# Excercise 3 - Grey-box models (continued)

Models for the heat dynamics of a building

*Marco Hernandez Velasco*

*August 2018*

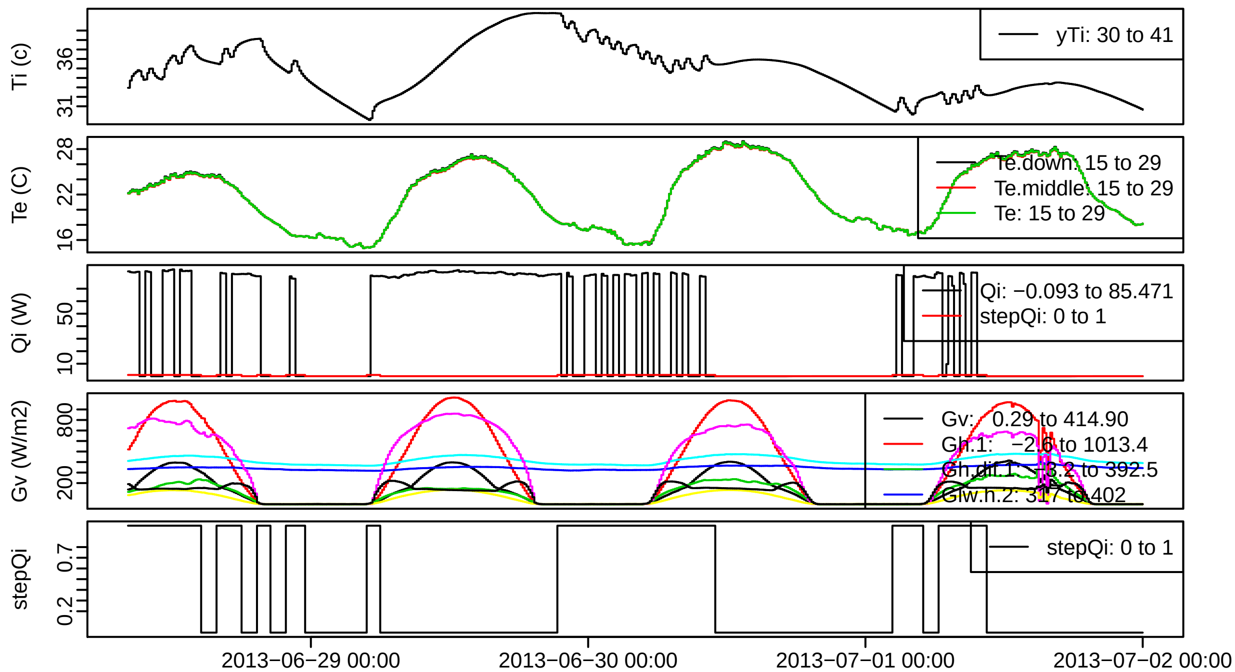
The exercise is focused on grey-box modelling of the heat dynamics of a (small) building using stochastic differential equations (SDEs). In addition to the first exercise on greybox modelling, we will in this exercise test different techniques to:

1. Alter the noise level or system uncertainty to account for e.g. non-linear phenomena.
2. Build a semi-parametric model to take into account that the solar penetration (i.e. relation between measured solar radiation and radiation entering into the building) as function of the position of the sun.
3. Balance heat gains to the air temperature and the temperature of the thermal mass.

The data consists of several measurement from a small test box with a single window. In this exercise the following signals are used:

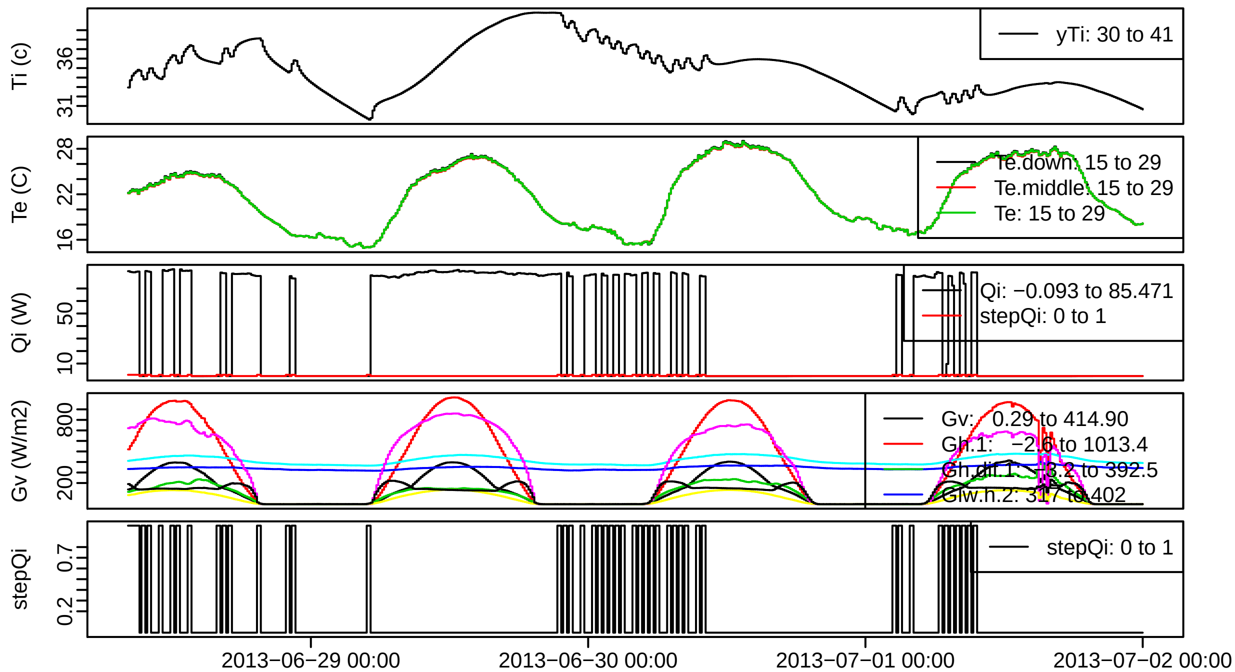
- Ti (yTi in data) the observed indoor temperatures. (C)
- Qi (Qi in data) the heat emitted by the electrical heaters in the test box (W)
- Te (Te in data) the ambient temperature (C)
- Gv (Gv in the data) the vertical south total solar radiation (W/m2)
- Gvn (Gvn in data) the vertical north total solar radiation (W/m2)

# Question 1




The lower time series plot is of stepQi, which goes from 0 to 1. Try to change the argument samples_after_Qi_step in the function preparing the data. How does it change stepQi?

Changing the stepQi makes the ON or OFF intervals longer or shorter and makes it able to correspond better to the actual heater's behavior. Like below with a step = 0.5




## Comparing the two models

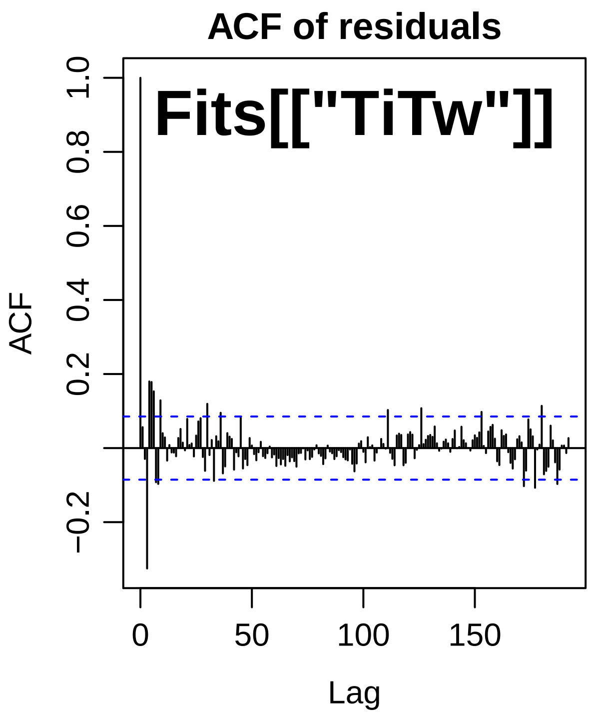
Now compare the two models implemented in **functions/sdeTiTw.R** and **functions/sdeTiTw_sigmalevels.R**. What is the difference?

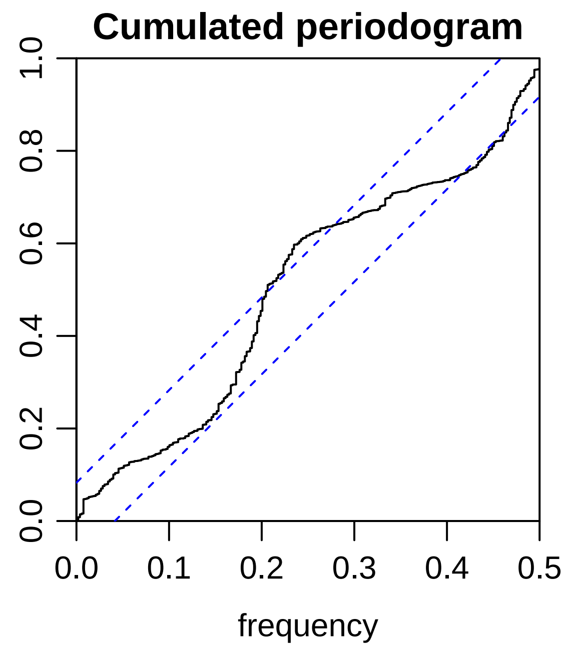
The second model includes with sigma levels include:

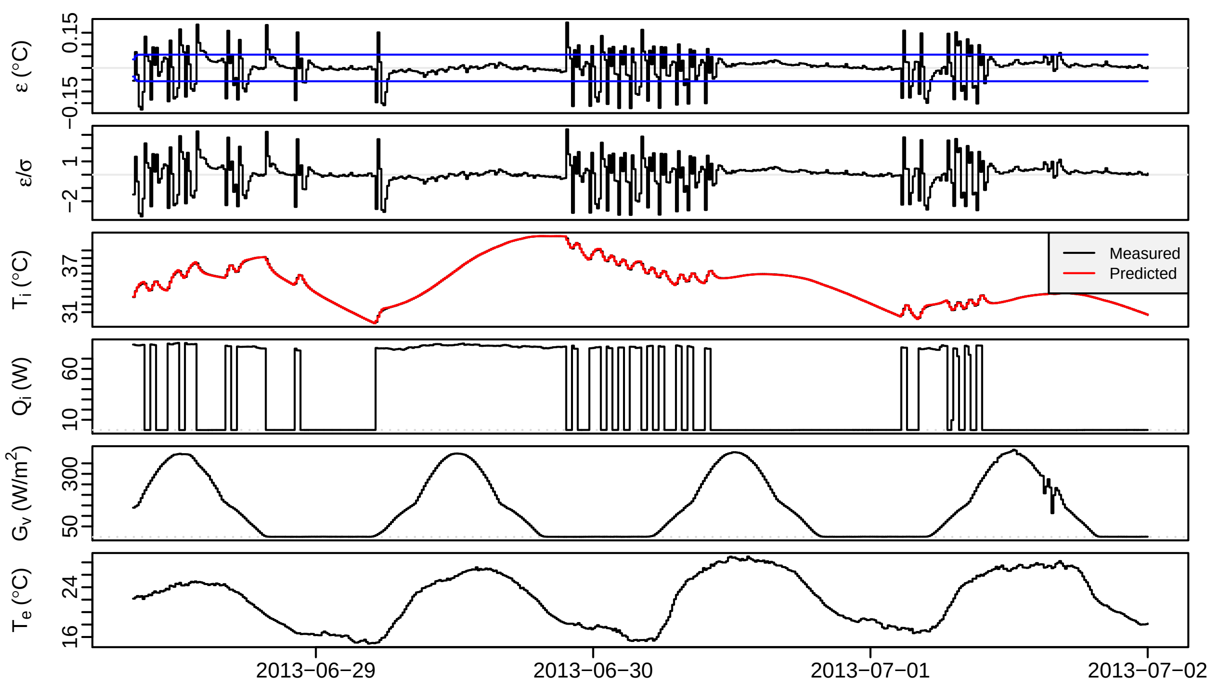
$(1 + (stepQi * sigmalevel))$

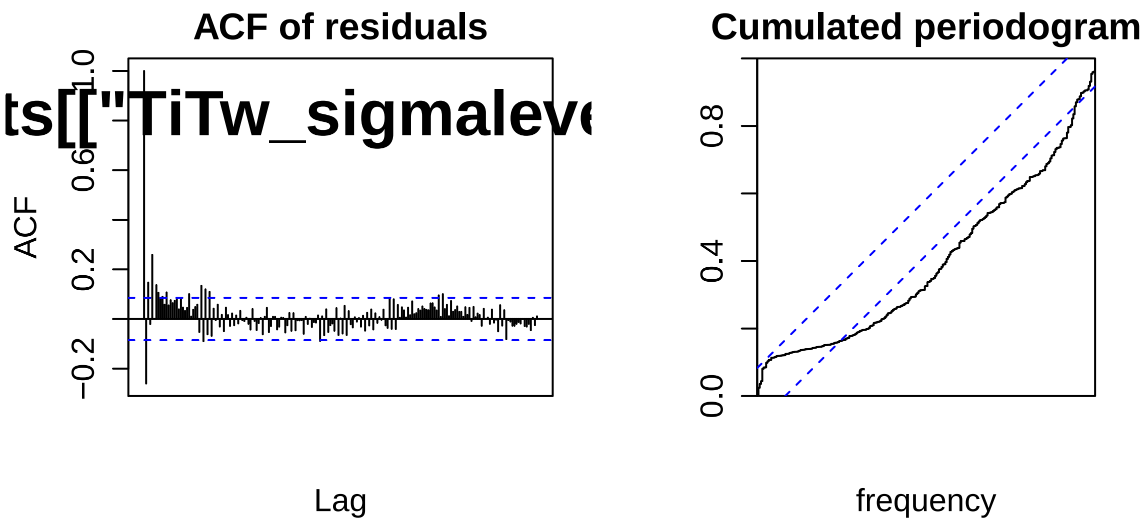
in the error term allowing to account for the noise of turning ON and OFF the heater.

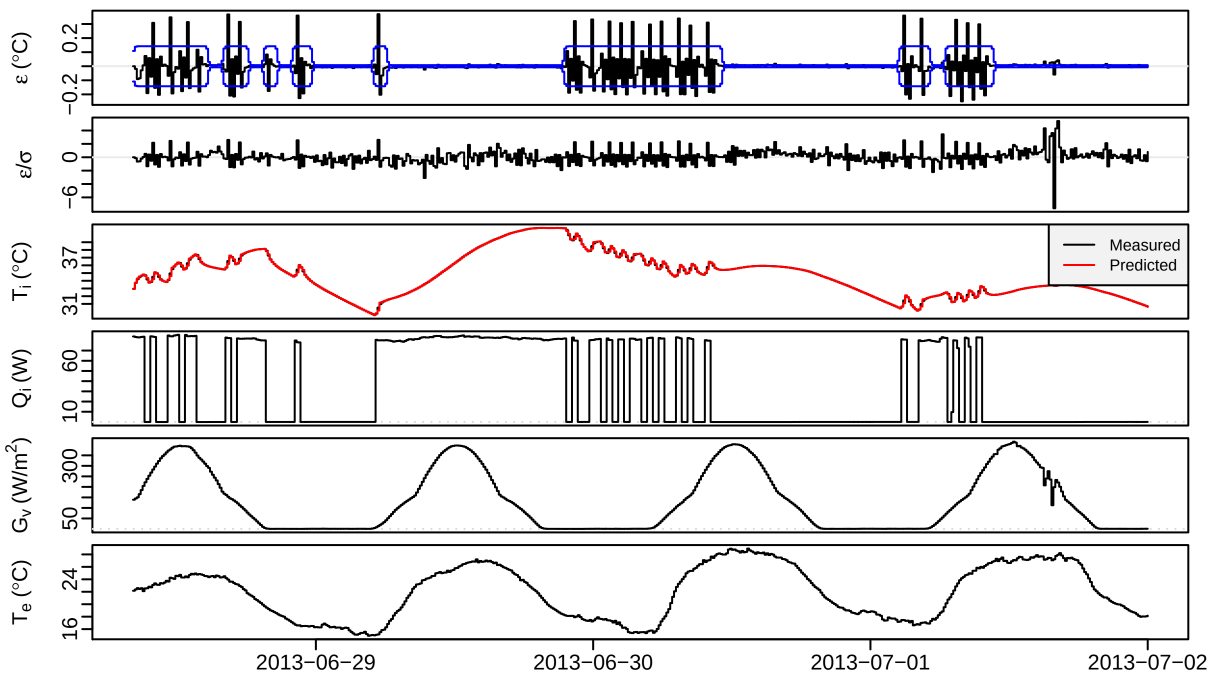
Now go to the script and fit the two models. Compare the results:


ACF of residuals
Fits[["TiTw"]]
ACF
Lag
0
50
100
150
−0.2
0.2
0.4
0.6
0.8
1.0


Cumulated periodogram
frequency
0.0
0.1
0.2
0.3
0.4
0.5
0.0
0.2
0.4
0.6
0.8
1.0


ε (°C)
ε/σ
Ti (°C)
Qi (W)
Gv (W/m2)
Te (°C)
Measured
Predicted
2013−06−29
2013−06−30
2013−07−01
2013−07−02




What is plotted in the upper two plots? (You maybe have to look into the analyzeFit() function).

The upper plot is the residuals of the predicted variable **yTi**. The lower plot is the standarized residuals (residuals / sd(yTi)), the y-axis changes as the errors are now divided by the standard deviation (sigma)

What is indicated by the blue lines in the upper plot? Step back in the plots and compare the results, and look at the summary output.

The blue lines in the upper plot are the standard deviation of the predicted variable (**yTi**) by which the residuals are divided to be standarized in the second graph below.

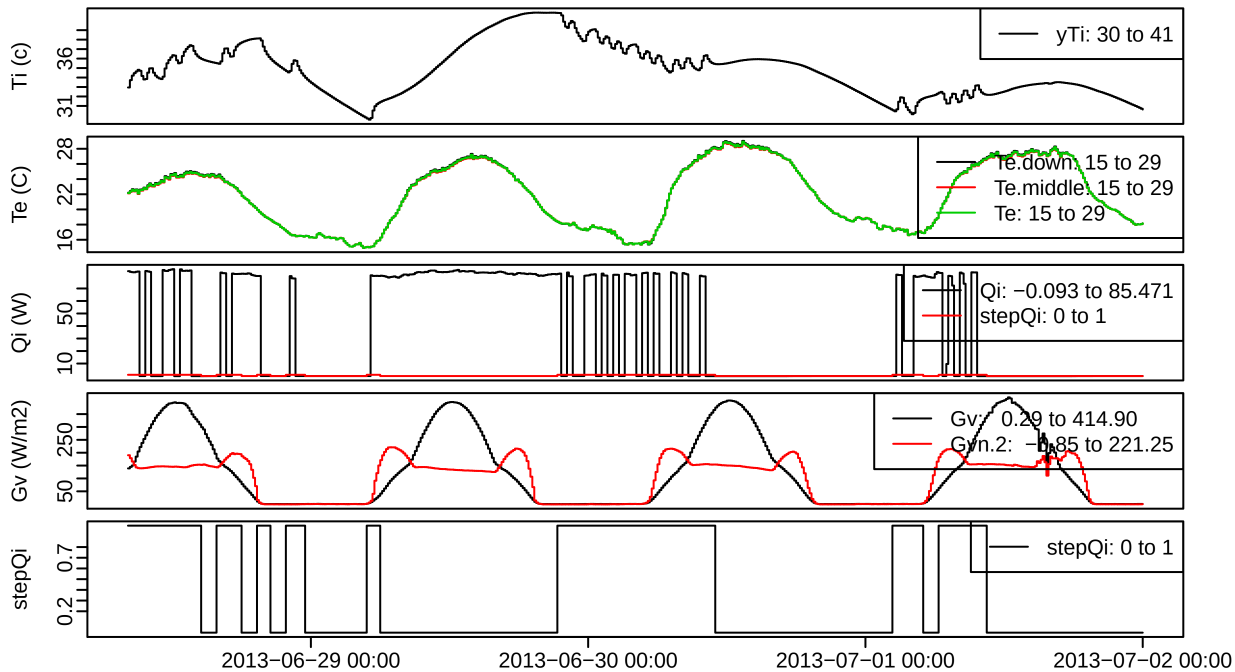
Step back in the plots and compare the results, and look at the summary output. Which of the two models will you prefer and why? The one with sigma level because it allows the variance to change with time and captures better the variability of the process and that is shown in its residuals.When comparing their Log-Likelihood, also the model with sigma-levels has a larger likelihood.

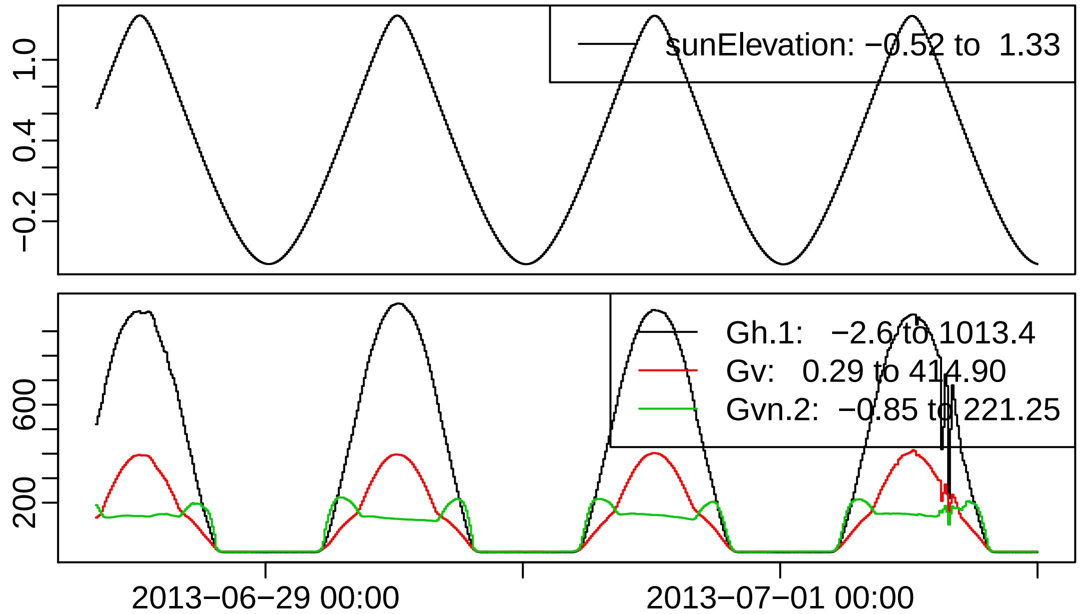
So it becomes clear that we have some (possible non-linear) dynamics when the heating turns on and off, which our models doesn't predict so well. But **instead of adding a more detailed description to the deterministic part of the model, we simply vary the system noise, or in other words, change the uncertainty level of our states under under different conditions**. This is a very useful thing, since there will be many phenomena in buildings, especially occupied buildings, which will lead different to levels of noise, e.g., solar radiation and occupants doing funny things.

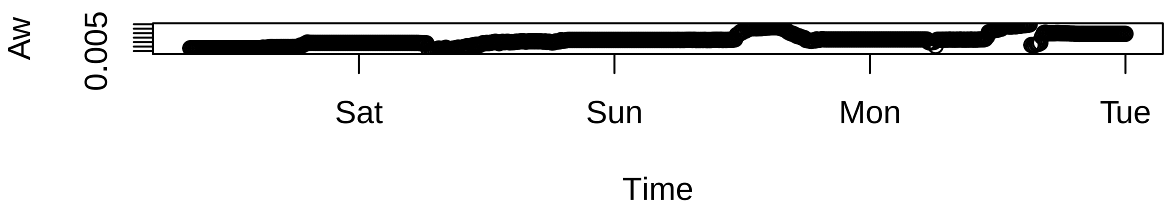
# Question 2

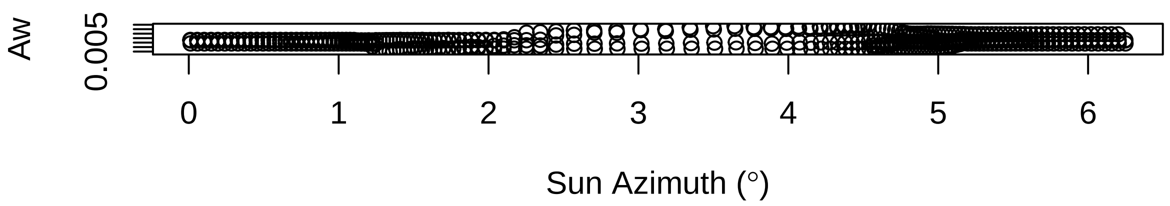
So far we have assumed that the solar gain is proportional to the radiation outside the test box. In reality, the heat gain from the sun depends highly on building geometry, surroundings, window properties, etc. In this part of the exercise, we will apply splines to estimate the solar heat gain as a function of solar position.

First we make a hidden state for $Aw$ (also called the gA-value) to investigate if it changes over time and as the function of the sun position.





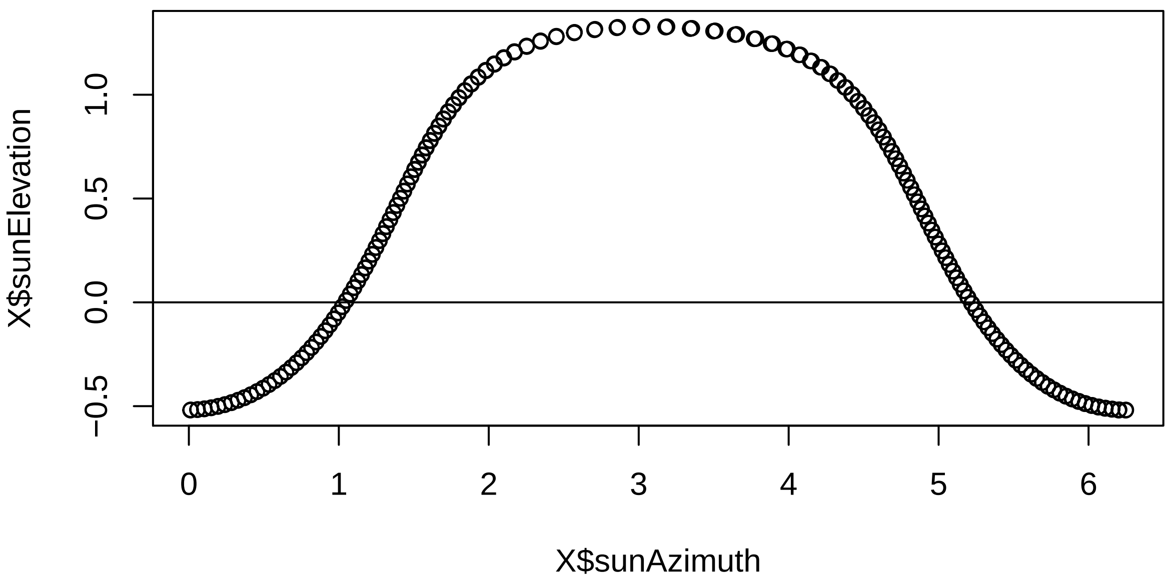



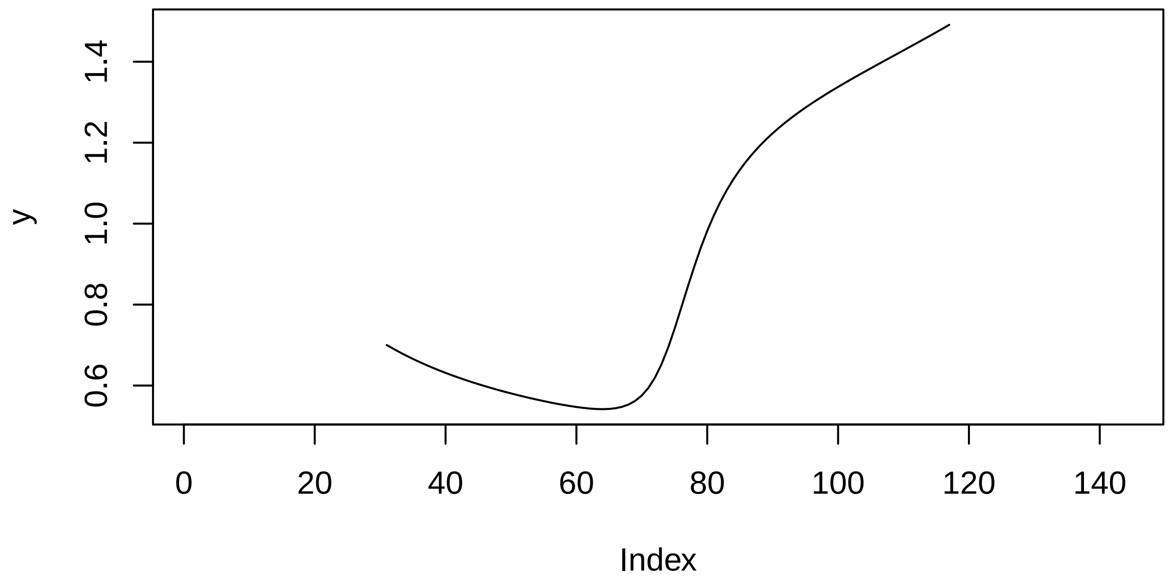

It is clear that the state of $Aw$ is not constant, but changes. Furthermore, it seems like there could be a relation to the sun azimuth.

Now, answer the questions below as you progress in modelling the solar radiation with use of splines. The

spline function we want to estimate is the **gA value** (e.g. the percentage of solar heat that enters through the window, multiplied with the window area) as a function of the sun azimuth.
First, plot the sun elevation as a function of the sun azimuth, as well as a horizontal line through 0 (notice that the angles is in radians).




Find the azimuth angles (in radians) that corresponds to the sunrise and sunset, and assign them to azumith_bound <- c(. . . , . . . ) below.




## References

JJ Allaire, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, and Winston Chang. *rmarkdown: Dynamic Documents for R*, 2018. URL https://CRAN.R-project.org/package=rmarkdown. R package version 1.10.

Peder Bacher and Henrik Madsen. Experiments and data for building energy performance analysis: Financed by the danish electricity saving trust. 2010.

Peder Bacher and Henrik Madsen. Identifying suitable models for the heat dynamics of buildings. *Energy and Buildings*, 43(7):1511–1522, 2011.

Dirk Eddelbuettel. *digest: Create Compact Hash Digests of R Objects*, 2018. URL https://CRAN.R-project.org/package=digest. R package version 0.6.15.

Keith R Godfrey. Correlation methods. In *System Identification*, pages 527–534. Elsevier, 1981.

Rune Juhl. *ctsmr: CTSM for R*, 2018. R package version 0.6.17.

Henrik Madsen. *Time series analysis*. Chapman and Hall/CRC, 2007.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018. URL https://www.R-project.org/.

Vitalie Spinu, Garrett Grolemund, and Hadley Wickham. *lubridate: Make Dealing with Dates a Little Easier*, 2018. URL https://CRAN.R-project.org/package=lubridate. R package version 1.7.4.

Hadley Wickham. *tidyverse: Easily Install and Load the 'Tidyverse'*, 2017. URL https://CRAN.R-project.org/package=tidyverse. R package version 1.2.1.

Yihui Xie. *knitr: A General-Purpose Package for Dynamic Report Generation in R*, 2018. URL https://CRAN.R-project.org/package=knitr. R package version 1.20.