価格vs評価

In [39]:
```
%matplotlib inline
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [54]:
```
smwatches = pd.read_csv("amazon_smart_watch.csv")
rev = pd.read_csv("review_data.csv")
print(smwatches.head())
print(rev.head())
```

```
         asin date        manufacturer  price      rank
0  B00YBY039O  NaN               DLAND  499.0   27558.0
1  B00XMUYASS  NaN  Leesentec(リーセンテック)  799.0   12795.0
2  B00XMTMTYG  NaN  Leesentec(リーセンテック)  799.0   59600.0
3  B0132J53JI  NaN           ポケットシステムズ  980.0  178966.0
4  B01FGVT0OE  NaN                IPRO  500.0  263855.0
         asin  average_rate  num_of_rate  ¥
0  B00XMUYASS           3.9           10
1  B0132J53JI           1.0            1
2  B01FGVT0OE           0.0            0
3  B0186P92W2           0.0            0
4  B01AIEBE3E           0.0            0

                                        review_url
0  http://www.amazon.jp/reviews/iframe?akid=AKIAI...
1  http://www.amazon.jp/reviews/iframe?akid=AKIAI...
2  http://www.amazon.jp/reviews/iframe?akid=AKIAI...
3  http://www.amazon.jp/reviews/iframe?akid=AKIAI...
4  http://www.amazon.jp/reviews/iframe?akid=AKIAI...
```

In [55]:
```
smwatches = smwatches.set_index("asin",drop=True)
rev = rev.set_index("asin",drop=True)
```

In [95]:
```
all_data = pd.concat([smwatches,rev],axis=1,join='inner')
```

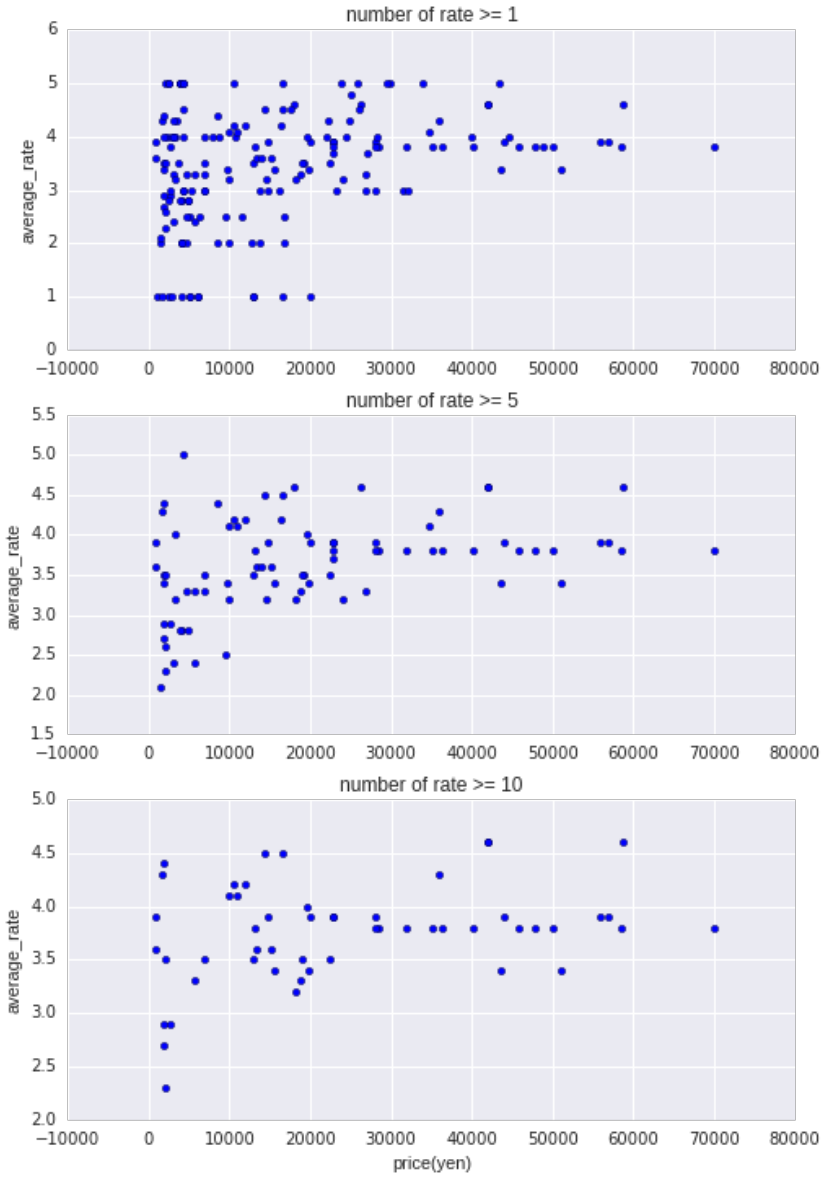
価格と評価の関係

- 縦軸平均評価
- 横軸価格
- 評価回数が1回以上、5回以上、10回以上の3段階で整理

In [167]:
```
fig = plt.figure(figsize=(8,12))
ax1 = fig.add_subplot(3,1,1)
ax2 = fig.add_subplot(3,1,2)
ax3 = fig.add_subplot(3,1,3)

ax1.scatter(all_data_2[all_data_2["num_of_rate"]>0]["price"],
            all_data_2[all_data_2["num_of_rate"]>0]["average_rate"])
ax2.scatter(all_data_2[all_data_2["num_of_rate"]>4]["price"],
            all_data_2[all_data_2["num_of_rate"]>4]["average_rate"])
ax3.scatter(all_data_2[all_data_2["num_of_rate"]>9]["price"],
            all_data_2[all_data_2["num_of_rate"]>9]["average_rate"])
ax1.set(ylabel="average_rate",title="number of rate >= 1")
ax2.set(ylabel="average_rate",title="number of rate >= 5")
ax3.set(ylabel="average_rate",title="number of rate >= 10",xlabel="price(yen)")

sns.plt.show()
```



In [96]:
```
all_data_2 = all_data.dropna(subset=["price",])
```

- 価格の高い安いと、評価の高い低いは関係がない模様
- 評価回数が1つの場合、評価が5または1の場合があるので、以降は除外

評価件数が5件以上の場合の線形回帰

In [141]:
```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(all_data_2[all_data_2["num_of_rate"]>4][["price"]],
          all_data_2[all_data_2["num_of_rate"]>4]["average_rate"])
```

Out[141]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)

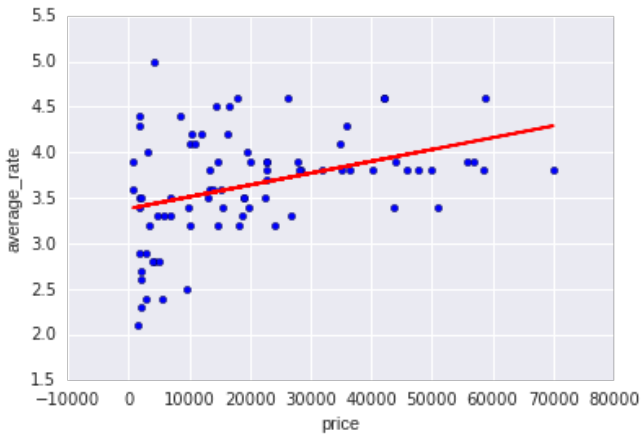
In [145]:
```
from scipy.stats import linregress
slope,intercept,r_value,p_value,std_err = linregress(all_data_2[all_data_2["num_of_rate"]>4]
["price"],
                                                     all_data_2[all_data_2["num_of_rate"]>4]
["average_rate"])
print("切片:%0.2f") %intercept
print("回帰係数:%0.6f") %slope
print("決定係数:%0.2f") %model.score(all_data_2[all_data_2["num_of_rate"]>4][["price"]],
                                    all_data_2[all_data_2["num_of_rate"]>4]["average_rate"])
print("p値:%0.5f") %p_value
print("¥n相関係数:%0.2f") %r_value
```

```
切片:3.38
回帰係数:0.000013
決定係数:0.14
p値:0.00054

相関係数:0.37
```

In [153]:
```
plt.scatter(all_data_2[all_data_2["num_of_rate"]>4]["price"],
            all_data_2[all_data_2["num_of_rate"]>4]["average_rate"])
plt.plot(all_data_2[all_data_2["num_of_rate"]>4][["price"]],
        model.predict(all_data_2[all_data_2["num_of_rate"]>4][["price"]]),c="r")
plt.xlabel("price")
plt.ylabel("average_rate")

sns.plt.show()
```



評価件数10件以上の線形回帰

In [171]:
```
model = LinearRegression()
model.fit(all_data_2[all_data_2["num_of_rate"]>9][["price"]],
          all_data_2[all_data_2["num_of_rate"]>9]["average_rate"])
```

Out[171]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)

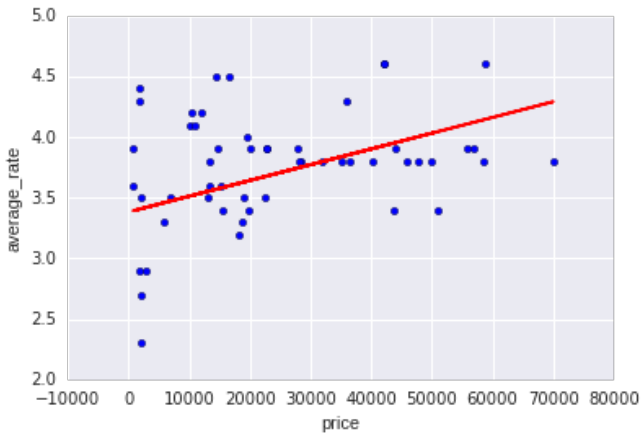
In [172]:
```
from scipy.stats import linregress
slope,intercept,r_value,p_value,std_err = linregress(all_data_2[all_data_2["num_of_rate"]>9]
["price"],
                                                     all_data_2[all_data_2["num_of_rate"]>9]
["average_rate"])
print("切片:%0.2f") %intercept
print("回帰係数:%0.6f") %slope
print("決定係数:%0.2f") %model.score(all_data_2[all_data_2["num_of_rate"]>9][["price"]],
                                    all_data_2[all_data_2["num_of_rate"]>9]["average_rate"])
print("p値:%0.5f") %p_value
print("¥n相関係数:%0.2f") %r_value
```

```
切片:3.58
回帰係数:0.000008
決定係数:0.09
p値:0.02863

相関係数:0.30
```

In [156]:
```
plt.scatter(all_data_2[all_data_2["num_of_rate"]>9]["price"],
            all_data_2[all_data_2["num_of_rate"]>9]["average_rate"])
plt.plot(all_data_2[all_data_2["num_of_rate"]>9][["price"]],
        model.predict(all_data_2[all_data_2["num_of_rate"]>9][["price"]]),c="r")
plt.xlabel("price")
plt.ylabel("average_rate")

sns.plt.show()
```



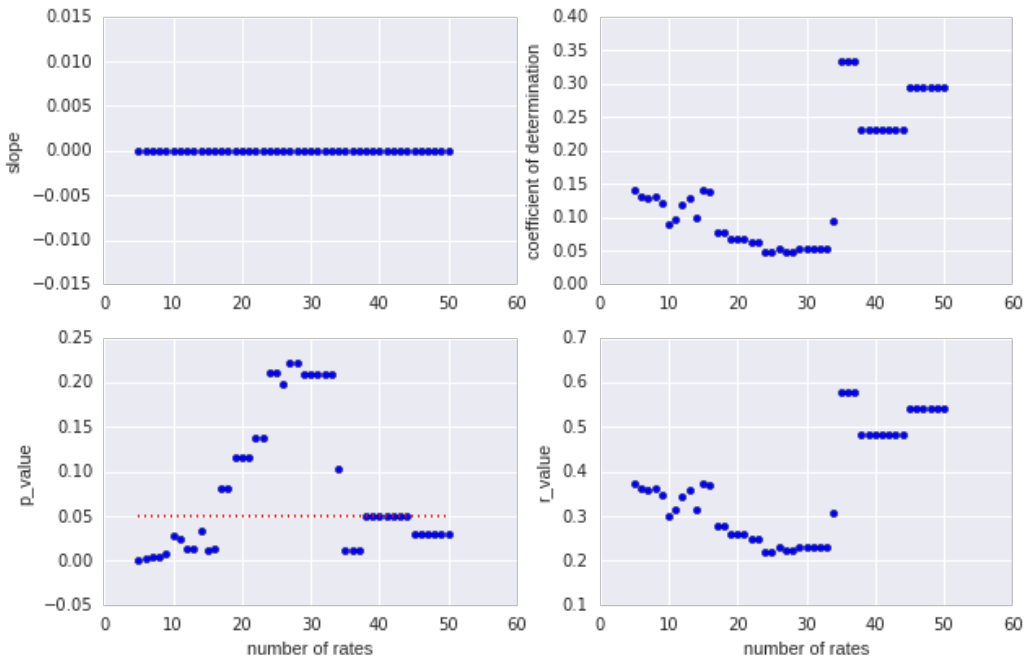
評価件数をパラメータに使って分析

- 横軸評価件数
- 縦軸、回帰係数、決定係数、p値、相関係数

In [176]:
```
stats = []
for i in np.linspace(5,50,46):
    slope,intercept,r_value,p_value,std_err = linregress(all_data_2[all_data_2["num_of_rate"
] >= i]["price"],
                                                         all_data_2[all_data_2["num_of_rate"] >=
 i]["average_rate"])
    model = LinearRegression()
    model.fit(all_data_2[all_data_2["num_of_rate"] >= i][["price"]],
              all_data_2[all_data_2["num_of_rate"] >= i]["average_rate"])
    determin = model.score(all_data_2[all_data_2["num_of_rate"] >= i][["price"]],
                           all_data_2[all_data_2["num_of_rate"] >= i]["average_rate"])
    stats.append([i,slope,intercept,determin,r_value,p_value,std_err])
columns=['number','slope','intercept','determin','r_value','p_value','std_err']
stats_df = pd.DataFrame(stats,columns=columns)
```

In [194]:
```
fig = plt.figure(figsize=(10,10))
ax1 = fig.add_subplot(3,2,1) #slope
ax2 = fig.add_subplot(3,2,2) #determin
ax3 = fig.add_subplot(3,2,3) #p_value
ax4 = fig.add_subplot(3,2,4) #r_value

ax1.scatter(stats_df["number"],stats_df["slope"])
ax1.set(ylabel="slope")
ax2.scatter(stats_df["number"],stats_df["determin"])
ax2.set(ylabel="coefficient of determination")
ax3.scatter(stats_df["number"],stats_df["p_value"])
ax3.plot([5,50],[0.05,0.05],ls=":",c='r')
ax3.set(ylabel="p_value",xlabel="number of rates")
ax4.scatter(stats_df["number"],stats_df["r_value"])
ax4.set(ylabel="r_value",xlabel="number of rates")
sns.plt.show()
```



- 回帰係数は評価件数を増やしても変化なし
- 回帰係数のp値は評価件数が17件～29件で0.05を超える
- 決定係数、相関係数は30件を超えると値が急増する
- 評価件数が30件以上のデータ数は32個

評価件数30件

In [197]:
```
model = LinearRegression()
model.fit(all_data_2[all_data_2["num_of_rate"] >29][["price"]],
          all_data_2[all_data_2["num_of_rate"] >29]["average_rate"])
plt.scatter(all_data_2[all_data_2["num_of_rate"]>29]["price"],
            all_data_2[all_data_2["num_of_rate"]>29]["average_rate"])
plt.plot(all_data_2[all_data_2["num_of_rate"]>29][["price"]],
        model.predict(all_data_2[all_data_2["num_of_rate"]>29][["price"]]),c="r")
plt.xlabel("price")
plt.ylabel("average_rate")

sns.plt.show()
```