Notes

2024-05-15

```
# devtools::install_github("mfasiolo/electBook")
library(electBook)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
data(Irish)

summary(Irish$survey)
```

```
##       ID               meanDem        SOCIALCLASS  OWNERSHIP
##  Length:2672        Min.   :0.02032   AB: 410     Length:2672
##  Class :character   1st Qu.:0.31820   C1: 730     Class :character
##  Mode  :character   Median :0.46698   C2: 449     Mode  :character
##                     Mean   :0.49938   DE:1018
##                     3rd Qu.:0.64220   F :  65
##                     Max.   :1.75077
##    BUILT.YEAR    HEAT.HOME          HEAT.WATER        WINDOWS.doubleglazed
##  Min.   :1674   Length:2672        Length:2672        Length:2672
##  1st Qu.:1962   Class :character   Class :character   Class :character
##  Median :1979   Mode  :character   Mode  :character   Mode  :character
##  Mean   :1972
##  3rd Qu.:1997
##  Max.   :2008
##  HOME.APPLIANCE..White.goods.      Code   ResTariffallocation
##  Min.   :0.00                 Min.   :1   A:733
##  1st Qu.:3.00                 1st Qu.:1   B:294
##  Median :4.00                 Median :1   C:744
##  Mean   :3.61                 Mean   :1   D:276
##  3rd Qu.:5.00                 3rd Qu.:1   E:568
##  Max.   :5.00                 Max.   :1   W: 57
##  ResStimulusallocation
##  1:506
##  2:531
##  3:493
##  4:517
##  E:568
##  W: 57
```

```
print("---------------------------------------")
```

```
## [1] "---------------------------------------"
```

```
print("----------------------------------------")
```

```
## [1] "----------------------------------------"
```

```
print("----------------------------------------")
```

```
## [1] "----------------------------------------"
```

```
summary(Irish$extra)
```

```
##       time            toy          dow          holy              tod
##  Min.   :    1   Min.   :0.0000   Sun:2208   Mode :logical   Min.   : 0.0
##  1st Qu.: 4200   1st Qu.:0.2411   Thu:2496   FALSE:16799     1st Qu.:12.0
##  Median : 8400   Median :0.5041   Mon:2400                   Median :24.0
##  Mean   : 8400   Mean   :0.4975   Tue:2400                   Mean   :23.5
##  3rd Qu.:12600   3rd Qu.:0.7452   Wed:2544                   3rd Qu.:35.5
##  Max.   :16799   Max.   :0.9918   Sat:2352                   Max.   :47.0
##                                   Fri:2399
##       temp            dateTime
##  Min.   :-10.000   Min.   :2009-12-29 23:00:00.00
##  1st Qu.:  4.000   1st Qu.:2010-03-31 10:45:00.00
##  Median :  9.000   Median :2010-07-05 22:30:00.00
##  Mean   :  8.616   Mean   :2010-07-03 00:08:03.46
##  3rd Qu.: 14.000   3rd Qu.:2010-10-01 10:15:00.00
##  Max.   : 24.000   Max.   :2010-12-31 22:30:00.00
##
```

Gaussian Process Regression

Gaussian process regression seems like a good fit for our noisy data, especially since we will have a distribution of possible `demand` points for each `dateTime`, corresponding to the uncertainty in our predictions.

Theory

A gaussian process is a collection of random variables, which have a joint Gaussian distribution. A Gaussian process is completely specified by its mean function and covariance function. We build the following model:

Let $y_i = f(x_i) + \varepsilon_i$, where $f(x) \sim \text{GP}(0, k(x, x'))$ and $\varepsilon_i \sim N(0, \sigma^2)$. Then we can find the posterior distribution of $f(x_*)$ given y as:

$$f(x_*)|y \sim N(\mu(x_*), \sigma^2(x_*))$$

where $\mu(x_*) = k(x_*, x)^T(K + \sigma^2 I)^{-1}y$ and $\sigma^2(x_*) = k(x_*, x_*) - k(x_*, x)^T(K + \sigma^2 I)^{-1}k(x_*, x)$. In pratice, to find the posterior distribution, we maximise the marginal log-likelihood.

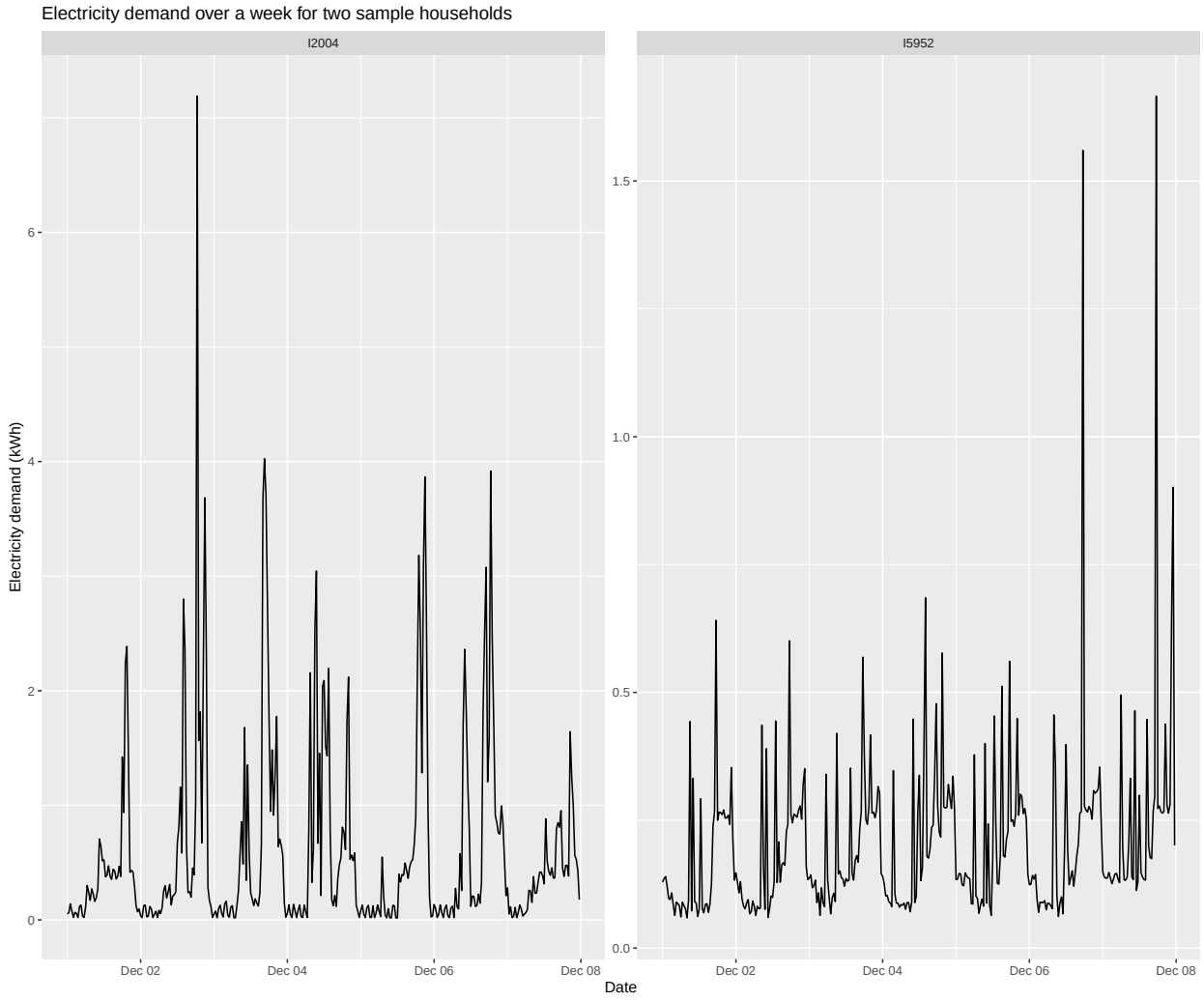
Implementation

```r
library(kernlab)
library(tidyverse)
```

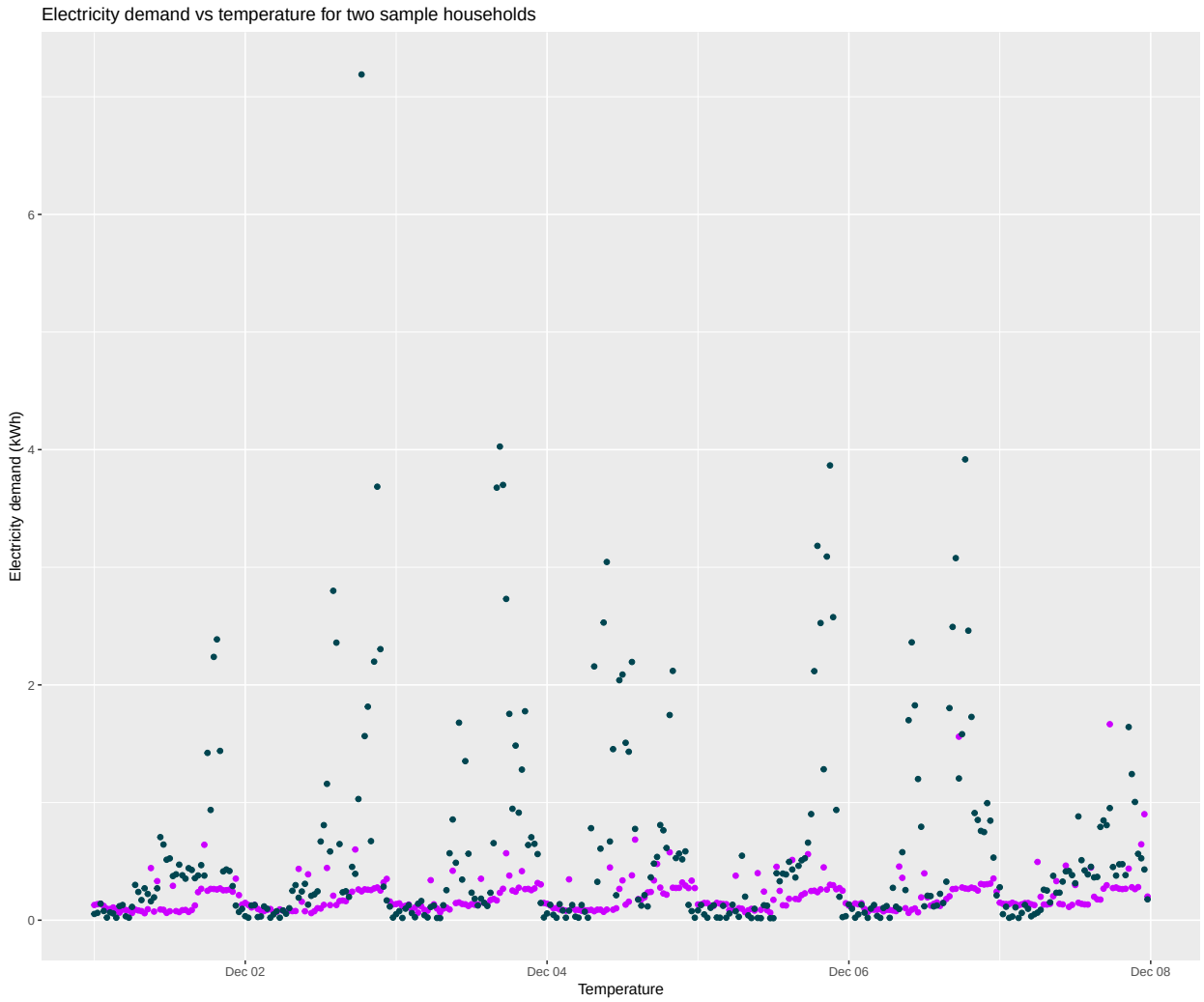
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x ggplot2::alpha() masks kernlab::alpha()
## x purrr::cross()   masks kernlab::cross()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
# Data cleaning
# Sample a subset of households for plotting
house_sample <- sample(colnames(Irish$indCons), 2)

irish_demand_sample <- Irish$indCons[,house_sample] %>%
  bind_cols(Irish$extra) %>% # add time-related variables
  pivot_longer(cols = all_of(house_sample), names_to = "house_sample", values_to = "demand") %>%
  # Data cleaning
  select(-time, -holy) %>%
  # Feature engineering
  mutate(
    hour = hour(dateTime),
    month = month(dateTime),
    weekend = ifelse(dow %in% c("Sat", "Sun"), 1, 0)
  ) %>%
  mutate(temp_sq = temp^2) %>%  # quadratic term for temperature
  # One-hot encode the day of the week
  bind_cols(model.matrix(~ dow - 1, data = .)) %>%
  select(-dow)


irish_demand_train <- irish_demand_sample %>%
  filter(dateTime < "2010-12-01")

irish_demand_test <- irish_demand_sample %>%
  filter(dateTime >= "2010-12-01")

house1 <-  filter(irish_demand_sample,house_sample == house_sample[1] & dateTime >= "2010-12-01" & date
house2 <-  filter(irish_demand_sample,house_sample == house_sample[2] & dateTime >= "2010-12-01" & date

#Plot demand over a week for two sample households
irish_demand_sample %>%
  filter(dateTime >= "2010-12-01" & dateTime < "2010-12-08") %>%
  ggplot(aes(x = dateTime, y = demand)) +
  geom_line() +
  facet_wrap(~house_sample, scales = "free_y") +
```

```
  labs(title = "Electricity demand over a week for two sample households",
       x = "Date", y = "Electricity demand (kWh)")
```




```
ggplot()+
  geom_point(data = house1, aes(x = dateTime, y = demand),color = "#cc00ff")+
  geom_point(data = house2, aes(x = dateTime, y = demand),color = "#004550")+
  labs(title = "Electricity demand vs temperature for two sample households",
       x = "Temperature", y = "Electricity demand (kWh)")
```




```
neg_marginal <- function(params){
  lambda <- params[1]
  psi <- params[2]
  #Compute K_n and K+lambdaI
  K <- kernelMatrix(rbfdot(sigma = psi), x)
  L <- K + lambda*diag(n)
  if (det(L) == 0){
    return(Inf)
  }
  #Compute alpha
  y <- as.matrix(y)
  alpha = solve(L,y)
  #Compute neg log marginal likelihood
  neg_marginal_val <- 0.5*(t(y)%*%alpha + sum(log(diag(L))))

  return(neg_marginal_val)
}
```