

```
In [1]: import matplotlib as plt
import seaborn as sns
import pandas as pd
import numpy as np
import json as js
```

```
In [2]: pwd
```

```
Out[2]: '/Users/mingyuanna/desktop/BA1'
```

```
In [58]: before = pd.read_csv("data/combined1.csv")
before = before.rename(columns = {"label": "label_dim1"})
```

```
In [59]: before_dim2 = pd.read_csv("data/before_dim2.csv")
before_dim2 = before_dim2.rename(columns = {"Unnamed: 2": "label"}).loc[:,["ResponseId", "label"]]
before = before_dim2.merge(before, how="right", left_on="ResponseId", right_on="ResponseId")
before
```

ResponseId	label	round	tick	orderId	taskId	taskTicks	workerId	workerTicks	label_dim1
0	R_DuOsl6BelnUee5	3	1	1	1	2	0	0	3
1	R_DuOsl6BelnUee5	3	1	1	2	1	2	1	0
2	R_DuOsl6BelnUee5	3	1	1	0	0	0	0	3
3	R_DuOsl6BelnUee5	3	1	2	3	1	2	2	0
4	R_DuOsl6BelnUee5	3	1	2	0	0	0	0	3
...	...	...	...	...	...	...	...	...	...
100561	R_zTBywJBicujhn	7	6	33	0	0	0	0	1
100562	R_zTBywJBicujhn	7	6	33	0	0	0	0	1
100563	R_zTBywJBicujhn	7	6	34	4	3	2	1	0
100564	R_zTBywJBicujhn	7	6	34	0	0	0	0	1
100565	R_zTBywJBicujhn	7	6	34	0	0	0	0	1

100566 rows x 10 columns

```
In [60]: after = pd.read_csv("data/combined2.csv")
after = after.rename(columns = {"label": "label_dim1"})
```

```
In [61]: after_dim2 = pd.read_csv("data/after_dim2.csv")
after_dim2 = after_dim2.rename(columns = {"Unnamed: 2": "label"}).loc[:,["ResponseId", "label"]]
after = after_dim2.merge(after, how="right", left_on="ResponseId", right_on="ResponseId")
after
```

ResponseId	label	round	tick	orderId	taskId	taskTicks	workerId	workerTicks	label_dim1
0	R_DhHbXVBuU69ED9QJ	4	1	1	1	2	0	0	2
1	R_DhHbXVBuU69ED9QJ	4	1	1	2	1	2	1	0
2	R_DhHbXVBuU69ED9QJ	4	1	1	3	1	2	2	0
3	R_DhHbXVBuU69ED9QJ	4	1	2	0	0	0	0	2
4	R_DhHbXVBuU69ED9QJ	4	1	2	0	0	0	0	2
...	...	...	...	...	...	...	...	...	...
141757	R_zQZiBEOLFD13yRX	5	6	37	0	0	0	0	3
141758	R_zQZiBEOLFD13yRX	5	6	37	0	0	0	0	3
141759	R_zQZiBEOLFD13yRX	5	6	38	4	3	2	1	0
141760	R_zQZiBEOLFD13yRX	5	6	38	0	0	0	0	3
141761	R_zQZiBEOLFD13yRX	5	6	38	0	0	0	0	3

141762 rows x 10 columns

## chi-square testing

```
In [62]: from scipy.stats import chi2_contingency
from collections import Counter

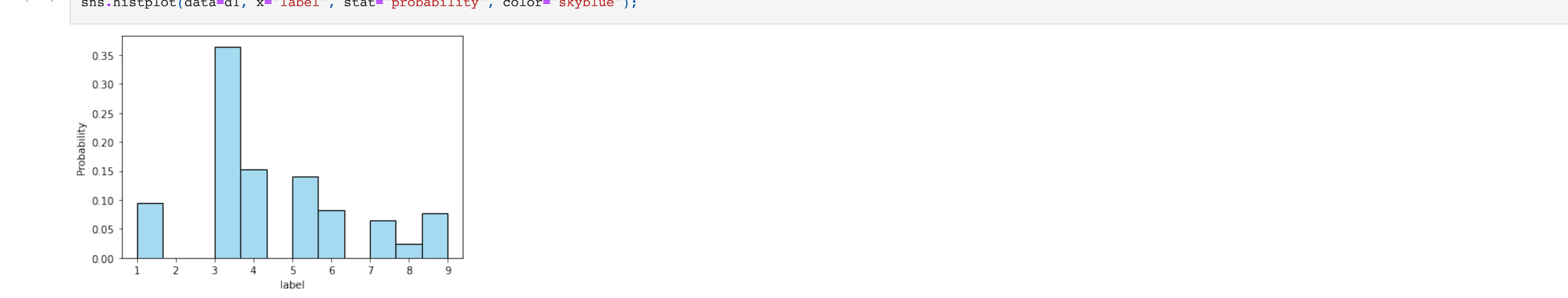
def chisquare(array1, array2, count=True):
    if not count:
        data = [array1, array2]
        stat, p, dof, expected = chi2_contingency(data)
    else:
        c1, c2 = Counter(array1), Counter(array2)
        before_dis, after_dis = [], []
        for i in set(c1).union(set(c2)):
            before_dis.append(c1[i])
            after_dis.append(c2[i])
        print(before_dis)
        print(after_dis)
        data = [before_dis, after_dis]
        stat, p, dof, expected = chi2_contingency(data)

    alpha = 0.05
    print('p value is ' + str(p))
    if p <= alpha:
        print('difference between the two distributions (reject H0)')
    else:
        print('no difference between the two distributions (H0 holds true)')
```

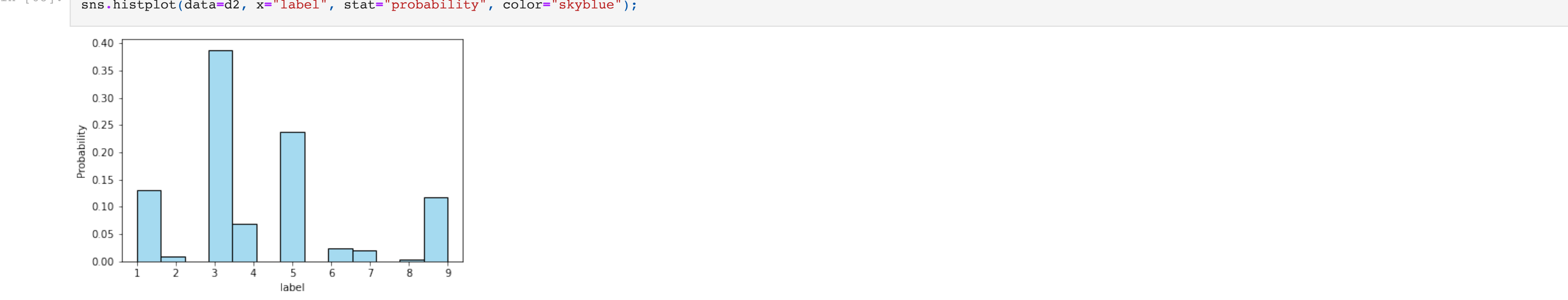
## Distribution of label before and after

```
In [63]: d1 = before.groupby("ResponseId", as_index=False).agg(lambda x:x.iloc[0])
d2 = after.groupby("ResponseId", as_index=False).agg(lambda x:x.iloc[0])
```

```
In [67]: sns.histplot(data=d1, x="label", stat="probability", color="skyblue");
```



```
In [68]: sns.histplot(data=d2, x="label", stat="probability", color="skyblue");
```



## Chi-Square Testing

```
In [69]: chisquare(d1["label"], d2["label"])

[16, 0, 62, 26, 24, 14, 11, 4, 13]
[32, 2, 95, 17, 58, 6, 5, 1, 29]
p value is 0.00017703873541072
difference between the two distributions (reject H0)
```

## Analysis of Compliance

```
In [70]: def compliance(df):
    ids = []
    rounds = []
    server = []
    label = []
    for player in set(np.array(df["ResponseId"])):
        tem = df[(df["ResponseId"] == player) & (df["round"] == 1)]
        if len(tem) != 0: # no response there
            ids.append(player)
            rounds.append(1)
            label.append(tem["label"].iloc[0])
            if i <= 2:
                num = sum((tem["workerId"] == 2) & (tem["taskId"] == 1))
            else:
                num = sum((tem["workerId"] == 1) & (tem["taskId"] == 1))
            server.append(num)
        else:
            print(player, i)

    d = {
        "ResponseId": ids,
        "round": rounds,
        "numServerCook": server,
        "label": label
    }
    return pd.DataFrame(d)
```

```
In [71]: before_compliance = compliance(before)
```

```
In [72]: after_compliance = compliance(after)
```

```
In [73]: before_compliance
```

ResponseId	round	numServerCook	label
0	R_3GdmKNZwu4kdP1	1	2
1	R_3GdmKNZwu4kdP1	2	2
2	R_3GdmKNZwu4kdP1	3	2
3	R_3GdmKNZwu4kdP1	4	2
4	R_3GdmKNZwu4kdP1	5	2
...	...	...	...
1016	R_3L6WNwqJup7i2Ws	2	1
1016	R_3L6WNwqJup7i2Ws	3	11
1017	R_3L6WNwqJup7i2Ws	4	3
1018	R_3L6WNwqJup7i2Ws	5	3
1019	R_3L6WNwqJup7i2Ws	6	9

1020 rows x 4 columns

```
In [74]: after_compliance
```

ResponseId	round	numServerCook	label
0	R_3QKNPgR0RmCwX	1	1
1	R_3QKNPgR0RmCwX	2	1
2	R_3QKNPgR0RmCwX	3	9
3	R_3QKNPgR0RmCwX	4	4
4	R_3QKNPgR0RmCwX	5	6
...	...	...	...
1466	R_3gMLEVUqUBNFXW	2	1
1466	R_3gMLEVUqUBNFXW	3	10
1467	R_3gMLEVUqUBNFXW	4	3
1468	R_3gMLEVUqUBNFXW	5	1
1469	R_3gMLEVUqUBNFXW	6	3

1470 rows x 4 columns

```
In [ ]:
```

## group by rounds

```
In [75]: round_before = before_compliance.groupby("round").agg(np.average).loc[:,["numServerCook"]]
round_before
```

numServerCook
round
1
2
3
4
5
6

```
In [76]: round_after = after_compliance.groupby("round").agg(np.average).loc[:,["numServerCook"]]
round_after
```

numServerCook
round
1
2
3
4
5
6

```
In [77]: a = np.array(round_before["numServerCook"])
```

```
In [78]: b = np.array(round_after["numServerCook"])
```

## chi-square testing

```
In [79]: chisquare(a, b, count=False)

p value is 0.999958667113752
no difference between the two distributions (H0 holds true)
```

```
In [ ]:
```

## group by tip

```
In [80]: tip_before = before_compliance.groupby("label", as_index=False).agg(np.average).loc[:,["label", "numServerCook"]]
tip_before
```

```
In [81]: tip_after = after_compliance.groupby("label", as_index=False).agg(np.average).loc[:,("label","numServerCook")]
tip_after
```

```
Out[81]:
```

	label	numServerCook
0	1	1.651042
1	2	1.759000
2	3	1.854396
3	4	1.909095
4	5	1.867916
5	6	1.816662

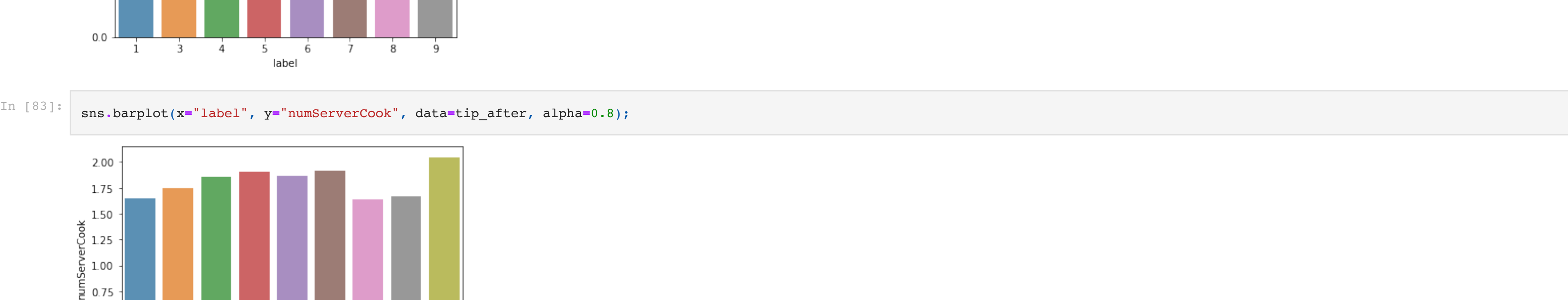
```
In [81]: tip_after = after_compliance.groupby("label", as_index=False).agg(np.average).loc[:,["label", "numServerCook"]]
tip_after
```

7	8	1.666567
8	9	2.045977

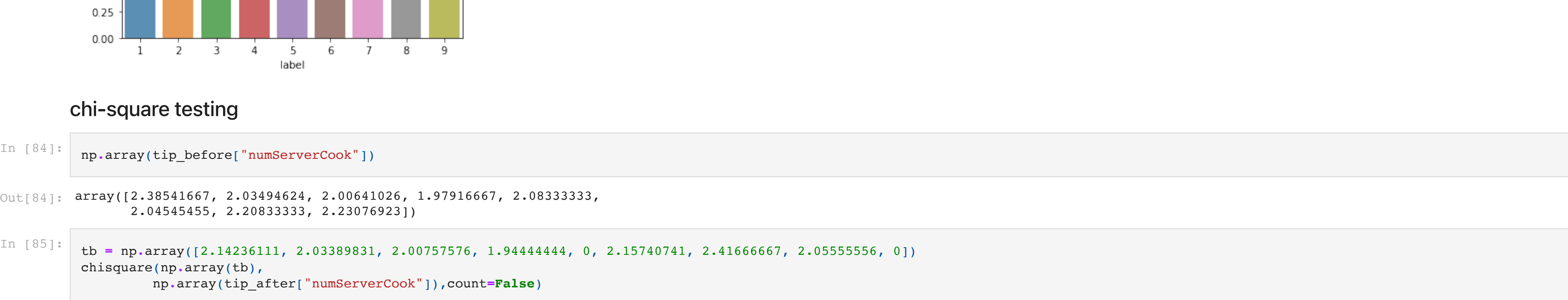
```
In [82]: sns.barplot(x="label", y="numServerCook", data=tip_before, alpha=0.8);
```

label	numServerCook
0	24
1	20
2	20
3	20
4	20
5	22
6	22

```
In [82]: sns.barplot(x="label", y="numServerCook", data=tip_before, alpha=0.8);
```



```
In [83]: sns.barplot(x="label", y="numServerCook", data=tip_after, alpha=0.8);
```



## chi-square testing

```
In [84]: np.array(tip_before["numServerCook"])

array([2.3854167, 2.0349424, 2.0641026, 1.9791667, 2.0833333,
       2.0454555, 2.2083333, 2.2307692])
```

```
In [85]: tb = np.array([2.14236111, 2.03389831, 2.00757576, 1.94444444, 0, 2.15740741, 2.41666667, 2.05555556, 0])
chisquare(np.array(tb),
np.array(tip_after["numServerCook"]), count=False)
```

p value is 0.8436870817093086  
no difference between the two distributions (H0 holds true)

```
In [ ]:
```

## Testing Aversion Distribution

```
In [86]: ave_before = before_compliance[before_compliance["numServerCook"] >= 3]
[before_compliance["numServerCook"] >= 3]
ave_before = ave_before.rename(columns = {"round": "numAversion"})
```

```
In [87]: ave_after = after_compliance[after_compliance["numServerCook"] >= 3]
[after_compliance["numServerCook"] >= 3]
ave_after = ave_after.rename(columns = {"round": "numAversion"})
```

```
In [88]: def counting(df):
    count = []
    for i in np.arange(1,10):
        cou = len(df[df["label"] == i])
        count.append(cou)
    return count
```

```
In [89]: counting(before_compliance), counting(after_compliance)
[_, for _ in counting(before_compliance) if _]
```

```
Out[89]: [96, 372, 156, 144, 84, 66, 24, 78]
```

```
In [90]: ave_before["total_num_label"] = [_ for _ in counting(before_compliance) if _]
ave_after["proportion_aversion"] = ave_before["numAversion"] / ave_before["total_num_label"]
ave_after["proportion_label"] = ave_after["total_num_label"] / after_compliance.shape[0]
ave_before
```

1	42	96	0.437500	0.094118
3	122	372	0.327957	0.364706
4	44	156	0.282051	0.152941
5	54	144	0.375000	0.141176
6	27	84	0.321429	0.082353
7	25	66	0.378788	0.064706
8	8	24	0.333333	0.023529
9	29	76	0.371995	0.076471

In [91]:

ave\_after['total\_num\_label'] = counting(after\_compliance)

ave\_after['proportion\_aversion'] = ave\_after['numAversion'] / ave\_after['total\_num\_label']

```
In [91]: ave_after["total_num_label"] = counting(after_compliance)
ave_after["proportion_aversion"] = ave_after["numAversion"] / ave_after["total_num_label"]
ave_after["proportion_label"] = ave_after["total_num_label"] / after_compliance.shape[0]
ave_after
```

Out[91]:

	numAversion	total_num_label	proportion_aversion	proportion_label
label				
1	38	192	0.197917	0.130612
2	5	12	0.416667	0.009163
3	145	570	0.254386	0.387755
4	29	102	0.284314	0.069388
5	81	348	0.232759	0.236735
6	6	36	0.166667	0.024490
7	3	30	0.100000	0.020408
8	1	6	0.166667	0.004082

```
In [ ]:
```

## Testing Compliance Distribution

```
In [92]: com_before = before_compliance[before_compliance["numServerCook"] >= 2].groupby(
"label").count().loc[:,["round"]]
com_before = com_before.rename(columns = {"round": "numCompliance"})
```

```
In [93]: com_after = after_compliance[after_compliance["numServerCook"] >= 2].groupby(
"label").count().loc[:,["round"]]
com_after = com_after.rename(columns = {"round": "numCompliance"})
```

numCompliance
label
1
2
3
4
5
6
7
8
9

```
In [95]: com_before["total_num_label"] = [_ for _ in counting(before_compliance) if _]
com_after["proportion_compliance"] = com_after["numCompliance"] / com_after["total_num_label"]
com_before["proportion_label"] = com_before["total_num_label"] / before_compliance.shape[0]
com_before
```

```
7      19
8      5
9      70

In [95]: com_before['total_num_label'] = 1_ for _ in counting(before_compliance) if _]
com_before['proportion_compliance'] = com_before['numCompliance'] / com_before['total_num_label']
com_before['proportion_label'] = com_before['total_num_label'] / before_compliance.shape[0]
com_before

Out[95]: numCompliance total_num_label proportion_compliance proportion_label

label
1      33      96      0.343750      0.094118
```

```
In [96]: com_after["total_num_label"] = counting(after_compliance)
com_after["proportion_compliance"] = com_after["numCompliance"] / com_after["total_num_label"]
com_after["proportion_label"] = com_after["total_num_label"] / after_compliance.shape[0]
com_after
```

5	61	144	0.423611	0.141176
6	38	84	0.452381	0.082353
7	33	66	0.500000	0.064706
8	11	24	0.458333	0.023529
9	26	78	0.333333	0.076471

```
In [96]: com_after['total_num_label'] = counting(after_compliance)
com_after['proportion_compliance'] = com_after['numCompliance'] / com_after['total_num_label']
com_after['proportion_label'] = com_after['total_num_label'] / after_compliance.shape[0]
com_after

Out[96]: numCompliance  total_num_label  proportion_compliance  proportion_label
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```