

Psychophysics: Experiment Report

The Role of Categorization in Visual Search for Orientation: 'Left' as a category

No. of Subjects who completely the survey satisfactorily : 8

Experiment Description

Experiment 1

 \wedge

□ p align="center"> Target = 'Left': -30 degree orientation.

 \wedge

☐ p align="center"> Distractors = 10, 90 degree

Experiment 2

 \wedge

□ p align="center"> Target = 'Steep': 10 degree.

 \wedge

□ p align="center"> Distractors = -50, 50 degree.

Experiment 3

 \wedge

□ p align="center"> Target = 'Steep-Left': -20 degree.

 \wedge

□□ p align="center"> Distractors = 20, -80 degree.

Average Reaction Time vs Set Size Plots

Reaction Time Statistics

Experiment 1

Set size	Mean	Median	Mode	Std. Dev.
6	679.575	620.5	629.412	299.859
12	727.127	616.0	559.46	389.43
18	817.25	687.0	545.45	360.0
24	895.038	776.0	573.33	407.857

Experiment 2

Set size	Mean	Median	Mode	Std. Dev.
6	728.125	564.0	474.074	363.399
12	940.962	784.0	680.0	507.805
18	886.387	670.5	575.0	619.54
24	821.887	686.5	588.88	453.911

Experiment 3

Set size	Mean	Median	Mode	Std. Dev.
6	874.112	657.0	477.777	530.9
12	1034.35	865.5	650.0	611.10
18	951.4	735.0	570.0	534.93
24	1033.425	786.0	655.55	546.257

Subject Reliability

No. of correct non-responses in target-absent trials:

Set size	Exp1	Exp2	Exp3
6	39	37	39
12	37	35	39

Set size	Exp1	Exp2	Exp3
24	40	37	38

No. of correct responses in target-present trials:

Set size	Exp1	Exp2	Exp3
6	80	80	79
12	79	79	80
18	80	80	79
24	80	80	77

Code Snippets

Reading Exp Data

```
data = pd.read_csv("data/data.csv", sep=",").values
exp_data = []
for i in range(len(data)):
    val = pd.read_csv("data/" + data[i, 4], sep=" ", header=None).values
    exp_data.append(val)
```

Preprocessing, separating trials by type and set size

```
data = []
for i in range(len(exp_data)):
    for j in range(len(exp_data[i])):
        data.append(exp_data[i][j][1:])

tar_sets = [[],[],[],[ ]]
emp_sets = [[],[],[],[ ]]
for i in range(len(data)):
    if(data[i][0] < 100):
        tar_sets[int((data[i][0]-6)/6)].append(data[i][1:])
    else:
        emp_sets[int((data[i][0]-106)/6)].append(data[i][1:])
```

Filtering correct trials

```

right_tar = [[],[],[],[ ]]
right_emp = [[],[],[],[ ]]
for i in range(len(tar_sets)):
    c = 0
    w = 0
    for j in range(len(tar_sets[i])):
        if(tar_sets[i][j][0] == 1):
            c = c + 1
            right_tar[i].append(tar_sets[i][j][1])
    for j in range(len(emp_sets[i])):
        if(emp_sets[i][j][0] != 1):
            w = w + 1
            right_emp[i].append(emp_sets[i][j][1])
print(c, w)

```

Getting statistics

```

for i in range(len(right_tar)):
    print(right_tar[i].sort())
    bins = [i for i in range(0, 4100, 100)]
    labels = ['{}-{}'.format(x, y-.1) for x, y in zip(bins[:], bins[1:])]
    frame = pd.Series(right_tar[i])
    ncut = pd.cut(frame, bins=bins, labels=labels, right=False)
    freq = lambda x: len(x) / x.sum()
    freq.__name__ = 'freq'
    out = pd.concat([ncut, frame], axis=1).groupby(0).agg(['size', 'std', 'mean', freq])
    print(frame.mean())
    print(frame.median())
    print(frame.std())
    print(out)

```