Replications of Brandsma 2016

JBvR

30 juli 2019

Intro

Replicate findings from Brandsma - De Bilt, 2016, Technical report; TR-356 - the homogenization of Dutch temperature records. The comments will be brief. The main purpoe is to have insight in and aa replication of the methods. We will not exactly replicate tables and figures.

Sources

On the KNMI site we found the unhomogenized and homogenized data from stations

1. Den Helder/De Kooy,
2. De Bilt,
3. Groningen/Eelde,
4. Vlissingen, and
5. Maastricht/Beek.

(235, 260, 280, 310 and 380). We did not yet find th parallel data so we cannot yet replicate the parallel part of the report.

Percentiles phase

When we have a sample without any knowledge of the distribution of the population, the only available tools come from the area of non-parametric statistics. That means we will start with the so-called empirical cumulative distribution function (e-cdf) for the observations $x_i, i = 1, ..., N$

$$F(x) := |x_i <= x|)/(N + 1)$$

where $|v|$ is the length of the vector v..

$F(x)$ is a stepfunction, so neither differentiable nor invertable.We have to approximate them, f.i. using interpolation. The *quantile* function in R does just that. Given all observations, it fixes $F(min(x)) := 0$ and $F(max(x)) := 1$ interpolates all other values.

The R quantile function has 9 different methods availble. The first 3 relate to count data, the other 6 to continuous data. The classical commercial statistical packages such as SAS, SPSS, Stata and S-(plus) all had slightly different algorithms, so R, the Open Source succesor to S-(plus) has all these methods and more implemented. The default is the same as S.

Which one is the best? In the theory of non-parametric statistics they are equally valid because they have the same asymtotic behaviour. But otherwise it is completely undecidable. Simulation is not applicable, because we do not know how to daw a random sample from any unknown distribution. Hyndman, the author of the above R-function has added his owen favorite ("type 8") which according to him has some advantges.

Reading and processing data

```
inv_F <- function(s) quantile(s, seq(0.05, 0.95, by=0.05))

pretty_inv_F <- function(aa, ...) {
  pretty_plot(data.frame(seq(0.05, 0.95, by=0.05), inv_F(aa)), ...)
  }

get_etm <- function(fname){
    a<- readLines(fname)
    b<- read.csv(text=a, skip=49,
                 header=FALSE, stringsAsFactors = FALSE)
    names(b)<-  gsub("\\s", "", strsplit(a[48], ",")[[1]])
    return(b)
}

ref_etm<- function(etm){
  # need yy mm dd TN, TX, TG
  etm$yy <- as.integer(substring(etm$YYYYMMDD,1,4))
  etm$mm <- as.integer(substring(etm$YYYYMMDD,5,6))
  etm$dd <- as.integer(substring(etm$YYYYMMDD,7,8))

  return(etm[, c("yy", "mm", "dd", "TX", "TN", "TG")])
}


get_pre <- function(fname){
    a<- readLines(fname)
    b<- read.csv(text=a, skip=13,
                 header=TRUE, stringsAsFactors = FALSE)
    return(b[, -ncol(b)])
}

ref_pre<- function(etm){
  # need yy mm dd TN, TX, TG
  etm$yy <- as.integer(substring(etm$YYYYMMDD,1,4))
  etm$mm <- as.integer(substring(etm$YYYYMMDD,5,6))
  etm$dd <- as.integer(substring(etm$YYYYMMDD,7,8))

  return(etm[, c("yy", "mm", "dd", "TX", "TN", "TG")])
}


DeBilt_h <- ref_etm(get_etm("./input/etmgeg_260.txt"))
DeBilt_u <- ref_pre(get_pre("./input/temp_260.txt"))
DeBilt_u <- rbind(DeBilt_u, DeBilt_h[-(1:24350),])
```

Lets do some monhtly aggregation, then look for a changepoint,

```
monthly_DeBilt_u <- aggregate(DeBilt_u[, c( "TX", "TN", "TG")],
  by = list(DeBilt_u$yy, DeBilt_u$mm ),
                             FUN=mean)
names(monthly_DeBilt_u) <- c("yy", "mm", "TX", "TN", "TG")
```

Do change point(s) analysis, first de-season it.

```r
b<- complete.cases(DeBilt_u)
m<- aggregate(DeBilt_u[b, c("TN", "TG", "TX")], by=list(DeBilt_u$mm[b]), FUN=mean)

DeBilt_u_des <- DeBilt_u
DeBilt_u_des$TN <- DeBilt_u_des$TN -
  rep(m$TN, 150*365)[1: nrow(DeBilt_u)]
DeBilt_u_des$TG <- DeBilt_u_des$TG -
  rep(m$TG, 150*365)[1: nrow(DeBilt_u)]
DeBilt_u_des$TX <- DeBilt_u_des$TX -
  rep(m$TX, 150*365)[1: nrow(DeBilt_u)]

monthly_DeBilt_u_des <- aggregate(DeBilt_u_des[, c( "TX", "TN", "TG")],
  by = list(DeBilt_u_des$yy, DeBilt_u_des$mm ),
                             FUN=mean)
names(monthly_DeBilt_u_des) <- c("yy", "mm", "TX", "TN", "TG")


monthly_DeBilt_u_des <- monthly_DeBilt_u_des[
  order(monthly_DeBilt_u_des$yy, monthly_DeBilt_u_des$mm),]
```

Analyse those TSeries

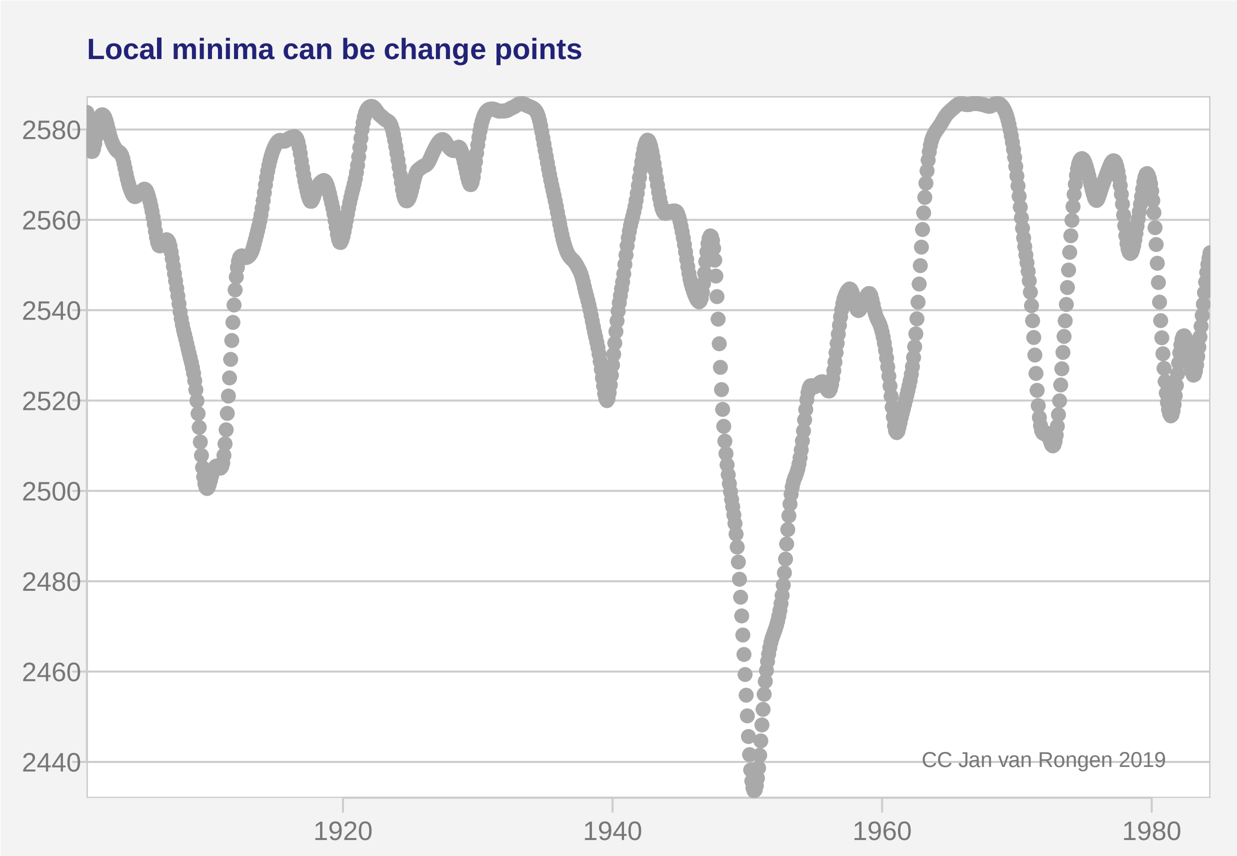
```r
work1<- ts(monthly_DeBilt_u_des$TX, start=1901, freq=12)/10
b<- stl(work1, s.window= "periodic")
work1 <- b$time.series[,2]
ns_points= 2
range <- 0:1000
require(splines)
```

```
## Loading required package: splines
```

```r
result<- sapply(range, function(N)
                {cpt <- c(rep(0, N),
                          rep(1,length(work1)-N))
                t<- time(work1)
                l<- lm(work1 ~ ns(t, ns_points)+ cpt)
                return(AIC(l))
})

dd<- range[which(result== min(result))]


pretty_plot(ts(result, start=1901, freq=12), type="p", kleur=4,
            cex=0.9, main="Local minima can be change points")
```

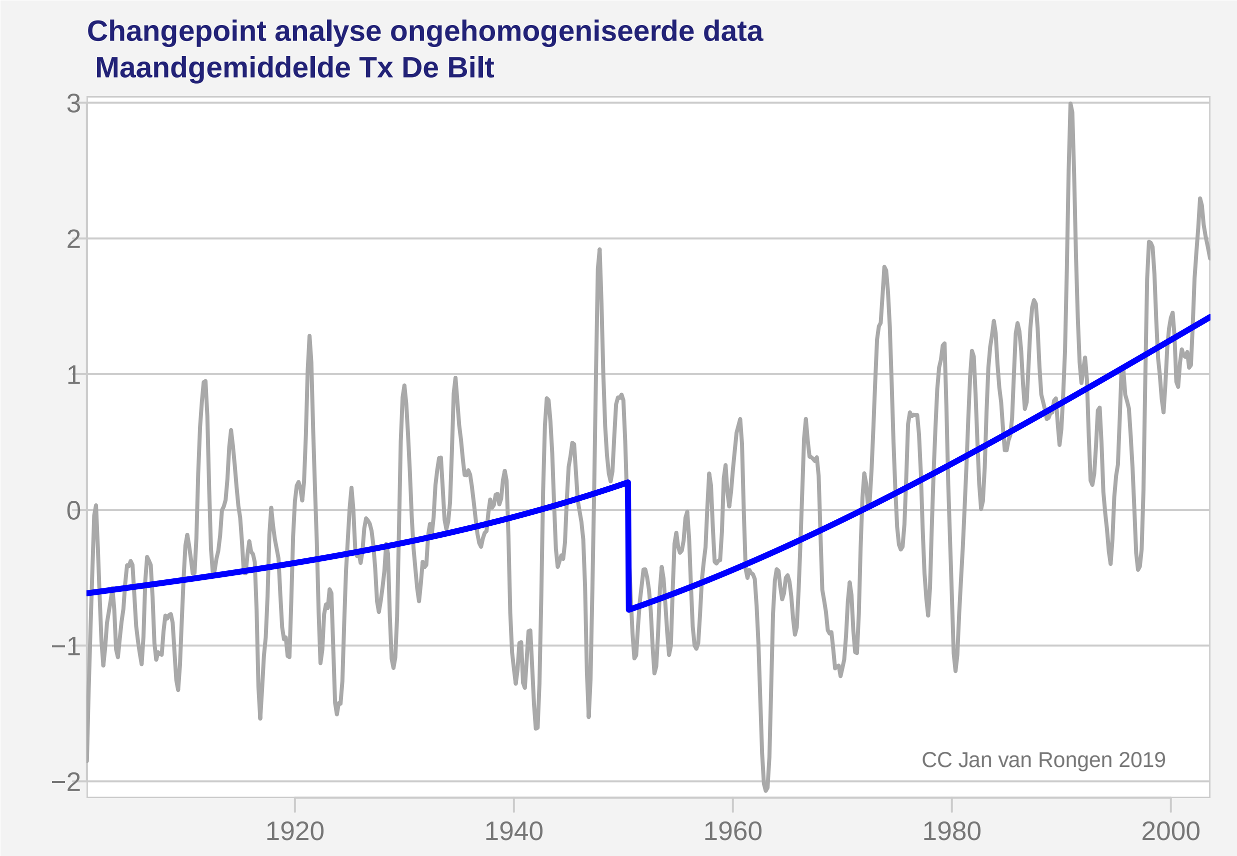



```
pretty_plot(work1, kleur=4, lwd=2, main="Changepoint analyse ongehomogeniseerde data\n Maandgemiddelde '

l<- lm(work1 ~ time(work1))
N = dd
cpt <- c(rep(0, N), rep(1,length(work1)-N))

t<- time(work1)
l<-lm(work1 ~ ns(t, ns_points)+ cpt )

pretty_plot(add=TRUE, work1-l$res, kleur=2, lwd=3)
```




```r
summary(l)
```

```
##
## Call:
## lm(formula = work1 ~ ns(t, ns_points) + cpt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7336 -0.3883  0.0280  0.4173  2.1694
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.61474    0.05630  -10.92   <2e-16 ***
## ns(t, ns_points)1   2.55113    0.18102   14.09   <2e-16 ***
## ns(t, ns_points)2   2.72308    0.10424   26.12   <2e-16 ***
## cpt                -0.94082    0.07404  -12.71   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6481 on 1228 degrees of freedom
## Multiple R-squared:  0.4213, Adjusted R-squared:  0.4199
## F-statistic:   298 on 3 and 1228 DF,  p-value: < 2.2e-16
```

```r
cat("\n*** Jump", dd, "Of size", l$fit[1+dd]- l$fit[dd])
```

```
##
## *** Jump 594 Of size -0.9385077
```

So this rather unsophisticated analysis of monthly means finds a change point in 1951 for TX, TG and TN - independent of each other. The size of the downward step is 0.96 for TX, 0.46 for TG and 0.25 for TN. It is very remarkable that the method actually show that the change points are very likely located in sept. 1951

```
b<- stl(ts(DeBilt_u$TX, start=1901, freq=365.25) , s.window = "periodic")

# pretty_plot(b$time.series[,2])

work1 <- b$time.series[,2]
ns_points= 2
range <- (0:1000)*30
require(splines)
result<- sapply(range, function(N)
                {cpt <- c(rep(0, N),
                          rep(1,length(work1)-N))
                t<- time(work1)
                l<- lm(work1 ~ ns(t, ns_points)+ cpt)
                return(AIC(l))
})

dd<- range[which(result== min(result))]
zoo::as.yearmon(time(work1))[dd]
```

```
## [1] "jul 1950"
```

```
pretty_plot(ts(result, start=1901, freq=365.25/30), type="p", kleur=4,
            cex=0.9, main="Local minima can be change points")
```