Energy consumption

02418 Statistical modelling– Anders Hørsted (s082382)

24-10-2011

In this case models for the heat consumption of houses are build. The models are fitted to a data set containing measurements of heat consumption along with 4 other environmental variables as well as the date of the measurement for 16 different houses. First a model is build that only uses data for a single house. Then this model is extended to try to make a general model that includes all 16 houses.

Model for one house

First a model for a single house is developed. The model will be based on the house with `id=5` and as a first step a plot of the heat consumption as function of the date is plotted in figure 1. As seen the heat consumption Q depends primary on the season with large Q in the winter and small Q in the summer.

For the heat loss across a wall we have that

$$Q_w = U_a(T_a - T_i)$$

where T_a is the ambient temperature, T_i the indoor temperature and U_a is the response from temperature differences. We regard T_i as a constant and therefore get the simple relationship $Q_w = U_a T_a + k$. If heat loss across the wall was the only way that energy could flow in/out of the house, a great fit for the data should be obtained by the linear model `Q~Ta`.

Fitting the simple model for one house

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	3.9542	0.0270	146.56	0.0000
Ta	-0.2123	0.0024	-87.41	0.0000

Table 1: Summary table for the simple model for house 5

Fitting the simple model `Q~Ta` in R gives the output shown in table 1 and the diagnostic plots in figure 2. From the plots the residuals are seen to be approximately normal distributed, but the variance do not seem to be independent of the fitted values. The

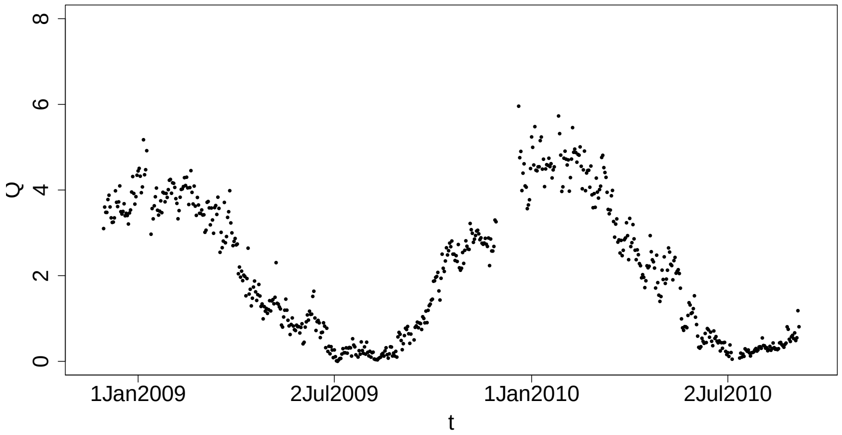

Figure 1: Heat consumption for house 5.

primary problem is that our model makes negative predictions which doesn't make sense since Q is the energy consumption of the house. This is also seen by the fact that the residuals for negative fitted values lies on an approximately straigth line with a slope of -1. Therefore all values that are predicted as negative have an observed value close to 0. More problematic is it that the residuals for fitted values in the interval $(0, 2)$ seems to have a negative mean and variance smaller than for fitted values larger than 2. To get a idea of how the model can be extended the residuals are plotted against the predictors. The most interesting predictor turn out to be T_a and this plot is shown in figure 3.

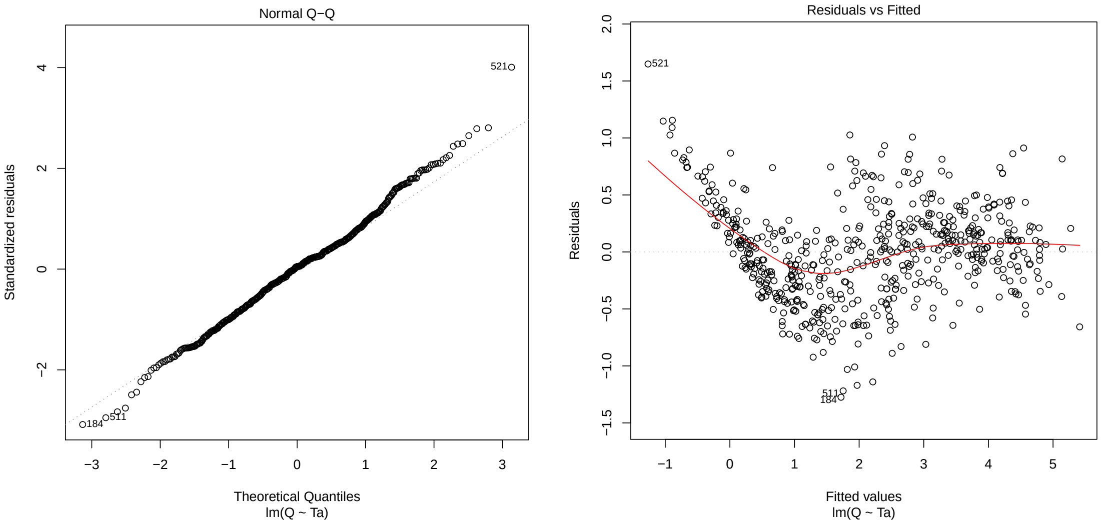

Figure 2: Diagnostic plots for simple model for house 5

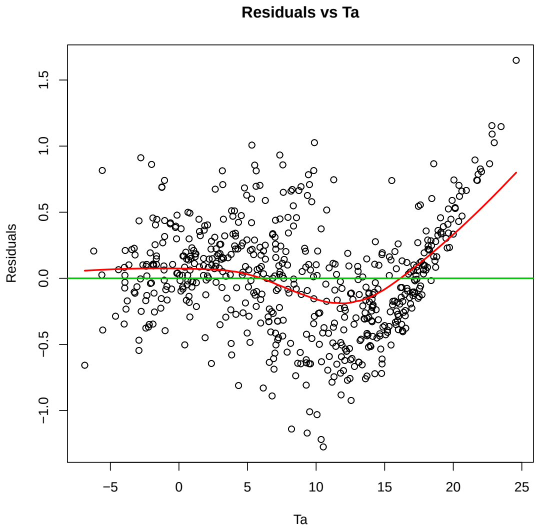

Figure 3: Residuals vs T_a for simple model for house 5

Fitting a higher order polynomial in T_a

From the plot of the residuals vs T_a it is seen that for high values of T_a the residuals are consistently above 0. A way to counteract this is by fitting a higher order polynomial in T_a. After testing a few different degrees a 3 degree polynomial seems to be of the right complexity. The model is defined as `Q~poly(Ta, 3, raw=TRUE))` which gives the coefficients in table 2 and the diagnostic plots in figure 4. It is seen that the residuals vs fitted values shows much less structure now which is as wished. Also the model no longer predicts negative values of Q which is great. On the other hand the residuals begins to be s-shaped in the QQ-plot which indicates a distribution with 'fat tails' compared to the normal distribution. The conclusion is that the 3 degree polynomial model do seem to fit the data well but the deviation from normallity shown in the QQ plot is unsatisfying. The next step is to use information from the other predictors.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	4.1276	0.0245	168.41	0.0000
poly(Ta, 3, raw = TRUE)1	-0.2035	0.0069	-29.53	0.0000
poly(Ta, 3, raw = TRUE)2	-0.0094	0.0010	-9.55	0.0000
poly(Ta, 3, raw = TRUE)3	0.0005	0.0000	13.08	0.0000

Table 2: Summary table for the third degree polynomial model for house 5

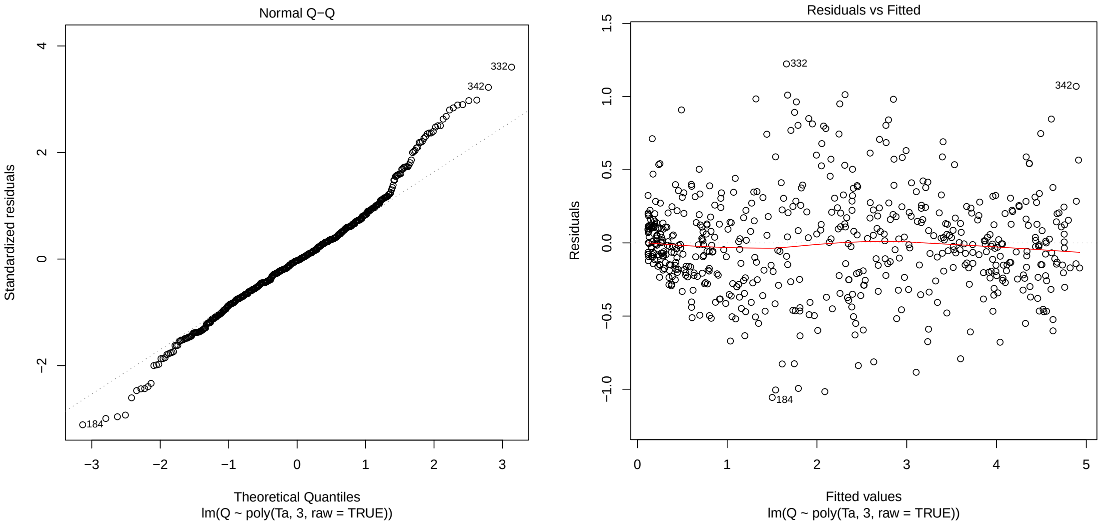

Figure 4: Diagnostic plots for the third degree polynomial model for house 5

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	4.4334	0.0525	84.46	0.0000
poly(Ta, 3, raw = TRUE)1	-0.1779	0.0067	-26.49	0.0000
poly(Ta, 3, raw = TRUE)2	-0.0070	0.0009	-7.75	0.0000
poly(Ta, 3, raw = TRUE)3	0.0004	0.0000	10.94	0.0000
Ws	0.0129	0.0118	1.09	0.2779
G	-0.0006	0.0002	-2.63	0.0088
sunElev	-1.4168	0.1933	-7.33	0.0000

Table 3: Summary table for the model in (1)

Adding predictors to the 3 degree polynomial model

Now the model given as

$$Q = \beta_1 T_a + \beta_2 T_a^2 + \beta_3 T_a^3 + \beta_4 W_s + \beta_5 G + \beta_6 S_{elev} + \beta_7 + \varepsilon \tag{1}$$

is fitted to the data. The coefficients are shown in table 3 and it is seen that some of the predictors might be redundant. To trim the model a stepwise selection is performed. The result is that W_s is dropped from the model. The model is then given as

$$Q = \beta_1 T_a + \beta_2 T_a^2 + \beta_3 T_a^3 + \beta_4 G + \beta_5 S_{elev} + \beta_6 + \varepsilon \tag{2}$$

When this model is fitted to the data the coefficients in table 4 are obtained. The diagnostic plots are shown in figure 5. From the QQ plot it seems as if the residuals are slightly closer to a normal distribution but it isn't very significant compared to the model with only a third degree polynomial in T_a. For now the predictors are kept in the model though and the next step is to look for possible interaction terms to include.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	4.4671	0.0424	105.47	0.0000
poly(Ta, 3, raw = TRUE)1	-0.1763	0.0066	-26.91	0.0000
poly(Ta, 3, raw = TRUE)2	-0.0072	0.0009	-8.05	0.0000
poly(Ta, 3, raw = TRUE)3	0.0004	0.0000	11.09	0.0000
G	-0.0006	0.0002	-2.62	0.0091
sunElev	-1.4239	0.1932	-7.37	0.0000

Table 4: Summary table for the model in (2)

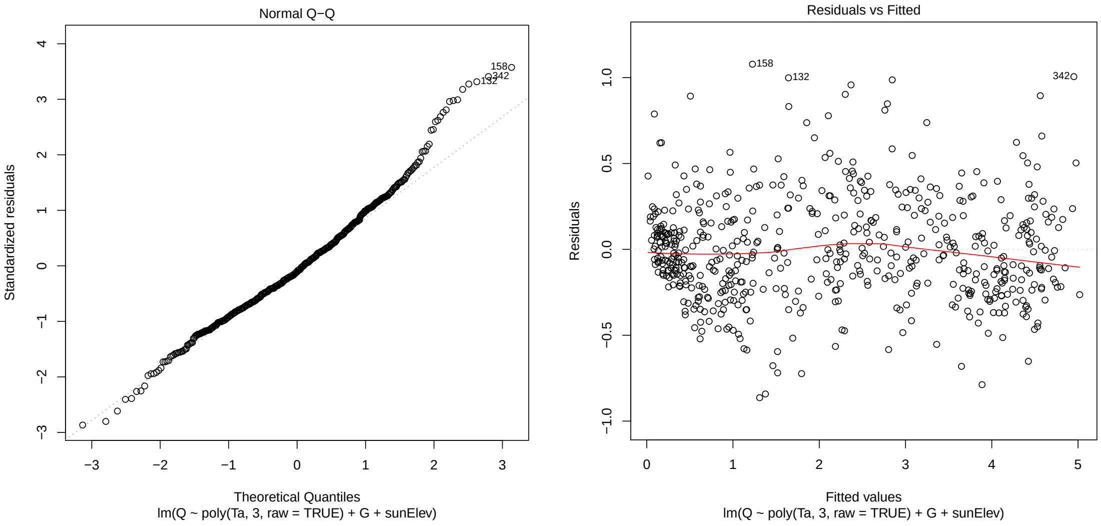

Figure 5: Diagnostic plots for the model in (2)

Including interaction terms

There are 4 predictor variables which gives 6 unique interaction pairs to test. One idea is to include all pairs in a model and then do a stepwise selection to get rid of redundant pairs. First a single interaction pair is tested though. The interaction between the ambient temperature and the wind speed might be interesting since low temperature combined with high wind speed could mean increased flow of cold air through windows. The model given by

$$Q = \beta_1 T_a + \beta_2 T_a^2 + \beta_3 T_a^3 + \beta_4 G + \beta_5 S_{elev} + \beta_6 W_s + \beta_7 T_a W_s + \beta_8 + \varepsilon \qquad (3)$$

Due to bad time management this is unfortunately how far I get before hand-in. The summary table and diagnostics plots for the model with interaction are shown in table 5 and figure 6.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	4.3764	0.0594	73.71	0.0000
poly(Ta, 3, raw = TRUE)1	-0.1710	0.0075	-22.78	0.0000
poly(Ta, 3, raw = TRUE)2	-0.0068	0.0009	-7.41	0.0000
poly(Ta, 3, raw = TRUE)3	0.0004	0.0000	10.51	0.0000
Ws	0.0378	0.0170	2.22	0.0268
G	-0.0006	0.0002	-2.58	0.0102
sunElev	-1.4215	0.1927	-7.37	0.0000
Ws:Ta	-0.0037	0.0018	-2.03	0.0424

Table 5: Summary table for the model in (3)

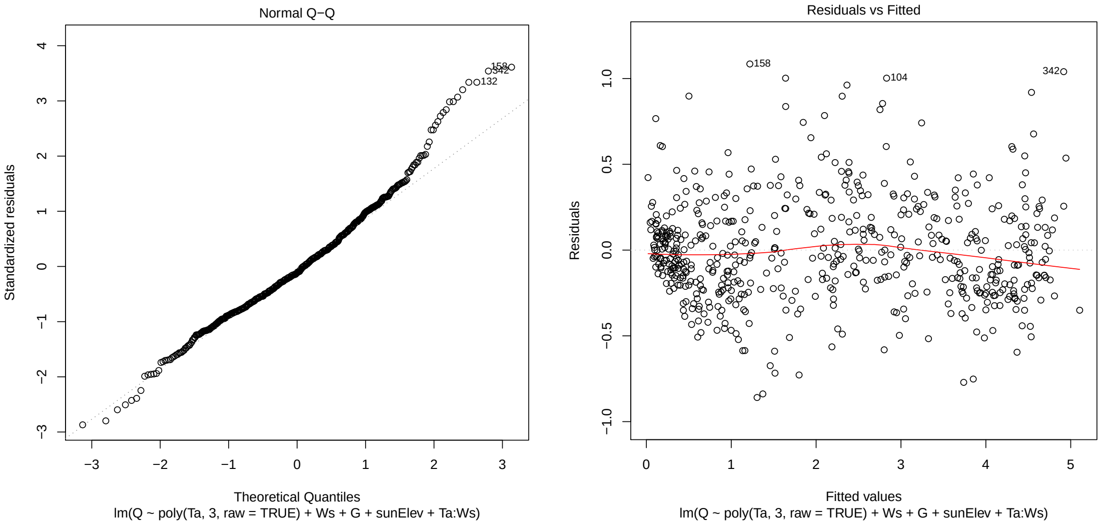

Figure 6: Diagnostic plots for the model in (3)

A Appendices

All R source code is included in the appendices. All the source code including the Latex code used for the report can also be found at `https://github.com/alphabits/dtu-fall-2011/tree/master/02418/heat-consumption`.

A.1 functions.R

```
plot.and.save = function(filename, width, height, plotfunction, ...) {
    save.the.plot = exists('SAVEPLOTS') && SAVEPLOTS
    if (save.the.plot) {pdf(sprintf('../plots/%s', filename), width, height)}
    plotfunction(...)
    if (save.the.plot) {dev.off()}
}

save.diagnostics = function(model, filenametmpl) {
    filename.summary = sprintf(filenametmpl, ’summary-xtable’)
    sink(sprintf(’../tables/%s.tex’, filename.summary))
    print(xtable(model))
    sink()

    filename.res = sprintf(filenametmpl, ’fit-res-plot’)
    pdf(sprintf(’../plots/%s.pdf’, filename.res))
    plot(model, which=1)
    dev.off()

    filename.res = sprintf(filenametmpl, ’qq-plot’)
    pdf(sprintf(’../plots/%s.pdf’, filename.res))
    plot(model, which=2)
    dev.off()

    filename.fit = sprintf(filenametmpl, ’fit-data-plot’)
    pdf(sprintf(’../plots/%s.pdf’, filename.fit))
    plot.model.with.data(model)
    dev.off()

    for (i in names(datH05)[4:7]){
        filename.fit = sprintf(filenametmpl, sprintf(’%s-data-plot’, i))
        pdf(sprintf(’../plots/%s.pdf’, filename.fit))
        plot(datH05[[i]], residuals(model), xlab=i, ylab=’Residuals’,
             main=sprintf(’Residuals vs %s’, i))
        panel.smooth(datH05[[i]], residuals(model), lwd=2)
        abline(h=0, col=3, lwd=2)
        dev.off()
    }
}

plot.model.with.data = function(model) {
    plot(t, Q, pch=21)
    lines(t, fitted(model))
}
```

A.2 loaddata.R

```
# Include dependecies
library(date)

if (!exists("dat")) {
    dat = read.table("../data/houseEnergy.txt", header=TRUE)
    dat$t = as.date(as.character(dat$t), order="ymd")
    dat$houseId = as.factor(sprintf("H%02i", dat$houseId))
    dat$daynum = (as.numeric(dat$t)-310) %% 365
    dat = na.omit(dat)
}

if (!exists("datH05")) {
    datH05 = subset(dat, houseId=="H05")
}
```

A.3 eda.R

```
# Include dependencies
source(’loaddata.R’)
source(’functions.R’)

# Should the plots be saved. Used by save.and.plot function
SAVEPLOTS = TRUE

attach(dat)

## Plotting the heat consumptions for all houses
png(’../plots/Q-house-grid.png’, 1600, 1000)
par(mfrow=c(4,4),mgp=c(2,0.7,0),mar=c(2,2,1,1))
for (i in levels(dat$houseId)){
    plot(Q~t, dat[dat$houseId==i,], xlim=c(17857,18520), ylim=c(0,8), cex=0.1)
    legend("topright", legend=i)
}
dev.off()

plot.and.save(’../plots/Q-h05.pdf’, 12, 7,
          plot, Q~t, datH05, xlim=c(17857,18520), ylim=c(0,8), cex=0.8,
          pch=20, cex.axis=1.8, cex.lab=1.8,
          main=’Heat consumption for house 5’)


detach("dat")
```

A.4 singlehouse.R

```
# Include dependencies
source(’loaddata.R’)
source(’functions.R’)
library(xtable)

attach(datH05)

# Fit models
m1 = lm(Q~Ta)
save.diagnostics(m1, ’one-house-m1-%s’)

m2 = lm(Q~poly(Ta, 3, raw=TRUE))
save.diagnostics(m2, ’one-house-m2-%s’)

m3 = lm(Q~poly(Ta, 3, raw=TRUE)+Ws+G+sunElev)
save.diagnostics(m3, ’one-house-m3-%s’)

m4 = step(m3, direction="both")
save.diagnostics(m4, ’one-house-m4-%s’)

m5 = lm(Q~poly(Ta, 3, raw=TRUE)+Ws+G+sunElev+Ta:Ws)
save.diagnostics(m5, ’one-house-m5-%s’)

# Not used
m6 = step(m5, direction="both")
```

```
m7 = lm(Q~poly(Ta, 3, raw=TRUE)+Ta:Ws + Ws)
m8 = lm(Q~Ta*Ws)
```

References

[1] N. H. Bingham & John M. Fry *Regression.* Springer-Verlag London, 1st Edition, 2010.