StormR

```r
library(StormR)
```

Introduction

This vignette teaches you how to use StormR package solving basic problems. The StormsDataset used within this document relies on the IBTrACS.SP.v04r00.nc. This dataset gathers all tropical depressions, storms and cyclones that occured in the South Pacific ocean over since 1980. It is available in this package and named IBTRACS_SP and is also the default setting to retrieve storms using getStorms function (See next section).

Solve common problems

Get data associated with storms

Once the StormsDataset is loaded, the first thing to do is to select storms we are interested in. This operation is done using getStorms function. It collects the storms coming from a StormsDataset (sds input) over a certain Location Of Interest (loi input). This searching location is extended using the max_dist input, and default value is set to 300km. Note that These 2 inputs are mandatory to perform this function (See getStorms Documentation). It is also possible to filter the storms by their cyclonic seasons and their names (season and input names). Finally, storms with maximum wind speed inferior to 18 m/s (Tropical Depressions in the Saffir Simpson Hurricane Scale SSHS) can be ignored using remove_TD logical input. Default value is set to TRUE. Here are some basic usage of the getStorms function.

In this case, we get the data associated with the tropical cyclone Harold that hit Vanuatu in 2020. Note here that the loi represents a whole country. (See getStorms documentation to get the full list of country available).

```r
harold <- getStorms(loi = "Vanuatu", names = "HAROLD")
```

```
## === getStorms processing ... ===
##
## -> Making buffer: Done
## -> Searching for HAROLD storm ...
##    -> Identifying Storms: Done
## -> Gathering storm(s) ...
##   |                                                                              |
##
## === DONE with run time 1.853966 sec ===
##
## SUMMARY:
## (*) LOI: Vanuatu
## (*) Buffer size: 300 km
## (*) Remove Tropical Depressions (< 18 m/s in sshs): yes
## (*) Number of Storms: 1
##         Name - Tropical season - SSHS - Number of observation within buffer:
##         HAROLD - 2020 - 5 - 26
```

In this second example, we collect data for all tropical storms and cyclones over the Exclusive Economic Zone of New Caledonia (eezNC) between 2000 and 2022. The loi here is a sf object, but it can also be a shapefile.

```
sts.nc <- getStorms(loi = eezNC, seasons = c(2000,2022))
```

```
## === getStorms processing ... ===
##
## -> Making buffer: Done
## -> Searching storms from 2000 to 2022 ...
##    -> Identifying Storms: 185 potential candidates...
## -> Gathering storm(s) ...
##   |                                                                              |
```

```
##         FREDA - 2013 - 3 - 42
##         SANDRA - 2013 - 3 - 49
##         JUNE - 2014 - 0 - 21
##         EDNA - 2014 - 0 - 19
##         HADI - 2014 - 0 - 1
##         LUSI - 2014 - 1 - 29
##         ITA - 2014 - 5 - 11
##         OLA - 2015 - 2 - 36
##         MARCIA - 2015 - 4 - 11
##         PAM - 2015 - 5 - 13
##         SOLO - 2015 - 0 - 29
##         ULA - 2016 - 4 - 46
##         TATIANA - 2016 - 0 - 31
##         WINSTON - 2016 - 5 - 79
##         ZENA - 2016 - 2 - 12
##         COOK - 2017 - 2 - 33
##         DONNA - 2017 - 4 - 44
##         FEHI - 2018 - 0 - 28
##         GITA - 2018 - 4 - 31
##         HOLA - 2018 - 3 - 33
##         LINDA - 2018 - 0 - 30
##         IRIS - 2018 - 0 - 66
##         JOSIE - 2018 - 0 - 7
##         OWEN - 2019 - 2 - 13
##         PENNY - 2019 - 0 - 22
##         OMA - 2019 - 1 - 126
##         ANN - 2019 - 0 - 19
##         UESI - 2020 - 1 - 43
##         GRETEL - 2020 - 1 - 19
##         HAROLD - 2020 - 5 - 25
##         LUCAS - 2021 - 1 - 29
##         NIRAN - 2021 - 5 - 16
##         RUBY - 2022 - 1 - 28
##         SETH - 2022 - 0 - 25
##         CODY - 2022 - 0 - 31
##         DOVI - 2022 - 2 - 21
##         FILI - 2022 - 0 - 43
##         GINA - 2022 - 0 - 24
```

In this last example, we retrieve all data associated with tropical storms and cyclones that occured since 1980 around the point coordinate 188.17: -13.92 (longitude, latitude decimal degree) within a 300km buffer. These coordinates are actually located in the American Samoa.

```
pt <- c(188.17,-13.92)
sts.pt <- getStorms(loi = pt)
```

```
## === getStorms processing ... ===
##
## -> Making buffer: Done
## -> Searching storms from 1980 to 2022 ...
##    -> Identifying Storms: 386 potential candidates...
## -> Gathering storm(s) ...
##   |                                                                            |
```

```
##
## SUMMARY:
## (*) LOI: 188.17 -13.92 lon-lat
## (*) Buffer size: 300 km
## (*) Remove Tropical Depressions (< 18 m/s in sshs): yes
## (*) Number of Storms: 30
##         Name - Tropical season - SSHS - Number of observation within buffer:
##         ESAU - 1981 - 0 - 5
##         TUSI - 1987 - 3 - 6
##         WINI - 1987 - 1 - 7
##         ZUMAN - 1987 - 0 - 8
##         GINA - 1989 - 0 - 11
##         OFA - 1990 - 4 - 11
##         VAL - 1992 - 4 - 29
##         GENE - 1992 - 0 - 10
##         LIN - 1993 - 2 - 10
##         MICK - 1993 - 0 - 20
##         EVAN - 1997 - 1 - 25
##         KELI - 1997 - 4 - 5
##         TUI - 1998 - 0 - 21
##         WES - 1998 - 0 - 3
##         HETA - 2004 - 5 - 9
##         OLAF - 2005 - 5 - 8
##         NANCY - 2005 - 4 - 2
##         URMIL - 2006 - 0 - 4
##         ARTHUR - 2007 - 1 - 7
##         NISHA - 2010 - 0 - 5
##         RENE - 2010 - 3 - 12
##         WILMA - 2011 - 4 - 23
##         EVAN - 2013 - 4 - 23
##         GARRY - 2013 - 2 - 12
##         TUNI - 2016 - 0 - 8
##         AMOS - 2016 - 2 - 10
##         ELLA - 2017 - 1 - 14
##         GITA - 2018 - 4 - 7
##         VICKY - 2020 - 0 - 10
##         WASI - 2020 - 0 - 13
```

Access data

The getStorms function returns data collected from tropical storms and cyclone in a Storms object especially designed for this purpose (See Storms class). Then , one can be interested in getting basics informations from a Storms object initialized with getStorms. However the structure of this object is quite complex and it can rapidly become overwhelming trying to reach data on your own. Here are some getters that will help you saving time to access data.

From now on, we demonstrate how to use it using the sts.nc Storms object initialized right above.

First of all, if you are interested in getting all the storms names just run the following getter:

```
getNames(sts = sts.nc)
```

```
##  [1] "IRIS"    "JO"      "VAUGHAN" "PAULA"   "SOSE"    "CLAUDIA" "DES"
##  [8] "ZOE"     "BENI"    "ERICA"   "ESETA"   "GINA"    "IVY"     "GRACE"
## [15] "KERRY"   "JIM"     "LARRY"   "WATI"    "YANI"    "BECKY"   "FUNA"
## [22] "GENE"    "INNIS"   "HAMISH"  "JASPER"  "RENE"    "ULUI"    "VANIA"
```

```
## [29] "ZELIA"    "WILMA"    "ANTHONY" "YASI"    "ATU"     "JASMINE" "DAPHNE"
## [36] "FREDA"    "SANDRA"   "JUNE"    "EDNA"    "HADI"    "LUSI"    "ITA"
## [43] "OLA"      "MARCIA"   "PAM"     "SOLO"    "ULA"     "TATIANA" "WINSTON"
## [50] "ZENA"     "COOK"     "DONNA"   "FEHI"    "GITA"    "HOLA"    "LINDA"
## [57] "IRIS"     "JOSIE"    "OWEN"    "PENNY"   "OMA"     "ANN"     "UESI"
## [64] "GRETEL"   "HAROLD"   "LUCAS"   "NIRAN"   "RUBY"    "SETH"    "CODY"
## [71] "DOVI"     "FILI"     "GINA"
```

Also, for each storms in your Storms object, the following getters will return the cyclonic season and the maximum category reached in the SSHS:

```
#Get cyclonic seasons
getSeasons(sts = sts.nc)
```

```
##    IRIS      JO VAUGHAN   PAULA    SOSE CLAUDIA     DES     ZOE    BENI   ERICA
##    2000    2000    2000    2001    2001    2002    2002    2003    2003    2003
##   ESETA    GINA     IVY   GRACE   KERRY     JIM   LARRY    WATI    YANI   BECKY
##    2003    2003    2004    2004    2005    2006    2006    2006    2007    2007
##    FUNA    GENE  INNIS  HAMISH  JASPER    RENE    ULUI   VANIA   ZELIA   WILMA
##    2008    2008    2009    2009    2009    2010    2010    2011    2011    2011
## ANTHONY    YASI     ATU JASMINE  DAPHNE   FREDA  SANDRA    JUNE    EDNA    HADI
##    2011    2011    2011    2012    2012    2013    2013    2014    2014    2014
##    LUSI     ITA     OLA  MARCIA     PAM    SOLO     ULA TATIANA WINSTON    ZENA
##    2014    2014    2015    2015    2015    2015    2016    2016    2016    2016
##    COOK   DONNA    FEHI    GITA    HOLA   LINDA    IRIS   JOSIE    OWEN   PENNY
##    2017    2017    2018    2018    2018    2018    2018    2018    2019    2019
##     OMA     ANN    UESI  GRETEL  HAROLD   LUCAS   NIRAN    RUBY    SETH    CODY
##    2019    2019    2020    2020    2020    2021    2021    2022    2022    2022
##    DOVI    FILI    GINA
##    2022    2022    2022
```

```
#Get maximum reached category in SSHS
getSSHS(sts = sts.nc)
```

```
##    IRIS      JO VAUGHAN   PAULA    SOSE CLAUDIA     DES     ZOE    BENI   ERICA
##       1       1       0       3       1       1       0       5       4       4
##   ESETA    GINA     IVY   GRACE   KERRY     JIM   LARRY    WATI    YANI   BECKY
##       3       2       3       0       2       1       4       1       1       1
##    FUNA    GENE  INNIS  HAMISH  JASPER    RENE    ULUI   VANIA   ZELIA   WILMA
##       3       3       0       4       0       3       5       0       2       4
## ANTHONY    YASI     ATU JASMINE  DAPHNE   FREDA  SANDRA    JUNE    EDNA    HADI
##       0       4       4       4       0       3       3       0       0       0
##    LUSI     ITA     OLA  MARCIA     PAM    SOLO     ULA TATIANA WINSTON    ZENA
##       1       5       2       4       5       0       4       0       5       2
##    COOK   DONNA    FEHI    GITA    HOLA   LINDA    IRIS   JOSIE    OWEN   PENNY
##       2       4       0       4       3       0       0       0       2       0
##     OMA     ANN    UESI  GRETEL  HAROLD   LUCAS   NIRAN    RUBY    SETH    CODY
##       1       0       1       1       5       1       5       1       0       0
##    DOVI    FILI    GINA
##       2       0       0
```

This getter simply returns the number of storms provided in your Storms Object:

```
getNbStorms(sts = sts.nc)
```

```
## [1] 73
```

The next 3 getters are useful to retrieve spatial informations on the Location Of Interest of your Storms object. The first command will return the LOI converted in sf format:

```
getLOI(sts = sts.nc)
```

```
## Simple feature collection with 1 feature and 0 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 156.2557 ymin: -26.20108 xmax: 174.2757 ymax: -14.82636
## Geodetic CRS:  WGS 84
##                           loi.sf
## 1 POLYGON ((164.1694 -15.9122...
```

This second command simply returns the size (in km) of the buffer used to extent the LOI:

```
getBufferSize(sts = sts.nc)
```

```
## [1] 300
```

Finally this third command provides the LOI extended with the buffer in sf format.

```
getBuffer(sts = sts.nc)
```

```
## Simple feature collection with 1 feature and 0 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 153.3918 ymin: -28.97837 xmax: 177.3811 ymax: -12.05625
## Geodetic CRS:  WGS 84
##                           loi.sf
## 1 POLYGON ((168.2932 -16.0239...
```

One can also be interested in getting all informations about a particular storm. This operation is achieved using the following getter:

```
niran <- getStorm(sts = sts.nc, name = "NIRAN")
```

Note: If serveral storms share the same name, you must specify the cyclonic season to differenciate them. For example, 2 storms named Evan are provided within the sts.pt Storms object initialized in the first section. This first command will then return an error, as we did not specify which one we are interested in.

```
#getStorm(sts = sts.pt, name = "EVAN")
```

We thus tackle this issue using the next 2 commands:

```
evan1997 <- getStorm(sts = sts.pt, name = "EVAN", season = 1997)
evan2013 <- getStorm(sts = sts.pt, name = "EVAN", season = 2013)
```

All theses getters are designed to retrieve general informations on first levels of Storms objects. However we can go further into the object getting data of a particular storm. Combined with the getStorm getter, these following command perform really well.

This command provides the cyclonic season of a particular storm:

```
#getStorm_season(niran)
#Equivalent to getSeason(getStorm(sts = sts.nc, name = "NIRAN"))
```

This command provides the maximum category reached in the sshs for a particular storm:

```
#getStorm_sshs(niran)
#Equivalent to getsshs(getStorm(sts = sts.nc, name = "NIRAN"))
```

This command provides the number of observations available for a particular storm:

```
#getStorm_nbObs(niran)
#Equivalent to getNbObs(getStorm(sts = sts.nc, name = "NIRAN"))
```

This command provides all the observations of a particular storm:

```
#getStorm_obs(niran)
#Equivalent to getObs(getStorm(sts = sts.nc, name = "NIRAN"))
```

This command provides the index of observations within the spatial buffer for a particular storm:

```
#getStorm_inObs(niran)
#Equivalent to getInObs(getStorm(sts = sts.nc, name = "NIRAN"))
```

Plot data associated with storms

An interesting feature of this package is the plotStorms function which let you plot track(s) of storm(s) provided in a Storms object over the Location Of Interest, using different settings (See plotStorms documentation to get all the available input). Here are some basics usages of this function.

In this example, we plot tropical cyclone Harold track over the Vanuatu alongside with the labeled observations. Default setting are used to plot labels: every 24h and on the right side of observations.

```
plotStorms(harold, labels = TRUE)
```

In this second example, we plot tropical cyclone Erica (2003) and Cook (2017), over the EEZ of New Caledonia alongside with the labeled observations (In this case every ??H).

```
plotStorms(sts.nc, names = c("ERICA", "COOK"), labels = TRUE, by = 12)
```

In this last example, we plot every tropical cyclone that reached category 5 (SSHS) around American Samoa, alongside with the labeled observations.

```
plotStorms(sts.pt, category = 5, labels = TRUE)
```

Computing rasterized products

The most important functionality provided by this package is by far the stormBehaviour_sp function.

```
prod.harold <- stormBehaviour_sp(harold, product = c("MSW", "PDI", "Exposure"))
```

```
## === stormBehaviour_sp processing ... ===
##
## Computation settings:
##   (*) Time interpolation: Every 60 min
##   (*) Space resolution: 2.5min
##   (*) Method used: Willoughby
##   (*) Product(s) to compute: MSW PDI Exposure
##   (*) Asymmetry used: Boose01
##
## Storm(s):
##   ( 1 )  HAROLD
##
## HAROLD  ( 1 / 1 )
##   |                                                                  |
```

```
## SpatRaster stack with 8 layers:
## index - name of layers
##   1     HAROLD_MSW
##   2     HAROLD_PDI
##   3     HAROLD_Exposure_18
##   4     HAROLD_Exposure_33
##   5     HAROLD_Exposure_42
##   6     HAROLD_Exposure_49
##   7     HAROLD_Exposure_58
##   8     HAROLD_Exposure_70
```

```r
prof.harold <- stormBehaviour_sp(harold, product = "Profiles")
```

```
## === stormBehaviour_sp processing ... ===
##
## Computation settings:
##   (*) Time interpolation: Every 60 min
##   (*) Space resolution: 2.5min
##   (*) Method used: Willoughby
##   (*) Product(s) to compute: Profiles
##   (*) Asymmetry used: Boose01
##
## Storm(s):
##   ( 1 )  HAROLD
##
## HAROLD  ( 1 / 1 )
##   |                                                                              |
```

```
## 23 HAROLD_Profiles_31.1
## 24 HAROLD_Profiles_31.2
## 25 HAROLD_Profiles_32
## 26 HAROLD_Profiles_32.1
## 27 HAROLD_Profiles_32.2
## 28 HAROLD_Profiles_33
## 29 HAROLD_Profiles_33.1
## 30 HAROLD_Profiles_33.2
## 31 HAROLD_Profiles_34
## 32 HAROLD_Profiles_34.1
## 33 HAROLD_Profiles_34.2
## 34 HAROLD_Profiles_35
## 35 HAROLD_Profiles_35.1
## 36 HAROLD_Profiles_35.2
## 37 HAROLD_Profiles_36
## 38 HAROLD_Profiles_36.1
## 39 HAROLD_Profiles_36.2
## 40 HAROLD_Profiles_37
## 41 HAROLD_Profiles_37.1
## 42 HAROLD_Profiles_37.2
## 43 HAROLD_Profiles_38
## 44 HAROLD_Profiles_38.1
## 45 HAROLD_Profiles_38.2
## 46 HAROLD_Profiles_39
## 47 HAROLD_Profiles_39.1
## 48 HAROLD_Profiles_39.2
## 49 HAROLD_Profiles_40
## 50 HAROLD_Profiles_40.1
## 51 HAROLD_Profiles_40.2
## 52 HAROLD_Profiles_41
## 53 HAROLD_Profiles_41.1
## 54 HAROLD_Profiles_41.2
## 55 HAROLD_Profiles_42
## 56 HAROLD_Profiles_42.1
## 57 HAROLD_Profiles_42.2
## 58 HAROLD_Profiles_43
## 59 HAROLD_Profiles_43.1
## 60 HAROLD_Profiles_43.2
## 61 HAROLD_Profiles_44
## 62 HAROLD_Profiles_44.1
## 63 HAROLD_Profiles_44.2
## 64 HAROLD_Profiles_45
## 65 HAROLD_Profiles_45.1
## 66 HAROLD_Profiles_45.2
## 67 HAROLD_Profiles_46
## 68 HAROLD_Profiles_46.1
## 69 HAROLD_Profiles_46.2
## 70 HAROLD_Profiles_47
## 71 HAROLD_Profiles_47.1
## 72 HAROLD_Profiles_47.2
## 73 HAROLD_Profiles_48
## 74 HAROLD_Profiles_48.1
## 75 HAROLD_Profiles_48.2
## 76 HAROLD_WindDirection_24
```

```
## 77     HAROLD_WindDirection_24.1
## 78     HAROLD_WindDirection_24.2
## 79     HAROLD_WindDirection_25
## 80     HAROLD_WindDirection_25.1
## 81     HAROLD_WindDirection_25.2
## 82     HAROLD_WindDirection_26
## 83     HAROLD_WindDirection_26.1
## 84     HAROLD_WindDirection_26.2
## 85     HAROLD_WindDirection_27
## 86     HAROLD_WindDirection_27.1
## 87     HAROLD_WindDirection_27.2
## 88     HAROLD_WindDirection_28
## 89     HAROLD_WindDirection_28.1
## 90     HAROLD_WindDirection_28.2
## 91     HAROLD_WindDirection_29
## 92     HAROLD_WindDirection_29.1
## 93     HAROLD_WindDirection_29.2
## 94     HAROLD_WindDirection_30
## 95     HAROLD_WindDirection_30.1
## 96     HAROLD_WindDirection_30.2
## 97     HAROLD_WindDirection_31
## 98     HAROLD_WindDirection_31.1
## 99     HAROLD_WindDirection_31.2
## 100     HAROLD_WindDirection_32
## 101     HAROLD_WindDirection_32.1
## 102     HAROLD_WindDirection_32.2
## 103     HAROLD_WindDirection_33
## 104     HAROLD_WindDirection_33.1
## 105     HAROLD_WindDirection_33.2
## 106     HAROLD_WindDirection_34
## 107     HAROLD_WindDirection_34.1
## 108     HAROLD_WindDirection_34.2
## 109     HAROLD_WindDirection_35
## 110     HAROLD_WindDirection_35.1
## 111     HAROLD_WindDirection_35.2
## 112     HAROLD_WindDirection_36
## 113     HAROLD_WindDirection_36.1
## 114     HAROLD_WindDirection_36.2
## 115     HAROLD_WindDirection_37
## 116     HAROLD_WindDirection_37.1
## 117     HAROLD_WindDirection_37.2
## 118     HAROLD_WindDirection_38
## 119     HAROLD_WindDirection_38.1
## 120     HAROLD_WindDirection_38.2
## 121     HAROLD_WindDirection_39
## 122     HAROLD_WindDirection_39.1
## 123     HAROLD_WindDirection_39.2
## 124     HAROLD_WindDirection_40
## 125     HAROLD_WindDirection_40.1
## 126     HAROLD_WindDirection_40.2
## 127     HAROLD_WindDirection_41
## 128     HAROLD_WindDirection_41.1
## 129     HAROLD_WindDirection_41.2
## 130     HAROLD_WindDirection_42
```

```
## 131      HAROLD_WindDirection_42.1
## 132      HAROLD_WindDirection_42.2
## 133      HAROLD_WindDirection_43
## 134      HAROLD_WindDirection_43.1
## 135      HAROLD_WindDirection_43.2
## 136      HAROLD_WindDirection_44
## 137      HAROLD_WindDirection_44.1
## 138      HAROLD_WindDirection_44.2
## 139      HAROLD_WindDirection_45
## 140      HAROLD_WindDirection_45.1
## 141      HAROLD_WindDirection_45.2
## 142      HAROLD_WindDirection_46
## 143      HAROLD_WindDirection_46.1
## 144      HAROLD_WindDirection_46.2
## 145      HAROLD_WindDirection_47
## 146      HAROLD_WindDirection_47.1
## 147      HAROLD_WindDirection_47.2
## 148      HAROLD_WindDirection_48
## 149      HAROLD_WindDirection_48.1
## 150      HAROLD_WindDirection_48.2
```

Computing point wise products

```
luganville.pt <- data.frame(lon = 167.1667 , lat = -15.5333)

ts.luganville <- stormBehaviour_pt(harold, points = luganville.pt)
```

Visualize products

```
plotBehaviour(harold, prod.harold[["HAROLD_MSW"]])
```

```
plotBehaviour(harold, prod.harold[["HAROLD_PDI"]])
```

```
plotBehaviour(harold, prod.harold[["HAROLD_Exposure_58"]])
```

##Save product

```
writeRast(prod.harold, path = paste0(tempdir(),"/"))
```