SMAP challenge 1

Benoit Fayolle

12 May 2018

loading and cleaning data

```
training<-read_delim(file = "../training_dataset_500.csv",delim=",")
```

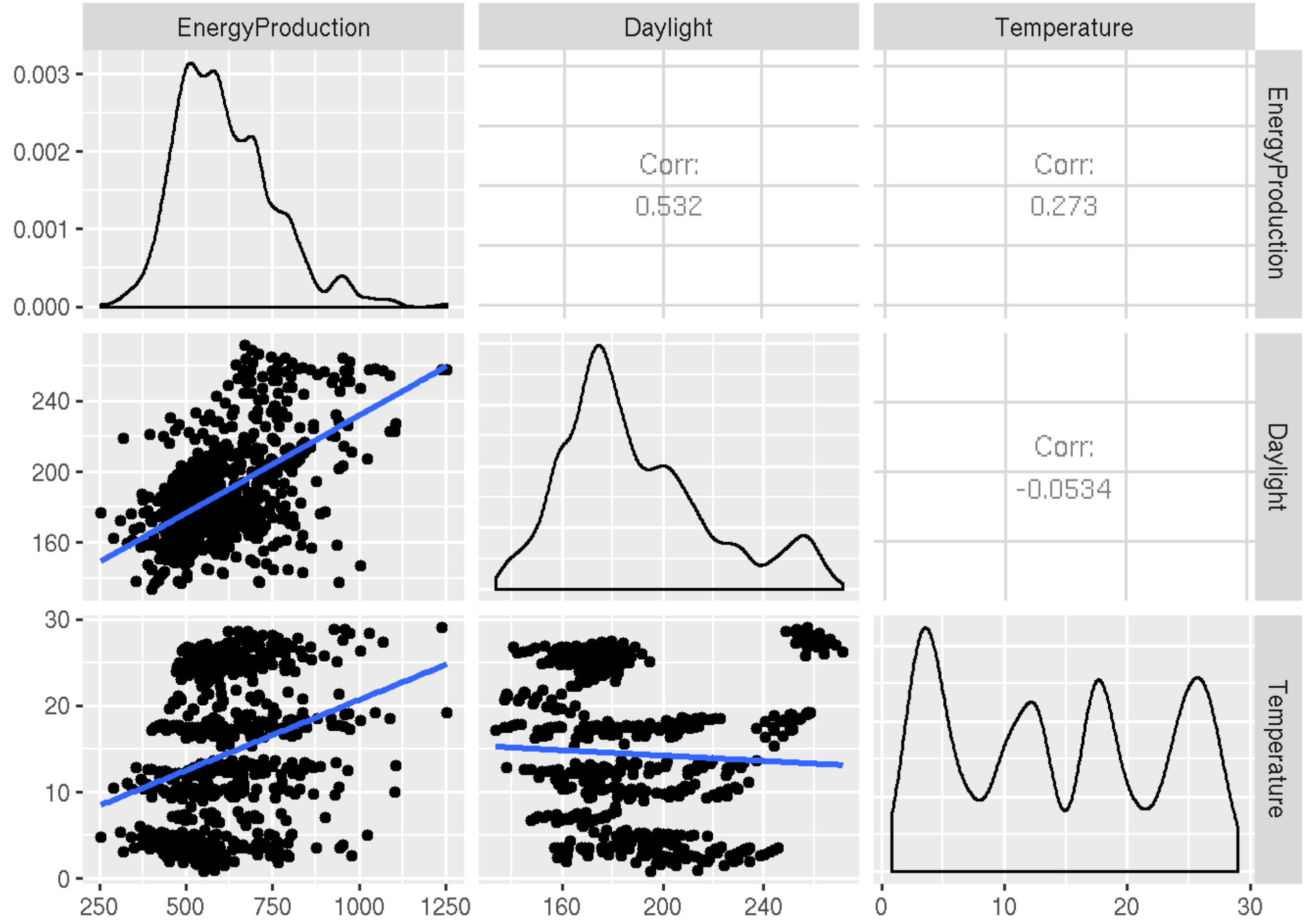
```
## Parsed with column specification:
## cols(
##   ID = col_integer(),
##   Label = col_integer(),
##   House = col_integer(),
##   Year = col_integer(),
##   Month = col_integer(),
##   Temperature = col_double(),
##   Daylight = col_double(),
##   EnergyProduction = col_integer()
## )
```

```
training$ym<-strftime(as.POSIXct(paste(training$Year,sprintf("%02d",training$Month),"01",sep="-"),format="%F",tz="GMT"),"%m/%y")
training<-training %>% group_by(House) %>% mutate(absmonth=1:length(Month)) %>% ungroup()
```

Exploratory Analysis

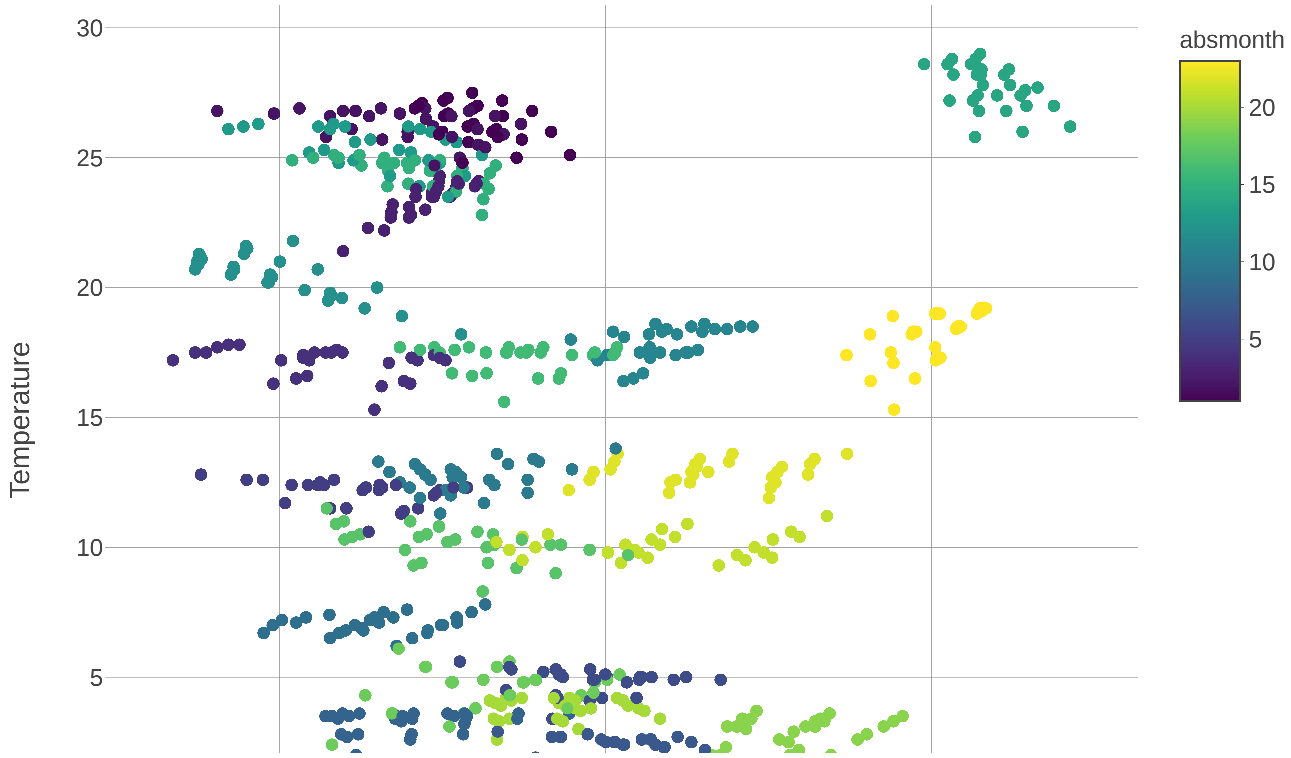
Daylight and Temperature are the only candidate covariates in the dataset

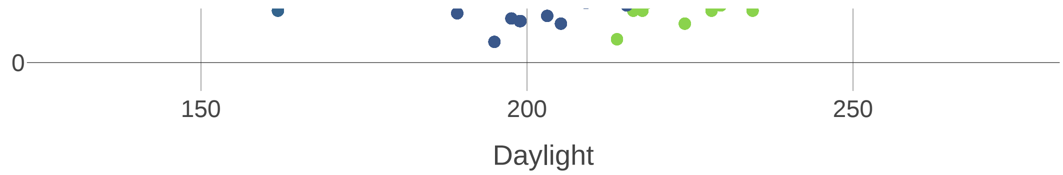
```
my_fn <- function(data, mapping, method="loess", ...){
  p <- ggplot(data = data, mapping = mapping) + geom_point() + geom_smooth(method=method, ...)
  p
}
ggpairs(select(training,EnergyProduction,Daylight,Temperature),lower=list(continuous=wrap(my_fn,method="lm")))
```



- Clear dependency between EnergyProduction and Daylight
- Less obvious for EnergyProduction and Temperature
- Unclear relation between Daylight and Temperature. Overall slightly negatively proportional
- Outliers for high temperatures (group of high temp and high daylight).

```
plot_ly(training,x=~Daylight,y=~Temperature,mode="markers",text=~ym,hoverinfo="House"
,color=~absmonth,type="scatter")
```

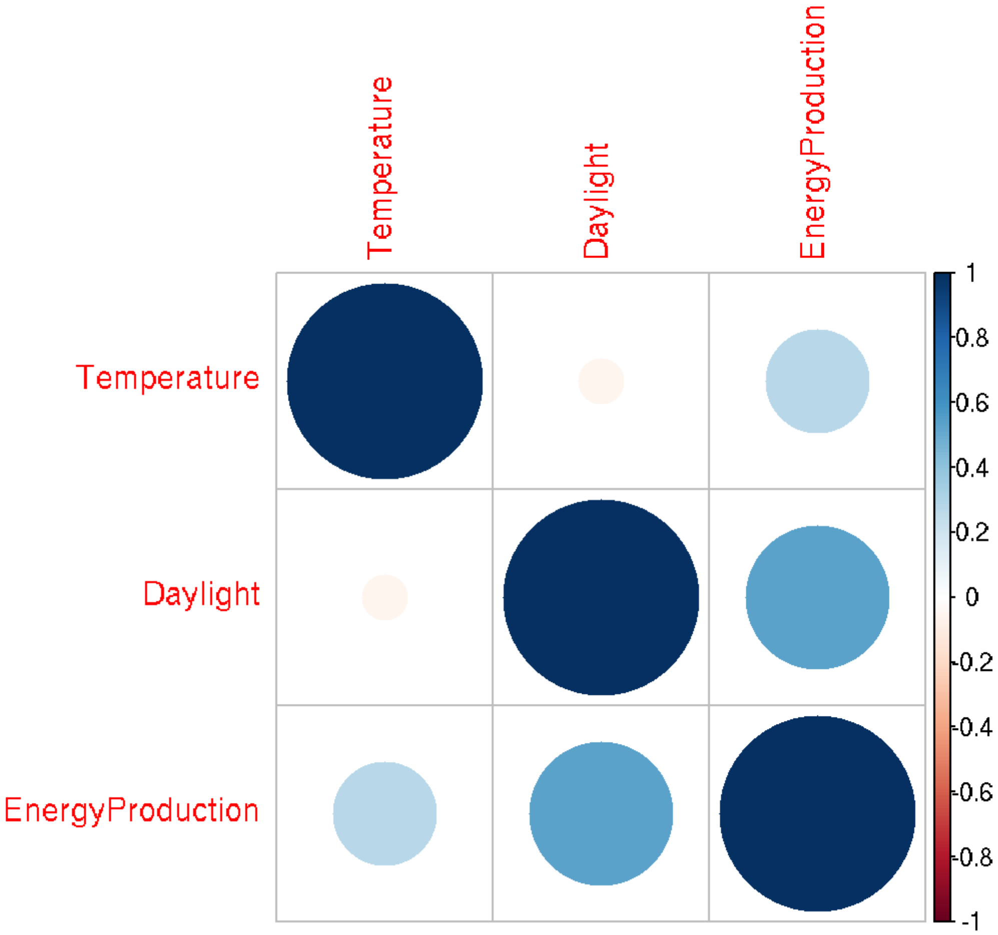




These outliers all originate from the month of august 2012

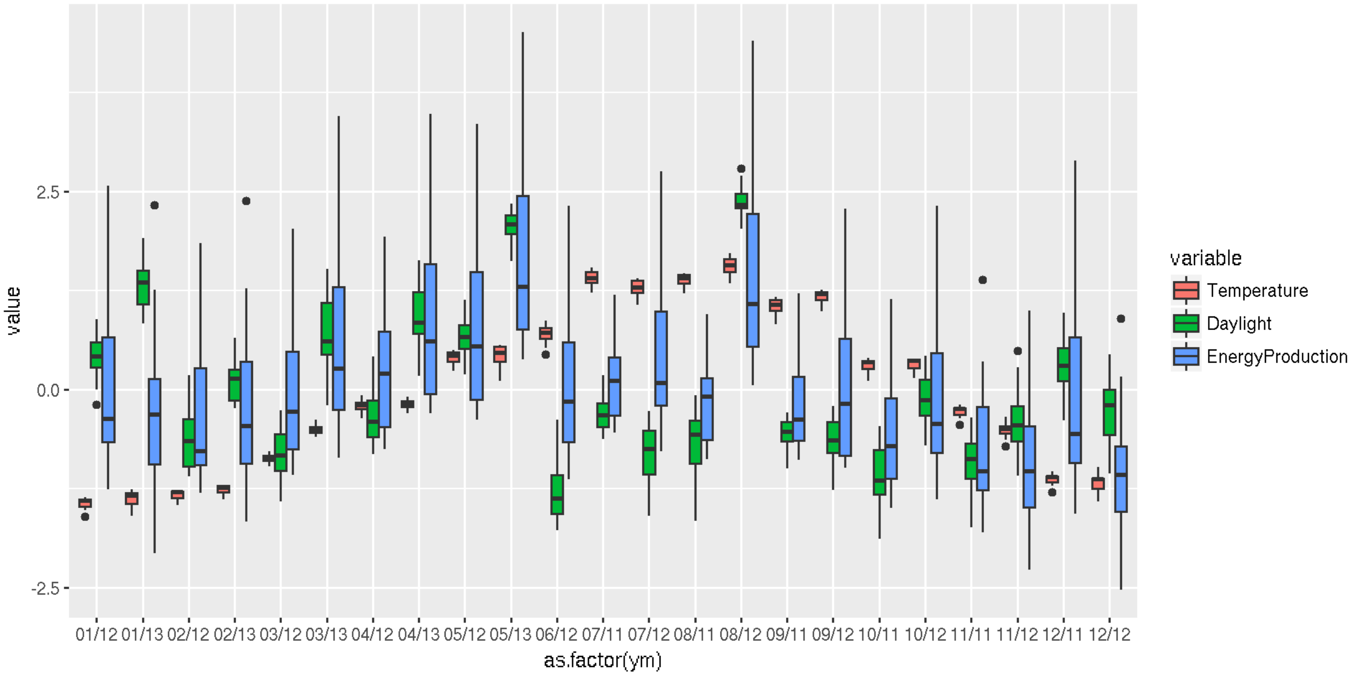
Above observations are strengtened by correlation plot:

```
M<-cor(select(training,Temperature,Daylight,EnergyProduction))
corrplot(M)
```



As well as time series plot:

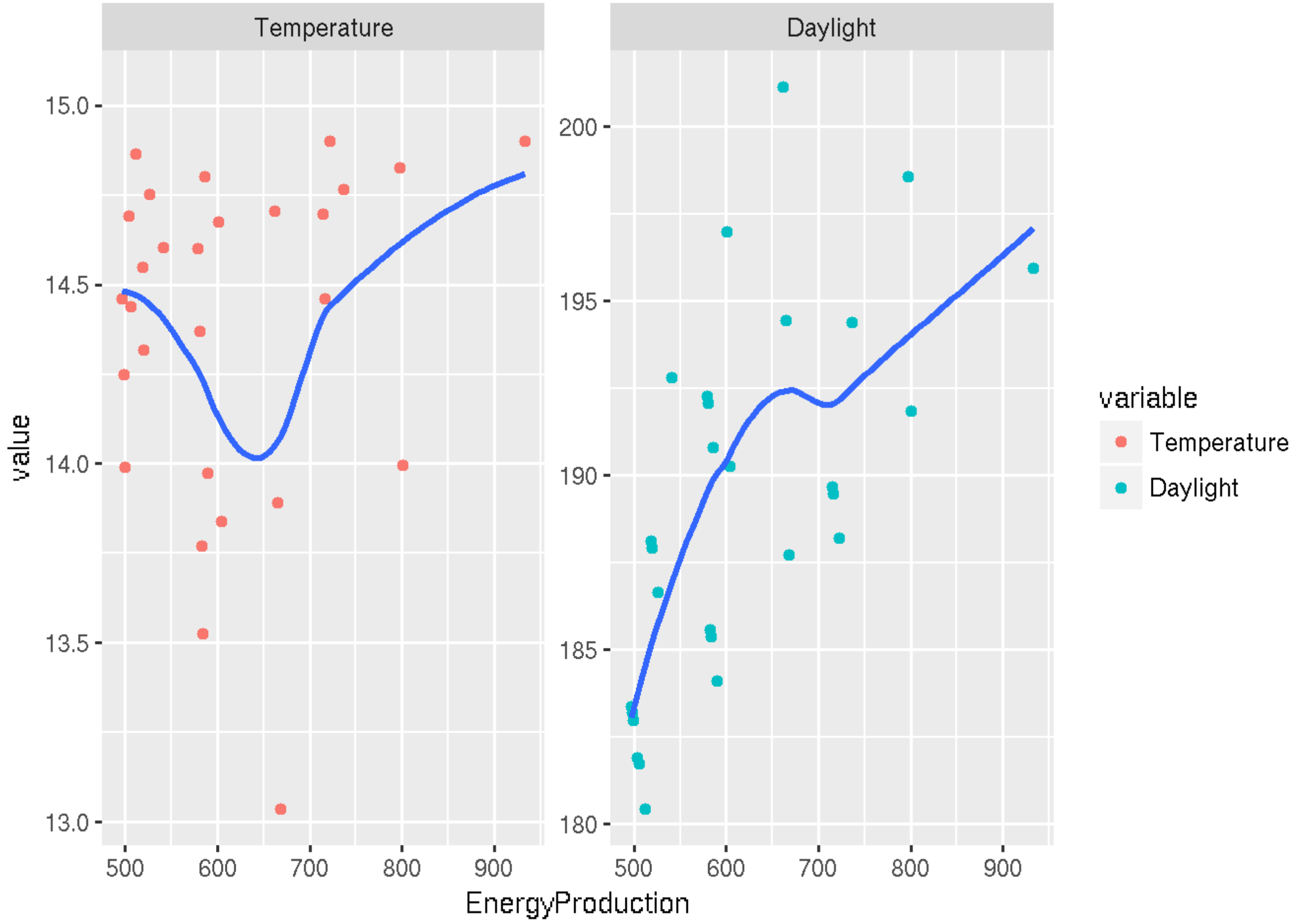
```
  DFplot<-select(training,ym,Daylight,Temperature,EnergyProduction)
  DFplot[,2:4]<-apply(DFplot[,2:4],2,scale)
  DFplot<-melt(DFplot,
               id.vars="ym",measure.vars=c("Temperature","Daylight","EnergyProductio
n"))
  DFplot<-DFplot[order(DFplot$ym),]
  ggplot(DFplot,aes(x=as.factor(ym),y=value))+geom_boxplot(aes(fill=variable))
```



- Temperature variance accross houses is small
- Energy Production and Daylight distribution across Houses are higher

```
DFplot<-melt(training %>% group_by(House) %>% summarise(Temperature=mean(Temperatur
e),Daylight=mean(Daylight),
                                                        EnergyProduction=mean(EnergyP
roduction)),
             id.vars="EnergyProduction",measure.vars=c("Temperature","Daylight"))

ggplot(DFplot,aes(x=EnergyProduction,y=value))+geom_point(aes(col=variable))+facet_wr
ap(~variable,scale="free_y")+geom_smooth()
```

```
## `geom_smooth()` using method = 'loess'
```



```
str(training)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    11500 obs. of  10 variables:
##  $ ID              : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ Label           : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ House           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Year            : int  2011 2011 2011 2011 2011 2011 2012 2012 2012 2012 ...
##  $ Month           : int  7 8 9 10 11 12 1 2 3 4 ...
##  $ Temperature     : num  26.2 25.8 22.8 16.4 11.4 4.2 1.8 2.8 6.7 12.6 ...
##  $ Daylight        : num  179 170 170 169 169 ...
##  $ EnergyProduction: int  740 731 694 688 650 763 765 706 788 831 ...
##  $ ym              : chr  "07/11" "08/11" "09/11" "10/11" ...
##  $ absmonth        : int  1 2 3 4 5 6 7 8 9 10 ...
```

Once again, more overall daylight for a house is associated with more overall Energy Production. Temperature influence is more complex

model selection

Linear regression

Most straight forward model is linear or multilinear regression. We want to try two different strategies:

- a local approach where we construct a model for each house. It would take into account the specificities of each house.
- a global approach where we train on the dataset disregarding the house variable

local approach

We use 10-fold cross-validation. We try daylight on model 2 as exploratory work showed Temperature influence on Energy production was not clear.

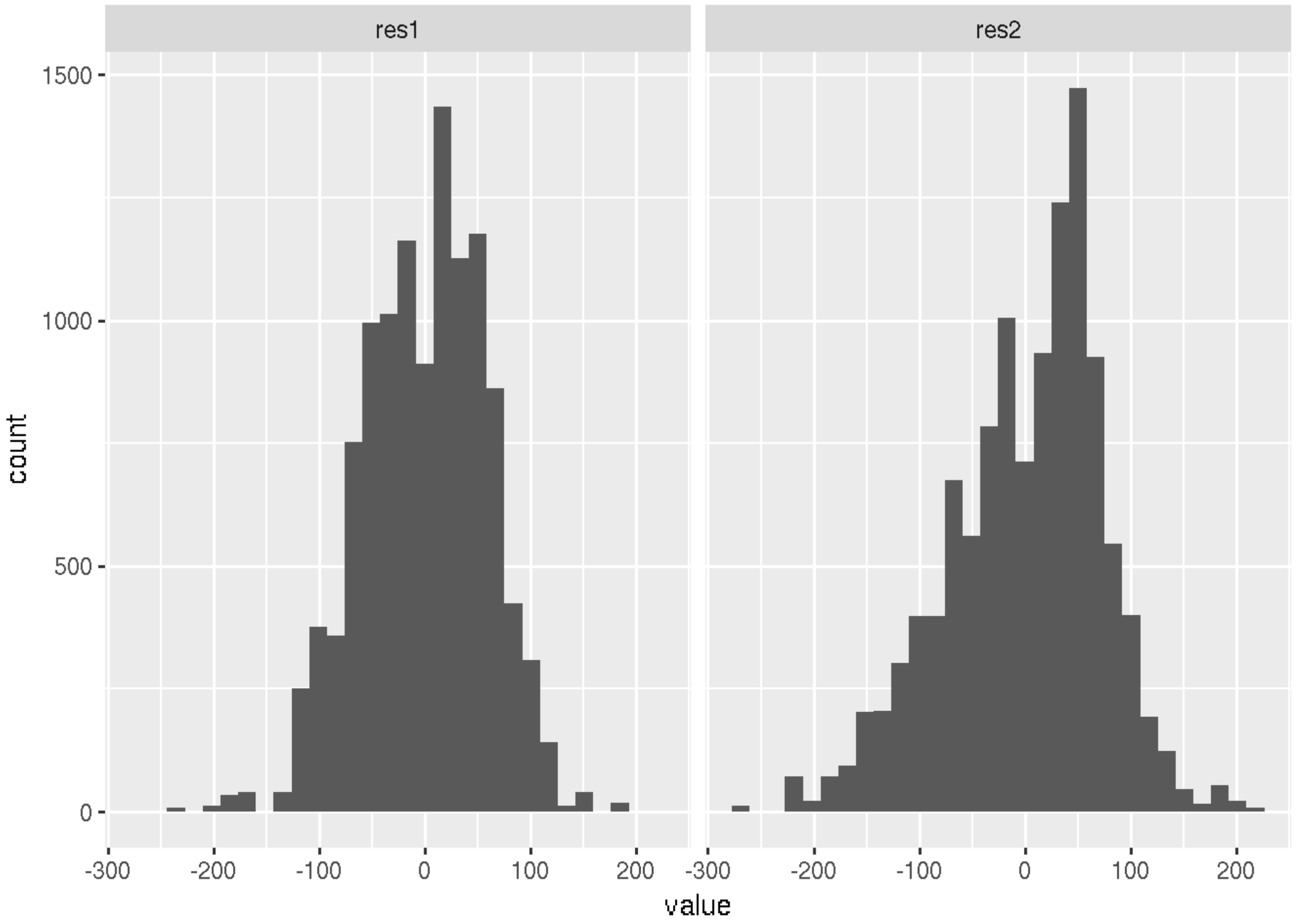
```
tryModels<-function(train){
  mod1<-train(EnergyProduction ~ Daylight+Temperature,data = train,method="lm",
              trControl=trainControl(method="cv",number=3,savePredictions=F,allowPara
llel = F))
  mod2<-train(EnergyProduction ~ Daylight,data = train,method="lm",
              trControl=trainControl(method="cv",number=3,savePredictions=F,allowPara
llel = F))
  data.frame(preds1=fitted.values(mod1),preds2=fitted.values(mod2),res1=residuals(mod
1),res2=residuals(mod2))
}

cl <- parallel::makeForkCluster(4)
doParallel::registerDoParallel(cl)

res<-foreach(i=1:500) %dopar%{
  tryModels(subset(training,House==i))
}
parallel::stopCluster(cl)

predsLMloc<-plyr::ldply(res,data.frame)[,1:2]
resLMloc<-plyr::ldply(res,data.frame)[,3:4]
MAPE_LMloc<-colSums(abs((training$EnergyProduction-predsLMloc)/training$EnergyProduct
ion))/nrow(training) * 100
MAPE_LMloc
```

```
##    preds1    preds2
##  8.281432 10.298170
```

```
qplot(value,data=melt(resLMloc),geom = "histogram",facets=~variable)
```

```
## No id variables; using all as measure variables
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Model 1 has normaly distributed residuals - Model 2 has skewed distribution to the right

Model 1 is then prefered ####global approach

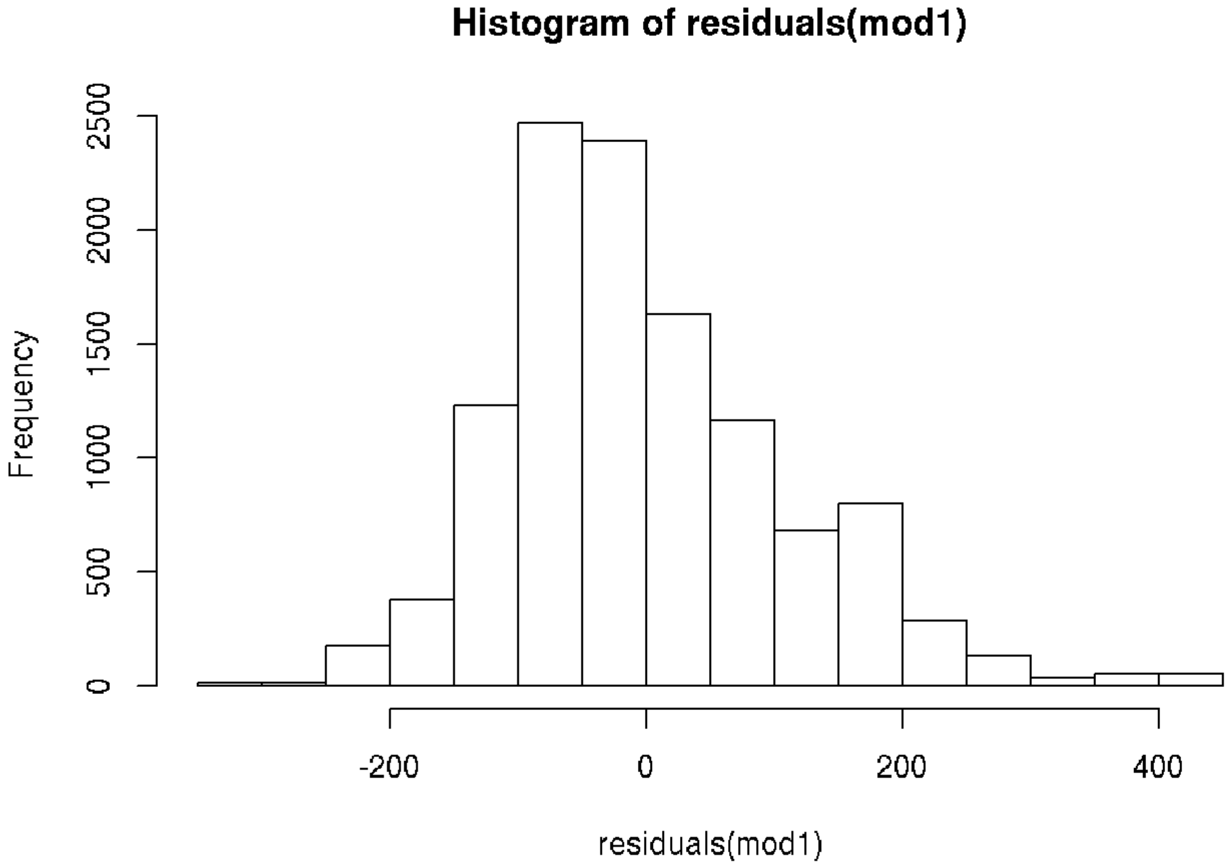
```
mod1<-train(EnergyProduction ~ Daylight+Temperature,data = training,method="lm",
            trControl=trainControl(method="cv",number=10,savePredictions=F,allowParal
lel = F))
summary(mod1)
```

```
## 
## Call:
## lm(formula = .outcome ~ ., data = dat)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -313.97  -76.07  -22.68   65.38  442.84 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 40.38919    7.14390   5.654 1.61e-08 ***
## Daylight     2.64255    0.03567  74.090  < 2e-16 ***
## Temperature  5.05111    0.12363  40.856  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 112.4 on 11497 degrees of freedom
## Multiple R-squared:  0.3735, Adjusted R-squared:  0.3734 
## F-statistic:  3427 on 2 and 11497 DF,  p-value: < 2.2e-16
```

```
predsLMg<-predict(mod1,training)

MAPE_LMg<-with(training,sum(abs((EnergyProduction-predict(mod1,training))/EnergyProdu
ction))/nrow(training) * 100)
MAPE_LMg
```

```
## [1] 14.65764
```

```
hist(residuals(mod1))
```



```
MAPE_LMg<-with(training,sum(abs((EnergyProduction-predict(mod1,training))/EnergyProdu
ction))/nrow(training) * 100)
MAPE_LMg
```

```
## [1] 14.65764
```

The residuals distribution is close to a normal distribution - although slightly skewed to the left ####conclusion The local approach gives us better MAPE. Taking house specificities into account seems to give better results.

Time Series Forecasting

We also try time series forecasting for two reasons: - trends or time dependant factors could play a role - We only want to predict the next month. Forecast can be good for short previsions

We here try forecasting only based on EnergyProduction variable. The simplest forecasting method is exponential smoothing. It weighs each point of the time series depending on how far in the past it is. The weight applied is represented by the parameter alpha. The higher alpha, the most weight is put on recent points. Function ses() estimates alpha based on the optimization of alpha on predicting earlier points in the time series. Here it is close to 0.8.

We train the model on the last point of the train set.

```
actual<-predsSES<-rep(NA,500)
for (i in unique(training$House)){
  trainn<-subset(training,House==i)
  dev<-tail(trainn,1);trainn<-trainn[-23,]
  actual[i]<-dev$EnergyProduction
  fit<-ses(y = trainn$EnergyProduction,h=1)
  predsSES[i]<-fit$mean
}
devMAPEses<-sum(abs((actual-predsSES)/actual))/500 * 100
devMAPEses
```

```
## [1] 15.24725
```

Not great results but interesting regarding the simplicity of the model

```
actual<-predsHW<-rep(NA,500)
for (i in unique(training$House)){
  trainn<-subset(training,House==i)
  trainn$ym<-as.POSIXct(paste(trainn$Year,sprintf("%02d",trainn$Month),"01",sep="-"),format="%F",tz="GMT")

  dev<-tail(trainn,1);trainn<-trainn[-23,]
  actual[i]<-dev$EnergyProduction
  fit<-hw(y = xts(trainn$EnergyProduction,order.by=as.yearmon(trainn$ym)),h=1,seasonal = "multiplicative")
  predsHW[i]<-fit$mean
}
devMAPEses<-sum(abs((actual-predsHW)/actual))/500 * 100
devMAPEses
```

```
## [1] 8.782207
```

Because the Holt-Winters method captures seasonality, it is able to predict relatively well the EnergyProduction. Daylight and temperature variations are of course associated with seasonality.

Conclusion and prediction on test data

We pick multiple linear regression as our model for two reasons : - it has slightly better training set MAPE than Holt-Winters forecasting - We believe it is more robust because no cross validation was performed on Holt-Winters method Holt Winters is interesting because it does not require covariates to predict. Put more simply, because it is forecasting

```
training<-read_delim(file = "../training_dataset_500.csv",delim=",")
```

```
## Parsed with column specification:
## cols(
##   ID = col_integer(),
##   Label = col_integer(),
##   House = col_integer(),
##   Year = col_integer(),
##   Month = col_integer(),
##   Temperature = col_double(),
##   Daylight = col_double(),
##   EnergyProduction = col_integer()
## )
```

```
test<-read_delim(file = "../test_dataset_500.csv",delim=",")
```

```
## Parsed with column specification:
## cols(
##   ID = col_integer(),
##   Label = col_integer(),
##   House = col_integer(),
##   Year = col_integer(),
##   Month = col_integer(),
##   Temperature = col_double(),
##   Daylight = col_double(),
##   EnergyProduction = col_integer()
## )
```

```
models<-mclapply(split(training,as.factor(training$House)),function(x){
  train(EnergyProduction ~ Daylight+Temperature,data = x,method="lm",
              trControl=trainControl(method="cv",number=3,savePredictions=F,allowPara
llel = F))
},mc.cores=4)

predsFinal<-mapply(FUN = predict,models,split(test,as.factor(test$House)),USE.NAMES =
 F)
finalMAPE<-sum(abs((test$EnergyProduction-predsFinal)/actual))/500 * 100
finalMAPE
```

```
## [1] 10.01736
```

Finale MAPE is 10%

An interesting further study would be to combine linear regression with forecasting to capture the trends in the data.