TER *2018/2019*

Étude de données atmosphériques extrêmes

Projet tutoré :

Mathis CORDIER

Amandine PEILLON

Sommaire

$$\hat{q}_n(\alpha \mid x)$$

Nos estimateurs

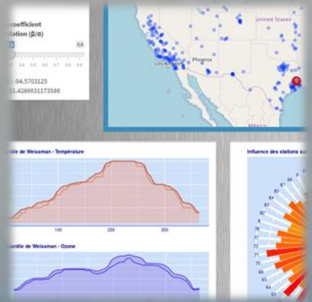
Introduction à la théorie

Présentation des estimateurs

Code R

Méthode

Application en R



Application

ShinyApps

Application interactive

$$\hat{\bar{F}}_n(y \mid x) := \sum_{i=1}^{n} K_h(x - X_i)\mathbb{1}\{Y_i > y\} \Big/ \sum_{i=1}^{n} K_h(x - X_i)$$

Estimateur de $P(Y > y \mid X = x)$

$$\hat{q}_n(\alpha \mid x) := \hat{\bar{F}}_n^{\leftarrow}(\alpha \mid x) = inf\left\{t, \hat{\bar{F}}_n(t \mid x) \leq \alpha\right\}$$

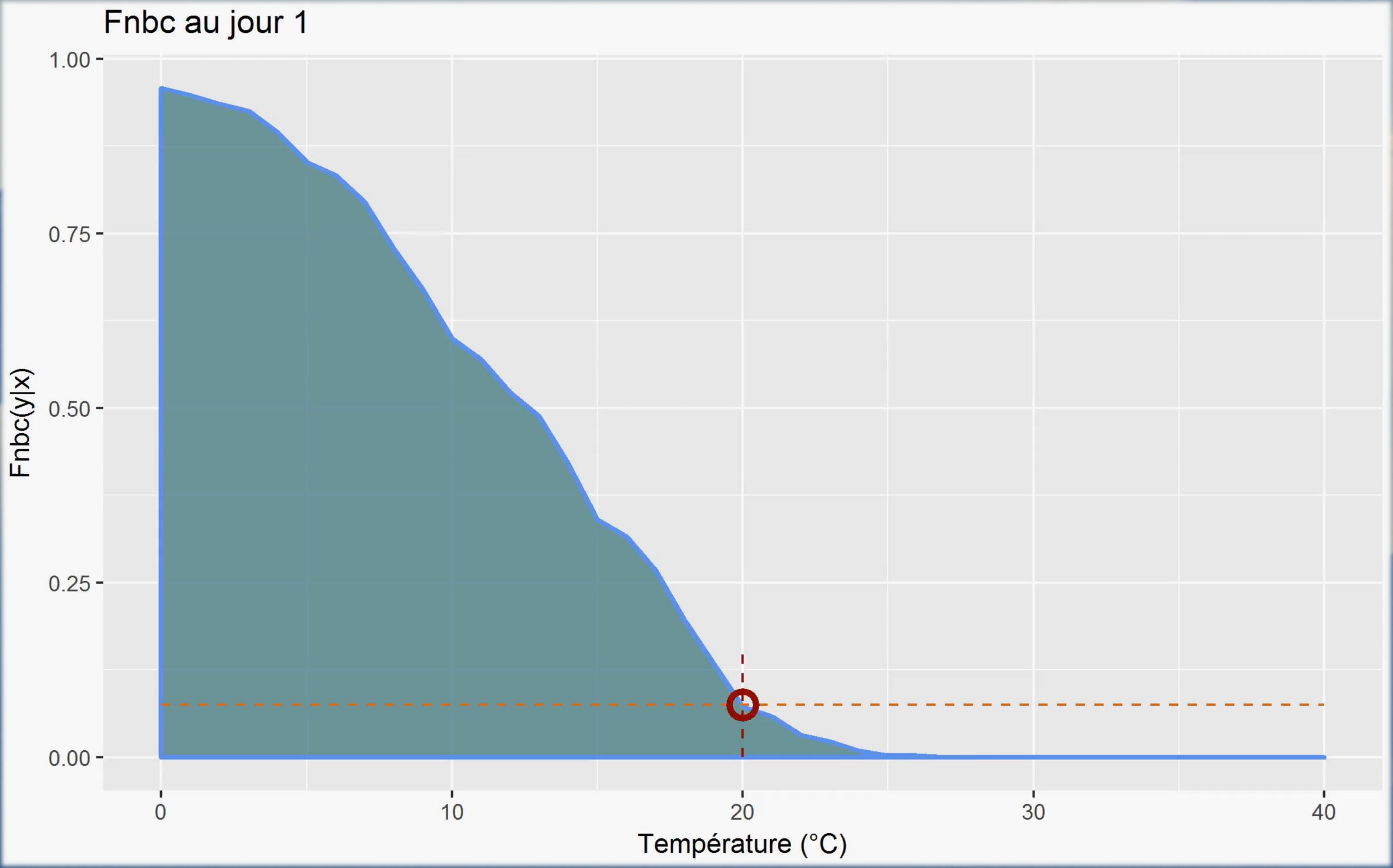
Estimateur du quantile classique

$$\hat{\gamma}_n^H(x) = \sum_{j=1}^{J} [\log \hat{q}_n(\tau_j \alpha_n \mid x) - \log \hat{q}_n(\alpha_n \mid x)] \Big/ \sum_{j=1}^{J} \log(1/\tau_j)$$

Estimateur de γ

$$\hat{q}_n^W(\beta_n \mid x) = \hat{q}_n(\alpha_n \mid x)(\alpha_n/\beta_n)^{\hat{\gamma}_n(x)}$$

Estimateur du quantile des valeurs extrêmes


Fnbc au jour 1
Fnbc(y|x)
1.00
0.75
0.50
0.25
0.00
0
10
20
30
40
Température (°C)

Optimisation pour le calcul de plusieurs quantiles

```r
normalize = function(df){
  df2 = df
  for (i in 3:5){
    df2[,i] = (df2[,i]-min(df2[,i])) / (max(df2[,i])-min(df2[,i]))
  }
  return(df2)
}

normalize_val = function(x,i){
  return((x-min(df_base[,i])) / (max(df_base[,i])-min(df_base[,i])) )
}

K = function(x){
  return(15/16*(1-x^2)^2)
}

Kh = function(x){
  tmp = x/h
  tmp[abs(tmp)>1]=1
  tmp = K(tmp)
  return(tmp/h)
}
```

```r
indic_oz = function(N,champ){
  Y = matrix(rep(df$oz,length(champ)),ncol=length(champ))
  tmp = t(matrix(rep(champ,N),ncol=N))
  return(Y>tmp)
}

indic_tp = function(N,champ){
  Y = matrix(rep(df$tp,length(champ)),ncol=length(champ))
  tmp = t(matrix(rep(champ,N),ncol=N))
  return(Y>tmp)
}
```

```r
# Réglages
df = df[df$lat<62,] # Suppression de l'Alaska
df$tp = (df$tp-32)*5/9 # Passage en Celsius
df_base = df
df = normalize(df)


# Paramètres
h = 0.12
N = dim(df)[1]
alpha = 0.001 # ~= 11 * log(N)/N
Npt = 60
Cpt = as.data.frame(table(df$tp))
Cpt$Freq = Cpt$Freq/N
Cpt$Var1 = as.numeric(as.character(Cpt$Var1))
Cpt = Cpt[(Cpt$Freq>0.001 & Cpt$Var1>=10 & Cpt$Var1<50),]
champ_tp = c(Cpt$Var1,seq(44,70,4))
champ_oz = 40:160/1000
J = 1:Npt/Npt
D = df[,3:5]
Mat = Kh(t(matrix(rep(J,N),ncol=N))-D$day)

# Calcul de l'indicatrice dès le début (ne dépend pas de lat et lng)
ind_tp = indic_tp(N,champ_tp)
ind_oz = indic_oz(N,champ_oz)


DF = as.data.frame(dplyr::summarise(dplyr::group_by(df_base,lat), lng=mean(lng)))
DF = cbind(rownames(DF),DF)
colnames(DF)[1]="station"
DF$station = paste("Station",DF$station)
```

```r
qnc = function(alpha,Fn,champ){
  return(champ[min(which(Fn<=alpha))])
}

gam_ncH = function(champ,Fn){
  tau = 1:9
  Sdiv = sum(log(tau))
  Snum = c()
  for (t in alpha/tau){
    Snum = c(Snum,qnc(t,Fn,champ))
  }
  return(sum(log(Snum)-log(Snum[1]))/Sdiv)
}

qnW = function(Fn,beta,champ){
  res = qnc(alpha,Fn,champ)*(alpha/beta)^gam_ncH(champ,Fn)
  return(res)
}
```

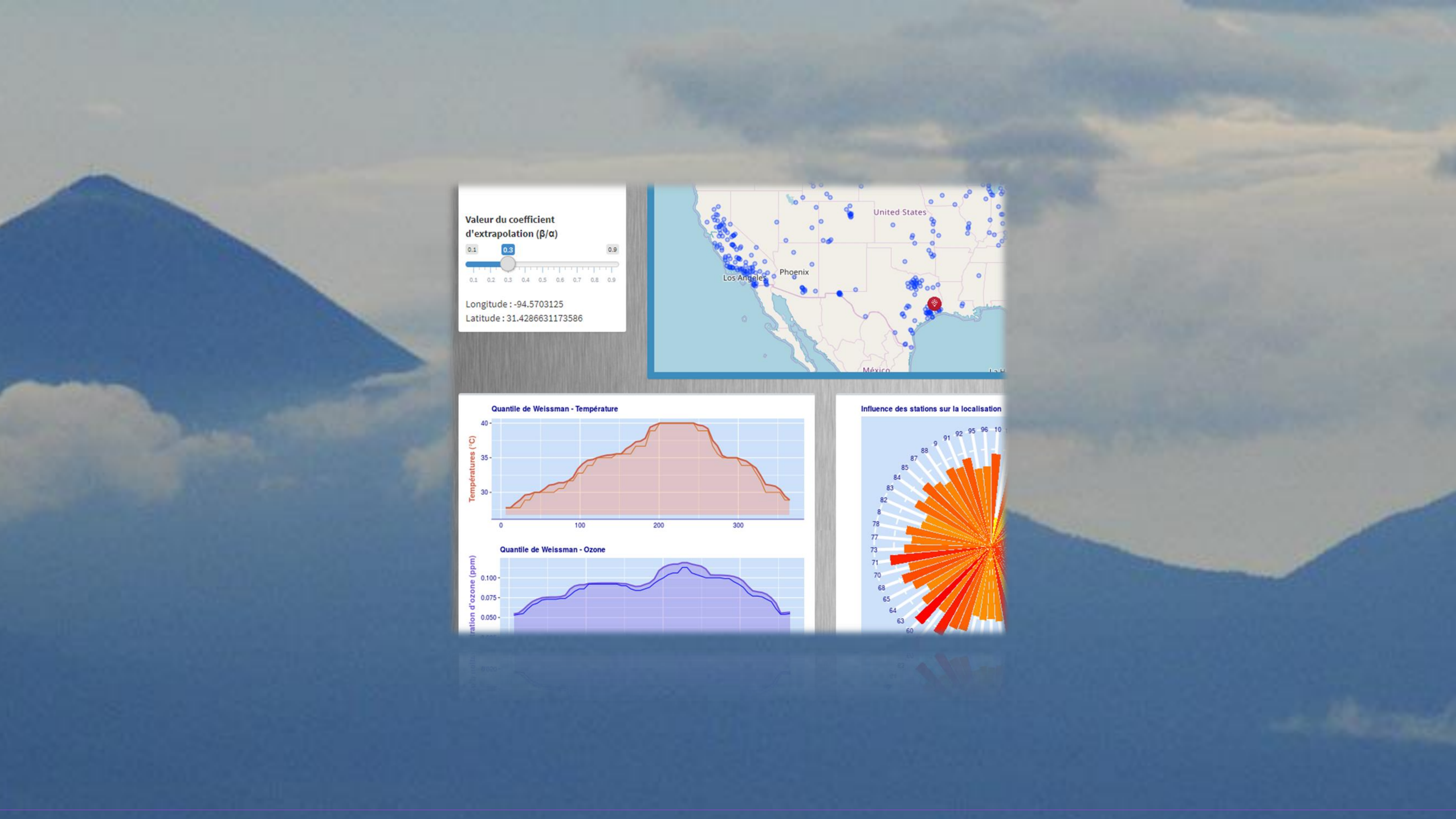
```
beta = l*alpha
latn = normalize_val(lieu[2],3)
lngn = normalize_val(lieu[1],4)
v = Kh(latn-D$lat)*Kh(lngn-D$lng)

M = t(v*Mat)
SK = apply(M,1,sum)


# Températures
Ki_tp = M%*%ind_tp
Fn_tp = Ki_tp/SK
QtpW = apply(Fn_tp,1,qnW,beta=beta,champ=champ_tp)
QtpC = apply(Fn_tp,1,qnc,alpha=alpha,champ=champ_tp)
q1 = min(QtpW,QtpC)


# Ozone
Ki_oz = M%*%ind_oz
Fn_oz = Ki_oz/SK
QozW = apply(Fn_oz,1,qnW,beta=beta,champ=champ_oz)
QozC = apply(Fn_oz,1,qnc,alpha=alpha,champ=champ_oz)
```

Application


Valeur du coefficient d'extrapolation (β/α)
0.1
0.3
0.9
Longitude : -94.5703125
Latitude : 31.4286631173586
United States
Phoenix
Los Angeles
México
Quantile de Weissman - Température
Températures (°C)
Quantile de Weissman - Ozone
Influence des stations sur la localisation