

# Visual Search Familiar Faces: Figures and Descriptive Statistics for Accuracy

*Matteo Visconti di Oleggio Castello*

## Contents

<b>Figures and Descriptive Statistics for Accuracy</b>	<b>1</b>
Setup . . . . .	1
Bootstrapping of the data . . . . .	2
Plot of Average Accuracy . . . . .	5
Compute Accuracy across conditions . . . . .	7

## Figures and Descriptive Statistics for Accuracy

This R Markdown file produces descriptive statistics and figures for accuracy.

### Setup

Load some useful libraries and load the data.

```
# return version information
version
```

```
##
## platform      x86_64-apple-darwin13.4.0
## arch          x86_64
## os            darwin13.4.0
## system        x86_64, darwin13.4.0
## status
## major         3
## minor         2.3
## year          2015
## month         12
## day           10
## svn rev       69752
## language      R
## version.string R version 3.2.3 (2015-12-10)
## nickname      Wooden Christmas-Tree
```

```
packages <- c('dplyr',
              'ggplot2',
              'doParallel',
              'foreach',
              'knitr')
```

```
for (package in packages) {
  require(package, character.only=T)
  cat(paste(package, packageVersion(package), '\n'))
}
```

```
## dplyr 0.4.3
## ggplot2 2.1.0
## doParallel 1.0.10
## foreach 1.4.3
## knitr 1.12.3
```

```
data <- read.csv('data.csv')
```

Set up some variables that will be used later.

```
nproc <- 4 # change this to use more/less processors for parallel use
seed <- 42 # seed for rng to obtain reproducible results in different runs
alpha <- .05 # significance level for confidence intervals
nbs <- 10000 # number of bootstrapping repetitions
```

## Bootstrapping of the data

Bootstrap data resampling within each condition. It could take a while depending on the number of processors used.

```
cl <- makeCluster(nproc)
registerDoParallel(cl)
# push required packages to each worker
clusterCall(cl, function() library(magrittr))
clusterCall(cl, function() library(plyr))
clusterCall(cl, function() library(dplyr))
bstrap <- data.frame()
set.seed(seed)
bstrap <- foreach(i = 1:nbs, .combine=rbind) %dopar% {
  tmp <-
    data %>%
    group_by(orientation, target_presence, familiarity,
              set_size, target_sex, subid) %>%
    sample_frac(1, replace=T) %>%
    group_by(orientation, target_presence, familiarity, set_size) %>%
    summarise(accuracy=sum(correct == 1)/n() * 100)

  tmp$index <- i
  tmp
}
stopCluster(cl)
```

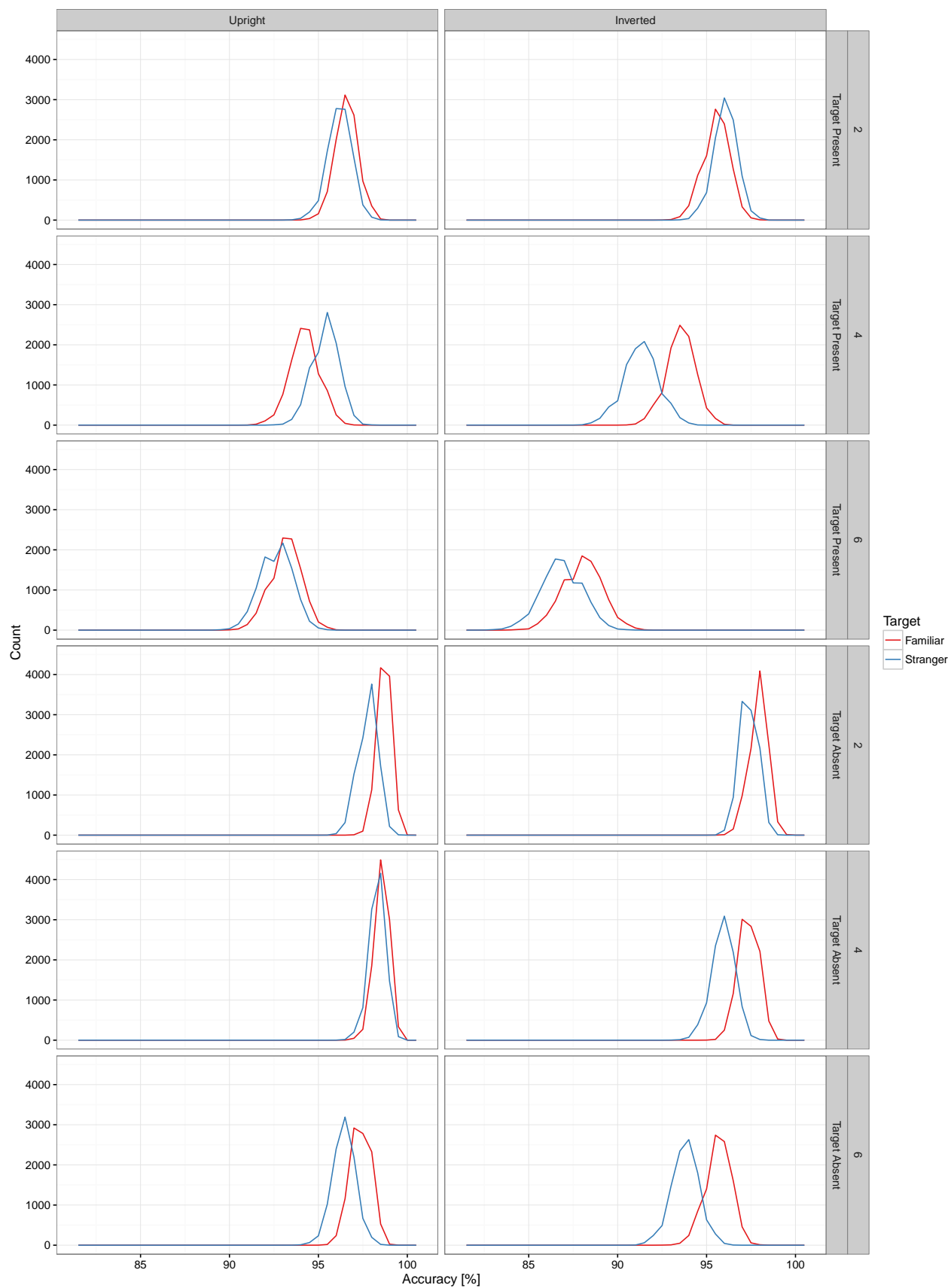
Visualize bootstrapping distribution

```

bstrap$orientation <- factor(bstrap$orientation,
                             levels=c('Upright', 'Inverted'))
bstrap$target_presence <- factor(bstrap$target_presence,
                                 levels=c('Target Present', 'Target Absent'))

ggplot(bstrap, aes(accuracy, color=familiarity)) +
  geom_freqpoly(binwidth=.5) +
  facet_grid(target_presence*set_size ~ orientation) +
  theme_bw(base_size=12) +
  labs(color='Target', x='Accuracy [%]', y='Count') +
  scale_color_brewer(palette='Set1')

```



## Plot of Average Accuracy

Now compute confidence intervals and averages from the bootstrapped samples.

```
cis <-
  bstrap %>%
  group_by(orientation, target_presence, familiarity, set_size) %>%
  summarise(low=quantile(accuracy, alpha/2),
            high=quantile(accuracy, 1-alpha/2))

# compute average from original data
avgs <- data %>%
  group_by(orientation, target_presence, familiarity, set_size) %>%
  summarise(accuracy=sum(correct == 1)/n() * 100)

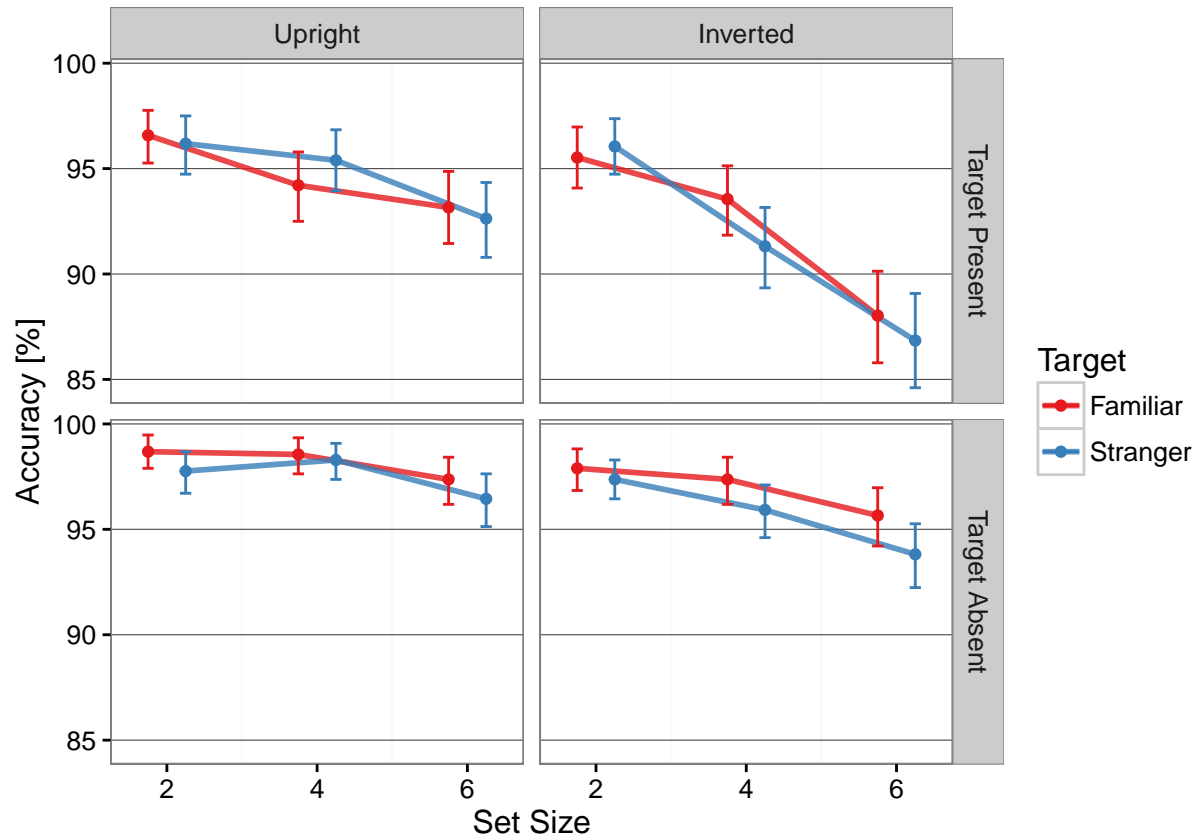
cis <- merge(avgs, cis)
```

Now plot the figure

```
# first reorder the levels of target_presence and orientation
cis$orientation <- factor(cis$orientation, levels=c('Upright', 'Inverted'))
cis$target_presence <- factor(cis$target_presence,
                             levels=c('Target Present', 'Target Absent'))

pd = position_dodge(width=1)

ggplot(cis, aes(set_size, accuracy, ymin=low, ymax=high,
               color=familiarity)) +
  geom_line(alpha=.8, size=1, position=pd) +
  geom_errorbar(width=0.3, position=pd) +
  geom_point(position=pd) +
  facet_grid(target_presence ~ orientation) +
  theme_bw(base_size=12) +
  theme(panel.grid.major.y = element_line(colour = "gray30"),
        panel.grid.major.x = element_blank(),
        panel.grid.minor.y = element_blank()) +
  scale_x_continuous(breaks=c(2, 4, 6)) +
  labs(x='Set Size', y='Accuracy [%]', color='Target',
       linetype='Orientation') +
  scale_color_brewer(palette='Set1')
```



These are the plotted values

```
kable(cis, digits=2)
```

orientation	target_presence	familiarity	set_size	accuracy	low	high
Inverted	Target Absent	Familiar	2	97.89	96.84	98.82
Inverted	Target Absent	Familiar	4	97.37	96.18	98.42
Inverted	Target Absent	Familiar	6	95.66	94.21	96.97
Inverted	Target Absent	Stranger	2	97.37	96.45	98.29
Inverted	Target Absent	Stranger	4	95.92	94.61	97.11
Inverted	Target Absent	Stranger	6	93.82	92.24	95.26
Inverted	Target Present	Familiar	2	95.53	94.08	96.97
Inverted	Target Present	Familiar	4	93.55	91.84	95.13
Inverted	Target Present	Familiar	6	88.03	85.79	90.13
Inverted	Target Present	Stranger	2	96.05	94.74	97.37
Inverted	Target Present	Stranger	4	91.32	89.34	93.16
Inverted	Target Present	Stranger	6	86.84	84.61	89.08
Upright	Target Absent	Familiar	2	98.68	97.89	99.47
Upright	Target Absent	Familiar	4	98.55	97.63	99.34
Upright	Target Absent	Familiar	6	97.37	96.18	98.42
Upright	Target Absent	Stranger	2	97.76	96.71	98.68
Upright	Target Absent	Stranger	4	98.29	97.37	99.08
Upright	Target Absent	Stranger	6	96.45	95.13	97.63
Upright	Target Present	Familiar	2	96.58	95.26	97.76
Upright	Target Present	Familiar	4	94.21	92.50	95.79
Upright	Target Present	Familiar	6	93.16	91.45	94.87

orientation	target_presence	familiarity	set_size	accuracy	low	high
Upright	Target Present	Stranger	2	96.18	94.74	97.50
Upright	Target Present	Stranger	4	95.39	93.95	96.84
Upright	Target Present	Stranger	6	92.63	90.79	94.34

## Compute Accuracy across conditions

Accuracy according to target presence.

```
accs_bytp <-
  avgs %>%
  group_by(target_presence) %>%
  summarise(accuracy=mean(accuracy))

cis_bytp <-
  bstrap %>% group_by(index, target_presence) %>%
  summarise(accuracy=mean(accuracy)) %>%
  group_by(target_presence) %>%
  summarise(low=quantile(accuracy, alpha/2),
            high=quantile(accuracy, 1-alpha/2))

bytp <- merge(accs_bytp, cis_bytp)

kable(bytp, digits=2)
```

target_presence	accuracy	low	high
Target Absent	97.09	96.78	97.41
Target Present	93.29	92.80	93.78

Accuracy according to target orientation.

```
accs_byto <-
  avgs %>%
  group_by(orientation) %>%
  summarise(accuracy=mean(accuracy))

cis_byto <-
  bstrap %>% group_by(index, orientation) %>%
  summarise(accuracy=mean(accuracy)) %>%
  group_by(orientation) %>%
  summarise(low=quantile(accuracy, alpha/2),
            high=quantile(accuracy, 1-alpha/2))

byto <- merge(accs_byto, cis_byto)

kable(byto, digits=2)
```

orientation	accuracy	low	high
Inverted	94.11	93.65	94.56
Upright	96.27	95.90	96.64

Accuracy according to set size.

```
accs_byss <-  
  avgs %>%  
  group_by(set_size) %>%  
  summarise(accuracy=mean(accuracy))  
cis_byss <-  
  bstrap %>% group_by(index, set_size) %>%  
  summarise(accuracy=mean(accuracy)) %>%  
  group_by(set_size) %>%  
  summarise(low=quantile(accuracy, alpha/2),  
            high=quantile(accuracy, 1-alpha/2))  
  
byss <- merge(accs_byss, cis_byss)  
  
kable(byss, digits=2)
```

set_size	accuracy	low	high
2	97.01	96.60	97.42
4	95.58	95.07	96.05
6	92.99	92.40	93.59

Accuracy according to familiarity.

```
accs_byfam <-  
  avgs %>%  
  group_by(familiarity) %>%  
  summarise(accuracy=mean(accuracy))  
cis_byfam <-  
  bstrap %>% group_by(index, familiarity) %>%  
  summarise(accuracy=mean(accuracy)) %>%  
  group_by(familiarity) %>%  
  summarise(low=quantile(accuracy, alpha/2),  
            high=quantile(accuracy, 1-alpha/2))  
  
byfam <- merge(accs_byfam, cis_byfam)  
  
kable(byfam, digits=2)
```

familiarity	accuracy	low	high
Familiar	95.55	95.14	95.93
Stranger	94.84	94.41	95.26

Accuracy according to target presence and set size.

```
accs_byssXtp <-  
  avgs %>%  
  group_by(target_presence, set_size) %>%  
  summarise(accuracy=mean(accuracy))  
cis_byssXtp <-
```



```

bstrap %>% group_by(index, set_size, target_presence) %>%
  summarise(accuracy=mean(accuracy)) %>%
  group_by(target_presence, set_size) %>%
  summarise(low=quantile(accuracy, alpha/2),
            high=quantile(accuracy, 1-alpha/2))

byssXtp <- merge(accs_byssXtp, cis_byssXtp)

kable(byssXtp, digits=2)

```

target_presence	set_size	accuracy	low	high
Target Absent	2	97.93	97.47	98.39
Target Absent	4	97.53	97.01	98.03
Target Absent	6	95.82	95.16	96.48
Target Present	2	96.09	95.43	96.74
Target Present	4	93.62	92.76	94.41
Target Present	6	90.16	89.14	91.15

Accuracy according to target orientation and target presence.

```

accs_bytoXtp <-
  avgs %>%
  group_by(target_presence, orientation) %>%
  summarise(accuracy=mean(accuracy))
cis_bytoXtp <-
  bstrap %>% group_by(index, target_presence, orientation) %>%
  summarise(accuracy=mean(accuracy)) %>%
  group_by(target_presence, orientation) %>%
  summarise(low=quantile(accuracy, alpha/2),
            high=quantile(accuracy, 1-alpha/2))

bytoXtp <- merge(accs_bytoXtp, cis_bytoXtp)

kable(bytoXtp, digits=2)

```

target_presence	orientation	accuracy	low	high
Target Absent	Inverted	96.34	95.83	96.82
Target Absent	Upright	97.85	97.43	98.25
Target Present	Inverted	91.89	91.12	92.63
Target Present	Upright	94.69	94.06	95.31

Accuracy according to familiarity and target presence.

```

accs_byfamXtp <-
  avgs %>%
  group_by(target_presence, familiarity) %>%
  summarise(accuracy=mean(accuracy))
cis_byfamXtp <-
  bstrap %>% group_by(index, target_presence, familiarity) %>%

```

```

summarise(accuracy=mean(accuracy)) %>%
group_by(target_presence, familiarity) %>%
summarise(low=quantile(accuracy, alpha/2),
          high=quantile(accuracy, 1-alpha/2))

byfamXtp <- merge(accs_byfamXtp, cis_byfamXtp)

kable(byfamXtp, digits=2)

```

target__presence	familiarity	accuracy	low	high
Target Absent	Familiar	97.59	97.15	98.00
Target Absent	Stranger	96.60	96.10	97.08
Target Present	Familiar	93.51	92.83	94.17
Target Present	Stranger	93.07	92.37	93.77