A guide to robust statistical methods in neuroscience: Figure 7

Rand R. Wilcox & Guillaume A. Rousselet

2023-03-02

Contents

Receptive Fields in Early Visual Cortex

These analyzes are based on data analyzed by Talebi and Baker (2016). The goal was to estimate visual receptive field (RF) models of cortical neurons using visually rich natural image stimuli and regularized regression system identification methods and characterize their spatial tuning, temporal dynamics, spatiotemporal behavior, and spiking properties.

Dependencies

```
library(ggplot2)
library(cowplot)
library(tibble)
source("./code/Rallfun-v40.txt")
source("./code/theme_gar.txt")
```

load data & set parameters

```
talebi = read.csv('./data/talebi-jn-2016-fig9.csv',header=T)
```

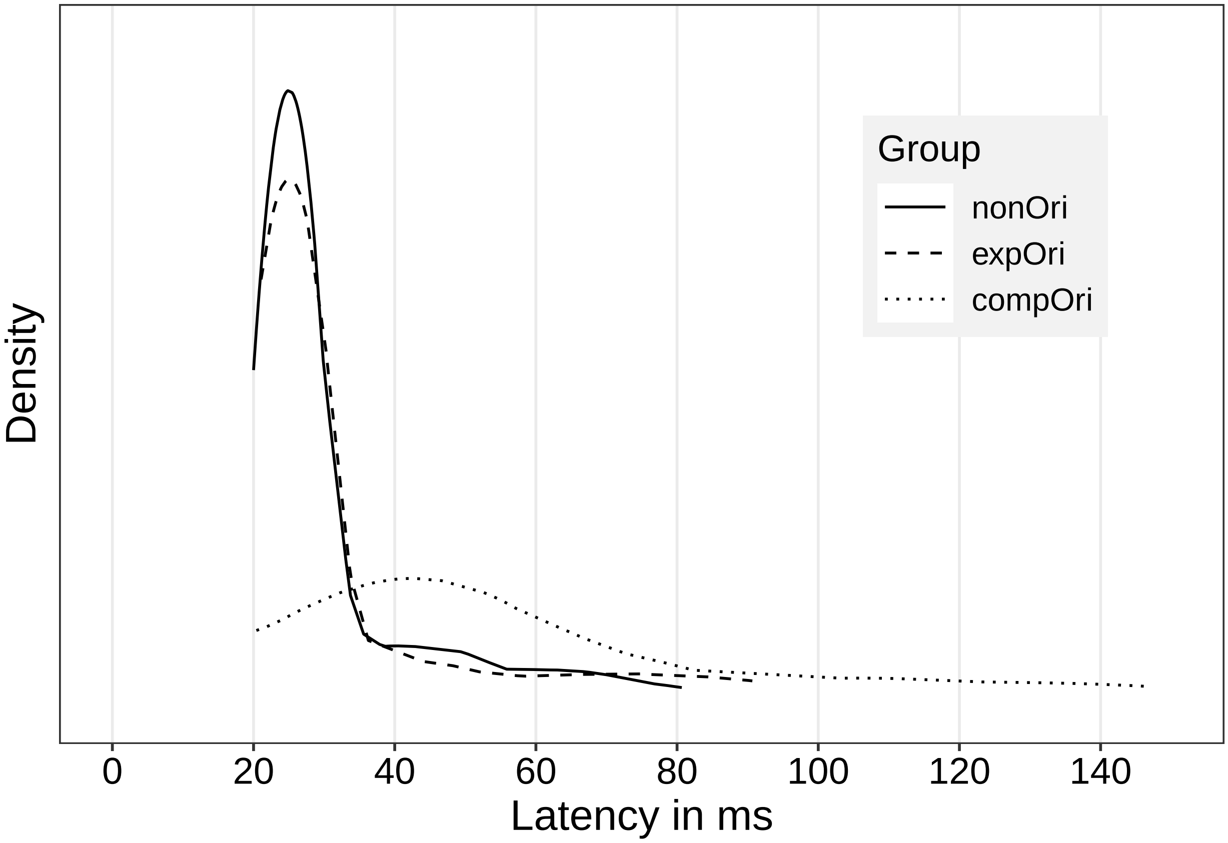
```
# parameters for kernel density estimation (KDE)
fr <- .8
aval <- .5

# parameters for plots
axis_size <- 14
axis_title_size <- 16

# number of bootstrap samples
nboot <- 2000
```

panel A: KDE + SF of LATENCY data

```
# get latency data
nonOri <- talebi[,1]
expOri <- talebi[,2]
compOri <- talebi[,3]

# KDE
# g5plot(talebi[,1:3],xlab='Latency')
z1 <- akerd(nonOri,aval=aval,fr=fr,pyhat=TRUE,plotit=FALSE)
z2 <- akerd(expOri,aval=aval,fr=fr,pyhat=TRUE,plotit=FALSE)
z3 <- akerd(compOri,aval=aval,fr=fr,pyhat=TRUE,plotit=FALSE)
# make data frame
Group <- factor(c(rep("nonOri", length(z1)),
                  rep("expOri", length(z2)),
                  rep("compOri", length(z3))))
Latency <- c(sort(nonOri), sort(expOri), sort(compOri))
Density <- c(z1, z2, z3)
df <- tibble(Group, Latency, Density)
df$Group <- factor(df$Group, levels = c("nonOri", "expOri", "compOri"))

kdeA <- ggplot(df, aes(Latency, Density, group=Group)) + theme_gar +
  geom_line(aes(linetype=Group, colour=Group), size=0.5) +
  scale_linetype_manual(values = c("solid","dashed","dotted")) +
  scale_color_manual(values = c('black','black','black')) +
  scale_x_continuous(breaks = seq(0,160,20),
                     minor_breaks = NULL,# seq(10,150,10),
                     limits = c(0, 150)) +
  scale_y_continuous(breaks = seq(0,0.07,0.01),
                     minor_breaks = NULL,
                     limits = c(0, 0.07)) +
  theme(axis.text = element_text(size = axis_size),
        axis.title = element_text(size = axis_title_size),
        legend.key.width = unit(1,"cm"),
        legend.position = c(.9, .7),
        legend.justification = 'right',
        legend.background = element_rect(fill="gray95"),
        legend.text = element_text(size = 12),
        legend.title = element_text(size = 14),
        axis.ticks.y = element_blank(),
        axis.text.y = element_blank(),
        panel.grid.major.y = element_blank()) +
  xlab("Latency in ms") +
  ylab("Density")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
```

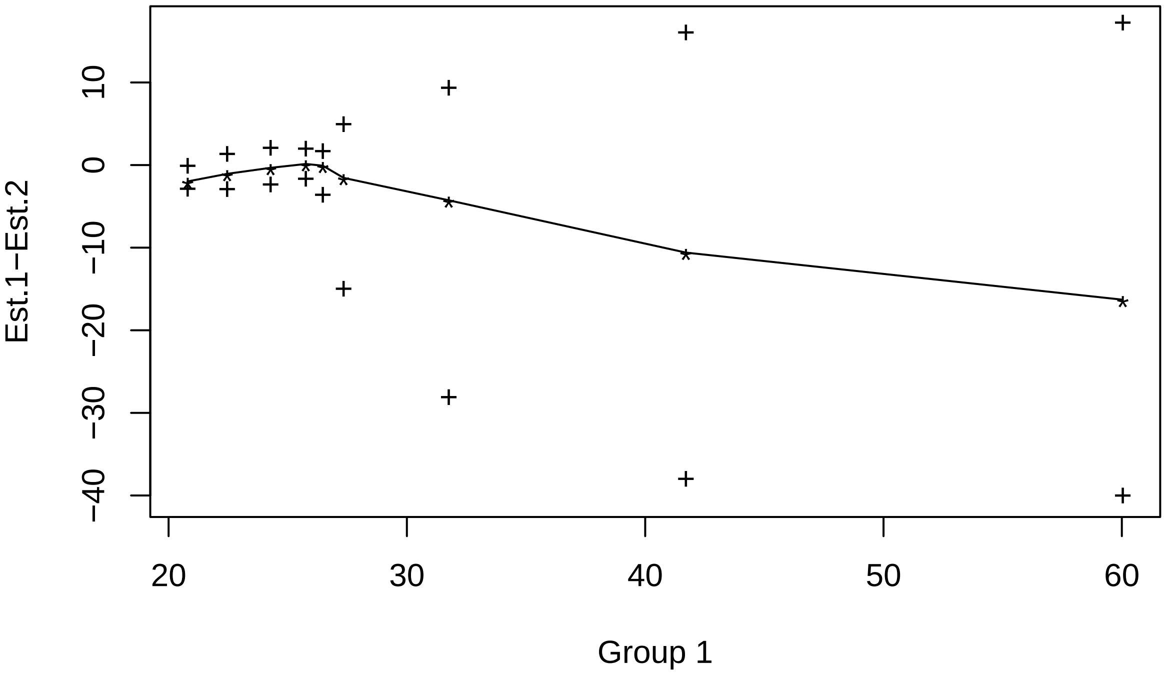
```
kdeA
```



shift function 1: nonOri vs. expOri

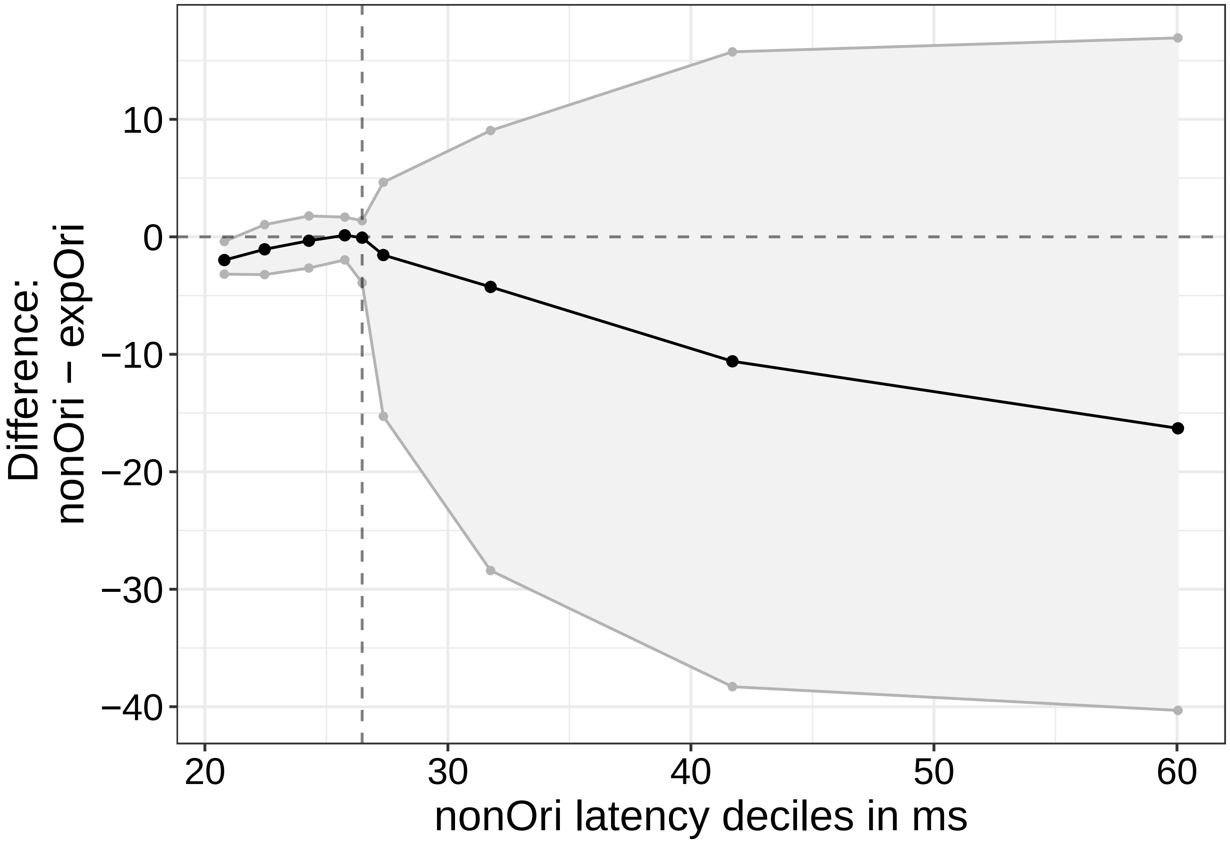
Compute and base R plot

```
sf1 <- qcomhd(nonOri,expOri,q=c(1:9)/10,nboot=nboot,plotit=TRUE)
```



ggplot2 version

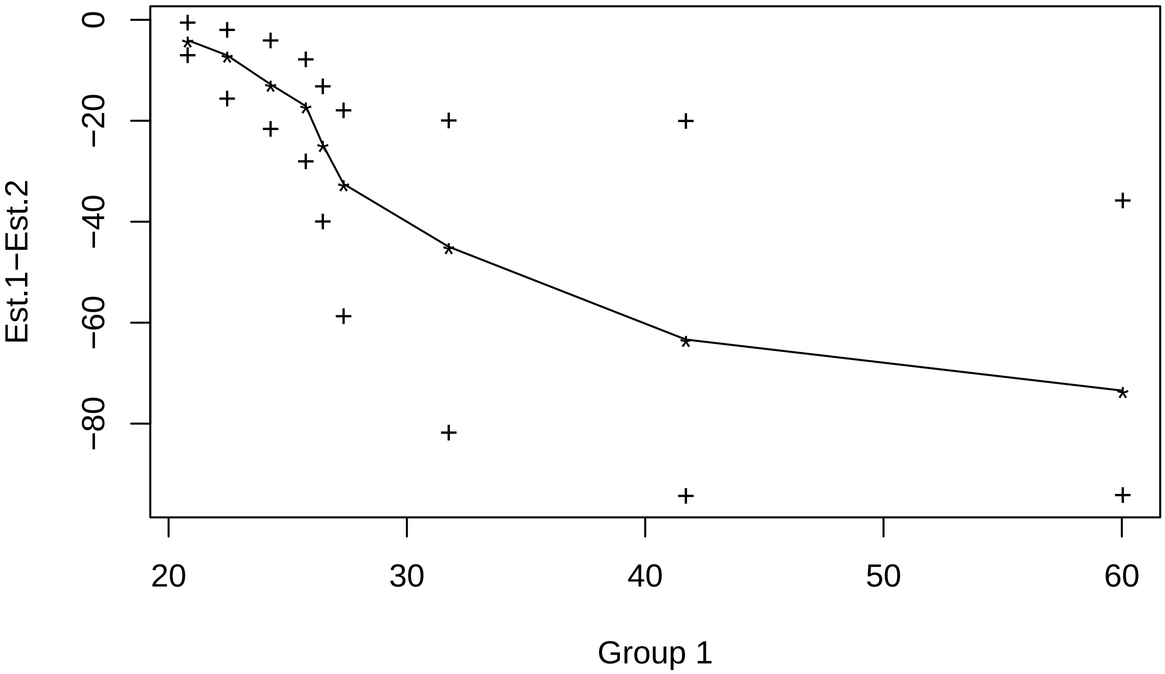
```r
# make data frame
dsf1 <- tibble(`Group1` = sf1[,"est.1"],
               `Difference` = sf1[,"est.1_minus_est.2"],
               `lower` = sf1[,"ci.low"],
               `upper` = sf1[,"ci.up"])
# make plot
xintercept <- dsf1[[5,1]]
sf1A <- ggplot(dsf1, aes(x = Group1, y = Difference)) +  theme_gar +
        # theme_bw() +
        geom_ribbon(aes(ymin = lower, ymax = upper), fill = "grey95") +
        geom_line(aes(y = lower), colour = "grey70") +
        geom_point(aes(y = lower), colour = "grey70", size = 1) +
        geom_line(aes(y = upper), colour = "grey70") +
        geom_point(aes(y = upper), colour = "grey70", size = 1) +
        geom_line(aes(y = Difference)) +
        geom_point(aes(y = Difference), size = 1.5) +
        geom_hline(yintercept = 0, linetype = 2, alpha = 0.5) +
        geom_vline(xintercept = xintercept, linetype = 2, alpha = 0.5) +
        theme(axis.text = element_text(size = axis_size),
          axis.title = element_text(size = axis_title_size)) +
        scale_y_continuous(breaks = seq(-40,20,10)) +
        xlab("nonOri latency deciles in ms") +
        ylab("Difference: \nnonOri - expOri")
sf1A
```



shift function 2: nonOri vs. compOri

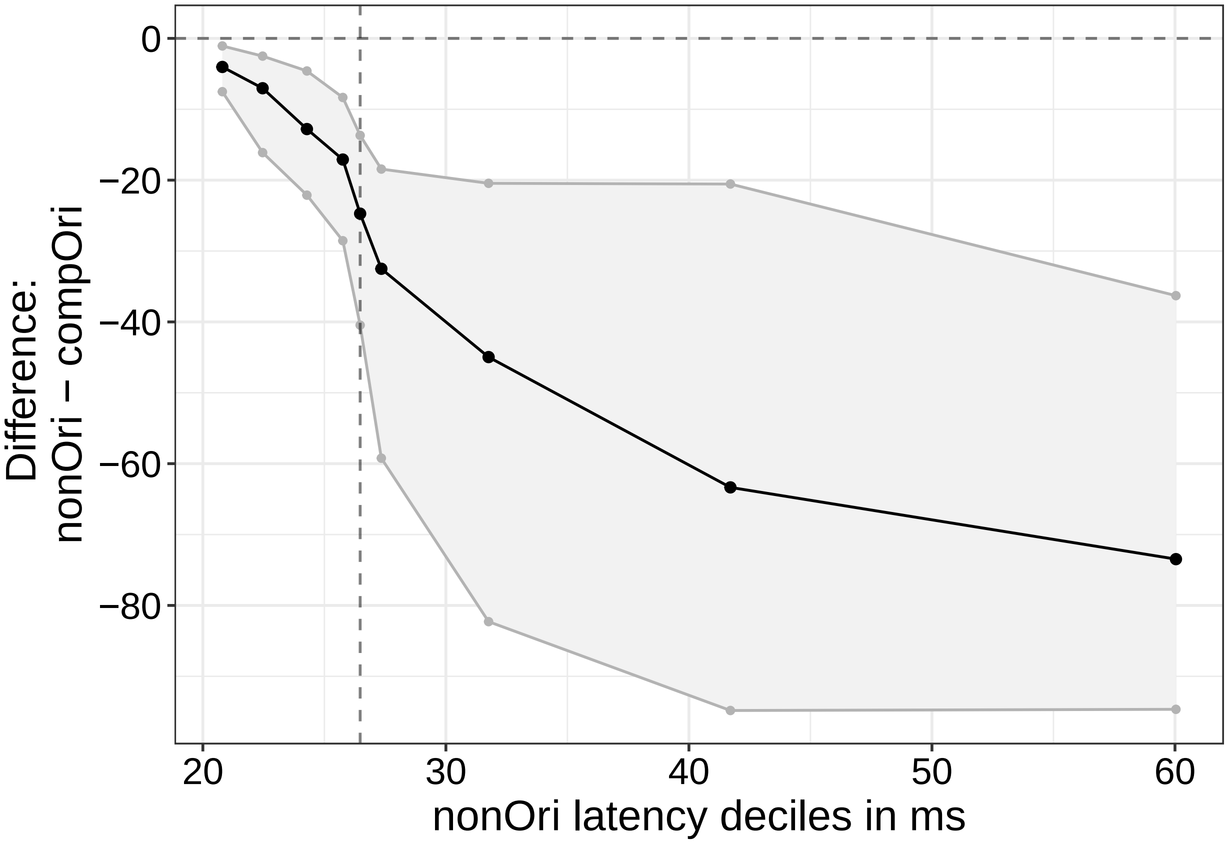
Compute and base R plot

```
sf2 <- qcomhd(nonOri,compOri,q=c(1:9)/10,nboot=nboot,plotit=TRUE)
```



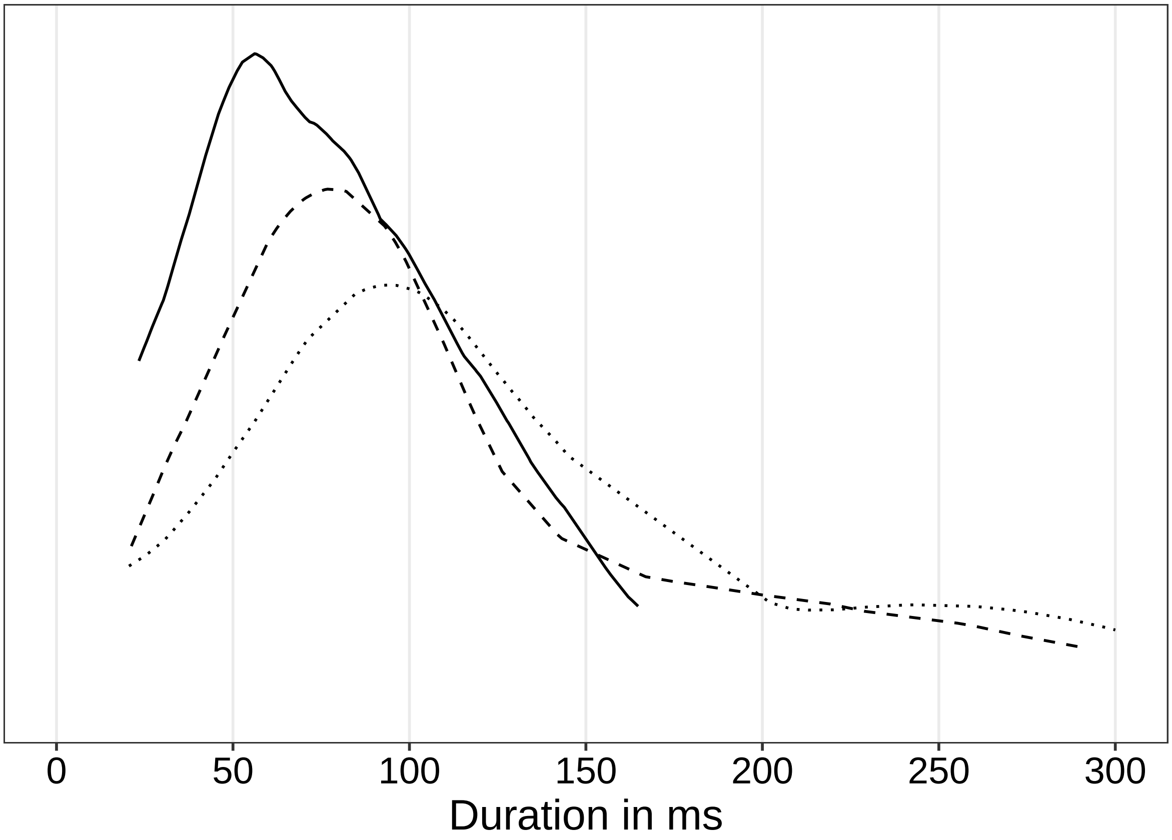
ggplot2 version

```r
# make data frame
dsf2 <- tibble(`Group1` = sf2[,"est.1"],
  `Difference` = sf2[,"est.1_minus_est.2"],
  `lower` = sf2[,"ci.low"],
  `upper` = sf2[,"ci.up"])
# make plot
xintercept <- dsf2[[5,1]]
sf2A <- ggplot(dsf2, aes(x = Group1, y = Difference)) + theme_gar +
  # theme_bw() +
  geom_ribbon(aes(ymin = lower, ymax = upper), fill = "grey95") +
  geom_line(aes(y = lower), colour = "grey70") +
  geom_point(aes(y = lower), colour = "grey70", size = 1) +
  geom_line(aes(y = upper), colour = "grey70") +
  geom_point(aes(y = upper), colour = "grey70", size = 1) +
  geom_line(aes(y = Difference)) +
  geom_point(aes(y = Difference), size = 1.5) +
  geom_hline(yintercept = 0, linetype = 2, alpha = 0.5) +
  geom_vline(xintercept = xintercept, linetype = 2, alpha = 0.5) +
  theme(axis.text = element_text(size = axis_size),
    axis.title = element_text(size = axis_title_size)) +
  scale_y_continuous(breaks = seq(-100,0,20)) +
  xlab("nonOri latency deciles in ms") +
  ylab("Difference: \nnonOri - compOri")
sf2A
```



panel B: KDE + SF of DURATION data

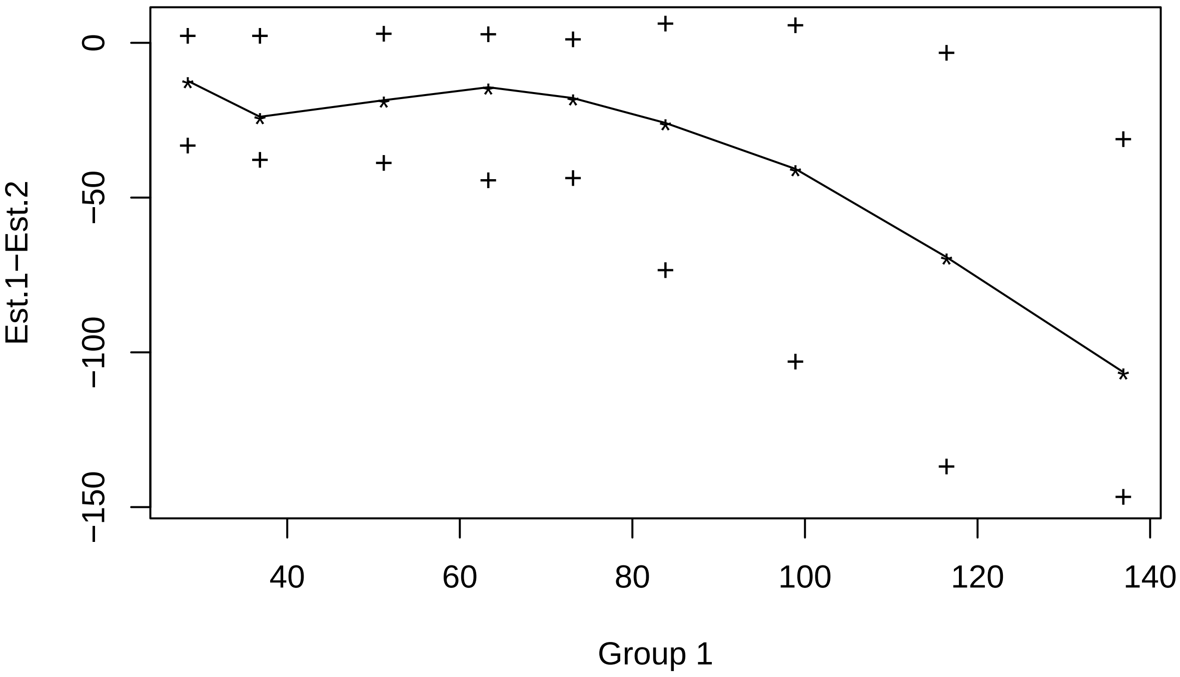
```
# get latency data
nonOri <- talebi[,4]
expOri <- talebi[,5]
compOri <- talebi[,6]

# KDE
# g5plot(talebi[,4:6],xlab='Duration')
z1 <- akerd(nonOri,aval=aval,fr=fr,pyhat=TRUE,plotit=FALSE)
z2 <- akerd(expOri,aval=aval,fr=fr,pyhat=TRUE,plotit=FALSE)
z3 <- akerd(compOri,aval=aval,fr=fr,pyhat=TRUE,plotit=FALSE)
# make data frame
Group <- factor(c(rep("nonOri", length(z1)),
  rep("expOri", length(z2)),
  rep("compOri", length(z3))))
Latency <- c(sort(nonOri), sort(expOri), sort(compOri))
Density <- c(z1, z2, z3)
df <- tibble(Group, Latency, Density)
df$Group <- factor(df$Group, levels = c("nonOri", "expOri", "compOri"))

kdeB <- ggplot(df, aes(Latency, Density, group=Group)) + theme_gar +
  geom_line(aes(linetype=Group, colour=Group), size=0.5) +
  scale_linetype_manual(values = c("solid","dashed","dotted")) +
  scale_color_manual(values = c('black','black','black')) +
  scale_x_continuous(breaks = seq(0,300,50),
    minor_breaks = NULL,# seq(10,150,10),
    limits = c(0, 300)) +
  scale_y_continuous(breaks = seq(0,0.01,0.002),
    minor_breaks = NULL,
    limits = c(0, 0.01)) +
  # theme_bw() +
  theme(axis.text = element_text(size = axis_size),
    axis.title = element_text(size = axis_title_size),
    # legend.key.width = unit(1,"cm"),
    # legend.position = c(.9, .7),
    # legend.justification = 'right',
    # legend.background = element_rect(fill="gray95"),
    # legend.text = element_text(size = 12),
    # legend.title = element_text(size = 14),
    legend.position = "none",
    axis.ticks.y = element_blank(),
    axis.text.y = element_blank(),
    panel.grid.major.y = element_blank()) +
    xlab("Duration in ms") +
# ylab("Density")
  ylab("")
kdeB
```



shift function 1: nonOri vs. expOri

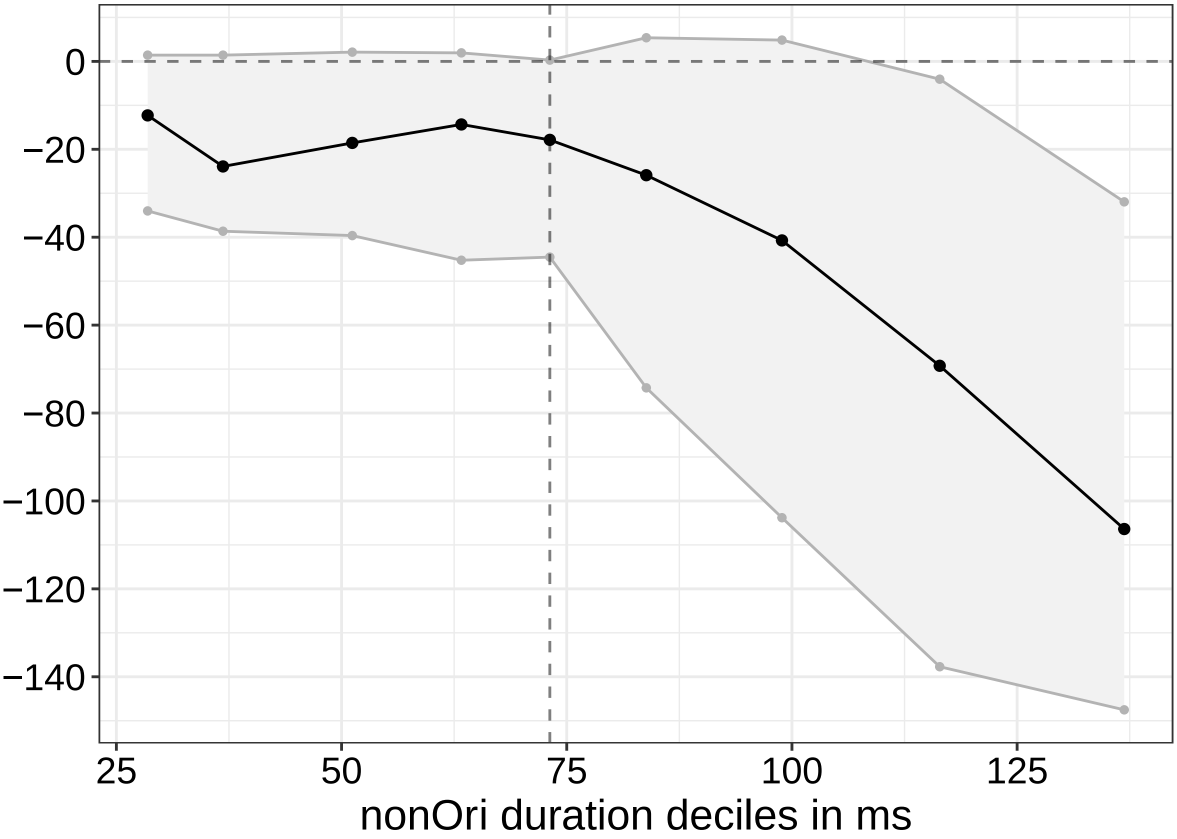
Compute and base R plot

```
sf1 <- qcomhd(nonOri,expOri,q=c(1:9)/10,nboot=nboot,plotit=TRUE)
```



ggplot2 version

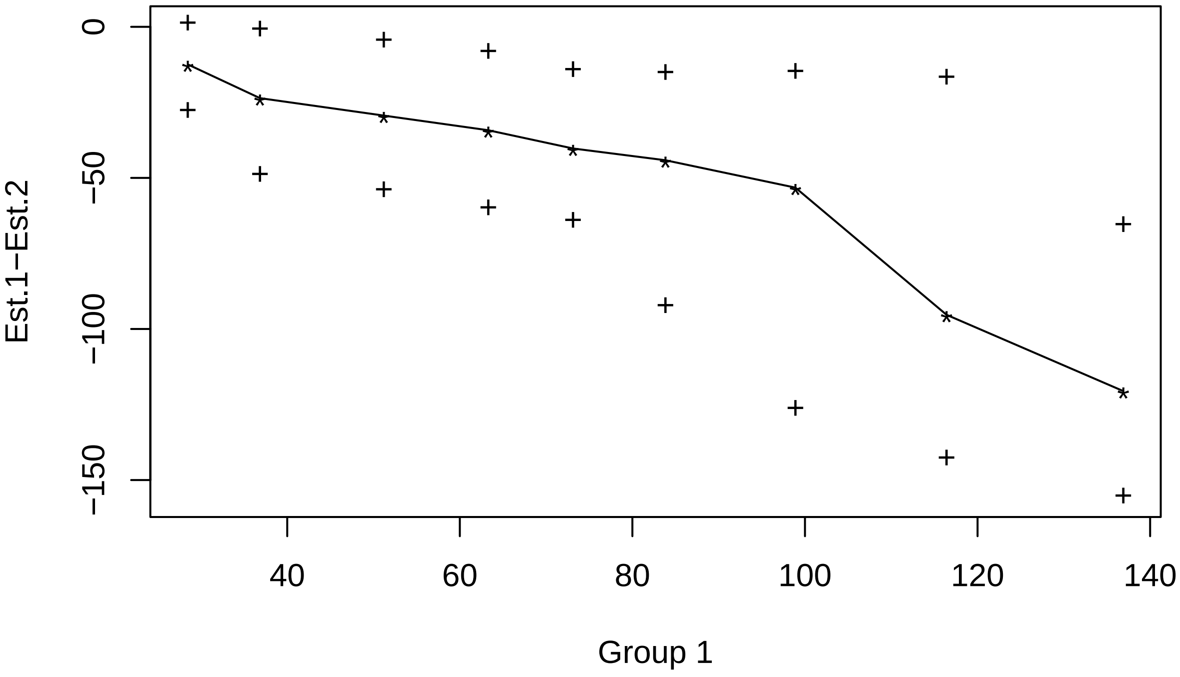
```
# make data frame
dsf1 <- tibble(`Group1` = sf1[,"est.1"],
  `Difference` = sf1[,"est.1_minus_est.2"],
  `lower` = sf1[,"ci.low"],
  `upper` = sf1[,"ci.up"])

# make plot
xintercept <- dsf1[[5,1]]
sf1B <- ggplot(dsf1, aes(x = Group1, y = Difference)) + theme_gar +
  geom_ribbon(aes(ymin = lower, ymax = upper), fill = "grey95") +
  geom_line(aes(y = lower), colour = "grey70") +
  geom_point(aes(y = lower), colour = "grey70", size = 1) +
  geom_line(aes(y = upper), colour = "grey70") +
  geom_point(aes(y = upper), colour = "grey70", size = 1) +
  geom_line(aes(y = Difference)) +
  geom_point(aes(y = Difference), size = 1.5) +
  geom_hline(yintercept = 0, linetype = 2, alpha = 0.5) +
  geom_vline(xintercept = xintercept, linetype = 2, alpha = 0.5) +
  theme(axis.text = element_text(size = 14),
    axis.title = element_text(size = 16)) +
  scale_y_continuous(breaks = seq(-160,0,20)) +
  xlab("nonOri duration deciles in ms") +
  # ylab("Difference: \nnonOri - expOri")
  ylab("")
sf1B
```



shift function 2: nonOri vs. compOri

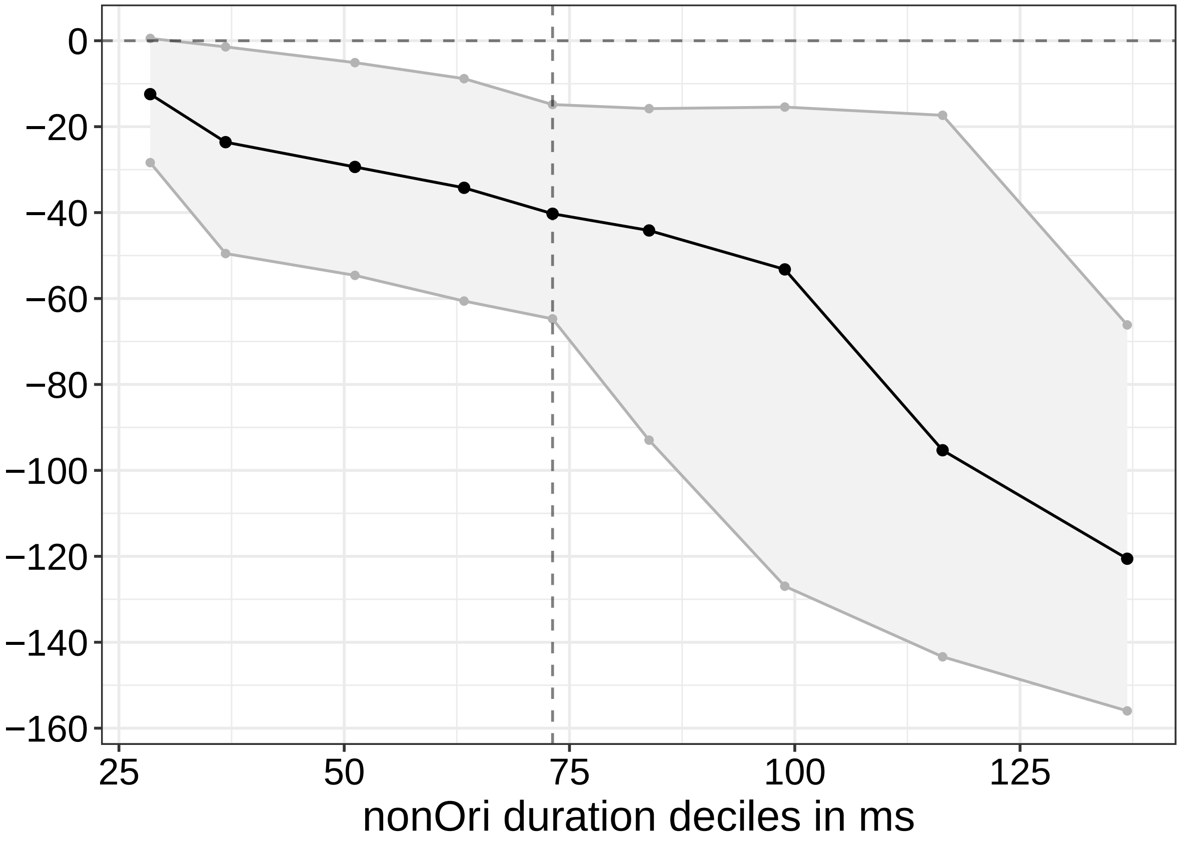
Compute and base R plot

```
sf2 <- qcomhd(nonOri,compOri,q=c(1:9)/10,nboot=nboot,plotit=TRUE)
```



ggplot2 version

```
# make data frame
dsf2 <- tibble(`Group1` = sf2[,"est.1"],
  `Difference` = sf2[,"est.1_minus_est.2"],
  `lower` = sf2[,"ci.low"],
  `upper` = sf2[,"ci.up"])

# make plot
xintercept <- dsf2[[5,1]]
sf2B <- ggplot(dsf2, aes(x = Group1, y = Difference)) + theme_gar +
  geom_ribbon(aes(ymin = lower, ymax = upper), fill = "grey95") +
  geom_line(aes(y = lower), colour = "grey70") +
  geom_point(aes(y = lower), colour = "grey70", size = 1) +
  geom_line(aes(y = upper), colour = "grey70") +
  geom_point(aes(y = upper), colour = "grey70", size = 1) +
  geom_line(aes(y = Difference)) +
  geom_point(aes(y = Difference), size = 1.5) +
  geom_hline(yintercept = 0, linetype = 2, alpha = 0.5) +
  geom_vline(xintercept = xintercept, linetype = 2, alpha = 0.5) +
  theme(axis.text = element_text(size = axis_size),
    axis.title = element_text(size = axis_title_size)) +
  scale_y_continuous(breaks = seq(-160,0,20)) +
  xlab("nonOri duration deciles in ms") +
  # ylab("Difference: \nnonOri - compOri")
  ylab("")
sf2B
```



combine panels into one figure

```
panelA <- cowplot::plot_grid(kdeA, sf1A, sf2A,
                   labels=c("A", "", ""),
                   ncol = 1,
                   nrow = 3,
                   rel_heights = c(1, 1, 1),
                   label_size = 20,
                   hjust = -0.75,
                   scale=.95,
                   align = "v")

panelB <- cowplot::plot_grid(kdeB, sf1B, sf2B,
                    labels=c("B", "", ""),
                    ncol = 1,
                    nrow = 3,
                    rel_heights = c(1, 1, 1),
                    label_size = 20,
                    hjust = -2,
                    scale=.95,
                    align = "v")

cowplot::plot_grid(panelA, panelB,
                    labels=c("", ""),
                    ncol = 2,
                    nrow = 1,
                    rel_widths = c(1, 1),
                    align = "h")

# save figure
ggsave(filename='./figures/figure7.pdf',width=10,height=9)
```

Results reported in the text

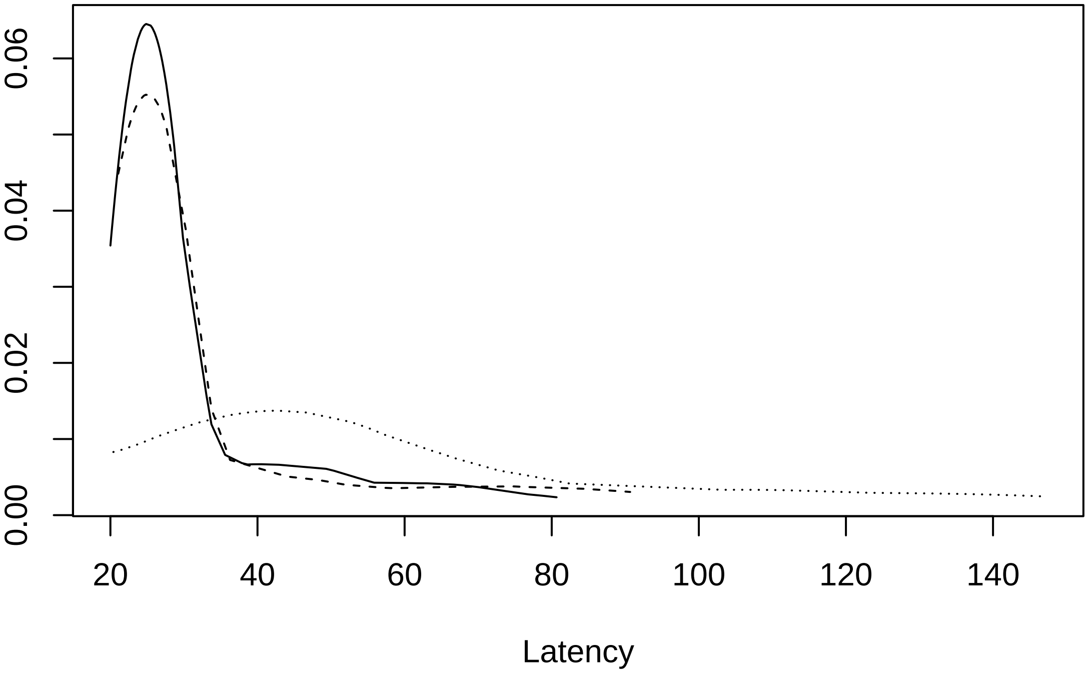
Plot estimates of the latency distributions (based on the R function g5plot). The solid line is non-oriented (nonOri) RFs, the dashed line is expansive oriented (expOri) and the dotted line is compressive oriented (compOri).

The first three columns are the response latency data, the next three are duration, and the final three are direction selectivity index (dsi).

Latencies

Here are the distributions of the latency category:

```
g5plot(talebi[,1:3],xlab='Latency')
```



The latency nonOri and expOir distributions are very similar suggesting that all methods will fail to reject, and this was found to be the case. That is, there is more compelling evidence of no significant differences beyond the results based on conventional methods.

Next, compare nonOri to expOri based on means, 20% trimmed means and medians:

means

```r
trimci(talebi[,1]-talebi[,2],tr=0) # means
```

```
## [1] "The p-value returned by this function is based on the"
## [1] "null value specified by the argument null.value, which defaults to 0"
## [1] "To get a measure of effect size using a Winsorized measure of scale,  use trimciv2"
```

```
## $estimate
## [1] -5.471529
##
## $ci
## [1] -13.339321   2.396263
##
## $test.stat
## [1] -1.399032
##
## $se
## [1] 3.910938
##
## $p.value
## [1] 0.1683683
##
## $n
## [1] 48
```

20% trimmed means

```r
trimci(talebi[,1]-talebi[,2],tr=0.2)  # 20% trim
```

```
## [1] "The p-value returned by this function is based on the"
## [1] "null value specified by the argument null.value, which defaults to 0"
## [1] "To get a measure of effect size using a Winsorized measure of scale,  use trimciv2"
```

```
## $estimate
## [1] -3.410001
##
## $ci
## [1] -8.588068  1.768067
##
## $test.stat
## [1] -1.34688
##
## $se
## [1] 2.531778
##
## $p.value
## [1] 0.1884508
##
## $n
## [1] 48
```

medians

```
sintv2(talebi[,1]-talebi[,2])  # medians
```

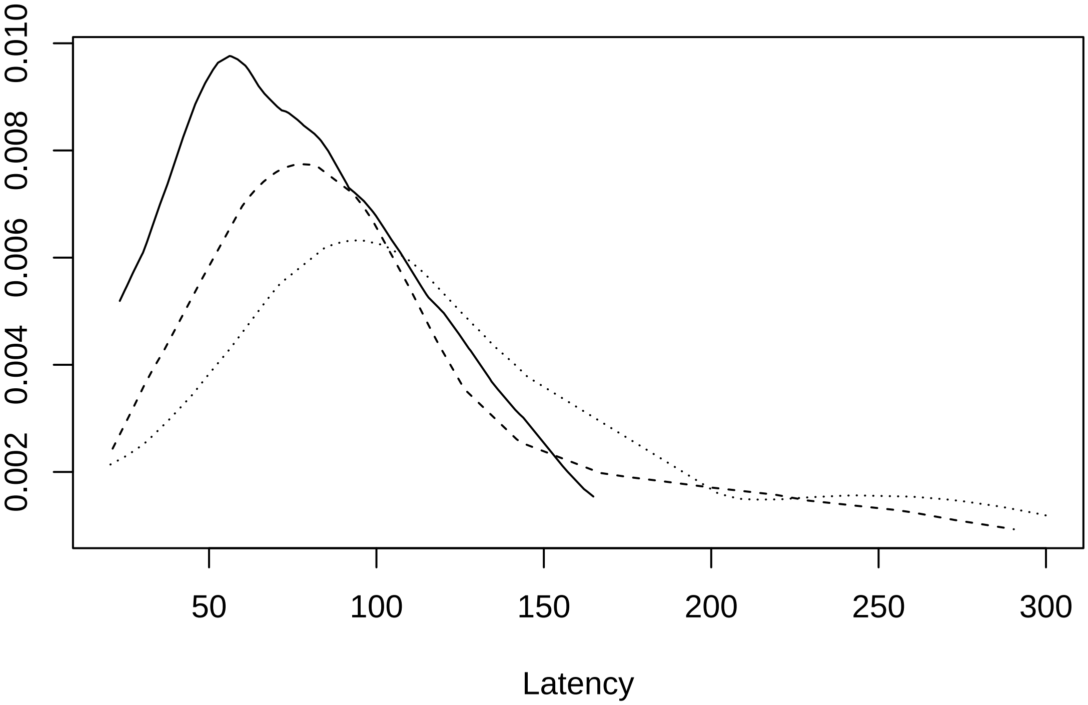
```
## $median
## [1] -1.366665
##
## $n
## [1] 48
##
## $ci.low
## [1] -6.506927
##
## $ci.up
## [1] 0.5600692
##
## $p.value
## [1] 0.39
```

So there is considerable evidence that no significant difference was missed due to the method used to compare the groups.

Durations

Now consider duration. Here is a plot of the distributions:

```
g5plot(talebi[,4:6],xlab='Latency')
```



The results when comparing means, 20% trimmed means and medians for nonOri versus expOri:

means

```
trimci(talebi[,4]-talebi[,5],tr=0) # means
```

```
## [1] "The p-value returned by this function is based on the"
## [1] "null value specified by the argument null.value, which defaults to 0"
## [1] "To get a measure of effect size using a Winsorized measure of scale,  use trimciv2"

## $estimate
## [1] -41.23194
##
## $ci
## [1] -65.33878 -17.12510
##
## $test.stat
## [1] -3.440848
##
## $se
## [1] 11.98308
##
## $p.value
## [1] 0.001226751
##
## $n
## [1] 48
```

20% trimmed means

```
trimci(talebi[,4]-talebi[,5],tr=0.2)  # 20% trim
```

```
## [1] "The p-value returned by this function is based on the"
## [1] "null value specified by the argument null.value, which defaults to 0"
## [1] "To get a measure of effect size using a Winsorized measure of scale,  use trimciv2"

## $estimate
## [1] -33.42667
##
## $ci
## [1] -55.69741 -11.15592
##
## $test.stat
## [1] -3.069732
##
## $se
## [1] 10.88912
##
## $p.value
## [1] 0.004617461
##
## $n
## [1] 48
```

medians

```
sintv2(talebi[,4]-talebi[,5])  # medians
```

```
## $median
## [1] -36.53334
##
## $n
## [1] 48
##
## $ci.low
## [1] -54.85607
##
## $ci.up
## [1] -17.16773
##
## $p.value
## [1] 0.009
```

Repeat the above for nonOri versus compOri:

means

```
trimci(talebi[,4]-talebi[,6],tr=0) # means
```

```
## [1] "The p-value returned by this function is based on the"
## [1] "null value specified by the argument null.value, which defaults to 0"
## [1] "To get a measure of effect size using a Winsorized measure of scale,  use trimciv2"

## $estimate
## [1] -51.48254
##
## $ci
## [1] -73.40913 -29.55595
##
## $test.stat
## [1] -4.693485
##
## $se
## [1] 10.96894
##
## $p.value
## [1] 1.52821e-05
##
## $n
## [1] 63
```

20% trimmed means

```
trimci(talebi[,4]-talebi[,6],tr=0.2)  # 20% trim
```

```
## [1] "The p-value returned by this function is based on the"
## [1] "null value specified by the argument null.value, which defaults to 0"
## [1] "To get a measure of effect size using a Winsorized measure of scale,  use trimciv2"
```

```
## $estimate
## [1] -34.8735
##
## $ci
## [1] -58.49752 -11.24948
##
## $test.stat
## [1] -2.988387
##
## $se
## [1] 11.66967
##
## $p.value
## [1] 0.004894304
##
## $n
## [1] 63
```

medians

```
sintv2(talebi[,4]-talebi[,6])  # medians
```

```
## $median
## [1] -20.8
##
## $n
## [1] 63
##
## $ci.low
## [1] -57.25379
##
## $ci.up
## [1] -7.115883
##
## $p.value
## [1] 0.002
```