

Whitfield test

R. Chlumsky

June 4, 2018

Whitfield merge

General comments

- there had been a consensus to use the Google guidelines for naming conventions, should switch the function names to booth.plot instead of booth_plot, etc.
- need to update all of the documentation to be more descriptive and complete
- need to update examples and remove where possible using built in sample data

Review of Paul's package by function below.

booth plot

- get_peaks is not clearly defined in package, not sure where this function is
- should consider merging the get_peaks function into the hydRology package and calling it internally from within booth_plot to generalize the input as single hydrograph series instead of linking function output -> function -> function
- can use general utility functions within the hydRology package to call the months, common colours, etc., rather than hardcoding into each function; colour schemes and other utilities will be common between functions

doys

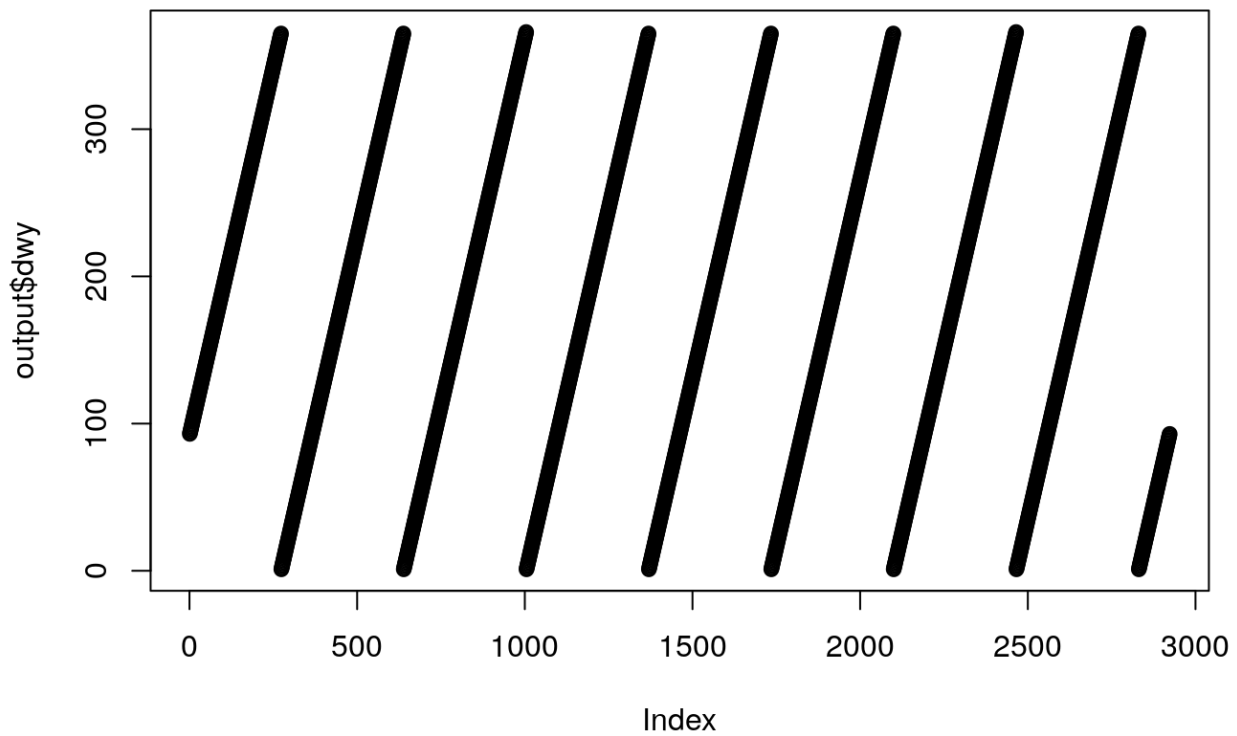
- description of the function should be updated to full sentence format
- could modify function to allow the specification of the water year with October 1st as a default, instead of hardcoding

doys calculates the numeric calendar year, the day of year, the water year, and the day of water year for each day of the date series supplied.

```
library(CSHShydRology)
dd <- seq.Date(as.Date("2010-01-01"), as.Date("2018-01-01"), by=1)
output <- doys(dd)
head(output)
```

```
##      Date year doy wyear dwy
## 1 2010-01-01 2010  1 2010  93
## 2 2010-01-02 2010  2 2010  94
## 3 2010-01-03 2010  3 2010  95
## 4 2010-01-04 2010  4 2010  96
## 5 2010-01-05 2010  5 2010  97
## 6 2010-01-06 2010  6 2010  98
```

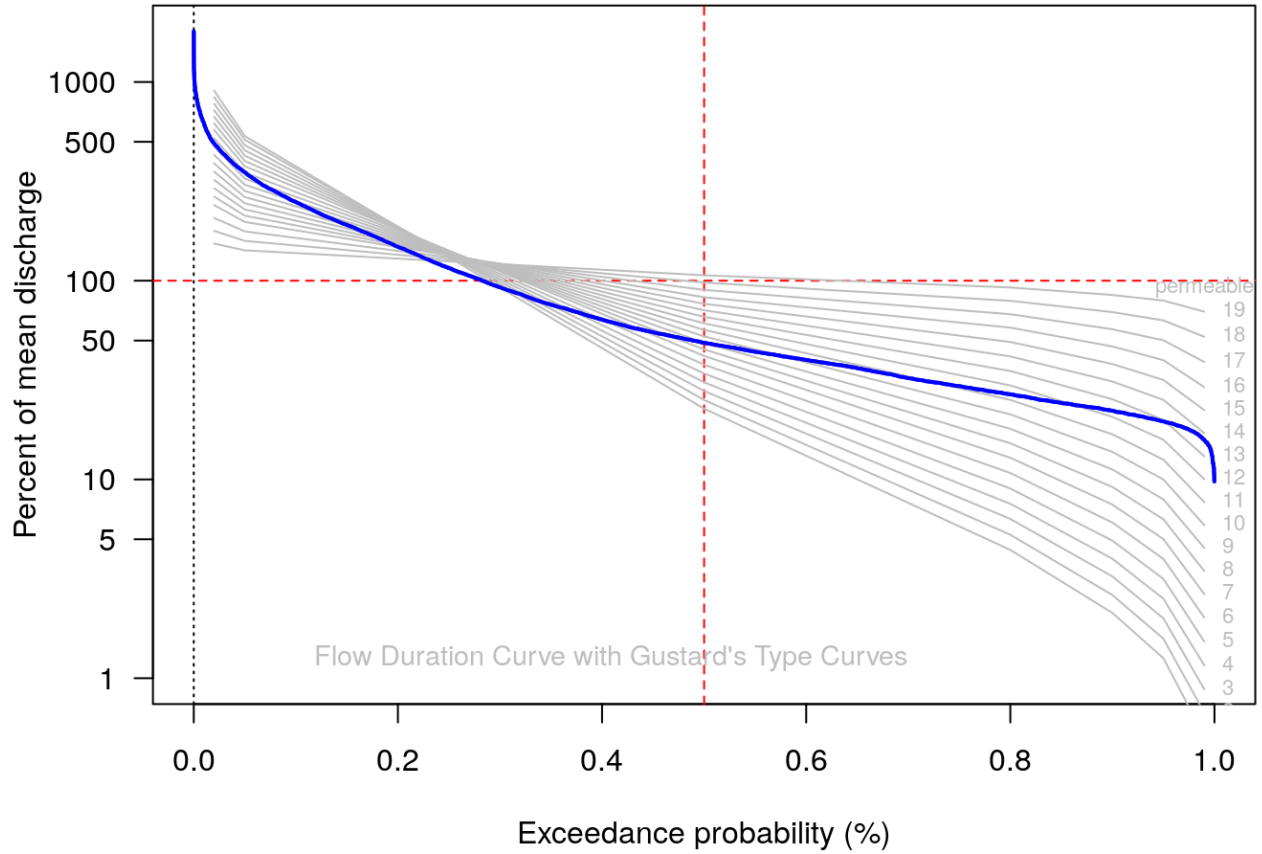
```
plot(output$dwy)
```



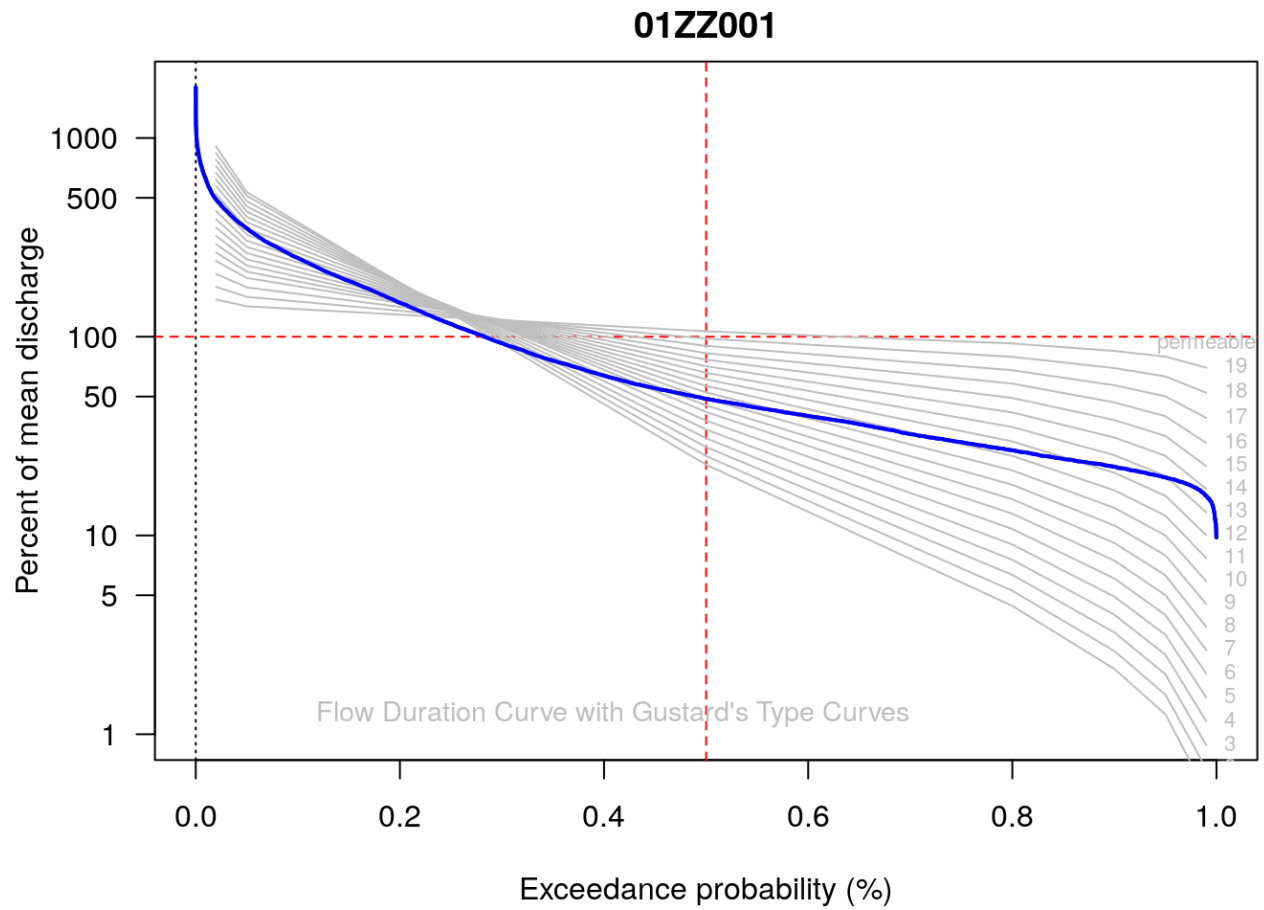
- update the example with something like the above, remove the

fdcurve

```
data("W05AA008")
dd <- W05AA008
# fdcurve(dd$Flow) # results in error with no title argument
fdcurve(dd$Flow,title="")
```

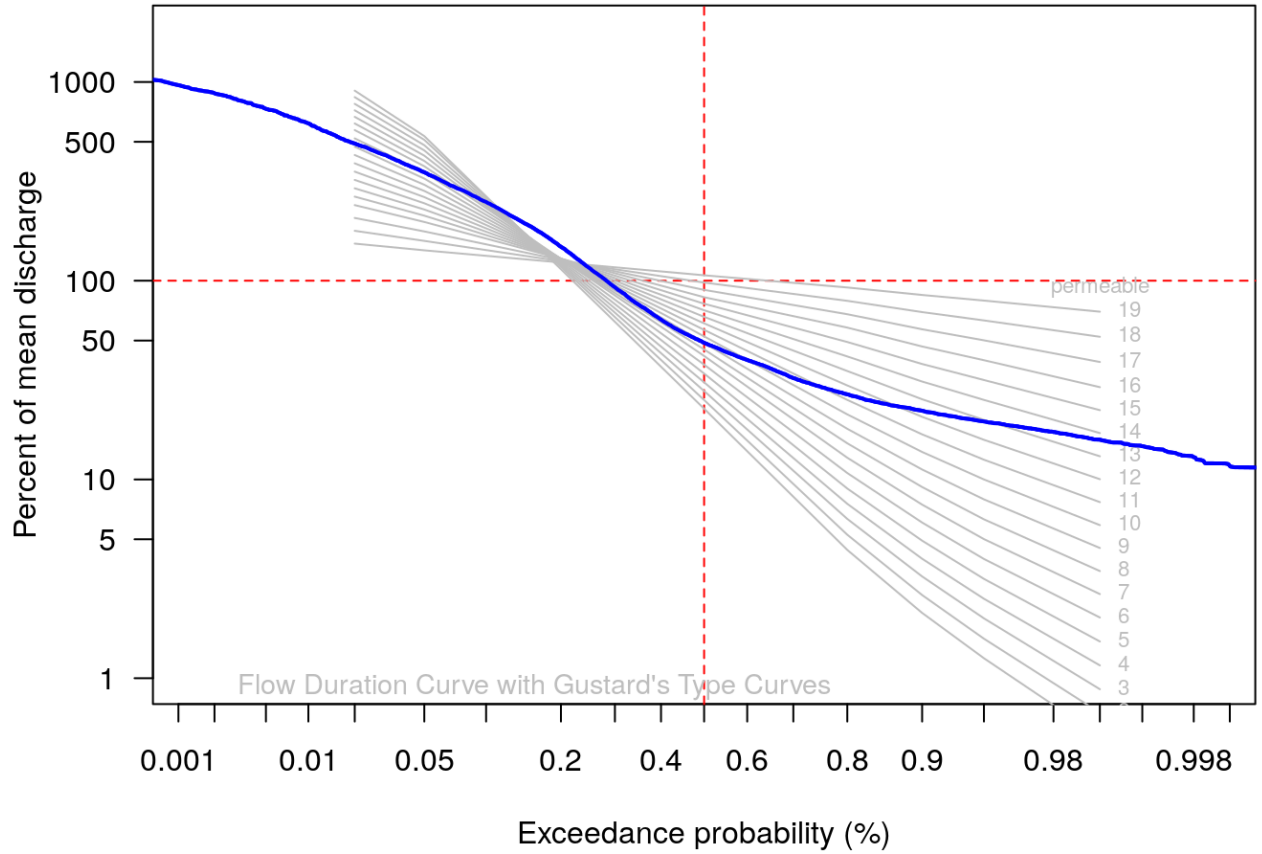


```
fdcurve(dd$Flow,title="01ZZ001") # adds any title
```



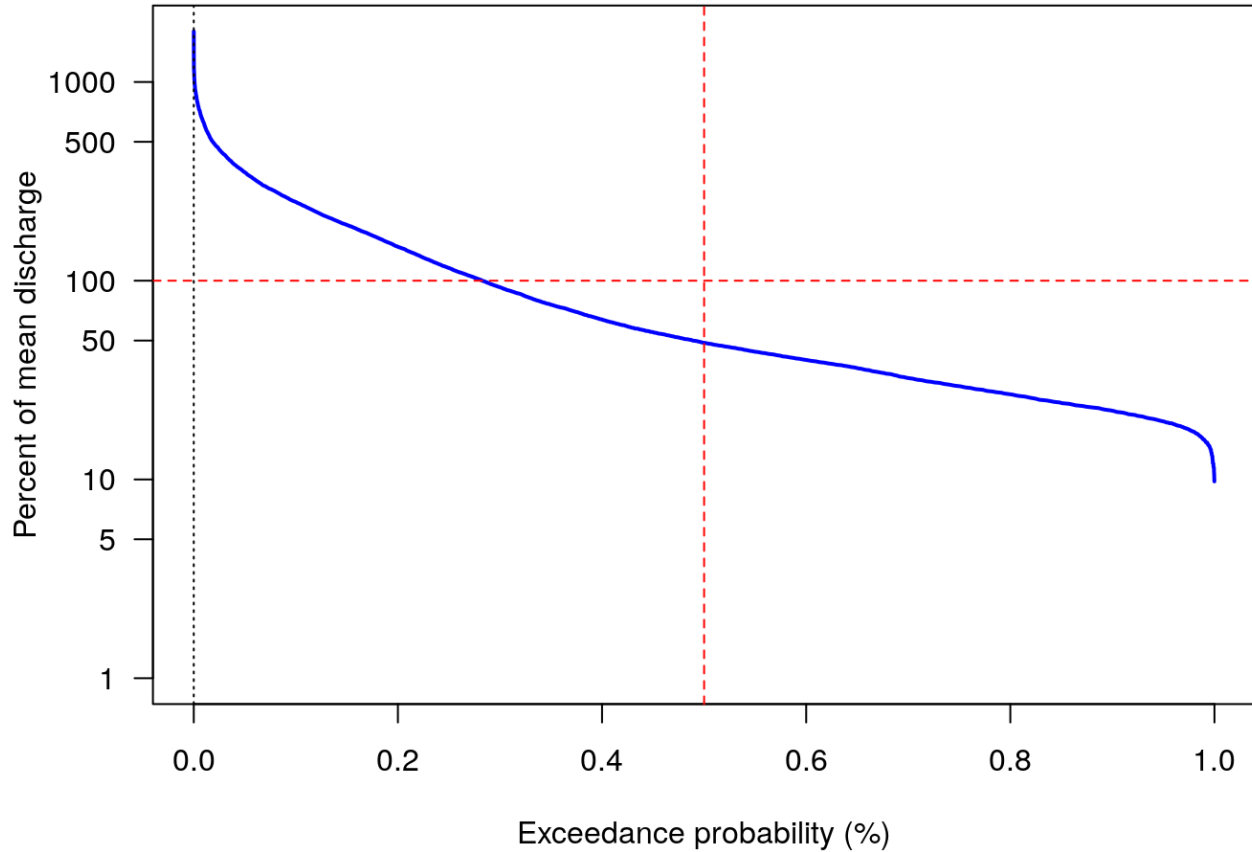
Can use the *normal = TRUE* argument to normalize the exceedance probability

```
fdcurve(dd$Flow,title="",normal =T) # normalized exceedance probability
```



Can also turn off the 'gust' curves from Gustard et al 1992.

```
fdcurve(dd$Flow,title="",gust = F) # Gustard et al 1992 curves turned off
```



Recommended modifications to the `fdcurve` function:

- recommend making `title=""` as a default argument OR reading it from the `names(flow)` (i.e. from the `flows` column name as default if nothing provided)
- explain how the normalization is calculated in the documentation

get_wscstation

Returns data from the internally stored `HYDAT_List`.

- `get_wscstation` does not seem to be recognized when the package is loaded, loaded locally for testing purposes;
- when markdown ran, returned an error with “Error: \$ operator is invalid for atomic vectors.”. Recommend reviewing this once function is recognized in package

```
# mystn <- get_wscstation(stnID="01AD004")
# head(mystn)
```

Try with multiple `stn ids` in a vector, does not seem to recognize them properly but also does not return an error.

```
# mystn <- get_wscstation(stnID=c("01AD004","01AA002","01AF006")) # causes an error
# head(mystn)
```

Recommended fixes:

- remove stn as a parameter; store HYDAT_list internally as per KS comments
- fix so that it is exported properly and recognized as a package function
- adjust to allow for use of multiple stnIDs to be queried simultaneously

HYDAT_list

- not clear if this is a static dataframe or retrieved with each run of the function, should make that clear
- usage - HYDAT_list shown twice
- typo in dataframe <- dataframe
- change the dontrun example

read_wsc, raster_trend, raster_qa

- not reviewed

slice

Converts data into bins based upon a chosen step size. Use the doy from sample data set

```
data("W05AA008")
dd <- doys(W05AA008$Date)
ss <- slice(dd$doy[1:100],step = 5)
```

```
## [1] "Bins = 73 The number of extra points in last bin is up to 1 per year"
```

```
head(ss)
```

```
## [1] 42 43 43 43 43 43
## 73 Levels: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 ... 73
```

```
ss <- slice(dd$doy[500:1000],step = 10)
```

```
## [1] "Bins = 36 The number of extra points in last bin is up to 6 per year"
```

```
head(ss)
```

```
## [1] 4 5 5 5 5 5  
## 36 Levels: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 ... 36
```

General comments:

- overall purpose of this function is not clear. Need better documentation+examples within function help