Lab5_EmilKlassonSvensson

Emil K Svensson

9 December 2016

```
set.seed(1234567890)
library(geosphere)
```

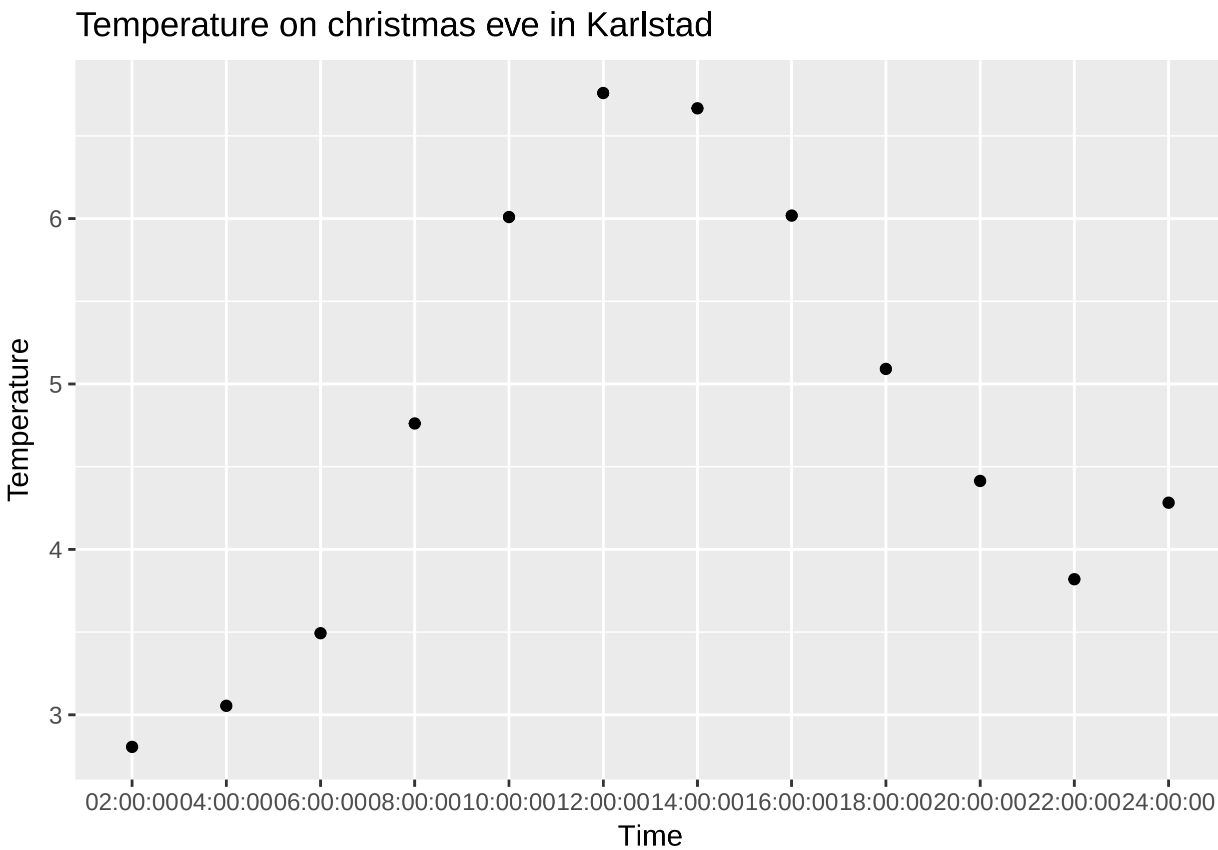
```
## Loading required package: sp
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
stations <- read.csv("stations.csv",header = TRUE)
temps <- read.csv("temps50k.csv")
st <- merge(stations,temps,by="station_number")


my_magic_kernel <- function(data ,time, date, longlat = c(59.4446, 13.3374), h_days = 6, h_time = 4, h_d

### Defining the kernel

gk <- function(x, xi){
  #for the days
  if( all(class(x) == "Date")) {

    xi <- as.Date(factor(xi),format = "%Y-%m-%d")
    return(exp(-((abs( as.numeric(x - xi) )^2) / (h_days) )))
  }

  #For the hours
  if(class(x)  == c("difftime") ) {

    xi <- strptime(xi,"%H:%M:%S")
    return(exp(-((abs( as.numeric(x) )^2) / (h_time))))
  }

  #For long and lat
  return(exp(-((abs( (x - xi))^2) / (h_distance))))
}


  #Initiatin objects for loop.
  predictions <- data.frame(time=1,temp=1)
  i <- 1

for (timme in times){


      mdate = strftime(paste(date,timme))
      data<-subset(st,  strftime(paste(st$date,st$time)) < mdate)
```

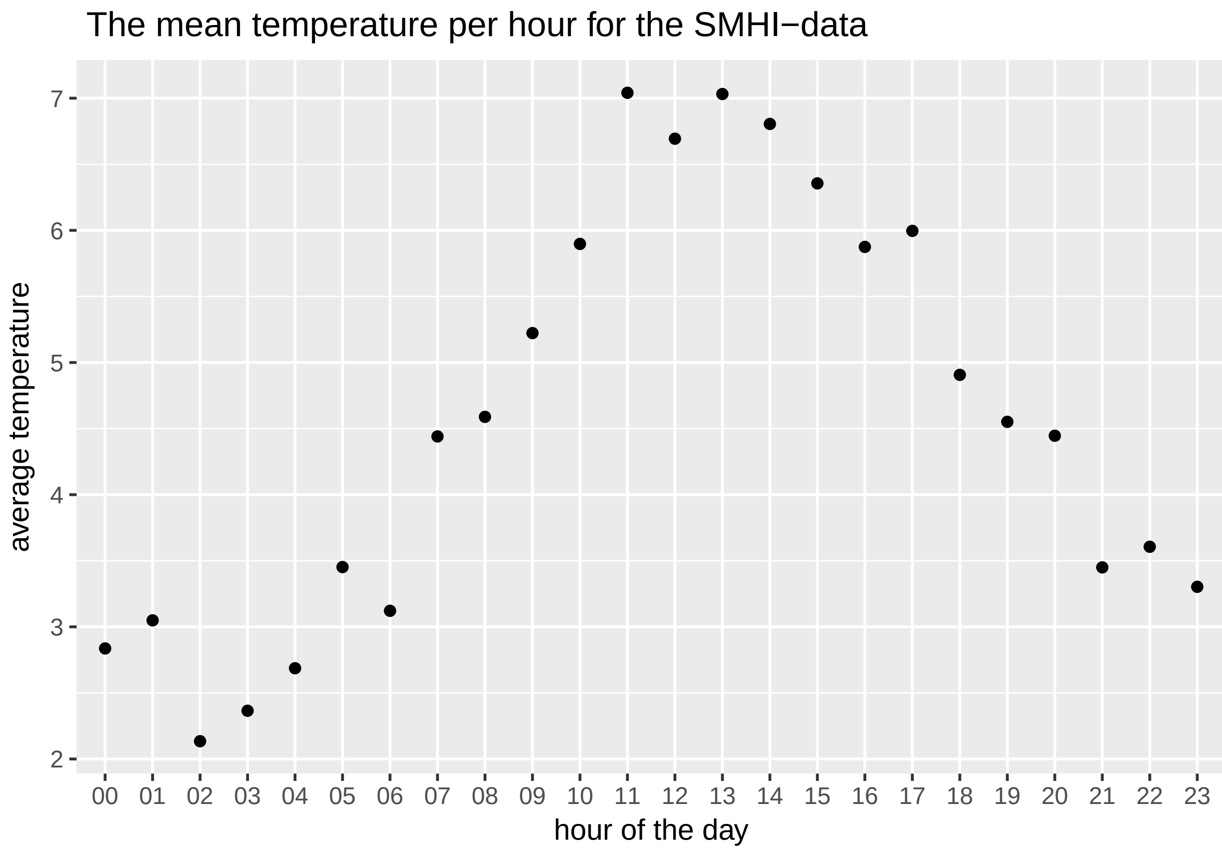
```
      #Longitude and Latitude distances.
      dmat <- geosphere::distHaversine(p1 = cbind(data$latitude,data$longitude) , p2 = longlat)
      gkdmat<- gk(dmat,0)

      #datum
      datevec <- as.Date(st$date)
      gkdate<-gk(datevec,date)


      #timme

      difftimes<-difftime(strptime(data$time,format="%H:%M:%S"),strptime(timme,format = "%H:%M:%S"),uni
      gktime<-gk(difftimes,0)

      alltemps <- rowSums(cbind(gkdmat,gktime,gkdate)*data$air_temperature)/sum((gkdmat + gkdate + gkti
      predictions[i,] <- c(timme,sum(alltemps))
      i <- i + 1

}

predictions[,1]<- as.factor(predictions[,1])
predictions[,2]<- as.numeric(predictions[,2])
return(predictions)

}


a <- 58.4274
b <- 14.826
times <- c(paste0("0",seq(2,9,2),":00:00"),paste0(seq(10,24,2),":00:00"))



as<-my_magic_kernel(data = st ,time = times, date = "2016-12-24", longlat = c(59.4446, 13.3374), h_days

as[,1] <- as.factor(as[,1])
as[,2] <- as.numeric(as[,2])
ggplot(data = as, aes(x= time,y=temp))+geom_point() +
  labs(x= "Time",y= "Temperature", title = "Temperature on christmas eve in Karlstad")
```



The temperature seem to be at warmest around noon and in almost a bell shaped pattern, which seems resonable when dealing with temperatures, it's cold during night time and warm during the day. Since the date I've chosen is christmas eve this year the kernel that takes in consideration of how long it was since the last observation isn't affecting the prediction so much, which is resonable since a observation six months ago alone doesn't say anything about the temperature six months in the future. In this case the kernels are indepentend wich renders the days kernel pretty bad since it won't give more importance to observations more close in time.

In this scenario the time-component is probably the one affecting the prediction the most since it only cares about the time of the day and nothing elese. Here is a plot of the averages of the different time-periods.

```
tempagg<- aggregate(st$air_temperature,list(factor(substr(st$time,start= 1,stop =2))),FUN = mean)

ggplot(data = tempagg, aes(x = Group.1, y =x )) +geom_point()+
  labs(x= "hour of the day", y = "average temperature", title = " The mean temperature per hour for the
```



The general shape of the means matches the pattern of the prediction very well. This servers as an indication that much of the weighting is controled by this kernel which seems sensible. The smoothing factor of two-hours might be a bit narrow, but it's a bell-shaped pattern it seems justified as we're using a gaussian kernel.

The last kernel was the distance to the station the observation was recorded. This is also a pretty bad predictior, temperatures are a very regional occurences, the kernel smoother should be set to a low value, 100 000 is in this case a low value sincie it corresponds to 100 km which isn't that much considering the size of Sweden.

A more apropriate kernel to include would be one similar to the time-variable but in regards to the day of the year it is, but since this isn't SMHI and I don't have the time to implement it I will consider this model very crude and shoudn't be used for predictions.