In [51]:
```python
import matplotlib as plt
import seaborn as sns
import pandas as pd
import numpy as np
import json as js
```

In [52]:
```python
pwd
```

Out[52]: '/Users/mingyuanma/Desktop/HAI/analysis'

In [53]:
```python
before = pd.read_csv("../../data/phase1/combined1.csv")
before = before.rename(columns = {"label":"label_dim1"})
```

In [54]:
```python
before_dim2 = pd.read_csv("../../data/phase1/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 = before[before["round"] >= 3]
before
```

Out[54]:

	ResponseId	label	round	tick	orderId	taskId	taskTicks	workerId	workerTicks	label_dim1
135	R_0uOsLe6BeLnUee5	3	3	1	2	1	2	0	0	3
136	R_0uOsLe6BeLnUee5	3	3	1	1	1	2	1	0	3
137	R_0uOsLe6BeLnUee5	3	3	1	0	0	0	0	0	3
138	R_0uOsLe6BeLnUee5	3	3	2	0	0	0	0	0	3
139	R_0uOsLe6BeLnUee5	3	3	2	0	0	0	0	0	3
...	...	...	...	...	...	...	...	...	...	...
100561	R_zTlbYwl8ijcujhn	7	6	33	0	0	0	0	0	1
100562	R_zTlbYwl8ijcujhn	7	6	33	0	0	0	0	0	1
100563	R_zTlbYwl8ijcujhn	7	6	34	4	3	2	1	0	1
100564	R_zTlbYwl8ijcujhn	7	6	34	0	0	0	0	0	1
100565	R_zTlbYwl8ijcujhn	7	6	34	0	0	0	0	0	1

76941 rows × 10 columns

In [55]:
```python
after = pd.read_csv("../../data/phase2/combined2.csv")
after = after.rename(columns = {"label":"label_dim1"})
```

In [56]:
```python
after_dim2 = pd.read_csv("../../data/phase2/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 = after[after["round"] >= 3]
after
```

Out[56]:

	ResponseId	label	round	tick	orderId	taskId	taskTicks	workerId	workerTicks	label_dim1
117	R_0HBxBV8U696D9QJ	4	3	1	1	1	2	0	0	2
118	R_0HBxBV8U696D9QJ	4	3	1	2	1	2	1	0	2
119	R_0HBxBV8U696D9QJ	4	3	1	0	0	0	0	0	2
120	R_0HBxBV8U696D9QJ	4	3	2	3	1	2	1	0	2
121	R_0HBxBV8U696D9QJ	4	3	2	0	0	0	0	0	2
...	...	...	...	...	...	...	...	...	...	...
141757	R_zZQIbE0LFD13yRX	5	6	37	0	0	0	0	0	3
141758	R_zZQIbE0LFD13yRX	5	6	37	0	0	0	0	0	3
141759	R_zZQIbE0LFD13yRX	5	6	38	4	3	2	1	0	3
141760	R_zZQIbE0LFD13yRX	5	6	38	0	0	0	0	0	3
141761	R_zZQIbE0LFD13yRX	5	6	38	0	0	0	0	0	3

108330 rows × 10 columns

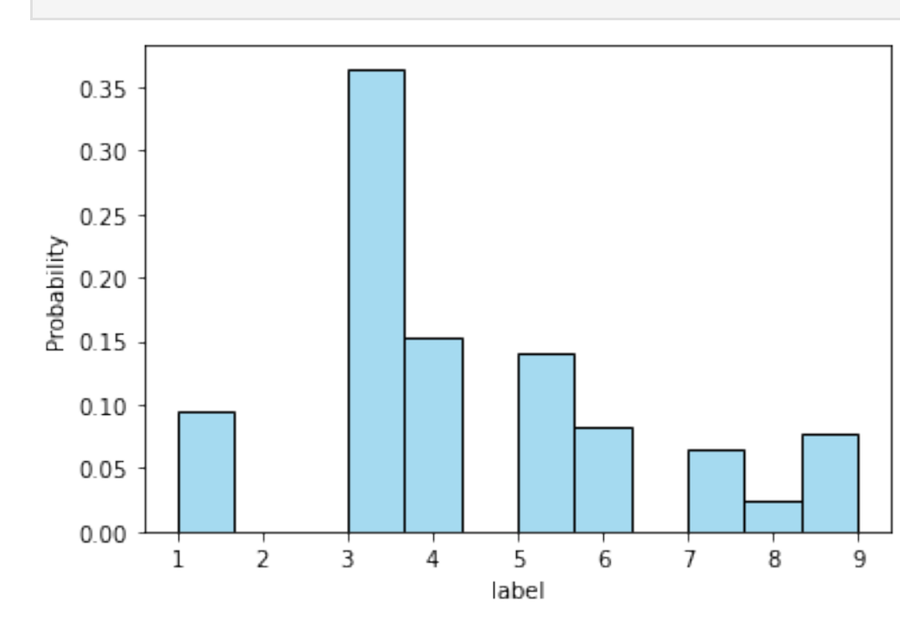
chi-square testing

In [57]:
```python
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)')
```

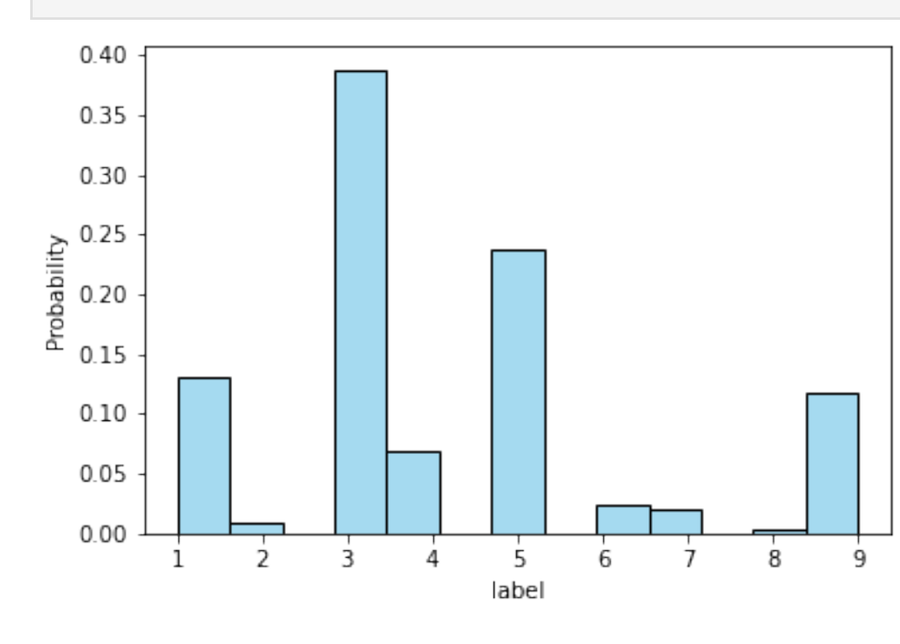
Distribution of label before and after

In [58]:
```python
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 [59]:
```python
sns.histplot(data=d1, x="label", stat="probability", color="skyblue");
```



In [60]:
```python
sns.histplot(data=d2, x="label", stat="probability", color="skyblue");
```



Chi-Square Testing

In [61]:
```python
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 0.0001373003673541072
difference between the two distributions (reject H0)
```

Analysis of Compliance

In [62]:
```python
def compliance(df):
    ids = []
    rounds = []
    server = []
    label = []
    for player in set(np.array(df["ResponseId"])):
        for i in np.arange(1,7):
            tem = df[(df["ResponseId"] == player) & (df["round"] == i)]
            if len(tem) != 0: # no response there
                ids.append(player)
                rounds.append(i)
                l = df[(df["ResponseId"] == player)]["label"].iloc[0]
                label.append(l)
                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 [63]:
```python
before_compliance = compliance(before)
```

In [64]:
```python
after_compliance = compliance(after)
```

In [65]:
```python
before_compliance
```

Out[65]:

	ResponseId	round	numServerCook	label
0	R_8nOXycxlDgZm6kx	3	2	5
1	R_8nOXycxlDgZm6kx	4	2	5
2	R_8nOXycxlDgZm6kx	5	1	5
3	R_8nOXycxlDgZm6kx	6	2	5
4	R_6JZr4lgBJ3Xo9gt	3	1	5
...	...	...	...	...
675	R_BYxHoWpYxArK6dP	6	3	4
676	R_1KiNAbM6l7sMKtU	3	2	3
677	R_1KiNAbM6l7sMKtU	4	2	3
678	R_1KiNAbM6l7sMKtU	5	2	3
679	R_1KiNAbM6l7sMKtU	6	2	3

680 rows × 4 columns

In [66]:
```python
after_compliance
```

Out[66]:

	ResponseId	round	numServerCook	label
0	R_1MX11huDrBtkGoB	3	2	5
1	R_1MX11huDrBtkGoB	4	2	5
2	R_1MX11huDrBtkGoB	5	2	5
3	R_1MX11huDrBtkGoB	6	2	5
4	R_1jYXGI34FVbha4S	3	2	3
...	...	...	...	...
975	R_ykkhkGYu1KplM8h	6	4	3
976	R_3gMLtEVUqUbNFXW	3	10	4
977	R_3gMLtEVUqUbNFXW	4	3	4
978	R_3gMLtEVUqUbNFXW	5	1	4
979	R_3gMLtEVUqUbNFXW	6	3	4

980 rows × 4 columns

In [ ]:

group by rounds

In [67]:
```python
round_before = before_compliance.groupby("round").agg(np.average).loc[:,["numServerCook"]]
round_before
```

Out[67]:

round	numServerCook
3	2.552941
4	2.670588
5	2.617647
6	2.500000

In [68]:
```python
round_after = after_compliance.groupby("round").agg(np.average).loc[:,["numServerCook"]]
round_after
```

Out[68]:

round	numServerCook
3	2.408163
4	2.232653
5	2.163265
6	2.044898

In [69]:
```python
a = np.array(round_before["numServerCook"])
```

In [70]:
```python
b = np.array(round_after["numServerCook"])
```

chi-square testing

In [71]:
```python
chisquare(a, b, count=False)
```

```
p value is 0.9994561534639331
no difference between the two distributions (H0 holds true)
```

In [ ]:

group by tip

In [72]:
```python
tip_before = before_compliance.groupby("label", as_index=False).agg(np.average).loc[:,["label","numServerCook"]]
tip_before
```

Out[72]:

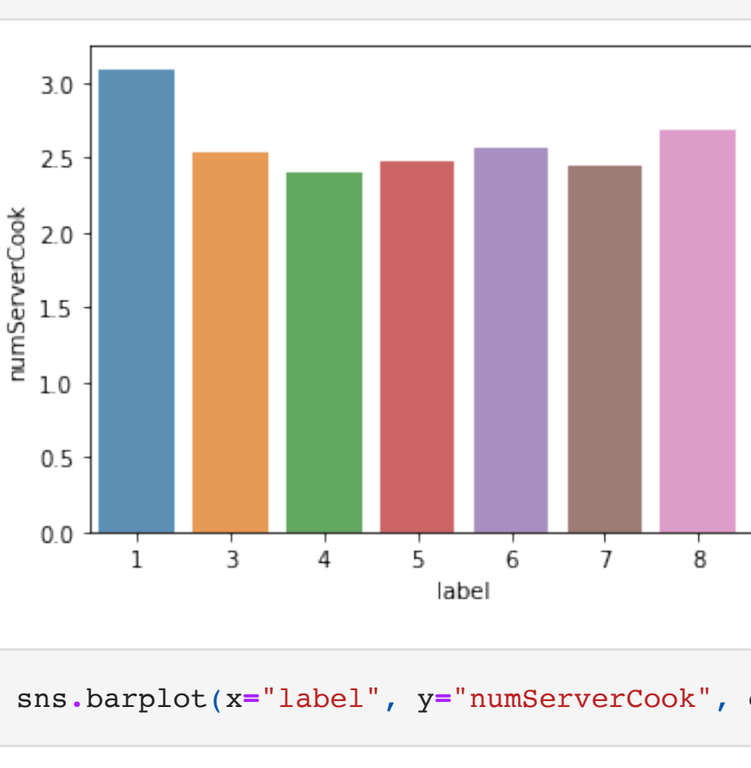
	label	numServerCook
0	1	3.093750
1	3	2.532258
2	4	2.403846
3	5	2.479167
4	6	2.571429
5	7	2.454545
6	8	2.687500
7	9	2.865385

In [73]:
```python
tip_after = after_compliance.groupby("label", as_index=False).agg(np.average).loc[:,["label","numServerCook"]]
tip_after
```

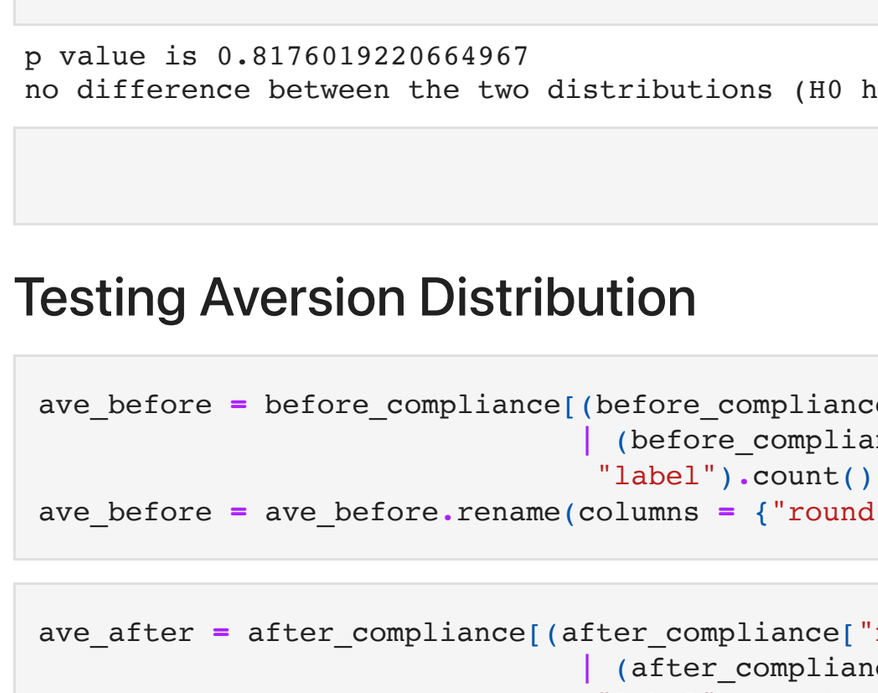
Out[73]:

	label	numServerCook
0	1	1.914062
1	2	2.375000
2	3	2.215789
3	4	2.382353
4	5	2.219828
5	6	2.166667
6	7	1.650000
7	8	2.000000
8	9	2.517241

In [74]:
```python
sns.barplot(x="label", y="numServerCook", data=tip_before, alpha=0.8);
```



In [75]:
```python
sns.barplot(x="label", y="numServerCook", data=tip_after, alpha=0.8);
```



chi-square testing

In [76]:
```python
np.array(tip_before["numServerCook"])
```

Out[76]:
```
array([3.09375   , 2.53225806, 2.40384615, 2.47916667, 2.57142857,
       2.45454545, 2.6875    , 2.86538462])
```

In [77]:
```python
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 0.8176019220664967
no difference between the two distributions (H0 holds true)
```

In [ ]:

Testing Aversion Distribution

In [78]:
```python
ave_before = before_compliance[(before_compliance["numServerCook"] >= 3)
                               | (before_compliance["numServerCook"] == 0)].groupby(
                                  "label").count().loc[:,["round"]]
ave_before = ave_before.rename(columns = {"round":"numAversion"})
```

In [79]:
```python
ave_after = after_compliance[(after_compliance["numServerCook"] >= 3)
                             | (after_compliance["numServerCook"] == 0)].groupby(
                                "label").count().loc[:,["round"]]
ave_after = ave_after.rename(columns = {"round":"numAversion"})
```

In [80]:
```python
def counting(df):
    count = []
    for i in np.arange(1,10):
        cou = len(df[df["label"] == i])
        count.append(cou)
    return count
```

In [81]:
```python
counting(before_compliance), counting(after_compliance)
[_ for _ in counting(before_compliance) if _]
```

Out[81]: [64, 248, 104, 96, 56, 44, 16, 52]

In [82]:
```python
ave_before['total_num_label'] = [_ for _ in counting(before_compliance) if _]
ave_before["proportion_aversion"] = ave_before["numAversion"] / ave_before['total_num_label']
ave_before["proportion_label"] = ave_before['total_num_label'] / before_compliance.shape[0]
ave_before
```

Out[82]:

label	numAversion	total_num_label	proportion_aversion	proportion_label
1	35	64	0.546875	0.094118
3	96	248	0.387097	0.364706
4	37	104	0.355769	0.152941
5	43	96	0.447917	0.141176
6	23	56	0.410714	0.082353
7	20	44	0.454545	0.064706
8	7	16	0.437500	0.023529
9	25	52	0.480769	0.076471

In [84]:
```python
ave_after['total_num_label'] = [_ for _ in counting(before_compliance) if _]
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[84]:

label	numAversion	total_num_label	proportion_aversion	proportion_label
1	26	64	0.406250	0.065306
2	3	248	0.012097	0.253061
3	108	104	1.038462	0.106122
4	21	96	0.218750	0.097959
5	57	56	1.017857	0.057143
6	5	44	0.113636	0.044898
7	2	16	0.125000	0.016327
9	41	52	0.788462	0.053061

In [ ]:

Testing Compliance Distribution

In [85]:
```python
com_before = before_compliance[before_compliance["numServerCook"] ==2].groupby(
                               "label").count().loc[:,["round"]]
com_before = com_before.rename(columns = {"round":"numCompliance"})
```

In [86]:
```python
com_after = after_compliance[after_compliance["numServerCook"] ==2].groupby(
                             "label").count().loc[:,["round"]]
com_after = com_after.rename(columns = {"round":"numCompliance"})
com_after
```

Out[86]:

label	numCompliance
1	61
2	4
3	193
4	39
5	130
6	18
7	12
8	4
9	51

In [87]:
```python
com_before['total_num_label'] = [_ 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[87]:

label	numCompliance	total_num_label	proportion_compliance	proportion_label
1	27	64	0.421875	0.094118
3	145	248	0.584677	0.364706
4	64	104	0.615385	0.152941
5	51	96	0.531250	0.141176
6	31	56	0.553571	0.082353
7	23	44	0.522727	0.064706
8	8	16	0.500000	0.023529
9	23	52	0.442308	0.076471

In [88]:
```python
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[88]:

label	numCompliance	total_num_label	proportion_compliance	proportion_label
1	61	128	0.476562	0.130612
2	4	8	0.500000	0.008163
3	193	380	0.507895	0.387755
4	39	68	0.573529	0.069388
5	130	232	0.560345	0.236735
6	18	24	0.750000	0.024490
7	12	20	0.600000	0.020408
8	4	4	1.000000	0.004082
9	51	116	0.439655	0.118367

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: