correlations

Correlations in `bdots`

This vignette is created to illustrate the use of the `bdotsCorr` function, which finds the correlation between a fixed value in our dataset and the collection of fitted curves at each time points for each of the groups fit in `bdotsFit`.

First, let's take an existing dataset and add a fixed value for each of the subjects

```
library(bdots)
library(data.table)

## Let's work with cohort_unrelated dataset, as it has multiple groups
dat <- as.data.table(cohort_unrelated)

## And add a fixed value for which we want to find a correlation
dat[, val := rnorm(1), by = Subject]

head(dat)
```

```
##    Subject Time DB_cond  Fixations LookType Group       val
## 1:       1    0      50 0.01136364   Cohort    50 0.5911445
## 2:       1    4      50 0.01136364   Cohort    50 0.5911445
## 3:       1    8      50 0.01136364   Cohort    50 0.5911445
## 4:       1   12      50 0.01136364   Cohort    50 0.5911445
## 5:       1   16      50 0.02272727   Cohort    50 0.5911445
## 6:       1   20      50 0.02272727   Cohort    50 0.5911445
```

Now, we go about creating our fitted object as usual

```
## Create regular fit in bdots
fit <- bdotsFit(data = dat,
                subject = "Subject",
                time = "Time",
                group = c("LookType", "Group"),
                y = "Fixations", curveType = doubleGauss2(),
                cores = 2)
```

Using this fit object, we now introduce the `bdotsCorr` function, taking four arguments:

1. `bdObj`, any object returned from a `bdotsFit` call
2. `val`, a length one character vector of the value with which we want to correlate. `val` should be a column in our original dataset, and it should be numeric
3. `ciBands`, a boolean indicating whether or not we want to return 95% confidence intervals. Default is `FALSE`
4. `method`, paralleling the `method` argument in `cor` and `cor.test`. The default is `pearson`.

```
## Returns a data.table of class bdotsCorrObj
corr_ci <- bdotsCorr(fit, val = "val", ciBands = TRUE)
head(corr_ci)
```

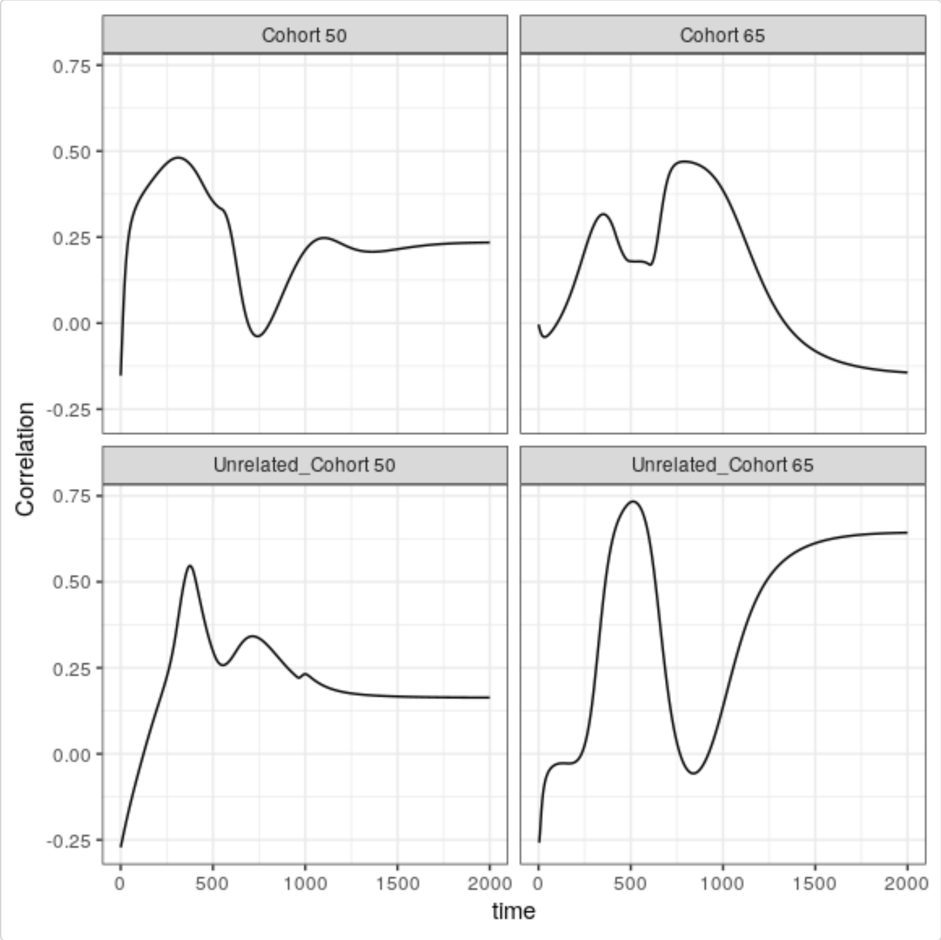
```
##    time Correlation       lower     upper     Group Group1 Group2
## 1:    0 -0.15244734 -0.7414972 0.5693137 Cohort 50 Cohort     50
## 2:    4 -0.09788149 -0.7154925 0.6056079 Cohort 50 Cohort     50
## 3:    8 -0.04195758 -0.6869378 0.6399976 Cohort 50 Cohort     50
## 4:   12  0.01181996 -0.6574628 0.6706769 Cohort 50 Cohort     50
## 5:   16  0.06087569 -0.6286621 0.6968255 Cohort 50 Cohort     50
## 6:   20  0.10392521 -0.6017271 0.7184596 Cohort 50 Cohort     50
```

```
## Same, without confidence intervals
corr_noci <- bdotsCorr(fit, val = "val")
head(corr_noci)
```

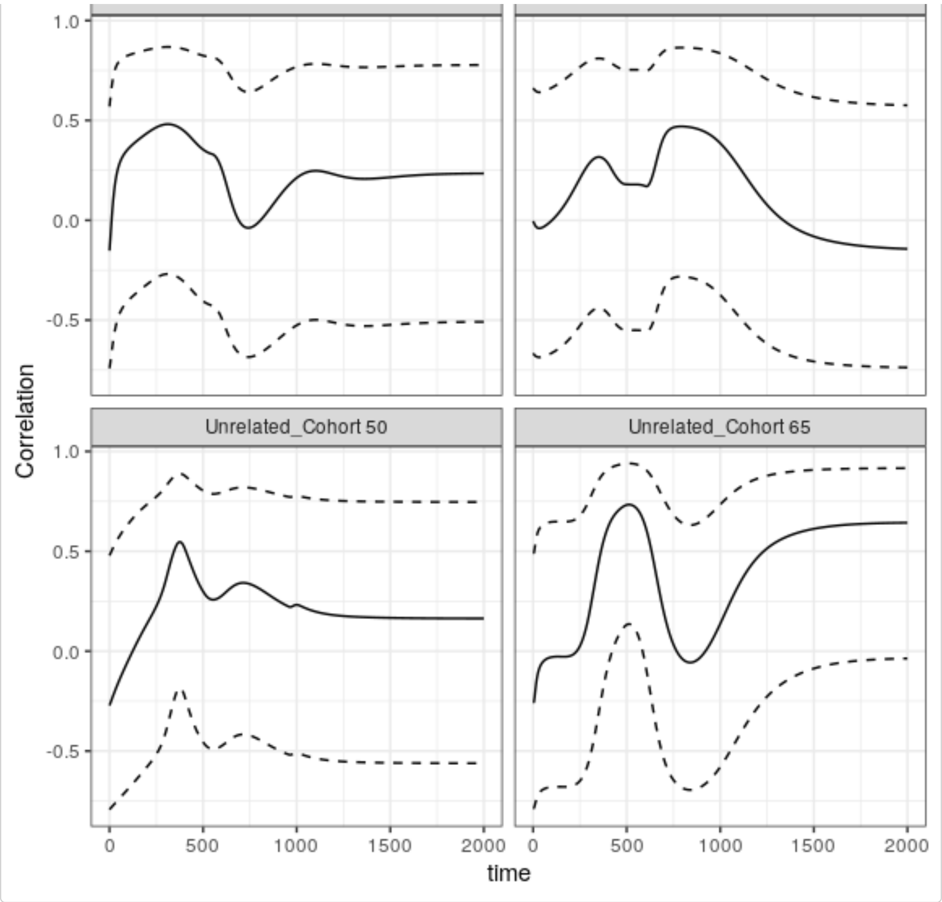
```
##    time Correlation     Group Group1 Group2
## 1:    0 -0.15244734 Cohort 50 Cohort     50
## 2:    4 -0.09788149 Cohort 50 Cohort     50
## 3:    8 -0.04195758 Cohort 50 Cohort     50
## 4:   12  0.01181996 Cohort 50 Cohort     50
## 5:   16  0.06087569 Cohort 50 Cohort     50
## 6:   20  0.10392521 Cohort 50 Cohort     50
```

From here, we are able to use the `data.tables` themselves for whatever we may be interested in. We also have a plotting method associated with this object

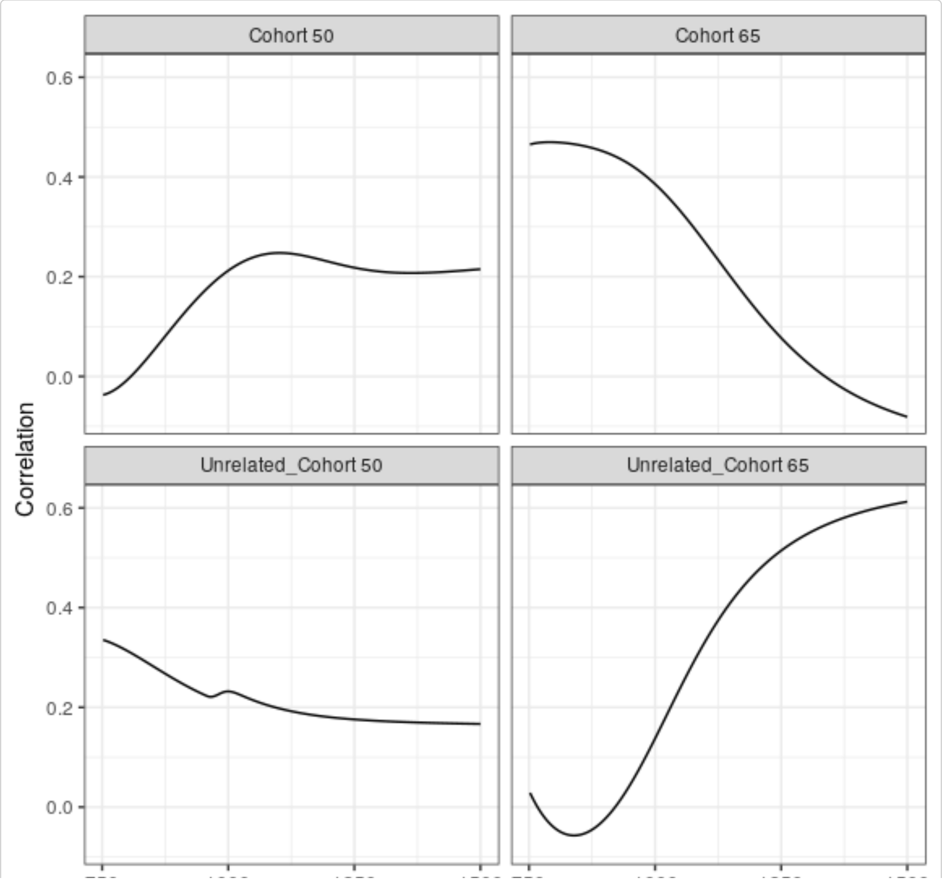
```
## Default is no bands
plot(corr_ci)
```

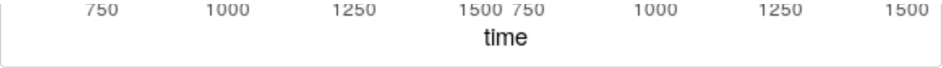


```
## Try again with bands
plot(corr_ci, ciBands = TRUE)
```



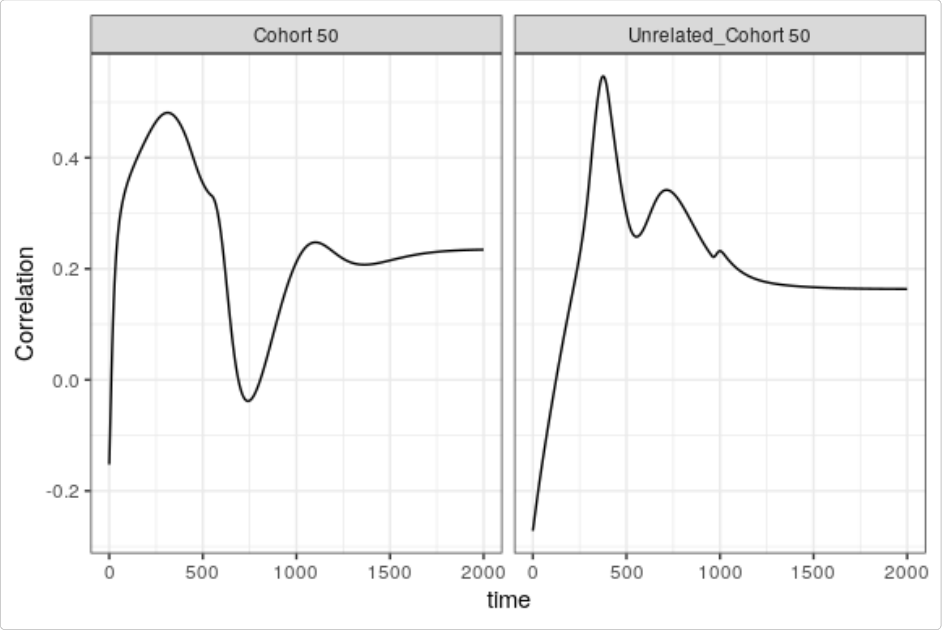
```
## Narrow in on a particular window
plot(corr_ci, window = c(750, 1500))
```





Because this object is a `data.table`, we have full use of subsetting capabilities for our plots

```r
plot(corr_ci[Group2 == "50", ])
```


Cohort 50
Unrelated_Cohort 50
Correlation
time