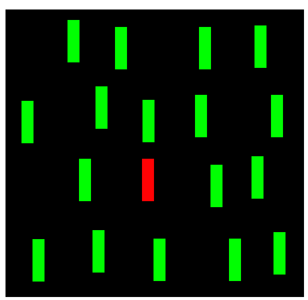
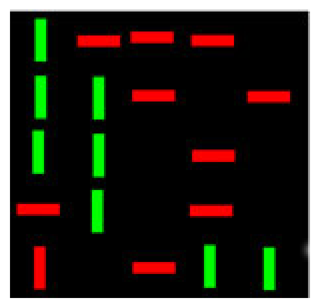
Demo 1: Search slopes

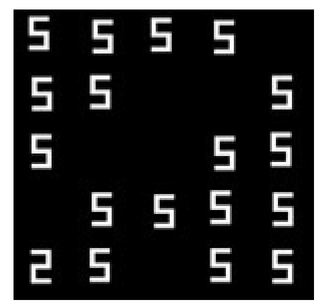
- Experiment data from: http://search.bwh.harvard.edu/new/data_set_files.html
- Three visual search tasks:



Feature



Conjunction



Configuration

Load the data

```
data <- read.table("your_path/PSYC6229_search_data.csv",
header=TRUE, sep=",")

View(data)
summary(data)

unique(data$Condition)
unique(data$Setsize)

summary(subset(data, Condition == "R_vs_G"))
summary(subset(data, Condition == "RV_vs_RHGV"))
summary(subset(data, Condition == "2_vs_5"))
```

RT distributions

```
par(mfrow=c(3,4))
for (i in unique(data$Condition)) {
    for (j in sort(unique(data$Setsize))) {
        hist(data$RT[data$Condition==i & data$Setsize==j],
        xlab="RT in msec",
        main=paste(i,", set size = ",j),
        col="blue")
    }
}
```

It’s hard to see the details. How about looking only at the fastest (< 2 sec) RTs?

Pattern of errors

```
aggregate(data$Error, list(data$Condition,data$Setsize), mean)

dev.new()
par(mfrow=c(3,4))
for (i in unique(data$Condition)) {
    for (j in sort(unique(data$Setsize))) {
        tmp <- data$Type[data$Condition==i & data$Setsize==j]
        counts <- table(tmp)
        barplot(100*counts[c("FA","MISS")]/sum(counts),
            ylim = c(0,5),
            main=paste(i,", set size = ",j))
    }
}
```

This is okay for investigating trends, but it’s not how you’d graph errors for a paper (by subjects, with error bars).

Pattern of errors (by subjects)

```
par(mfrow=c(3,4))
for (i in unique(data$Condition)) {
   for (j in sort(unique(data$Setsize))) {
      tmp <- data[data$Condition==i & data$Setsize==j,]
      bySub <- aggregate(tmp$Error,
list(tmp$Subject,tmp$Tpresent), mean)
      meansBySub <- aggregate(bySub$x, list(bySub$Group.2),
mean)
      means <- 100 * meansBySub$x
      sds <- aggregate(bySub$x, list(bySub$Group.2), sd)
      sderr <- 100 * sds$x / sqrt(length(unique(bySub$Group.1)))
      barHeights <- barplot(means, names.arg = list("FA","MISS"),
         ylim = c(0,12), main=paste(i,", set size = ",j))
      arrows(barHeights, means - sderr, barHeights, means + sderr,
         lwd = 1.5, angle = 90, code = 3, length = 0.05)
   }
}
```

d prime (by subjects)

```
par(mfrow=c(3,4))
for (i in unique(data$Condition)) {
   for (j in sort(unique(data$Setsize))) {
      tmp <- data[data$Condition==i & data$Setsize==j,]
      bySub <- aggregate(tmp$Error,
        list(tmp$Subject,tmp$Tpresent), mean)
      bySub$x[bySub$x < 0.005] <- 0.005
      bySub$x[bySub$x > 0.995] <- 0.995
      zHIT <- qnorm(1-bySub$x[bySub$Group.2==1])
      zFA <- qnorm(bySub$x[bySub$Group.2==0])
      dprimeBySub <- zHIT - zFA
      barHeights <- barplot(dprimeBySub, names.arg =
         unique(bySub$Group.1), ylim = c(0,7),
         main=paste(i,", set size = ",j))
   }
}
```

Are RT distributions normal?

```
p <- matrix(ncol=5, nrow=4*29*2)
colnames(p) <- list("cond", "setsize", "subject", "tpres", "pval")
counter <- 1
for (i in unique(data$Condition)) {
    for (j in sort(unique(data$Setsize))) {
        for (ii in unique(data$Subject[data$Condition==i &
data$Setsize==j])) {
            for (jj in unique(data$Tpresent)) {
                tmp <- data[data$Condition==i & data$Setsize==j &
data$Subject==ii & data$Tpresent==jj,]
                result <- shapiro.test(tmp$RT)
                p[counter,] <- c(i, j, ii, jj, result$p.value)
                counter <- counter + 1
}}}}
testresults = as.data.frame(p)
View(testresults)
```

RT trimming

```
flag <- 1
for (i in unique(data$Condition)) {
   for (j in sort(unique(data$Setsize))) {
      for (ii in unique(data$Subject[data$Condition==i &
data$Setsize==j])) {
         for (jj in unique(data$Tpresent)) {
            tmp <- data[data$Condition==i & data$Setsize==j &
data$Subject==ii & data$Tpresent==jj,]
            thHigh <- mean(tmp$RT) + (2.5*sd(tmp$RT))
            thLow <- mean(tmp$RT) - (2.5*sd(tmp$RT))
            new <- tmp[tmp$RT > thLow & tmp$RT < thHigh,]
            if (flag==1) {
               data2 <- new
               flag <- 0
            } else {
               data2 <- rbind(data2, new) }
}}}}
```

Although this is a standard technique to make RT distributions normal, it does have some issues:

Miller, J. (1991). Reaction time analysis with outlier exclusion: Bias varies with sample size. The Quarterly Journal of Experimental Psychology, 43A(4), 907-912.

Search slopes

```
x <- sort(unique(data$Setsize))
for (i in unique(data$Condition)) {
   dev.new()
   tmp <- data[data$Condition==i,]
   bySub <- aggregate(tmp$RT,
      list(tmp$Subject,tmp$Tpresent,tmp$Setsize), mean)
   means <- aggregate(bySub$x,
      list(bySub$Group.2, bySub$Group.3), mean)
   means <- means[order(means$Group.2),]
   y0 = means$x[means$Group.1==0]
   y1 = means$x[means$Group.1==1]
   plot(x, y0, col="blue", ylim=c(0,2500),
      xlab="Set size", ylab="RT (ms)", main=i)
   points(x, y1, col="red")
   regressline <- lm(y0 ~ x)
   abline(regressline)
}
```

Discussion

- Does the difference in conditions match the predictions of the two-stage model?
- How do target-absent slopes relate to target-present?
- Is a linear fit correct for this data?

Demo 2: SDT simulation

```
dprime <- 2

targs <- data.frame(response=rnorm(500, mean=(dprime/2), sd=1))
dists <- data.frame(response=rnorm(500, mean=(-dprime/2), sd=1))
targs$label <- "Target"
dists$label <- "Distractor"
stimuli <- rbind(targs,dists)
ggplot(stimuli, aes(response, fill=label)) + geom_density(alpha=0.5)
```

How does the plot change as you change d prime?

Maximum Rule

```
dprime <- 2
setsize <- 18
targs <- matrix(ncol=1, nrow=500)
dists <- matrix(ncol=1, nrow=500)
accuracy <- matrix(ncol=1, nrow=500)
```

Maximum Rule

```
counter <- 1
while (counter <= nrow(targs)) {
    nitems <- setsize
    tsample <- c(rnorm(1, mean=(dprime/2), sd=1),
        rnorm(nitems-1, mean=(-dprime/2), sd=1))
    dsample <- rnorm(nitems, mean=(-dprime/2), sd=1)
    targs[counter,] <- max(tsample)
    dists[counter,] <- max(dsample)
    accuracy[counter,] <- (max(tsample) > max(dsample))
    counter <- counter+1
}
targs <- data.frame(response=targs, label="Tpresent")
dists <- data.frame(response=dists, label="Tabsent")
stimuli <- rbind(targs,dists)
ggplot(stimuli, aes(response, fill=label)) + geom_density(alpha=0.5)
mean(accuracy)
```

Maximum Rule over set sizes

```
dprime <- 2
setsize <- c(3,6,12,18)
targs <- matrix(ncol=4, nrow=500)
dists <- matrix(ncol=4, nrow=500)
```

Maximum Rule over set sizes

```
for (i in 1:length(setsize)) {
    counter <- 1
    nitems <- setsize[i]
    while (counter <= nrow(targs)) {
        tsample <- c(rnorm(1, mean=(dprime/2), sd=1),
            rnorm(nitems-1, mean=(-dprime/2), sd=1))
        dsample <- rnorm(nitems, mean=(-dprime/2), sd=1)
        targs[counter,i] <- max(tsample)
        dists[counter,i] <- max(dsample)
        counter <- counter+1
    }

…
```

Maximum Rule over set sizes

```
…

    targtmp <- data.frame(response=targs[,i], label="Tpresent")
    disttmp <- data.frame(response=dists[,i], label="Tabsent")
    stimuli <- rbind(targtmp,disttmp)
    ggp <- ggplot(stimuli, aes(response, fill=label)) +
geom_density(alpha=0.5) + ggtitle(paste("Setsize = ",nitems))
    dev.new()
    print(ggp)
}
```

Accuracy over set size

```
diffMeans <- colMeans(targs) - colMeans(dists)
sds <- sqrt(( (apply(targs,2,sd)^2) + (apply(dists,2,sd)^2) ) / 2)
dSetsize <- diffMeans / sds

dev.new()
plot(setsize, dSetsize, xlab="Set size", ylab="Simulated d prime",
   main=paste("d prime = ",dprime))
```

RT over set size

```
nonsearchRT <- 400
beta <- 500
maxDprime <- 6.2
simulatedRT <- nonsearchRT + (beta *
   (1 - (log(dSetsize)/log(maxDprime))) )

dev.new()
plot(setsize, simulatedRT, xlab="Set size", ylab="RT (msec)",
   main=paste("Simulated RT over set size, d prime = ",dprime))
regressline <- lm(simulatedRT ~ setsize)
abline(regressline)
summary(regressline)
```

Loosely based on speed-accuracy tradeoff from:

McElree, B. & Carrasco, M. (1999). The temporal dynamics of visual search: Evidence for parallel processing in feature and conjunction searches. Journal of Experimental Psychology: Human Perception & Performance, 25(6), 1517-1539.