

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 -1.417
## 2:         1    4      50  0.011364   Cohort    50 -1.417
## 3:         1    8      50  0.011364   Cohort    50 -1.417
## 4:         1   12      50  0.011364   Cohort    50 -1.417
## 5:         1   16      50  0.022727   Cohort    50 -1.417
## 6:         1   20      50  0.022727   Cohort    50 -1.417
```

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    -0.40588 -0.84282  0.35354 Cohort  50 Cohort    50
```

```
## 2:    4    -0.39775 -0.83999 0.36199 Cohort 50 Cohort    50
## 3:    8    -0.38174 -0.83434 0.37829 Cohort 50 Cohort    50
## 4:   12    -0.35976 -0.82643 0.39991 Cohort 50 Cohort    50
## 5:   16    -0.33438 -0.81706 0.42387 Cohort 50 Cohort    50
## 6:   20    -0.30798 -0.80703 0.44772 Cohort 50 Cohort    50
```

```
## Same, without confidence intervals
```

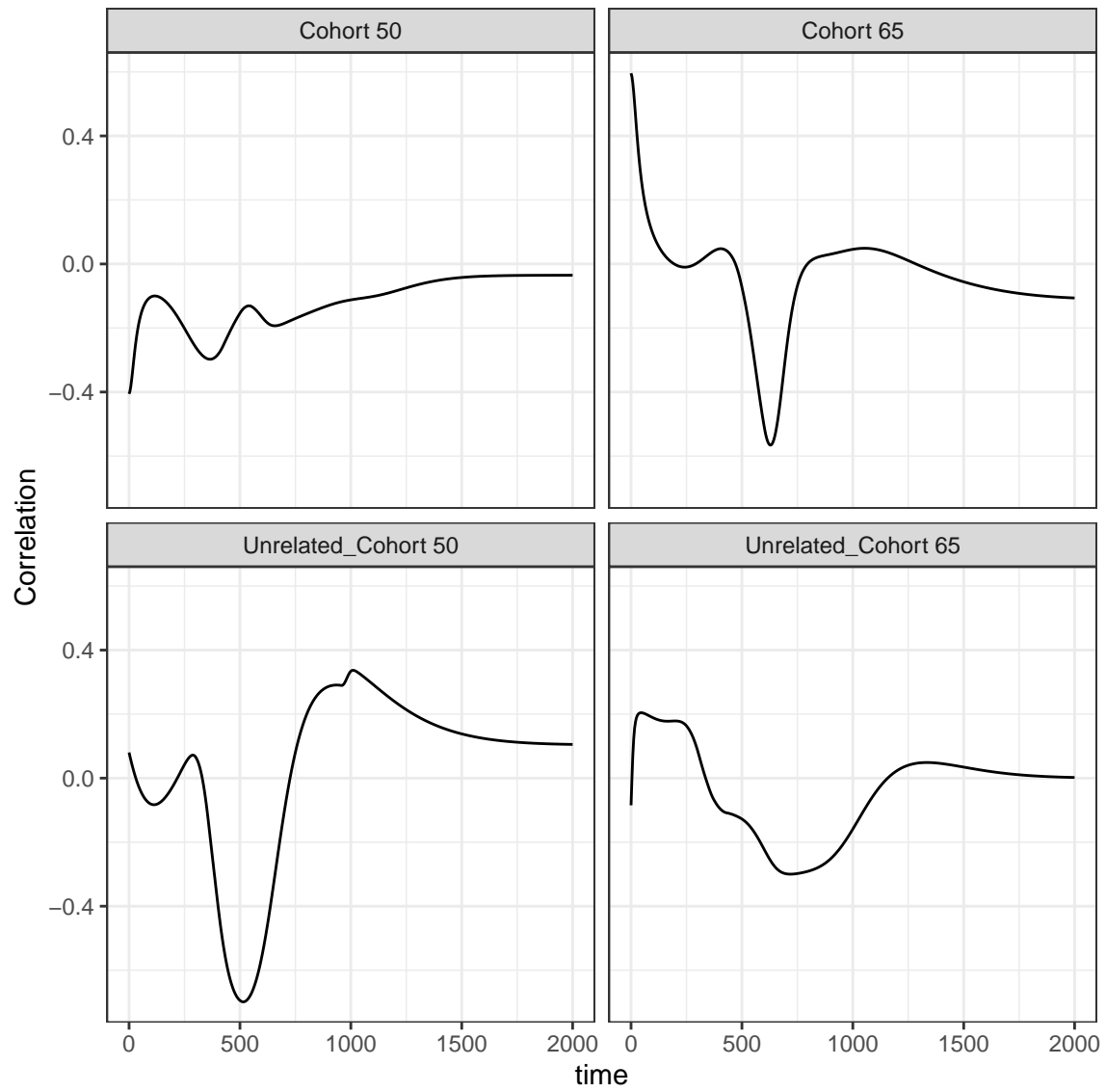
```
corr_noci <- bcorr(fit, val = "val")
head(corr_noci)
```

```
##      time Correlation      Group Group1 Group2
## 1:    0    -0.40588 Cohort 50 Cohort    50
## 2:    4    -0.39775 Cohort 50 Cohort    50
## 3:    8    -0.38174 Cohort 50 Cohort    50
## 4:   12    -0.35976 Cohort 50 Cohort    50
## 5:   16    -0.33438 Cohort 50 Cohort    50
## 6:   20    -0.30798 Cohort 50 Cohort    50
```

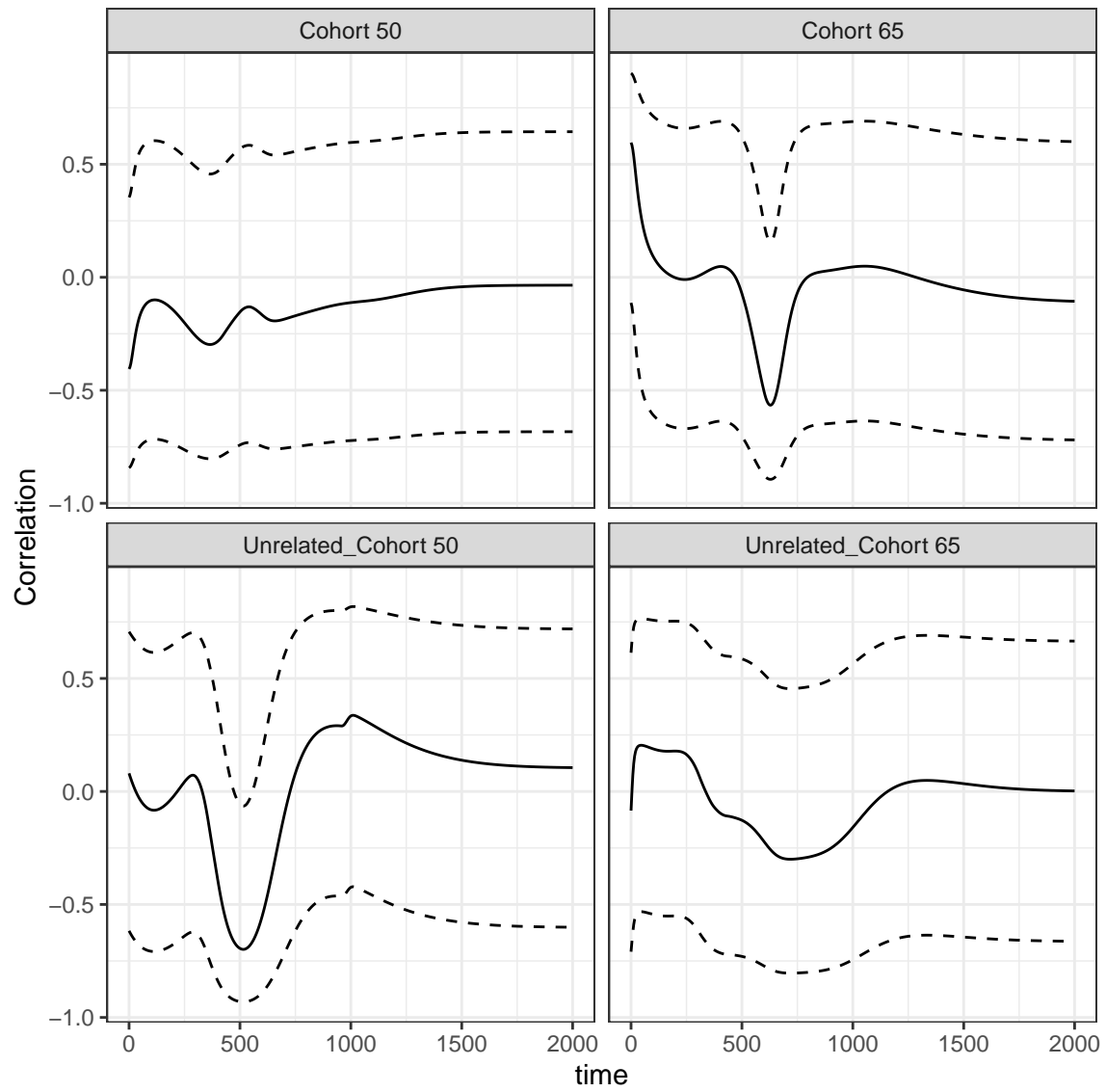
From here, we are able to use the `data.tables` themselves for whatever we may be interested in. We also have a plotting method associated with this object

```
## Default is no bands
```

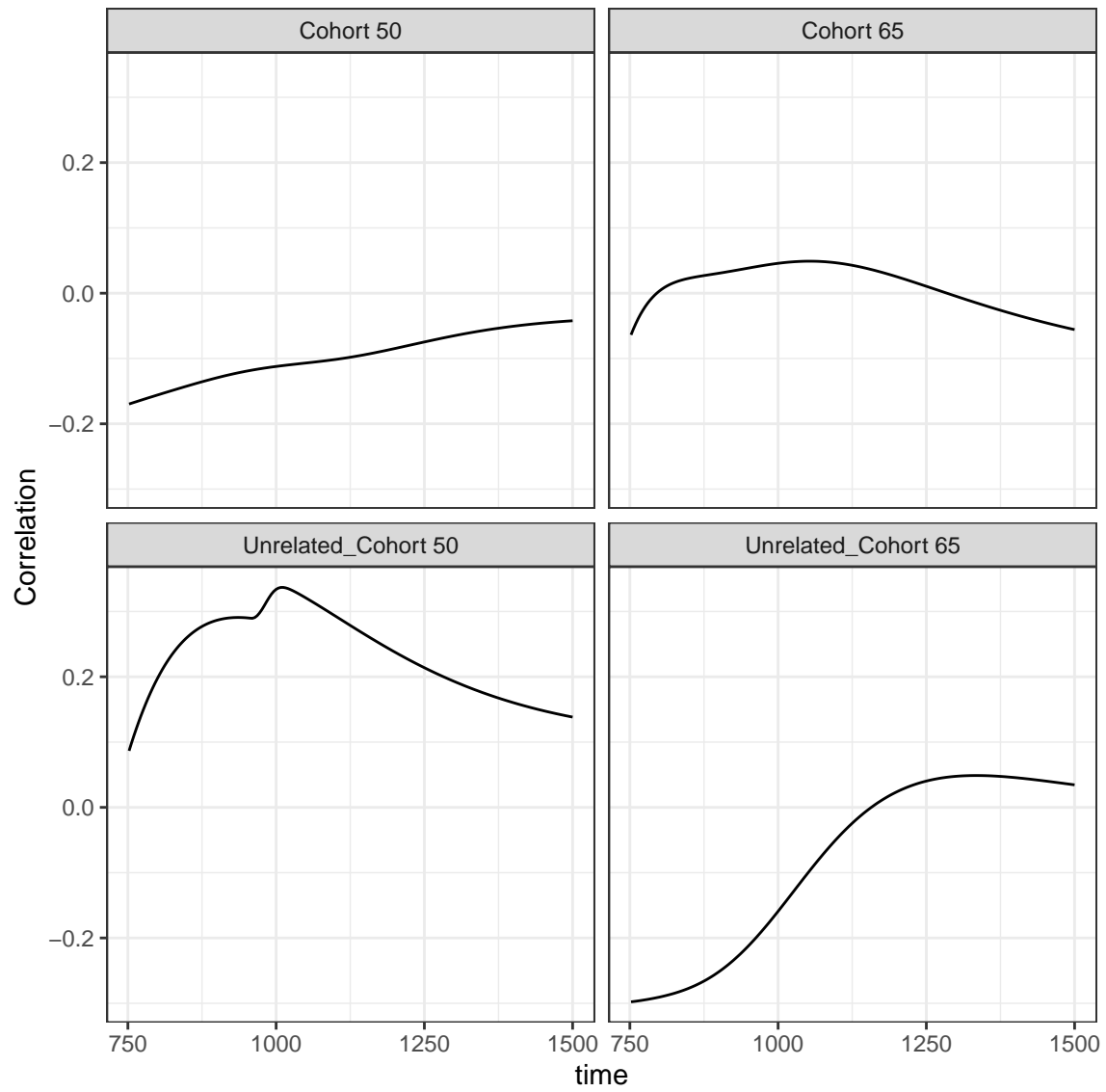
```
plot(corr_ci)
```



```
## 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

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

