Extreme-Value Analysis of Wind Speed data with R-INLA

Author: Daniela Castro Camilo
Contact address: daniela.castro@kaust.edu.sa
Version: 1.0, March 2018 **Software requirement**: R (https://cran.r-project.org/)

1 Extreme-Value Theory

Extreme Value Theory emerges as the natural tool to assess probabilities of events that are rare, and whose occurrence involves great risk (Coles, 2001). Extreme rainfall, high and low temperatures, droughts, massive earthquakes, and stock market crashes are all extreme events whose consequences can be catastrophic. In the case of natural hazards, both observational data and computer climate models suggest that the frequencies and sizes of such events will increase in the future (Davison et al., 2012). Therefore, the statistical analysis of extreme events plays a fundamental role, since it can be used to quantify the changing probability of specific catastrophic events.

Figure 1 illustrate two types of extreme events that occurred recently. The right image shows Seine water levels during January 2018. When this picture was taken, more than 90mm of rain had falled, which is almost twice as much as normal, causing floods in several departments in France (center image). On the same month, Sidney was having the hottest day in 78 days (left image).

Figure 1: Examples of extreme events in the world during January 2018. Left: Seine high water levels. Center: Floods caused by heavy precipitation in France. Right: Heat waves in Sidney.

When modeling extreme events, we typically need to extrapolate beyond observed data, into the tail of the distribution, where the available data are often limited, and classical inference methods usually fail. The plausibility of the extrapolation is subject to certain stability conditions that constrain the class of possible distributions on which extrapolation should be based. In this tutorial, we will study a specific type of extreme events defined as *exceedances over high thresholds*. This approach is specially suitable when dealing with environmental data since it relates to the risk of observing a succession of large values over a period of time. Exceedances over high thresholds can reasonably be modeled only by a generalized Pareto distribution, which is the focus of our next section.

2 The Generalized Pareto distribution

As mentioned before, here we are interested in a particular type of extreme events: the so-called *threshold exceedances* i.e., values that exceed a certain high threshold u. A description of the stochastic behavior of

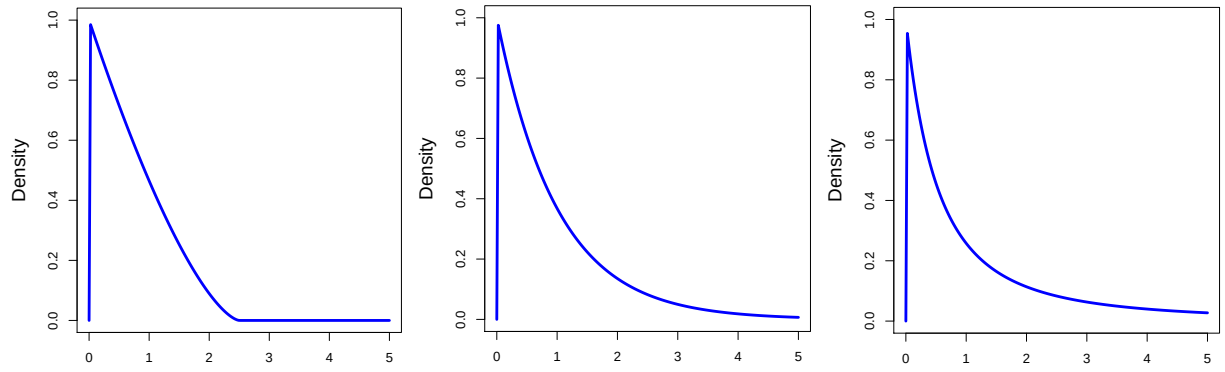


Figure 2: Generalized Pareto distributions for three different values of the shape parameter: -0.5 (left), 0 (center), and 5 (right).

these extreme events is given by the conditional probability

$$\Pr(Y > u + y | Y > u) = \frac{1 - F(u + y)}{1 - F(u)}, \quad y > 0,$$

where Y denote an arbitrary term in a sequence of i.i.d. variables having marginal distribution F. Under mild conditions, as u become large and the sample size n increases, it can be shown that

$$\Pr(Y > u + y | Y > u) \approx 1 - H_{GP}(y) = (1 + \xi y/\sigma_u)_+^{-1/\xi}, \quad \xi \neq 0,$$

where H_{GP} is the Generalized Pareto (GP) distribution with scale parameter $\sigma_u > 0$, shape parameter $\xi \in \mathbb{R}$, and $a_+ = \max\{0, a\}$. When $\xi = 0$, the above expression is interpreted as the limit $\xi \to 0$ and corresponds to the exponential survivor function $\exp(-y/\sigma_u)$, $y > 0$. When $\xi > 0$, the GP distribution has a power-law decay with infinite upper endpoint, and when $\xi < 0$, it has a finite and parameter-dependent upper endpoint $-\sigma/\xi$. The shape parameter ξ is dominant in determining the qualitative behavior of the GPD, as is shown in Figure 2.

Note that for sufficiently large u and n, we have that

$$\Pr(Y > u + y) = \zeta_u (1 + \xi y/\sigma_u)_+^{-1/\xi}, \quad \xi \neq 0,$$

with $\zeta_u = \Pr(Y > u)$. From this expression we can see that the GP distribution characterizes the intensity of exceedances, whereas ζ_u relates with the frequency of exceedances.

In practice, if extremes are defined as threshold exceedances in the i.i.d. setting, then the following framework for extreme modeling is suggested: extremes are identified by defining a high threshold u, for which the exceedances are $\{y_i : y_i > u, i = 1, \ldots, n\}$. If we label these exceedances as $y_{(1)}, \ldots, y_{(k)}$ with $k < n$, then threshold excesses are defined as $x_j = y_{(j)} - u$, $j = 1, \ldots, k$. By the previous results, $\{x_j\}_{j=1}^k$ may be regarded as independent realizations of a random variable whose distribution can be approximated by a member of the GP family. Inference is therefore performed by fitting the GP distribution to the observed threshold exceedances. Unfortunately, in many realistic applications, the i.i.d. setting cannot be assumed. This is mainly due to the presence of autocorrelation structures, or more general, some form of non-stationarity. To address these issues, in Section 4 we propose a three-stage Bayesian model for wind speeds assuming that there is a latent process that drives the extreme wind speeds and can be well described by suitable regression equations. But before we move to the application section, let's *warm-up* with a small simulated example.

3 A simulated example

We start by loading the R-INLA package and the simulated data:

```
library(INLA)
```

```
## Loading required package: sp
```

```
## Loading required package: Matrix
```

```
## This is INLA_18.01.26 built 2018-01-26 15:08:02 UTC.
## See www.r-inla.org/contact-us for how to get help.
```

```
DATA <- paste0(getwd(), "/Data/") # data directory
load(paste0(DATA,"sim_data.Rdata"))
```

We can assume, for instance, that $\{y_i\}$ correspond to precipitation (in mm) in a given area, while $\{x_i\}$ is a set of covariates of interest. The goal is to fit a GP distribution to $\{y_i\}$ assuming some functional relation with $\{x_i\}$. Let's assume that $y_i \sim \text{GP}(\xi, q_{\alpha,i})$ where $q_{\alpha,i} = \exp(1 + x_i)$ is the α-quantile of the GP distribution. Here we re-parametrized the GP distribution in term of the its $\alpha-$quantile is to avoid confounding problems due to the correlation between estimated GP parameters.

To run this model in `R-INLA` we first specify the formula, and then we call the `inla` function:

```
form = y ~ 1+x
alpha = 0.99
options(warn=-1)
fit1 = inla(form, data = sim_data,
         family = "gp",
         control.family = list(control.link = list(quantile = alpha)),
         control.predictor = list(compute=TRUE),
         verbose=FALSE)
```

By default, minimally informative priors are specified in the hyperparameter ξ. Different priors can be specified through the option hyper in the formula specification, for instance

```
fit2 = inla(form, data = sim_data,
         family = "gp",
         control.family = list(control.link = list(quantile = alpha),
                               hyper = list(theta = list(prior = "loggamma",
                                                         param = c(2, 2),
                                                         initial = log(2)))),
         control.predictor = list(compute=TRUE),
         verbose=FALSE)
```

We can take a look at the results using the `summary` function:

```
summary(fit1)
```

```
##
## Call:
## c("inla(formula = form, family = \"gp\", data = sim_data, verbose = FALSE, ",  "    control.predicto
##
## Time used:
##  Pre-processing    Running inla Post-processing           Total
##          1.7413          0.3676          0.2760          2.3850
##
## Fixed effects:
##               mean     sd 0.025quant 0.5quant 0.975quant   mode   kld
## (Intercept) 0.4048 0.5249    -0.6102   0.3996     1.4493 0.3889 1e-04
## x           1.1321 0.1266     0.8831   1.1322     1.3803 1.1325 0e+00
##
```

```
## The model has no random effects
##
## Model hyperparameters:
##                                                    mean     sd 0.025quant
## Shape parameter for the genPareto observations 0.2495 0.0601     0.1408
##                                                0.5quant 0.975quant   mode
## Shape parameter for the genPareto observations   0.2463      0.376 0.2391
##
## Expected number of effective parameters(std dev): 2.001(1e-04)
## Number of equivalent replicates : 149.91
##
## Marginal log-Likelihood:  -1210.47
## Posterior marginals for linear predictor and fitted values computed
```

4 Application to wind speed data

4.1 Data and motivation

Our data consist of hourly measurements of wind speed and wind direction over 3 years (2012-2014) in 20 towers along the border between Oregon and Washington. The Bonneville Power Administration maintains and stores these data, which are publicly available at http://transmission.bpa.gov/Business/Operations-/Wind/MetData.aspx. Figure 3 maps the location of the 20 towers labelled by acronym. Wind turbines located in each one of the 20 stations are able to produce wind power from wind speed according to the GE 1.5 MW power curve in Figure 4.

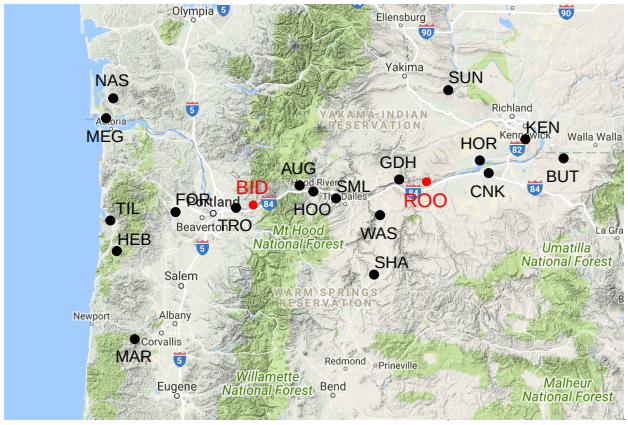


Figure 3: Map of the location of the 20 towers along the Columbia River.

Due to the intermittent and unstable nature of wind, there are significant challenges associated with the use of wind energy. One of such challenges is to obtain highly accurate short-term wind speed (and therefore wind power) forecasts. To do so we need to consider the following characteristics:

1. Winds are correlated both in time and space.
2. Wind patterns tend to show *persistence* (tendency to maintain its current state).
3. Wind speeds are highly non-stationary, usually exhibiting seasonal patterns of different magnitudes.

Our main goal is to propose a model able to produce short-term extreme and non-extreme wind speed forecasts. To do that we need a model that is able to describe the dynamic nature of the wind speeds, including all the characteristics mentioned before.

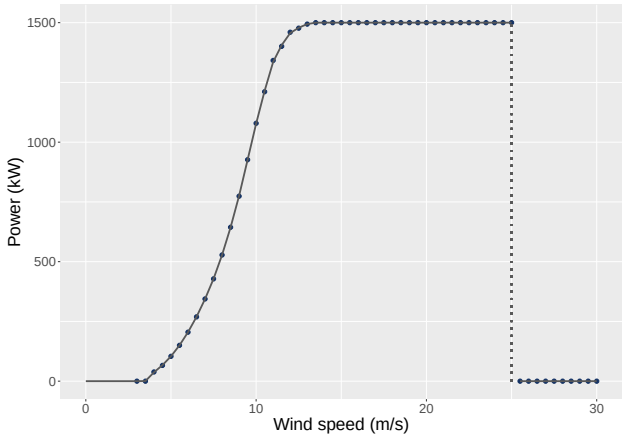


Figure 4: The GE 1.5 MW power curve. Dots correspond to the manufacturer data, and the solid line is a non-parametric fit to those data (cubic spline). The turbines have a cut-in speed of 3.5 m/s and a cut-off speed of 25 m/s.

To simplify our analysis, in this tutorial we will construct a model for wind speed only at location BID (Biddle Butte), therefore leaving aside the spatial component of wind speeds.

4.2 Exploratory analysis

As we know, an exploratory analysis is always needed before we can make any modeling assumptions. Let's start by loading the data and the require packages:

```
library(INLA)
library(ggplot2)
```

```
DATA <- paste0(getwd(), "/Data/") # data directory
load(paste0(DATA,"Wind_data_BID.Rdata"))
```

Let's take a look at the structure of the data:

```
head(data)
```

```
##   month day year hr    Speed      Dir
## 1     1   1 2012  9 17.24035 86.99180
## 2     1   1 2012 10 16.39685 85.29987
## 3     1   1 2012 11 17.26713 86.45008
## 4     1   1 2012 12 17.25374 84.91693
## 5     1   1 2012 13 17.16894 84.75838
## 6     1   1 2012 14 17.70449 85.09190
```

We can use ggplot to produce an enhanced plot of our data. First, let's define some nice colors (this is purely cosmetic and can be avoided):

```
myjblue = rgb(20/256, 50/256, 100/256)
my.colors = c("#F0B1AEB3", "#A9A06EB3", "#8DD198B3", "#8DD3D7B3", "#A9C2F4B3",
              "#E19FD8B3", "#595959B3")
my.colors.ext = rep(my.colors, 6)
```

To index each observation with time, we need to create a vector of dates in POSIXlt format

```
tmp = paste(paste(data$month, data$day, data$year, sep = "/"), data$hr, sep = " ")
dates = strptime(tmp, "%m/%d/%Y %H", tz = "US/Pacific")
data. = data.frame(Date = dates, Speed = data$Speed)
```

Now we can produce the plot:

```
gg <- ggplot(data = data., aes(data.$Date, data.$Speed))
gg <- gg + geom_line(col = myjblue, lwd = .1)
gg <- gg  + labs(x = "", y = "Speed") + ylim(c(0,30))
gg <- gg + ggtitle('Biddle Butte')
gg <- gg + theme(plot.title = element_text(size = 25, hjust = 0.5))
gg <- gg + theme(axis.text.x = element_text(size = 15),
                 axis.text.y = element_text(size = 15))
gg <- gg + theme(axis.text=element_text(size=12),axis.title=element_text(size=20))
print(gg)
```

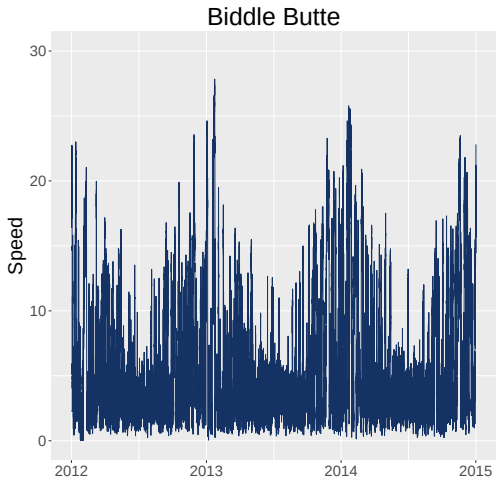
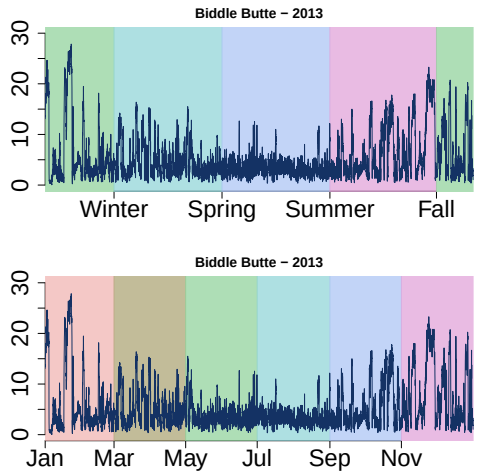


Figure 5: Wind speed at Biddle Butte.

As we can see, the data is non-stationary, with clear seasonal patterns. To have a better idea of the kind of seasonality we are dealing with, let's look at the data over one year, say 2013, by seasons and every two months:



Finally, let's take a look at the distribution of wind speeds:

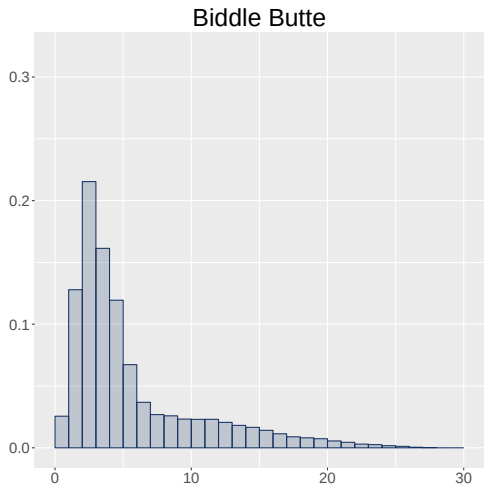


Figure 6: Histogram of wind speeds at Biddle Butte.

4.3 Modeling

4.3.1 Likelihood

Our strategy for modeling the temporal patterns observed at Biddle Butte can be decomposed into three stages, each consisting of a suitable univariate response distribution combined with a regression equation (latent Gaussian model) that captures systematic temporal effects.

Stage 1: Weibull distribution

We assume that (positive) wind speeds $Y(t)$ can be characterized by a Weibull distribution with time-varying scale parameter $\lambda(t)$ and fixed shape parameter κ:

$$h_{\mathrm{W}}(y) = \frac{\kappa}{\lambda(t)} \left[\frac{y}{\lambda(t)} \right]^{\kappa-1} \exp\{-[y/\lambda(t)]^{\kappa}\}, \quad y \geq 0.$$

This is a suitable model to describe the distribution of non-extreme wind speeds, and it can be used to define extremes. To see this, note that the threshold u after which observations are consider to be extreme can be changing in time due the the non-stationary nature of wind speeds. Therefore, a sensible approach is to select $u = u(t)$ by fitting a quantile regression to the 95%-quantile of the Weibull distribution. Figure **??** illustrates this idea. Extreme wind speeds are then defined as those which exceed the high time-varying threshold $u(t)$.

Stage 2: Bernoulli distribution

As we know from Section, to properly describe the distribution of extreme wind speeds we need to estimate the *exceedance probability*. At time t, an exceedance might or might not occur, therefore we model exceedance indicators $I(t) = 1\{Y(t) > u(t)\}$ as Bernoulli variables:

$$h_{\mathrm{B}}(y) = \mathrm{p}[u(t)]^{y}(1 - \mathrm{p}[u(t)])^{1-y}, \quad y \in \{0, 1\},$$

where $\mathrm{p}[u(t)] = \Pr\{Y(t) > u(t)\}$.

Stage 3: Generalized Pareto distribution

Threshold exceedances defined as $X(t) = [Y(t) - u(t)]_{+}$, are characterized by a GP distribution.

4.3.2 Latent Gaussian models

The exploratory analysis in Section 4.2 showed two possible sources of non-stationarity in time: one in terms of seasons (winter, spring, summer, fall) and another one in terms of seasonal patterns every two months. We could then consider a latent model that includes a smooth seasonal or bimonthly random effect. Another source of non-stationarity in time is the one given within a day: wind speeds tend to behave differently at morning, afternoon, evening, and night time. Therefore, our latent model can also includes an intra-daily effect (hourly, for example). With this in mind, we define two possible linear predictors:

$$\eta_1(t) = \mu + f_1[w_1(t)] + f_2[w_2(t)], \quad t = 1, \ldots, T,$$
$$\eta_2(t) = \mu + f_1[w_1(t)] + f_3[w_3(t)], \quad t = 1, \ldots, T,$$

where μ is a fixed intercept and $f_1[w_1(t)]$, $f_2[w_2(t)]$, and $f_3[w_3(t)]$ describe different sources of non-stationary. Specifically, $f_1[w_1(t)]$ is a temporal random effect defined on a hourly basis and cyclic with a daily period (this is the intra-daily effect), $f_2[w_2(t)]$ is a temporal random effect defined on a seasonal basis and cyclic with a yearly period, and $f_3[w_3(t)]$ is a temporal random effect defined on a bimonthly basis and cyclic with a yearly period. The cyclic part in $f_1[w_1(t)]$ means that we assume that the effect at 9 am on day 1 is the same as the one at 9 am on day 2. Similar arguments apply for $f_2[w_2(t)]$ and $f_3[w_3(t)]$. We choose every $f_i[w_i(t)]$ to be a Gaussian random walk of order 2 with precision parameters $\tau_i > 0$, $i = 1, 2, 3$.

4.3.3 Priors

Our model have two hyperparameters controlling the likelihood, namely κ and ξ, and three hyperparameters controlling the latent Gaussian model, namely $\tau_i > 0$, $i = 1, 2, 3$.. For simplicity, we use the minimally informative priors specified by default, which are:

- κ ~Gamma(25,25).
- ξ ~Gamma(1,15).
- τ_i ~Gamma(1,5 $\times$ 10^{-5}).

Note that all these hyperparameters are estimated within the model.

4.3.4 Running our three-stage Bayesian model in R-INLA

Here we will describe the coding of our three-stage model using the first linear predictor, η_1. A similar implementation can be used for the second linear predictor. We start by defining the response variable, the temporal random effects, and the initial precisions τ_i, $i = 1, 2$. This part is common to each stage of our model:

```
y.wind = data$Speed # response vector
#- Defining seasonal effect
# 1 = winter; 2 = spring; 3 = summer; 4 = fall
data$season = rep(NA, nrow(data))
for(i in 1:nrow(data)){
  if(data$month[i] == 12 || data$month[i] == 1 || data$month[i] == 2) data$season[i] = 1
  if(data$month[i] == 3 || data$month[i] == 4 || data$month[i] == 5) data$season[i] = 2
  if(data$month[i] == 6 || data$month[i] == 7 || data$month[i] == 8) data$season[i] = 3
  if(data$month[i] == 9 || data$month[i] == 10 || data$month[i] == 11) data$season[i] = 4
}
idx.seasons = data$season

#- Defining hourly effect
hours = dates$hour #- Extract all hours
idx.hours = inla.group(hours, method = "cut", n = 24) #- Subdivide the days into hours
```

```
#- Initial precisions for temporal effects
initialsd.hour = initialsd.season = 0.025
initialprec.hours = 1/initialsd.hour^2
initialprec.seasons = 1/initialsd.season^2
```

The next common step is to define the `inla` formula:

```
#- INLA formula
form = y ~ -1 + intercept +
  f(id.season, model = 'rw2', cyclic = TRUE,
    hyper = list(prec = list(initial = log(initialsd.season), fixed = FALSE)),
    constr = TRUE) +
  f(id.hour, model = 'rw2', cyclic = TRUE,
    hyper = list(prec = list(initial = log(initialprec.hours), fixed = TRUE)),
    constr = TRUE)
```

In the following we describe the `inla` implementation specific for each one of the three stages.

Stage 1: Weibull distribution

```
#- Weibull fit with quantile regression
fit.qr <- inla(form, data=data.wei, family ="weibull",
               control.family = list(variant = 1,control.link = list(quantile = 0.95,
                                                                      model = "quantile")),
               control.inla = list(int.strategy = "ccd"),
               control.predictor = list(compute = TRUE, link = 1),
               verbose = verbose)
#- Obtain time-varying threhsold
thr.wei = exp(fit.qr$summary.linear.predictor$mean)
```

Stage 2: Bernoulli distribution

```
#- Exceeds above threshold
y.exc = as.numeric(y.wind)
for (j in 1:length(y.exc)){
  thr.j = thr.wei[j]
  y.exc[j] = ifelse(y.exc[j] > thr.j, (y.exc[j] - thr.j), NA)
}
#- Define Bernoulli vector y.pexc with TRUE if an exceedance occurs
y.pexc = logical(length(y.exc))
y.pexc[!is.na(y.exc)] = TRUE; y.pexc[is.na(y.exc)] = FALSE
# INLA data
data.ber = data.wei
data.ber$y = as.integer(y.pexc)
data.ber$ntrials = rep(1, length(y.pexc))

#- Bernoulli fit
fit.ber <- inla(form, data = data.ber, family = "binomial",
                Ntrials = data.ber$ntrials,
                control.inla = list(int.strategy = "ccd"),
                control.predictor = list(compute = TRUE, link = 1),
                verbose = verbose)
```

Stage 3: Generalized Pareto distribution

```
#- Filter NAs
is.na.exc = is.na(y.exc)
```

```
y.exc = y.exc[!is.na.exc]

#- INLA data
data.gpd = data.wei[!is.na.exc, ]
data.gpd$y = y.exc
#- GPD fit
fit.gp <- inla(form, data = data.gpd, family = "gp",
               control.family = list(control.link = list(quantile = 0.5)),
               control.inla = list(int.strategy = "eb"),
               control.predictor = list(compute = TRUE, link = 1),
               verbose = verbose)
```

4.3.5 Results

Figure 8 shows posterior means and 95% pointwise credible intervals estimated for the temporal random effects in each of the three stages. Overall, the uncertainty in the Weibull model is much lower than the Bernoulli and GP models, as it uses more information. The hourly random effects seems significantly different from zero for the Weibull model, while the GP model seems close to stationary when taking the uncertainty into account. For the Weibull model, we can see that wind speeds tend to be higher between 6am and 6pm, whereas for the Bernoulli and GP model we can see a smooth transition between moderate and high wind speeds before and after noon, respectively. The seasonal random effect is less smooth (in part, because we only have 4 effects to estimate) and overall seems to indicate that wind speeds are stronger during winter and fall. Figure **??** show the same plot for the second linear predictor, where similar conclusions can be drawn.

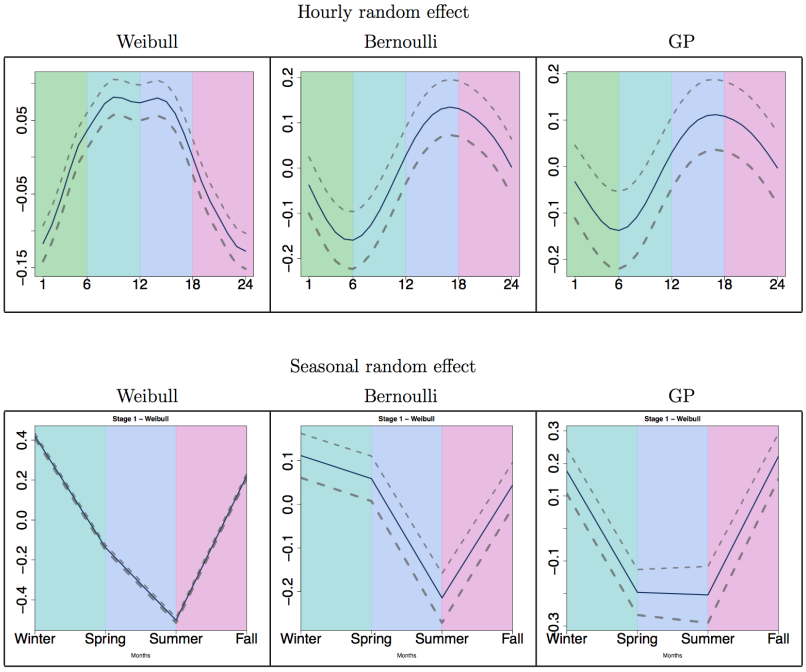


Figure 7: Temporal random effects displayed for the three stages (Weibull, Bernoulli, GP) using the first linear predictor. Curves show the posterior mean and 95 pointwise credible intervals.

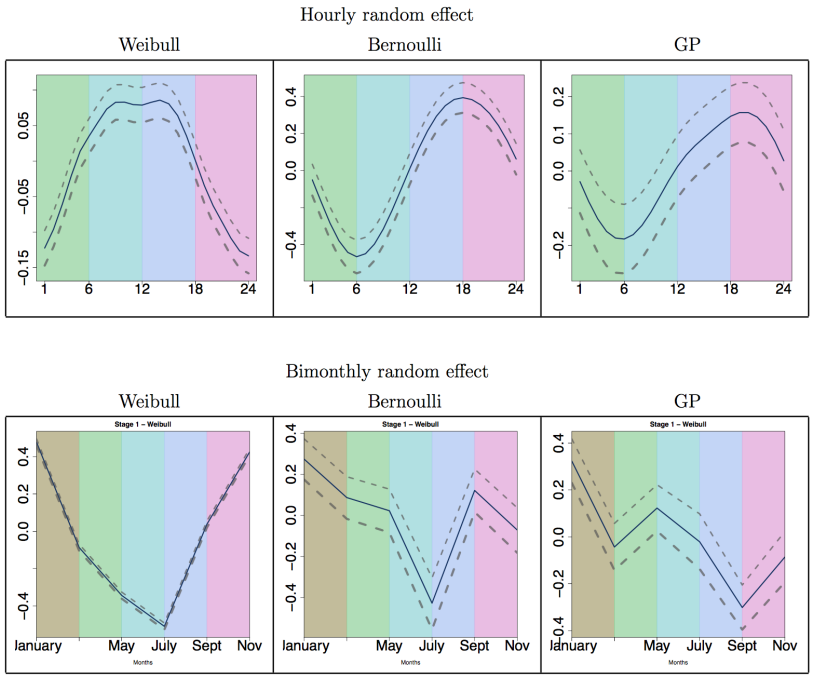


Figure 8: Temporal random effects displayed for the three stages (Weibull, Bernoulli, GP) using the second linear predictor. Curves show the posterior mean and 95 pointwise credible intervals.

4.3.6 Remarks

Here we develop a flexible temporal hierarchical model belonging to the wide class of latent Gaussian models. Our model can handle non-extreme and extreme observations at the same time, and is able to incorporate different sources of non-stationarity observed in the data. Model parameters are quickly estimated taking advantage of the very powerful and efficient `R-INLA` software.

To see extensions of this model and applications to short-term forecasts of wind speeds, please come to my talk on Wednesday, oral session number 4 :).