Correlation Function

Correlations in bdots

This vignette is created to illustrate the use of the `bcorr` 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 `bfit`.

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  0.011364   Cohort    50 0.014774
## 2:       1    4      50  0.011364   Cohort    50 0.014774
## 3:       1    8      50  0.011364   Cohort    50 0.014774
## 4:       1   12      50  0.011364   Cohort    50 0.014774
## 5:       1   16      50  0.022727   Cohort    50 0.014774
## 6:       1   20      50  0.022727   Cohort    50 0.014774
```

Now, we go about creating our fitted object as usual

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

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

1. `bdObj`, any object returned from a `bfit` 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 <- bcorr(fit, val = "val", ciBands = TRUE)
head(corr_ci)
```

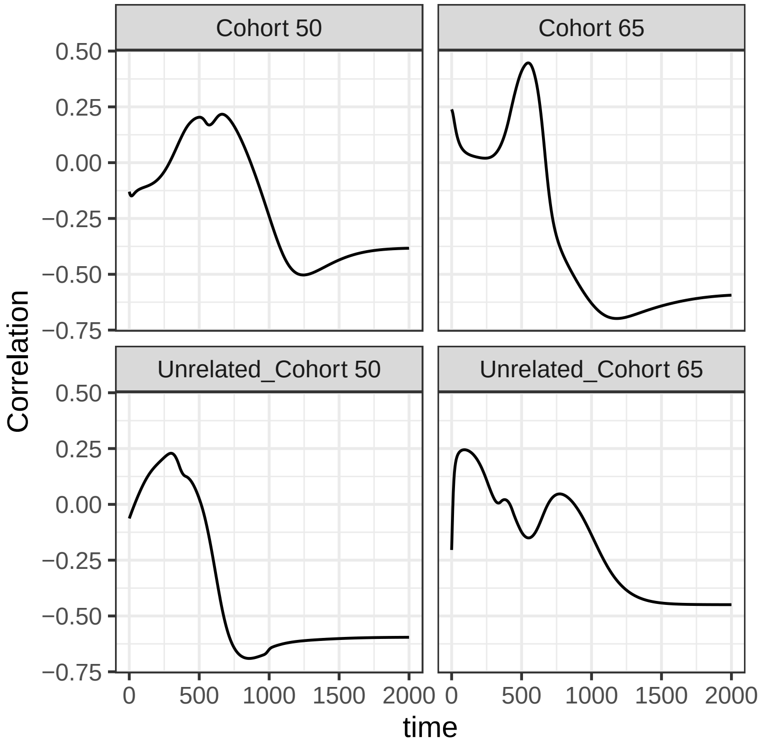
```
##    time Correlation    lower   upper     Group Group1 Group2
## 1:    0    -0.13010 -0.73106 0.58453 Cohort 50 Cohort     50
## 2:    4    -0.13903 -0.73526 0.57851 Cohort 50 Cohort     50
## 3:    8    -0.14515 -0.73812 0.57434 Cohort 50 Cohort     50
## 4:   12    -0.14842 -0.73963 0.57209 Cohort 50 Cohort     50
## 5:   16    -0.14927 -0.74003 0.57151 Cohort 50 Cohort     50
## 6:   20    -0.14836 -0.73961 0.57213 Cohort 50 Cohort     50
```

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

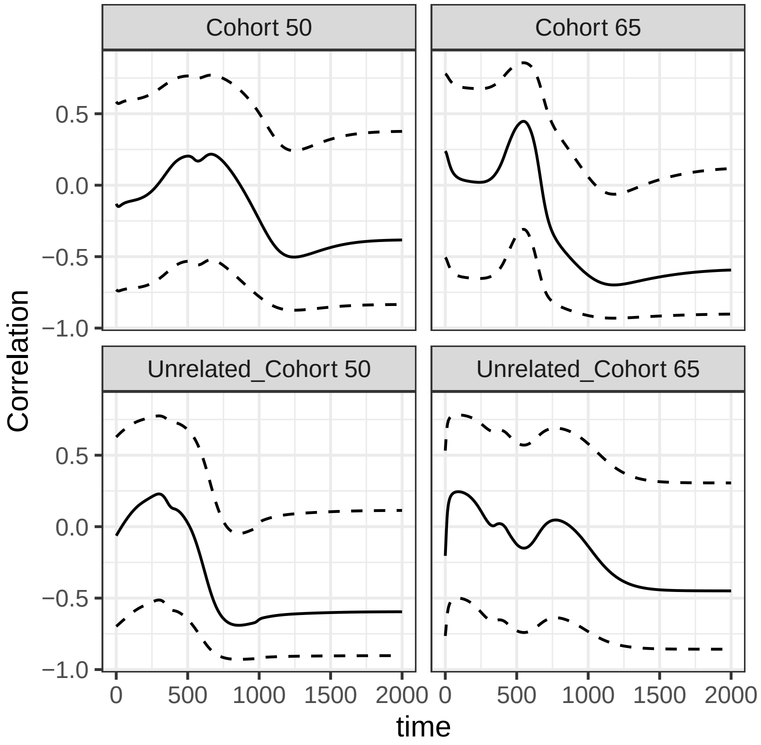
```
##    time Correlation     Group Group1 Group2
## 1:    0    -0.13010 Cohort 50 Cohort     50
## 2:    4    -0.13903 Cohort 50 Cohort     50
## 3:    8    -0.14515 Cohort 50 Cohort     50
## 4:   12    -0.14842 Cohort 50 Cohort     50
## 5:   16    -0.14927 Cohort 50 Cohort     50
## 6:   20    -0.14836 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

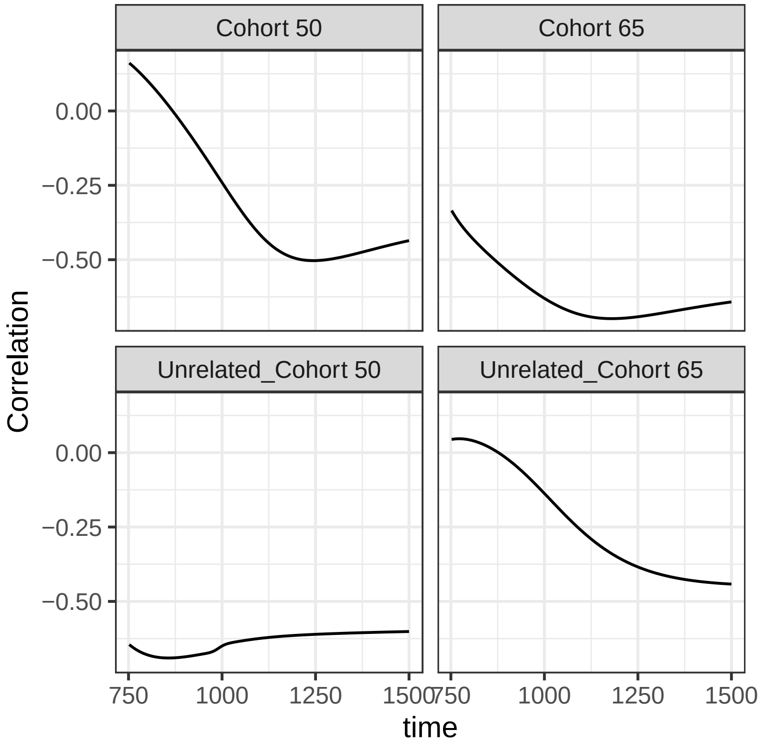
```
## Default is no bands
plot(corr_ci)
```



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

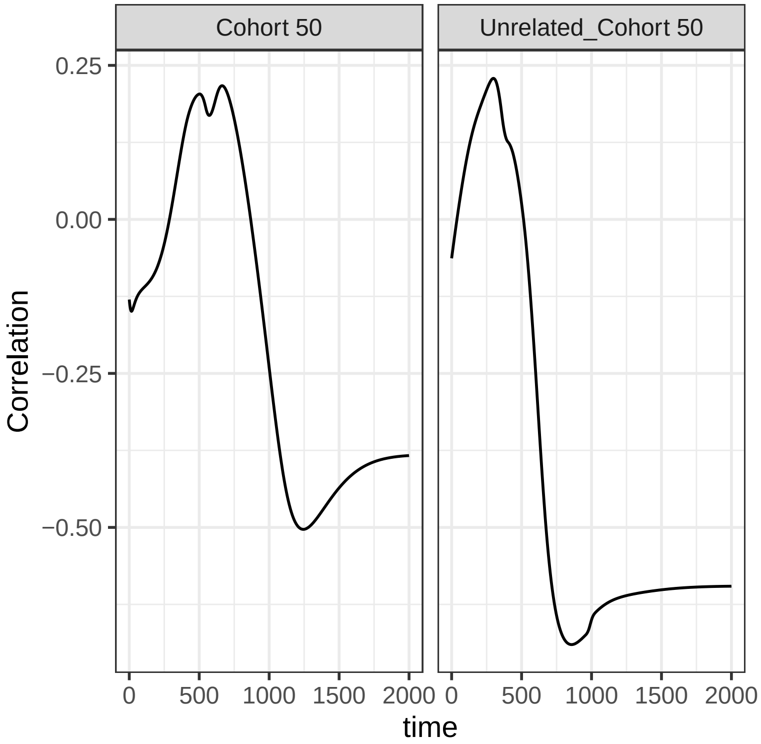


```r
## Narrow in on a particular window
plot(corr_ci, window = c(750, 1500))
```



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

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


Cohort 50
Unrelated_Cohort 50
Correlation
time