Statistical learning og programmering

1. Semesterprojekt. Antal ord:

Af Kenneth Gottfredsen, Eva Imad og Sanne Sørensen

2022-12-20

Indhold

The CRoss Industry Standard Process for Data Mining (CRISP-DM) is a process model that serves as the base for a data science process. It has six sequential phases:

- Business understanding – What does the business need?
- Data understanding – What data do we have / need? Is it clean?
- Data preparation – How do we organize the data for modeling?
- Modeling – What modeling techniques should we apply?
- Evaluation – Which model best meets the business objectives?
- Deployment – How do stakeholders access the results?

I første omgang indlæses `pacman::load`:

```
# Inden vi går i gang bruger vi pacman() til at installere og indhente relevante pa
pacman::p_load("tidyverse", "magrittr", "nycflights13", "gapminder",
               "Lahman", "maps", "lubridate", "pryr", "hms", "hexbin",
               "feather", "htmlwidgets", "broom", "pander", "modelr",
               "XML", "httr", "jsonlite", "lubridate", "microbenchmark",
               "splines", "ISLR2", "MASS", "testthat", "leaps", "caret",
               "RSQLite", "class", "babynames", "nasaweather",
               "fueleconomy", "viridis", "readxl", "timeDate", "tinytex",
               "ggbeeswarm", "palmerpenguins", "hms", "RColorBrewer", "boot",
               "openxlsx", "writexl", "pacman")
```

Import

I første omgang vil vi importere det datasæt vi har fået udleveret til eksamen:

```
# Indlæser datasæt og gemmer det nye datasæt i et objekt.
data1 <- read_excel("data/stud_exam_data.xlsx")
 # Dernæst undersøges strukturen i datasættet.
str(data1)
```

```
## tibble [152 x 4] (S3: tbl_df/tbl/data.frame)
##  $ date             : POSIXct[1:152], format: "2022-04-01" "2022-04-02" ...
##  $ efterspørgsel    : num [1:152] 367 361 376 47 367 402 416 355 283 454 ...
##  $ kammerjunkere    : chr [1:152] "0" "0" "0" "1" ...
##  $ forventet_l_lager: chr [1:152] "3" "3" "3" "3" ...
```

Nu har vi fået indlæst datasættet. Det næste skridt er at transformere de forskellige variable:

```
# I denne kode vil vi rekode og transformere de udvalgte variable så de stemmer ove

# 1 Vi bruger mutate() til at lave en ny kolonne ud fra data 1. Først laver vi en d

# 2 I den næste del anvendes mutate() til, at lave en kolonne der hedder dag, som b

# 3 Her bruger vi igen mutate() til at danne en ny weekend-variabel der hedder week

# 4 Måned, dag, kamjunk, forvent_lager og weekend_1 er alle kategoriske faktorer. F

data1 <- read_excel("data/stud_exam_data.xlsx") %>%
mutate(date = ymd(date), måned = factor(month(date)),
kamjunk = factor(kammerjunkere), forvent_lager = factor(forventet_l_lager)) %>% mutat
glimpse(data1)
```

```
## Rows: 152
```

```
## Columns: 7
## $ date              <dttm> 2022-04-01, 2022-04-02, 2022-04-03, 2022-04-04, 202~
## $ måned             <fct> april, april, april, april, april, april, april, apr~
## $ dag               <fct> fredag, lørdag, søndag, mandag, tirsdag, onsdag, tor~
## $ efterspørgsel     <dbl> 367, 361, 376, 47, 367, 402, 416, 355, 283, 454, 129~
## $ kamjunk           <fct> ja, ja, ja, nej, ja, ja, ja, ja, nej, ja, nej, ja, j~
## $ forvent_lager     <fct> høj, høj, høj, høj, høj, høj, høj, høj, høj, høj, hø~
## $ weekend_helligdag <fct> 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1~
```

```
# I denne kodechunk vil vi transformere den data vi har hentet fra vores apikald ti

# 2 Først bruger vi base_url og info_url til at anmode om vejrdata fra DMI's API. r



data2 <- as.data.frame(do.call(cbind, list_dmi))
data2 <- dplyr::select(data2, features.properties.observed, features.properties.value
rename(værdi = features.properties.value, parameter = features.properties.parameterId
målingstidspunkt = features.properties.observed) %>%
pivot_wider(names_from = parameter, values_from = værdi) %>%
mutate(målingstidspunkt = as_datetime(målingstidspunkt)) %>%
separate(målingstidspunkt, into = c('date', 'time'), sep = " ") %>%
filter(str_sub(time, 1, 4) == "12:0") %>%
mutate(date = as_date(date)) %>%
mutate(time = as.hms(time)) %>%
dplyr::select(-(temp_max_past12h:temp_min_past12h))
```

```
## Warning: `as.hms()` was deprecated in hms 0.5.0.
## Please use `as_hms()` instead.
```

```
glimpse(data2)
```

```
## Rows: 152
## Columns: 9
```

```
## $ date             <date> 2022-08-30, 2022-08-29, 2022-08-28, 2022-08-27, 2022~
## $ time             <time> 12:00:00, 12:00:00, 12:00:00, 12:00:00, 12:00:00, 12~
## $ temp_min_past1h  <dbl> 19.7, 16.9, 20.0, 19.9, 25.4, 24.3, 24.0, 23.7, 20.5,~
## $ humidity         <dbl> 47, 67, 79, 96, 58, 53, 40, 37, 47, 46, 56, 75, 52, 6~
## $ temp_dry         <dbl> 20.9, 17.9, 21.6, 20.0, 26.4, 24.5, 25.5, 24.4, 21.7,~
## $ temp_dew         <dbl> 9.0, 11.6, 17.8, 19.3, 17.5, 14.5, 11.1, 9.0, 10.0, 1~
## $ temp_max_past1h  <dbl> 21.0, 17.9, 21.6, 20.2, 26.7, 25.7, 26.1, 25.8, 23.2,~
## $ humidity_past1h  <dbl> 44, 72, 82, 97, 58, 53, 41, 39, 47, 46, 56, 75, 54, 5~
## $ temp_mean_past1h <dbl> 20.5, 17.4, 20.5, 20.0, 26.0, 24.9, 25.1, 24.4, 21.5,~
```

```
# Når man merger vha. leftjoin beholder man alle observationer i x.
data3 <- data1 %>%
left_join(data2, data1, by = c("date" = "date"))
dplyr::select(data3, date, time, weekend_helligdag, everything())
```

```
glimpse(data3)
```

```
## Rows: 152
## Columns: 15
## $ date             <dttm> 2022-04-01, 2022-04-02, 2022-04-03, 2022-04-04, 202~
## $ måned            <fct> april, april, april, april, april, april, april, apr~
## $ dag              <fct> fredag, lørdag, søndag, mandag, tirsdag, onsdag, tor~
## $ efterspørgsel    <dbl> 367, 361, 376, 47, 367, 402, 416, 355, 283, 454, 129~
## $ kamjunk          <fct> ja, ja, ja, nej, ja, ja, ja, ja, nej, ja, nej, ja, j~
## $ forvent_lager    <fct> høj, høj, høj, høj, høj, høj, høj, høj, høj, høj, hø~
## $ weekend_helligdag <fct> 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1~
## $ time             <time> 12:00:00, 12:00:00, 12:00:00, 12:00:00, 12:00:00, 1~
## $ temp_min_past1h  <dbl> 4.2, 4.0, 6.4, 3.4, 6.4, 7.6, 6.0, 5.3, 7.9, 7.8, 8.~
## $ humidity         <dbl> 36, 36, 38, 90, 44, 92, 94, 66, 44, 44, 47, 38, 53, ~
## $ temp_dry         <dbl> 5.0, 5.0, 8.0, 4.0, 7.2, 8.3, 6.1, 5.3, 9.1, 8.9, 9.~
## $ temp_dew         <dbl> -8.8, -8.9, -5.4, 2.5, -4.2, 7.2, 5.2, -0.6, -2.4, -~
## $ temp_max_past1h  <dbl> 5.7, 5.3, 9.1, 4.0, 7.6, 8.3, 7.7, 7.0, 9.3, 9.7, 10~
## $ humidity_past1h  <dbl> 38, 37, 41, 90, 45, 94, 91, 59, 45, 47, 49, 37, 52, ~
```

```
## $ temp_mean_past1h  <dbl> 4.9, 4.6, 7.5, 3.6, 7.0, 7.9, 6.9, 6.5, 8.6, 8.8, 9.~
```

```
data3 <- data3 %>%
  mutate(temp1 = lag(temp_max_past1h, 1),
         temp2 = lag(temp_max_past1h, 2),
         temp3 = lag(temp_max_past1h, 3),
         temp1 = if_else(is.na(temp1), 0, temp1),
         temp2 = if_else(is.na(temp2), 0, temp2),
         temp3 = if_else(is.na(temp3), 0, temp3),
         temp_gt25_3_dage = if_else(temp1 >= 25 & temp2 >= 25 & temp3 >= 25, 1, 0))
glimpse(data3)
```

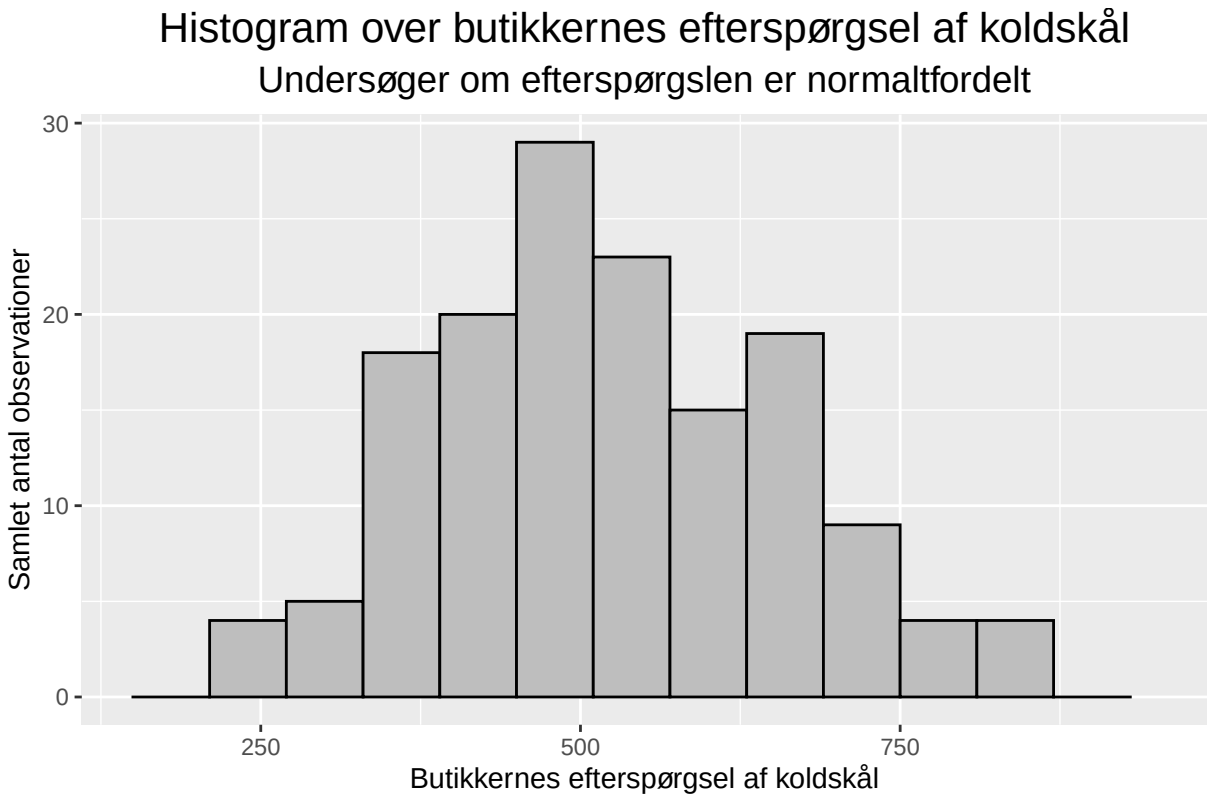
```
## Rows: 152
## Columns: 19
## $ date              <dttm> 2022-04-01, 2022-04-02, 2022-04-03, 2022-04-04, 202~
## $ måned             <fct> april, april, april, april, april, april, april, apr~
## $ dag               <fct> fredag, lørdag, søndag, mandag, tirsdag, onsdag, tor~
## $ efterspørgsel     <dbl> 367, 361, 376, 47, 367, 402, 416, 355, 283, 454, 129~
## $ kamjunk           <fct> ja, ja, ja, nej, ja, ja, ja, ja, nej, ja, nej, ja, j~
## $ forvent_lager     <fct> høj, høj, høj, høj, høj, høj, høj, høj, høj, høj, hø~
## $ weekend_helligdag <fct> 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1~
## $ time              <time> 12:00:00, 12:00:00, 12:00:00, 12:00:00, 12:00:00, 1~
## $ temp_min_past1h   <dbl> 4.2, 4.0, 6.4, 3.4, 6.4, 7.6, 6.0, 5.3, 7.9, 7.8, 8.~
## $ humidity          <dbl> 36, 36, 38, 90, 44, 92, 94, 66, 44, 44, 47, 38, 53, ~
## $ temp_dry          <dbl> 5.0, 5.0, 8.0, 4.0, 7.2, 8.3, 6.1, 5.3, 9.1, 8.9, 9.~
## $ temp_dew          <dbl> -8.8, -8.9, -5.4, 2.5, -4.2, 7.2, 5.2, -0.6, -2.4, -~
## $ temp_max_past1h   <dbl> 5.7, 5.3, 9.1, 4.0, 7.6, 8.3, 7.7, 7.0, 9.3, 9.7, 10~
## $ humidity_past1h   <dbl> 38, 37, 41, 90, 45, 94, 91, 59, 45, 47, 49, 37, 52, ~
## $ temp_mean_past1h  <dbl> 4.9, 4.6, 7.5, 3.6, 7.0, 7.9, 6.9, 6.5, 8.6, 8.8, 9.~
## $ temp1             <dbl> 0.0, 5.7, 5.3, 9.1, 4.0, 7.6, 8.3, 7.7, 7.0, 9.3, 9.~
## $ temp2             <dbl> 0.0, 0.0, 5.7, 5.3, 9.1, 4.0, 7.6, 8.3, 7.7, 7.0, 9.~
## $ temp3             <dbl> 0.0, 0.0, 0.0, 5.7, 5.3, 9.1, 4.0, 7.6, 8.3, 7.7, 7.~
## $ temp_gt25_3_dage  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

```r
# Vi skal lave en model der underfitter, en vi synes er sej og en der underfitter.
```

```r
# Først vil vi fjerne to outliers. Nemlig den observationer på 47 liter og 129.
data3 <- data3 %>%
filter(efterspørgsel < 47 | efterspørgsel > 129)
data3 # Observation 1 med 47 og 129 er væk. Der er i alt 150.
```

```r
ggplot(data = data3) +
geom_histogram(mapping = aes(x = efterspørgsel), color = "black", fill = "grey", binw
labs(title = "Histogram over butikkernes efterspørgsel af koldskål",
subtitle = "Undersøger om efterspørgslen er normaltfordelt",
y = "Samlet antal observationer",
x = "Butikkernes efterspørgsel af koldskål",
caption = "Kilde: Thise Mejeri 2022") +
ggeasy::easy_center_title() + # Centrerer titlen.
theme( plot.title = element_text(hjust = 0.5, size = 16),
plot.subtitle = element_text(hjust = 0.5, size = 14),
plot.caption = element_text(hjust = 1, face = "italic", size = 10)) +
xlim(150, 950)
```

```
## Warning: Removed 1 rows containing missing values (geom_bar).
```



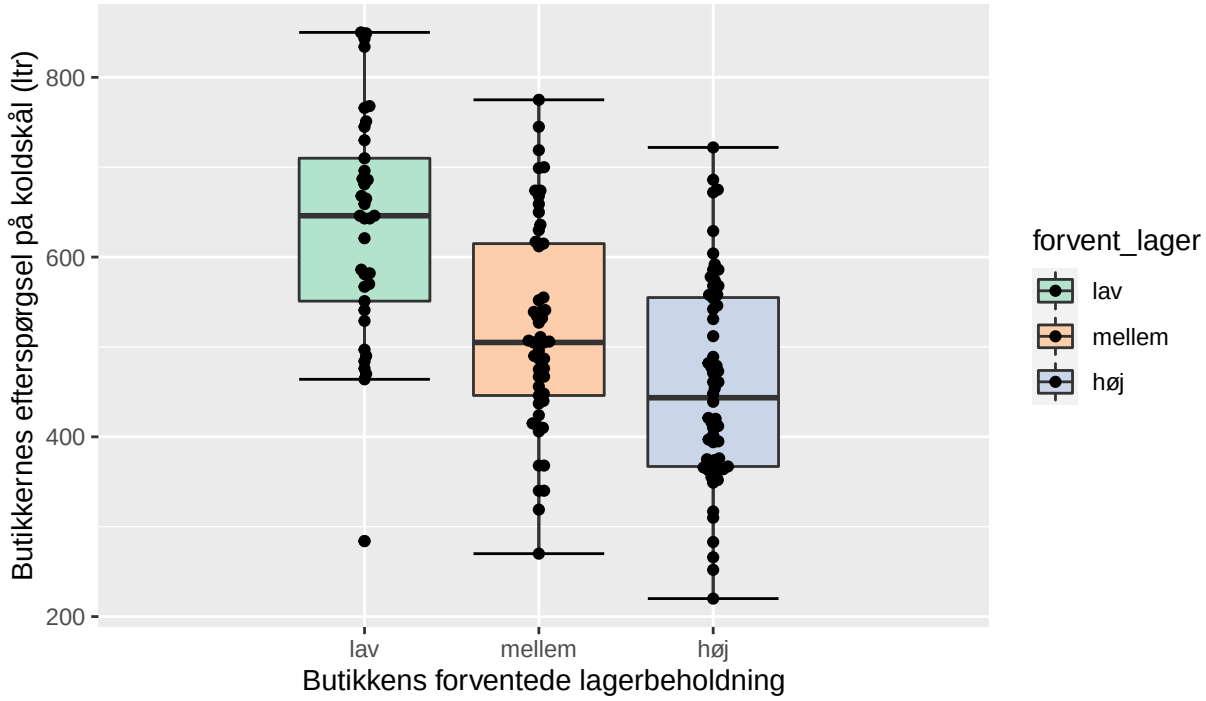
Kilde: Thise Mejeri 2022

```
ggplot(data = data3, mapping = aes(x = forvent_lager, y = efterspørgsel, fill = forve
stat_boxplot(geom = 'errorbar') + # whiskers.
geom_boxplot() +
labs(title = "Sammenhængen mellem lagerbeholdningen og efterspørgsel af koldskål",
subtitle = "Boxplox der viser variationen ift. den forventede lagerbeholdning og kold
caption = "Kilde: tal fra DMI 2002. Fra perioden 1/4/22-30/8/22",
y = "Butikkernes efterspørgsel på koldskål (ltr)",
x = "Butikkens forventede lagerbeholdning") + # Undgår overplotting
geom_beeswarm(dodge.width=3, cex =1, color = "black") + # Justerer boksbredden.
ggeasy::easy_center_title() + # Centrerer titlen.
theme( plot.title = element_text(hjust = 0.5, size = 14),
plot.subtitle = element_text(hjust = 0.5, size = 12),
plot.caption = element_text(hjust = 1.5, face = "italic", size = 10 )) + scale_fill_b
```

```
## Warning: position_dodge requires non-overlapping x intervals
```

mmenhængen mellem lagerbeholdningen og efterspørgsel af koldskål

Boxplox der viser variationen ift. den forventede lagerbeholdning og koldskål

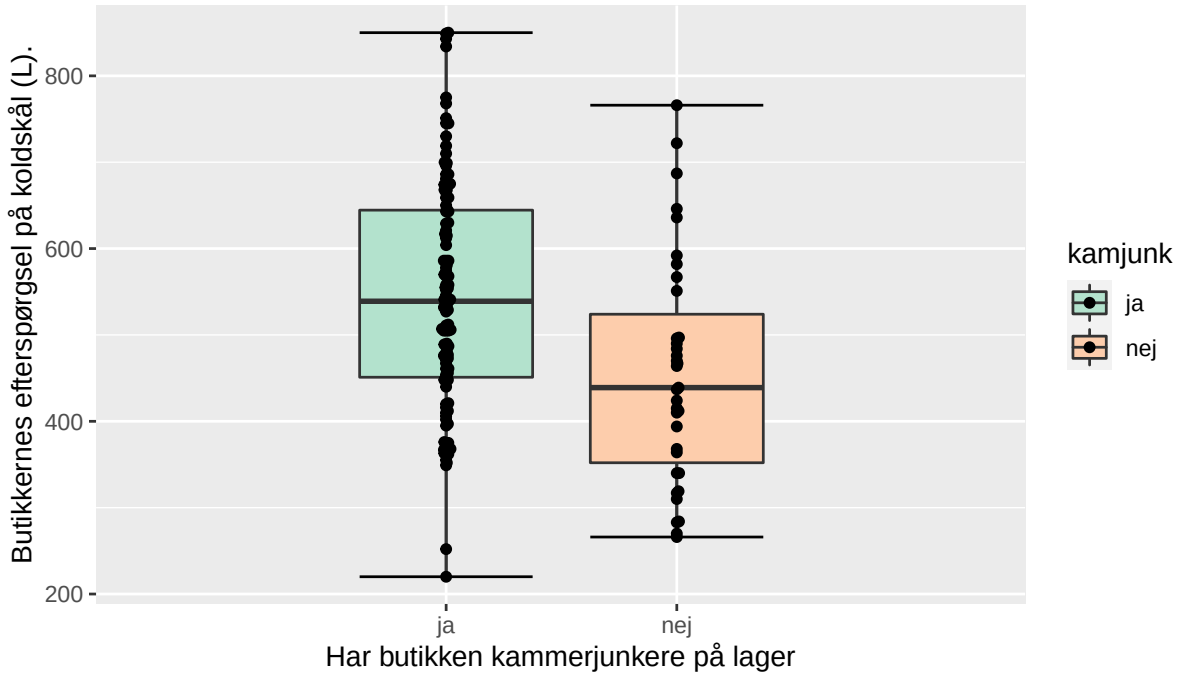


Kilde: tal fra DMI 2002. Fra perioden 1/4/22−30/8/22

```
# Undersøger spredningen af data
ggplot(data = data3, mapping = aes(x = kamjunk, y = efterspørgsel, fill = kamjunk)) +
stat_boxplot(geom = 'errorbar') +
geom_boxplot() +
labs(title = "Sammenhængen mellem lagerbeholdningen og efterspørgsel af koldskål",
subtitle = "Boxplox der viser variationen ift. den forventede lagerbeholdning og kold
caption = "Kilde: tal fra DMI 2002. Fra perioden 1/4/22-30/8/22",
y = "Butikkernes efterspørgsel på koldskål (L).",
x = "Har butikken kammerjunkere på lager") +
ggeasy::easy_center_title() + # Centrerer titlen.
geom_beeswarm(dodge.width=3,cex=0.5, color = "black") + # Justerer boksbredden.
theme( plot.title = element_text(hjust = 0.5, size = 16),
plot.subtitle = element_text(hjust = 0.5, size = 14),
plot.caption = element_text(hjust = 1.5, face = "italic", size = 10 )) + scale_fill_b
```

```
## Warning: position_dodge requires non-overlapping x intervals
```

nenhængen mellem lagerbeholdningen og efterspørgsel af koldskå
xplox der viser variationen ift. den forventede lagerbeholdning og koldskål



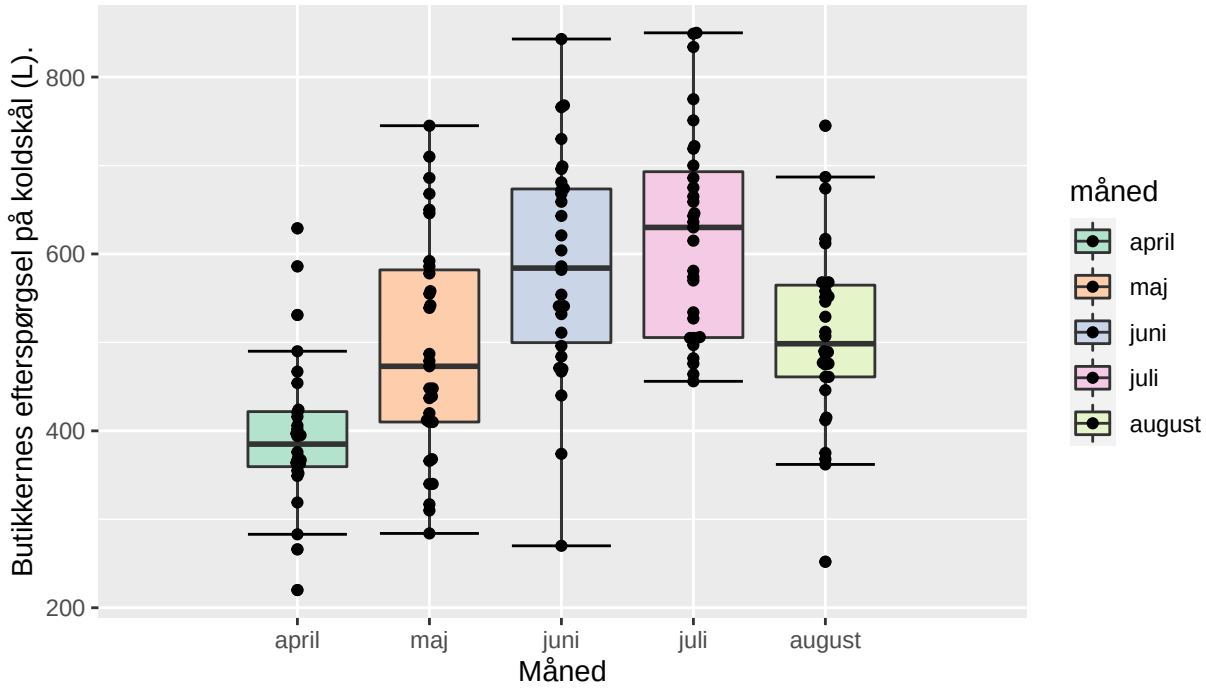
Kilde: tal fra DMI 2002. Fra perioden 1/4/22−30/8/22

```
ggplot(data = data3, mapping = aes(x = måned, y = efterspørgsel, fill = måned)) +
stat_boxplot(geom = 'errorbar') +
geom_boxplot() +
labs(title = "Sammenhængen mellem måned og efterspørgsel af koldskål",
subtitle = "Boxplox der viser variationen ift. den specifikke periode og koldskål",
caption = "Kilde: tal fra DMI 2002. Fra perioden 1/4/22-30/8/22",
y = "Butikkernes efterspørgsel på koldskål (L).",
x = "Måned") +
ggeasy::easy_center_title() + # Centrerer titlen.
geom_beeswarm(dodge.width=3,cex=0.5, color = "black") + # Justerer boksbredden.
theme( plot.title = element_text(hjust = 0.5, size = 16),
plot.subtitle = element_text(hjust = 0.5, size = 14),
plot.caption = element_text(hjust = 1.5, face = "italic", size = 10 )) + scale_fill_b
```

```
## Warning: position_dodge requires non-overlapping x intervals
```

Sammenhængen mellem måned og efterspørgsel af koldskål

Boxplox der viser variationen ift. den specifikke periode og koldskål



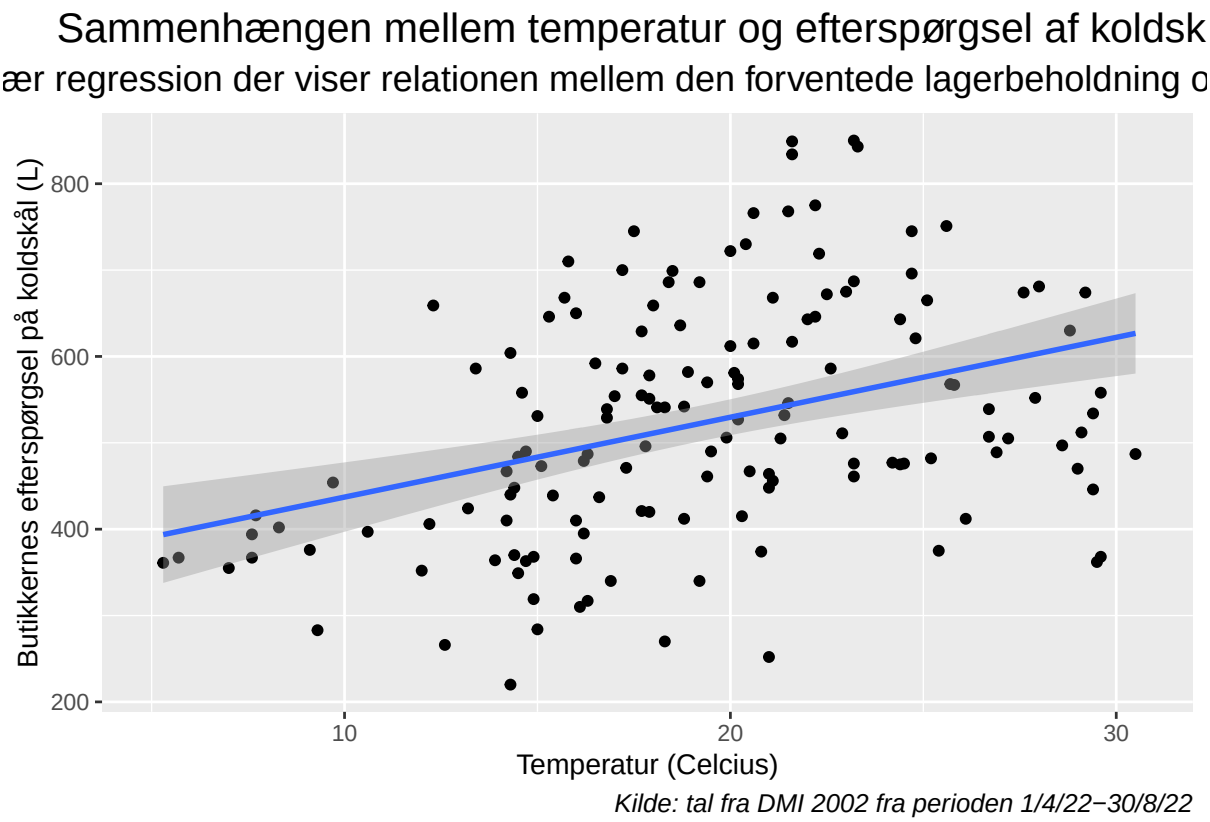
Kilde: tal fra DMI 2002. Fra perioden 1/4/22−30/8/22

```
# Basic scatter plot.
ggplot(data3, aes(x = temp_max_past1h, y = efterspørgsel)) +
geom_point() +
geom_smooth(method = lm, se = TRUE) +
labs(title = "Sammenhængen mellem temperatur og efterspørgsel af koldskål",
subtitle = "Linelær regression der viser relationen mellem den forventede lagerbehold
caption = "Kilde: tal fra DMI 2002 fra perioden 1/4/22-30/8/22",
y = "Butikkernes efterspørgsel på koldskål (L)",
x = "Temperatur (Celcius)") +
ggeasy::easy_center_title() + # Centrerer titlen.
theme( plot.title = element_text(hjust = 0.5, size = 16),
plot.subtitle = element_text(hjust = 0.5, size = 14),
plot.caption = element_text(hjust = 1, face = "italic", size = 10 ))+
xlim(5, 30.70) + ylim(220, 850) +
  scale_fill_brewer(palette = "Pastel2")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

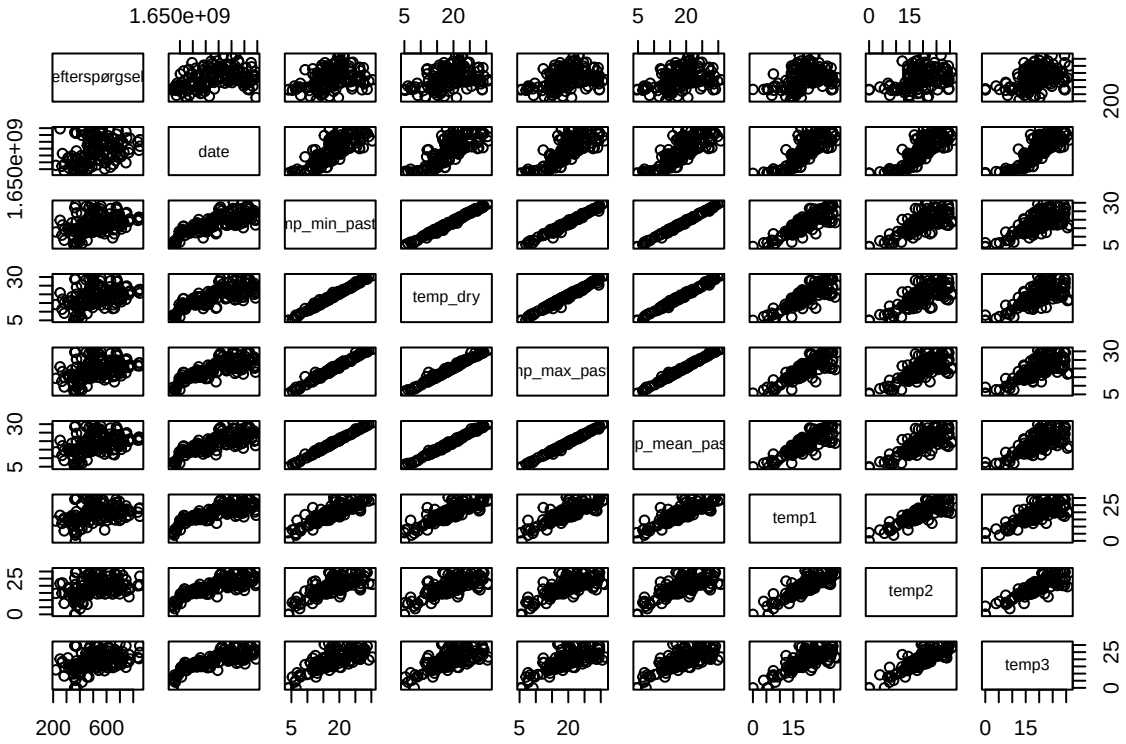


Kilde: tal fra DMI 2002 fra perioden 1/4/22−30/8/22

I dette afsnit vil vi gå i gang med analysen.

```
# LOOCV metoden er mindre biased end validation set metoden; hver gang vi fit'er en
```

```
cor_matrice <- data3 |>  dplyr::select(efterspørgsel, date, temp_min_past1h, temp_dry
plot(cor_matrice)
```




```
# Vi finder den model med den mindste MSE.
# n-fold=LOOVC

attach(data3)
lm.fit1 = lm(efterspørgsel ~ forvent_lager + weekend_helligdag + kamjunk + temp_gt25_
summary(lm.fit1)
```

```
##
## Call:
## lm(formula = efterspørgsel ~ forvent_lager + weekend_helligdag +
##     kamjunk + temp_gt25_3_dage + dag + temp_dry + temp_mean_past1h +
##     humidity_past1h + temp_max_past1h)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -221.484  -55.094   -5.611   61.093  202.873
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)          350.1534    69.8325   5.014 1.66e-06 ***
## forvent_lagermellem -106.4717    21.3102  -4.996 1.79e-06 ***
## forvent_lagerhøj    -138.5366    25.6921  -5.392 3.05e-07 ***
## weekend_helligdag1    93.6368    49.1033   1.907   0.0587 .
## kamjunknej           -71.1101    26.3685  -2.697   0.0079 **
## temp_gt25_3_dage    -113.8780    37.1083  -3.069   0.0026 **
## dagtirsdag            39.2619    38.0291   1.032   0.3037
## dagonsdag             29.1231    39.0244   0.746   0.4568
## dagtorsdag             3.5966    41.0026   0.088   0.9302
## dagfredag             63.8170    56.7860   1.124   0.2631
## daglørdag             29.5122    57.6743   0.512   0.6097
## dagsøndag             12.5491    56.9962   0.220   0.8261
## temp_dry             -14.5961    13.6972  -1.066   0.2885
## temp_mean_past1h       2.8638    24.9470   0.115   0.9088
## humidity_past1h        0.5827     0.5804   1.004   0.3172
## temp_max_past1h       20.9653    17.8515   1.174   0.2423
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 91.35 on 134 degrees of freedom
## Multiple R-squared:  0.5965, Adjusted R-squared:  0.5514
## F-statistic: 13.21 on 15 and 134 DF,  p-value: < 2.2e-16
```

```r
lm.fit2 = lm(efterspørgsel ~ 1) # simpel model
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = efterspørgsel ~ 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -302.83 -105.83  -16.33  104.17  327.17
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   522.83      11.14   46.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 136.4 on 149 degrees of freedom
```

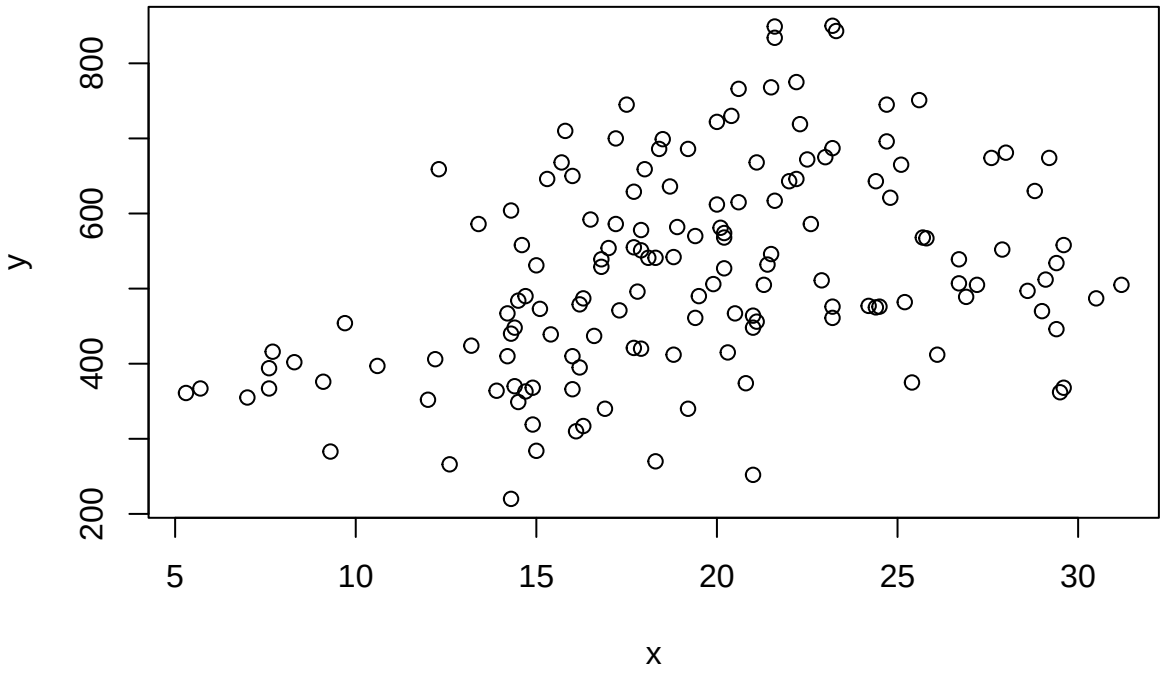
```r
lm.fit3 = lm(efterspørgsel ~ temp_max_past1h^2) # melllem model.
summary(lm.fit3)
```

```
##
## Call:
## lm(formula = efterspørgsel ~ temp_max_past1h^2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -285.50  -89.75  -13.14   82.08  306.14
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      350.064     37.352   9.372  < 2e-16 ***
## temp_max_past1h    8.926      1.854   4.815 3.59e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 127.2 on 148 degrees of freedom
## Multiple R-squared:  0.1355, Adjusted R-squared:  0.1296
## F-statistic: 23.19 on 1 and 148 DF,  p-value: 3.594e-06
```

```r
lm.fit4 = lm(efterspørgsel ~ temp_max_past1h + temp_max_past1h^5) # ekstrem model
summary(lm.fit4)
```

```
##
## Call:
## lm(formula = efterspørgsel ~ temp_max_past1h + temp_max_past1h^5)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -285.50  -89.75  -13.14   82.08  306.14
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      350.064     37.352   9.372  < 2e-16 ***
## temp_max_past1h    8.926      1.854   4.815 3.59e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 127.2 on 148 degrees of freedom
## Multiple R-squared:  0.1355, Adjusted R-squared:  0.1296
## F-statistic: 23.19 on 1 and 148 DF,  p-value: 3.594e-06
```

```r
x <- data3$temp_max_past1h
y <- data3$efterspørgsel
plot(x, y)
```




```
data <- data.frame(y, x)
data
```

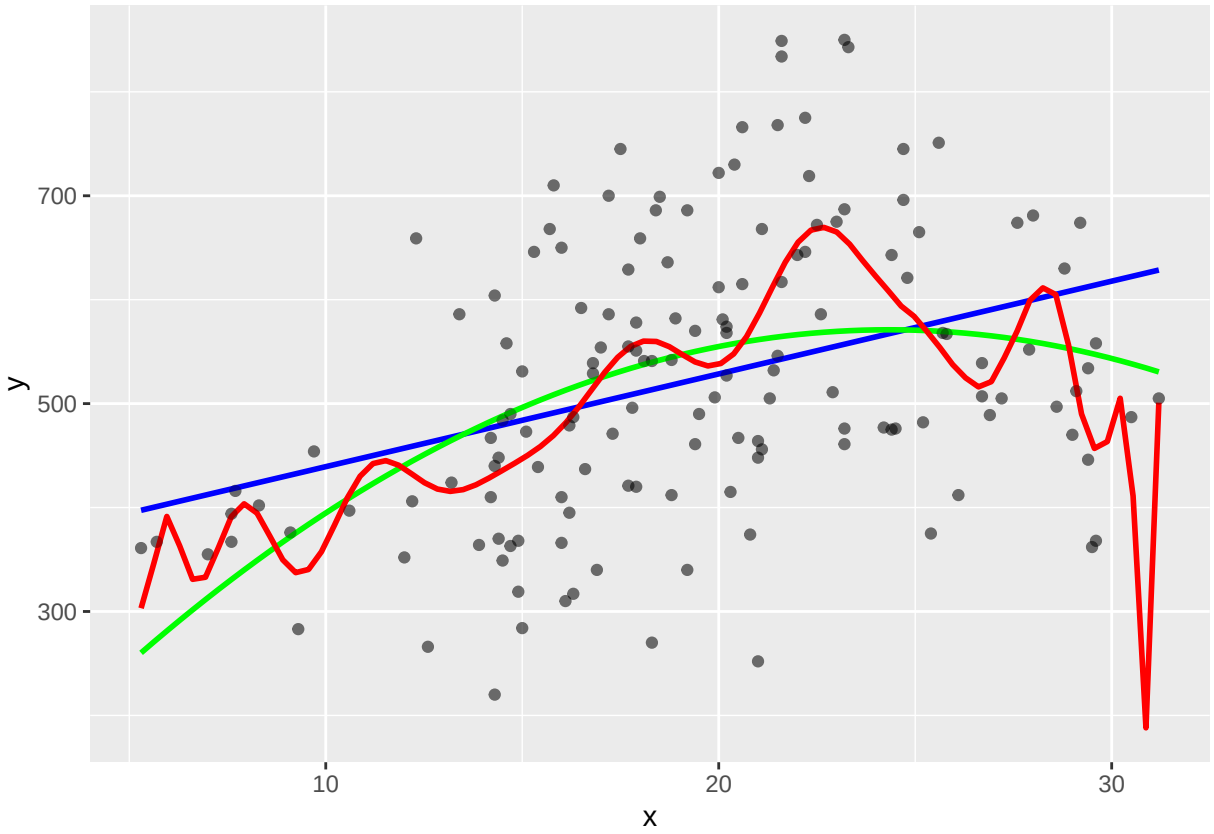
```
set.seed(1)
cv.error <- rep(0, 5)
for (i in 1:5) {
  glm.fit <- glm(y~poly(x , i), data = data)
  cv.error[i] <- cv.glm(data , glm.fit)$delta [1]
}
cv.error
```

```
## [1] 16364.47 15193.48 14296.16 14422.42 14346.71
```

```
ggplot(data3, mapping = aes(x=x, y=y)) +
  geom_point(alpha=1/3) +
  geom_smooth(method="glm", formula = y ~ poly(x, 1, raw=TRUE), se=FALSE, colour="blu
  geom_smooth(method="glm", formula = y ~ poly(x, 2, raw=TRUE), se=FALSE, colour="gre
  geom_smooth(method="glm", formula = y ~ poly(x, 22, raw=TRUE), se=FALSE, colour="re
geom_point(data=data3, mapping = aes(x=x, y=y), alpha=1/3)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
```

```
## prediction from a rank-deficient fit may be misleading
```



Tidy

Transformer

Visualiser

Model

Kommunikér/analyse

Sessioninformation

For at højne reproducerbarheden printes der en udskrift om den nuværende R session:

```
SI <- sessionInfo(package = NULL) # Udskriver en liste om denne R session.
```

Litteratur

Bilag