Olivier CÔME
Gueladio NIASSE
N° étudiant Olivier CÔME: 22110708
N° étudiant Gueladio NIASSE: 21714307
olivier.come@etu.umontpellier.fr
gueladio.niasse@etu.umontpellier.fr
Master 2
Statistique et Science des Données
Université de Montpellier

Projet: Statistique des évènements extêmes et applications

HAX005X : Valeurs extrêmes
Rédigé le 26 Février 2023 en LaTeX

Sommaire

1 Introduction **2**

2 Présentation des données **3**

3 Partie Univariée **4**
3.1 Approche GEV 4
3.1.1 Niveaux de retour associés aux périodes de retour 100, 500 et 1000 6
3.2 Approche GPD 6
3.2.1 Niveaux de retour associés aux périodes de retour 100, 500 et 1000 7

4 Partie bivariée **8**
4.1 Analyse entre les stations 2 et 9 8
4.2 Analyse entre les stations 2 et 7 10
4.3 Analyse entre les stations 9 et 4 12

5 Conclusion **14**

6 Annexes **15**

7 Bibliographie **19**

1 Introduction

Dans le cadre de ce projet, nous allons modéliser le comportement extrême des vagues dans le golfe du lion à l'aide des méthodes vue à travers l'unité d'enseignement HAX005X "valeurs extrêmes". D'après la source [1], le golf s'étale sur 220 kilomètres de la Camargue à la frontière espagnole. La côte, essentiellement sableuse, a été façonnée par la houle (la mer gagnant souvent les terres par élévation du niveau marin) et l'érosion côtière. L'apport de sédiments en provenance des fleuves a également permis de faire avancer le rivage pendant de longues périodes. Des formations de lagunes "comme les graus" (parfois temporaires) ont pu apparaitre et ont permis de faire communiquer les étangs littoraux avec la mer. Le golf du Lion est donc un milieu naturellement dynamique. C'est dans ce contexte que nous étudions le comportement extrême des vagues à cet endroit, de façon univariée dans un premier temps puis de façon bivariée dans un second temps.

2 Présentation des données

Les données que nous avons à notre disposition pour ce projet sont les suivantes :

— **DonneesStations**

— **DonneesVagues**

Pour réaliser les analyses, nous utiliserons essentiellement les données en provenance du dataframe *DonneesVagues* qui correspondent à des enregistrements de hauteurs de vagues significatives horaires de 20 stations situées dans le golf du Lion. Ces mesures horaires ont été enregistrées de 1961 à 2012. Ce dataframe est constitué de 464280 observations (en lignes) et de 21 variables (en colonnes). Décrivons un peu plus en détail les colonnes de ce dataframe *DonneesVagues* :

— **date** nous renseigne sur la date (format année mois jours) et l'heure précise (format heure minute seconde) à laquelle a été enregistrée la mesure.

— **station 1 à 20** nous renseigne sur les hauteurs de vagues mesurées

Le deuxième dataframe *DonneesStations*, quant à lui nous renseigne sur les coordonnées géographiques des 20 stations. Il est composé de 20 observations (lignes) et de 5 variables (colonnes). Décrivons un peu plus en détail les colonnes de ce dataframe.

— **lon :** floatant correspondant à la longitude de la station

— **lat :** floatant correspondant à la latitude de la station

— **depth :** floatant correspondant à la profondeur de la station

— **stationName :** Chaine de caractère indiquant le nom de la station

Les figures 1 et 2 sont des captures d'écran d'une portion des dataframe *DonneesVagues* et *DonneesStations*. La figure 3 correpond au nuage de points des données du dataframe *DonneesVagues*.

	date	station1	station2	station3	station4	station5	station6	station7	station8
1	1961-01-01 02:00:00	0.5120391	0.13847668	0.1775285	0.1967454	0.3011920	0.5258080	0.2734331	0.3345910
2	1961-01-01 03:00:00	0.5465476	0.08934399	0.1447777	0.1692832	0.3607951	0.5600442	0.2940980	0.3694929
3	1961-01-01 04:00:00	0.5894736	0.06477053	0.1429445	0.1745569	0.4378489	0.6025565	0.3268866	0.4230243
4	1961-01-01 05:00:00	0.6452147	0.05305877	0.1727725	0.2093817	0.5270881	0.6563302	0.3789678	0.4797138
5	1961-01-01 06:00:00	0.7143217	0.04773302	0.2210754	0.2612950	0.6135621	0.7224682	0.4465642	0.5400973
6	1961-01-01 07:00:00	0.7871489	0.04815382	0.2775594	0.3227399	0.6846716	0.7935590	0.5217415	0.6003126
7	1961-01-01 08:00:00	0.8617782	0.05834556	0.3420269	0.3958148	0.7518139	0.8599985	0.5987842	0.6319415
8	1961-01-01 09:00:00	0.9462464	0.08159746	0.4178586	0.4796206	0.8359310	0.9354858	0.6765540	0.6421875
9	1961-01-01 10:00:00	1.0216480	0.11668154	0.4989500	0.5657550	0.9561845	1.0071346	0.7399866	0.6414675
10	1961-01-01 11:00:00	1.0811893	0.16052766	0.5806396	0.6535683	1.1047519	1.0663446	0.7936877	0.6346608
11	1961-01-01 12:00:00	1.1294807	0.21032791	0.6651449	0.7451116	1.2539893	1.1148334	0.8430105	0.6261106
12	1961-01-01 13:00:00	1.1729137	0.26388204	0.7553571	0.8425921	1.4017688	1.1577642	0.8891504	0.6206052
13	1961-01-01 14:00:00	1.2247154	0.32077914	0.8476549	0.9395052	1.4871451	1.2081485	0.9299158	0.6271747
14	1961-01-01 15:00:00	1.2902025	0.38350457	0.9268072	1.0191746	1.5202062	1.2706363	0.9638701	0.6566994

FIGURE 1 – Extrait du dataframe *DonneesVagues*

	indexStation	lon	lat	depth	stationName
1	1	4.6400	42.0600	2326.11743	Lion
2	3	4.1400	43.4600	8.91505	EspigobsLa
3	4	6.7070	43.2960	146.92639	StMaximeLa
4	6	6.7400	43.3000	221.32100	StMaximeGr
5	7	7.8000	43.4000	2324.22217	61001
6	8	4.7000	42.1000	2337.85327	61002
7	9	6.2070	42.9333	622.45752	61004
8	10	9.2810	43.0680	368.80164	61005
9	11	9.0833	43.8500	1119.25525	61010
10	12	7.2290	43.6349	201.81612	61187
11	13	3.1683	42.4883	18.77695	61188
12	14	3.7785	43.3710	29.17235	61190
13	15	3.1250	42.9167	39.00000	61191
14	16	3.6500	41.9170	1222.56653	61196

FIGURE 2 – Extrait du dataframe *DonneesStations*

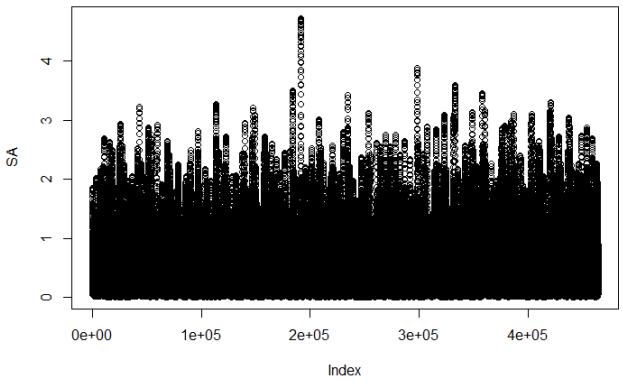


FIGURE 3 – Nuage de points des données *DonneesVagues*

Dans ce projet nous réaliserons dans un premier temps les analyses dans le cas univarié, c'est à dire en ne considérant qu'une seule station puis nous ferons une analyse bivariée en considérant cette fois-ci plusieurs stations.

3 Partie Univariée

Dans cette partie nous serons dans le cas univarié et nous considérerons la station 2. Pour mener à bien les analyses nous mettrons en œuvre deux approches. Dans un premier temps nous utiliserons la méthode *GEV* (Generalized Extreme Value) puis dans un second temps nous utiliserons la méthode *GPD*(Generalized Pareto distribution).

3.1 Approche GEV

Pour réaliser l'étude avec l'approche GEV nous allons nous aider de la méthode par blocs.
Au départ, on suppose avoir des réalisations indépendantes et de même loi F d'un certain phénomène d'intérêt : $X_1, X_2, \ldots, X_k$.
Dans notre cas nous avons la station 2 comme station de référence (SA). Pour obtenir un échantillon de max, on découpe les données en m blocs de même taille n :

$$\underbrace{X_1, X_2, \ldots, X_n}_{Z1} \,|\, \underbrace{X_{n+1}, X_{n+2}, \ldots, X_{2n}}_{Z2} \,|\, \ldots \,|\, \underbrace{X_{(m-1)n+1}, \ldots, X_{mn}}_{Zm} \,|$$

On obtient alors un échantillon de m réalisations de loi GEV : $Z_1, Z_2, \ldots, Z_m$.
Afin de faire des analyses de bonnes qualités il est nécessaire de sélectionner une taille de bloc n adéquat. comme il a été dit précédemment, le dataframe *DonneesVagues* est composé de 464280 lignes. Afin de choisir une bonne valeur de n, nous avons fait un script (disponible en annexe) qui affiche tous les diviseurs de 464280, ces derniers sont affichés en figure 4. À partir de là nous avons pu tester plusieurs valeurs de m et nous avons finalement pris $m = 2980$. Nous aurons donc 2980 blocs tous de même taille $n = 159$. Nous allons maintenant justifier pourquoi nous avons décidé de retenir cette valeur.
Les points du graphique de la figure 5, correspondent aux maximums de chaque bloc. Comme on peut le voir, le nuage de points obtenu est assez dispersé ce qui nous conforte dans l'idée que le paramètre n a été bien choisi.
D'autre part, nous allons également nous appuyer sur le graphique du *Quantile plot* afin de voir si le paramètre n a bien été choisi. On rappelle que le Quantile plot est le nuage de points :

$$\left\{\left(\widehat{G}^{-1}\left(\frac{i}{m+1}\right), z_{i,m}\right), i = 1, \ldots, m\right\}$$

Où

$$\widehat{G}(\frac{i}{m+1}) = \widehat{\mu} - \frac{\widehat{\sigma}}{\widehat{\gamma}}\Big[1 - \Big(-log\Big(\frac{i}{i+1}\Big)\Big)^{-\widehat{\gamma}}\Big]$$

Si l'on observe le graphique de la figure 6, on s'aperçoit que les points du nuage sont proches de la diagonale (ligne bleue) donc l'ajustement est de bonne qualité, la taille des blocs retenue pour l'approche GEV a donc été bien choisie.

```
> print(li_divisor)
 [1]     1     2     3     4     5     6     8    10    12    15    20
[12]    24    30    40    53    60    73   106   120   146   159   212
[23]   219   265   292   318   365   424   438   530   584   636   730
[34]   795   876  1060  1095  1272  1460  1590  1752  2120  2190  2920
[45]  3180  3869  4380  6360  7738  8760 11607 15476 19345 23214 30952
[56] 38690 46428 58035 77380 92856 116070 154760 232140 464280
```

FIGURE 4 – liste des diviseurs de 464280

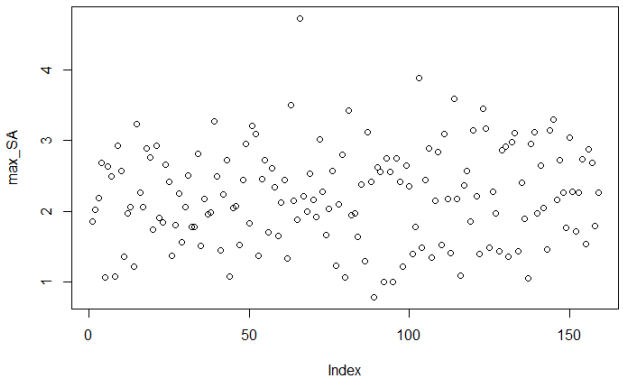


FIGURE 5 – nuage de points obtenu avec la méthode par bloc

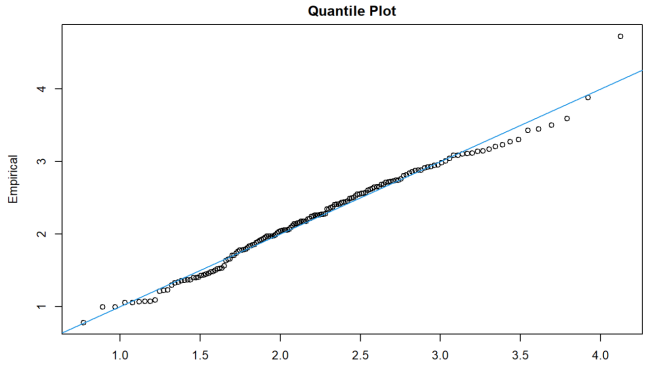


FIGURE 6 – Quantile plot

3.1.1 Niveaux de retour associés aux périodes de retour 100, 500 et 1000

Dans cette sous-section, nous allons nous intéresser aux niveaux de retour $x_{\frac{1}{T}}$ associés aux périodes de retour $T = \frac{1}{q}$ pour $T \in \{100, 500, 1000\}$. L'on souhaite connaître le niveau qui sera dépassé tous les 100, 500 et 1000 ans.

Périodes ou Années T / Niveau de retour	$T = 100$	$T = 500$	$T = 1000$
$x_{\frac{1}{T}}$	3.99003	4.413046	4.563588

TABLE 1 – Niveaux de retour associés aux périodes de retour T

Périodes ou Années T / Intervalles de confiance	$T = 100$	$T = 500$	$T = 1000$
IC de niveau $1 - \alpha = 95\%$	$[3.691942, 4.288119]$	$[3.974492, 4.851601]$	$[4.059768, 5.067409]$

TABLE 2 – Intervalles de confiance des niveaux de retour $x_{\frac{1}{T}}$

En regardant les valeurs des niveaux de retour affichées dans le tableau 1, on constate qu'elles sont bien toutes comprises dans leurs intervalles de confiance respectifs (voir tableau 2). Cela nous confirme une fois de plus que la taille n des blocs choisis pour l'approche GEV est bonne.
D'autre part, on remarque que plus la période de retour T est grande et plus le niveau de retour augmente. Cette augmentation des hauteurs de vagues peut s'expliquer par la montée du niveau de la mer au fil des années. On pourrait donc se demander si des plages comme celles de Carnon, Palavas ou encore La Grande-Motte existeront toujours dans 1000 ans ?

3.2 Approche GPD

Pour mener cette fois-ci l'étude avec l'approche GPD, il faut déterminer la valeur du seuil à utiliser. Une fois de plus, si l'on souhaite réaliser de bonnes analyses, il faut être vigilant au seuil que l'on choisit. Afin de déterminer ce dernier, nous allons nous aider de la figure 7, sur laquelle est tracé le diagramme de durée de vie résiduelle moyenne (Mean residual life plot).

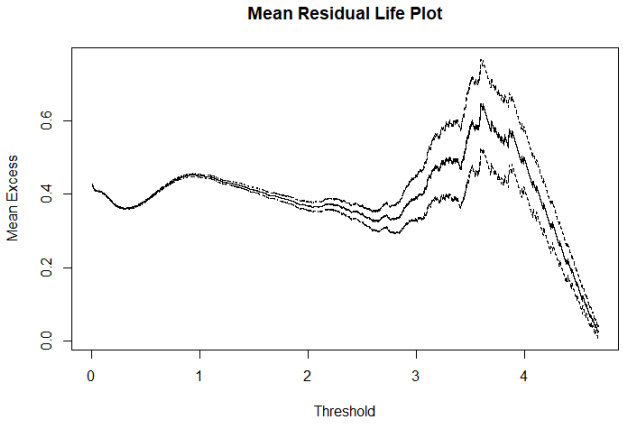
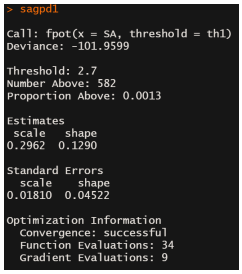
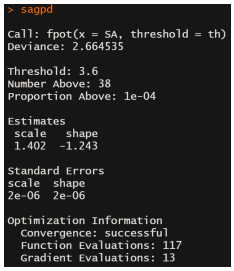


FIGURE 7 – diagramme de durée de vie résiduelle moyenne

Sur la figure 7, on voit que les pics significatifs sont atteints pour des valeurs de seuil (Threshold) valant approximativement 2.7 et 3.65. Nous avons donc testé ces deux valeurs de seuil et nous avons finalement retenu la valeur 3.65. Nous allons maintenant justifier pourquoi nous avons décidé de retenir cette dernière valeur.

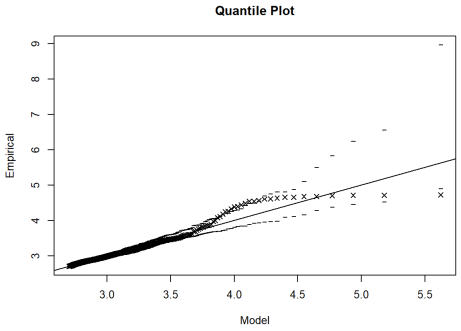


(a) seuil ≈ 2.7

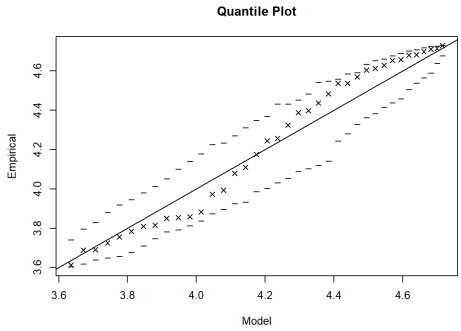


(b) seuil ≈ 3.65

FIGURE 8 – Résultats donnés en sortie de la fonction *fpot* pour les 2 seuils



(a) seuil ≈ 2.7



(b) seuil ≈ 3.65

FIGURE 9 – Quantile plot des 2 seuils

En observant les résultats affichés sur les captures d'écran de la figure 8, on voit que la proportion de valeurs dépassant le seuil (proportion above) est plus faible dans le cas où ce dernier vaut ≈ 3.65 que dans le cas où il vaut ≈ 2.7.
De plus, en observant les graphiques (*Quantile plot*) de la figure 9, on remarque que le nuage de points du seuil ≈ 3.65 est plus proche de la droite en diagonale que le nuage de points du seuil ≈ 2.7.
Ces résultats nous ont donc poussés à choisir le seuil= 3.65 car il nous permettra de faire une analyse de meilleure qualité qu'avec l'autre valeur de seuil.

3.2.1 Niveaux de retour associés aux périodes de retour 100, 500 et 1000

Maintenant que nous avons déterminer le seuil (threshold), nous allons nous intéresser comme dans le cas de l'approche GEV, aux niveaux de retour $x_{\frac{1}{T}}$ associés aux périodes de retour $T = \frac{1}{q}$ pour $T \in \{100, 500, 1000\}$. L'on souhaite connaître le niveau qui sera dépassé tous les 100, 500 et 1000 ans avec l'approche GPD.

Périodes ou Années T / Niveau de retour	$T = 100$	$T = 500$	$T = 1000$
$x_{\frac{1}{T}}$	4.686475	4.72502	4.723653

TABLE 3 – Niveaux de retour associés aux périodes de retour T

Quand on regarde le tableau 3, on constate que le niveau de retour pour la période $T = 500$ est supérieur à celui pour la période $T = 100$. Cependant, pour la période $T = 1000$, la valeur de $x_{\frac{1}{T}}$ est inférieur à celle trouvée pour $T = 500$. Ce qui est différent des résultats obtenus avec l'approche GEV. En effet, pour rappel, avec l'approche GEV on avait trouvé que plus la période de retour T était grande et plus le niveau de retour augmentait, ce qui était cohérent avec la montée des eaux. Dans le cas de ces données, l'approche GEV est donc peut être préférable car elle fournit des résultats plus cohérents que ceux obtenus

avec l'approche GPD.

4 Partie bivariée

Dans cette nouvelle partie, on se place maintenant dans le cas bivarié. L'objectif sera de prédire la valeur du quantile extrême z_p vérifiant :

$$\mathbb{P}(X > z_p | Y > y) = \frac{\mathbb{P}(X > z_p \cap Y > y)}{\mathbb{P}(Y > y)} \tag{1}$$

avec p petit et où Y correspond à la hauteur de vagues associées à la station 2 (S_A) et X à la hauteur de vagues associées à la station 9 (S_B).
Pour mener à bien les analyses dans toute la suite de ce projet, nous mettrons en œuvre uniquement l'approche *GEV* (Generalized Extreme Value) car nous avons vu précédemment que cette dernière fournissait de meilleurs résultats que l'approche GPD dans le cas de nos données. Tout comme dans le cas univarié, nous allons nous aider de la méthode par blocs. Afin de réaliser des analyses de bonnes qualités, nous devons correctement choisir la taille n des blocs. En nous aidant encore une fois du script (disponible en annexe) affichant tous les diviseurs de 464280, on décide de prendre $m = 2980$ blocs, tous de même taille $n = 159$.

4.1 Analyse entre les stations 2 et 9

Sur la figure 10 nous avons tracé deux nuages de points. Les points du nuage rouge correspondent aux maximums de chaque bloc dans le cas de la station S_B et ceux du nuage bleu aux maximums de chaque bloc dans le cas de la station S_A. Les points des deux nuages sont assez bien dispersés.

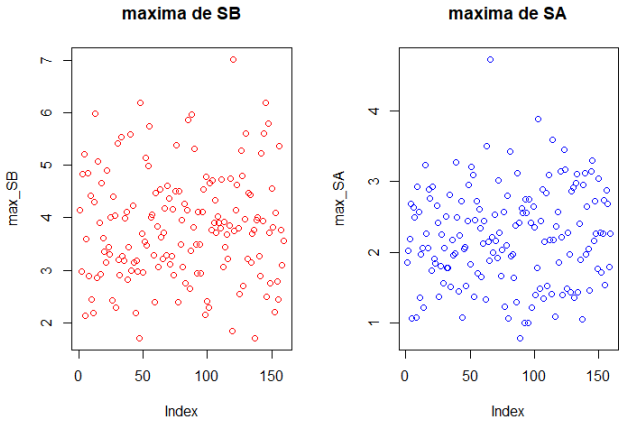


FIGURE 10 – Nuage de points obtenu avec la methode par bloc pour S_B et S_A

Nous allons maintenant effectuer un ajustement par une loi GEV sur chacune de deux marginales et nous allons récupérer les paramètres pour chaque ajustement des marginales. Nous les avons affichés dans le tableau 4.

	Paramètres S_A	Paramètres S_B
$\widehat{\mu}$	1.9483809	3.4137728
$\widehat{\sigma}$	0.6317954	0.9391569
$\widehat{\gamma}$	−0.1636501	−0.1620597

TABLE 4 – paramètres des ajustement des marginales

Calculons la probabilité de l'équation 1.
Les paramètres du tableau 4 vont nous aider à calculer cette probabilité. En effet, ils seront utilisés dans

la fonction *pgev* de R qui va nous permettre de déterminer le dénominateur $\mathbb{P}(Y > y)$.
on rappelle qu'ici, à titre d'exemple nous avons choisi la valeur $y = 4$.
Après calcul, on trouve que $\mathbb{P}(Y > 4) = 0.4041958$.
Nous avons également besoin de calculer le numérateur à savoir $\mathbb{P}(X > z_p \cap Y > y)$.
La fonction *fbvevd* de R nous permettra de récupérer les paramètres (*loc*,*scale* et *shape*) nécessaires au calcul du numérateur. La valeur de ce dernier est ensuite obtenue via la fonction *pbvevd* et on obtient que $\mathbb{P}(X > z_p \cap Y > 4) = 0.01589988$.
On en déduit donc que $\mathbb{P}(X > z_p | Y > 4) = 0.03933707$. C'est à dire que la probabilité que la hauteur de la vague de la station 9 soit supérieure z_p sachant que la hauteur de la vague de la station 2 est supérieure à 4 vaut environ 3.93%.

Nous allons maintenant procéder à une analyse bivariée entre les stations 2 et 9. La figure 11 représente les hauteurs maximales des vagues selon les blocs pour les stations 2 et 9. Après observation de ce nuage de points, il semblerait que les données des deux stations ne soient pas corrélées entre elles car le nuage est assez dispersé.

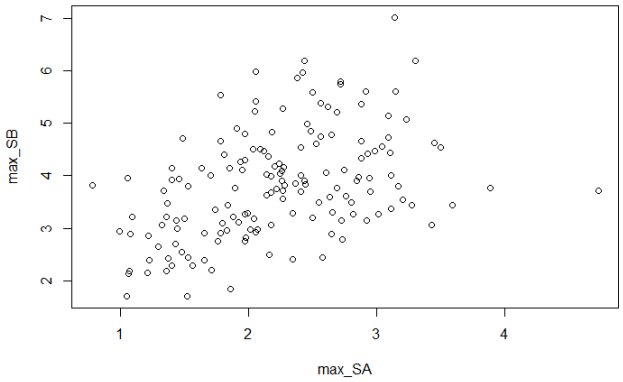


FIGURE 11 – hauteurs maximales des vagues selon les blocs pour les station 2 et 9

Dans la suite de cette partie, nous allons faire des analyses plus fines afin de vérifier si notre hypothèse d'indépendance entre les données des deux stations est juste ou pas. Pour ce faire nous allons utiliser deux différents modèles d'estimation paramétrique à savoir le modèle logistique et celui logistique asymétrique. Nous allons ensuite sélectionner le meilleur de ces deux modèles à l'aide du critère AIC. D'après la source [2], le critère AIC se définit de la manière suivante :

$$AIC = 2k - 2ln(L)$$

Où k est le nombre de paramètres à estimer du modèle et L est le maximum de la fonction de vraisemblance du modèle.
D'après la capture d'écran de la figure 12, le modèle logistique est celui ayant l'AIC le plus faible avec une valeur de 763.3703 contre 766.2632 pour le modèle logistique asymétrique. On en déduit donc que le meilleur des deux modèles est celui logistique (*log*) et c'est donc ce dernier que nous sélectionnerons.

```
[1] "log"
      loc1      scale1      shape1       loc2      scale2      shape2         dep
 1.93762130  0.62531368 -0.11469253  3.39859713  0.94585461 -0.08400814  0.70145482
[1] "alog"
      loc1      scale1      shape1       loc2      scale2      shape2        asy1
 1.94660170  0.64184084 -0.15993820  3.38147275  0.96650792 -0.05906914  0.71829377
      asy2         dep
 0.99973585  0.61771259
> aic
                   log     alog
  0.0000    0.0000 763.3703 766.2632
```

FIGURE 12 – AIC des modèles *log* et *alog*

Ajustons maintenant le modèle *log*. En observant la figure 13, on voit que la valeur du paramètre dépendance vaut 0.3738562. Il est proche de zéro, on en déduit donc que les données des stations 9 et 2 ne sont pas corrélées entre elles. Notre hypothèse d'indépendance des données est donc valide.

```
> mod_log

Call: fbvevd(x = maxSASB, model = "log")
Deviance: 749.3703
AIC: 763.3703
Dependence: 0.3738562

Estimates
    loc1    scale1    shape1      loc2    scale2    shape2       dep
 1.93762   0.62531  -0.11469   3.39860   0.94585  -0.08401   0.70145

Standard Errors
    loc1    scale1    shape1      loc2    scale2    shape2       dep
0.05480  0.03722  0.05042  0.08433  0.05718  0.06128  0.05245

Optimization Information
  Convergence: successful
  Function Evaluations: 46
  Gradient Evaluations: 11
```

FIGURE 13 – Vérification de la validité de l'hypothèse d'indépendance

4.2 Analyse entre les stations 2 et 7

Dans cette partie, nous cherchons dans un premier temps une station notée S_C qui est plus proche de S_A que S_B. Une fois que nous l'aurons trouvée, nous ferons une analyse entre notre station de référence S_A et la nouvelle station S_C. Pour déterminer S_C, nous avons utilisé la norme euclidienne présentée dans le document [3] et nous avons appliqué les formules 2 et 3 sur les coordonnées géographiques des stations (longitude et latitude) disponibles dans le dataframe *DonneesStations*.

$$\sqrt{\Big((longitude[S_A] - longitude[S_C])^2 + (latitude[S_A] - latitude[S_C])^2\Big)} \tag{2}$$

$$\sqrt{\Big((longitude[S_C] - longitude[S_B])^2 + (latitude[S_C] - latitude[S_B])^2\Big)} \tag{3}$$

En prenant $S_C = station7$, nous obtenons après application des formules 2 et 3 que :

$$||\overrightarrow{S_A S_C}|| = 2.13305$$
$$||\overrightarrow{S_C S_B}|| = 3.018848$$

Donc le choix de la station 7 est valide car cette dernière est effectivement plus proche de S_A que S_B.

Sur la figure 14 nous avons tracé deux nuages de points. Les points du nuage magenta correspondent aux maximums de chaque bloc dans le cas de la station S_C et ceux du nuage bleu clair aux maximums de chaque bloc dans le cas de la station S_A. Les points des deux nuages sont assez bien dispersés.

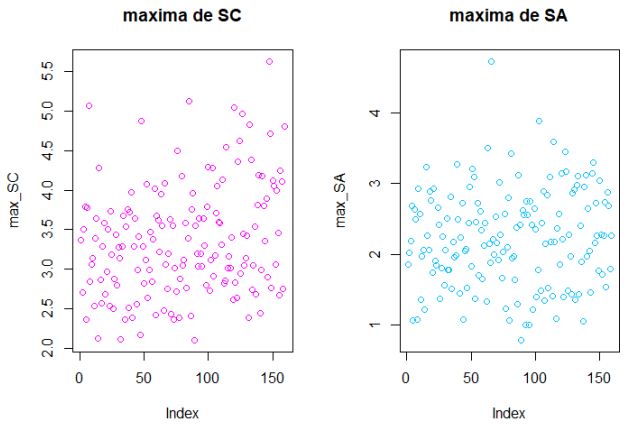


FIGURE 14 – Nuage de points obtenu avec la méthode par bloc pour S_A et S_C

Nous allons maintenant procéder à une analyse bivariée entre les stations 2 et 7. La figure 15 représente les hauteurs maximales des vagues selon les blocs pour les stations 2 et 7. Après observation de ce nuage

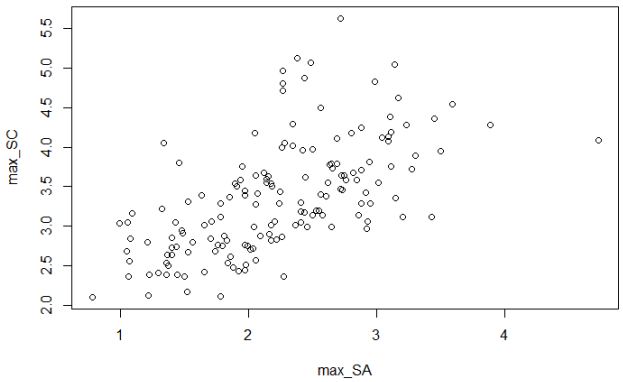


FIGURE 15 – hauteurs maximales des vagues selon les blocs pour les station 2 et 7

de points, il semblerait que les données des deux stations soient moins dispersées que celles des stations 2 et 9 (voir figure 11).
Dans la suite de cette partie, nous allons faire des analyses plus fines afin de voir s'il y a plus de corrélations entre les données des stations 2 et 7. Logiquement vu que les stations 2 et 7 sont plus proches l'une de l'autre que les stations 2 et 9, il devrait y avoir ici plus de corrélations. Vérifions le. Pour ce faire nous allons encore une fois utiliser les modèles d'estimation paramétrique logistique et celui logistique asymétrique. Nous allons ensuite sélectionner le meilleur de ces deux modèles à l'aide du critère AIC.
D'après la capture d'écran de la figure 16, le modèle logistique est celui ayant l'AIC le plus faible avec une valeur de 590.3655 contre 596.5881 pour le modèle logistique asymétrique. On en déduit donc que le meilleur des deux modèles est celui logistique (*log*) et c'est donc ce dernier que nous sélectionnerons.

```
[1] "log"
      loc1       scale1       shape1        loc2       scale2       shape2         dep
 1.93879299  0.63285466 -0.09476498  3.01701707  0.58282072  0.04314712  0.55673975
[1] "alog"
      loc1       scale1       shape1        loc2       scale2       shape2        asy1
 1.94957198  0.65968002 -0.11870856  3.02894423  0.61959334  0.06156629  0.99900242
      asy2         dep
 0.77985602  0.47994666
> aic
                   log     alog
  0.0000    0.0000 590.3655 596.5881
```

FIGURE 16 – AIC des modèles *log* et *alog*

Ajustons maintenant le modèle *log*. En observant la figure 17, on voit que la valeur du paramètre dépendance vaut 0.5290586. Il est donc plus grand que dans le cas des stations 2 et 9 (voir figure 13). Ceci est parfaitement cohérent avec ce que l'on avait dit précédemment. En effet, plus l'on va prendre des stations proches entre elles et plus il y aura de corrélations entre les données de ces dernières.

```
> mod_log <- fbvevd(maxSASC, model = "log")
> mod_log

Call: fbvevd(x = maxSASC, model = "log")
Deviance: 576.3655
AIC: 590.3655
Dependence: 0.5290586

Estimates
    loc1    scale1    shape1      loc2    scale2    shape2       dep
 1.93879   0.63285  -0.09476   3.01702   0.58282   0.04315   0.55674

Standard Errors
   loc1   scale1   shape1    loc2   scale2   shape2      dep
0.05542  0.03747  0.05050  0.05254  0.03681  0.06190  0.04339

Optimization Information
  Convergence: successful
  Function Evaluations: 56
  Gradient Evaluations: 12
```

FIGURE 17 – Vérification de la validité de l'hypothèse d'indépendance

4.3 Analyse entre les stations 9 et 4

Dans cette partie, nous cherchons dans un premier temps une station notée S_D qui est plus proche de S_B que S_A. Une fois que nous l'aurons trouvée, nous ferons une analyse entre la station S_B et la nouvelle station S_D. Pour déterminer S_D, nous avons utiliser comme dans la partie précédente, la norme euclidienne et nous avons appliqué les formules 4 et 5 sur les coordonnées géographiques des stations (longitude et latitude) disponibles dans le dataframe *DonneesStations*.

$$\sqrt{\Big((longitude[S_D] - longitude[S_B])^2 + (latitude[S_D] - latitude[S_B])^2\Big)} \tag{4}$$

$$\sqrt{\Big((longitude[S_A] - longitude[S_D])^2 + (latitude[S_A] - latitude[S_D])^2\Big)} \tag{5}$$

En prenant $S_D = station4$, nous obtenons après application des formules 2 et 3 que :

$$||\overrightarrow{S_B S_D}|| = 2.40698$$
$$||\overrightarrow{S_A S_D}|| = 2.604918$$

Donc le choix de la station 4 est valide car cette dernière est effectivement plus proche de S_B que S_A.

Sur la figure 18 nous avons tracé deux nuages de points. Les points du nuage orange correspondent aux maximums de chaque bloc dans le cas de la station S_C et ceux du nuage vert aux maximums de chaque bloc dans le cas de la station S_A. Les points des deux nuages sont assez bien dispersés.

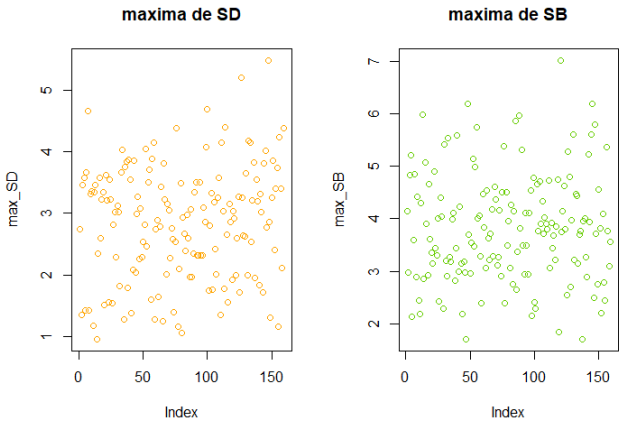


FIGURE 18 – Nuage de points obtenu avec la méthode par bloc pour S_B et S_D

Nous allons maintenant procéder à une analyse bivariée entre les stations 9 et 4. La figure 19 représente les hauteurs maximales des vagues selon les blocs pour les stations 9 et 4. Après observation de ce nuage de points, il semblerait que les données des deux stations soient moins dispersées que celles des stations 2 et 9 (voir figure 11).
Dans la suite de cette partie, nous allons faire des analyses plus fines afin de voir s'il y a plus de corrélations entre les données des stations 9 et 4. Logiquement vu que les stations 9 et 4 sont plus proches l'une de l'autre que les stations 2 et 9, il devrait y avoir ici plus de corrélations. Vérifions le. Pour ce faire nous allons encore une fois utiliser les modèles d'estimation paramétrique logistique et celui logistique asymétrique. Nous allons ensuite sélectionner le meilleur de ces deux modèles à l'aide du critère AIC.

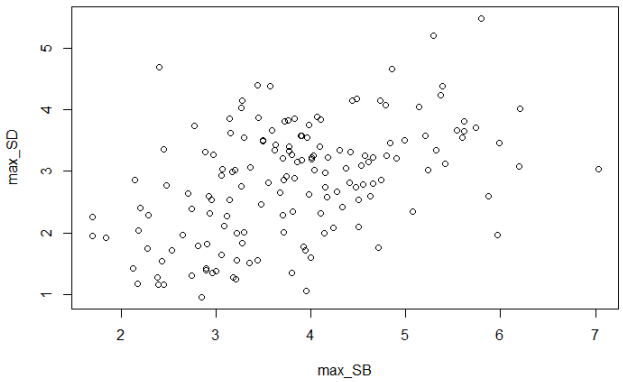


FIGURE 19 – hauteurs maximales des vagues selon les blocs pour les station 9 et 4

D'après la capture d'écran de la figure 20, le modèle logistique est celui ayant l'AIC le plus faible avec une valeur de 857.6410 contre 862.1909 pour le modèle logistique asymétrique. On en déduit donc que le meilleur des deux modèles est celui logistique (*log*) et c'est donc ce dernier que nous sélectionnerons.

```
[1] "log"
     loc1     scale1     shape1      loc2     scale2     shape2       dep
 3.4025050  0.9372093 -0.1128537  2.4828524  0.9052501 -0.2014719  0.6931814
[1] "alog"
     loc1     scale1     shape1      loc2     scale2     shape2      asy1
 3.4329581  0.9435664 -0.1228468  2.4943941  0.9059510 -0.1777674  0.9991988
     asy2       dep
 0.9954272  0.6968118
> aic
                  log     alog
  0.0000   0.0000 857.6410 862.1909
```

FIGURE 20 – AIC des modèles *log* et *alog*

Ajustons maintenant le modèle *log*. En observant la figure 21, on voit que la valeur du paramètre dépendance vaut 0.383155. Il est donc légèrement plus grand que dans le cas des stations 2 et 9 (voir figure 13). Ici la station S_D est légèrement plus proche de S_B qu'elle ne l'est de S_A, c'est pourquoi la corrélation entre les données de S_B et S_D n'est que légèrement plus grande que celle entre les données de S_A et S_B. Si l'on voulait avoir une corrélation plus élevée, il faudrait choisir une station qui serait à la fois encore plus proche de S_B et encore plus éloignée de S_A. Au final, même si ici la différence est moins marquante que pour la station 2 et 7, on constate toujours que plus l'on va prendre des stations proches entre elles et plus il y aura de corrélations entre les données de ces dernières.

```
> mod_log

Call: fbvevd(x = maxSBSD, model = "log")
Deviance: 843.641
AIC: 857.641
Dependence: 0.383155

Estimates
   loc1   scale1   shape1     loc2   scale2   shape2      dep
 3.4025   0.9372  -0.1129   2.4829   0.9053  -0.2015   0.6932

Standard Errors
   loc1   scale1   shape1     loc2   scale2   shape2      dep
0.08270  0.05662  0.05584  0.07923  0.05409  0.05001  0.05019

Optimization Information
  Convergence: successful
  Function Evaluations: 46
  Gradient Evaluations: 11
```

FIGURE 21 – Vérification de la validité de l'hypothèse d'indépendance

5 Conclusion

À travers ce projet, nous avons pu voir qu'au fil des années, nous serons confrontés à des valeurs extrêmes de vagues de plus en plus grandes. Cela est sans aucun doute une conséquence de la hausse des températures et en particulier de la montée du niveau des océans. Nous comprenons donc pourquoi la lutte contre le réchauffement climatique est au cœur des enjeux scientifiques actuels. D'autre part, nous avons pu constater que plus nous prendrons des mesures issues de stations proches et plus les données associées à ces dernières seront fortement corrélées.

6 Annexes

Ci dessous, l'export de code permettant d'afficher la liste des diviseurs de 464280.

```
li_divisor = c()
s = length(SA)
for(d in 1:s){
  if(s%%d == 0){
    li_divisor = c(li_divisor, d)
  }
}
print(li_divisor)
```

Dans l'export de code ci dessous on applique la méthode par bloc puis on affiche le nuage de points des maximums des blocs.

```
for(i in 1:(length(SA)/n)){ max_SA[i]<-max(SA[((i-1)*n+1):(i*n)])}
maxfit<-gev.fit(max_SA)
plot(max_SA)
```

La fonction affichée ci-dessous permet d'afficher les graphiques de diagnostics tels que le *Quantile Plot* et le *Probability Plot*.

```
gev.diag(maxfit)
```

Dans l'export de code ci-dessous, on affiche la valeur du niveau de retour $x_{\frac{1}{T}}$ associée à la période de retour $T = 100$ ainsi qu'un intervalle de confiance de niveau $1 - \alpha = 95\%$

```
T=100
q=1/T
dep100=fgev(max_SA,prob=q)
cat(" le montant qui sera depasse tous les 100 ans est ", dep100$estimate[1])

#IC 95%
cat(
  " l'intervalle de confiance de l'estimateur du depassement de 100 ans est:"
  ,"\n","[" ,dep100$estimate[1]- 1.96*dep100$std.err[1],","
  ,dep100$estimate[1]+ 1.96*dep100$std.err[1],"]","\n"
  )
```

Dans l'export de code ci-dessous, on affiche la valeur du niveau de retour $x_{\frac{1}{T}}$ associée à la période de retour $T = 500$ ainsi qu'un intervalle de confiance de niveau $1 - \alpha = 95\%$

```
T=500
q=1/T
dep500=fgev(max_SA,prob=q)
cat(" le montant qui sera depasse tous les 500 ans est ", dep500$estimate[1])

#IC 95%
cat(
  " l'intervalle de confiance de l'estimateur du depassement de 500 ans est:",
  "\n","[" ,dep500$estimate[1]- 1.96*dep500$std.err[1],"," ,
  dep500$estimate[1]+ 1.96*dep500$std.err[1],"]","\n"
    )
```

Dans l'export de code ci-dessous, on affiche la valeur du niveau de retour $x_{\frac{1}{T}}$ associée à la période de retour $T = 1000$ ainsi qu'un intervalle de confiance de niveau $1 - \alpha = 95\%$

```
T=1000
q=1/T
dep1000=fgev(max_SA,prob=q)
cat(" le montant qui sera depasse tous les 1000 ans est ", dep1000$estimate[1])

#IC 95%
cat(
  " l'intervalle de confiance de l'estimateur du depassement de 100 ans est:",
  "\n","[" ,dep1000$estimate[1]- 1.96*dep1000$std.err[1],"," ,
  dep1000$estimate[1]+ 1.96*dep1000$std.err[1],"]","\n"
    )
```

La fonction ci-dessous affiche le *Mean Residual Life Plot*

```
mrlplot(SA)
```

Via l'export de code ci-dessous, on met en œuvre l'approche GPD avec la fonction *fpot* et on affiche les graphiques de diagnostics (th correspond au seuil threshold).

```
th=3.6
sagpd=fpot(SA,threshold=th)
sagpd

par(mfrow=c(2,2))
plot(sagpd)
```

L'export de code ci-dessous permet de calculer le niveau de retour $x_{\frac{1}{T}}$ associé à la période de retour $T = 100$ avec l'approche GPD, $n = 2920$ correspond au nombre de blocs.

```
depas100=fpot(SA,threshold=th,npp=1,mper = 100*n, std.err = FALSE)
cat(
  " le montant qui sera depasse tous les 100 ans est ", depas100$estimate[1],
  "par l'autre methode on avait ",dep100$estimate[1]
  )
```

L'export de code ci-dessous permet de calculer le niveau de retour $x_{\frac{1}{T}}$ associé à la période de retour $T = 500$ avec l'approche GPD, $n = 2920$ correspond au nombre de blocs.

```
depas500=fpot(SA,threshold=th,npp=1,mper = 500*n,std.err = FALSE)
cat(
  " le montant qui sera depasse tous les 500 ans est ", depas500$estimate[1],
  "par l'autre methode on avait",dep500$estimate[1]
  )
```

L'export de code ci-dessous permet de calculer le niveau de retour $x_{\frac{1}{T}}$ associé à la période de retour $T = 1000$ avec l'approche GPD, $n = 2920$ correspond au nombre de blocs.

```
depas1000=fpot(SA,threshold=th,npp=1,mper = 1000*n,std.err = F)
cat(
  " le montant qui sera depasse tous les 1000 ans est  ", depas1000$estimate[1],
  "par l'autre methode on avait",dep1000$estimate[1]
  )
```

Ajustement par une loi GEV pour chacune des deux marginales.
Récupération des paramètres chaque ajustement des marginales.

```
gev_SA=gev.fit(max_SA)
gev_SB=gev.fit(max_SB)
summary(gev_SA)


parametre_SA=c(gev_SA$mle[1],gev_SA$mle[2],gev_SA$mle[3])
parametre_SB=c(gev_SB$mle[1],gev_SB$mle[2],gev_SB$mle[3])

parametres<-data.frame(parametre_SA,parametre_SB)
rownames(parametres)=c("mu","sigma","gamma")
colnames(parametres)=c("parametre SA","parametre SB")
parametres
```

Calcul du dénominateur $\mathbb{P}(Y > y)$

```
model_Y=pgev(
  4,loc=gev_SB$mle[1], scale = gev_SB$mle[2], shape =gev_SB$mle[3],
  lower.tail = FALSE
  )
```

Avec la commande ci dessous, on récupère les paramètres utiles pour calculer le numérateur $\mathbb{P}(X > z_p \cap Y > y)$

```
gevfb_SAB= fbvevd(maxSASB)
```

Via l'export de code ci-dessous, on calcule le numérateur $\mathbb{P}(X > z_p \cap Y > y)$

```
model_X_Y= pbvevd(c(1,4), dep = 0.70145,
                  mar1 = c(loc1=1.93762, scale1=0.62531, shape1=-0.11469),
                  mar2=c(loc2=3.39860, scale2=0.94585, shape2=-0.08401)
                  )
```

Affichage du nuage de points bivarié

```
plot(maxSASB)
```

L'export de code ci-dessous nous permet de sélectionner le meilleur modèle à l'aide du critère AIC

```
a=c("log", "alog")
aic= rep(0,length(a))
for(i in  a){
  print(i)
  print( fbvevd(maxSASB, model = i)$estimate)
  aic[i]=AIC(fbvevd(maxSASB, model = i))
}
aic
```

L'export de code ci-dessous nous permet d'ajuster le meilleur modèle et de vérifier le niveau de corrélation des données

```
mod_log <- fbvevd(maxSASB, model = "log")
mod_log
par(mfrow=c(3,2))
plot(mod_log)
```

Dans l'export de code ci-dessous, on calcule la distance séparant la station 2 et la station 7 ainsi que la distance séparant la station 9 avec la station 7.

```
sqrt((data2$lon[2] - data2$lon[7])^2 + (data2$lat[2] - data2$lat[7])^2)
sqrt((data2$lon[7] - data2$lon[9])^2 + (data2$lat[7] - data2$lat[9])^2)
```

Affichage du nuage de points bivarié pour les stations 2 et 7

```
plot(maxSASC)
```

L'export de code ci-dessous nous permet de sélectionner le meilleur modèle à l'aide du critère AIC

```
a=c("log", "alog")
aic= rep(0,length(a))
for(i in  a){
  print(i)
  print( fbvevd(maxSASC, model = i)$estimate)
  aic[i]=AIC(fbvevd(maxSASC, model = i))
}
aic
```

L'export de code ci-dessous nous permet d'ajuster le meilleur modèle et de vérifier le niveau de corrélation des données

```
mod_log <- fbvevd(maxSASC, model = "log")
mod_log
par(mfrow=c(3,2))
plot(mod_log)
```

Dans l'export de code ci-dessous, on calcule la distance séparant la station 4 et la station 9 ainsi que la distance séparant la station 4 avec la station 2.

```
sqrt((data2$lon[4] - data2$lon[9])^2 + (data2$lat[4] - data2$lat[9])^2)
sqrt((data2$lon[2] - data2$lon[4])^2 + (data2$lat[2] - data2$lat[4])^2)
```

Affichage du nuage de points bivarié pour les stations 4 et 9

```
plot(maxSBSD)
```

L'export de code ci-dessous nous permet de sélectionner le meilleur modèle à l'aide du critère AIC

```
a=c("log", "alog")
aic= rep(0,length(a))
for(i in  a){
  print(i)
  print( fbvevd(maxSBSD, model = i)$estimate)
  aic[i]=AIC(fbvevd(maxSBSD, model = i))
}
aic
```

L'export de code ci-dessous nous permet d'ajuster le meilleur modèle et de vérifier le niveau de corrélation des données

```
mod_log <- fbvevd(maxSBSD, model = "log")
mod_log
par(mfrow=c(3,2))
plot(mod_log)
```

7 Bibliographie

[1] https://reporterre.net/Le-golfe-du-Lion-est-tres-vulnerable-a-la-montee-des-eaux page web rédigée par Alexandre Brun et Benoît Devillers

[2] https://fr.wikipedia.org/wiki/Crit%C3%A8re_d%27information_d%27Akaike page web Wikipédia

[3] https://www.normalesup.org/~simonet/teaching/caml-prepa/tp-caml-2001-02.pdf TP MPSI – Option Informatique rédigé par Vincent Simonet en 2021