

Untitled

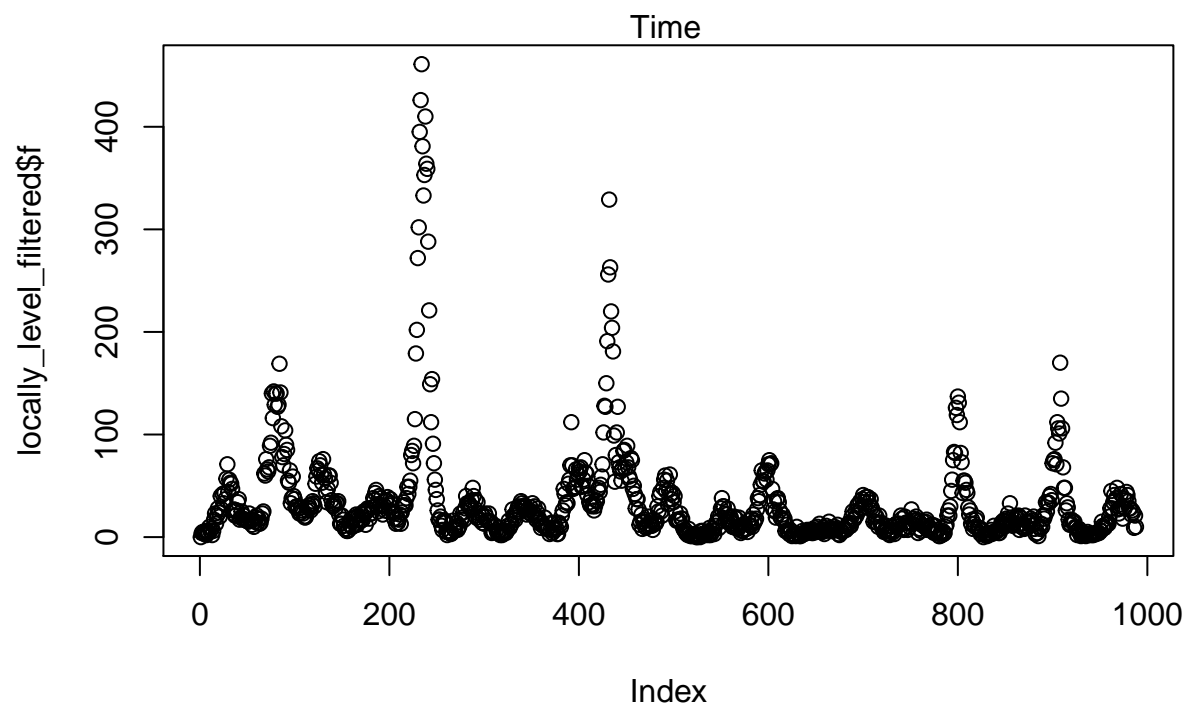
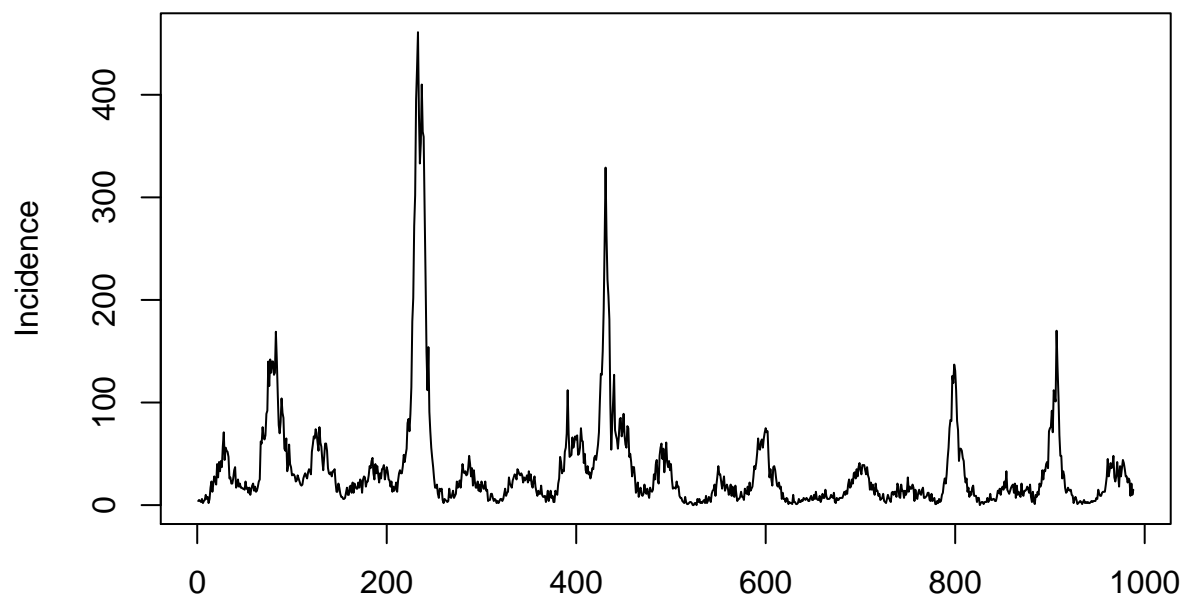
November 30, 2017

Introduction to Dengue

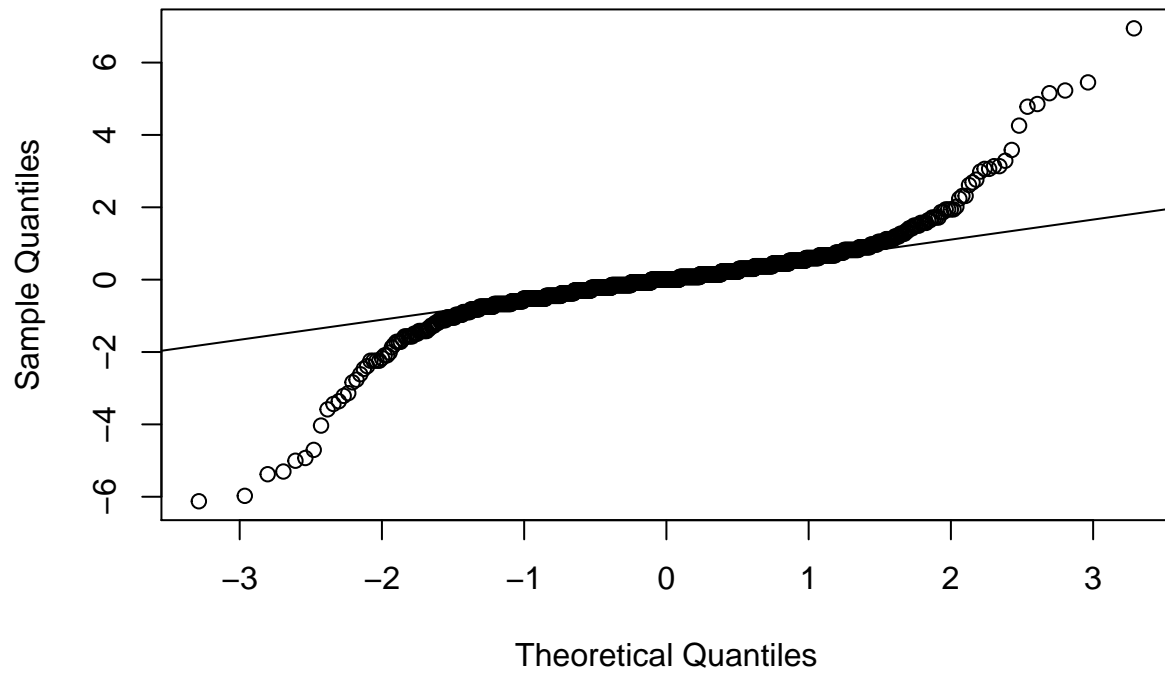
Introduction: Incidence of Dengue follows seasonal transmission patterns, and tend to have outbreaks every few years. Since these epidemics are currently hard to predict and they affect to the population in areas where dengue is endemic. Early recognition and prompt treatment of severe cases can substantially lower the risk of medical complications and death. Accurate forecasts of cases of infected individuals, or incidence, are key to planning and resource allocation. For example, knowing well in advance the numbers of cases that are expected and when they will occur allows preparation via education and community mobilization campaigns, reallocation of resources (people, insecticide, diagnostic reagents) to high-risk areas, or re-training of physicians to recognize symptoms and to treat appropriately (Kuhn et al., 2005; Degallier et al., 2010; Thomson et al., 2008) in advance of peak transmission. Our goal is to develop an optimal dlm model to predict the epidemics of dengue using only data from time periods prior to the historical dengue seasons.

Data: The provided data include weekly dengue incidence and linked environmental variables, and the training and testing sets may be downloaded from <http://dengueforecasting.noaa.gov/> The dengue incidence portion is comprised of historical surveillance data at Iquitos, Peru and San Juan, Puerto Rico, summarized weekly. Cases in the data set include laboratory-confirmed and serotype-specific cases. The data are reflecting the total number of cases in each week, possibly revised or estimated ex post. A breakdown of incidence into strata of four serotypes, with a fifth un-serotyped category, were also provided. However, we only trained on total_cases in the data file. #Basic attempts at modelling

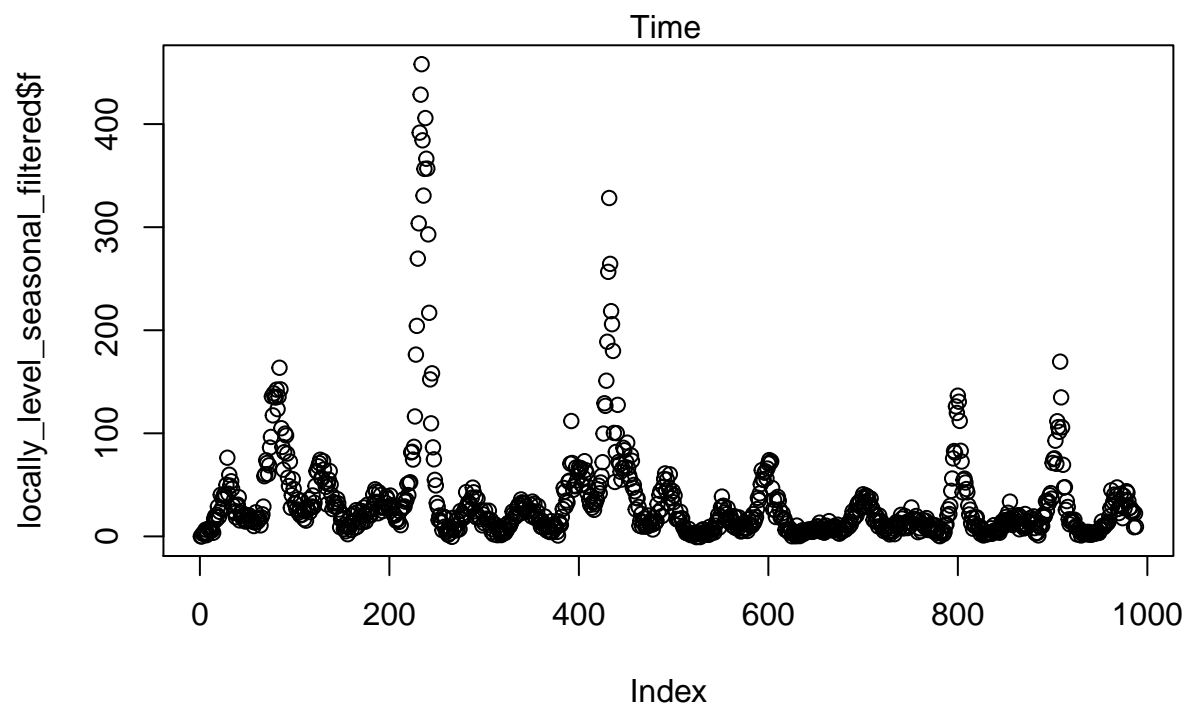
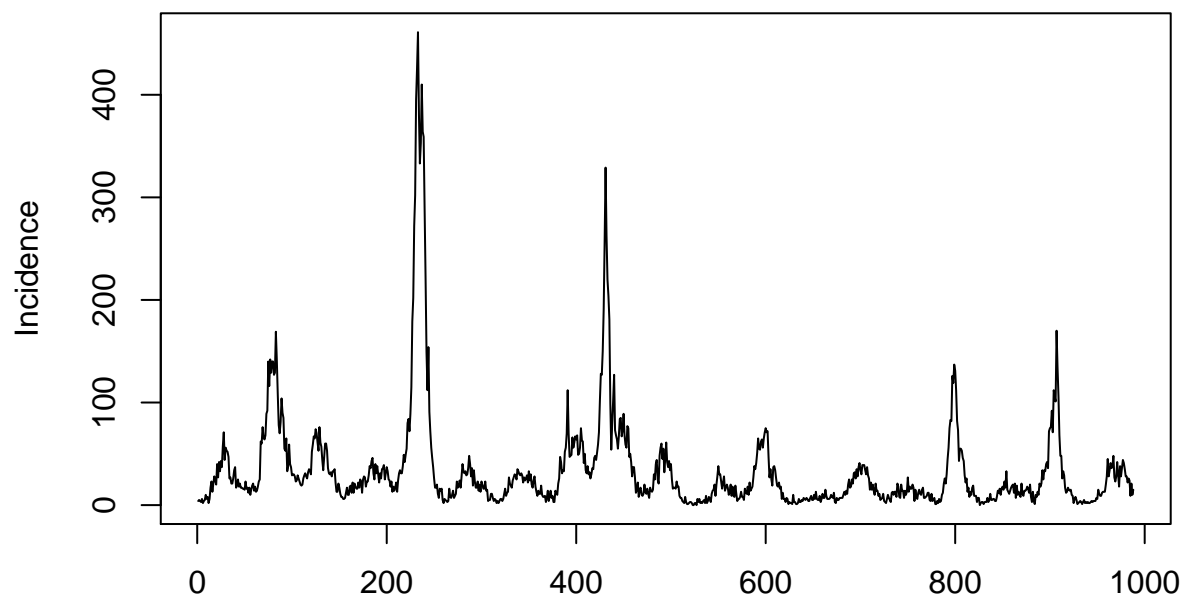
Locally-level



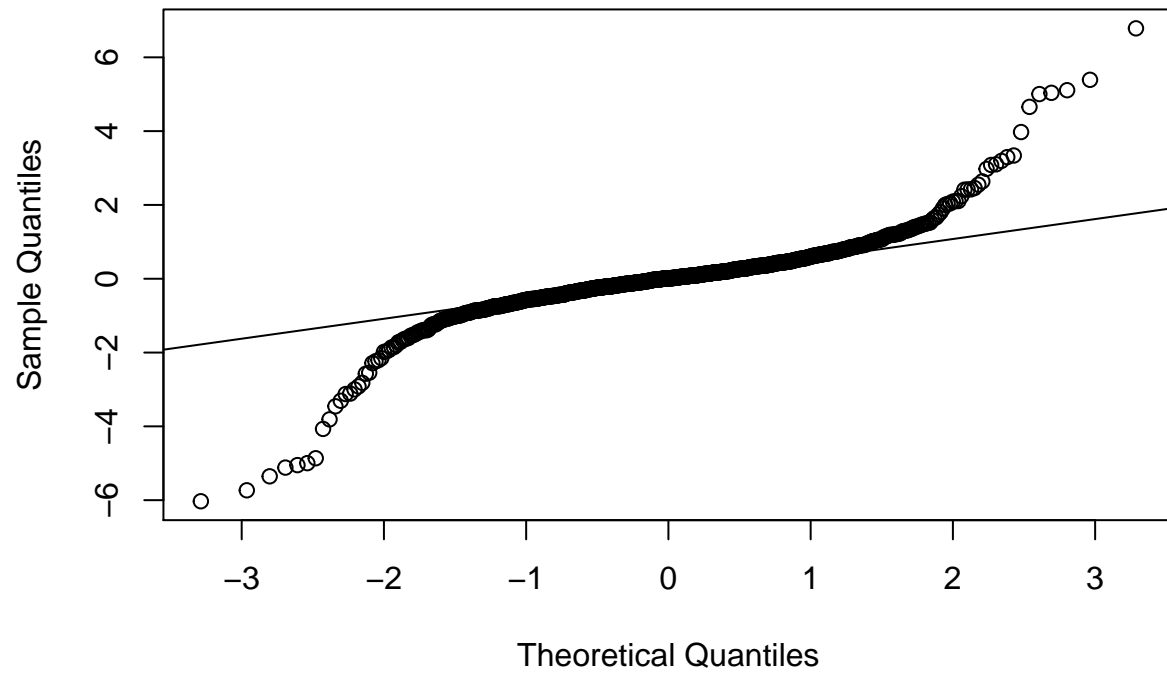
Normal Q-Q Plot



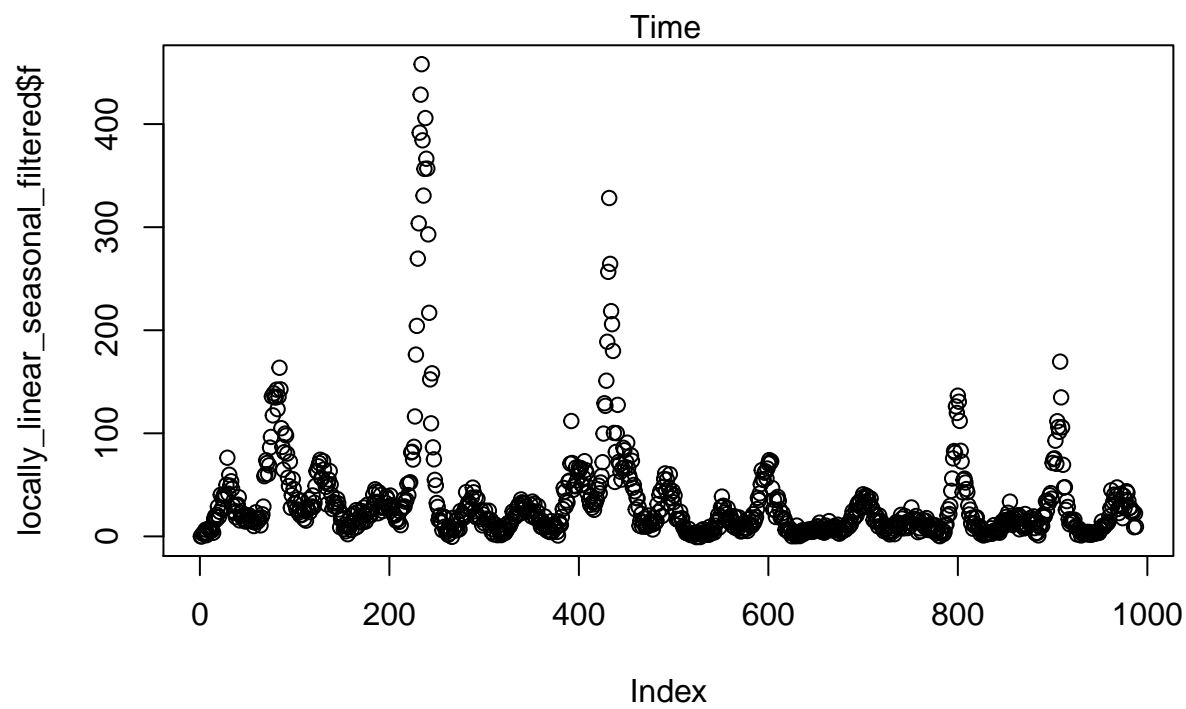
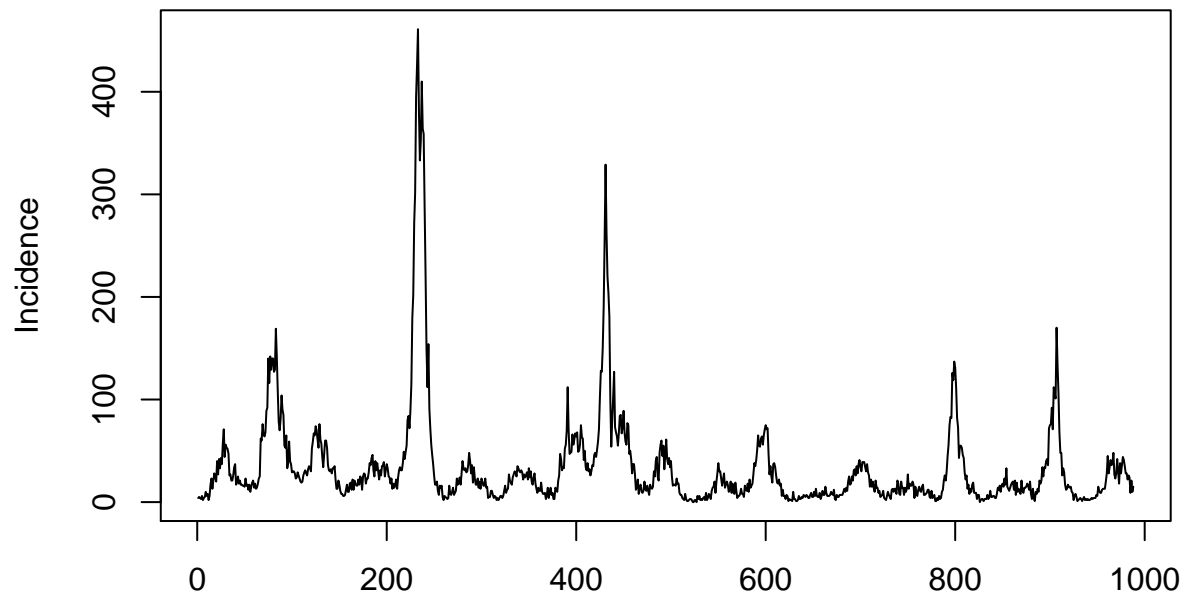
Locally-level+Seasonal



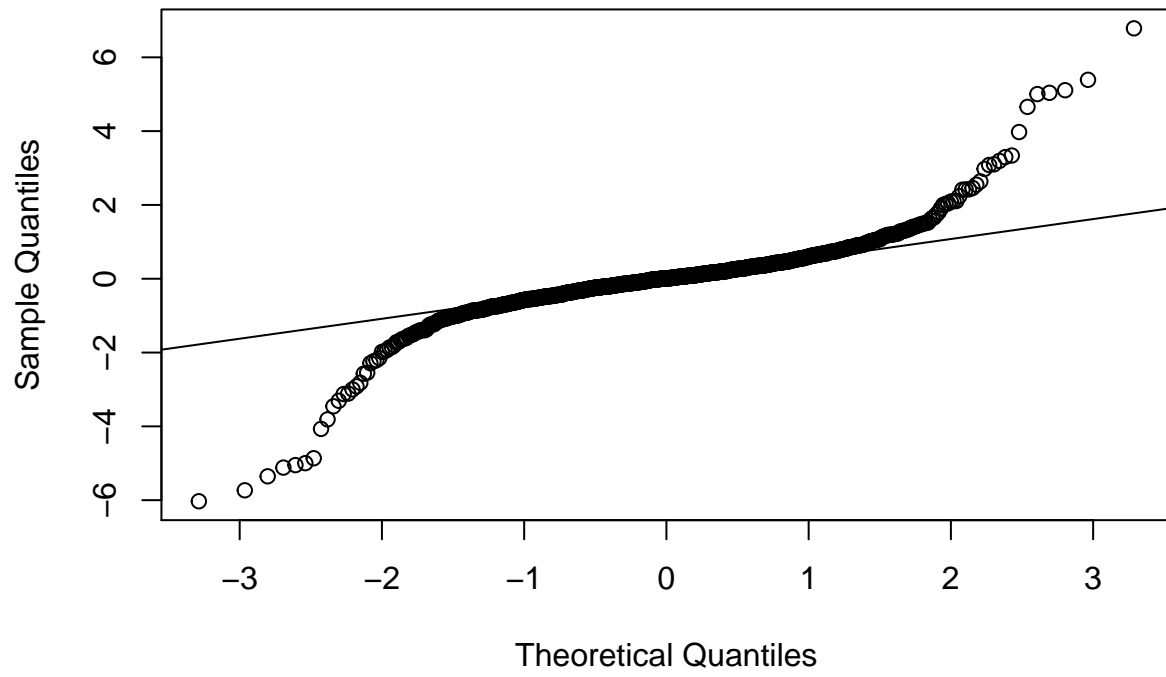
Normal Q-Q Plot



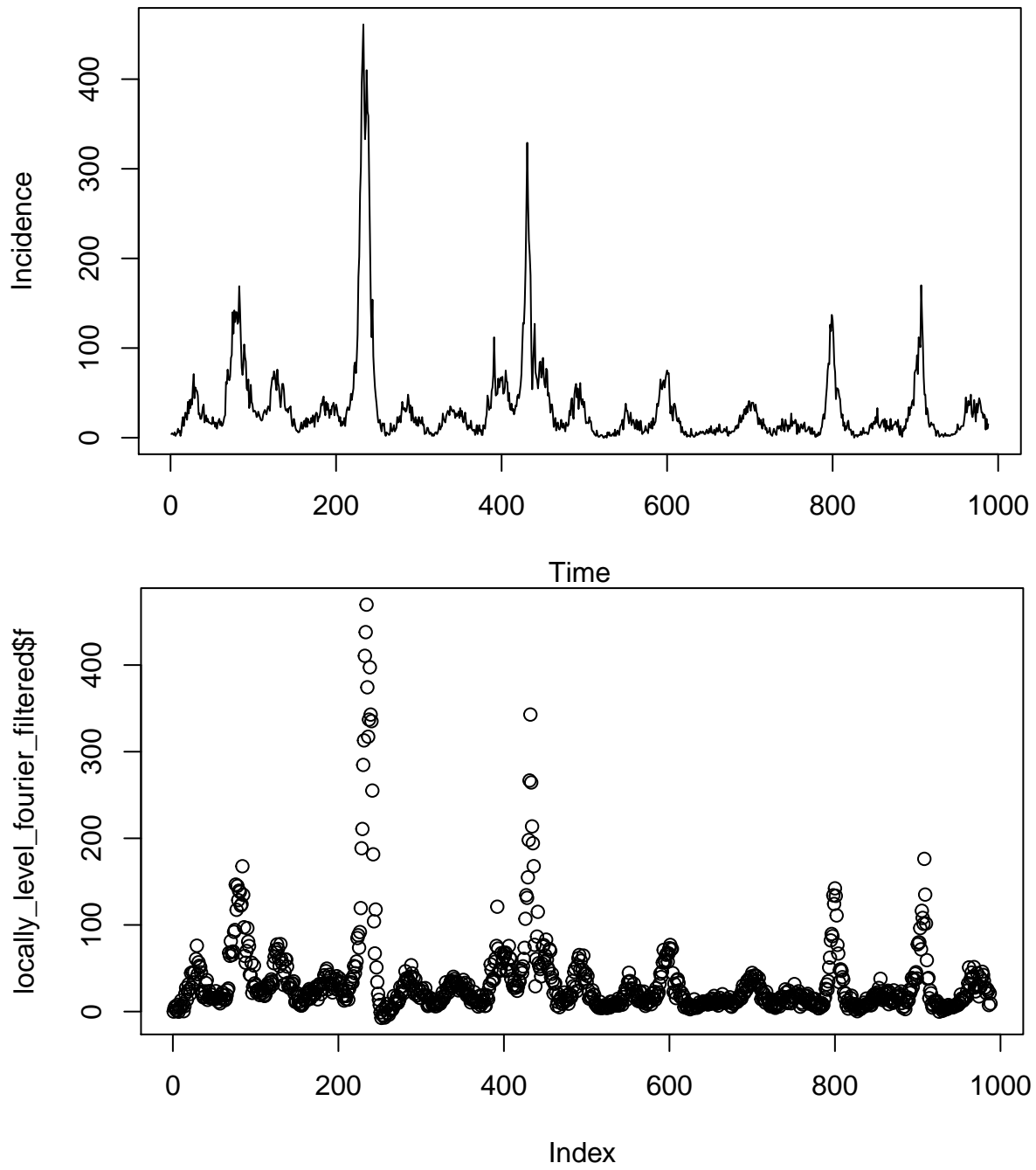
Locally-linear + Seasonal



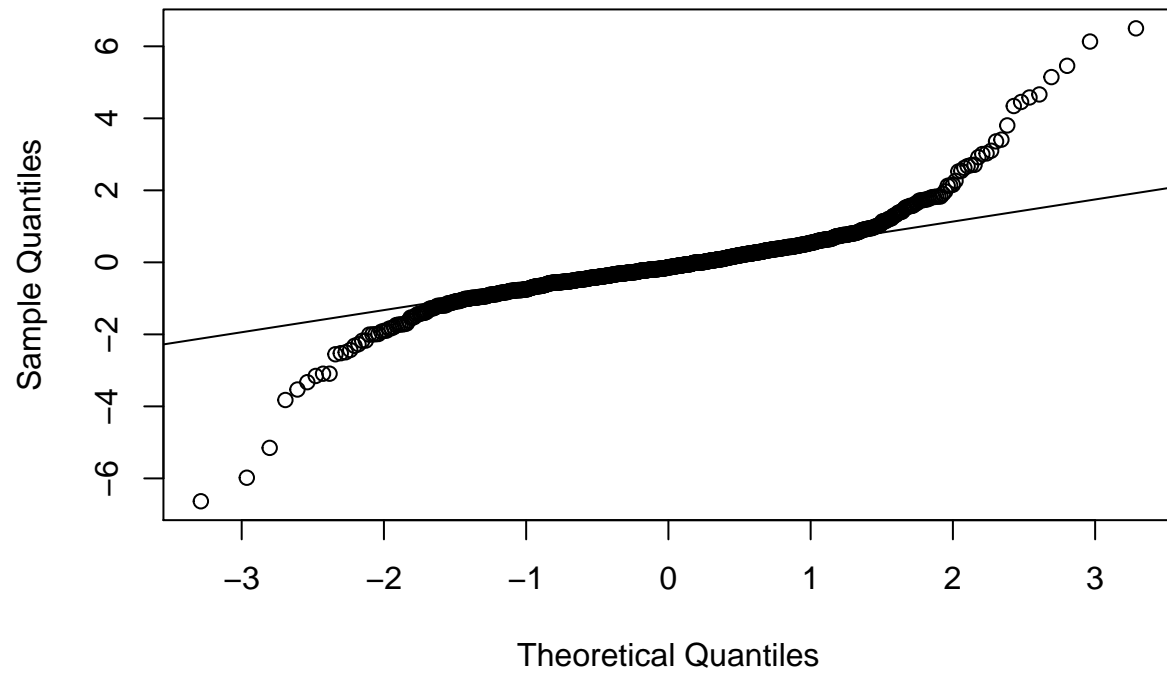
Normal Q-Q Plot



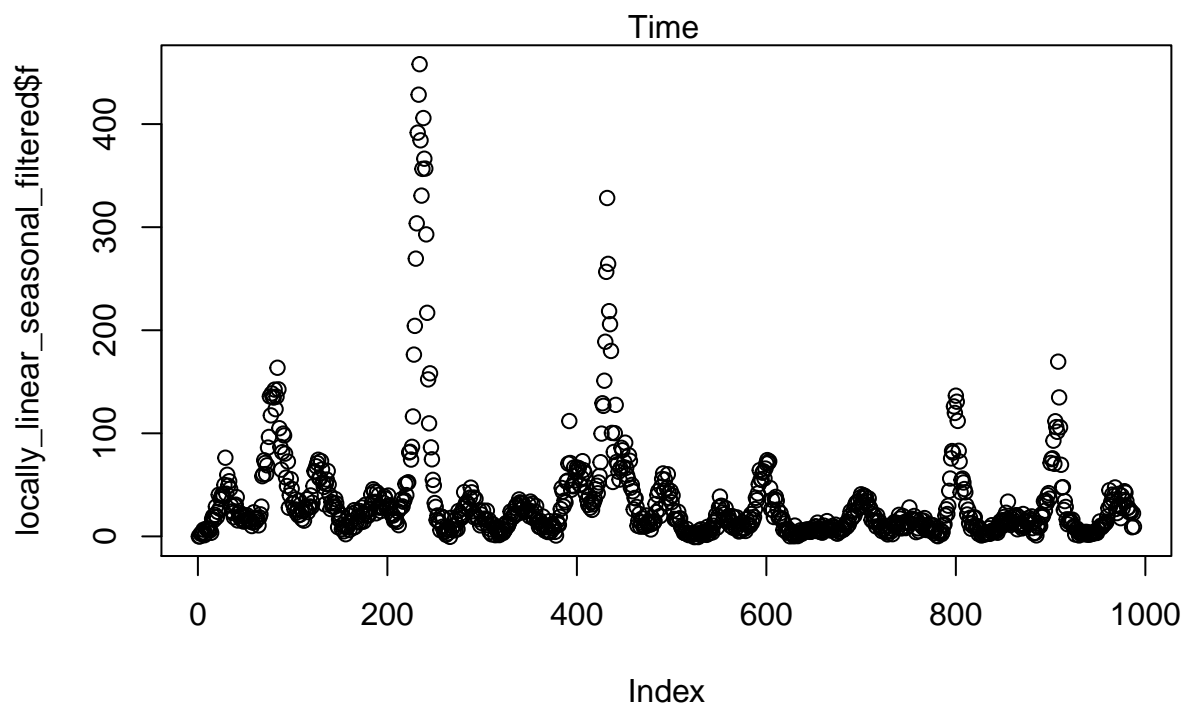
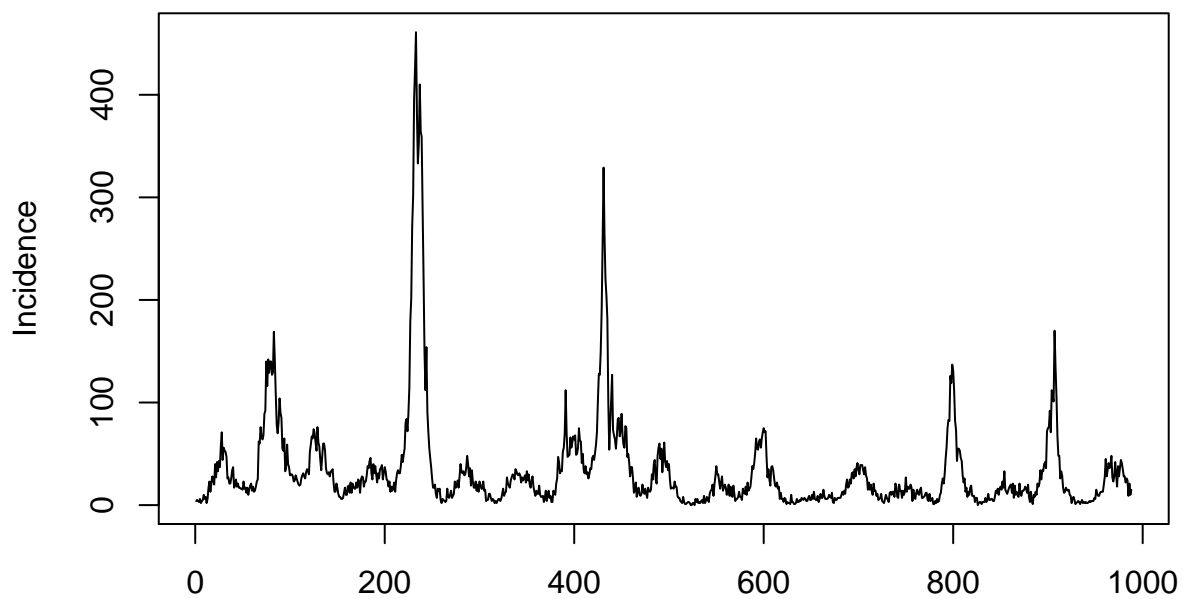
Seasonal vs Fourier Transform



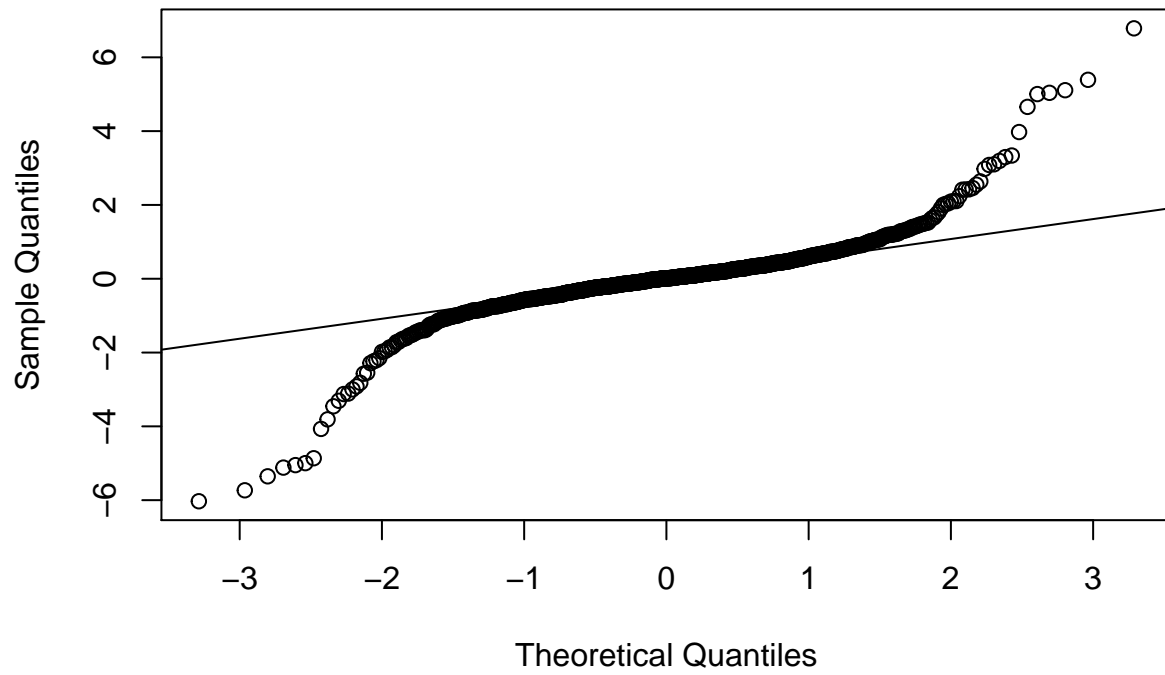
Normal Q-Q Plot



Multiple seasonality

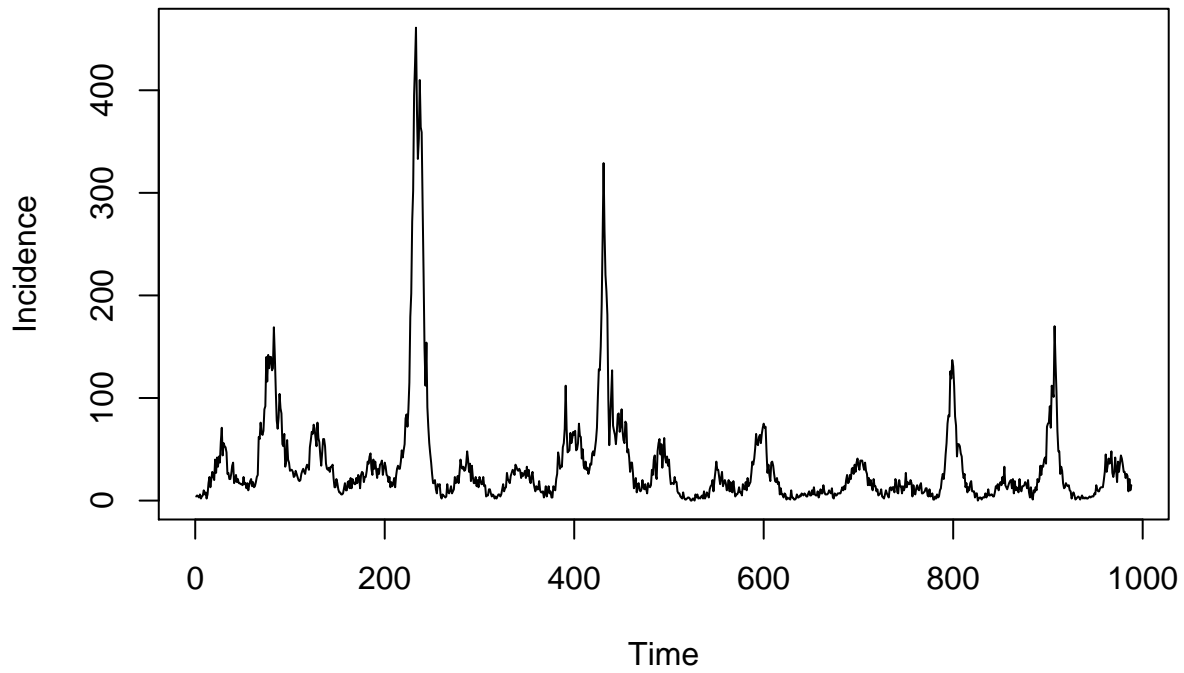


Normal Q-Q Plot



Alpha model

Standard



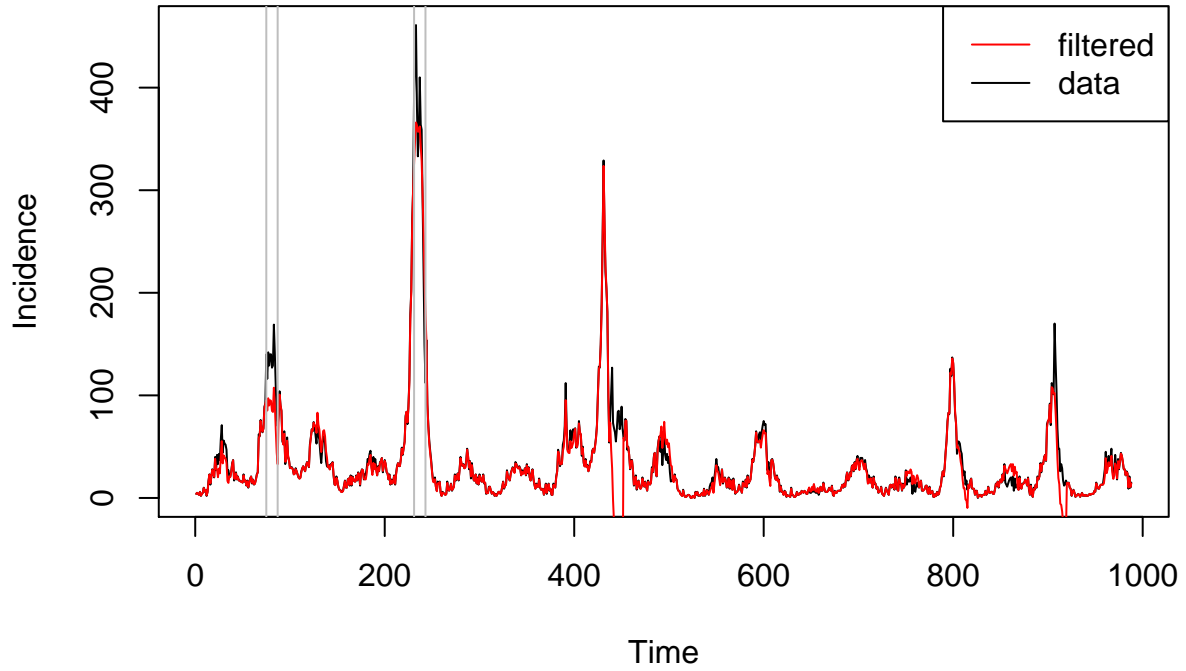
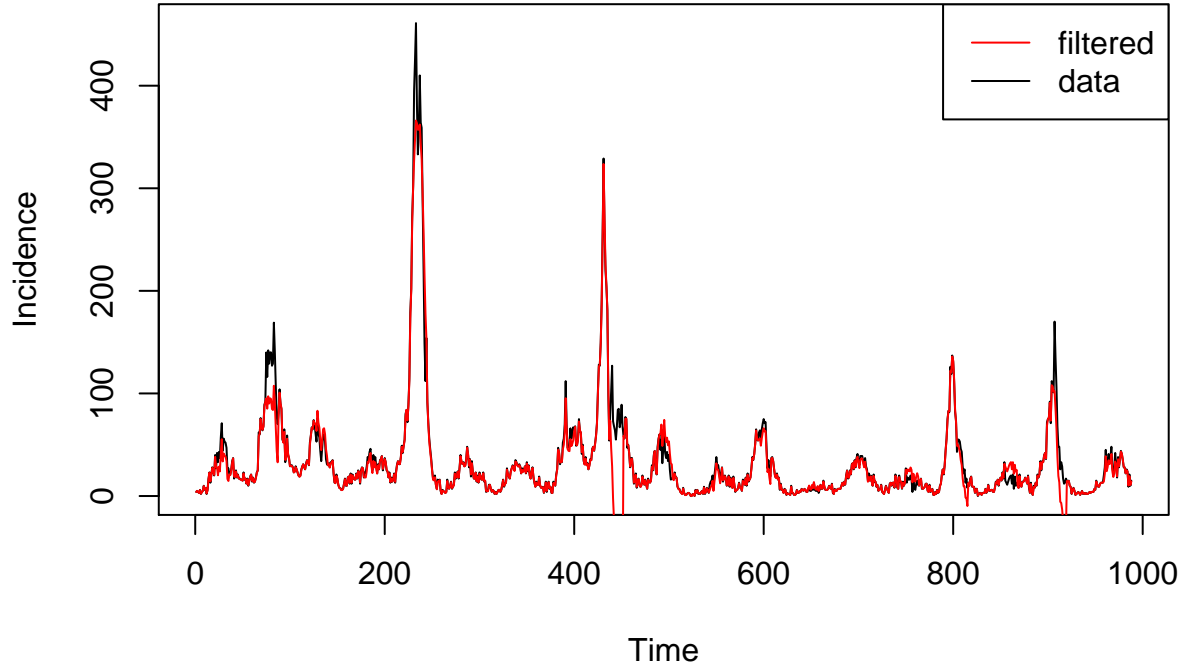
$$Y_t = F_t \Theta_t + W_t$$

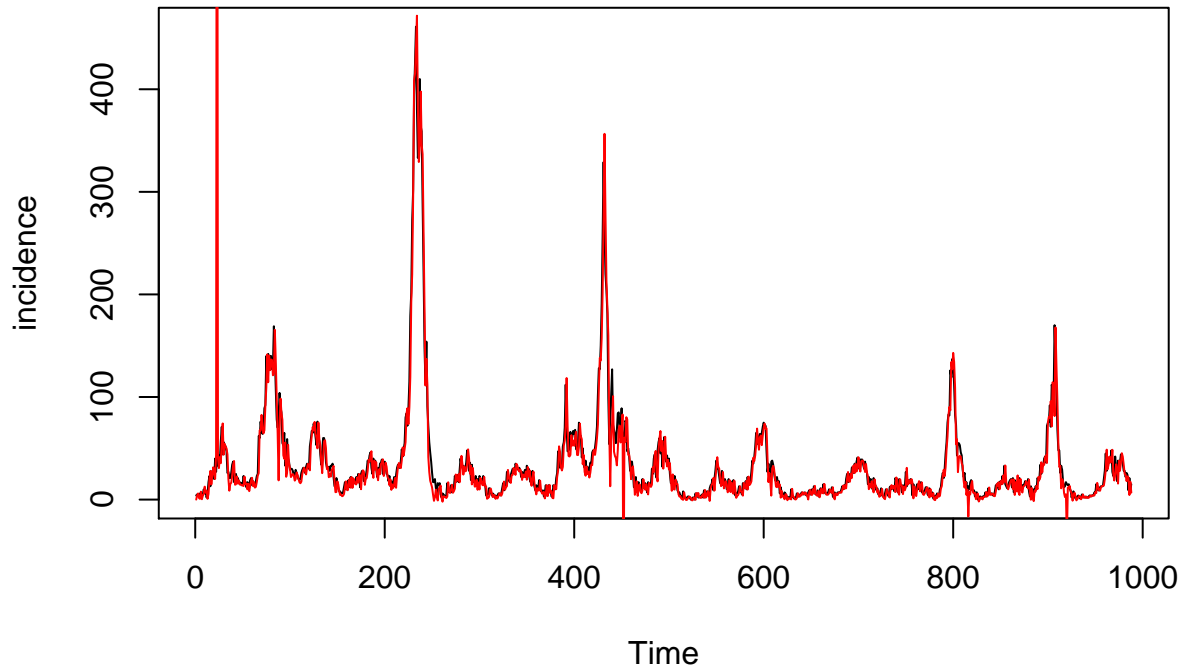
$$\Theta_t = G \Theta_{t-1} + V_t$$

When t is not in summer, $F_t = F = 1$,

when t is in summer, $F_t = \alpha_s F$.

where α_s is the maximum incidence in each year, $s = 1, \dots, 20$.





##log-alpha model dlm If we take a log transformation,

$$\log y = \log F\theta_t + \log \alpha$$

that is

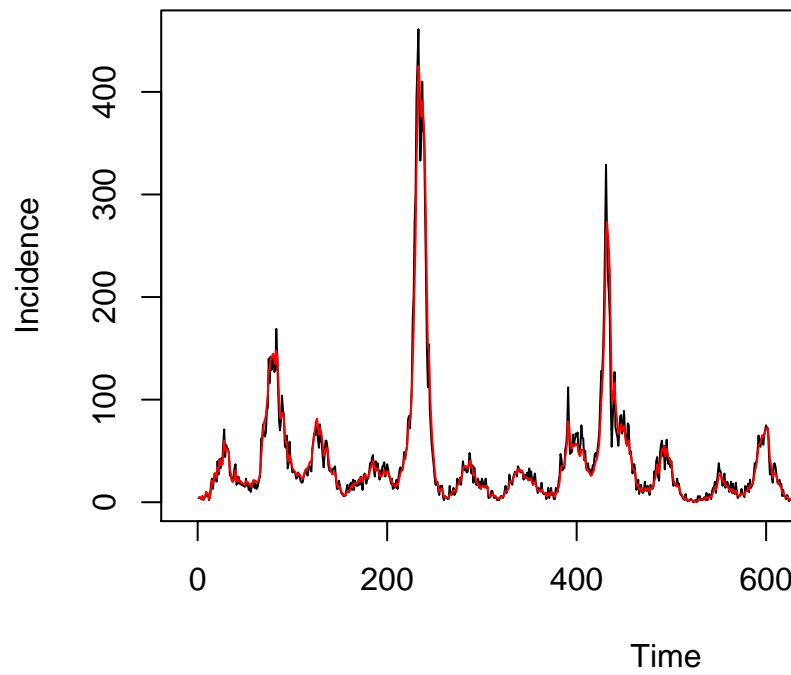
$$y' = F'\theta'_t + \alpha'$$

Let

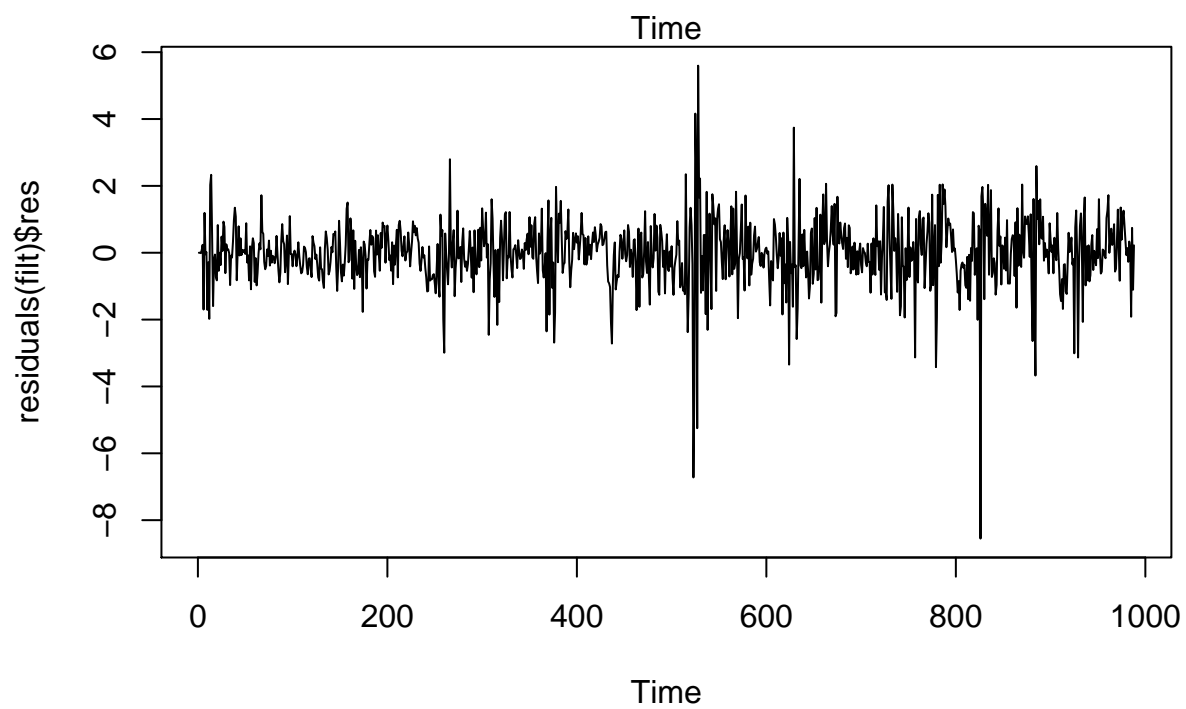
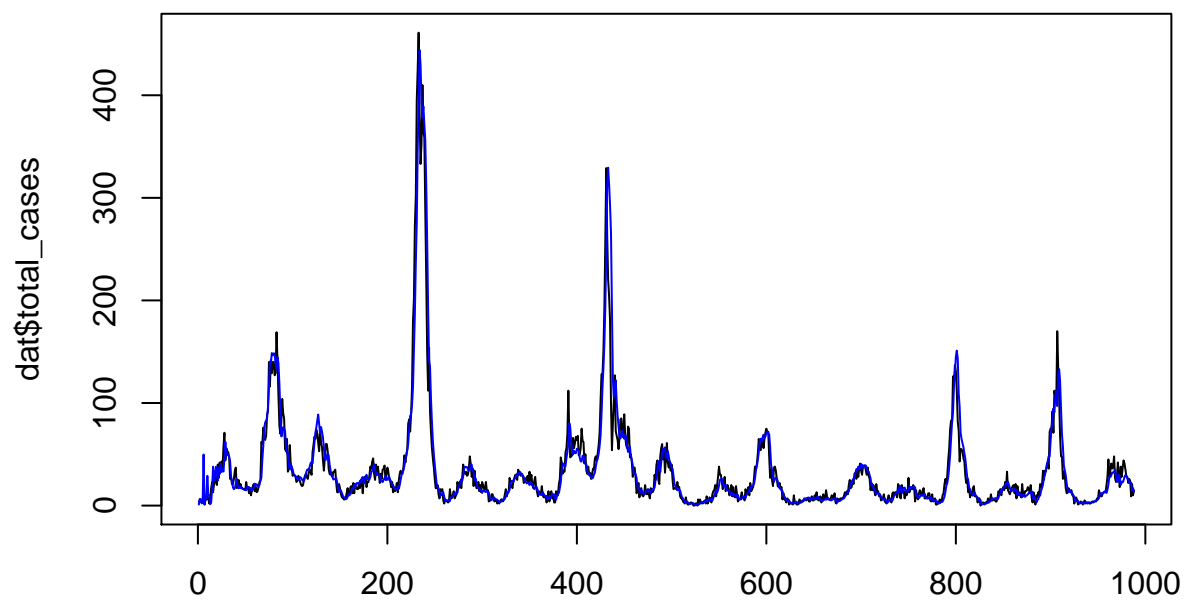
$$\theta_t^* = (\theta'_t, \alpha')$$

, and

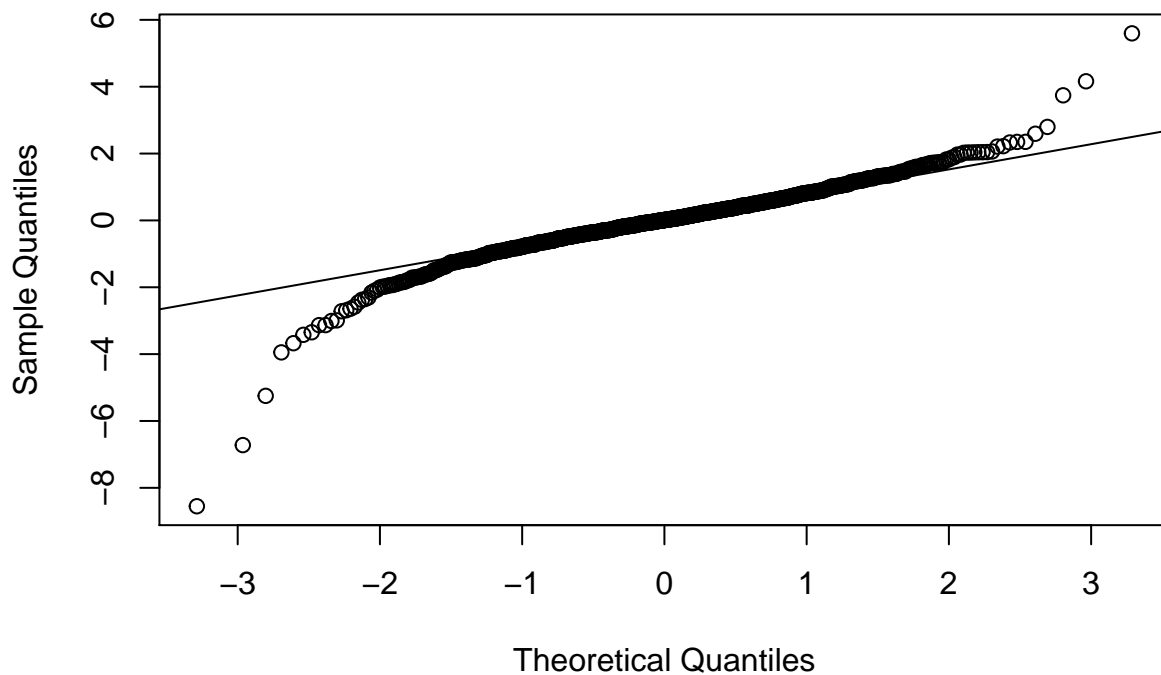
$$F^* = (F', f^\alpha)$$



, where $f^\alpha = \begin{cases} 1, & \text{in summer} \\ 0, & \text{not in summer} \end{cases}$ $w^\alpha = \begin{cases} w, & \text{week 1} \\ 0, & \text{otherwise} \end{cases}$



Normal Q-Q Plot



log-alpha model Particle filtering

```
if (FALSE){
  SJdat<-read.csv("http://dengueforecasting.noaa.gov/Training/San_Juan_Training_Data.csv")
  dat <- SJdat[c("season_week","total_cases")]

  cases <- dat$total_cases[1:100]
  require(rbiips)
  library(MCMCpack)
  model_file = '/Users/gcgibson/DLM_final_project/log_alpha.bug' # BUGS model filename
  cat(readLines(model_file), sep = "\n")

  par(bty='l')
  light_blue = rgb(.7, .7, 1)
  light_red = rgb(1, .7, .7)

  t_max = length(cases)
  n_burn = 500 # nb of burn-in/adaptation iterations
  n_iter = 1000 # nb of iterations after burn-in
  thin = 5 # thinning of MCMC outputs

  param_names = c('alpha') # names of the variables updated with MCMC (others are updated with SMC)
  latent_names = c('x') # names of the variables updated with SMC and that need to be monitored

  G = matrix(c(cos(2*pi),sin(2*pi),-sin(2*pi),cos(2*pi)), nrow=2, byrow=TRUE)
```

```

#setting the mean value of the initial count to 1400
data = list(t_max=t_max, y = cases, G = G, mean_sigma_init = c(0,0), cov_sigma_init=diag(2) ,mean_x_in
model = biips_model(model_file, data=data,sample_data = FALSE)

##fixing variance for now, will extend model to handle inference over variance later
n_burn = 5000 # nb of burn-in/adaptation iterations
n_iter = 10000 # nb of iterations after burn-in
thin = 5 # thinning of MCMC outputs
param_names = c('alpha') # names of the variables updated with MCMC (others are updated with SMC)
latent_names = c('x') # names of the variables updated with SMC and that need to be monitored
inits = list(10)

obj_pmmh = biips_pmmh_init(model, param_names, inits=list(1),
                           latent_names=latent_names) # creates a pmmh object
biips_pmmh_update(obj_pmmh, n_burn, n_part) # adaptation and burn-in iterations
out_pmmh = biips_pmmh_samples(obj_pmmh, n_iter, n_part, thin=thin) # samples

summ_pmmh = biips_summary(out_pmmh, probs=c(.025, .975))
}

```

Markov chain switching model

In order to incorporate various epidemic onset times we consider a model that permits time-varying.

```

if (FALSE){
model_file = '/Users/gcgibson/DLM_Final_Project/switching.bug' # BUGS model filename
cat(readLines(model_file), sep = "\n")

require(rbiips)
par(bty='l')
light_blue = rgb(.7, .7, 1)
light_red = rgb(1, .7, .7)
hot_colors = colorRampPalette(c('black', 'red', 'yellow', 'white'))

set.seed(0)
library(ggplot2)
SJdat<-read.csv("http://dengueforecasting.noaa.gov/Training/San_Juan_Training_Data.csv")
dat <- SJdat[c("season_week","total_cases")]

cases <- dat$total_cases[seq(1,10)]

sigma = .4; alpha = c(-2.5, -1); phi = .5; c0 = 1; x0 = 0; t_max = length(cases)
pi = matrix(c(.9, .1, .1, .9), nrow=2, byrow=TRUE)
data = list(t_max=t_max, sigma=sigma,y=cases,
            alpha=alpha, phi=phi, pi=pi, c0=c0, x0=x0)

model = biips_model(model_file, data, sample_data=FALSE)

n_burn = 20
n_iter = 100
thin = 1

```



```

obj_pimh = biips_pimh_init(model, variables)

biips_pimh_update(obj_pimh, n_burn, n_part) # Burn-in iterations

out_pimh = biips_pimh_samples(obj_pimh, n_iter, n_part, thin=thin) # Return samples
summ_pimh = biips_summary(out_pimh, probs=c(.025, .975))
}

```

Mechanistic Integration

$$\beta \sim \text{Exp}(\lambda_1)$$

$$\gamma \sim \text{Exp}(\lambda_2)$$

$$(S_t, I_t, R_t) \sim \text{Dirichlet}(f(S_{t-1}, I_{t-1}, R_{t-1}, \beta, \gamma))$$

$$f = \text{rkf approx to ode}$$

$$Y_t \sim \text{Pois}(NI_t)$$