Debug negative log-likelihood

2024-09-04

Simulation

```r
true_param <- c(0.4, 0.2, 1.5, 1)
ngrid <- 5
spa <- 1:ngrid
nsites <- ngrid^2 # if the grid is squared
temp <- 1:300

# load the simulations
file_path <- paste0("../data/simulations_BR/sim_", ngrid^2, "s_",
                                length(temp), "t/br_",
                      ngrid^2, "s_", length(temp), "t_", 1, ".csv")
simu_df <- read.csv(file_path)

nsites <- ncol(simu_df)
sites_coords <- generate_grid_coords(sqrt(nsites))
dist_mat <- get_dist_mat(sites_coords,
                         latlon = FALSE) # distance matrix
df_dist <- reshape_distances(dist_mat) # reshape the distance matrix
```

Set of lags

```r
get_lag_vectors <- function(df_coords, params, hmax = NA, tau_vect = 1:10) {

  # Advection vector
  adv <- if (length(params) == 6) params[5:6] else c(0, 0)

  n <- nrow(df_coords)
  tau_len <- length(tau_vect)

  # Create index combinations
  indices <- combn(n, 2)
  i_vals <- indices[1, ]
  j_vals <- indices[2, ]

  # Calculate lags
  lag_latitudes <- df_coords$Latitude[j_vals] - df_coords$Latitude[i_vals]
  lag_longitudes <- df_coords$Longitude[j_vals] - df_coords$Longitude[i_vals]
  # Calculate hnorm for all pairs
  hnorms <- sqrt(lag_latitudes^2 + lag_longitudes^2)

  # Filter based on hmax
  if (!is.na(hmax)) {
    valid_indices <- which(hnorms <= hmax)
    i_vals <- i_vals[valid_indices]
    j_vals <- j_vals[valid_indices]
    lag_latitudes <- lag_latitudes[valid_indices]
    lag_longitudes <- lag_longitudes[valid_indices]
    hnorms <- hnorms[valid_indices]
  }

  # Replicate for tau_vect
  num_pairs <- length(i_vals)
  i_vals <- rep(i_vals, each = tau_len)
  j_vals <- rep(j_vals, each = tau_len)
  lag_latitudes <- rep(lag_latitudes, each = tau_len)
  lag_longitudes <- rep(lag_longitudes, each = tau_len)
  hnorms <- rep(hnorms, each = tau_len)
  taus <- rep(tau_vect, times = num_pairs)

  # Apply advection
  if (all(adv != c(0, 0))) {
    lag_latitudes <- lag_latitudes - adv[1] * taus
    lag_longitudes <- lag_longitudes - adv[2] * taus
    hnorms <- sqrt(lag_latitudes^2 + lag_longitudes^2)
  }

  # Create final dataframe
  lags <- data.frame(
    s1 = i_vals,
    s2 = j_vals,
    h1 = lag_latitudes,
    h2 = lag_longitudes,
    tau = taus,
    hnorm = hnorms
  )

  return(lags)
}
```

```r
sites_coords <- generate_grid_coords(sqrt(nsites))
df_lags <- get_lag_vectors(sites_coords, true_param,
                          hmax = sqrt(17), tau_vect = 0:10)

print(head(df_lags))
```

```
##   s1 s2 h1 h2 tau hnorm
## 1  1  2  0  1   0     1
## 2  1  2  0  1   1     1
## 3  1  2  0  1   2     1
## 4  1  2  0  1   3     1
## 5  1  2  0  1   4     1
## 6  1  2  0  1   5     1
```

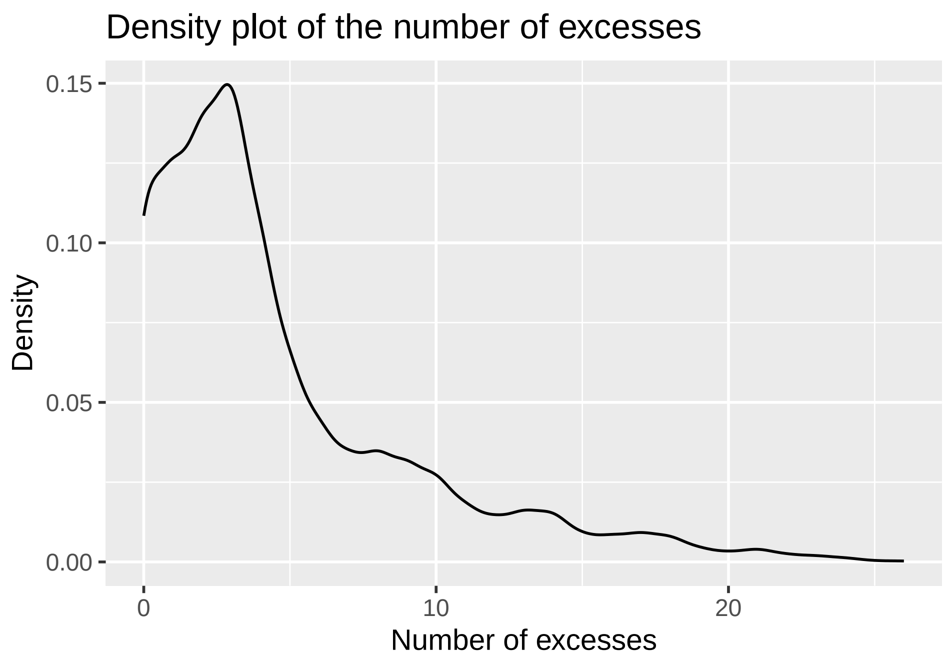
Optimisation

Get excesses

```r
empirical_excesses <- function(data_rain, quantile, df_lags) {
  excesses <- df_lags # copy the dataframe
  unique_tau <- unique(df_lags$tau) # unique temporal lags

  for (t in unique_tau) { # loop over temporal lags
    df_h_t <- df_lags[df_lags$tau == t, ] # get the dataframe for each lag

    for (i in seq_len(nrow(df_h_t))) { # loop over each pair of sites
      # get the indices of the sites
      ind_s2 <- as.numeric(as.character(df_h_t$s2[i]))
      ind_s1 <- df_h_t$s1[i]

      # get the data for the pair of sites
      rain_cp <- data_rain[, c(ind_s1, ind_s2), drop = FALSE]
      rain_cp <- as.data.frame(na.omit(rain_cp))
      colnames(rain_cp) <- c("s1", "s2")

      Tmax <- nrow(rain_cp) # number of time steps
      rain_unif <- cbind(rank(rain_cp$s1) / (Tmax + 1),
                         rank(rain_cp$s2) / (Tmax + 1))
      marginal_excesses <- sum(rain_unif[, 2] > quantile) # number of excesses

      rain_nolag <- rain_cp$s1[1:(Tmax - t)] # get the data without lag
      rain_lag <- rain_cp$s2[(1 + t):Tmax] # get the data with lag

      Tobs <- length(rain_nolag) # number of observations T - tau
      # transform the data in uniform data
      rain_unif <- cbind(rank(rain_nolag) / (Tobs + 1),
                         rank(rain_lag) / (Tobs + 1))
      # get the conditional excesses on s2
      cp_cond <- rain_unif[rain_unif[, 2] > quantile,]
      joint_excesses <- sum(cp_cond[, 1] > quantile) # number of excesses for s1
                                                      # given those of s2

      # store the number of excesses
      excesses$Tobs[excesses$s1 == ind_s1
                    & excesses$s2 == ind_s2
                    & excesses$tau == t] <- Tobs
      excesses$nj[excesses$s1 == ind_s1
                    & excesses$s2 == ind_s2
                    & excesses$tau == t] <- marginal_excesses
      excesses$kij[excesses$s1 == ind_s1
                  & excesses$s2 == ind_s2
                  & excesses$tau == t] <- joint_excesses
    }
  }
  return(excesses)
}

q <- 0.9
excesses <- empirical_excesses(simu_df, quantile = q, df_lags = df_lags)
print(head(excesses))
```

```
##   s1 s2 h1 h2 tau hnorm Tobs nj kij
## 1  1  2  0  1   0     1  300 30  15
## 2  1  2  0  1   1     1  299 30  14
## 3  1  2  0  1   2     1  298 30  13
## 4  1  2  0  1   3     1  297 30  13
## 5  1  2  0  1   4     1  296 30  13
## 6  1  2  0  1   5     1  295 30  11
```

```
# density plot of the number of excesses
ggplot(excesses, aes(x = kij)) +
  geom_density() +
  labs(title = "Density plot of the number of excesses",
       x = "Number of excesses", y = "Density")
```



Verif

```
# For a couple (s1, s2)
s1 <- 1
s2 <- 2
tau <- 2
```

```r
rain_cp <- simu_df[, c(s1, s2)]
rain_cp <- na.omit(rain_cp)
colnames(rain_cp) <- c("s1", "s2")
# quantile
q <- 0.9
rain_cp_q <- quantile(rain_cp$s2, probs = q)

Tmax <- nrow(rain_cp) # number of time steps == length(temp)

# get the number of marginal excesses
n_marg <- sum(rain_cp$s2 > rain_cp_q)
p_hat <- n_marg / Tmax # probability of marginal excesses

rain_lag <- rain_cp$s2[(1 + tau):Tmax] # get the data with lag
rain_nolag <- rain_cp$s1[1:(Tmax - tau)] # get the data without lag
# get the number of joint excesses
Tobs <- length(rain_nolag) # T - tau
rain_unif <- cbind(rank(rain_nolag) / (Tobs + 1),
                   rank(rain_lag) / (Tobs + 1))
# get the conditional excesses on s2
cp_cond <- rain_unif[rain_unif[, 2] > q,]
joint_excesses <- sum(cp_cond[, 1] > q) # number of excesses for s1

print(paste("Number of marginal excesses: ", n_marg))
```

```
## [1] "Number of marginal excesses:  30"
```

```r
print(paste("Number of joint excesses: ", joint_excesses))
```

```
## [1] "Number of joint excesses:  13"
```

Avec la fonction `empirical_excesses` on retrouve bien les mêmes résultats:

```r
q <- 0.9
excesses <- empirical_excesses(simu_df, quantile = q, df_lags = df_lags)
excesses_s1_s2 <- excesses[excesses$s1 == 1 & excesses$s2 == 2, ]
print(excesses_s1_s2)
```

```
##    s1 s2 h1 h2 tau hnorm Tobs nj kij
## 1   1  2  0  1   0     1  300 30  15
## 2   1  2  0  1   1     1  299 30  14
## 3   1  2  0  1   2     1  298 30  13
## 4   1  2  0  1   3     1  297 30  13
## 5   1  2  0  1   4     1  296 30  13
## 6   1  2  0  1   5     1  295 30  11
## 7   1  2  0  1   6     1  294 30  10
## 8   1  2  0  1   7     1  293 30   9
## 9   1  2  0  1   8     1  292 30   8
## 10  1  2  0  1   9     1  291 30   8
## 11  1  2  0  1  10     1  290 30   7
```

Pour un meme site s1 et s2, on doit avoir le meme nombre de dépassements marginale et conjoint sans décalage temporel. Prenons:

```
# For a couple (s1, s2)
s1 <- 1
s2 <- 1
tau <- 0
```

```
## [1] "Number of marginal excesses:  30"
```

```
## [1] "Number of joint excesses:  30"
```

Theorical chi

```
theorical_chi_ind <- function(params, h, tau) {
  # get variogram parameter
  beta1 <- params[1]
  beta2 <- params[2]
  alpha1 <- params[3]
  alpha2 <- params[4]
  # if (length(params) == 6) {
  #   hnorm <-
  # }
  # Get vario and chi for each lagtemp
  varioval <- 2 * (beta1 * h^alpha1 + beta2 * tau^alpha2)
  phi <- pnorm(sqrt(0.5 * varioval))
  chival <- 2 * (1 - phi)

  return(chival)
}


theorical_chi <- function(params, df_lags) {
  chi_df <- df_lags # copy the dataframe
  chi_df$chi <- theorical_chi_ind(params, df_lags$hnorm, df_lags$tau)
  return(chi_df)
}
```

```
chi_theorical <- theorical_chi(true_param, df_lags)
print(tail(chi_theorical))
```

```
##      s1 s2 h1 h2 tau hnorm       chi
## 2965 24 25  0  1   5     1 0.2367236
## 2966 24 25  0  1   6     1 0.2059032
## 2967 24 25  0  1   7     1 0.1797125
## 2968 24 25  0  1   8     1 0.1572992
## 2969 24 25  0  1   9     1 0.1380107
## 2970 24 25  0  1  10     1 0.1213353
```

Log likelihood

```r
neg_ll <- function(params, simu, df_lags, locations, quantile,
                   latlon = FALSE, simu_exp = FALSE, excesses = NULL) {
  hmax <- max(df_lags$hnorm)
  tau <- unique(df_lags$tau)

  # print(params)
  if (length(params) == 6) {
    adv <- params[5:6]
  } else {
    adv <- c(0, 0)
  }

  # Bounds for the parameters
  lower.bound <- c(1e-6, 1e-6, 1e-6, 1e-6)
  upper.bound <- c(Inf, Inf, 1.999, 1.999)
  if (length(params) == 6) {
    lower.bound <- c(lower.bound, -Inf, -Inf)
    upper.bound <- c(upper.bound, Inf, Inf)
  }

  # Check if the parameters are in the bounds
  if (any(params < lower.bound) || any(params > upper.bound)) {
    message("out of bounds")
    return(1e9)
  }

  if (!all(adv == c(0, 0))) { # if we have the advection parameters
    # then the lag vectors are different
    df_lags <- get_lag_vectors(locations, params, hmax = hmax, tau_vect = tau)
  }

  if (is.null(excesses)) {
    excesses <- empirical_excesses(simu, quantile, df_lags)
  }

  nj <- excesses$nj # number of marginal excesses
  Tobs <- excesses$Tobs # T - tau
  p <- nj[1] / nrow(simu) # probability of marginal excesses
  kij <- excesses$kij # number of joint excesses
  chi <- theorical_chi(params, df_lags) # get chi matrix
  # transform in chi vector
  chi_vect <- as.vector(chi$chi)
  chi_vect <- ifelse(chi_vect <= 0, 0.000001, chi_vect) # avoid log(0)

  non_excesses <- Tobs - kij # number of non-excesses
  # log-likelihood vector
  ll_vect <- kij * log(chi_vect) + non_excesses * log(1 - p * chi_vect)

  # final negative log-likelihood
  nll <- -sum(ll_vect, na.rm = TRUE)
  return(nll)
}
```

```
q <- 0.9
nll <- neg_ll(true_param, simu_df, df_lags, sites_coords, quantile = q)
print(nll)
```

```
## [1] 37860.6
```

```
nll <- neg_ll(true_param + 0.05, simu_df, df_lags, sites_coords, quantile = q)
print(nll)
```

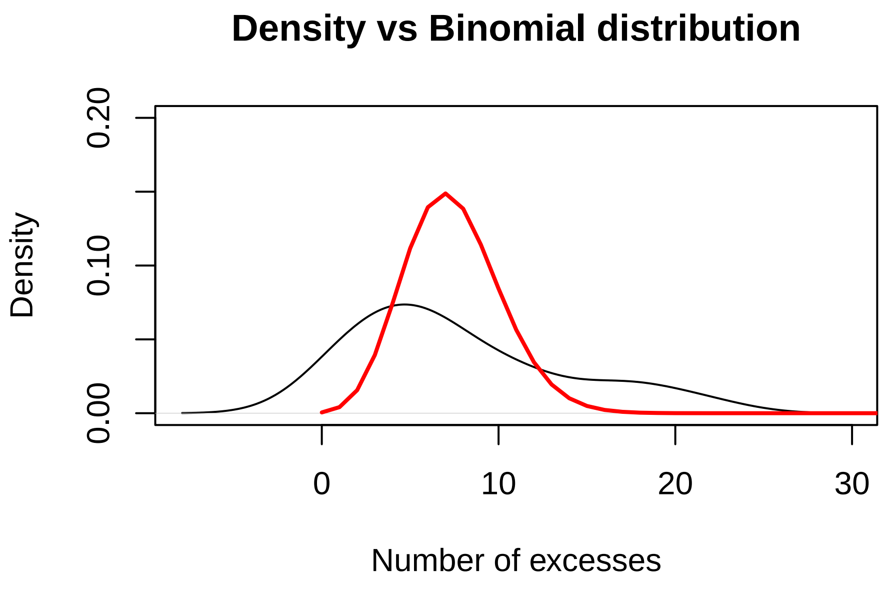
```
## [1] 39438.09
```

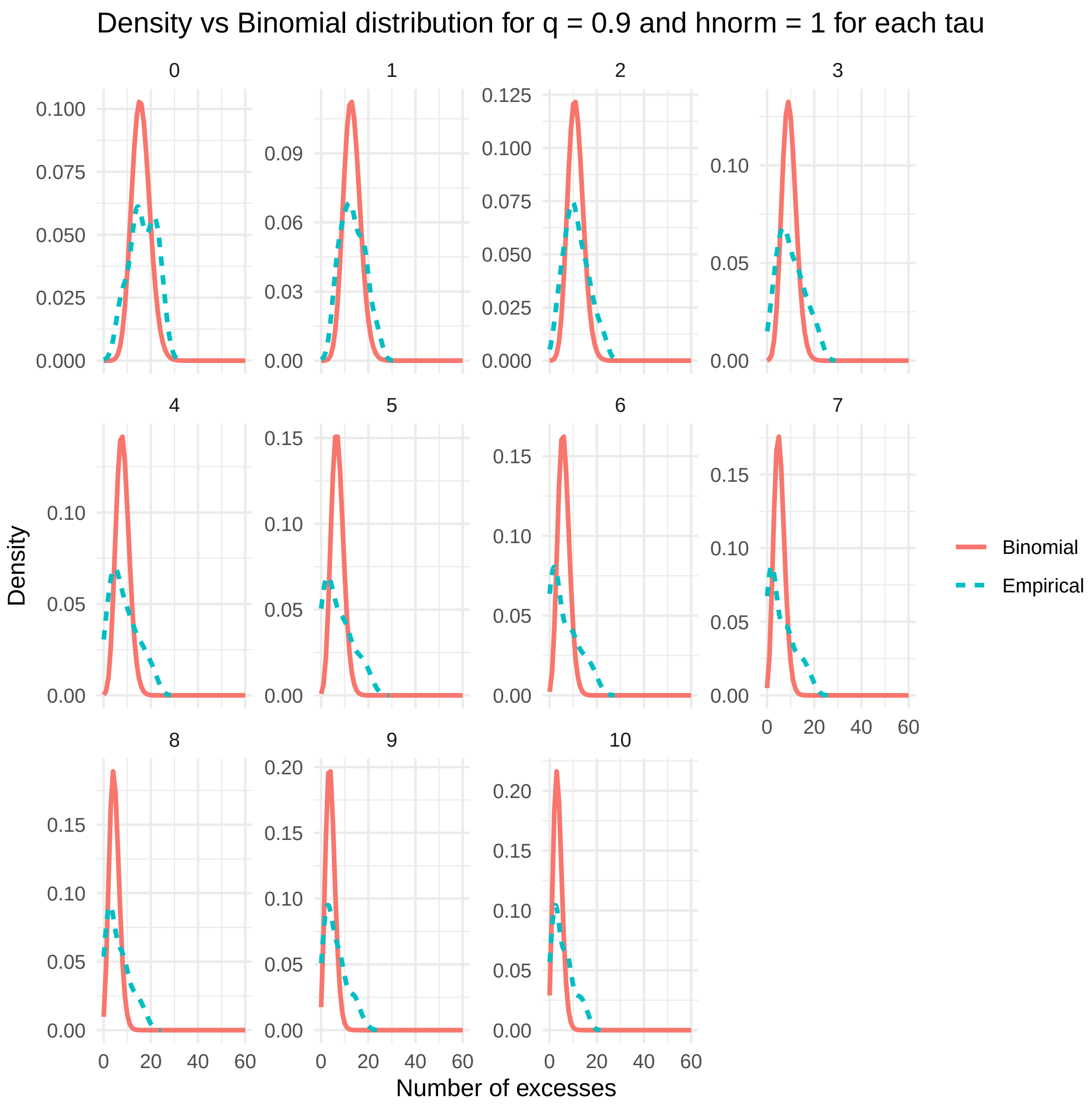
```
nll <- neg_ll(true_param - 0.1, simu_df, df_lags, sites_coords, quantile = q)
print(nll)
```

```
## [1] 39184.37
```

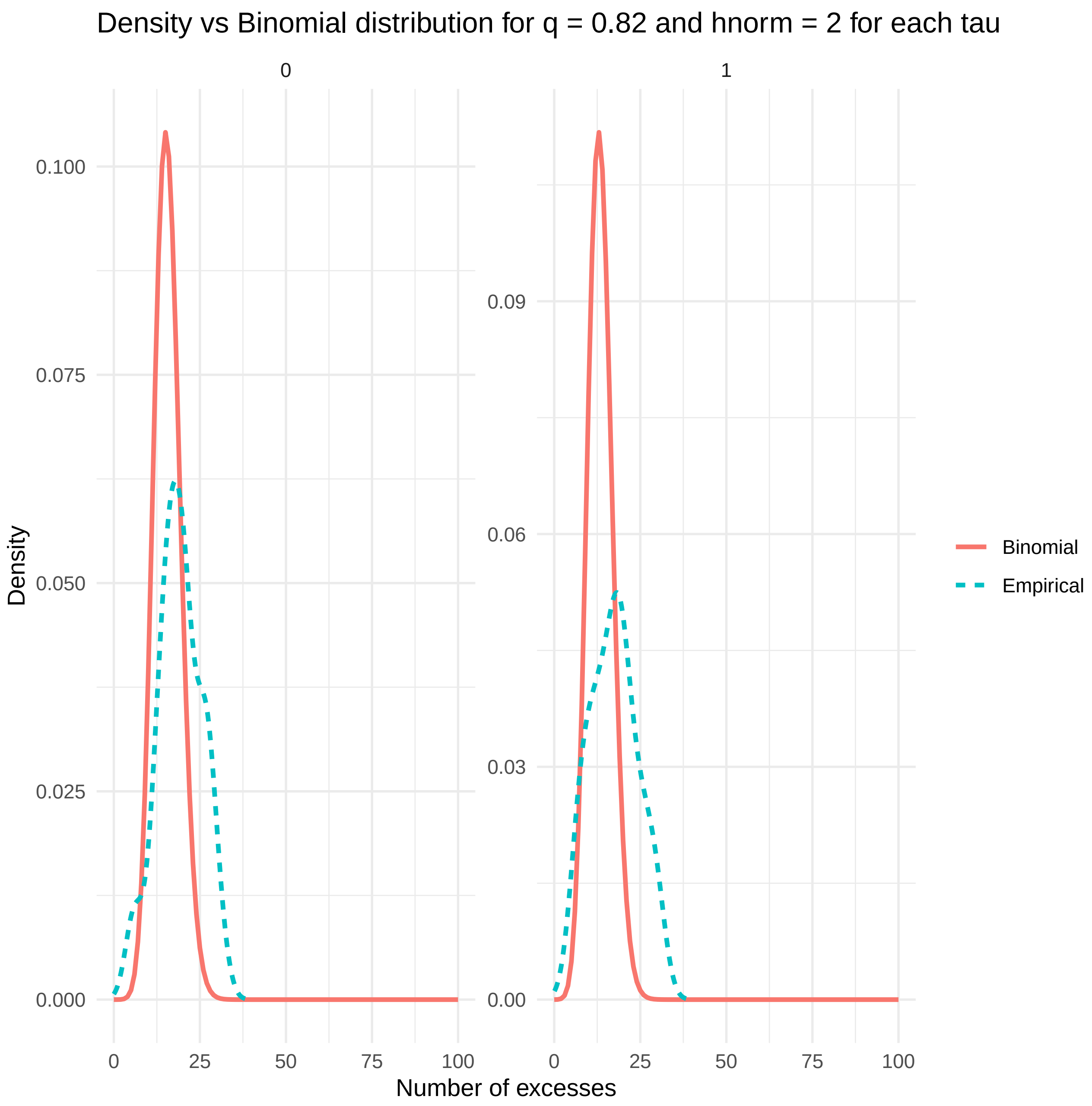
Verif optimisation : distribution des dépassements

```
q <- 0.9
excesses <- empirical_excesses(simu_df, quantile = q, df_lags = df_lags)

n_marg <- max(excesses$nj)
Tobs <- excesses$Tobs # T - tau
Tmax <- nrow(simu_df)
p_hat <- n_marg / Tmax # probability of marginal excesses
kij <- excesses$kij # number of joint excesses

chi_theorical <- theorical_chi(true_param, df_lags)

tau <- 1
hnorm <- 2
chi_tau_h <- chi_theorical$chi[chi_theorical$tau == tau &
                               chi_theorical$hnorm == hnorm]
k_tau_h <- excesses$kij[excesses$tau == tau &
                          excesses$hnorm == hnorm]
proba_tau_h <- unique(chi_tau_h * p_hat)
n <- Tmax - tau
Tobs_tau_h <- unique(Tobs[excesses$tau == tau &
                   excesses$hnorm == hnorm])
par(mfrow = c(1, 1))
x <- 0:Tobs_tau_h
# Density
plot(density(k_tau_h), main = "Density vs Binomial distribution",
     xlab = "Number of excesses", ylim = c(0, 0.2))
# binomial density
lines(x, dbinom(x, size = Tobs_tau_h, prob = proba_tau_h), col = "red", lwd = 2)
```


Density vs Binomial distribution
Density
0.00
0.10
0.20
0
10
20
30
Number of excesses



Density vs Binomial distribution for q = 0.9 and hnorm = 1 for each tau
0
1
2
3
4
5
6
7
8
9
10
Density
Number of excesses
Binomial
Empirical
0
20
40
60



Si je réduis le nombre de décalage temporel c'est beaucoup plus stable et j'ai de meilleures estimations.

```
tau1 <- 0:1 # 0:tmax
df_lags <- get_lag_vectors(sites_coords, true_param,
                           hmax = sqrt(17), tau_vect = tau1)

q <- 0.9
excesses <- empirical_excesses(simu_df, quantile = q, df_lags = df_lags)
true_param <- c(0.4, 0.2, 1.5, 1)
result_tau1 <- optim(par = c(true_param), fn = neg_ll,
                        simu = simu_df,
                        quantile = q,
                        df_lags = df_lags,
```

Table 1: RMSE for each parameter and different sets of temporal lags 0:tmax

tmax	rmse_beta1	rmse_beta2	rmse_alpha1	rmse_alpha2
1	0.0174320	0.0107351	0.0269555	0.0000000
2	0.0293007	0.0070692	0.0603444	0.1272188
4	0.0525518	0.0050158	0.1278235	0.1698772
6	0.0786232	0.0031435	0.2006046	0.1228936
8	0.0885474	0.0330338	0.2392121	0.2526097
10	0.0876895	0.0687326	0.2475781	0.3536453

```
                        excesses = excesses,
                        locations = sites_coords,
                        method = "CG",
                        control = list(parscale = c(1, 1, 1, 1),
                                       maxit = 10000))
# print(result_tau1$convergence) # 0 if it has converged
# print(result_tau1$par)
estim_tau1 <- result_tau1$par
rmse_tau1 <- sqrt((estim_tau1 - true_param)^2)
```

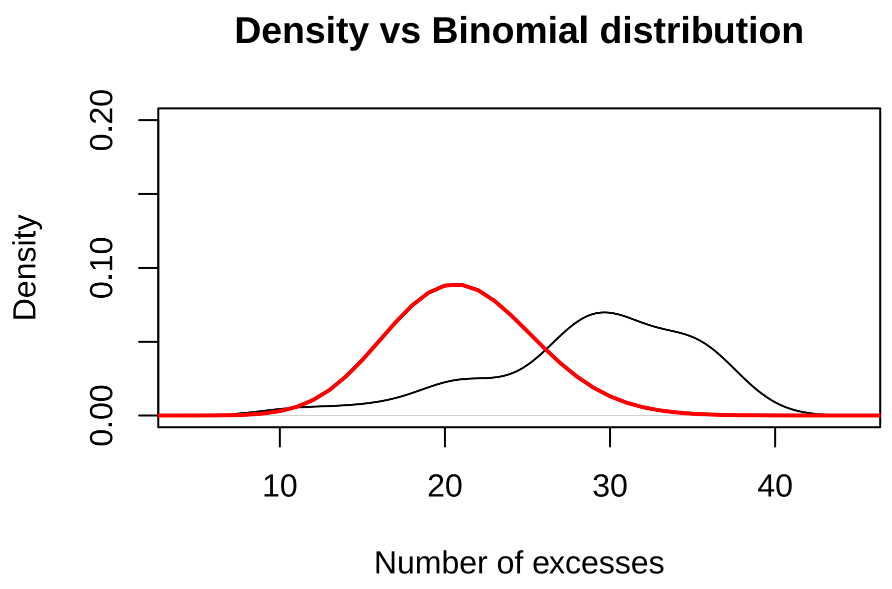
Other simulations

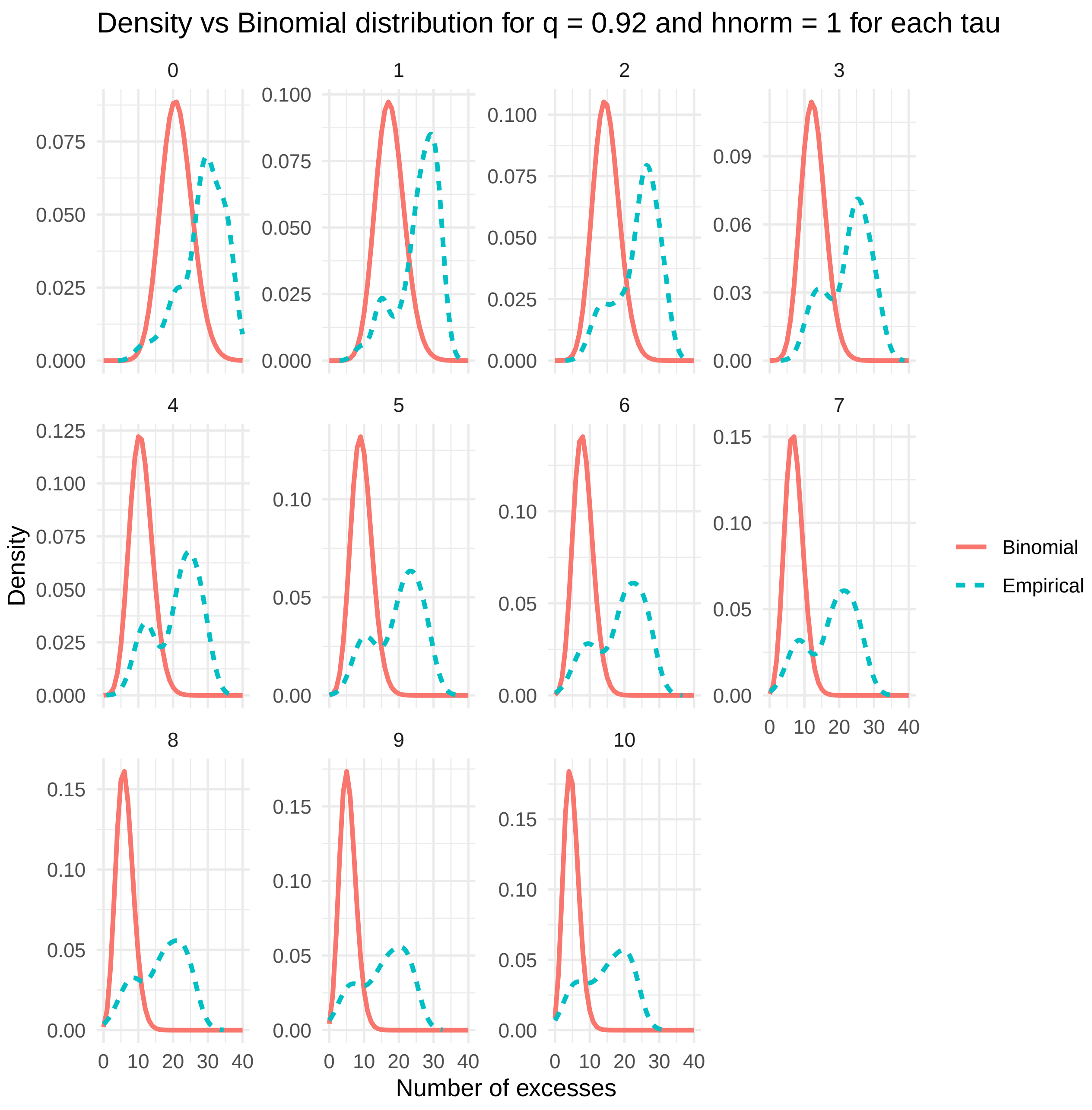
```
true_param <- c(0.4, 0.2, 1.5, 1)
ngrid <- 5
spa <- 1:ngrid
nsites <- ngrid^2 # if the grid is squared
temp <- 1:500

# load the simulations
file_path <- paste0("../data/simulations_BR/sim_", ngrid^2, "s_",
                                length(temp), "t/br_",
                      ngrid^2, "s_", length(temp), "t_", 1, ".csv")
simu_df <- read.csv(file_path)

nsites <- ncol(simu_df)
sites_coords <- generate_grid_coords(sqrt(nsites))
dist_mat <- get_dist_mat(sites_coords,
                         latlon = FALSE) # distance matrix
df_dist <- reshape_distances(dist_mat) # reshape the distance matrix
```

Pour tout quantile ce n'est pas binomial… il y a un problème.

```
q <- 0.92
df_lags <- get_lag_vectors(sites_coords, true_param,
                           hmax = sqrt(17), tau_vect = 0:10)
excesses <- empirical_excesses(simu_df, quantile = q, df_lags = df_lags)

n_marg <- max(excesses$nj)
```

```r
Tobs <- excesses$Tobs # T - tau
Tmax <- nrow(simu_df)
p_hat <- n_marg / Tmax # probability of marginal excesses
kij <- excesses$kij # number of joint excesses

chi_theorical <- theorical_chi(true_param, df_lags)

tau <- 0
hnorm <- 1
chi_tau_h <- chi_theorical$chi[chi_theorical$tau == tau &
                               chi_theorical$hnorm == hnorm]
k_tau_h <- excesses$kij[excesses$tau == tau &
                          excesses$hnorm == hnorm]
proba_tau_h <- unique(chi_tau_h * p_hat)
n <- Tmax - tau
Tobs_tau_h <- unique(Tobs[excesses$tau == tau &
                   excesses$hnorm == hnorm])
par(mfrow = c(1, 1))
x <- 0:Tobs
# Density
plot(density(k_tau_h), main = "Density vs Binomial distribution",
     xlab = "Number of excesses", ylim = c(0, 0.2))
# binomial density
lines(x, dbinom(x, size = Tobs_tau_h, prob = proba_tau_h), col = "red", lwd = 2)
```




Density vs Binomial distribution for q = 0.92 and hnorm = 1 for each tau
0
1
2
3
4
5
6
7
8
9
10
Density
Number of excesses
Binomial
Empirical