 波・光 線・放 射線) セン サー	0.129496	0.0444444	0.085052	0.085052	0.111116
G99	機能 の連 携非 限定・ その他	0.0791367	0	0.0791367	0.0791367	0.0516342
H02	医療 介護 用途	0.0647482	0	0.0647482	0.0647482	0.0800683
H05	生活 用途	0.028777	0	0.028777	0.028777	0.249922
	装着					

マイナス域

```
In [13]: compare_category.sort_values('diff', ascending=True).head(20).style.bar(subset=['succeed', 'normal', 'diff_abs'], color='#d65f5f')
```


Out[13]:

	中区分	succeed	normal	diff	diff_abs	pvalue
I01	基本データ	0.568345	0.911111	-0.342766	0.342766	2.71625e-05
G01	情報設備・機器との連携	0.280576	0.577778	-0.297202	0.297202	0.000288621
I06	オケーショナル・データ	0	0.288889	-0.288889	0.288889	4.92111e-11
F06	クライメイト(気候)・センサー	0.028777	0.311111	-0.282334	0.282334	3.00585e-08
I04	気候データ	0	0.266667	-0.266667	0.266667	3.03383e-10
H03	ヘルスケア用途	0.0143885	0.266667	-0.252278	0.252278	2.89435e-08
J04	訪問地	0.0215827	0.266667	-0.245084	0.245084	1.76868e-07
J99	入出力情報非限定・その他	0.0647482	0.266667	-0.201918	0.201918	0.000213485
D07	通信方式	0.179856	0.377778	-0.197922	0.197922	0.00596939
I05	場所・周辺状況データ	0.028777	0.222222	-0.193445	0.193445	2.10034e-05
F02	バイタル(生	0.18705	0.377778	-0.190727	0.190727	0.00859396

絶対値

```
In [14]: compare_category[compare_category['diff_abs'] >= 0.2].sort_values('diff_abs', ascending=False)
         .style.bar(subset=['succeed', 'normal', 'diff_abs'], color='#d65f5f')
```

Out[14]:

	中区分	succeed	normal	diff	diff_abs	pvalue
B99	種別 非限定・その他	0.402878	0.0444444	0.358433	0.358433	6.85551e-06
I01	基本データ	0.568345	0.911111	-0.342766	0.342766	2.71625e-05
D04	入力装置	0.654676	0.355556	0.299121	0.299121	0.000407243
G01	情報設備・機器との連携	0.280576	0.577778	-0.297202	0.297202	0.000288621
I06	オケージョナル・データ	0	0.288889	-0.288889	0.288889	4.92111e-11
F06	クライメイト（気候）・センサー	0.028777	0.311111	-0.282334	0.282334	3.00585e-08
I04	気候データ	0	0.266667	-0.266667	0.266667	3.03383e-10
D05	出力装置	0.748201	0.488889	0.259313	0.259313	0.00113704
H03	ヘルスケア用途	0.0143885	0.266667	-0.252278	0.252278	2.89435e-08
J04	訪問地	0.0215827	0.266667	-0.245084	0.245084	1.76868e-07
J99	入出力情報非限定・その他	0.0647482	0.266667	-0.201918	0.201918	0.000213485

7. 分析結果(大区分毎)

7.1 用途

- ヘルスケア用途で非成功企業の出願比率が高かった。スポーツ用途でも小さいながら差がでた。

In [18]: `compare_category[compare_category.index.str.startswith("H")].style.bar(subset=['succeed', 'normal', 'diff_abs'], color='#d65f5f')`

Out[18]:

	中 区 分	succeed	normal	diff	diff_abs	pvalue
H01	業 務 向 け・ 個 人 向 け	1	1	0	0	nan
H02	医 療 介 護 用 途	0.0647482	0	0.0647482	0.0647482	0.0800683
H03	ヘル ス ケ ア 用 途	0.0143885	0.266667	-0.252278	0.252278	2.89435e-08
H04	ス ポ ー ツ 用 途	0	0.0666667	-0.0666667	0.0666667	0.00214601
H05	生 活 用 途	0.028777	0	0.028777	0.028777	0.249922
H06	業 務 用 途	0.00719424	0	0.00719424	0.00719424	0.568315
H07	エン ター テイ メント 用 途	0	0	0	0	nan
H99	用 途 非 限 定・ そ の 他	0.028777	0.133333	-0.104556	0.104556	0.00716563

7.2 情報

- 基本データで非成功企業の出願件数比率がたかった。特徴のある区分ではないので、ここでは注目しなかった。
- 気候データ、場所・周辺データ、オケーショナルデータ、バイタルデータで非成功企業の出願比率がたかった。

```
In [16]: compare_category[compare_category.index.str.startswith("I")].style.bar(subset=['succeed', 'normal', 'diff_abs'], color='#d65f5f')
```


Out[16]:

	中 区 分	succeed	normal	diff	diff_abs	pvalue
I01	基本データ	0.568345	0.911111	-0.342766	0.342766	2.71625e-05
I02	蓄積データ	0.0719424	0.111111	-0.0391687	0.0391687	0.403949
I03	時間データ	0.0431655	0.0222222	0.0209432	0.0209432	0.523273
I04	気候データ	0	0.266667	-0.266667	0.266667	3.03383e-10
I05	場所・周辺状況データ	0.028777	0.222222	-0.193445	0.193445	2.10034e-05
I06	オケージョナルデータ	0	0.288889	-0.288889	0.288889	4.92111e-11
I07	バイタルデータ	0.0719424	0.177778	-0.105835	0.105835	0.0377899
I08	個人データ	0.0431655	0.0222222	0.0209432	0.0209432	0.523273
I09	商品データ	0	0	0	0	nan
I10	拡張・仮想現実	0	0	0	0	nan

7.3 センサー

- バイタルセンサ、ポジションセンサ、気候センサで差が出た。

```
In [17]: compare_category[compare_category.index.str.startswith("F")].style.bar(subset=['succeed', 'normal', 'diff_abs'], color='#d65f5f')
```

Out[17]:

	中区分	succeed	normal	diff	diff_abs	pvalue
F01	モーション(動作)・センサー	0.230216	0.355556	-0.12534	0.12534	0.0960604
F02	バイタル(生体)・センサー	0.18705	0.377778	-0.190727	0.190727	0.00859396
F03	セント(嗅覚)・テースト(味覚)センサー	0	0	0	0	nan
F04	ボジション(位置)・センサー	0.0791367	0.266667	-0.18753	0.18753	0.000946032
F05	ペリフェラル(周辺)・センサー	0.0359712	0.0888889	-0.0529177	0.0529177	0.152578
F06	クライメイト(気候)・センサー	0.028777	0.311111	-0.282334	0.282334	3.00585e-08
F07	タイム(時刻)・センサー	0	0	0	0	nan
	ケミストリー					

