# Simulations

2024-10-16

## Simulation Brown-Resnick

Comparison with and without advection on a 2x2 grid, 30 time steps and 2 sites. The advection is set to 0.5 in the x direction and 0.2 in the y direction. Seed is fixed.

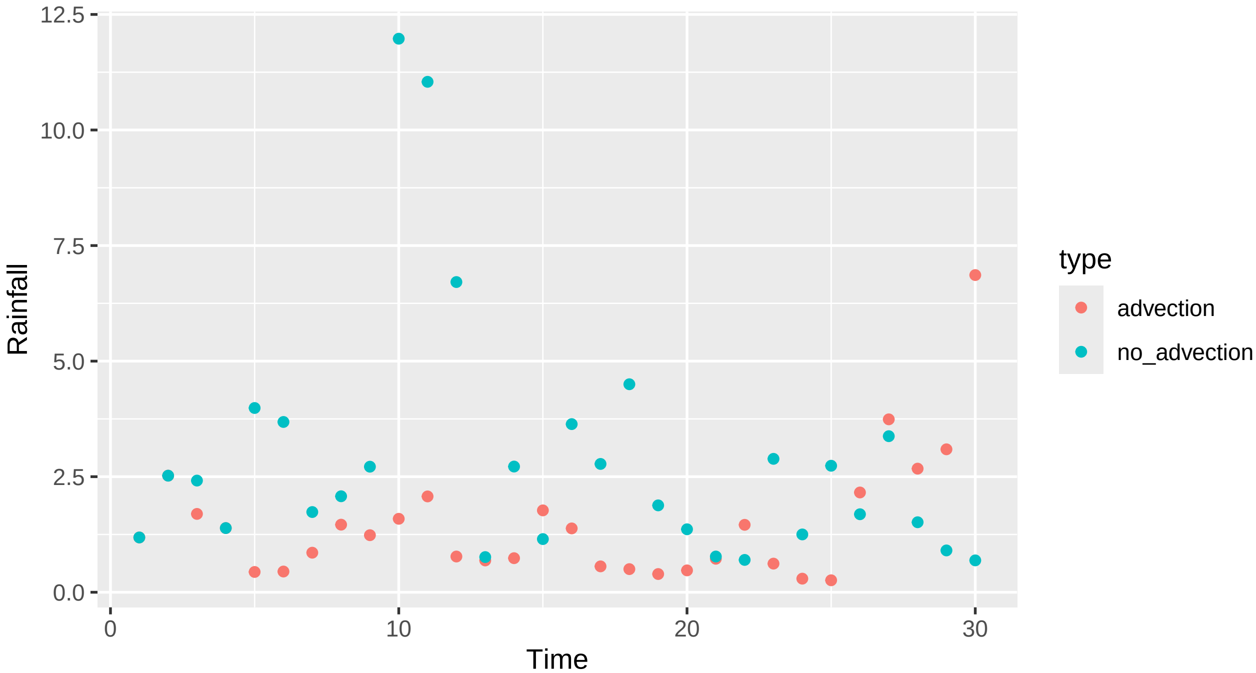


Figure 1: Rainfall simulation with and without advection

## Simulation with 25 sites and 300 time steps without advection

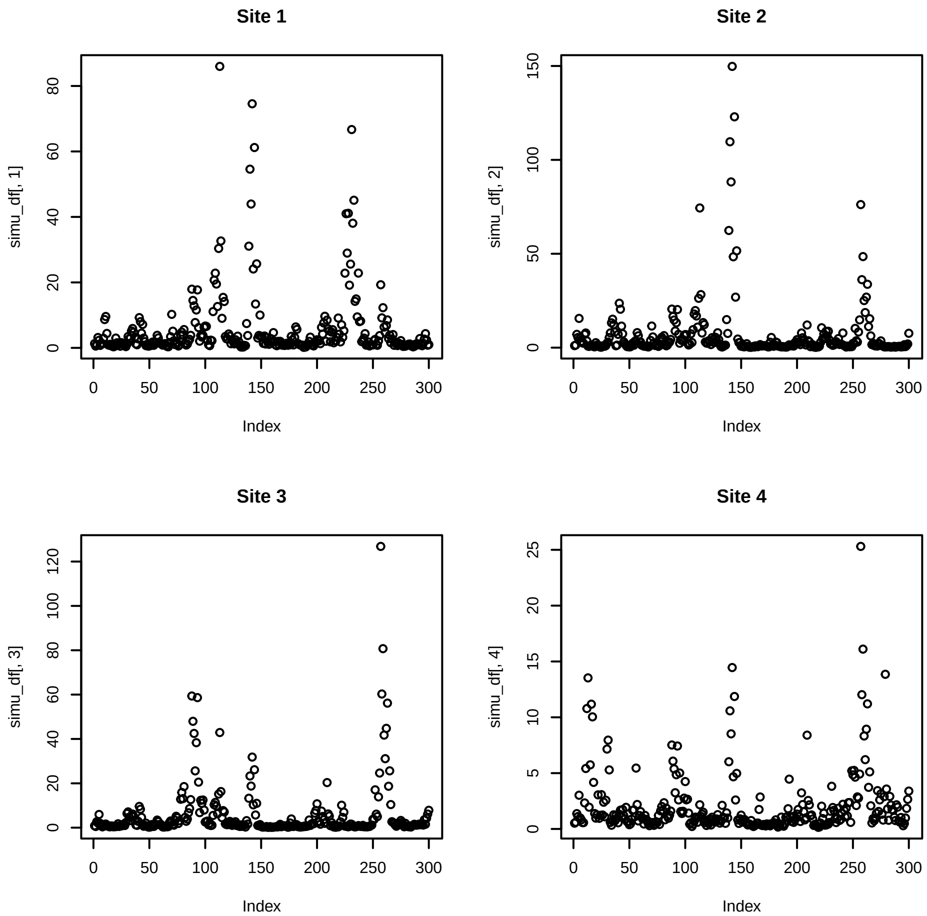


Figure 2: Rainfall simulation with 25 sites and 300 time steps

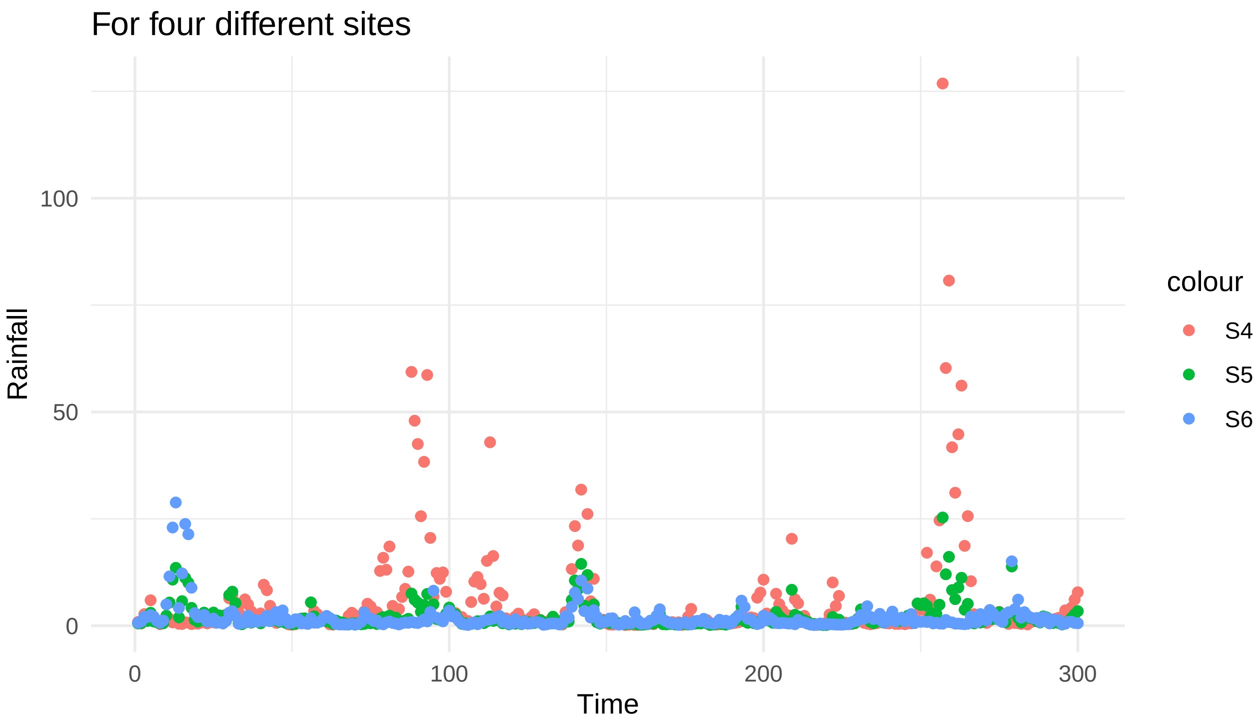


Figure 3: Rainfall simulation with 25 sites and 300 time steps on the same plot

## Verification of marginal distributions (Gumbel)

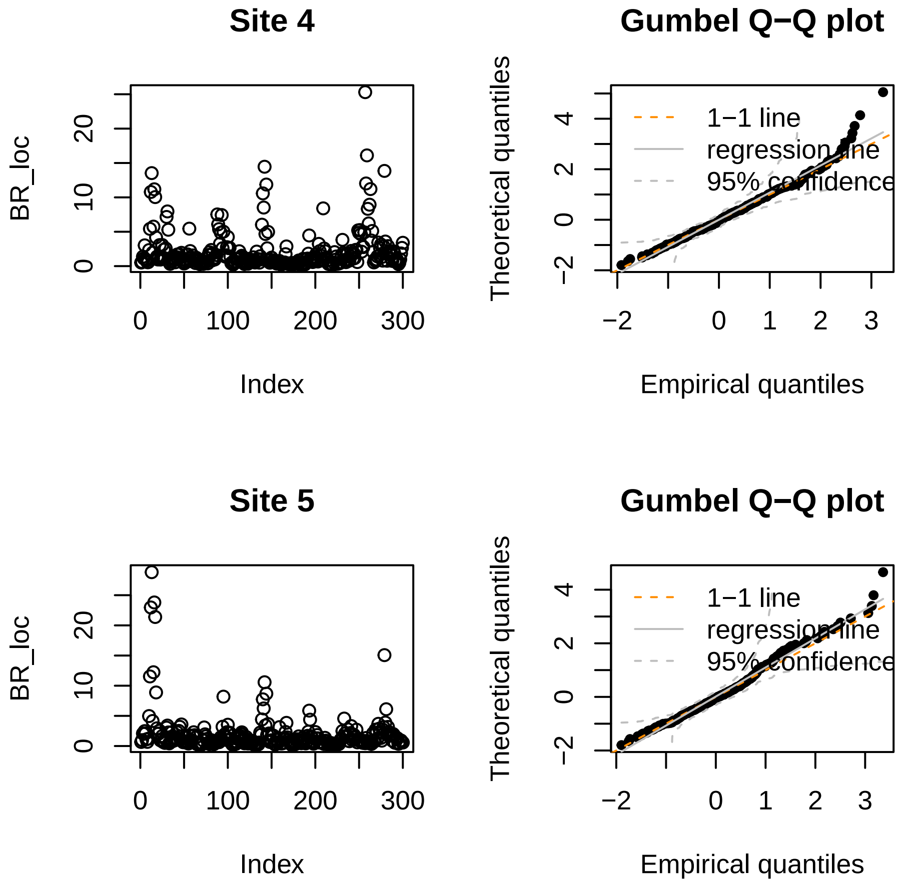


Figure 4: QQ-plot of the Gumbel distribution for two sites

## Simulation of r-pareto with advection

The optimisation of the variogram parameters without advection works. So I need to focus on the advection part and see if my r-pareto advected simulation works. I will simulate a 5x5 grid with 30 time steps and 20 realizations.

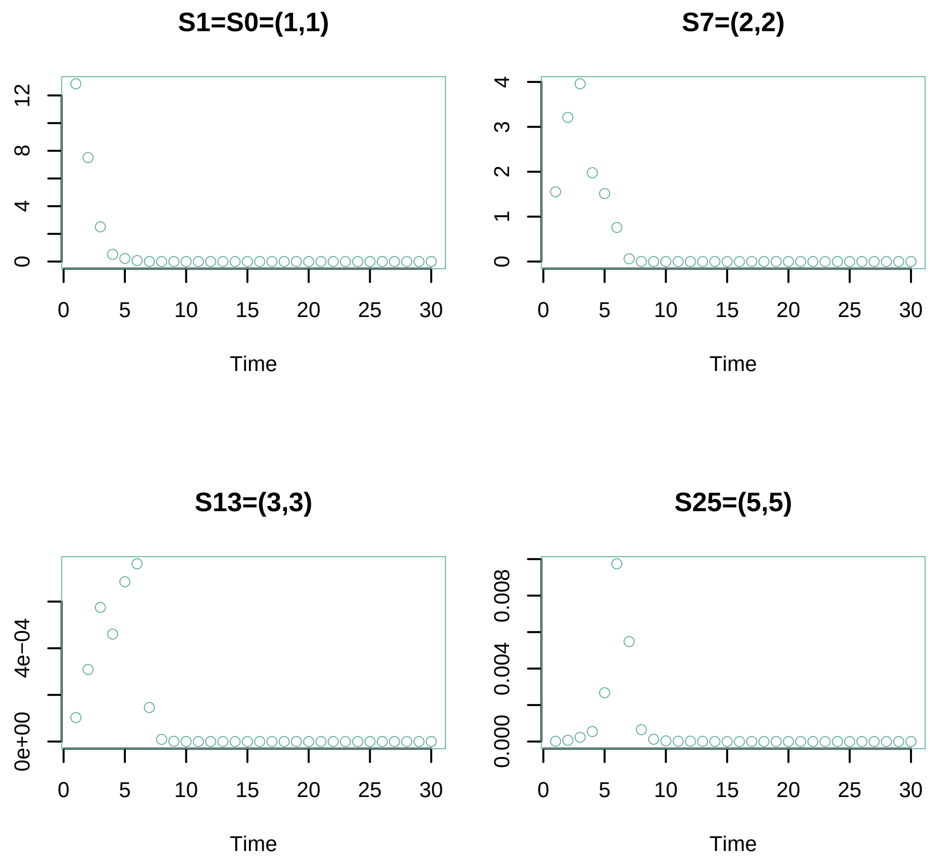


Figure 5: Time series for 4 sites of the first realization in the advection direction

## Verification of marginal distributions (GPD)

```
## $threshold
##        80%
## 0.03388002
##
## $nexc
## [1] 6
##
## $conv
## [1] 0
##
## $nllh
## [1] -19.70164
##
## $mle
## [1]  0.3760663 -1.5154808
##
## $rate
## [1] 0.2
##
## $se
## [1] 1.99997e-06 6.24615e-04
```

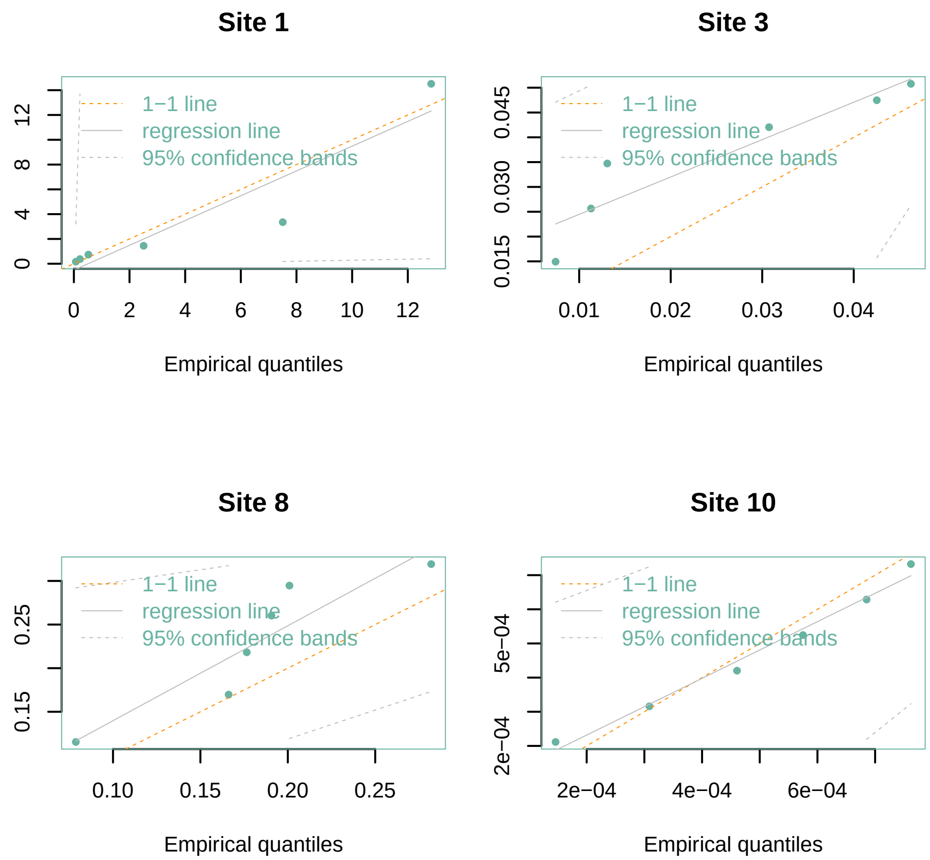


Figure 6: QQ-plot of the GPD distribution for four sites, with quantile at 0.8 and threshold is 0.147

With the threshold $u = 1$, some sites do not have exceedances.

## Check variogram inside the simulation

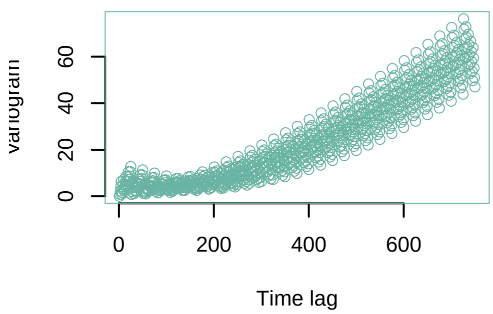


Figure 7: Variogram relative to the conditional spatio-temporal point for all sites

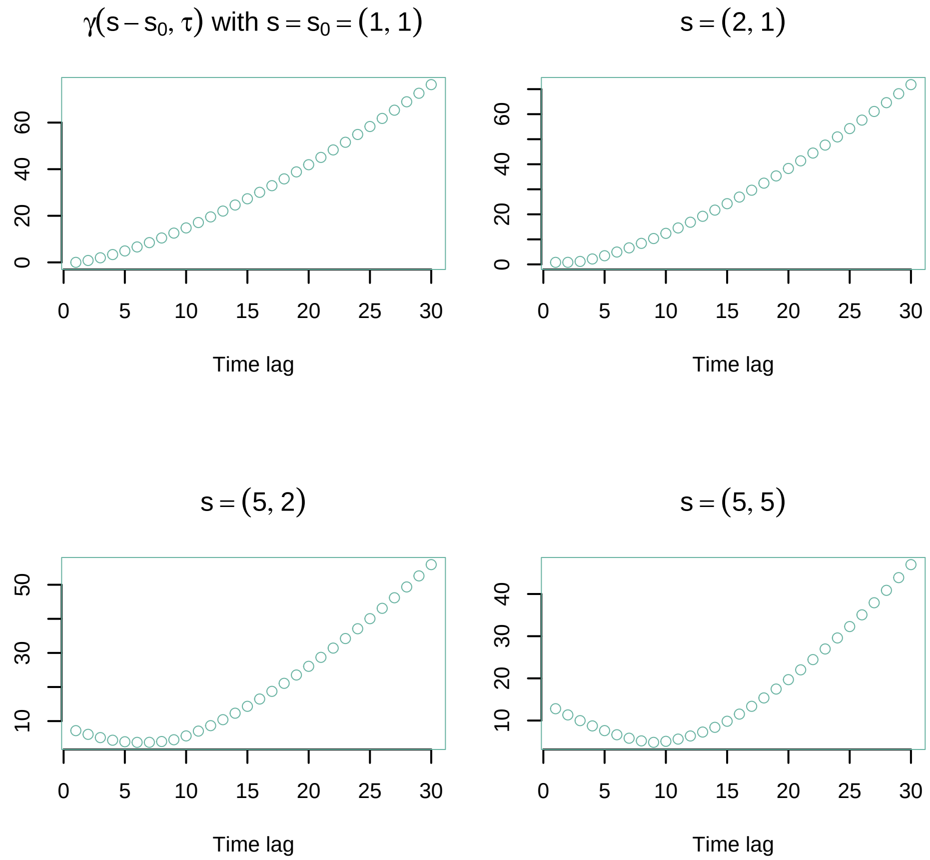


Figure 8: Variogram relative to the conditional spatio-temporal point for four sites

```
s0 <- c(1, 1)
s0_x <- s0[1]
s0_y <- s0[2]
t0 <- 1
print(gamma[s0_x, s0_y, t0] == 0)
```

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

```
s_x <- 1
s_y <- 1
time <- 2
semivario_s_t <- beta1 * abs(s0_x - s_x)^alpha1 + # no advection
                 beta1 * abs(s0_y - s_y)^alpha1 +
                 beta2 * abs(t0 - time)^alpha2
print(gamma[s_x, s_y, time] != 2*semivario_s_t) # as to be different
```

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

```
tau <- t0 - time
semivario_s_t_adv <- beta1 * abs(s0_x - s_x - adv[1] * tau)^alpha1 +
                 beta1 * abs(s0_y - s_y - adv[2]*tau)^alpha1 +
                 beta2 * abs(tau)^alpha2
print(gamma[s_x, s_y, time] == 2 * semivario_s_t_adv)
```

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

```
semivario_s_t_adv <- beta1 * abs(s0_x - s_x - adv[1] * abs(tau))^alpha1 +
                 beta1 * abs(s0_y - s_y - adv[2]*abs(tau))^alpha1 +
                 beta2 * abs(tau)^alpha2
print(gamma[s_x, s_y, time] == 2 * semivario_s_t_adv)
```

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

```
s_x <- 2
s_y <- 1
time <- 2
tau <- t0 - time
semivario_s_t_adv <- beta1 * abs(s0_x - s_x - adv[1] * tau)^alpha1 +
                 beta1 * abs(s0_y - s_y - adv[2]*tau)^alpha1 +
                 beta2 * abs(tau)^alpha2
print(gamma[s_x, s_y, time] == 2 * semivario_s_t_adv)
```

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

```
semivario_s_t_adv <- beta1 * abs(s0_x - s_x - adv[1] * abs(tau))^alpha1 +
                 beta1 * abs(s0_y - s_y - adv[2]*abs(tau))^alpha1 +
                 beta2 * abs(tau)^alpha2
print(gamma[s_x, s_y, time] == 2 * semivario_s_t_adv)
```

```
## [1] FALSE
```

```
s_x <- 2
s_y <- 1
time <- 10
tau <- time - t0
semivario_s_t_adv <- beta1 * abs(s_x  - s0_x - adv[1] * tau)^alpha1 +
                 beta1 * abs(s_y - s0_y - adv[2]*tau)^alpha1 +
                 beta2 * abs(tau)^alpha2
print(gamma[s_x, s_y, time] == 2 * semivario_s_t_adv)
```

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

```
semivario_s_t_adv <- beta1 * abs(s_x - s0_x - adv[1] * abs(tau))^alpha1 +
                 beta1 * abs(s_y - s0_y - adv[2] * abs(tau))^alpha1 +
                 beta2 * abs(tau)^alpha2
print(gamma[s_x, s_y, time] == 2 * semivario_s_t_adv)
```

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

Problem: J'ai consideré seulement des tau positifs tels que tau=t-t0 avec $t \geq t0$. Or il faut faire correspondre dans le bon sens les s-s0 et t-t0. Il faut donc que mette s0-s et non s-s0 quand je fais l'optim. FAIT.