TDDE01: Lab 5

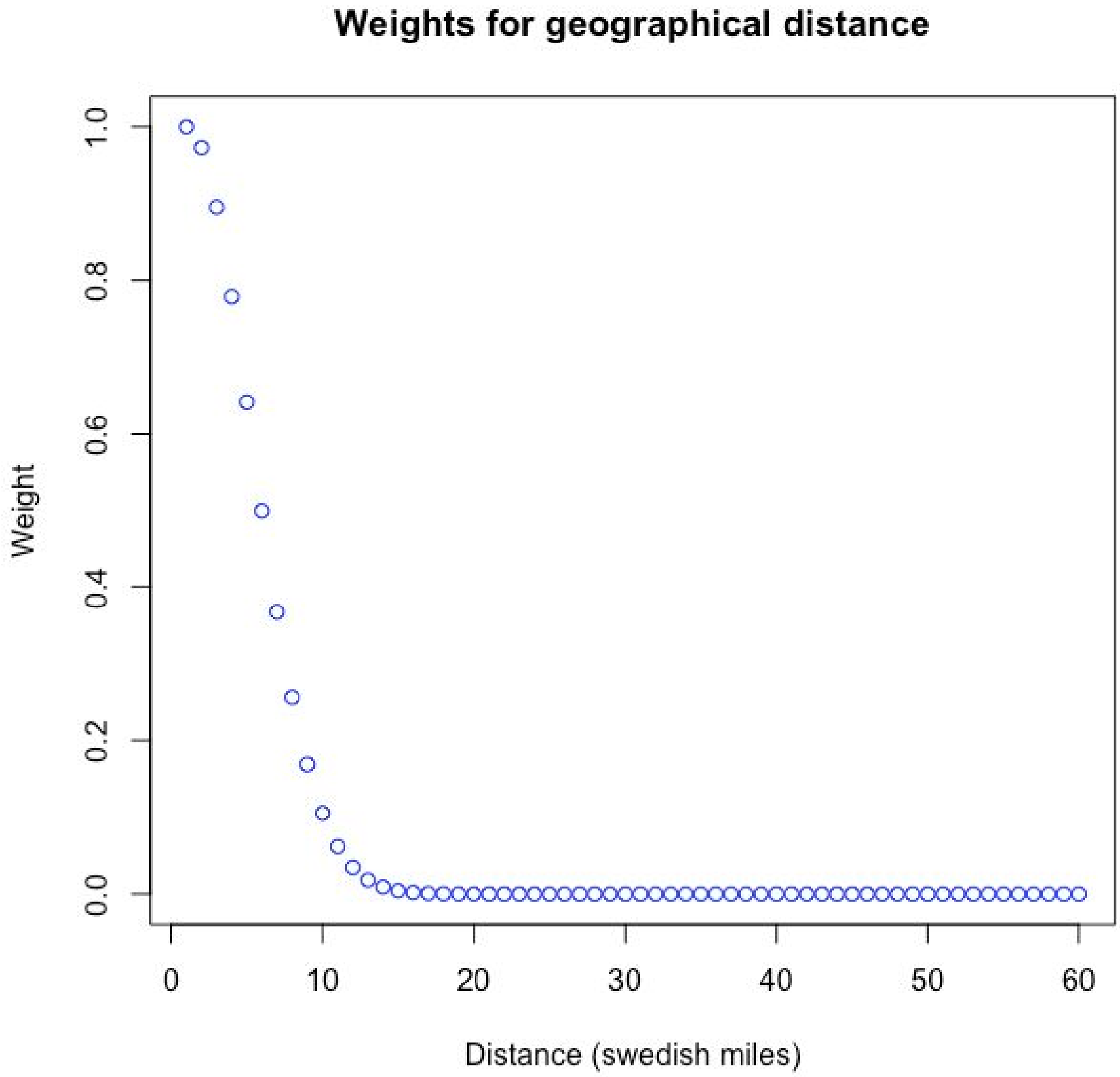
A report by

Adam Nyberg
adany869

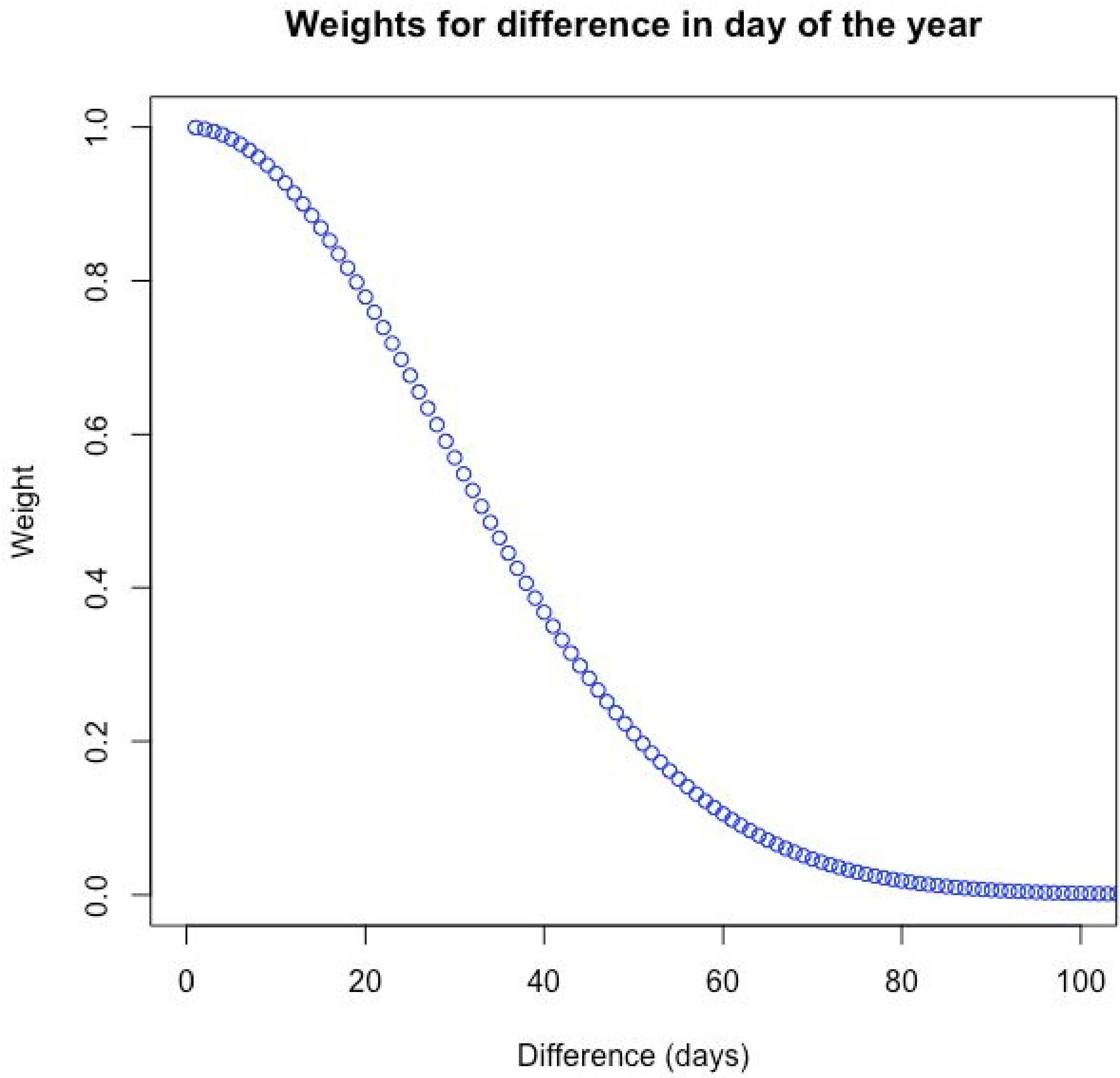
In this assignment we were to predict weather. We got two files with data that included temperature, date and geographical location. Based on this our goal was to return the temperature for one day with two hour intervals.

Kernels

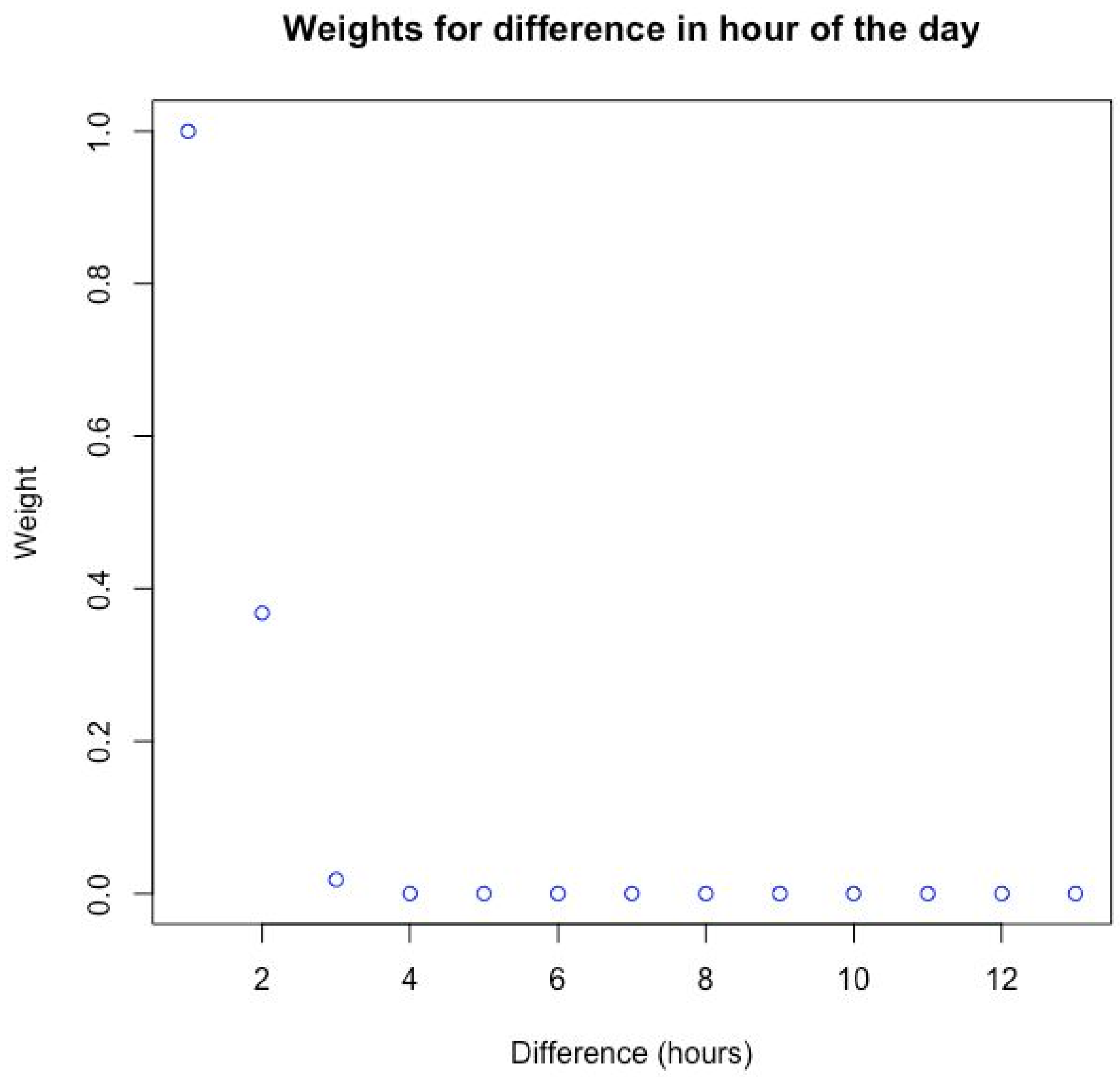
We did this with three guassian kernels that we sum together. I chose the width of the kernel by trying different values and then confirming that it's reasonable by plotting it. The first kernel was for the geographical distance with a width constant h_distance=60000. The kernel is shown below.



The second kernel was for the date distance and here I chose a width constant h_date=40. The kernel is shown below.



The third kernel was for the time distance and here I chose a width constant h_time=3*3600. The kernel is shown below.



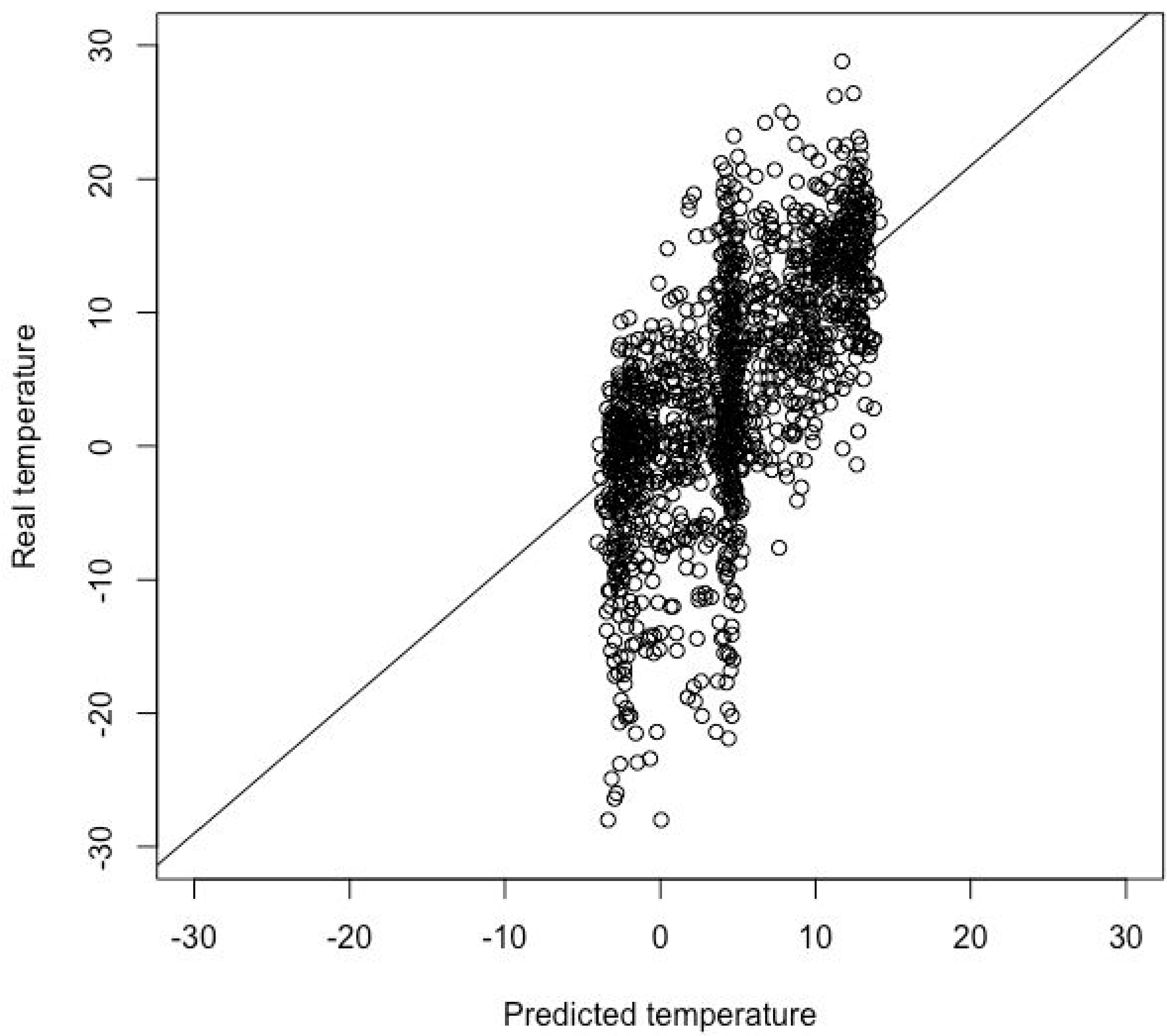
We sum the different kernels together with the following formula.

```
sum_dist = c_distance*station_dist + c_date*date_dist + c_time*hour_dist
```

Where c_distance=0.1, c_date=1 and c_time=0.1.

Train test

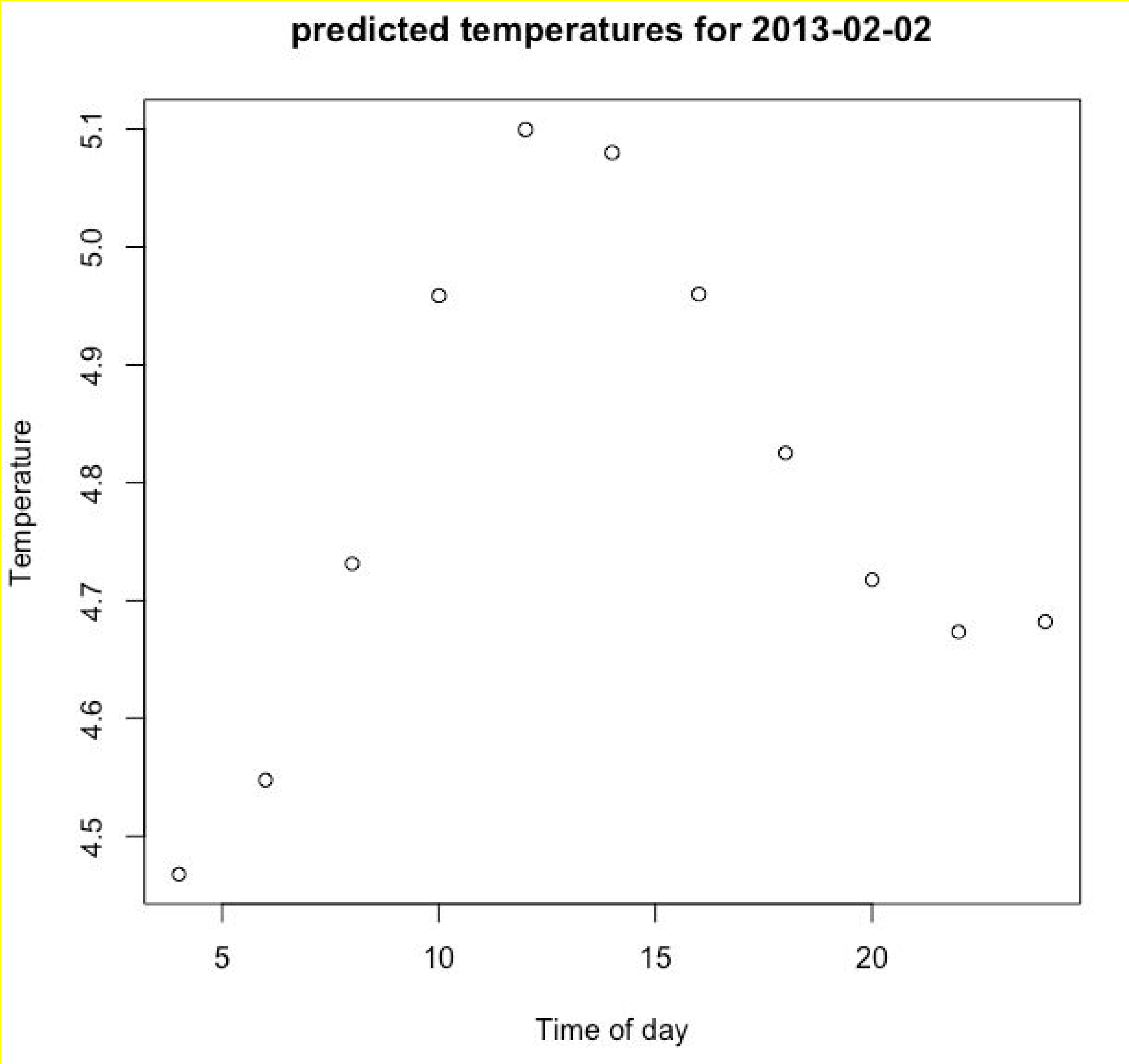
We now tried to test how well our prediction were compared to the real temperatures. We did it by first getting 5000 samples of our data and then dividing our samples into training and test sets. We then got the following plot.



As we can see we predict a lot of values around 5 degrees. That is because we sum the kernels and 5 is close to the average.

Results

I now use our kernels to predict the temperatures for one day as seen below.



As we can see here as well, the temperature is around 5 degrees. Forgot to add this plot in the group report.

Questions

- Show that your choice for the kernels' width is sensible, i.e. that it gives more weight to closer points. Discuss why your of definition of closeness is reasonable.

As we could see when plotting our kernels our widths are very reasonable. We only care about geographical distances that are closer than 150km, date of year distance that is closer than 60 days and time of day distance that is closer than 3 hours.

- It is quite likely that the predicted temperatures are too low. Do you think that the reason may be that the three Gaussian kernels are independent one of another?

Because we sum up our kernels that makes the kernels independent. That will also make our result not very good.

Added code below. Forgot to add the plot in the group report.

```
setwd("~/code/skola/tdde01/adam")

library(geosphere)

set.seed(1234567890)

stations = read.csv("lab5/stations.csv")
temps = read.csv("lab5/temps50k.csv")
st = merge(stations, temps, by="station_number")[sample(1:50000, 50000),]

n=dim(st)[1]
id=sample(1:n, floor(n*0.75))
train=st[id,]
test=st[-id,]

difftime_distance = function(x, y, unit_, format_) {
  date_strings = c(x, y)
  datetimes = strptime(date_strings, format=format_)
  diff = as.double(difftime(datetimes[2], datetimes[1], units=unit_))
  return(diff)
}

diff_hour = function(t1, t2) {
  return(abs(as.numeric(difftime(strptime(t1, "%H:%M:%S"),
                           strptime(t2, "%H:%M:%S")))))
}
diff_date = function(t1, t2) {
  return(as.numeric(difftime(strptime(t1, "%Y-%m-%d "),
                           strptime(t2, "%Y-%m-%d "))))
}

k_gaussian = function(u, h) {
  return(exp(-(abs(u/h)^2)))
}

station_distance = function(p1, p2) {
  return(k_gaussian(distHaversine(p1,p2), h_distance))
}

date_distance = function(d1, d2) {
  return(k_gaussian((diff_date(d1, d2) %% 365), h_date))
}

hour_distance = function(h1, h2) {
  return(k_gaussian(diff_hour(h1, h2), h_time))
}
```

```
calc_temp = function(dist, Y) {
  sumsum_dist = sum(dist)
  norm_dist = dist/sumsum_dist
  temp = as.vector(t(norm_dist))%*%as.vector(Y)
}

h_distance <- 60000 # These three values are up to the students
h_date <- 40
h_time <- 3*3600

c_distance = 0.1
c_date = 1
c_time = 0.1

a = 58.413497  # The point to predict (up to the students)
b = 15.582597
point = c(a, b)
date = "2013-02-02" # The date to predict (up to the students)
times = c("04:00:00",
          "06:00:00",
          "08:00:00",
          "10:00:00",
          "12:00:00",
          "14:00:00",
          "16:00:00",
          "18:00:00",
          "20:00:00",
          "22:00:00",
          "23:00:00")
temp = vector(length=length(times))
# Students??? code here

predict_temps = function(point_, date_, hour_) {
  station_dist = station_distance(point_, train[4:5])
  date_dist = date_distance(date_, train$date)

  temps_ = c()
  if (missing(hour_)) {
    for(i in 1:length(times)) {
      hour_dist = hour_distance(times[i], train$time)

      sum_dist = c_distance*station_dist + c_date*date_dist + c_time*hour_dist

      temps_[i] = calc_temp(sum_dist, train$air_temperature)
    }
```

```
    return(temps_)
  } else {
    hour_dist = hour_distance(hour_, train$time)
    sum_dist = c_distance*station_dist + c_date*date_dist + c_time*hour_dist
    temp = calc_temp(sum_dist, train$air_temperature)
    return(temp)
  }
}



# Predict only one day
pred_one_day = predict_temps(point, date)
plot(seq(4, 24, 2), pred_one_day, main="predicted temperatures for 2013-02-02",
ylab="Temperature", xlab="Time of day")

# train test

Yhat = c()
for(i in 1:nrow(test)) {
  test_row = test[i,]
  Yhat[i] = predict_temps(c(test_row$latitude, test_row$longitud), test_row$date,
test_row$time)
}

plot(Yhat, test$air_temperature, ylim=range(-30,30), xlim=range(-30,30), ylab="Real
temperature", xlab="Predicted temperature")
abline(1,1)


plot_h_values = function() {

  plot(k_gaussian(matrix(seq(1,600000,10000)), h_distance), main="Weights for
geographical distance", ylab="Weight", xlab="Distance (swedish miles)", col="blue")

  plot(k_gaussian(matrix(seq(1,365,1)), h_date), xlim=range(0,100), main="Weights for
difference in day of the year", ylab="Weight", xlab="Difference (days)",  col="blue")

  plot(k_gaussian(matrix(seq(0,24*3600,2*3600)), h_time), main="Weights for difference in
hour of the day", ylab="Weight", xlab="Difference (hours)", col="blue")
}
```