use extreme model

Comfort

8/17/2020

```
library(readxl)
library(extRemes)
```

```
## Loading required package: Lmoments
```

```
## Loading required package: distillery
```

```
##
## Attaching package: 'extRemes'
```

```
## The following objects are masked from 'package:stats':
##
##     qqnorm, qqplot
```

```
DODOMA_RAIN <- read_excel("DODOMA_RAIN.xlsx")
View(DODOMA_RAIN)
```

fitting a model assuming stationarity

```
attach(DODOMA_RAIN)
fit1 = fevd(max_rain, type="GEV", method = "MLE")
fit1
```

```
##
## fevd(x = max_rain, type = "GEV", method = "MLE")
##
## [1] "Estimation Method used: MLE"
##
##
##  Negative Log-Likelihood Value:  337.1421
##
##
##  Estimated parameters:
##    location       scale       shape
## 61.82464460 14.90230824 -0.03147776
##
##  Standard Error Estimates:
##  location     scale     shape
## 1.9392155 1.4298998 0.1004641
##
##  Estimated parameter covariance matrix.
##             location       scale       shape
## location  3.76055683  1.08875506 -0.08211196
```

```
## scale     1.08875506  2.04461335 -0.06009078
## shape    -0.08211196 -0.06009078  0.01009303
##
##  AIC = 680.2842
##
##  BIC = 687.3925
```

summary of the fitted model

```
summary.fevd(fit1)
```

```
##
## fevd(x = max_rain, type = "GEV", method = "MLE")
##
## [1] "Estimation Method used: MLE"
##
##
##  Negative Log-Likelihood Value:  337.1421
##
##
##  Estimated parameters:
##    location       scale       shape
## 61.82464460 14.90230824 -0.03147776
##
##  Standard Error Estimates:
##  location     scale     shape
## 1.9392155 1.4298998 0.1004641
##
##  Estimated parameter covariance matrix.
##             location       scale       shape
## location  3.76055683  1.08875506 -0.08211196
## scale     1.08875506  2.04461335 -0.06009078
## shape    -0.08211196 -0.06009078  0.01009303
##
##  AIC = 680.2842
##
##  BIC = 687.3925
```

Confidence interval for parameters

```
ci(fit1, type = "parameter")
```

```
## fevd(x = max_rain, type = "GEV", method = "MLE")
##
## [1] "Normal Approx."
##
##          95% lower CI    Estimate 95% upper CI
## location   58.0238520 61.82464460   65.6254372
## scale      12.0997562 14.90230824   17.7048603
## shape      -0.2283837 -0.03147776    0.1654282
```

```
distill(fit1, cov=TRUE, FUN="mean")
```

```
##          location             scale             shape              nllh
##       61.82464460       14.90230824       -0.03147776      337.14207708
## location.location    scale.location    shape.location    location.scale
##        3.76055683        1.08875506       -0.08211196        1.08875506
##       scale.scale       shape.scale    location.shape       scale.shape
##        2.04461335       -0.06009078       -0.08211196       -0.06009078
##       shape.shape
##        0.01009303
```

To confirm if the fitted model is stationary

```
is.fixedfevd(fit1)
```

```
## [1] TRUE
```

fitting a non-stationary model

```
attach(DODOMA_RAIN)
```

```
## The following objects are masked from DODOMA_RAIN (pos = 3):
##
##     max_rain, year
```

```
fit2 = fevd(max_rain, location.fun=~year,type="GEV", method = "MLE")
fit2
```

```
##
## fevd(x = max_rain, location.fun = ~year, type = "GEV", method = "MLE")
##
## [1] "Estimation Method used: MLE"
##
##
##  Negative Log-Likelihood Value:  336.8901
##
##
##  Estimated parameters:
##         mu0         mu1       scale       shape
## 25.68592787  0.01831569 15.01216832 -0.03507236
##
##  Standard Error Estimates:
##          mu0          mu1        scale        shape
## 195.44556694   0.09928921   1.57989882   0.12147849
##
##  Estimated parameter covariance matrix.
##               mu0           mu1         scale        shape
## mu0   38198.96964 -19.404680670 -112.90664014 12.769984143
## mu1     -19.40468   0.009858347    0.05791028 -0.006528674
## scale  -112.90664   0.057910276    2.49608029 -0.103292028
## shape    12.76998  -0.006528674   -0.10329203  0.014757022
##
##  AIC = 681.7801
```

```
##
##  BIC = 691.2579
```

The code below is important when we are fitting a non-stationary model

```
findpars(fit2)
```

```
## $location
##        1        2        3        4        5        6        7        8
## 61.12680 61.14511 61.16343 61.18174 61.20006 61.21837 61.23669 61.25501
##        9       10       11       12       13       14       15       16
## 61.27332 61.29164 61.30995 61.32827 61.34658 61.36490 61.38322 61.40153
##       17       18       19       20       21       22       23       24
## 61.41985 61.43816 61.45648 61.47479 61.49311 61.51143 61.52974 61.54806
##       25       26       27       28       29       30       31       32
## 61.56637 61.58469 61.60300 61.62132 61.63964 61.65795 61.67627 61.69458
##       33       34       35       36       37       38       39       40
## 61.71290 61.73121 61.74953 61.76785 61.78616 61.80448 61.82279 61.84111
##       41       42       43       44       45       46       47       48
## 61.85942 61.87774 61.89606 61.91437 61.93269 61.95100 61.96932 61.98763
##       49       50       51       52       53       54       55       56
## 62.00595 62.02427 62.04258 62.06090 62.07921 62.09753 62.11584 62.13416
##       57       58       59       60       61       62       63       64
## 62.15248 62.17079 62.18911 62.20742 62.22574 62.24405 62.26237 62.28069
##       65       66       67       68       69       70       71       72
## 62.29900 62.31732 62.33563 62.35395 62.37226 62.39058 62.40889 62.42721
##       73       74       75       76       77       78       79
## 62.44553 62.46384 62.48216 62.50047 62.51879 62.53710 62.55542
##
## $scale
##  [1] 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217
##  [9] 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217
## [17] 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217
## [25] 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217
## [33] 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217
## [41] 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217
## [49] 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217
## [57] 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217
## [65] 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217
## [73] 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217 15.01217
##
## $shape
##  [1] -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236
##  [7] -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236
## [13] -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236
## [19] -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236
## [25] -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236
## [31] -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236
## [37] -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236
## [43] -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236
## [49] -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236
## [55] -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236
## [61] -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236
```

```
## [67] -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236
## [73] -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236 -0.03507236
## [79] -0.03507236
```

Covariate matrix for non-stationary EVD projections

```
make.qcov(fit2)
```

```
##      mu0 mu1 scale shape threshold
## [1,]   1   0     1     1         0
```

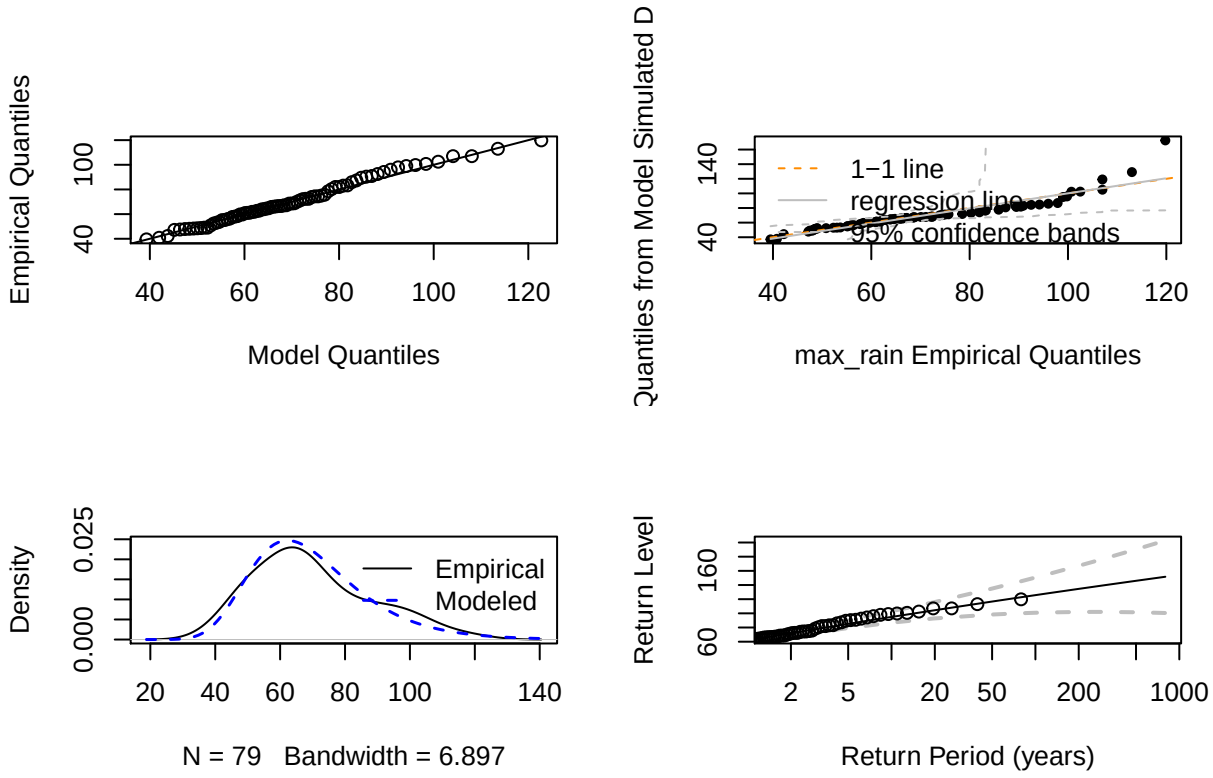
Likelihood Ratio test

```
lr.test(fit1,fit2, alpha = 0.05)
```

```
##
##  Likelihood-ratio Test
##
## data:  max_rainmax_rain
## Likelihood-ratio = 0.50402, chi-square critical value = 3.8415, alpha =
## 0.0500, Degrees of Freedom = 1.0000, p-value = 0.4777
## alternative hypothesis: greater
```

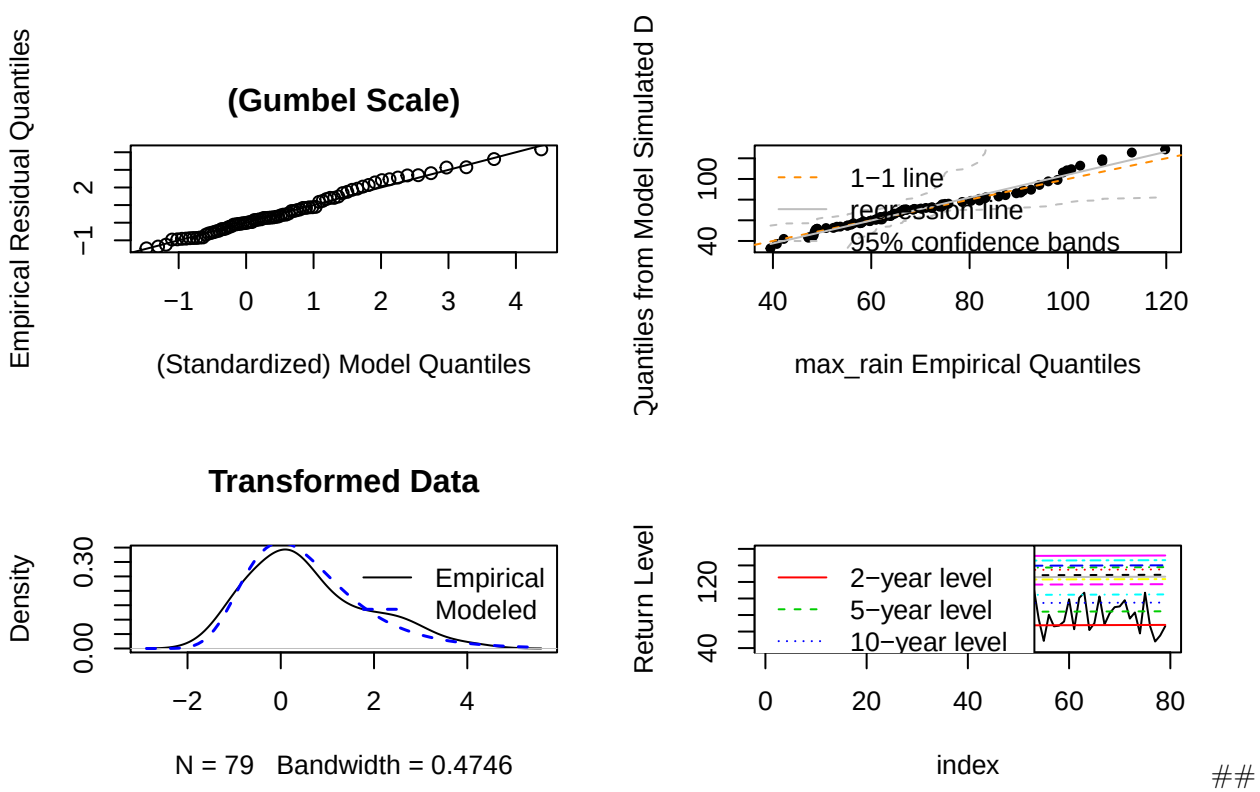
Diagnostic plot for a model assuming stationarity

```
plot(fit1)
```



Diagnostic plot for non-stationary model

```
plot(fit2)
```



##

Return level estimate for a model assuming stationarity

```
return.level(fit1,return.period = c(10,20,50,100,200))
```

```
## fevd(x = max_rain, type = "GEV", method = "MLE")
## get(paste("return.level.fevd.", newcl, sep = ""))(x = x, return.period = return.period)
##
##  GEV model fitted to  max_rain
## Data are assumed to be  stationary
## [1] "Return Levels for period units in years"
##  10-year level  20-year level  50-year level 100-year level 200-year level
##       94.20009      104.08124      116.54337      125.64532      134.51727
```

```
ci(fit1, alpha=0.05,return.period =10 ,qcov=NULL)
```

```
## fevd(x = max_rain, type = "GEV", method = "MLE")
##
## [1] "Normal Approx."
##
## [1] "10-year return level: 94.2"
##
## [1] "95% Confidence Interval: (86.3823, 102.0179)"
```

Return level estimate for non-stationary model

```
return.level(fit2,return.period = c(10,20,50,100,200))
```

```
## fevd(x = max_rain, location.fun = ~year, type = "GEV", method = "MLE")
## get(paste("return.level.fevd.", newcl, sep = ""))(x = x, return.period = return.period)
```

```
##
##  GEV model fitted to  max_rain
## Data are assumed to be  non-stationary
## [1] "Return Levels for period units in years"
##       10-year level 20-year level 50-year level 100-year level 200-year level
##  [1,]      93.61091      103.4720      115.8720       124.9020       133.6819
##  [2,]      93.62923      103.4903      115.8903       124.9204       133.7002
##  [3,]      93.64754      103.5086      115.9086       124.9387       133.7185
##  [4,]      93.66586      103.5269      115.9269       124.9570       133.7368
##  [5,]      93.68418      103.5453      115.9453       124.9753       133.7551
##  [6,]      93.70249      103.5636      115.9636       124.9936       133.7735
##  [7,]      93.72081      103.5819      115.9819       125.0119       133.7918
##  [8,]      93.73912      103.6002      116.0002       125.0303       133.8101
##  [9,]      93.75744      103.6185      116.0185       125.0486       133.8284
## [10,]      93.77575      103.6368      116.0368       125.0669       133.8467
## [11,]      93.79407      103.6552      116.0552       125.0852       133.8650
## [12,]      93.81239      103.6735      116.0735       125.1035       133.8834
## [13,]      93.83070      103.6918      116.0918       125.1218       133.9017
## [14,]      93.84902      103.7101      116.1101       125.1401       133.9200
## [15,]      93.86733      103.7284      116.1284       125.1585       133.9383
## [16,]      93.88565      103.7467      116.1467       125.1768       133.9566
## [17,]      93.90396      103.7650      116.1650       125.1951       133.9749
## [18,]      93.92228      103.7834      116.1834       125.2134       133.9933
## [19,]      93.94060      103.8017      116.2017       125.2317       134.0116
## [20,]      93.95891      103.8200      116.2200       125.2500       134.0299
## [21,]      93.97723      103.8383      116.2383       125.2684       134.0482
## [22,]      93.99554      103.8566      116.2566       125.2867       134.0665
## [23,]      94.01386      103.8749      116.2749       125.3050       134.0848
## [24,]      94.03217      103.8933      116.2933       125.3233       134.1031
## [25,]      94.05049      103.9116      116.3116       125.3416       134.1215
## [26,]      94.06881      103.9299      116.3299       125.3599       134.1398
## [27,]      94.08712      103.9482      116.3482       125.3783       134.1581
## [28,]      94.10544      103.9665      116.3665       125.3966       134.1764
## [29,]      94.12375      103.9848      116.3848       125.4149       134.1947
## [30,]      94.14207      104.0031      116.4032       125.4332       134.2130
## [31,]      94.16038      104.0215      116.4215       125.4515       134.2314
## [32,]      94.17870      104.0398      116.4398       125.4698       134.2497
## [33,]      94.19702      104.0581      116.4581       125.4881       134.2680
## [34,]      94.21533      104.0764      116.4764       125.5065       134.2863
## [35,]      94.23365      104.0947      116.4947       125.5248       134.3046
## [36,]      94.25196      104.1130      116.5130       125.5431       134.3229
## [37,]      94.27028      104.1314      116.5314       125.5614       134.3413
## [38,]      94.28859      104.1497      116.5497       125.5797       134.3596
## [39,]      94.30691      104.1680      116.5680       125.5980       134.3779
## [40,]      94.32523      104.1863      116.5863       125.6164       134.3962
## [41,]      94.34354      104.2046      116.6046       125.6347       134.4145
## [42,]      94.36186      104.2229      116.6229       125.6530       134.4328
## [43,]      94.38017      104.2413      116.6413       125.6713       134.4511
## [44,]      94.39849      104.2596      116.6596       125.6896       134.4695
## [45,]      94.41680      104.2779      116.6779       125.7079       134.4878
## [46,]      94.43512      104.2962      116.6962       125.7263       134.5061
## [47,]      94.45343      104.3145      116.7145       125.7446       134.5244
## [48,]      94.47175      104.3328      116.7328       125.7629       134.5427
## [49,]      94.49007      104.3511      116.7512       125.7812       134.5610
```

```
## [50,]      94.50838      104.3695      116.7695      125.7995      134.5794
## [51,]      94.52670      104.3878      116.7878      125.8178      134.5977
## [52,]      94.54501      104.4061      116.8061      125.8361      134.6160
## [53,]      94.56333      104.4244      116.8244      125.8545      134.6343
## [54,]      94.58164      104.4427      116.8427      125.8728      134.6526
## [55,]      94.59996      104.4610      116.8610      125.8911      134.6709
## [56,]      94.61828      104.4794      116.8794      125.9094      134.6893
## [57,]      94.63659      104.4977      116.8977      125.9277      134.7076
## [58,]      94.65491      104.5160      116.9160      125.9460      134.7259
## [59,]      94.67322      104.5343      116.9343      125.9644      134.7442
## [60,]      94.69154      104.5526      116.9526      125.9827      134.7625
## [61,]      94.70985      104.5709      116.9709      126.0010      134.7808
## [62,]      94.72817      104.5893      116.9893      126.0193      134.7991
## [63,]      94.74649      104.6076      117.0076      126.0376      134.8175
## [64,]      94.76480      104.6259      117.0259      126.0559      134.8358
## [65,]      94.78312      104.6442      117.0442      126.0742      134.8541
## [66,]      94.80143      104.6625      117.0625      126.0926      134.8724
## [67,]      94.81975      104.6808      117.0808      126.1109      134.8907
## [68,]      94.83806      104.6991      117.0992      126.1292      134.9090
## [69,]      94.85638      104.7175      117.1175      126.1475      134.9274
## [70,]      94.87470      104.7358      117.1358      126.1658      134.9457
## [71,]      94.89301      104.7541      117.1541      126.1841      134.9640
## [72,]      94.91133      104.7724      117.1724      126.2025      134.9823
## [73,]      94.92964      104.7907      117.1907      126.2208      135.0006
## [74,]      94.94796      104.8090      117.2090      126.2391      135.0189
## [75,]      94.96627      104.8274      117.2274      126.2574      135.0372
## [76,]      94.98459      104.8457      117.2457      126.2757      135.0556
## [77,]      95.00291      104.8640      117.2640      126.2940      135.0739
## [78,]      95.02122      104.8823      117.2823      126.3124      135.0922
## [79,]      95.03954      104.9006      117.3006      126.3307      135.1105
```