dm_bcfs-2_subject

Egor Ananyev

2016-03-17

```
# The list of subjects, the order of conditions, and the thresholds are derived from Subjects.xlsx
# Read in the Excel worksheet:
library(xlsx)
```

```
## Loading required package: rJava
```

```
## Warning: package 'rJava' was built under R version 3.2.3
```

```
## Loading required package: xlsxjars
```

```
library(ggplot2)
library(plyr)
source('/Users/Egor/Dropbox/Prog/R/myFunctions/blackTheme.R')
```

```
## Warning: package 'gridExtra' was built under R version 3.2.4
```

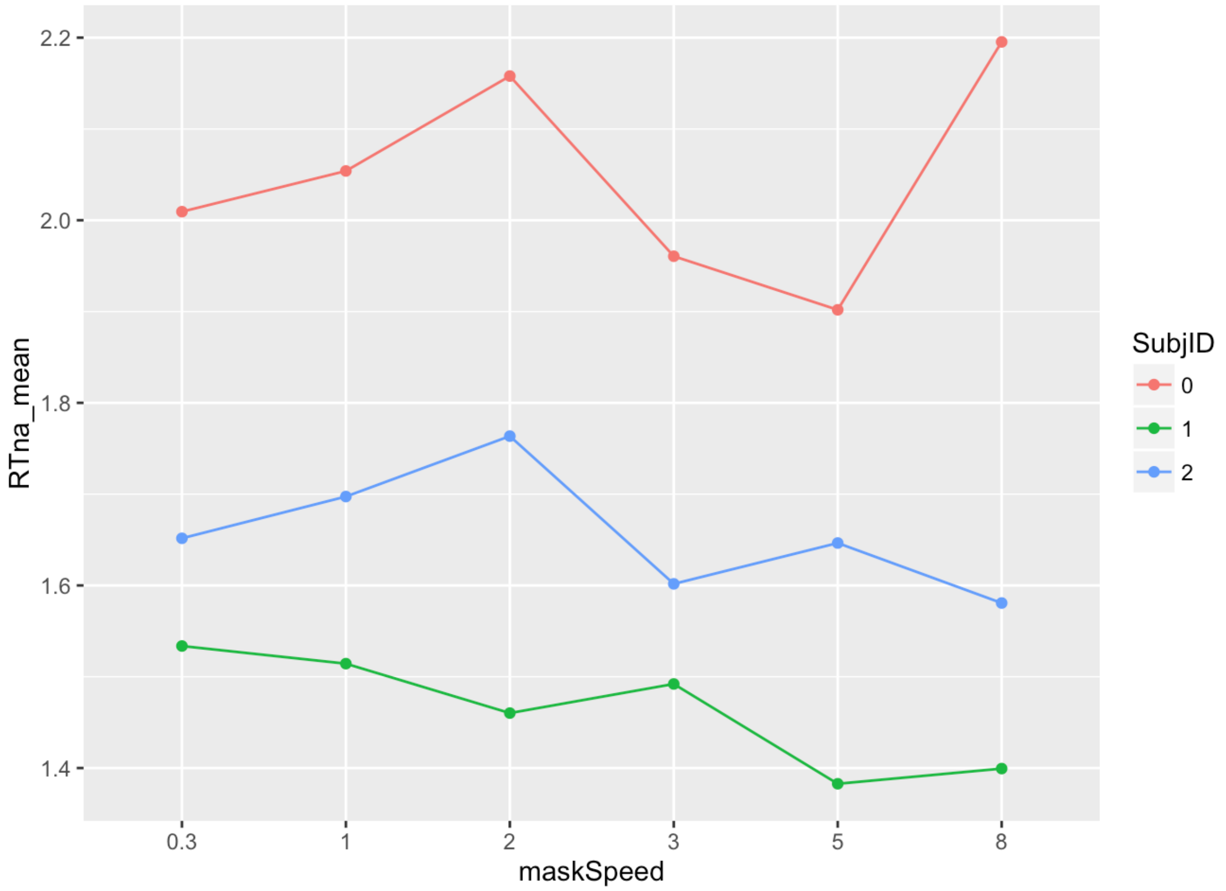
```
dataDir <- '/Users/Egor/Dropbox/Projects/supr-eff/data/'
subjList <- read.xlsx(file=paste(dataDir,'Subjects.xlsx',sep=''), 1, colIndex=
c(1:8), header=T)
subjList$Subj<- as.factor(subjList$Subj)
subjList$Sequence <- as.factor(subjList$Sequence)
condList <- c('ts1','ts8')
outlFac <- 3 # outlier factor
# Gathering the directories from the data directory:
allDirs <- dir(dataDir)
# Setting variables for the loop:
ds <- data.frame()
curSubj <- 1 #temp
curCond <- 1 #temp
for(curSubj in 1:length(subjList$Subj)){ #going through all dirs
    for(curCond in 1:length(condList)){
        grepPattern <- paste('dm_', condList[curCond], '_t0_', subjList$Subj[c
urSubj], sep='')
        subjDir <- allDirs[grep(grepPattern, allDirs)]
        # With this directory, extracting data:
        subjDataFN <- paste(dataDir, subjDir, '/', subjDir, '_trials.tsv', sep
='')
        if(file.exists(subjDataFN)){
            nConds <- 7
            nc <- nConds-1 # the number of conditions without the blank
            ss <- read.table(subjDataFN, sep='\t', nrows=nConds, header=T)
            sumss <- ss[,1:29]
            # Inserting subject ID
            sumss$SubjID <- subjList$Subj[curSubj]
            # I can't take RT_mean, because it doesn't exclude zeros
            # Need to get the RTs on my own, and since the nTrials might
            # differ...
            nTrials <- ss$n[1]
            rts <- ss[,30:(29+nTrials)]
            # Counting and removing RTs<.6 and non-responses
            shortRT <- .6
            sumss$cnt_RTshort <- c(rowSums(rts[1:nc,]<shortRT),0)
            sumss$cnt_RTna <- rowSums(rts==0)
            # Handling non-responses in two different ways: either equating th
em to 3s (max RT) or to NA
            rt3 <- rts
            rt3[rt3==0] <- 3 #maximum rt
            sumss$RT3_mean <- c(rowMeans(rt3[1:nc,],na.rm=T),0)
            sumss$RT3_norm <- c(sumss$RT3_mean[1:nc]/mean(sumss$RT3_mean[1:nc]
,na.rm=T),0)
            rtna <- rts
            rtna[rtna==0] <- NA
            rtna[rtna<shortRT] <- NA
            # Dealing with the outliers:
            sumss$RTna_mean <- c(rowMeans(rtna[1:nc,],na.rm=T),0)
            sumss$RTna_sd <- c(apply(rtna[1:nc,],1,sd,na.rm=T),0)
            sumss$RTna_outlLow <- rowSums(rtna<(sumss$RTna_mean-outlFac*sumss$
RTna_sd),na.rm=T)
```

```
            sumss$RTna_outlHigh <- rowSums(rtna>(sumss$RTna_mean+outlFac*sumss
$RTna_sd),na.rm=T)
            rtna[rtna<(sumss$RTna_mean-outlFac*sumss$RTna_sd)] <- NA
            rtna[rtna>(sumss$RTna_mean+outlFac*sumss$RTna_sd)] <- NA
            sumss$RTna_mean <- c(rowMeans(rtna[1:nc,],na.rm=T),0)
            sumss$RTna_sd <- c(apply(rtna[1:nc,],1,sd,na.rm=T),0)
            sumss$RTna_norm <- c(sumss$RTna_mean[1:nc]/mean(sumss$RTna_mean[1:
nc],na.rm=T),0)
            # Binding the summary statistics with the empty data frame
            ds <- rbind(ds, sumss)
            print(paste('Processed file',subjDataFN))
        }else{
            print(paste('File does not exist:',subjDataFN))
        }
    }
}
```

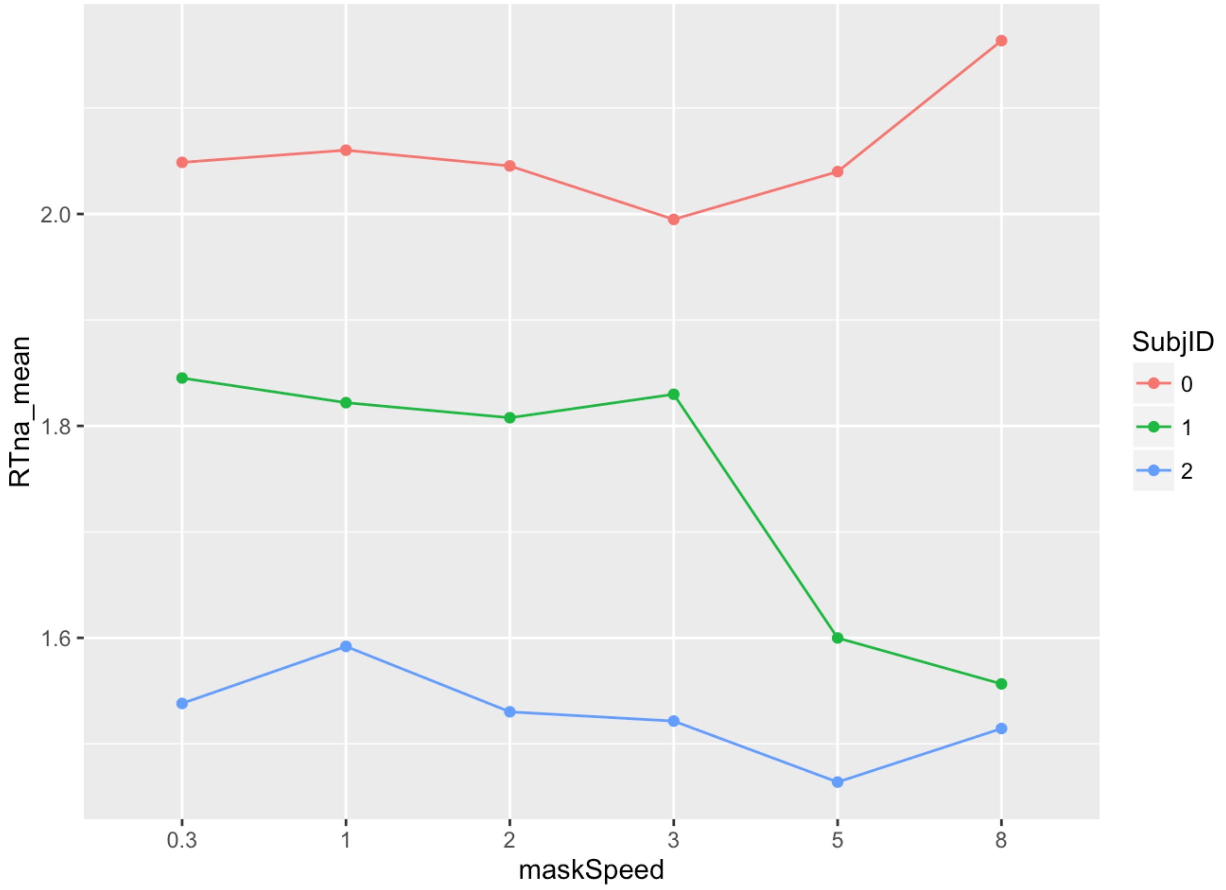
```
## [1] "Processed file /Users/Egor/Dropbox/Projects/supr-eff/data/dm_ts1_t0_1_
dom-r_2016-04-14_1412/dm_ts1_t0_1_dom-r_2016-04-14_1412_trials.tsv"
## [1] "Processed file /Users/Egor/Dropbox/Projects/supr-eff/data/dm_ts8_t0_1_
dom-r_2016-04-14_1435/dm_ts8_t0_1_dom-r_2016-04-14_1435_trials.tsv"
## [1] "Processed file /Users/Egor/Dropbox/Projects/supr-eff/data/dm_ts1_t0_2_
dom-r_2016-04-14_1655/dm_ts1_t0_2_dom-r_2016-04-14_1655_trials.tsv"
## [1] "Processed file /Users/Egor/Dropbox/Projects/supr-eff/data/dm_ts8_t0_2_
dom-r_2016-04-14_1616/dm_ts8_t0_2_dom-r_2016-04-14_1616_trials.tsv"
## [1] "Processed file /Users/Egor/Dropbox/Projects/supr-eff/data/dm_ts1_t0_0_
dom-r_2016-04-15_1555/dm_ts1_t0_0_dom-r_2016-04-15_1555_trials.tsv"
## [1] "Processed file /Users/Egor/Dropbox/Projects/supr-eff/data/dm_ts8_t0_0_
dom-r_2016-04-15_1730/dm_ts8_t0_0_dom-r_2016-04-15_1730_trials.tsv"
## [1] "File does not exist: /Users/Egor/Dropbox/Projects/supr-eff/data//_tria
ls.tsv"
## [1] "File does not exist: /Users/Egor/Dropbox/Projects/supr-eff/data//_tria
ls.tsv"
## [1] "File does not exist: /Users/Egor/Dropbox/Projects/supr-eff/data//_tria
ls.tsv"
## [1] "File does not exist: /Users/Egor/Dropbox/Projects/supr-eff/data//_tria
ls.tsv"
## [1] "File does not exist: /Users/Egor/Dropbox/Projects/supr-eff/data//_tria
ls.tsv"
## [1] "File does not exist: /Users/Egor/Dropbox/Projects/supr-eff/data//_tria
ls.tsv"
## [1] "File does not exist: /Users/Egor/Dropbox/Projects/supr-eff/data//_tria
ls.tsv"
## [1] "File does not exist: /Users/Egor/Dropbox/Projects/supr-eff/data//_tria
ls.tsv"
## [1] "File does not exist: /Users/Egor/Dropbox/Projects/supr-eff/data//_tria
ls.tsv"
## [1] "File does not exist: /Users/Egor/Dropbox/Projects/supr-eff/data//_tria
ls.tsv"
## [1] "File does not exist: /Users/Egor/Dropbox/Projects/supr-eff/data//_tria
ls.tsv"
## [1] "File does not exist: /Users/Egor/Dropbox/Projects/supr-eff/data//_tria
ls.tsv"
## [1] "File does not exist: /Users/Egor/Dropbox/Projects/supr-eff/data//_tria
ls.tsv"
## [1] "File does not exist: /Users/Egor/Dropbox/Projects/supr-eff/data//_tria
ls.tsv"
```

```
ds$maskSpeed <- as.factor(ds$maskSpeed)
ds <- ds[ds$targSpeed>0,]
```

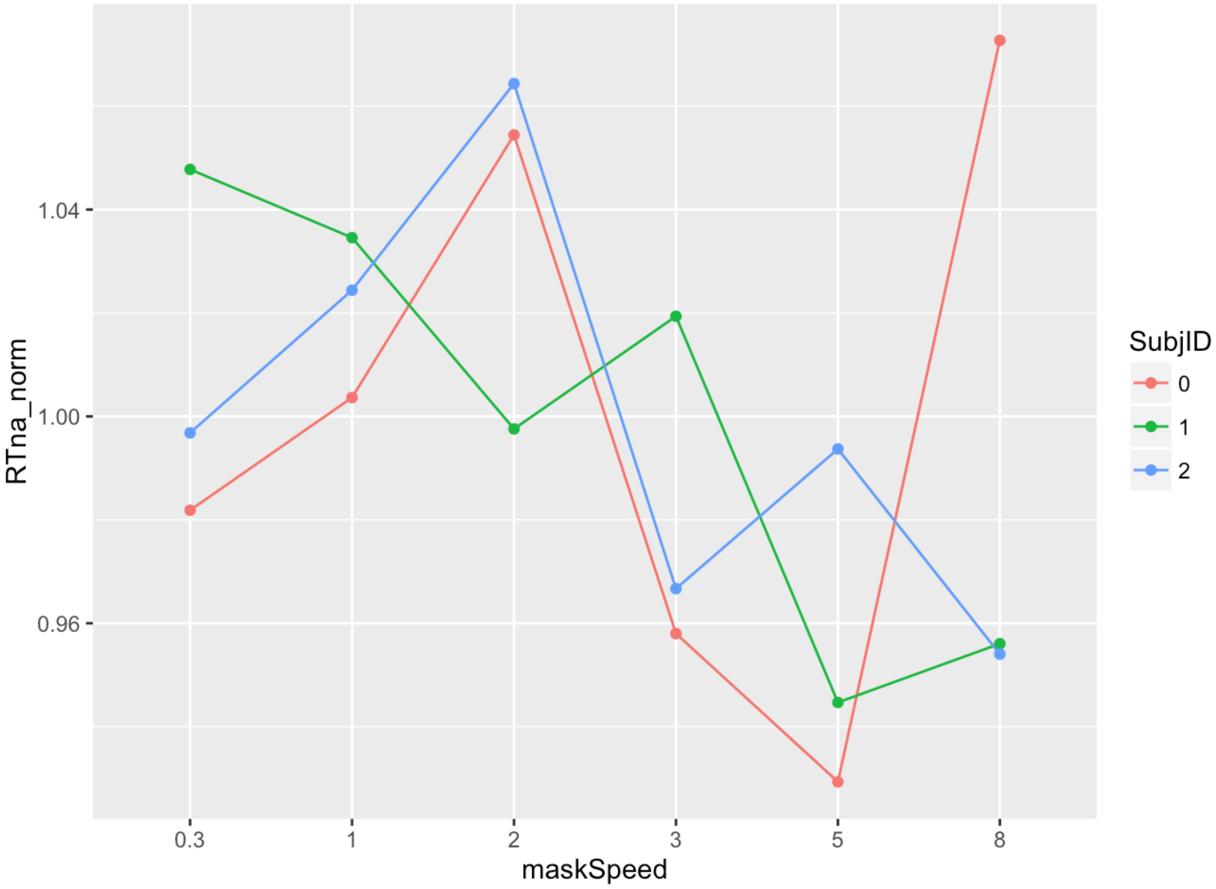
```
# Line plots of raw RTs per subject
ggplot(ds[ds$cond=='ts1',], aes(x=maskSpeed, y=RTna_mean, group=SubjID, colour
=SubjID)) +  geom_line() + geom_point()
```



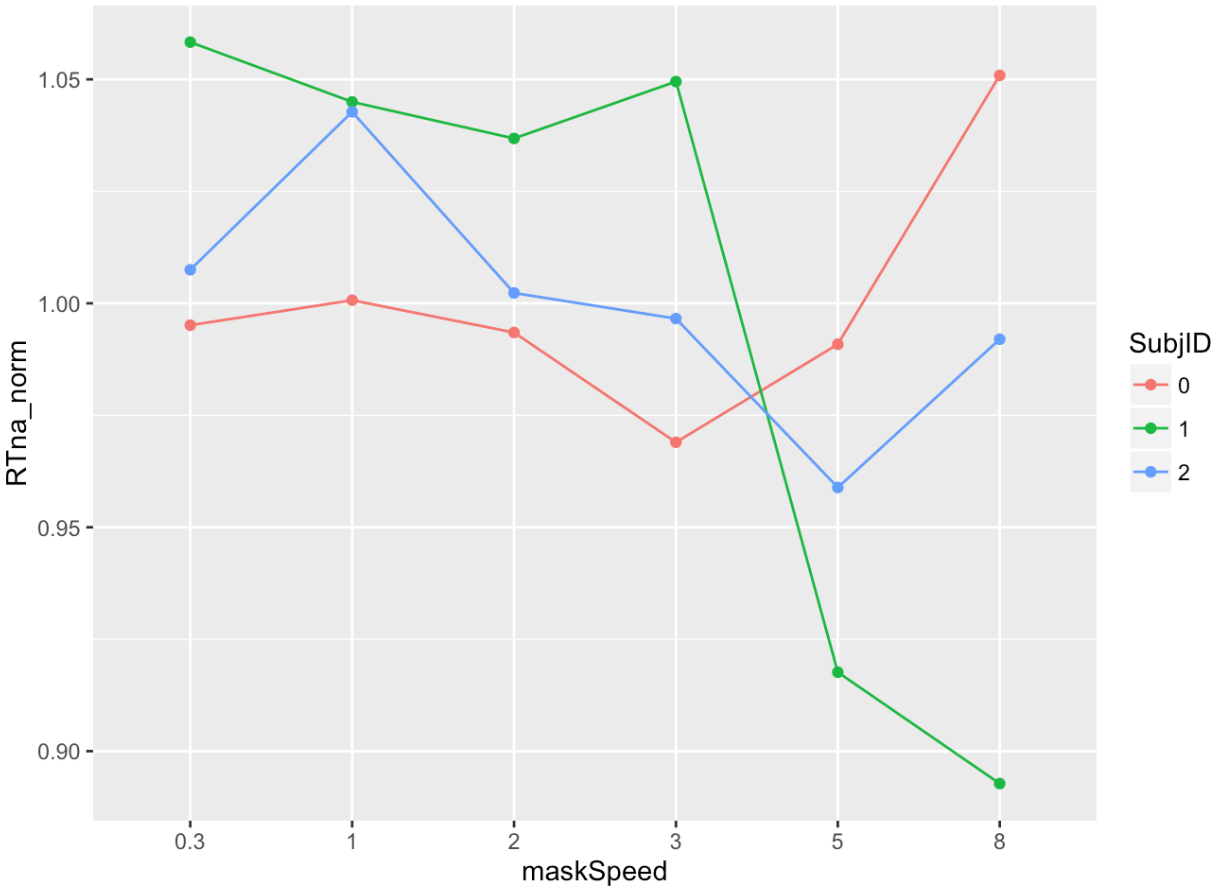
```
ggplot(ds[ds$cond=='ts8',], aes(x=maskSpeed, y=RTna_mean, group=SubjID, colour
=SubjID)) +  geom_line() + geom_point()
```



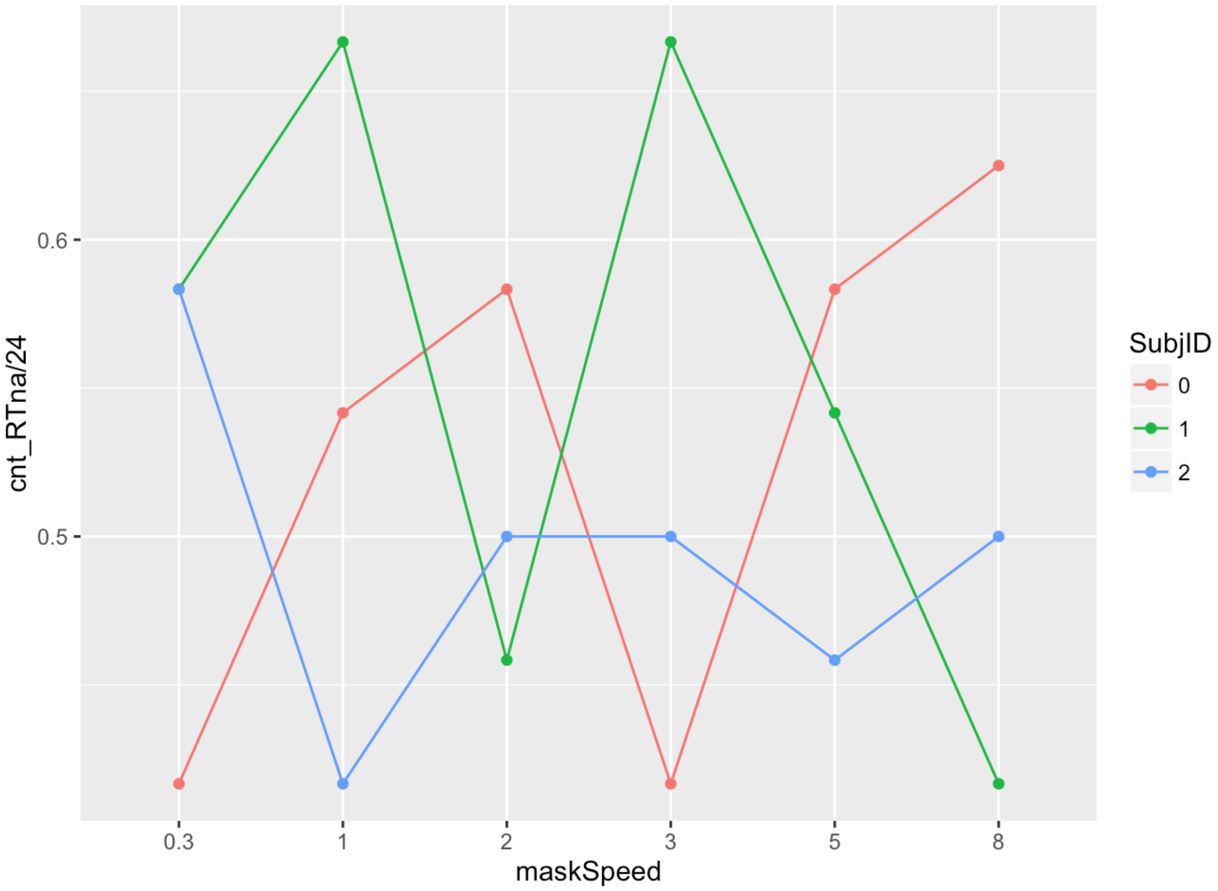
```
# Line plots of normalized RTs
ggplot(ds[ds$cond=='ts1',], aes(x=maskSpeed, y=RTna_norm, group=SubjID, colour
=SubjID)) +  geom_line() + geom_point()
```



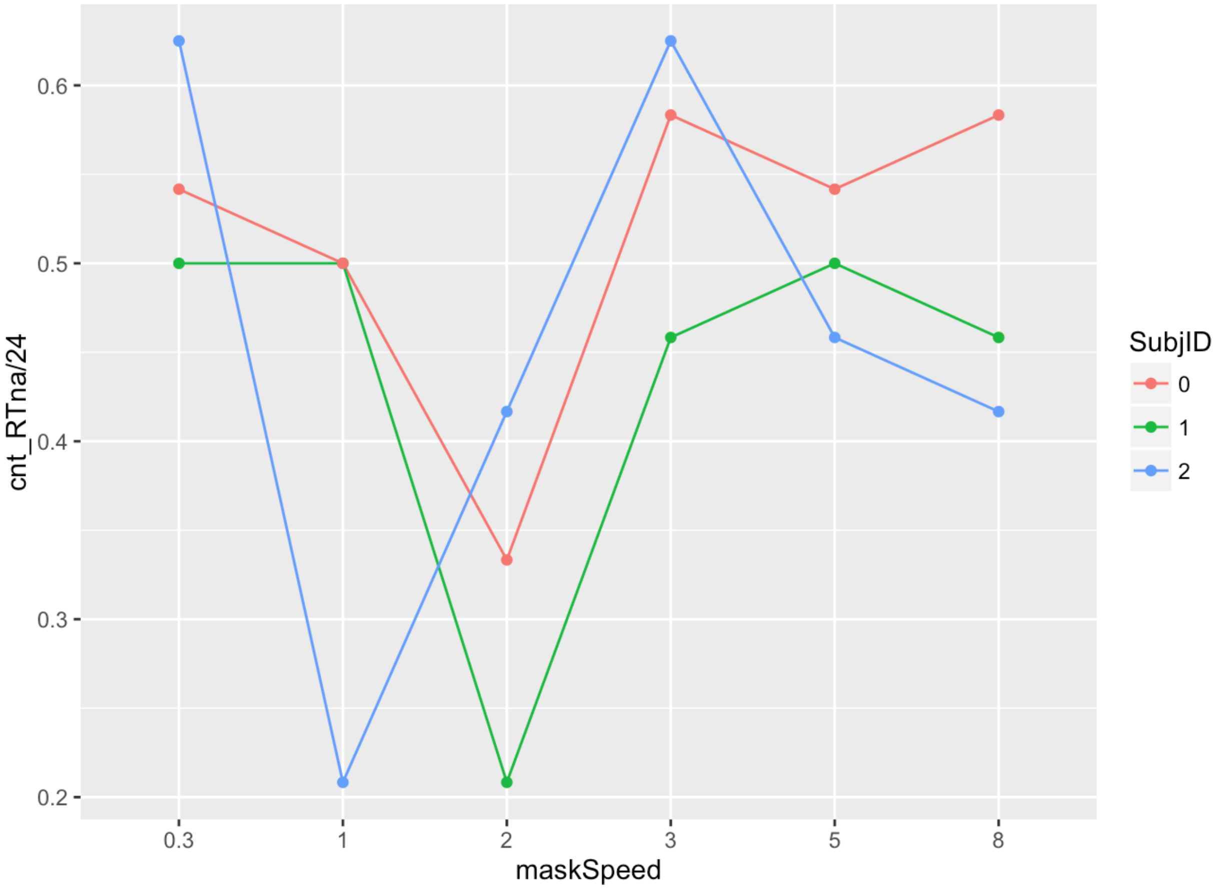
```
ggplot(ds[ds$cond=='ts8',], aes(x=maskSpeed, y=RTna_norm, group=SubjID, colour
=SubjID)) +  geom_line() + geom_point()
```



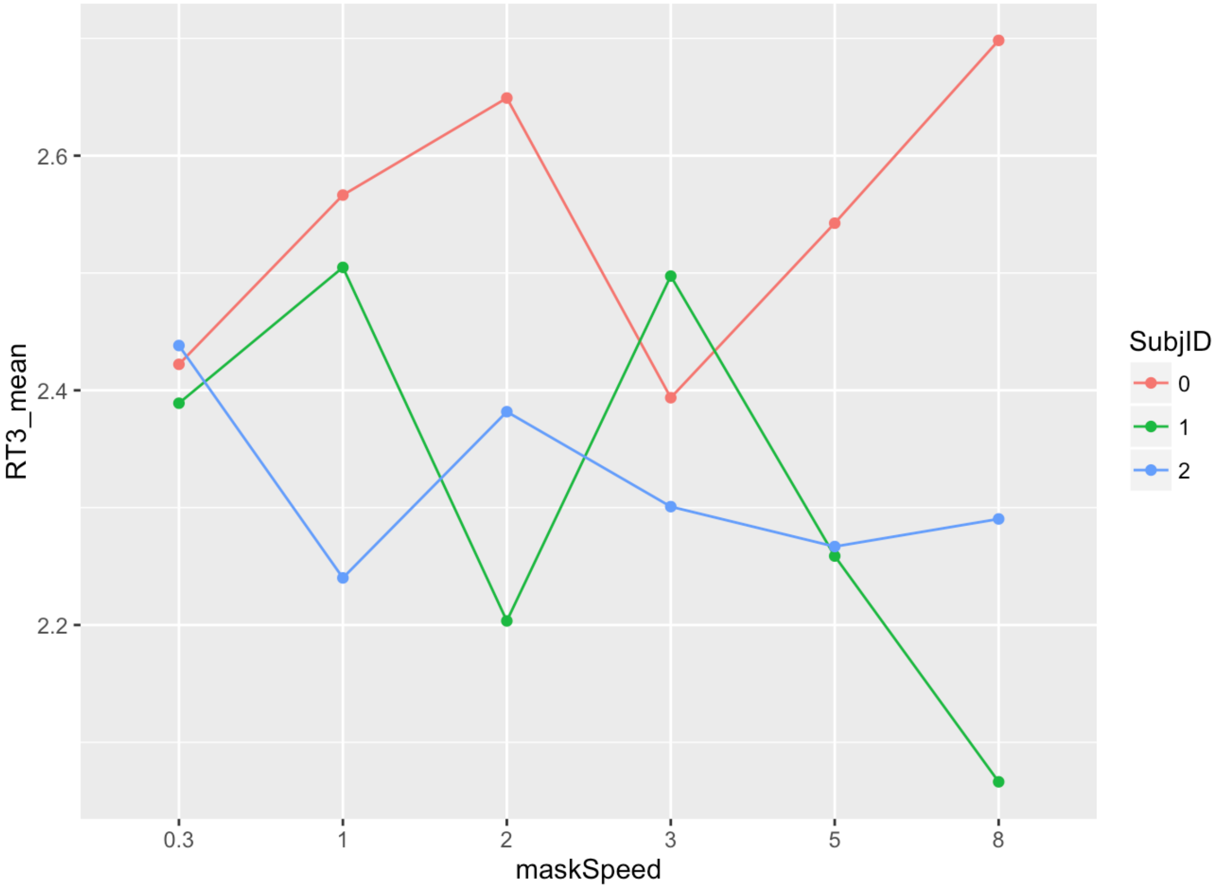
```
# Line plots for non-broken trials:
ggplot(ds[ds$cond=='ts1',], aes(x=maskSpeed, y=cnt_RTna/24, group=SubjID, colour=SubjID)) +  geom_line() + geom_point()
```



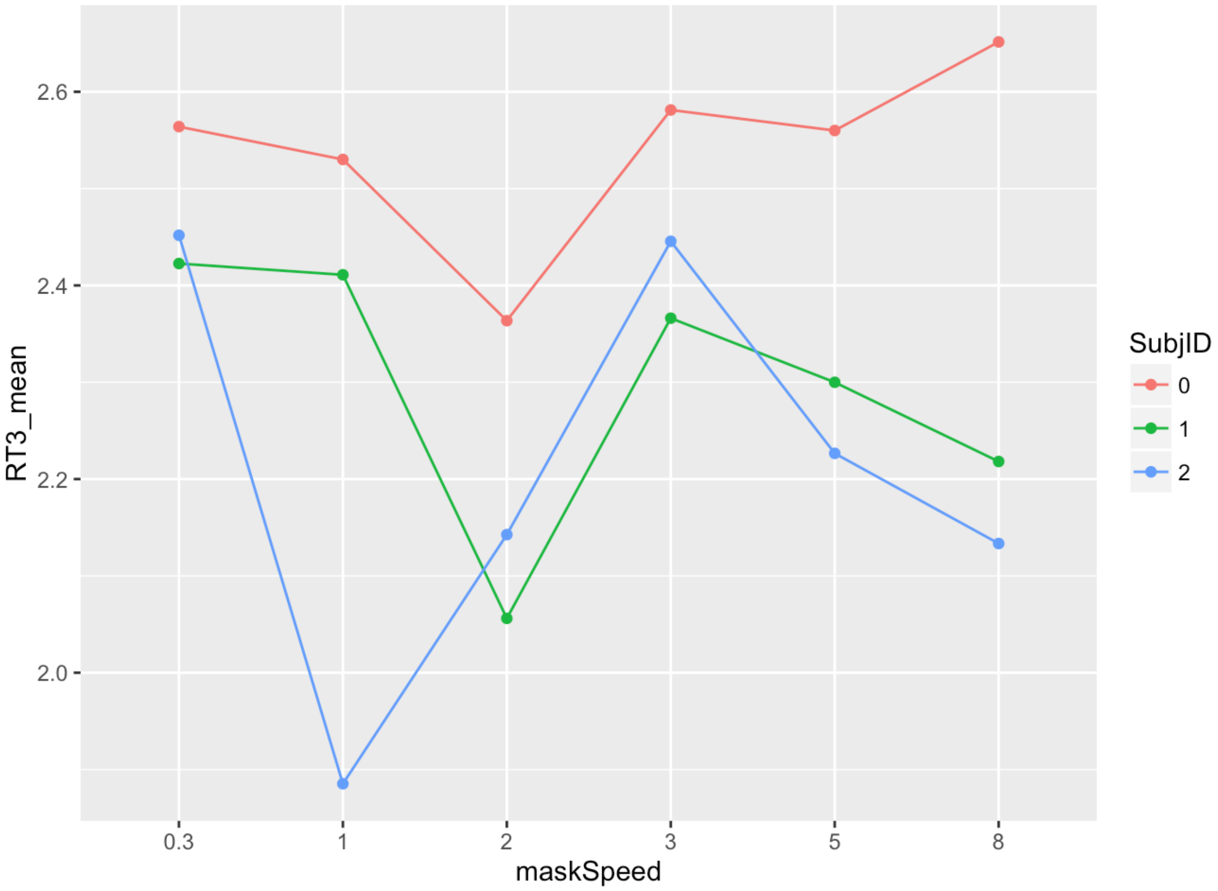
```
ggplot(ds[ds$cond=='ts8',], aes(x=maskSpeed, y=cnt_RTna/24, group=SubjID, colo
ur=SubjID)) +  geom_line() + geom_point()
```



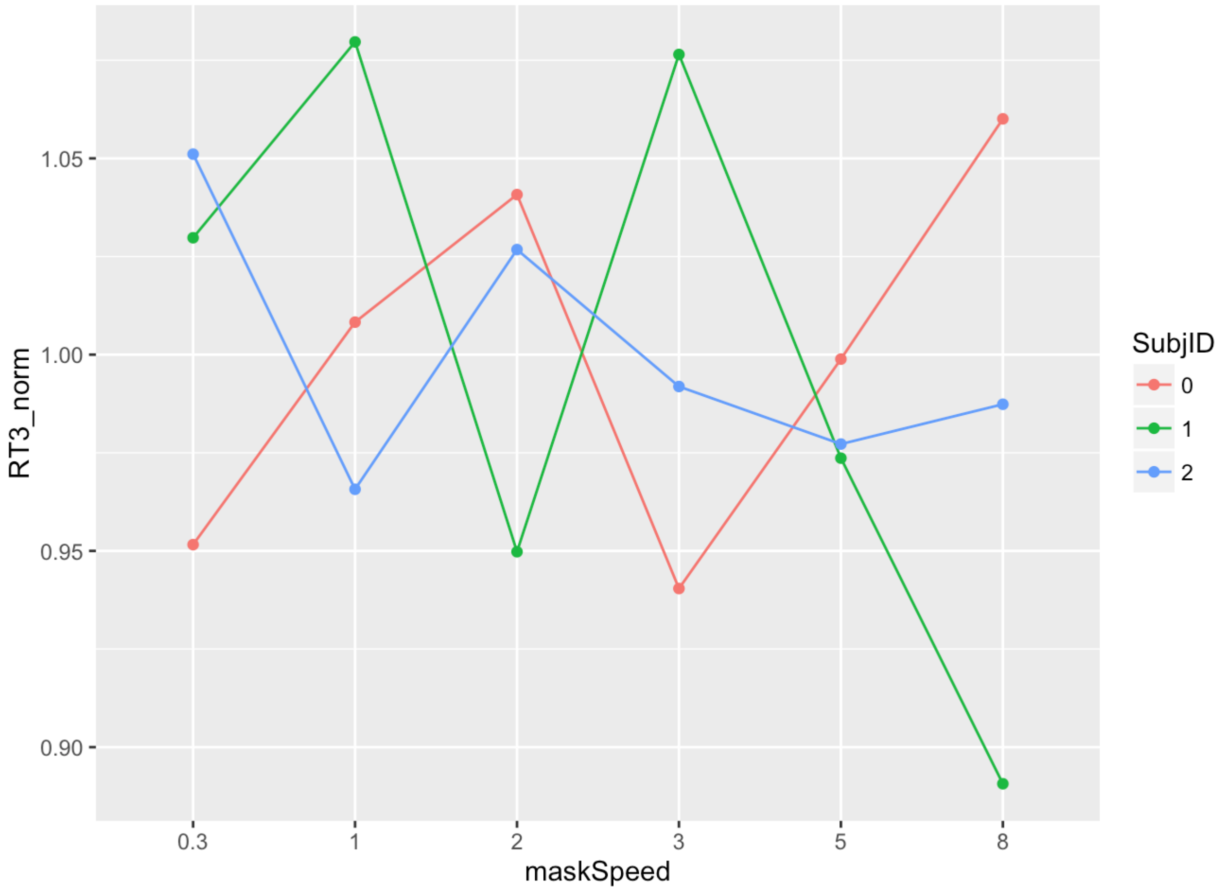
```
# Line plots for RT3 means
ggplot(ds[ds$cond=='ts1',], aes(x=maskSpeed, y=RT3_mean, group=SubjID, colour=
SubjID)) +  geom_line() + geom_point()
```



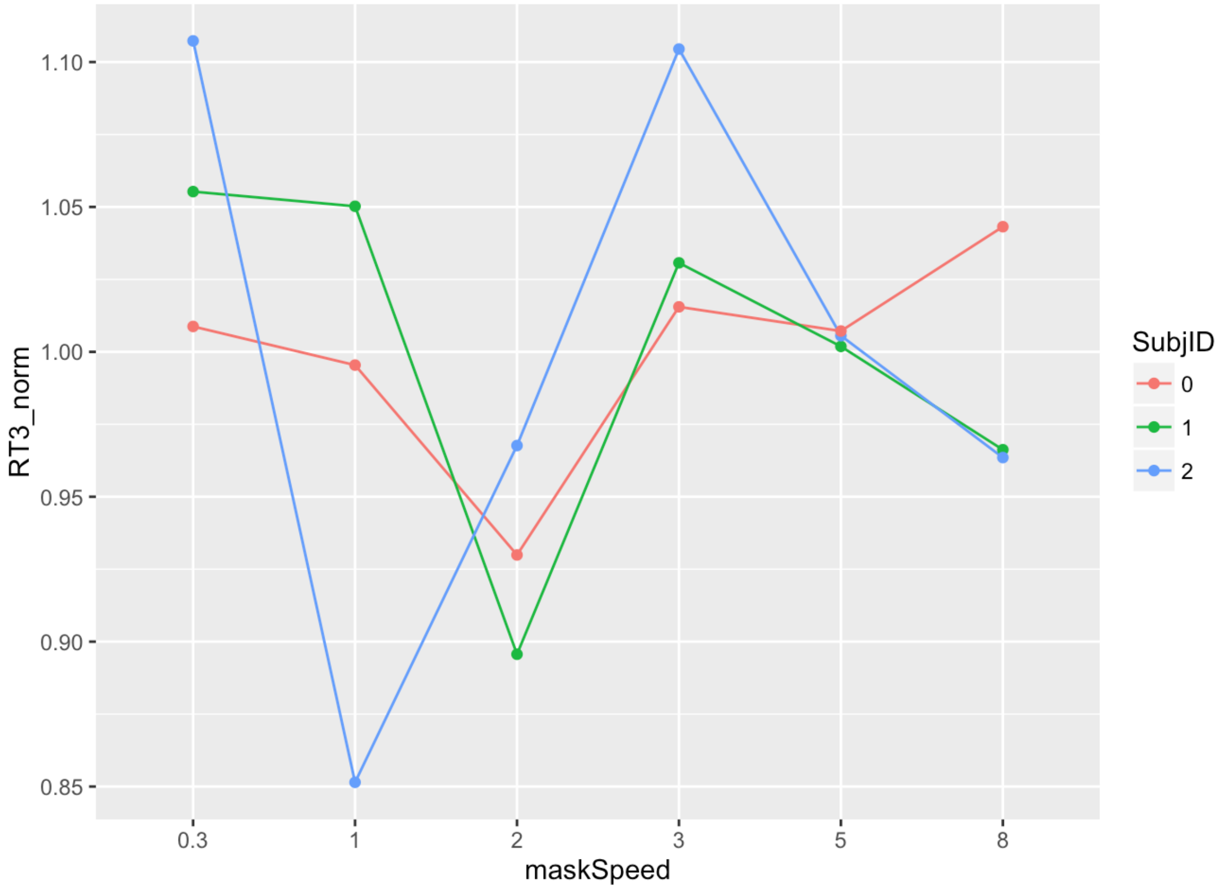
```
ggplot(ds[ds$cond=='ts8',], aes(x=maskSpeed, y=RT3_mean, group=SubjID, colour=
SubjID)) +  geom_line() + geom_point()
```



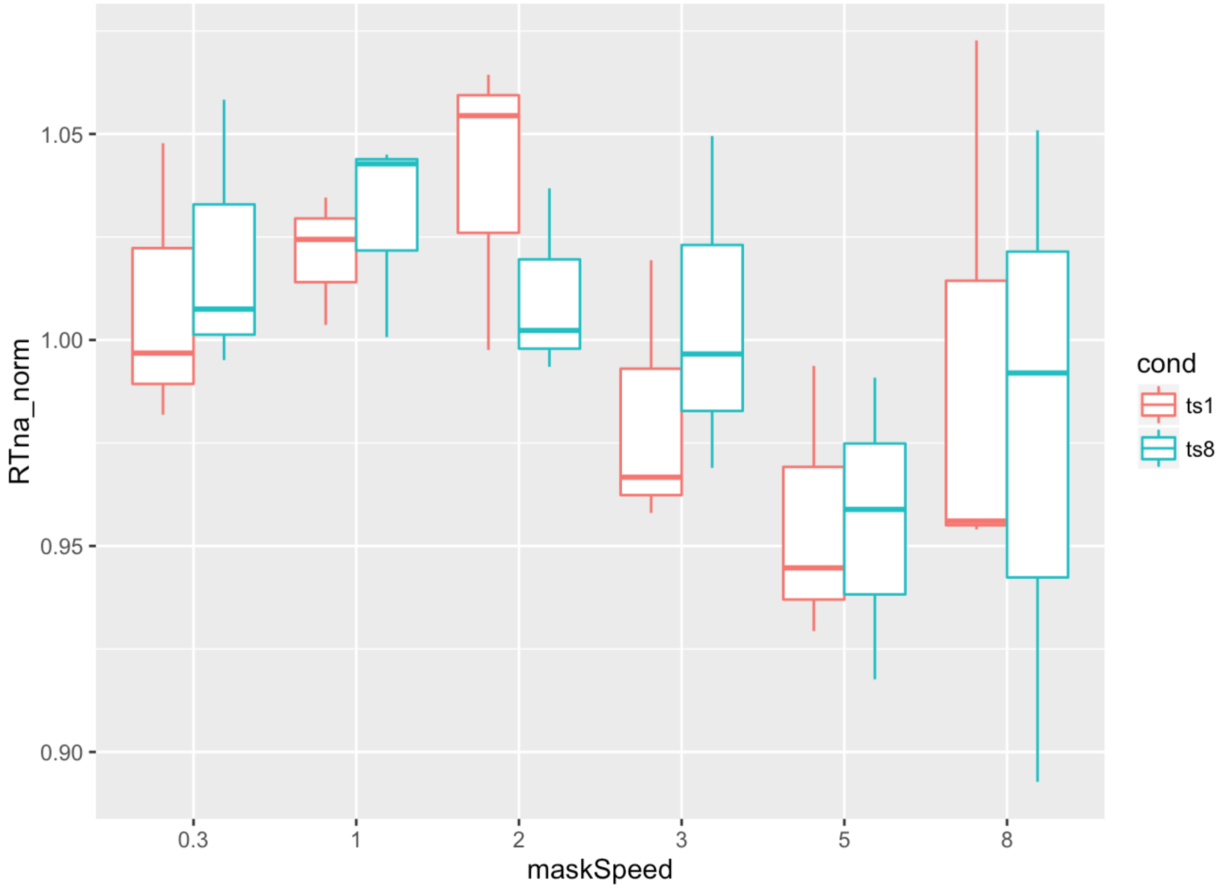
```
# Line plots for RT3 normalized means
ggplot(ds[ds$cond=='ts1',], aes(x=maskSpeed, y=RT3_norm, group=SubjID, colour=
SubjID)) +  geom_line() + geom_point()
```



```
ggplot(ds[ds$cond=='ts8',], aes(x=maskSpeed, y=RT3_norm, group=SubjID, colour=
SubjID)) +  geom_line() + geom_point()
```

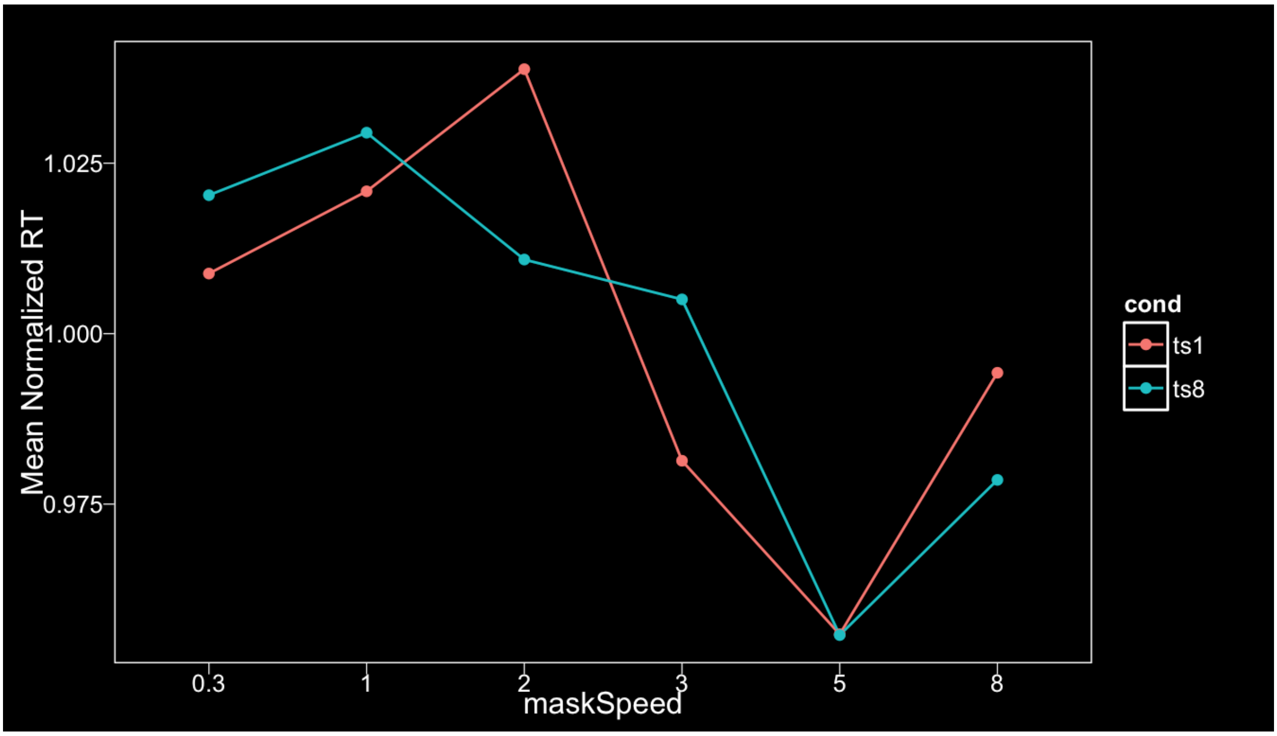


```
# Box plots of normalized RTs
ggplot(ds, aes(x=maskSpeed, y=RTna_norm, colour=cond)) +  geom_boxplot()
```



```
ss = ddply(ds, .(maskSpeed,cond), summarise, `Mean Normalized RT`=mean(RTna_no
rm))
ggplot(ss, aes(x=maskSpeed, y=`Mean Normalized RT`, colour=cond, group=cond))
+  geom_line() + geom_point() +  theme_black()
```

```
## Warning: `axis.ticks.margin` is deprecated. Please set `margin` property of
## `axis.text` instead
```


Mean Normalized RT
1.025
1.000
0.975
0.3
1
2
3
5
8
maskSpeed
cond
ts1
ts8