



DATAANALYSE, ERHVERVSAKADEMIET DANIA

VEJLEDER: Simon Bjerrum Eilersen



GRUPPE 1:

Nicklas Helmundt

Nicklas Wollesen

Nikolaj Aaby Larsen

Nicolaj Andersen

Mathias Ødegaard Andersen

6. JANUAR 2026

CASE: VIBORG FF

1. SEMESTER

ANTAL ANSLAG: xxxx

Indholdsfortegnelse

0.1	Problemstilling	1
0.1.1	Problemformulering	1
0.1.2	Afgrænsning (osv.... [@sec-pf] hvis du har brug for at referere til snittet problemformulering)	1
0.2	Videnskabsteori og metode	1
0.3	Analyse	2
0.4	Væsentligste konklusioner	2
0.5	Literaturliste	3
0.6	Bilag	4
.1	Dataforståelse (husk tankegang fra CRISP-DM)	4
.2	Dataklargørelse	4
.2.1	Data fra Superstats	4
.2.2	Data fra DMI	18
.2.3	Data fra nager.at (helligdage og feriedage)	21
.2.4	Transfer data	24
.2.5	Bjarnes data	26
.2.6	RSQLight	26
.2.7	Download joinet datasæt	30
.2.8	Preprocess	30
.3	Modellering	30
.3.1	Stor model - linæær regression med alle mulige variabler	31
.3.2	Stor model for 3 dage før - linæær regression med alle mulige variabler	37
.3.3	Stor model for 7 dage før - linæær regression med alle mulige variabler	40
.3.4	Stor model for 10 dage før - linæær regression med alle mulige variabler	40
.3.5	Stor model for flere måneder før - linæær regression med alle mulige variabler	41
.3.6	Samlet for lineære modeller	41
.3.7	Lasso / Ridge regression (shrinkage-metoder)	44
.3.8	Best subset selection	47
.3.9	Sammenligning af nøgletal	50
.3.10	Cross Validation	51
.3.11	Prædiktion	54
.3.12	Evaluering af vores model på en allerede spillet kamp (fra datasættet)	59
.4	Eksempler på markdown-sproget	61
.5	Transskriberede interviews og præsentationer	62
.5.1	Daniel	62
.5.2	Praktikanter	67
.5.3	Palle	70
.5.4	Palle & Olga	77
.5.5	Daniel & praktikanterne	81

Figuroversigt

1	Faserne i CRISP-DM.	61
2	Australisk BNP pr capita	62

Tabeloversigt

1	Outliers og indflydelsesrige observationer.	62
---	---	----

0.1 Problemstilling

Det helt centrale i udarbejdelsen af dette projekt kredser sig om at kunne forudsige antallet af tilskuere til Viborg FF's hjemmekampe i Superligaen. På forskellige tidspunkter før kampstart. Det gælder flere måneder før, 10 dage før, 7 dage før og sluteligt 3 dage før start fløjtet lyder fra dommeren på Energi Viborg Arena. Evnen til at kunne forudsige fremmødet før kampstart er central for planlægningen af kampafviklingen, som Palle Nielsen, billet- og stadionansvarlig hos VFF beskriver er der "altså markant forskel på, om man skal servicere 8.500, eller skal servicere 6.500", og "at jo tidligere, at man ved noget om det, så er der jo hele klargøringen til boderne og sådan nogle ting" [5.3](#). Palle beskriver hvordan en god kampafvikling forudsætter, at der ikke opstår problemer i forhold til normeringen af ansatte, herunder fremhæver han sikkerheds- og bodpersonale, som skal matche antallet af tilskuere på stadion før der er tale om en god kampafvikling [5.3](#). Palle fortæller helt generelt hvordan klubbens brug af data er vokset markant. Han fortæller blandt meget andet om brugen af data, at de skal have kigget ind i normeringen af bodpersonalet i halvlegene, da de gennemsnitligt kun har otte transaktioner i halvlegen [5.4](#). Palle fortæller også, at han har lavet prædiktioner på tilskuertallet på den tilnærmedesvis samme måde gennem 20 år [5.3](#). Han nævner også, at han primært arbejder i Excel, men at "det Olga (Dataanalytiker) arbejder i er jo 100 gange smartere. Så det er jo helt klart den vej, vi skal" [5.4](#). Brugen af data til at forudsige tilskuertallet til fremtidige hjemmekampe er altså vejen frem for Viborg FF.

0.1.1 Problemformulering

Hvordan kan ML-modeller forudsige tilskuerantallet på forskellige tidspunkter før Viborg FF's hjemmekampe, og hvordan kan løsningen implementeres i praksis?

Undersørgsmål

Hvilke faktorer påvirker antallet af tilskuere?

Hvordan undersøges hvilken model der vil klare sig bedst?

Hvordan kan løsningen implementeres i praksis, med deres nuværende datamodenhed?

0.1.2 Afgrænsning (osv.... [@sec-pf] hvis du har brug for at referere til afsnittet problemformulering)

0.2 Videnskabsteori og metode

Opgaven tager udgangspunkt i et kritisk realistisk paradigme, da tilskuerantallet påvirkes af underliggende mekanismer, og derfor skal opfattes som et observerbart fænomen. Kritisk realisme bygger fundamentalt på, at virkeligheden eksisterer uafhængigt af observatøren, med at vores erkendelse om, at viden om virkeligheden opnås gennem empiriske observationer, data og analytiske modeller. Tilskuerantallet estimeres på baggrund af valgte kvantitative variabler, som vejret, modstandere og helligdage, der fungerer som indikatorer for disse mekanismer. I forhold til ontologi, så skal tilskuerantallet ikke forstås som tilfældigt, da antallet bliver påvirket af vores tidligere nævnte variabler. Vi må dog antage, at der altid vil være underliggende mekanismer, som vi ikke kender, som alligevel kan have betydning for antallet af tilskuere. I forhold til epistemologi har vi opnået viden empirisk ved at analysere historiske data om tilskuertal og kamprelaterede faktorer ved hjælp af machine learning-modeller, som identificerer statistiske mønstre og sammenhænge. Metodologisk anvender vi en kvantitativ tilgang baseret på sekundære data. De anvendte machine learning-modeller er bygget op på et positivistisk videnskabssyn, ved at de har til formål at identificere statistiske sammenhænge mellem inputvariable og tilskuerantal. Inden for en kritisk realistisk ramme forstås modellerne imidlertid primært som redskaber til at indikere, frem for at forklare de underliggende mekanismer.

Undersøgelsesdesignet er i opgaven overvejende deduktiv, da eksisterende machine learning-teori anvendes til at teste forskellige modellers præstation. Samtidig indgår der også abduktive elementer, hvor resultaterne anvendes til justering af variabelvalg, og fortolkning af mulige mekanismer. Vores data består primært af kvantitative data, såsom tilskuerantal og kamprelaterede variable, men inkluderer også kvalitative data fra et interview med Viborg FF, som bidrager til forståelsen af bagvedliggende mekanismer. Der er altså inddraget regressionsbaserede

machine learning-modeller, som er effektive til forudsigelse af en numerisk variabel. Modellernes præstation vurderes ved hjælp af MSE, RMSE og R^2 på opdelte trænings- og testdatasæt. Validiteten styrkes ved test på ukendte data, mens reliabiliteten understøttes af en ensartet og reproducerbar analyseproces. Resultaterne er i væsentlig grad afhængige af datakvaliteten og de tilgængelige variabler, hvilket fremhæver betydningen af systematisk dataforberedelse.

0.3 Analyse

0.4 Væsentligste konklusioner

0.5 Literaturliste

- Bang, Claus Grand. 2024. *Data-Driven Decision-Making for Business*. 1. udg. Routledge.
- DMI. 2022. "DMI API". <https://www.dmi.dk/frie-data>.
- JaseZiv. 2025. "Ubegrænset lyd- og videotransskription". <https://github.com/JaseZiv/worldfootballR>.
- Leif Foged. 2025. "TurboScribe: Ubegrænset lyd- og videotransskription". <https://turboscribe.ai/da/>.
- nager.at. 2025. "Worldwide Public Holidays". <https://date.nager.at/>.
- SuperStats. 2007. "Superligaen i tal, fodbold, statistik, resultater og tabeller". <https://superstats.dk/>.

0.6 Bilag

.1 Dataforståelse (husk tankegang fra CRISP-DM)

.2 Dataklargørelse

.2.1 Data fra Superstats

Upload af relevante pakker.

```
pacman::p_load("tidyverse", "rvest", "stringr", "lubridate")
```

Mappen VFFdata til opbevaring af data. Hvis den ikke eksistere, så bliver den oprettet.

```
if (!dir.exists("VFFdata")) dir.create("VFFdata")
```

Først skal sæsonerne hvor VFF har været i Superligaen webscrapes. Så der ikke bliver hentet unødvenigt meget data ned. Desuden er det reproducerbart og man slipper for at definere en vektor med år manuelt. Læser HTML-siden direkte fra URL'en med read_html og gemmer den i containeren html_superstats. Her hentes dataen fra (SuperStats 2007).

```
html_superstats <- read_html("https://superstats.dk/hold/alltime?id=11&utm", encoding = "UTF-8")
```

Finder alle table-elementer på siden med html_elements.

```
tables_sæsoner <- html_superstats |> html_elements("table")
```

map() anvender her funktionen html_table på hver tabel i tables_sæsoner. Det giver en liste af dataframes, en for hver html-tabel på siden.

```
tables_df <- map(tables_sæsoner, html_table)
```

Det er tabel 3 der er den rigtige tabel, som indeholder de sæsoner VFF er i Superligaen. [[3]] trækker det tredje element ud af listen. Funktionen html_table() fra rvest konverterer html-tabellen til en dataframe.

```
tables_VFF <- (tables_sæsoner[[3]] |> html_table())
```

Fjerner den første række i tabellen den med NA-værdier. [-1,] Tag alle rækker undtagen række 1, da 1 har - foran betyder det fjern.

```
tables_VFF <- tables_VFF[-1, ]
```

Returnerer TRUE hvis tabellen indeholder mindst en NA. any er der nogen værdier? som er NA i tables_VFF? Output her er lig med FALSE, som ønsket.

```
any(is.na(tables_VFF))
```

```
[1] FALSE
```

Udtrækker slutåret for hver sæson fra teksten 1999/2000, 2016/2017 osv. URL'en på Superstats bruger slutåret (f.eks. 2017 for sæsonen 2016/2017), så de sidste fire cifre i sæsonstrenge skal bruges i et loop senere for at finde de rigtige sæsoner Laver dem til integer (heltal). \$ efter tables_VFF bruges til at hente kolonnen sa_son. |> efter betyder og så... gör noget mere. Resultatet: [1] 2026 2025 2024 2023 2022 2017 2016 2014 2008 2007 2006 2005 2004 2003 2002 2001 2000 1999 1997 1996 1994. Så nu er VFF_sæsoner en vektor med start året for alle de år VFF har været i superligaen.

```
VFF_sæsoner <- tables_VFF$Sæson |>
#> Regex-forklaring af "\\d{4}$".
#> \\d betyder cifre mellem 0-9.
#> {4} betyder præcis fire cifre f.eks. 2026.
#> $ slutningen af strengen.
#> $ tvinger regex til at tage de sidste 4 cifre, ikke de første.
#> Uden $ ville str_extract tage startåret (f.eks. 2016 i 2016/2017).
str_extract("\\\\d{4}$") |>
as.integer()
```

Opretter en liste og gemmer i en container, som skal indeholde kampdata for superligakampe i de valgte år/sæsoner.

```
kampliste_samlede <- list()
```

For-loopet looper gennem alle årene i vektoren VFF_sæsoner. VFF_sæsoner indeholder slutåret for hver sæson (f.eks. 2024 = sæson 2023/2024). Her hentes dataen fra (SuperStats 2007).

```
for (y in VFF_sæsoner) {

#> Bygger URL'en til superstads for en bestemt sæson.
#> y svarer til året.
#> paste0 binder URL'en og året sammen uden mellemrum.
#> Resultatet bliver f.eks. "https://superstats.dk/program?aar=2026" hvis y = 2026.
url <- paste0("https://superstats.dk/program?aar=", y)

#> read_html() henter hele hjemmesidens html-kode med det tilføjede år.
#> encoding = "UTF-8" sikre korrekt visning af alle tegn, som f.eks. æ, ø og å.
kampliste_alle <- read_html(url, encoding = "UTF-8") |>

# html_elements("table") finder alle tabeller på siden i html koden.
html_elements("table") |>

#> html_table() konverterer hver tabel til en tibble.
#> convert = FALSE = alle kolonner læses som tekst, så kan datatyper bestemmes senere.
#> header = FALSE betyder at den første række i tabellen ikke skal bruges som kolonnenavn.
#> Da det er Runde 1 f.eks. og ikke hold, resultat osv.
html_table(header = FALSE, convert = FALSE)

#> På hver side med kampprogrammet ligger der flere tabeller, men de sidste 4 tabeller
#> er ikke kampe. Det er f.eks. stilling, optakt osv.
#> 1:(length(kampliste_alle) - 4) laver en sekvens fra 1 og frem til
#> det totale antal tabeller i kampliste_alle minus de sidste 4 tabeller.
kampliste_valgte <- kampliste_alle[1:(length(kampliste_alle) - 4)] 

#> Tilføjer en sæson-kolonne til hver tabel i listen.
#> lapply() = løber alle tabeller igennem en ad gangen.
#> hver_tabel_i_listen = betyder bare den aktuelle tabel.
```

```

#> Sæson skal f.eks. være 2023/2024 når y = 2024 derfor paste0(y-1, "/", y).
#> Paste0 sætter sæsonerne sammen f.eks. "2024/2025".
#> function(hver_tabel_i_listen) { ... } er en funktion,
#> som modtager et element ad gangen (en tabel).
kampliste_valgte <- lapply(kampliste_valgte, function(hver_tabel_i_listen) {
  hver_tabel_i_listen |> mutate(sæson = paste0(y-1, "/", y))
})
# Gemmer kamp-tabellerne i en stor liste, hvor navnet på hvert element er sæsonen.
# F.eks. kampliste_samlede[["2023/2024"]] = alle runder for sæsonen 2023/2024.
kampliste_samlede[[paste0(y-1, "/", y)]] <- kampliste_valgte
}

```

Gemmer data

```

# Gemmer data i en RDS-fil i data mappen.
saveRDS(kampliste_samlede, "VFFdata/superstatsliste.rds")

```

Indlæs data igen fra RDS-filen. Alle andre steps før nu kan udkommenteres.

```
gemt_kampliste <- readRDS("VFFdata/superstatsliste.rds")
```

Konverterer til dataframe. Kombinerer alle lister til en samlet dataframe. bind_rows() binder alle tabellerne sammen under hinanden og laver et stort datasæt i en tibble.

```

kampe_df <- gemt_kampliste |>
  bind_rows()

```

Rensning af data.

```

# Først skal rækkerne der kun indeholder runde nummer transformeres til en kolonne for sig selv.
kampe_df <- kampe_df |>
  mutate(
    #> Opretter en hjælpekollonne find_runde med mutate, som kun indeholder tekst fra de rækker,
    #> hvor X1 starter med "Runde" (f.eks. "Runde 1", "Runde 2", osv.).
    #> ^ = betyder "i starten af teksten".
    #> str_detect(X1, "^Runde"): returnerer TRUE for rækker,
    #> hvor X1 starter med ordet "Runde" (f.eks. "Runde 1", "Runde 12").
    #> if_else(TRUE) = gemmer i containeren find_runde, ellers NA = FALSE gemmer ikke.
    find_runde = if_else(str_detect(X1, "^Runde"), X1, NA)) |>

    #> fill(find_runde) kopierer den seneste ikke-NA værdi nedad.
    #> Resultat: alle kampe under "Runde 1" får find_runde = runde 1,
    #> alle kampe under "Runde 2" får find_runde = runde 2, osv. Hele vejen igennem.
    fill(find_runde) |>

    #> Opretter selve runde-kolonnen der skal være i dataframen igen med mutate.
    #>
    #> str_extract henter tallet (f.eks. "1") fra "Runde 1".
    #> \d+ finder det første tal (en eller flere cifre) i teksten, f.eks. "1" i "Runde 1".
    #> as.integer() gør det numerisk, altså konverterer tallet fra tekst til heltal (integer).
    mutate(runde = str_extract(find_runde, "\d+")) |> as.integer() |>

    #> Fjerner rækker der selv er runde-overskrifter:

```

```

#> str_detect(X1, "^Runde") finder rækker hvor kolonnen X1 starter med ordet "Runde"
#> man kunne også have valgt X2.
#> ! foran betyder ikke, så filter(!str_detect) bevarer alle rækker,
#> undtagen rækkerne med "Runde 1", "Runde 2", osv.
#> 1 Runde 1 Runde 1      ""      ""      ""      ""      ""      ""      2025/2026
#> 2 Fre     18/07 19:00 "VFF-FCK" "2-3" "6.696" "Sandi Putros"      ""      2025/2026
#> Så her fjernes række 1, osv.
filter(!str_detect(X1, "^Runde")) |>

#> Fjerner hjælpeakolonnen find_runde, så den ikke optræder i dataframen.
#> Kolonnen blev kun brugt som mellemtrin til at finde og udfylde rundenummeret,
#> select(...) svarer til at fravælge.
select(-find_runde)

```

Her renser vi dataene, giver kolonnerne passende navne, fjerner kolonne 7 (TV-logoer i billedformat på hjemmesiden), Vi fjerner overflødige rækker, filtrerer kun de kampe hvor der findes et resultat, og rydder op i tekstformater såsom tilskuertal og ugedage.

```

kampe_df <- kampe_df |>
  # # Fjerner kolonnen X7, kolonnen indeholdte TV-logoer i billedformat på hjemmesiden.
  #> select(...) svarer til at fravælge.
  select(-X7) |>
    #> Beholder kun rækker hvor kolonnen X5 indeholder et tal (tilskuertallet)
    #> viser at kampen er spillet.
    #> Regex-forklaring: ^ = start af teksten.
    #> [0-9\\\.]+ = en eller flere cifre og eller punktum.
    #> $ = slut af teksten.
    #> Så værdien må kun bestå af tal og punktummer.
    #> str_detect(X5, "^[0-9\\\.]+$"): returnerer TRUE for rækker:
    #> Med f.eks. 12.000, 100, 0 osv. men returner FALSE ved f.eks. NA, "", "--" osv.
    #> Filtrerer kampene så kun færdigspillede kampe (med tilskuertal) bevares.
    filter(str_detect(X5, "^[0-9\\\.]+$")) |>
    #> Giver kolonnerne nye navne.
    #> rename() syntaks: new name = old name f.eks. ugedag = X1.
    #> Så hedder kolonnen nu ugedag osv.
    rename(
      ugedag = X1,
      dato_og_tidspunkt = X2,
      kamp = X3,
      resultat = X4,
      tilskuere = X5,
      dommer = X6) |>
  mutate(
    #> Fuldender ugedagsnavnene:
    #> paste0() sætter teksten sammen uden mellemrum.
    #> Her tilføjes ordet "dag" direkte efter den eksisterende ugedag.
    #> Lør bliver til Lørdag osv.
    ugedag = paste0(ugedag, "dag"),
    tilskuere = tilskuere |>
      #> Fjerner punktummet i tal f.eks. 1.000, så det bliver 1000.
      #> "\\." "et punktum" og "" "ingenting" = byt punktum ud med ingenting = fjern punktummet.
      #> str_replace_all(fjerner alle ., erstatter alle med "ingenting").
      str_replace_all("\\.", "") |>
      # Konverterer tekststrengen f.eks. 1000 til et rigtigt heltal (integer).
      as.integer())

```

Nye variabler og fortsat rensning af data. Her opdeles variablen dato_og_tidspunkt, som består af både dato og klokkeslæt, i to separate kolonner: dato og klokkeslæt.

```
kampe_df <- kampe_df |>
  # separate() splitter teksten ved mellemrum (sep = " ").
  separate(dato_og_tidspunkt, into = c("dato", "klokkeslæt"), sep = " ") |>
    #> Efter split konverteres klokkeslæt fra en tekst streng til tidsformatet hms.
    #> med hm(), så tiden behandles korrekt i beregninger og ved dannelse af dttm senere.
    mutate(klokkeslæt = hm(klokkeslæt))
```

Trækker månederne ud - med unique, så f.eks. 05 kun optræder en gang i outputted osv. substr(dato, 4, 5) henter tegn 4 og 5 i dato-strenge, dvs. måneden. Superligaen starter typisk i juli/07 og slutter typisk i maj/05 / start juni/06

```
unique(substr(kampe_df$dato, 4, 5))
```

```
[1] "07" "08" "09" "10" "11" "12" "02" "03" "04" "05" "06"
```

Tilføj år til dato og ændrer formatet fra til . Dato-kolonnen indeholder kun dag og måned (dd/mm), derfor skal det bestemmes, hvilket år kampene hører til. Det afhænger af Superligaens struktur, som blev kigget på lige før.

```
kampe_df <- kampe_df |>
  mutate(dato = paste0(
    # Sætter året ind efter dato'en og / med paste0.
    dato, "/",
    if_else(
      #> Hent månedstallene fra dato'en.
      #> substr(dato, 4, 5) tager tegn 4 og 5 i dd/mm dvs. mm.
      #> as.integer(...) konverterer "07" til 7, "03" til 3 osv.
      #> Hvis måneden er 7 eller større (juli-december),
      #> så betyder det, at kampen hører til sæsonens startår.
      as.integer(substr(dato, 4, 5)) >= 7,
      #> Hvis ovenstående betingelse er TRUE:
      #> så bruges sæsonens startår (før "/")
      #> f.eks. "2023/2024" så det "2023"
      as.integer(substr(sæson, 1, 4)),
      #> Hvis betingelsen er FALSE (måned < 7 = januar-juni):
      #> så bruges sæsonens slutår (efter "/")
      #> f.eks. "2023/2024" så det "2024"
      as.integer(substr(sæson, 6, 9)))) |>
      #> Efter årstallet er tilføjet bruges dmy() til at lave dato'en om til et
      #> rigtigt datoobjekt med formatet <date>.
      dmy())
```

Opretter en samlet datetime-variabel (dttm) gör det nemmere når datasættet skal merges med andre datasæt senere. dato er et objekt og klokkeslæt er et objekt. Når de lægges sammen, så bliver det et objekt i stedet.

```
kampe_df <- kampe_df |>
  mutate(
    #> Opretter et samlet tidsstempel (date + time i dttm-format)
    #> as_datetime konverterer til dato + klokkeslæt
    tidsstempel = as_datetime(dato + klokkeslæt)) |>
    # Fjerner de gamle kolonner med select(-x, -x1), da tidsstempel nu indeholder begge dele
    select(-dato, -klokkeslæt)
```

Trækker timerne ud med unique, så f.eks. 15 kun optræder en gang i outputted osv.

```
unique(hour(kampe_df$tidsstempel))
```

```
[1] 19 14 16 18 20 15 12 17 21 13 11
```

Laver en ny kategorivariabel for tidsperiode i forhold til kampens starttidspunkt på dagen.

```
kampe_df <- kampe_df |>  
  mutate(  
    #> hour(tidsstempel) henter klokkeslættets time-del som et heltal (0-23).  
    #> if-else før kl. 14 -> formiddag, Kl. 14-17 -> eftermiddag og efter kl. 17 -> aften.  
    #> paste0() sætter ugedag + tidskategori(_for-, eftermiddag og aften) sammen,  
    #> f.eks. Søndag + _aften = Søndag_aften osv.  
    tidsperiode = if_else(  
      hour(tidsstempel) < 14,  
      paste0(ugedag, "_formiddag"),  
      if_else(  
        hour(tidsstempel) <= 17,  
        paste0(ugedag, "_eftermiddag"),  
        paste0(ugedag, "_aften"))))
```

Her deles resultat-kolonnen op i to nye kolonner: separate() splitter tekst baseret på et separatortegn, her er det “-”, som er mellem mål for hjemmeholdet og udeholdet. c(“mål_hjemme”, “mål_ude”) er en charactervektor med navnet på de to nye kolonner, som resultatet skal splittes ud i. convert = TRUE betyder, at R automatisk konverterer tekst til tal, altså i stedet for .

```
kampe_df <- kampe_df |>  
  separate(resultat, into = c("mål_hjemme", "mål_ude"), sep = "-", convert = TRUE)
```

Laver to nye kolonner med point til hjemme- og udeholdet. point_hjemme med if_else 3 point hvis hjemmeholdets mål > udeholdets mål. 1 point hvis mål_hjemme == mål_ude, ellers 0 point. point_ude fungerer omvendt for udeholdet.

```
kampe_df <- kampe_df |>  
  mutate(  
    point_hjemmehold = ifelse(mål_hjemme > mål_ude, 3, ifelse(mål_hjemme == mål_ude, 1, 0)),  
    point_udehold = ifelse(mål_hjemme < mål_ude, 3, ifelse(mål_hjemme == mål_ude, 1, 0)))
```

separate() deler kampnavnet i hjemme- og udehold, men på superstads hjemmeside er der mellem rum efter “-” ved hold med kun to bogstaver. str_trim() fjerner mellemrum i starten og slutningen af teksten, så ”OB” bliver til ”OB” og ”VB” bliver til ”VB”.

```
kampe_df <- kampe_df |>  
  separate(kamp, into = c("hjemmehold", "udehold"), sep = "-") |>  
  mutate(hjemmehold = str_trim(hjemmehold), udehold = str_trim(udehold))
```

Sum af mål i de sidste 3 hjemmekampe lag. Mål i sidste tre hjemmekampe. Skal vise målene scoret i de sidste tre hjemmekampe op til den nye hjemmekamp.

```

kampe_df <- kampe_df |>
  #> arrange(sæson, tidsstempel): Sorterer alle kampe kronologisk inden for hver sæson.
  #> Det er vigtigt, fordi lag() ellers henter fra en forkert kamp.
  arrange(desc(sæson), tidsstempel) |>
  #> group_by(sæson, hjemmehold): beregningen laves separat for:
  #> Hver sæson så sæsonerne ikke overlapper hinanden.
  #> hvert hjemmehold så hvert hold, så FCK's og VFF's hjemmekampes mål ikke lægges sammen.
  group_by(sæson, hjemmehold) |>
  mutate(
    mål_seneste_3_hjemme =
      # lag(mål_hjemme, 1) Mål i den forrige hjemmekamp.
      lag(mål_hjemme, 1, default = 0) +
      # lag(mål_hjemme, 2) Mål i de to forrige hjemmekampe.
      lag(mål_hjemme, 2, default = 0) +
      # lag(mål_hjemme, 3) Mål i de tre forrige kampe.
      lag(mål_hjemme, 3, default = 0)
    #> Det der sker:
    #> Runde 1: vil altid vise 0 (ingen tidligere kampe).
    #> Runde 2: mål i kamp 1
    #> Runde 3: mål i kamp 1 + kamp 2
    #> Fra runde 4 i hver sæson vises summen af de tre seneste hjemmekampe.
  ) |>
  #> ungroup() fjerner den grouping, der blevet lavet med group_by().
  #> Så det igen bliver et almindeligt datasæt uden grupper.
  ungroup()

```

desc = descending, så datasættet vises i faldende rækkefølge igen.

```

kampe_df <- kampe_df |>
  arrange(desc(sæson), tidsstempel)

```

Antallet af tilskuere ved seneste møde. Antallet af tilskuere sidst hjemmeholdet mødte samme modstander Indtil videre bliver default sat til 0.

```

kampe_df <- kampe_df |>
  #> Sorterer alle kampe kronologisk på tværs af alle sæsoner.
  #> Dette er vigtigt, fordi lag()-funktionen skal vide hvilken kamp der kronologisk var den seneste.
  arrange(tidsstempel) |>
  #> Grupperer data efter hjemmehold og udehold.
  #> Det betyder, at lag() kan finde den seneste hjemmekamp mod præcis samme modstander.
  group_by(hjemmehold, udehold) |>
  #> lag(tilskuere) Henter tilskuertallet fra den foregående kamp mod samme modstander.
  #> default = 0 Bruges, intil en bedre løsning findes, når der ikke er en tidligere hjemmekamp mod modstander.
  #> Sker ved det historiske første møde med en modstander.
  mutate(
    tilskuere_hold_lag =
      lag(tilskuere, default = 0)) |>
  # Fjerner grupering.
  ungroup()

```

desc = descending, så datasættet vises i faldende rækkefølge igen.

```
kampe_df <- kampe_df |>
  arrange(desc(sæson), tidsstempel)
```

Point i sidste 3 hjemmekampe. Viser summen af point i de tre seneste hjemmekampe før den aktuelle kamp.

```
kampe_df <- kampe_df |>
  #> Sorterer kampe kronologisk inden for hver sæson.
  #> Gør at lag() henter point fra de rigtige hjemmekampe.
  arrange(desc(sæson), tidsstempel) |>
  #> Gruppen laves pr. sæson og hjemmehold.
  #> Hver sæson behandles for sig og hvert hjemmehold får sin egen hjemmebane historik.
  group_by(sæson, hjemmehold) |>
  mutate(
    form_seneste_3_hjemme =
      # Point i seneste hjemmekamp.
      lag(point_hjemmehold, 1, default = 0) +
      # Point i de to seneste.
      lag(point_hjemmehold, 2, default = 0) +
      # Point i de tre seneste.
      lag(point_hjemmehold, 3, default = 0)
    #> Det der sker:
    #> Runde 1: 0 (ingen tidligere hjemmekampe).
    #> Runde 2: point fra runde 1.
    #> Runde 3: point fra runde 1 + runde 2.
    #> Runde 4+: point fra tre seneste hjemmekampe.
  ) |>
  #> Fjerner grouping og går tilbage til normalt datasæt.
  ungroup()
```

Placering lag. Fjerner sæsonen 1993/1994 fra datasættet. I denne sæson spiller VFF kun 18 kampe.

```
kampe_df <- kampe_df |>
  # Beholder kun de kampe der IKKE!= er 1993/1994.
  filter(sæson != "1993/1994")
```

Her ændres datasættet til en række pr. kamp pr. hold med pivot_longer() = dobbelt så mange rækker. Det giver 2 rækker pr. kamp (en for hjemmeholdet og en for udeholdet) Det gør det muligt at beregne point, målscore osv. pr. hold.

```
kampe_kloning <- kampe_df |>
  pivot_longer(cols = c(hjemmehold, udehold),           # De to kolonner laves om til rækker.
                names_to = "hjemme_udé",                  # Her gemmes navnet på kolonerne hjemmehold og udeholde.
                values_to = "hold") |>                   # Denne nye kolonne indeholder selve holdnavnet f.eks.
  #> Opretter en ny kolonne = point.
  #> if_else tjekker om denne række er hjemmeholdets række?
  #> Hvis ja, så bruges point_hjemmehold. Hvis nej, så bruges point_udéhold.
  mutate(point = if_else(hjemme_udé == "hjemmehold", point_hjemmehold, point_udéhold),
        #> mål_for = mål scorert af dette hold.
        #> Hvis denne række er hjemmeholdets række, så bruges mål_hjemme.
        #> Ellers udeholdet, så bruges mål_udé.
        mål_for = if_else(hjemme_udé == "hjemmehold", mål_hjemme, mål_udé),
        #> mål_imod = antal mål modstanderen scorede mod holdet i kampen.
        #> Hvis det er hjemmeholdets række, så bruges mål_imod = mål_udé
        #> (modstanderen = udeholdet).
```

```

#> Hvis dette er udeholdets række, så bruges mål imod = mål_hjemme
#> (modstanderen = hjemmeholdet).
mål_imod = if_else(hjemme_ude == "hjemmehold", mål_ude, mål_hjemme)

```

Her beregnes stillingen løbende for hver runde.

```

stillinger <- kampe_kloning |>
#> Starter med en række pr. hold pr. kamp og der grupperes efter sæson og hold.
#> Så stillingen bliver indefor de separate sæsoner og hold pr. kamp.
group_by(sæson, hold) |>
#> Kampene kommer i kronologisk rækkefølge for hvert hold.
#> arrange efter tidspunktet for kampene, indefor grupperne.
#> Så sorteres kampene kronologisk for hvert hold i hver sæson med .by_group = TRUE.
arrange(tidststempel, .by_group = TRUE) |>
#> Beregn løbende totaler cumulative summer, altså løbende sum.
#> Løbende point, løbende mål for, løbende mål i mod og løbende mål difference for holdende.
#> Efterfølgende fjernes grouping for at omgruppere på sæson, runde og hold.
mutate(
  point_sum      = cumsum(point),
  mål_for_sum   = cumsum(mål_for),
  mål_imod_sum = cumsum(mål_imod),
  måldiff_sum   = mål_for_sum - mål_imod_sum) |>
ungroup() |>
#> Laver en række pr. sæson, pr. runde, pr. hold.
#> For at få point sum osv. for en bestemt runde for holdene i en given sæson.
#> last() henter de nyeste summer for holdet i den pågældende runde.
group_by(sæson, runde, hold) |>
# summarise() laver en række for denne kombination af point, mål_for osv.
summarise(
  point_sum      = last(point_sum),
  mål_for_sum   = last(mål_for_sum),
  mål_imod_sum = last(mål_imod_sum),
  måldiff_sum   = last(måldiff_sum),
  # .groups = "drop" gør, at resultatet ikke fortsætter med at være grupperet.
  .groups = "drop") |>

#> For hver runde rangeres holdene korrekt.
#> 1) flest point
#> 2) bedst måldifference
#> 3) flest scorede mål
#> 4) alfabetisk (som sidste udvej. Kan forbedres)
#> I faldende rækkefølge med desc, så det er dem med flest point,
#> mål difference osv. som kommer først.
group_by(sæson, runde) |>
arrange(
  desc(point_sum),
  desc(måldiff_sum),
  desc(mål_for_sum),
  hold,
  # Sorterer inde i gruppen .by_group = TRUE.
  .by_group = TRUE) |>

# Tildeler holdenes normale placering, altså den rigtige stilling.
mutate(plac_norm = row_number()) |> # giver korrekt placering i stillingen pr. runde.
ungroup() # fjerner grupperingen bagefter.

```

Strukturen i superligaen ændrer sig igennem årene. For VFF er sæsonen 2016/2017 helt unik, mens sæsonerne de er i Superligaen. Før 2016/2017 er anderlæs og sæsonerne efter også er en kategori for sig. I sæsonen 2016/2017 er der et split i tabellen og det samme i sæsonerne efter. Først skal split runderne identificeres for hver sæson.

```
splitinfo <- stillinger |>
  # Gruppere pr. sæson hvor splittet skal findes.
  group_by(sæson) |>
  summarise(
    max_runde = max(runde),
    split_runde = if_else(max_runde == 32, 22,                      # Split ved runde 22 i 12-holdssæsoner med
                       # mesterskabsspil og nedrykningsspil.
                       if_else(max_runde == 36, 26,                      # split ved runde 26 i sæsonen 2016/17,
                           # sæson med 14 hold og mesterskabs.. osv.
                           NA))) |>                                     # Ellers ingen split.
  ungroup()                                              # Fjerner gruppering efter summarise
```

Tilføjer split-information til stillingerne for hver sæson. left_join sikrer at alle rækker i stillinger bevares, og at max_runde samt split_runde bliver tilføjet ud fra sæsonen.

```
stillinger_split <- stillinger |>
  left_join(splitinfo, by = "sæson")
```

Top og bund skal låses fast ved split i de sæsoner med opdeling efter x-antal runder. Sådan at et hold i f.eks. bund 6 eller bund 8 ikke kan komme i top 6 og omvendt.

```
stillinger_split <- stillinger_split |>

  #> Grupper efter sæson og hold, så hvert holds runder behandles separat.
  #> Altså et hold ad gangen. Inden for en sæson.
  group_by(sæson, hold) |>

  #> Sørger for at runderne står i korrekt rækkefølge INDEN FOR hver gruppe.
  #> .by_group = TRUE betyder: arrange fungerer separat for hver runde.
  #> Det vil altså sige at kampene sorteres kronologisk for hvert hold hver for sig.
  arrange(runde, .by_group = TRUE) |>
  mutate(

    #> Ved splitrunden: Her indeles i 2 grupper kategorier, top og bund.
    #> Top hvis plac_norm >= 6 bund" hvis plac_norm > 6 og NA i alle andre runder.
    #> Split runde er NA i sæsoner uden split.
    gruppe = if_else(!is.na(split_runde) & runde == split_runde & plac_norm <= 6, "top",
                    if_else(!is.na(split_runde) & runde == split_runde & plac_norm > 6, "bund", NA))

    #> Hvis et hold f.eks. bliver top ved splitrunden, så bliver det top i alle de efterfølgende runder.
    #> Dette er efter strukturen i de sæsoner med split.
    #> fill() går nedad i rækkefølgen af runder (I stigende orden nedad).
    tidyverse::fill(gruppe, .direction = "down") |>
    # Fjern grouping - resten af pipeline kræver et ikke-grupperet datasæt.
    ungroup()
```

Efter splitrunden er top og bund grupperne låst. Nu skal holdene sorteres inden for deres gruppe i hver runde.

```
stillinger_split <- stillinger_split |>
```

```

#> Grupper efter (sæson, runde, gruppe).
#> Så rangeres top- og bund-grupperne separat i hver runde.
group_by(sæson, runde, gruppe) |>

#> Sorter hver gruppe med arrange:
#> 1) flest point.
#> 2) bedst måldifference.
#> 3) flest scorede mål.
#> 4) alfabetisk (som sidste udvej. Kan forbedres).
#> I faldende rækkefølge med desc, så det er dem med flest point, mål difference osv.
#> som kommer først.
arrange(desc(point_sum),
         desc(måldiff_sum),
         desc(mål_for_sum),
         hold,
         .by_group = TRUE) |>
#> Intern placering i gruppen:
#> top-gruppe = 1,2,3,4,5,6.
#> bund-gruppe = 1,2,3,4,5,6 og ...8 i 2016/2017.
#> Hvis gruppe = NA (sæsoner uden split) gives NA.
mutate(intern = if_else(!is.na(gruppe), row_number(), NA),

       #> Global placering:
       #> I sæsoner uden split bruges plac_norm.
       #> Ellers top-gruppe med intern placering (1-6).
       #> Ellers bund-gruppe → intern + 6 (7-12 og 7-14 i 2016/2017).
       global_plac = if_else(is.na(gruppe), plac_norm,
                             if_else(gruppe == "top", intern, intern + 6))) |>
# Fjern grupperingen igen.
ungroup()

```

Henter bund-placeringer i runde 32 for 2016/2017. Da de efter 32 runde spiller play-off, som ikke giver point.

```

bund32 <- stillinger_split |>

#> Her udtrækkes derfor bund holdenes globale placering i runde 32.
#> Den gemmes som frys_plac, så den senere kan bruges som fast låst placering.
filter(sæson=="2016/2017", runde==32, gruppe=="bund") |>
select(sæson, hold, frys_plac = global_plac)

```

Fryser bundholdenes placering efter runde 32 i sæsonen 2016/2017. I denne sæson spiller kun top-6 videre efter runde 32 om placeringer. Joiner først deres placering fra runde 32 (frys_plac), og bruger den til at overskrive global_plac i runder efter 32.

```

stillinger_split <- stillinger_split |>
left_join(bund32, by = c("sæson", "hold")) |>
mutate(

#> Derefter bruges if_else til at overskrive global_plac KUN for:
#> Sæsonen 2016/2017. Gruppe == "bund" (altså bund-8) efter 32 runder.
#> Alle andre sæsoner, top-hold, eller runder 1-32 forbliver uændrede.
global_plac = if_else(
  sæson=="2016/2017" & gruppe=="bund" & runde>32,
  frys_plac, global_plac)) |>

```

```
#> Til sidst fjernes frys_plac igen, da den har udtjent sin værnepligt.
select(-frys_plac)
```

Beregner placeringen før kampen (lag).

```
stilling_lag <- stillinger_split |>

# Grupper pr. sæson og hold, så hver klub behandles separat.
group_by(sæson, hold) |>

# Sorterer runder kronologisk inden for hvert hold.
arrange(runde, .by_group = TRUE) |>

#> lag(global_plac) henter placeringen fra runden før.
#> default = 0L betyder: ingen placering før runde 1.
mutate(placerings_lag = lag(global_plac, default = 0)) |>
ungroup()
```

Udtrækker lagget placering for hjemme- og udehold separat, så de kan joines tilbage på det originale kampdatasæt i to separate kolonner.

```
plac_hjemme <- stilling_lag |>
  select(sæson, runde, hjemmehold = hold,
         placering_lag_hjemme = placering_lag)
```

```
plac_ude <- stilling_lag |>
  select(sæson, runde, udehold = hold,
         placering_lag_ude = placering_lag)
```

Tilføjer placering-lag til kampdata. Joiner på (sæson, runde, hjemmehold) og (sæson, runde, udehold).

```
kampe_df <- kampe_df |>
  left_join(plac_hjemme, by = c("sæson", "runde", "hjemmehold")) |>
  left_join(plac_ude, by = c("sæson", "runde", "udehold"))
```

Form i sidste 3 kampe ude og hjemme.

```
form_total <- kampe_kloning |>
  # Sorterer alle kampe kronologisk inden for hver sæson.
  arrange(sæson, tidsstempel) |>
  # Grupperer efter sæson og hold, så hvert hold behandles for sig.
  group_by(sæson, hold) |>
  mutate(
    # Summen af point i de seneste 3 kampe (uanset hjemme/ude).
    # lag(point, 1) = kamp før den aktuelle.
    # lag(point, 2) = to kampe før.
    # lag(point, 3) = tre kampe før.
    form_seneste_3_total =
      lag(point, 1, default = 0) +
      lag(point, 2, default = 0) +
      lag(point, 3, default = 0)) |>
  ungroup()
```

```

form_hjemme <- form_total |>
  select(sæson, runde, hjemmehold = hold,
         form_seneste_3_hjemmehold = form_seneste_3_total)

form_ude <- form_total |>
  select(sæson, runde, udehold = hold,
         form_seneste_3_udehold = form_seneste_3_total)

kampe_df <- kampe_df |>
  left_join(form_hjemme, by = c("sæson", "runde", "hjemmehold")) |>
  left_join(form_ude,     by = c("sæson", "runde", "udehold"))

```

Mål i sæsonen hjemme + ude lagget før kampen. Beregn løbende sæsonmål pr. hold uanset hjemme/ude.

```

mål_sæson_lag <- kampe_kloning |>
  # Kampene sorteres kronologisk inden for hver sæson.
  arrange(sæson, tidsstempel) |>

  # Et samlet forløb pr. sæson og hold.
  group_by(sæson, hold) |>

  mutate(
    # Samlede mål i sæsonen EFTER kampen.
    mål_sæson_total = cumsum(mål_for),

    # Lagget værdi = mål i sæsonen inden kampen.
    mål_sæson_lag = lag(mål_sæson_total, default = 0)) |>
  ungroup()

```

Klargør data til join med hjemmehold.

```

mål_sæson_hjemme <- mål_sæson_lag |>
  select(
    sæson,
    runde,
    hjemmehold = hold,
    mål_sæson_lag_hjemmehold = mål_sæson_lag)

```

Klargør data til join med udehold

```

mål_sæson_ude <- mål_sæson_lag |>
  select(
    sæson,
    runde,
    udehold = hold,
    mål_sæson_lag_udehold = mål_sæson_lag)

```

Join mål lag tilbage på kampdatasættet.

```

kampe_df <- kampe_df |>
  left_join(mål_sæson_hjemme, by = c("sæson", "runde", "hjemmehold")) |>
  left_join(mål_sæson_ude,   by = c("sæson", "runde", "udehold"))

```

Gemmer ændringer.

```
saveRDS(kampe_df, "VFFdata/superstats.rds")
```

Load ændringer.

```
kampe_df <- readRDS("VFFdata/superstats.rds")
```

Udvælger alle rækker hvor hjemmeholdet er VFF. Det giver en oversigt over alle VFF's hjemmekampe i datasættet.

```
vff_hjemmekampe <- kampe_df |>  
  filter(hjemmehold == "VFF")
```

Test af hjemmekampene. Output: 318. Det er klasse, da VFF ifølge Superstats har spillet 327 hjemmekampe all time. Sæsonen 1993/1994 indeholdt 9 hjemmekampe, så alle kampene er succesfuldt kommet med i tabellen.

```
paste("Antal hjemmekampe i perioden for VFF er:", nrow(vff_hjemmekampe))
```

```
[1] "Antal hjemmekampe i perioden for VFF er: 318"
```

Tilføjer A, B, C og D kampe. Samt en beregning af uddnyttelsesgraden på Energi Viborg Arena.

```
vff_hjemmekampe <- vff_hjemmekampe |>  
  # Opretter "udehold_rang" baseret på modstanderens styrke.  
  mutate(udehold_rang = case_when(  
    udehold %in% c("AGF", "FCM", "BIF", "FCK") ~ "A",  
    udehold %in% c("Aab", "RFC", "SIF", "VB", "OB") ~ "B",  
    udehold %in% c("FCN", "SJF", "FCF", "LBK") ~ "C",  
    TRUE ~ "D")) |> # Alle andre.  
  # Uddnyttelsesgrad: en simpel proxy = tilskuere / stadionkapaciteten.  
  mutate(udnyttelsesgrad_tilskuere = tilskuere / 9695)
```

Ser navnene på alle kolonnerne.

```
names(vff_hjemmekampe)
```

```
[1] "ugedag"                      "hjemmehold"  
[3] "udehold"                     "mål_hjemme"  
[5] "mål_ude"                     "tilskuere"  
[7] "dommer"                       "sæson"  
[9] "runde"                        "tidsstempel"  
[11] "tidsperiode"                 "point_hjemmehold"  
[13] "point_udehold"                "mål_seneste_3_hjemme"  
[15] "tilskuere_hold_lag"          "form_seneste_3_hjemme"  
[17] "placering_lag_hjemme"        "placering_lag_ude"  
[19] "form_seneste_3_hjemmehold"   "form_seneste_3_udehold"  
[21] "mål_sæson_lag_hjemmehold"   "mål_sæson_lag_udehold"  
[23] "udehold_rang"                "udnyttelsesgrad_tilskuere"
```

Tæller antallet af kolonner.

```
ncol(vff_hjemmekampe) # Output: 24.
```

```
[1] 24
```

Vælger rækkefølgen af kolonnerne med select.

```
vff_hjemmekampe <- vff_hjemmekampe |>
  select(ugedag, tidsperiode, tidsstempel, runde, sæson,
         hjemmekold, udehold, mål_hjemme, mål_ude, mål_sæson_lag_hjemmekold, mål_sæson_lag_udehold,
         tilskuer, point_hjemmekold, point_udehold, placering_lag_hjemme,
         placering_lag_ude, mål_seneste_3_hjemme, form_seneste_3_hjemme,
         form_seneste_3_hjemmekold, form_seneste_3_udehold, tilskuer_hold_lag,
         udnyttelsesgrad_tilskuer, udehold_rang, dommer
  ) |>
  arrange(desc(tidsstempel)) # Faldende rækkefølge.
```

Gemmer ændringer.

```
saveRDS(vff_hjemmekampe, "VFFdata/superstatsvff.rds")
```

.2.2 Data fra DMI

Upload af relevante pakker.

```
pacman::p_load("tidyverse", "rvest", "stringr", "janitor", "lubridate", "RSQLite")
```

Vi starter med at indlæse Superstatsdata, da kampdatoerne skal bruges til at hente vejrdata.

```
vff_hjemmekampe <- read_rds("VFFdata/superstatsvff.rds")
```

Hvis der findes en .Renvironment-fil, så læses den ind her.

```
if (file.exists(".Renvironment")) readRenvironment(".Renvironment")
```

En .Renvironment-fil bruges til, at R automatisk kan læse værdier feks. API-nøgler, selvom de ikke er synlige i selve koden.

Sys.getenv() bruges i R til at hente oplysninger gemt i f.eks. .Renvironment. Nøglen er døbt MY_API_KEY i .Renvironment og derfor den der hentes. Den returnerer i dette tilfælde en API-nøgle til DMI, da det er den der er gemt i filen.

```
api_nøgle <- Sys.getenv("MY_API_KEY")
usethis::edit_r_environment()
```

```
[ ] Edit '/Users/mathiasandersen/.Renvironment'.
```

```
[ ] Restart R for changes to take effect.
```

Kampdatoerne for VFF's hjemmekampe gemmes i kamp_dataer. Skal bruges i et loop, så der kun kommer vejrobservationer fra DMI på kampdage.

```
kamp_datoer <- vff_hjemmekampe$tidsstempel
```

Opretter en tom liste og gemmer i dmi_list.

```
dmi_list <- list()
```

Looper gennem alle kampdatoer en ad gangen og henter DMI data. Seq_along(kamp_datoer) laver en sekvens af 1, 2, 3, osv. i bliver derfor hvert enkelt kampnummer i rækkefølgen.

Formaterer kampens tidsstempel til det format DMI-API'en kræver. format() omdanner dato/tid til en tekststreng i et bestemt format. Her hentes dataen fra (DMI 2022)

```
for (i in seq_along(kamp_datoer)) {  
  dato_format <- format(kamp_datoer[i], "%Y-%m-%dT%H:%M:%SZ")  
  #> URL-koder dato'en så den kan bruges i API-kaldet.  
  #> URLencode() oversætter tegn som specialtegn.  
  #> F.eks. bliver 2023-07-12 19:00 til "2023-07-12%2019%3A00". så datoerne de kan bruges i URL'en.  
  dato_encode <- URLencode(dato_format)  
  #> Bygger URL'en der henter temperatur og vind for kampen index i. Alle kampe.  
  #> paste0() sætter tekst sammen uden mellemrum  
  url_temp_vind <- paste0(  
    "https://dmigw.govcloud.dk/v2/metObs/collections/observation/items?",  
    "stationId=06060",  
    "&datetime=", dato_encode,  
    "&api-key=", api_nøgle)  
  #> Henter DMI's data for kampene.  
  #> httr::GET() sender en HTTP-forespørgsel af typen GET.  
  #> Det betyder, at serveren bliver bedt om at sende data fra det angivne URL.  
  api_kald <- httr::GET(url_temp_vind)  
  #> httr::content() trækker indholdet ud som tekst.  
  #> Læser API-svaret som ren JSON-tekst  
  tekst <- httr::content(api_kald, "text", encoding = "UTF-8")  
  #> jsonlite::fromJSON() omdanner JSON-tekst til en R-liste  
  #> Konverterer data så de kan gemmes i dmi_list.  
  dmi_list[[i]] <- jsonlite::fromJSON(tekst)  
  #> message() skriver til konsollen.  
  #> Viser løbende hvilket tidspunkt der hentes data for.  
  message("Henter data for:", dato_format, "\n")  
}
```

lapply() kører en funktion på hvert element listen og returnerer en ny liste. function(x) ... hvad der skal trækkes ud fra hvert liste-element. x\$features\$properties henter den del af JSON-dataen, hvor DMI gemmer selve målingerne (temperatur, vind og andre variabler) \$ bruges til at tilgå properties i listen. Resultatet bliver en ny liste (dmi_properties), hvor hvert element indeholder properties for en kamp.

```
dmi_properties <- lapply(dmi_list, function(x) x$features$properties)
```

bind_rows() samler alle liste-elementer i en samlet data frame. .id = "kamp" tilføjer en kolonne med kampnummer. mutate() tilføjer/ændrer kolonner: kamp = konverterer kamp-id til heltal as.integer. kamp_dato = henter korrekt kampdato med kampnummeret.

```
dmi_df <- bind_rows(dmi_properties, .id = "kamp") |>  
  dplyr::mutate(kamp = as.integer(kamp), kamp_dato = kamp_datoer[kamp])
```

filter() vælger kun rækker, der opfylder betingelsen: %in% tester om parameterId findes i den angivne liste og beholder kun de to parametre, som skal bruges = temperatur og vind.

```
dmi_df <- dmi_df |>
  filter(parameterId %in% c("temp_dry", "wind_speed"))
```

group_by() grupperer data efter kamp og parameter = temperatur/vind. slice_max() vælger den senest observerede måling. ungroup() fjerner gruppering så df fungerer normalt igen. Ender med eb temperatur og en vindmåling per kamp.

```
dmi_df <- dmi_df |>
  group_by(kamp, parameterId) |>
  slice_max(observed) |>
  ungroup()
```

select() vælger kun de kolonner, der skal bruges videre i tabellen. value = den målte værdi = temperatur eller vindstyrke. pivot_wider() omdanner data fra vertikalt format til horisontalt format. id_cols angiver hvilke kolonner der skal være unikke rækker = kamp + dato. names_from bestemmer hvilke værdier der bliver til nye kolonnennavne = temp/vind.

values_from siger hvor måleværdierne skal hentes fra.

```
dmi_df <- dmi_df |>
  select(kamp, kamp_dato, parameterId, value) |>
  pivot_wider(id_cols = c(kamp, kamp_dato),
              names_from = parameterId,
              values_from = value)
```

Funktion der henter den samlede nedbør 7 timer før en kamp.

```
nedbør_7t <- function(dt) {

  # format() laver dato/tid om til tekst i API-formatet "YYYY-MM-DDTHH:MM:SSZ"
  start <- format(dt - 7*3600, "%Y-%m-%dT%H:%M:%SZ")
  slut   <- format(dt,           "%Y-%m-%dT%H:%M:%SZ")

  # paste0() sætter URL'en sammen.
  url_nedbør <- paste0(
    "https://dmigw.govcloud.dk/v2/metObs/collections/observation/items?",
    "stationId=06060",                                # Station Karup.
    "&parameterId=precip_past1h",                     # Nedbør målt pr time.
    "&datetime=", start, "/", slut,                  # Tidsinterval.
    "&limit=8",                                       # Max antal observationer.
    "&api-key=", api_nøgle) # API-nøgle.

  #> httr::GET() henter data fra API.
  #> httr::content() trækker selve svaret ud som tekst.
  tekstu1 <- httr::content(httr::GET(url_nedbør), "text", encoding = "UTF-8")

  # jsonlite::fromJSON() forsøger at omdanne JSON-tekst til en liste.
  js <- try(jsonlite::fromJSON(tekstu1, simplifyVector = TRUE), silent = TRUE)

  # Hvis der er fejl i JSON'en (fx tomt svar) returnerer vi NA.
  if (inherits(js, "try-error")) return(NA)}
```

```

if (is.null(js$features) || length(js$features) == 0) return(NA)

# Henter selve nedbørsværdierne fra properties$value.
values <- js$features$properties$value

# Udskriver status i konsollen.
message("Henter 7-timers nedbør for:", format(dt, "%Y-%m-%d %H:%M"), "\n")

# Summerer nedbør over perioden (NA ignoreres).
sum(values, na.rm = TRUE)
}

```

Beregn nedbør for alle kampdatoer ved at køre funktionen på hver dato.

```
nedbør7t <- sapply(kamp_datoer, nedbør_7t)
```

Tilføj nedbørsværdien til dmi_df ved at matche på kampnummer.

```
dmi_df$nedbør_seneste_7_timer <- nedbør7t[dmi_df$kamp]
```

Sorter tabellen, så de nyeste kampe kommer øverst = faldende rækkefølge.

```
dmi_df <- dmi_df |>
  arrange(desc(kamp_dato))
```

Join med de andre to variabler.

```
dmi_df <- dmi_df |>
  left_join(
    vff_hjemmekampe |>
      mutate(kamp = row_number()) |>
      select(kamp),
    by = "kamp")
```

Her gemmes DMI dataerne, som en RDS-fil

```
saveRDS(dmi_df, "VFFdata/dmi.rds")
```

.2.3 Data fra nager.at (helligdage og feriedage)

Her defineres URL og parameteren. Her hentes dataen fra (nager.at 2025)

```
BASE_URL <- "https://date.nager.at/api/v3/PublicHolidays/"
LAND <- "/DK" # Landekode for Danmark
```

Her defineres årsintervallet for dataindsamlingen (Matcher kamp-data perioden)

```
aar_super <- 2003:2026
```

Her oprettes en tom tibble under "helligdage" objektet. Variablene konveteres til de rigtige formater (navn = "chr" og dato = "date")

```

helligdage <- tibble(
  navn = character(),
  dato = as.Date(character())
)

```

Loop gennem hvert år og hent hellige dage fra date.nager Url laves, så den kører de specifikke år igennem

```

for (aar in aar_super) {
  url <- paste0(BASE_URL, aar, LAND)
  response <- httr::GET(url)

  data_raw <- httr::content(response, as = "text", encoding = "UTF-8")
  parsed <- jsonlite::fromJSON(data_raw)

  helligdage_aar <- parsed |>
    as_tibble() |>
    transmute(
      navn = localName,
      dato = as.Date(date)
    )

  helligdage <- bind_rows(helligdage, helligdage_aar)
}

```

Load af vff hjemmekamp-data RDS-fil

```
VFF_Hjemmekampe <- read_rds("VFFdata/superstatsvff.rds")
```

Vi konverterer tidsstempel til date-format for at muliggøre join mellem de to tabeller

```

VFF_Hjemmekampe1 <- VFF_Hjemmekampe |>
  mutate(
    Dato_join = as.Date(tidsstempel)
  )

```

Left join for at tilføje helligdagsinformation

```

VFF_Komplet <- VFF_Hjemmekampe1 |>
  left_join(
    helligdage,
    by = c("Dato_join" = "dato")
  ) |>

  mutate(
    Er_helligdag = if_else(!is.na(navn), 1, 0),
    Helligdag_navn = navn
  ) |>
  dplyr::select(-navn)

```

Opgør antal kampe på helligdage vs. almindelige dage. Her skal resultatet være 15

```
VFF_Komplet |>
  count(Er_helligdag)
```

```
# A tibble: 2 x 2
  Er_helligdag     n
  <dbl> <int>
1          0    303
2          1     15
```

Fjern hjælpekolonne og filtrér data fra sæson 2003/2004 og fremefter

```
VFF_Komplet <- VFF_Komplet |>
  dplyr::select(-Dato_join) |>
  filter(as.numeric(str_extract(sæson, "^\d{4}")) >= 2003)
```

Tilføjelse af feriedage

```
VFF_Komplet1 <- VFF_Komplet |>
  mutate(
    # Konvertér tidsstempel til Date-objekt
    Dato_kamp = as_date(ymd_hms(tidsstempel)), # eller ymd_hms(tidsstempel) for POSIXct

    # Udtræk måned og ISO-ugenummer fra kampdato
    maaned = month(Dato_kamp),
    uge = isoweek(Dato_kamp),

    # Definér ferieperioder baseret på danske skoleferier

    # Sommerferie: 1. juli til 15. august
    i_sommerferie = Dato_kamp >= make_date(year(Dato_kamp), 7, 1) &
      Dato_kamp < make_date(year(Dato_kamp), 8, 15),

    # Efterårsferie: Uge 42 + weekend i uge 41 (lørdag-søndag)
    i_efterårsferie = uge == 42 | (uge == 41 & wday(Dato_kamp, week_start = 1) >= 6),

    # Vinterferie: Uge 7 (typisk i februar)
    i_vinterferie = uge == 7,

    # Kategorisér ferieperiode i tekstformat
    ferieperiode = case_when(
      i_sommerferie ~ "Sommerferie",
      i_efterårsferie ~ "Efterårsferie",
      i_vinterferie ~ "Vinterferie",
      TRUE ~ "Ingen"
    ),

    # Ekstraher klokkeslæt fra tidsstempel (format HH:MM)
    Tid_kamp = format(ymd_hms(tidsstempel), "%H:%M"))
  )
```

Her gemmes den endelige fil i RDS-formatet

```
saveRDS(VFF_Komplet1, file = "VFFdata/VFF_Komplet.rds")
```

.2.4 Transfer data

Transferdata har været meget svære at få fingrene i, da de fleste hjemmesider med data på området er utilgængelige for os. Vi har derfor fundet et open-source-projekt, skabt af en eller flere privatpersoner, som gør det lettere at hente fodbolddata i R. Pakken hedder worldfootballR (JaseZiv 2025). Pakken blev den 18. september 2025 arkiveret og bliver ikke længere vedligeholdt. Det kan give udfordringer i forhold til reproducerbarhed fremadrettet. Hvis det viser sig, at transferdata har en signifikant betydning for antallet af tilskuere til VFF's hjemmekampe, er antagelsen dog, at VFF selv har adgang til deres interne transferdata. Disse data har vi dog ikke adgang til på nuværende tidspunkt. Kør installationen herunder, hvis du ikke allerede har installeret devtools. devtools bruges til at installere R-pakker direkte fra GitHub. worldfootballR indeholder blandt andet transferdata fra Transfermarkt(en af verdens største offentligt tilgængelige databaser for spillertransfers), som ikke kan webscrapes direkte. Derfor benyttes denne metode i stedet. Argumentet force = TRUE betyder, at pakken geninstalleres, selv hvis den allerede er installeret i samme version. Hvis R spørger om opdatering af afhængige pakker, vælges 3 for "None", så eksisterende pakker ikke opdateres, da det ikke er nødvendigt. Her er koden der kan køres for at installere pakken: devtools::install_github("JaseZiv/worldfootballR", force = TRUE).

Indlæsning af nødvendige pakker.

```
pacman::p_load(worldfootballR, tidyverse, janitor)
```

Mappen VFFdata til opbevaring af data. Hvis den ikke eksistere, så bliver den oprettet.

```
if (!dir.exists("VFFdata")) dir.create("VFFdata")
```

Indlæser datasæt med Viborgs hjemmekampe i Superligaen, som blandt andet indeholder informationer om hvilke sæsoner de har været i Superligaen, som skal bruges til at hente transferdata fra de korrekte år.

```
vff_hjemmekampe <- readRDS("VFFdata/superstatsvff.rds")
sæsoner <- vff_hjemmekampe |>
  distinct(sæson) |> # distinct(sæson) sikrer, at hver sæson kun optræder en gang.
  pull(sæson) # pull(sæson) konverterer kolonnen til en vektor.
```

Når der hentes data fra Transfermarkt, identificeres sæsoner ved den pågældende sæsons startår. substr(1, 4) udtrækker de første fire tegn i sæson strengen, f.eks. 2002 fra 2002/2003 osv. as.integer() konverterer værdien fra tekst til en numerisk værdi for årstallet.

```
år <- sæsoner |>
  substr(1, 4) |>
  as.integer()
```

Fjerner år før 2002, da det er fra dette år og frem vi ser på VFF's kampe. Ved kun at vælge år, som er større eller lig med 2002.

```
år <- år[år >= 2002]
```

Oprettet en tom liste. Listen udvides løbende i loopet, hvor hvert element bliver en dataframe med transferdata for hver sæson.

```
list_of_transfers <- list()
```

Gennemløber alle startår og henter transferdata fra Transfermarkt. seq_along(år) opretter en sekvens fra 1 til længden af vektoren år. For loopet kører derfor en gang for hvert år i vektoren år.

```

for (i in seq_along(år)) {
  # Henter et startår ad gangen fra vektoren år.
  y <- år[i]
  #> Opbygger URL til Viborg FF's Transfermarkt-side for den givne sæson.
  #> paste0 sammensætter året og URL'en uden mellemrum til en samlet streng.
  url <- paste0("https://www.transfermarkt.com/viborg-ff/startseite/verein/1063/saison_id/", y)
  #> Henter alle transfers (sommer + vinter) for den pågældende sæson.
  #> og gemmer resultatet i listen, som f.eks. Viborg_2002.
  #> tm_team_transfers() henter alle spillertransfers for VFF.
  list_of_transfers[[paste0("Viborg_", y)]] <- tm_team_transfers()
  team_url = url,                      # team_url angiver URL'en til holdets Transfermarkt side.
  transfer_window = "all")             # transfer_window = "all" betyder, at både sommer og vintertransfers
  # Udskriver status i konsollen, så kan man se når data er hentet..
  message("Henter data fra året :", y, "\n")
}

```

Kombinerer alle lister til en samlet dataframe. bind_rows() binder alle tabellerne sammen under hinanden og laver et stort datasæt i en tibble.

```
total_transfers <- bind_rows(list_of_transfers)
```

Fjern kolonner der ikke skal bruges. Med select og - foran de kolonner der ikke skal bruges, så de fravælges. dplyr:: er benytet da nogen får fejl ved select uden dplyr:: foran, da andre pakker i R har en select funktion.

```

total_transfers <- total_transfers |>
  dplyr::select(-team_name, -league, -country, -player_url, -player_position,
                -player_age, -player_nationality, -club_2, -league_2,
                -country_2, -in_squad, -appearances, -goals,
                -minutes_played, -transfer_notes)

```

Gør kolonnenavn konsistente og nemme at arbejde med i R med janitor pakken. clean_names() omdanner f.eks. Transfer Fee til transfer_fee.

```
total_transfers_clean <- total_transfers |>
  clean_names()                                # Renser kolonnenavnene med clean_names() fra janitor.
```

Sæson står lige nu som et enkelt årstal. Laves om til korrekt sæsonformat f.eks. fra 2025 til 2025/2026. Opretter en ny sæson kolonne med den nuværende ved hjælp af mutate, som bruger den oprindelige kolonne til at lave en ny.

```

total_transfers_clean <- total_transfers_clean |>
  #> Opretter sæsonformatet YYYY/YYYY+1 ud fra sæsonens startår.
  #> Sammensættes med paste0 med / imellem start og slut år. + 1 giver korrekt slut år i sæson strengen
  #> as.numeric, som numerisk værdi.
  mutate(season_format = paste0(season, "/", as.numeric(season) + 1))

```

Fjerner den oprindelige sæson kolonne med kun et enkelt årstal. Ingen bruges select(med - foran den kolonne, som defor fravælges).

```
total_transfers_clean <- total_transfers_clean |>
  dplyr::select(-season)
```

Fjerner alle transfers der er registreret som lån. Ved at filtrere på kolonnen is_loan og kun beholde rækker hvor is_loan == FALSE.

```
transfers <- total_transfers_clean |>
  filter(is_loan == FALSE)
```

Fjerner observationer uden registreret transfersum ved at filtrere alle rækker fra hvor transfer_fee er NA. !is.na filtrerer rækker fra med manglende transfersum og beholder kun observationer med en transfer_fee.

```
transfers <- transfers |>
  filter(!is.na(transfer_fee))
```

Beregner samlet transferøkonomi pr. sæson samt antallet af transfers.

```
total_all <- transfers |>
  # Grupperer data på sæson, så beregningerne foretages separat for hver sæson.
  group_by(season_format) |>
  # Opsummerer transferdata for hver sæson.
  # Tager mange rækker pr. sæson og samler dem til en række pr. sæson.
  summarise(
    #> Beregner samlede transferindtægter pr. sæson.
    #> Udvælger kun transfers hvor transfer_type er "Departures" som er spiller afgang.
    #> og summerer transfer_fee for dem.
    total_income = sum(transfer_fee[transfer_type == "Departures"]),
    #> Beregner samlede transferudgifter for sæsonen (spillerkøb).
    #> Samme koncept som før. Bare med spiller tilgange (Arrivals)
    total_expense = sum(transfer_fee[transfer_type == "Arrivals"]),
    # Nettobalance beregnes som indtægter minus udgifter.
    net_balance = total_income - total_expense,
    #> Antal transfers i sæsonen (både køb og salg).
    #> Tæller antallet af observationer for hver sæson.
    #> n() svarer til det samlede antal transfers i sæsonen.
    antal_transfers = n()) |>
  #> Fjerner gruppering efter summarise.
  #> Så datasættet igen behandles som et almindeligt ugrupperet datasæt.
  ungroup() |>
  #> Sorterer sæsonerne i faldende rækkefølge (nyeste først).
  #> arrange(desc) sørger for at arrangere i faldende rækkefølge.
  arrange(desc(season_format))
```

Gemmer transfer data for VFF, som en RDS-fil.

```
saveRDS(total_all, "VFFdata/TransfermarktData.rds")
```

.2.5 Bjarnes data

Indlæsning af uleverede RDS-filer fra Bjarne.

```
data_vffkort01_rds <- readRDS("VFFdata/vffkort01.rds")
```

```
data_fcidk_rds <- readRDS("VFFdata/fcidk.rds")
```

.2.6 RSQLight

Vi starter med at indlæse de nødvendige pakker. Tidyverse bruges til databehandling. RSQlite bruges til at arbejde med databaser.

```
pacman::p_load("tidyverse", "RSQLite")
```

Oprettelse af SQLite database

```
if (!dir.exists("VFFdata")) dir.create("VFFdata")
```

Vi opretter forbindelse til SQLite databasen.

```
connection_VFFdata <- dbConnect(SQLite(), "VFFdata/fodbolddata.sqlite")
```

Vi indlæser VFF kampdata og fjerner helligdag-navn.

```
VFF_Komplet <- readRDS("VFFdata/VFF_Komplet.rds") |>
  select(-Helligdag_navn)
```

Vi indlæser DMI vejrdata og fjerner rækker uden kampdatoer.

```
DMI_data <- readRDS("VFFdata/dmi.rds") |>
  filter(!is.na(kamp_dato))
```

Indløser Transfermarkt_data og fjerner rækker uden sæson informationer

```
Transfermarkt_data <- readRDS("VFFdata/TransfermarktData.rds") |>
  filter(!is.na(season_format))
```

Indlæser informationer om tilskuerne vffkort

```
data_vffkort01_rds <- readRDS("VFFdata/vffkort01.rds")
```

Indlæser informationer om modstanderne med fcidk.

```
data_fcidk_rds <- readRDS("VFFdata/fcidk.rds")
```

Vi smider nu data i SQLite dbWriteTable = Vi gemmer en dataframe permanent i en database som en tabel.
overwrite = Vi sikre at den gamle data overskrives hver gang vi kører koden.

```
dbWriteTable(connection_VFFdata, "vff", data_vffkort01_rds, overwrite = TRUE)
dbWriteTable(connection_VFFdata, "fci", data_fcidk_rds, overwrite = TRUE)
dbWriteTable(connection_VFFdata, "DMI_data", DMI_data, overwrite = TRUE)
dbWriteTable(connection_VFFdata, "VFF_Komplet", VFF_Komplet, overwrite = TRUE)
dbWriteTable(connection_VFFdata, "Transfermarkt_data", Transfermarkt_data, overwrite = TRUE)
```

Vi ser alle tabeller der findes i databasen.

```
dbListTables(connection_VFFdata)
```

```
[1] "DMI_data"           "Transfermarkt_data" "VFF_Komplet"
[4] "db_fcidk"          "db_vff"            "fci"
[7] "vff"
```

Vi lukker databasen.

```
dbDisconnect(connection_VFFdata)
```

vi genåbner forbindelsen til databasen.

```
connection_VFFdata <- dbConnect(SQLite(), "VFFdata/fodbolddata.sqlite")
```

dbGetQuery henter data fra databasen og laver dem om til en dataframe. Vi henter alle kolonner med SELECT

```
SQL_DMI <- dbGetQuery(connection_VFFdata, "SELECT * FROM DMI_data")
SQL_VFF_Komplet <- dbGetQuery(connection_VFFdata, "SELECT * FROM VFF_Komplet")
SQL_Transfermarkt <- dbGetQuery(connection_VFFdata, "SELECT * FROM Transfermarkt_data")
SQL_vff <- dbGetQuery(connection_VFFdata, "SELECT * FROM vff")
SQL_fci <- dbGetQuery(connection_VFFdata, "SELECT * FROM fci")
```

Vi lukker databasen.

```
dbDisconnect(connection_VFFdata)
```

Vi åbner forbindelsen.

```
connection_VFFdata <- dbConnect(SQLite(), "VFFdata/fodbolddata.sqlite")
```

Vi samler alle kampdata og får en komplet tabel over alle kampe.

```
sql_samlet <- dbGetQuery(connection_VFFdata, "
SELECT
    datetime(D.kamp_dato, 'unixepoch') AS kampdato,
    D.temp_dry,
    D.wind_speed,
    D.nedbør_seneste_7_timer,
    V.ugedag,
    V.hjemmehold,
    V.udehold,
    V.mål_hjemme,
    V.mål_udet,
    V.tilskuere,
    dommer,
    V.point_hjemmehold,
    V.point_udehold,
    V.placerings_lag_hjemme,
    V.placerings_lag_udet,
    V.mål_seneste_3_hjemme,
    V.mål_sæson_lag_hjemmehold,
    V.mål_sæson_lag_udehold,
    V.form_seneste_3_hjemmehold,
    V.form_seneste_3_udet,
    V.udnyttelsesgrad_tilskuere,
    V.tilskuere_hold_lag,
    V.udehold_rang,
    V.sæson,
    V.runde,
    V.tidsperiode,
    V.Er_helligdag,
```

```

V.i_sommerferie,
V.i_efterårsferie,
V.i_vinterferie,
V.ferieperiode,
T.total_income,
T.total_expense,
T.net_balance,
T.antal_transfers,
C.d10_tilskuere,
C.d7_tilskuere,
C.d3_tilskuere,
F.navn AS udehold_navn,
F.stadion,
F.sponsor_stadion_navn,
F.tilskuere AS stadion_kapacitet

FROM VFF_Komplet AS V

LEFT JOIN DMI_data AS D
    ON D.kamp_dato = V.tidsstempel

LEFT JOIN Transfermarkt_data AS T
    ON V.sæson = T.season_format

LEFT JOIN (
    SELECT
        sæson,
        runde,
        tilskuere,
        MAX(d10_tilskuere) AS d10_tilskuere,
        MAX(d7_tilskuere) AS d7_tilskuere,
        MAX(d3_tilskuere) AS d3_tilskuere
    FROM vff
    GROUP BY sæson, runde, tilskuere HAVING COUNT(*) >= 1
) AS C
    ON V.sæson = C.sæson
    AND V.runde = C.runde
    AND V.tilskuere = C.tilskuere

LEFT JOIN fci AS F
    ON CASE
        WHEN V.udehold = 'SJF' THEN 'SDR'
        ELSE V.udehold
    END = F.kort

WHERE V.sæson BETWEEN '2003/2004' AND '2025/2026'
")

```

Vi lukker databasen.

```
dbDisconnect(connection_VFFdata)
```

Vi gemmer filen, som en RDS

```
saveRDS(sql_samlet, "VFFdata/VFF_samlet_variabler.rds")
```

.2.7 Download joinet datasæt

Indlæser nødvendige pakker.

```
pacman::p_load("tidyverse", "forcats", "ggplot2", "glmnet", "leaps", "caret", "purrr", "car", "tibble")
```

Indlæser joinet datasæt.

```
VFF_samlet_variabler <- readRDS("VFFdata/VFF_samlet_variabler.rds")
```

.2.8 Preprocess

Vi konverter relevante variabler til faktorer. Vi bruger mutate til at bygge videre på de samlet variabler, og definer dem “as factor”. Dette er vigtigt i forbindelse med lineær regression, hvor faktorvariable behandles som kategoriske og derfor omsættes til dummyvariable i matricen.

```
VFF_samlet_variabler <- VFF_samlet_variabler |>
  mutate(
    udehold_rang      = as.factor(udehold_rang),
    Er_helligdag     = as.factor(Er_helligdag),
    i_sommerferie    = as.factor(i_sommerferie),
    i_efterårsferie  = as.factor(i_efterårsferie),
    i_vinterferie    = as.factor(i_vinterferie),
    ferieperiode     = as.factor(ferieperiode),
    udehold          = as.factor(udehold)
  )
```

Vi samler udehold der kun optræder få gange i en fælles kategori. Vi tæller hvor mange gange et hold opretter, og laver en ny kolonne der hedder “n_udehold” som tæller antallet af gange holdet gentages i datasættet. Vi bruger mutate og if_else til at definerer at et “stor hold” er et hold hvor observation gentages mere end 2 gange. Dvs hold der gentages 2 eller færre gange defineres efterfølgende som faktoren “små hold”. Til sidst bruger vi select til at fjerne kolonnen “n_udehold”.

```
VFF_samlet_variabler <- VFF_samlet_variabler |>
  add_count(udehold, name = "n_udehold") |>
  mutate(
    små_hold = if_else(n_udehold > 2, "stor hold", as.character(udehold)),
    små_hold = as.factor(små_hold)
  ) |>
  select(-n_udehold)
```

.3 Modellerung

Vi bruger set.seed til at definere, at vi er gruppe 1. nrow tæller hvor rækker (dvs. kampe) der er i vores datasæt. vi gemmer det i et objekt vi kalder “n”. Vi opdeler efterfølgende dataene helt tilfældigt 70/30 procent. Vi definer vores træningsindex som 70%, og vores test data som minus træningsindex, dvs 100 procent minus 70 procent, som gir 30 procent.

```

set.seed(1)
n <- nrow(VFF_samlet_variabler)
train_index <- sample(1:n, size = round(0.7*n))
train_data <- VFF_samlet_variabler[train_index, ]
test_data <- VFF_samlet_variabler[-train_index, ]

```

Vi sikrer at test_data kun har niveauer, som modellen kender. Vi bruger mutate til at bygge videre på test_dataene, og sørger for, udehold gendannes som faktor med samme levels som i træningsdataene. Levels(train_data\$udehold henter alle de niveauer, som udehold har i trænings-datasættet.

```

test_data <- test_data |>
  mutate(
    udehold = factor(udehold, levels = levels(train_data$udehold)))

```

.3.1 Stor model - lineær regression med alle mulige variabler

Vi laver en stor lineær regressionsmodel for at afdække potentielle sammenhænge i vores mange forklarende variabler.

```

stor_lm_model <- lm(
  tilskuere ~ temp_dry + wind_speed + nedbør_seneste_7_timer +
  ugedag + placering_lag_hjemme + placering_lag_udet + mål_seneste_3_hjemme +
  mål_sæson_lag_hjemmehold + mål_sæson_lag_udet + form_seneste_3_hjemmehold +
  form_seneste_3_udet + tilskuere_hold_lag + udehold_rang + sæson + runde + tidsperiode +
  Er_helligdag + i_sommerferie + i_efterårsferie + i_vinterferie +
  ferieperiode + total_income + total_expense + net_balance +
  antal_transfers + d3_tilskuere + stadion_kapacitet,
  data = train_data
)

```

Test for multikollinearitet - VIF (Variance inflation factor) via car-pakken VIF-testen fejler, fordi vi har perfekt multikollinearitet

```

# vif(stor_lm_model)
# Udkommenteret da Quattro ikke kan håndtere multikollinearitet.

```

Vi finder de perfekte lineære afhængigheder

```
alias(stor_lm_model)
```

```

Model :
tilskuere ~ temp_dry + wind_speed + nedbør_seneste_7_timer +
  ugedag + placering_lag_hjemme + placering_lag_udet + mål_seneste_3_hjemme +
  mål_sæson_lag_hjemmehold + mål_sæson_lag_udet + form_seneste_3_hjemmehold +
  form_seneste_3_udet + tilskuere_hold_lag + udehold_rang +
  sæson + runde + tidsperiode + Er_helligdag + i_sommerferie +
  i_efterårsferie + i_vinterferie + ferieperiode + total_income +
  total_expense + net_balance + antal_transfers + d3_tilskuere +
  stadion_kapacitet

```

```

Complete :
          (Intercept) temp_dry wind_speed

```

tidsperiodeLørdag_eftermiddag	0	0	0
tidsperiodeMandag_formiddag	0	0	0
tidsperiodeOnsdag_aften	0	0	0
tidsperiodeSøndag_formiddag	0	0	0
tidsperiodeTordag_eftermiddag	0	0	0
ferieperiodeIngen	1	0	0
ferieperiodeSommerferie	0	0	0
ferieperiodeVinterferie	0	0	0
total_income	8e+05	0	0
total_expense	120000	0	0
net_balance	680000	0	0
antal_transfers	13	0	0
	nedbør_seneste_7_timer ugedagLørdag ugedagMandag		
tidsperiodeLørdag_eftermiddag	0	1	0
tidsperiodeMandag_formiddag	0	0	1
tidsperiodeOnsdag_aften	0	0	0
tidsperiodeSøndag_formiddag	0	0	0
tidsperiodeTordag_eftermiddag	0	0	0
ferieperiodeIngen	0	0	0
ferieperiodeSommerferie	0	0	0
ferieperiodeVinterferie	0	0	0
total_income	0	0	0
total_expense	0	0	0
net_balance	0	0	0
antal_transfers	0	0	0
	ugedagOnsdag ugedagSøndag ugedagTordag		
tidsperiodeLørdag_eftermiddag	0	0	0
tidsperiodeMandag_formiddag	0	0	0
tidsperiodeOnsdag_aften	1	0	0
tidsperiodeSøndag_formiddag	0	1	0
tidsperiodeTordag_eftermiddag	0	0	1
ferieperiodeIngen	0	0	0
ferieperiodeSommerferie	0	0	0
ferieperiodeVinterferie	0	0	0
total_income	0	0	0
total_expense	0	0	0
net_balance	0	0	0
antal_transfers	0	0	0
	placering_lag_hjemme placering_lag_udé		
tidsperiodeLørdag_eftermiddag	0	0	0
tidsperiodeMandag_formiddag	0	0	0
tidsperiodeOnsdag_aften	0	0	0
tidsperiodeSøndag_formiddag	0	0	0
tidsperiodeTordag_eftermiddag	0	0	0
ferieperiodeIngen	0	0	0
ferieperiodeSommerferie	0	0	0
ferieperiodeVinterferie	0	0	0
total_income	0	0	0
total_expense	0	0	0
net_balance	0	0	0
antal_transfers	0	0	0
	mål_seneste_3_hjemme mål_sæson_lag_hjemmehold		
tidsperiodeLørdag_eftermiddag	0	0	0
tidsperiodeMandag_formiddag	0	0	0
tidsperiodeOnsdag_aften	0	0	0
tidsperiodeSøndag_formiddag	0	0	0

tidsperiodeTordag_eftermiddag	0	0	
ferieperiodeIngen	0	0	
ferieperiodeSommerferie	0	0	
ferieperiodeVinterferie	0	0	
total_income	0	0	
total_expense	0	0	
net_balance	0	0	
antal_transfers	0	0	
	mål_sæson_lag_udehold	form_seneste_3_hjemmehold	
tidsperiodeLørdag_eftermiddag	0	0	
tidsperiodeMandag_formiddag	0	0	
tidsperiodeOnsdag_aften	0	0	
tidsperiodeSøndag_formiddag	0	0	
tidsperiodeTordag_eftermiddag	0	0	
ferieperiodeIngen	0	0	
ferieperiodeSommerferie	0	0	
ferieperiodeVinterferie	0	0	
total_income	0	0	
total_expense	0	0	
net_balance	0	0	
antal_transfers	0	0	
	form_seneste_3_udehold	tilskuere_hold_lag	
tidsperiodeLørdag_eftermiddag	0	0	
tidsperiodeMandag_formiddag	0	0	
tidsperiodeOnsdag_aften	0	0	
tidsperiodeSøndag_formiddag	0	0	
tidsperiodeTordag_eftermiddag	0	0	
ferieperiodeIngen	0	0	
ferieperiodeSommerferie	0	0	
ferieperiodeVinterferie	0	0	
total_income	0	0	
total_expense	0	0	
net_balance	0	0	
antal_transfers	0	0	
	udehold_rangB	udehold_rangC	udehold_rangD
tidsperiodeLørdag_eftermiddag	0	0	0
tidsperiodeMandag_formiddag	0	0	0
tidsperiodeOnsdag_aften	0	0	0
tidsperiodeSøndag_formiddag	0	0	0
tidsperiodeTordag_eftermiddag	0	0	0
ferieperiodeIngen	0	0	0
ferieperiodeSommerferie	0	0	0
ferieperiodeVinterferie	0	0	0
total_income	0	0	0
total_expense	0	0	0
net_balance	0	0	0
antal_transfers	0	0	0
	sæson2004/2005	sæson2005/2006	sæson2006/2007
tidsperiodeLørdag_eftermiddag	0	0	0
tidsperiodeMandag_formiddag	0	0	0
tidsperiodeOnsdag_aften	0	0	0
tidsperiodeSøndag_formiddag	0	0	0
tidsperiodeTordag_eftermiddag	0	0	0
ferieperiodeIngen	0	0	0
ferieperiodeSommerferie	0	0	0
ferieperiodeVinterferie	0	0	0

	-8e+05	320000	-8e+05
total_income	-8e+05	320000	-8e+05
total_expense	-120000	280000	430000
net_balance	-680000	40000	-1230000
antal_transfers	0	1	0
	sæson2007/2008	sæson2013/2014	sæson2015/2016
tidsperiodeLørdag_eftermiddag	0	0	0
tidsperiodeMandag_formiddag	0	0	0
tidsperiodeOnsdag_aften	0	0	0
tidsperiodeSøndag_formiddag	0	0	0
tidsperiodeTordag_eftermiddag	0	0	0
ferieperiodeIngen	0	0	0
ferieperiodeSommerferie	0	0	0
ferieperiodeVinterferie	0	0	0
total_income	-130000	-8e+05	-8e+05
total_expense	-120000	-50000	-120000
net_balance	-10000	-750000	-680000
antal_transfers	3	-1	-4
	sæson2016/2017	sæson2021/2022	sæson2022/2023
tidsperiodeLørdag_eftermiddag	0	0	0
tidsperiodeMandag_formiddag	0	0	0
tidsperiodeOnsdag_aften	0	0	0
tidsperiodeSøndag_formiddag	0	0	0
tidsperiodeTordag_eftermiddag	0	0	0
ferieperiodeIngen	0	0	0
ferieperiodeSommerferie	0	0	0
ferieperiodeVinterferie	0	0	0
total_income	-4e+05	141000	4e+06
total_expense	-20000	-120000	1652000
net_balance	-380000	261000	2348000
antal_transfers	5	1	4
	sæson2023/2024	sæson2024/2025	sæson2025/2026
tidsperiodeLørdag_eftermiddag	0	0	0
tidsperiodeMandag_formiddag	0	0	0
tidsperiodeOnsdag_aften	0	0	0
tidsperiodeSøndag_formiddag	0	0	0
tidsperiodeTordag_eftermiddag	0	0	0
ferieperiodeIngen	0	0	0
ferieperiodeSommerferie	0	0	0
ferieperiodeVinterferie	0	0	0
total_income	6850000	1697000	9685000
total_expense	1939000	1301000	1389000
net_balance	4911000	396000	8296000
antal_transfers	0	-1	-1
runde	tidsperiodeLørdag_aften		
tidsperiodeLørdag_eftermiddag	0	-1	
tidsperiodeMandag_formiddag	0	0	
tidsperiodeOnsdag_aften	0	0	
tidsperiodeSøndag_formiddag	0	0	
tidsperiodeTordag_eftermiddag	0	0	
ferieperiodeIngen	0	0	
ferieperiodeSommerferie	0	0	
ferieperiodeVinterferie	0	0	
total_income	0	0	
total_expense	0	0	
net_balance	0	0	
antal_transfers	0	0	

	tidsperiodeMandag_aften	
tidsperiodeLørdag_eftermiddag	0	
tidsperiodeMandag_formiddag	-1	
tidsperiodeOnsdag_aften	0	
tidsperiodeSøndag_formiddag	0	
tidsperiodeTordag_eftermiddag	0	
ferieperiodeIngen	0	
ferieperiodeSommerferie	0	
ferieperiodeVinterferie	0	
total_income	0	
total_expense	0	
net_balance	0	
antal_transfers	0	
	tidsperiodeMandag_eftermiddag	
tidsperiodeLørdag_eftermiddag	0	
tidsperiodeMandag_formiddag	-1	
tidsperiodeOnsdag_aften	0	
tidsperiodeSøndag_formiddag	0	
tidsperiodeTordag_eftermiddag	0	
ferieperiodeIngen	0	
ferieperiodeSommerferie	0	
ferieperiodeVinterferie	0	
total_income	0	
total_expense	0	
net_balance	0	
antal_transfers	0	
	tidsperiodeSøndag_aften	
tidsperiodeLørdag_eftermiddag	0	
tidsperiodeMandag_formiddag	0	
tidsperiodeOnsdag_aften	0	
tidsperiodeSøndag_formiddag	-1	
tidsperiodeTordag_eftermiddag	0	
ferieperiodeIngen	0	
ferieperiodeSommerferie	0	
ferieperiodeVinterferie	0	
total_income	0	
total_expense	0	
net_balance	0	
antal_transfers	0	
	tidsperiodeSøndag_eftermiddag	
tidsperiodeLørdag_eftermiddag	0	
tidsperiodeMandag_formiddag	0	
tidsperiodeOnsdag_aften	0	
tidsperiodeSøndag_formiddag	-1	
tidsperiodeTordag_eftermiddag	0	
ferieperiodeIngen	0	
ferieperiodeSommerferie	0	
ferieperiodeVinterferie	0	
total_income	0	
total_expense	0	
net_balance	0	
antal_transfers	0	
	tidsperiodeTordag_aften Er_helligdag1	
tidsperiodeLørdag_eftermiddag	0	0
tidsperiodeMandag_formiddag	0	0
tidsperiodeOnsdag_aften	0	0

tidsperiodeSøndag_formiddag	0	0
tidsperiodeTordag_eftermiddag	-1	0
ferieperiodeIngen	0	0
ferieperiodeSommerferie	0	0
ferieperiodeVinterferie	0	0
total_income	0	0
total_expense	0	0
net_balance	0	0
antal_transfers	0	0
	i_sommerferie1	i_efterårsferie1
tidsperiodeLørdag_eftermiddag	0	0
tidsperiodeMandag_formiddag	0	0
tidsperiodeOnsdag_aften	0	0
tidsperiodeSøndag_formiddag	0	0
tidsperiodeTordag_eftermiddag	0	0
ferieperiodeIngen	-1	-1
ferieperiodeSommerferie	1	0
ferieperiodeVinterferie	0	1
total_income	0	0
total_expense	0	0
net_balance	0	0
antal_transfers	0	0
	d3_tilskuere	stadion_kapacitet
tidsperiodeLørdag_eftermiddag	0	0
tidsperiodeMandag_formiddag	0	0
tidsperiodeOnsdag_aften	0	0
tidsperiodeSøndag_formiddag	0	0
tidsperiodeTordag_eftermiddag	0	0
ferieperiodeIngen	0	0
ferieperiodeSommerferie	0	0
ferieperiodeVinterferie	0	0
total_income	0	0
total_expense	0	0
net_balance	0	0
antal_transfers	0	0

Designmatrix-rank (bekræftelse af multikollinearitet problemet)

```
kampdag_X <- model.matrix(stor_lm_model)
qr(kampdag_X)$rank
```

[1] 45

```
ncol(kampdag_X)
```

[1] 57

Vi har 57 kolonner, 45 kolonner er ikke lineært afhængige af andre kolonner. Det betyder, at 12 kolonner er linerært afhængige af andre kolonner.

I samlet_variabler har vi vores rå data, her har vi 43 kolonner, som repræsenterer vores 43 variabler. De 57 kolonner her antallet af kolonner i designmatricen efter vi har kørt dummy-kodningen. De numeriske variabler giver 1 kolonne. Faktorvariablerne bliver omdannet til dummies, og giver k - 1, hvor k er antallet af dummykolonner.

Vores model lider af multikollinearitet, fordi flere af vores variabler beskriver det samme fænomen. Det giver perfekt lineær afhængighed mellem variablerne (dette kaldes også aliasing). Nogle af de variabler der forårsager

den perfekt multikollinearitet er bl.a. følgende: ugedag / tidsperiode (tidsperiode indeholder informationer om ugedag) Ferievariablerne (ferieperioden gentages flere gange, de er derfor perfekt afhængige af hinanden) Transfervariablerne (balance = indkomst - omkostninger, dvs perfekt afhængighed) Lagged tilskueretal (d3,d7,d10). Vi inkluderer dem alle og lager dem, det er lagged versioner af samme variabel, og de er derfor meget stærkt korrelerede. Dette giver perfekt multikollinearitet. Det vi gør nu, er at vi beholder en af disse variabler, så vi ikke beskriver det samme igen og igen.

.3.2 Stor model for 3 dage før - lineær regression med alle mulige variabler

```
stor_lm_model_ny <- lm(
  tilskuere ~
    temp_dry + wind_speed + nedbør_seneste_7_timer +
    tidsperiode +
    placering_lag_hjemme + placering_lag_udet +
    mål_seneste_3_hjemme +
    mål_sæson_lag_udehold +
    form_seneste_3_hjemmehold + form_seneste_3_udehold + d3_tilskuere +
    udehold_rang +
    sæson + runde +
    ferieperiode +
    net_balance +
    stadion_kapacitet,
  data = train_data
)
```

Vi tester igen for multikollinearitet - VIF (Variance inflation factor) via car-pakken Model.matrix udtrækker desginmatricen fra modellen qr() laver en QR-dekomposition af designmatricen \$rank er antallet af lineært uafhængige kolonner i X (som vi har defineret som modelmatricen) ncol tæller antallet af regressorer i modellen, dvs det antal af kolonner, som ikke er lineært afhængige af hinanden. alias() identificerer hvilke variabler der er 100% lineære afhængige af hinanden.

```
X <- model.matrix(stor_lm_model_ny)
qr(X)$rank
```

```
[1] 42
```

```
ncol(X)
```

```
[1] 43
```

```
alias(stor_lm_model_ny)
```

```
Model :
tilskuere ~ temp_dry + wind_speed + nedbør_seneste_7_timer +
  tidsperiode + placering_lag_hjemme + placering_lag_udet +
  mål_seneste_3_hjemme + mål_sæson_lag_udehold + form_seneste_3_hjemmehold +
  form_seneste_3_udehold + d3_tilskuere + udehold_rang + sæson +
  runde + ferieperiode + net_balance + stadion_kapacitet
```

```
Complete :
(Intercept) temp_dry wind_speed nedbør_seneste_7_timer
```

```

net_balance 680000      0      0      0
            tidsperiodeLørdag_aften tidsperiodeLørdag_eftermiddag
net_balance 0          0
            tidsperiodeMandag_aften tidsperiodeMandag_eftermiddag
net_balance 0          0
            tidsperiodeMandag_formiddag tidsperiodeOnsdag_aften
net_balance 0          0
            tidsperiodeSøndag_aften tidsperiodeSøndag_eftermiddag
net_balance 0          0
            tidsperiodeSøndag_formiddag tidsperiodeTordag_aften
net_balance 0          0
            tidsperiodeTordag_eftermiddag placering_lag_hjemme
net_balance 0          0
            placering_lag_ude mål_seneste_3_hjemme mål_sæson_lag_udhold
net_balance 0          0          0
            form_seneste_3_hjemmehold form_seneste_3_udhold d3_tilskuere
net_balance 0          0          0
            udehold_rangB udehold_rangC udehold_rangD sæson2004/2005
net_balance 0          0          0      -680000
            sæson2005/2006 sæson2006/2007 sæson2007/2008 sæson2013/2014
net_balance 40000      -1230000     -10000      -750000
            sæson2015/2016 sæson2016/2017 sæson2021/2022 sæson2022/2023
net_balance -680000     -380000      261000      2348000
            sæson2023/2024 sæson2024/2025 sæson2025/2026 runde
net_balance 4911000     396000      8296000      0
            ferieperiodeIngen ferieperiodeSommerferie ferieperiodeWinterferie
net_balance 0          0          0
            stadion_kapacitet
net_balance 0

```

Det kan ses, at net_balance skyldes fejlen. Den fjerner vi nu

```

stor_lm_model_ny <- lm(
  tilskuere ~
    temp_dry + wind_speed + nedbør_seneste_7_timer +
    tidsperiode +
    placering_lag_hjemme + placering_lag_ude +
    mål_seneste_3_hjemme +
    mål_sæson_lag_udhold +
    form_seneste_3_hjemmehold + form_seneste_3_udhold +
    d3_tilskuere +
    udehold_rang +
    sæson + runde +
    ferieperiode +
    stadion_kapacitet,
  data = train_data
)

```

Vi tester igen for multikollinearitet - VIF (Variance inflation factor) via car-pakken Model.matrix udtrækker desginmatricen fra modellen qr() laver en QR-dekomposition af designmatricen \$rank er antallet af lineært uafhængige kolonner i X (som vi har defineret som modelmatricen) ncol tæller antallet af regressorer i modellen, dvs det antal af kolonner, som ikke er lineært afhængige af hinanden. alias() identificerer hvilke variabler der er 100% lineære afhængige af hinanden.

```
X <- model.matrix(stor_lm_model_ny)
qr(X)$rank
```

[1] 42

```
ncol(X)
```

[1] 42

Nu er der 42/42, så ingen lineære afhængighed.

```
alias(stor_lm_model_ny)
```

```
Model :
tilskuere ~ temp_dry + wind_speed + nedbør_seneste_7_timer +
    tidsperiode + placering_lag_hjemme + placering_lag_udet +
    mål_seneste_3_hjemme + mål_sæson_lag_udet + form_seneste_3_hjemmehold +
    form_seneste_3_udet + d3_tilskuere + udet_rang + sæson +
    runde + ferieperiode + stadion_kapacitet
```

Vi tester nu for VIF igen.

```
vif(stor_lm_model_ny)
```

	GVIF	Df	GVIF^(1/(2*Df))
temp_dry	1,764200	1	1,328232
wind_speed	1,493931	1	1,222265
nedbør_seneste_7_timer	1,458720	1	1,207775
tidsperiode	25,418330	11	1,158431
placering_lag_hjemme	5,041781	1	2,245391
placering_lag_udet	4,145643	1	2,036085
mål_seneste_3_hjemme	2,153552	1	1,467499
mål_sæson_lag_udet	13,691301	1	3,700176
form_seneste_3_hjemmehold	1,954839	1	1,398156
form_seneste_3_udet	2,210132	1	1,486651
d3_tilskuere	3,049122	1	1,746174
udet_rang	5,045422	3	1,309633
sæson	103,383835	12	1,213209
runde	10,939411	1	3,307478
ferieperiode	3,605810	3	1,238323
stadion_kapacitet	2,450885	1	1,565530

Det kan ses, at ingen af variablerne har et kritisk VIF-niveau (over 5). De fleste ligger på et moderat niveau mellem 1 og 5. Vi skal dog være OBS på, at mål_sæson_lag_udet og runde nærmere sig et højt niveau. Vi har allerede en tårnhøj R^2, så vi prøver lige at fjerne mål_sæson_lag_udet.

```
stor_lm_model_3_dage <- lm(
  tilskuere ~
    temp_dry + wind_speed + nedbør_seneste_7_timer +
    tidsperiode +
    placering_lag_hjemme + placering_lag_udet +
```

```

mål_seneste_3_hjemme +
form_seneste_3_hjemmehold + form_seneste_3_udehold +
d3_tilskuere +
udehold_rang +
sæson + runde +
ferieperiode +
stadion_kapacitet,
data = train_data
)

```

Vi tester nu for VIF igen.

```
vif(stor_lm_model_3_dage)
```

	GVIF	Df	GVIF^(1/(2*Df))
temp_dry	1,741086	1	1,319502
wind_speed	1,492148	1	1,221535
nedbør_seneste_7_timer	1,422966	1	1,192882
tidsperiode	22,467320	11	1,151951
placering_lag_hjemme	4,920812	1	2,218290
placering_lag_udé	2,530418	1	1,590729
mål_seneste_3_hjemme	2,153361	1	1,467433
form_seneste_3_hjemmehold	1,914527	1	1,383664
form_seneste_3_udehold	2,207073	1	1,485622
d3_tilskuere	3,045546	1	1,745149
udehold_rang	5,009718	3	1,308084
sæson	92,380421	12	1,207533
runde	2,278849	1	1,509586
ferieperiode	3,533438	3	1,234145
stadion_kapacitet	2,408279	1	1,551863

.3.3 Stor model for 7 dage før - linenær regression med alle mulige variabler

Model 7 dage før

```

stor_lm_model_7_dage <- lm(
tilskuere ~
  temp_dry + wind_speed + nedbør_seneste_7_timer +
tidsperiode +
  placering_lag_hjemme + placering_lag_udé +
mål_seneste_3_hjemme +
  form_seneste_3_hjemmehold + form_seneste_3_udehold +
d7_tilskuere +
udehold_rang +
sæson + runde +
ferieperiode +
  stadion_kapacitet,
data = train_data
)

```

.3.4 Stor model for 10 dage før - linenær regression med alle mulige variabler

Model 10 dage før

```

stor_lm_model_10_dage <- lm(
  tilskuere ~
    temp_dry + wind_speed + nedbør_seneste_7_timer +
    tidsperiode +
    placering_lag_hjemme + placering_lag_udet +
    mål_seneste_3_hjemme +
    form_seneste_3_hjemmehold + form_seneste_3_udehold +
    d10_tilskuere +
    udehold_rang +
    sæson + runde +
    ferieperiode +
    stadion_kapacitet,
  data = train_data
)

```

.3.5 Stor model for flere måneder før - lineær regression med alle mulige variabler

```

stor_lm_model_flere_måneder <- lm(
  tilskuere ~ sæson + runde + tidsperiode
  + ferieperiode + stadion_kapacitet,
  data = train_data
)

```

Her beregner vi VIF på modellen der kan forudsige flere måneder i forvejen

```
vif(stor_lm_model_flere_måneder)
```

	GVIF	Df	GVIF^(1/(2*Df))
sæson	4,989881	12	1,069269
runde	1,707010	1	1,306526
tidsperiode	7,142289	11	1,093480
ferieperiode	2,103220	3	1,131916
stadion_kapacitet	1,391654	1	1,179684

.3.6 Samlet for lineære modeller

Predict på testdata Vi bruger predict til at tage koefficienter fra den estimerede model, (som vi har defineret foroven).

```

tilskuere_hat3 <- predict(stor_lm_model_3_dage, newdata = test_data)
tilskuere_hat7 <- predict(stor_lm_model_7_dage, newdata = test_data)
tilskuere_hat10 <- predict(stor_lm_model_10_dage, newdata = test_data)
tilskuere_hat_flere_måneder <- predict(stor_lm_model_flere_måneder, newdata = test_data)

```

Funktion til beregning af RMSE og MSE Vi opretter en funktion, som vi kalder "calc_metrics". obs er de observerede værdier (dvs faktiske tilskuerantal) pred er forudsagtes værdier fra vores model Efter det skriver vi formlerne til at beregne hhv. MSE, RMSE og R^2. Til sidst returnerer vi en liste med alle tre metrics.

```

calc_metrics <- function(obs, pred) {
  mse <- mean((obs - pred)^2)
  rmse <- sqrt(mse)
}

```

```

ss_res <- sum((obs - pred)^2)
ss_tot <- sum((obs - mean(obs))^2)
r2 <- 1 - ss_res/ss_tot
return(list(MSE = mse, RMSE = rmse, R2 = r2))
}

```

Vi definer vores modeller, og hvad der skal beregnes

```

metrics_3_dage    <- calc_metrics(test_data$tilskuere, tilskuere_hat3)
metrics_7_dage    <- calc_metrics(test_data$tilskuere, tilskuere_hat7)
metrics_10_dage   <- calc_metrics(test_data$tilskuere, tilskuere_hat10)
metrics_flere_måneder <- calc_metrics(test_data$tilskuere, tilskuere_hat_flere_måneder)

```

Vi ser resultateterne for vores lineære regressionsmodeller

```
metrics_3_dage
```

```
$MSE
[1] 45326,65

$RMSE
[1] 212,9006

$R2
[1] 0,9823316
```

```
metrics_7_dage
```

```
$MSE
[1] 182685,7

$RMSE
[1] 427,4174

$R2
[1] 0,9287889
```

```
metrics_10_dage
```

```
$MSE
[1] 267665,8

$RMSE
[1] 517,3643

$R2
[1] 0,8956636
```

```
metrics_flere_måneder
```

```
$MSE
[1] 1595157
```

```
$RMSE  
[1] 1262,995
```

```
$R2  
[1] 0,3782062
```

Vi samler resultaterne i en tibble. Vi smider de i en tibble, og bruger c (combine) til kombiner de forskellige modeller for de forskellige perioder, og nøgletalene for hver model.

```
metrics_samlet <- tibble(  
  Model = c("3 dage før", "7 dage før", "10 dage før", "flere måneder"),  
  MSE = c(metrics_3_dage$MSE, metrics_7_dage$MSE, metrics_10_dage$MSE, metrics_flere_måneder$MSE),  
  RMSE = c(metrics_3_dage$RMSE, metrics_7_dage$RMSE, metrics_10_dage$RMSE, metrics_flere_måneder$RMSE),  
  R2 = c(metrics_3_dage$R2, metrics_7_dage$R2, metrics_10_dage$R2, metrics_flere_måneder$R2)  
)
```

Vi laver et ggplot til at se residualerne, dvs hvor meget prikkerne afviger fra linjen og hvor præcis modellen er.

Vi starter med at samle dataene til plottet. Vi bruger bind_rows til at sætter rækkerne fra de forskellige tabeller sammen. Vi har vores model, vores observeret værdi og vores forudsagte (predicted) Værdi.

```
plot_data <- bind_rows(  
  tibble(  
    Model = "3 dage før",  
    Observed = test_data$tilskuere,  
    Predicted = tilskuere_hat3  
)  
,  
  tibble(  
    Model = "7 dage før",  
    Observed = test_data$tilskuere,  
    Predicted = tilskuere_hat7  
)  
,  
  tibble(  
    Model = "10 dage før",  
    Observed = test_data$tilskuere,  
    Predicted = tilskuere_hat10  
)  
,  
  tibble(  
    Model = "Flere måneder før",  
    Observed = test_data$tilskuere,  
    Predicted = tilskuere_hat_flere_måneder  
)  
)
```

Vi laver vores ggplot ud fra vores plot_data. x er = observed og y = predicted geom_point() tegner et punkt for hver observation. alpha = 0.4 gør punkterne halvgenemsigtige geom_abline tegner en 45 graders linje. Hvor vi definerer at den skal være stiplet eller "dashed", og at farven skal være "rød". Vi bruger facet_wrap til at lave separate plots for hver model ~ betyder, at hver model får sit eget plit scales = "free" betydet, at hver plt kan have sin egen skal på x- og y-aksen, så alle punkterne passer. Vi bruger labs til at lave overskrifterne.

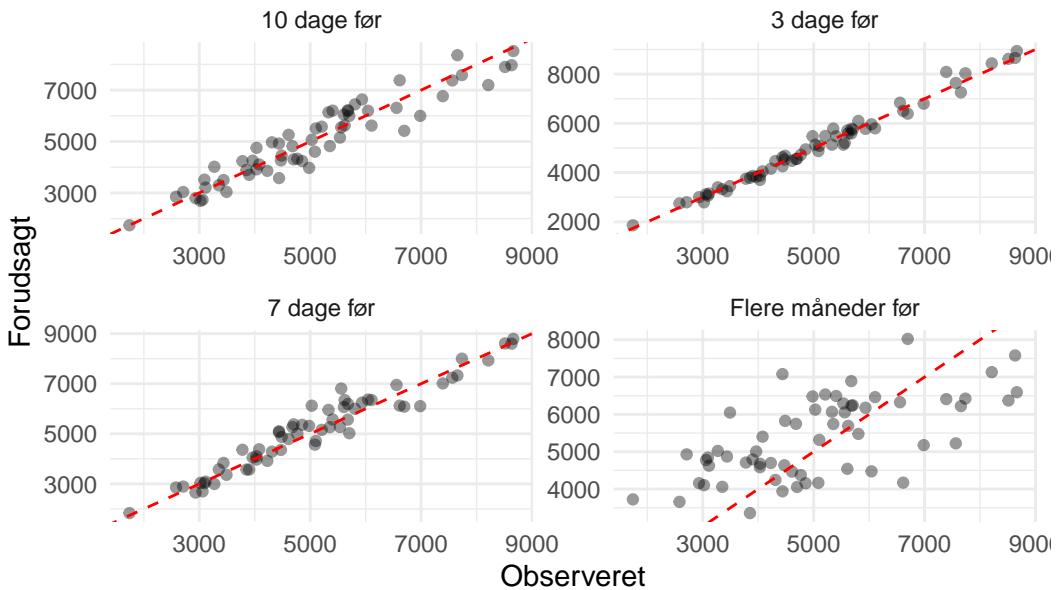
```
ggplot(plot_data, aes(x = Observed, y = Predicted)) +  
  geom_point(alpha = 0.4) +  
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +  
  facet_wrap(~ Model, scales = "free") +
```

```

  labs(
    title = "Observeret vs. forudsagt tilskuertal",
    x = "Observeret",
    y = "Forudsagt"
  ) +
  theme_minimal()

```

Observeret vs. forudsagt tilskuertal



Jo tættere på kampdagen vi er, jo mindre bliver residualerne. Dette betyder, at modellen bliver mere og mere præcis, jo tættere på kampdagen vi kommer. Årsagen til denne udvikling, er at vi har flere variabler jo tættere på kampdagen vi er. Dette betyder, at vores model kan forklare mere af variationen, hvilket bekræftes ved at R^2 stiger jo tættere på kampdagen vi kommer.

.3.7 Lasso / Ridge regression (shrinkage-metoder)

Ridge og lasso regression

Lambda-grid Vi opretter 100 lamda-værdier fra 10^{-5} til 10^{-3} Vi bruger det til regularisering i shrinkage metoder (Ridge og Lasso). Jo større lamda-værdien er, jo større er straffen. En lille lamda-værdi, er næsten det samme som almindelig lineær regression. Vi vurder 100 lamda-værdier, til at være tilstrækkeligt.

```
lambda_grid <- 10^seq(5, -3, length = 100)
```

Funktion til metrics Vi opretter en funktion, som vi kalder "calc_metrics". Obs er de observerede værdier (dvs faktiske tilskuertal) pred er forudsagtes værdier fra vores model Efter det skriver vi formlerne til at beregne hhv. MSE, RMSE og R^2 . Til sidst returnerer vi en liste med alle tre metrics.

```

calc_metrics <- function(obs, pred) {
  mse <- mean((obs - pred)^2)
  rmse <- sqrt(mse)
  ss_res <- sum((obs - pred)^2)
  ss_tot <- sum((obs - mean(obs))^2)
  r2 <- 1 - ss_res / ss_tot
}

```

```

  list(MSE = mse, RMSE = rmse, R2 = r2)
}

```

Funktion til at lave metrics tibble Vi pakker vores resultater i en tibble med vores værdier, og hvilke modeltype det er. Formålet her, at at gøre resultaterne lettere at sammenligne og samle.

```

make_metrics_tibble <- function(obs, pred, model_name, lambda, periode) {
  metrics <- calc_metrics(obs, pred)
  tibble(
    Model = model_name,
    Lambda = lambda,
    MSE = metrics$MSE,
    RMSE = metrics$RMSE,
    R2 = metrics$R2,
    Periode = periode
  )
}

```

Base features Vi definerer de fælles forklarende variabler for de fleste modeller. Vi bruger combine (c) til at sammensætte de forskellige variabler.

```

base_features <- c(
  "temp_dry", "wind_speed", "nedbør_seneste_7_timer",
  "ugedag",
  "placering_lag_hjemme", "placering_lag_udet",
  "form_seneste_3_hjemmekampe", "form_seneste_3_udet",
  "udehold_rang",
  "sæson", "runde", "tidsperiode",
  "ferieperiode",
  "antal_transfers",
  "stadion_kapacitet"
)

```

Features for hvert tidspunkt Vi laver fire modeller med forskellige tidshorisonter. Vi bruger combine (c) til at kombinerer base_features, som vi defineret foroven, samt billetsalg varabel og mål i seneste 3 hjemmekampe. I "flere måneder"-modellen definerer hvilke variabler der skal med via combine (c).

```

features_list <- list(
  "3 dage" = c(base_features, "d3_tilskuere", "mål_seneste_3_hjemme"),
  "7 dage" = c(base_features, "d7_tilskuere", "mål_seneste_3_hjemme"),
  "10 dage" = c(base_features, "d10_tilskuere", "mål_seneste_3_hjemme"),
  "flere måneder" = c(
    "ugedag", "sæson", "runde", "tidsperiode",
    "ferieperiode", "antal_transfers", "stadion_kapacitet"
  )
)

```

Vi kører en Ridge-funktion, der vælger den bedste lambda. Vi kører den for vores forskellige features for alle modeller, og den periode som modellerne er i. Vi laver designmatricer (defineret som x_train og x_test) og sikrer, at vores test-matrice har samme kolonner som train-matricen. Vi træner vores Ridge-model til af alpha = 0 for alle lambda-værdier i lambda-grind. Vi beregner vores nøgletal (MSE, RMSE og R²) på testdata for hver lambda. Vores Ridge-regressionsmodel returnerer den lambda og de metrics, som giver den laveste MSE for den valgte periode.

```

run_ridge_best_lambda <- function(features, periode) {

  formula <- as.formula(paste("tilskuere ~", paste(features, collapse = "+")))

  x_train <- model.matrix(formula, train_data)[, -1]
  y_train <- train_data$tilskuere

  x_test_raw <- model.matrix(formula, test_data)[, -1]

  x_test <- matrix(0, nrow = nrow(x_test_raw), ncol = ncol(x_train))
  colnames(x_test) <- colnames(x_train)
  common_cols <- intersect(colnames(x_test_raw), colnames(x_train))
  x_test[, common_cols] <- x_test_raw[, common_cols]

  ridge_mod <- glmnet(x_train, y_train, alpha = 0, lambda = lambda_grid)

  metrics <- bind_rows(lapply(seq_along(ridge_mod$lambda), function(i) {
    pred <- predict(ridge_mod, s = ridge_mod$lambda[i], newx = x_test)
    make_metrics_tibble(
      obs = test_data$tilskuere,
      pred = pred,
      model = "Ridge",
      lambda = ridge_mod$lambda[i],
      periode = periode
    )
  }))
  metrics[which.min(metrics$MSE), ]
}

```

Vi kører en Lasso-funktion, der vælger den bedste lambda. Vi kører den for vores forskellige features for alle modeller, og den periode som modellerne er i. Vi laver designmatricer (defineret som x_train og x_test) og sikrer, at vores test-matrice har samme kolonner som train-matricen. Vi træner vores Ridge-model til af alpha = 0 for alle lambda-værdier i lambda-grid. Vi beregner vores nøgletal (MSE, RMSE og R²) på testdata for hver lambda. Vores Ridge-regressionsmodel returnerer den lambda og de metrics, som giver den laveste MSE for den valgte periode.

```

run_lasso_best_lambda <- function(features, periode) {

  formula <- as.formula(paste("tilskuere ~", paste(features, collapse = "+")))

  x_train <- model.matrix(formula, train_data)[, -1]
  y_train <- train_data$tilskuere

  x_test_raw <- model.matrix(formula, test_data)[, -1]

  x_test <- matrix(0, nrow = nrow(x_test_raw), ncol = ncol(x_train))
  colnames(x_test) <- colnames(x_train)
  common_cols <- intersect(colnames(x_test_raw), colnames(x_train))
  x_test[, common_cols] <- x_test_raw[, common_cols]

  lasso_mod <- glmnet(x_train, y_train, alpha = 1, lambda = lambda_grid)

  metrics <- bind_rows(lapply(seq_along(lasso_mod$lambda), function(i) {
    pred <- predict(lasso_mod, s = lasso_mod$lambda[i], newx = x_test)
  }))

```

```

make_metrics_tibble(
  obs = test_data$tilskuere,
  pred = pred,
  model = "Lasso",
  lambda = lasso_mod$lambda[i],
  periode = periode
)
}))
```

metrics[which.min(metrics\$MSE),]

}

Vi kører nu alle periodemodellerne for Ridge- og Lasso- regression, og samler resulterne i en tabel.

Vi laver et loop over alle perioder i features_listen, og kører run_ridge_best_lambda. Og vi bruger bind_rows funktioenn til at binde resultaterne i en tibble med den bedste lambda og de bedste metrics for hver periode.

```

metrics_ridge_best <- bind_rows(
  lapply(names(features_list), function(p) {
    run_ridge_best_lambda(features_list[[p]], p)
  })
)
```

Vi laver et loop over alle perioder i features_listen, og kører run_lasso_best_lambda. Og vi bruger bind_rows funktioenn til at binde resultaterne i en tibble med den bedste lambda og de bedste metrics for hver periode.

```

metrics_lasso_best <- bind_rows(
  lapply(names(features_list), function(p) {
    run_lasso_best_lambda(features_list[[p]], p)
  })
)
```

Vi ser resultaterne

```
#view(metrics_ridge_best)
```

.3.8 Best subset selection

Metrics-funktion Funktion til metrics Vi opretter en funktion, som vi kalder “calc_metrics”. Obs er de observeerde værdier (dvs faktiske tilskuertal) pred er forudsagtes værdier fra vores model Efter det skriver vi formlerne til at beregne hhv. MSE, RMSE og R². Til sidst returnerer vi en liste med alle tre metrics.

```

calc_metrics <- function(obs, pred) {
  mse <- mean((obs - pred)^2)
  rmse <- sqrt(mse)
  ss_res <- sum((obs - pred)^2)
  ss_tot <- sum((obs - mean(obs))^2)
  r2 <- 1 - ss_res / ss_tot
  list(MSE = mse, RMSE = rmse, R2 = r2)
}
```

Alignment-funktion Vi sørger for, at vores designmatrix (newx) får samme kolonner og rækkefælge som vores reference-matrix (refx). Vi opretter et tom matrix med det samme antal kolonner og kolonneanavne som refx.

Vi finder kolonner, der findes i både designmatrixen og reference-matrixen. Vi kopier værdier fra newx til de matchende kolonner til refx. Funktioenn returnerer en matrix, hvor alle kolonner fra refx er til stede, og evt manglende kolonner fra newx fyldes med "0". Formålet med dette er at sikre, at train- og test- matricen har samme struktur, hvilket er nødtvendigt for at glmnet-pakken kan lave forudsigelser.

```


```

Base features Vi definer de fælles forklarende variabler for de fleste modeller. Vi bruger combine (c) til at sammensætte de forskellige variabler.

```
base_features <- c(
  "temp_dry", "wind_speed", "nedbør_seneste_7_timer",
  "ugedag",
  "placering_lag_hjemme", "placering_lag_udet",
  "form_seneste_3_hjemmekampe", "form_seneste_3_udehold",
  "udehold_rang",
  "sæson", "runde", "tidsperiode",
  "ferieperiode",
  "antal_transfers",
  "stadion_kapacitet"
)
```

Features for hvert periode Vi laver fire modeller med forskellige tidshorisonter. Vi bruger combine (c) til at kombinerer base_features, som vi defineret foroven, samt billetsalg varabel og mål i seneste 3 hjemmekampe. I "flere måneder"-modellen definerer hvilke variabler der skal med via combine (c).

```
features_list <- list(
  "3 dage" = c(base_features, "d3_tilskuere", "mål_seneste_3_hjemme"),
  "7 dage" = c(base_features, "d7_tilskuere", "mål_seneste_3_hjemme"),
  "10 dage" = c(base_features, "d10_tilskuere", "mål_seneste_3_hjemme"),
  "flere måneder" = c(
    "ugedag", "sæson", "runde", "tidsperiode",
    "ferieperiode", "antal_transfers", "stadion_kapacitet"
  )
)
```

Best Subset funktion Vi finder den model ved best subset regression. Vi udvælger outcome og features fra train- og testdata. Vi opretter designmatricer uden intercept og fjerner perfekt multikollinearitet, det sker ved qrX og X_train + X_test. Vi tilpasser test-matrix, så den har samme kolonner som train. Vi tester alle kombinationer af op til max_vars variable (exhaustive search). Vi beregner MSE, RMSE og R² på testdata for hver model. Til sidst eturnerer koden den model, der giver lavest RMSE, med antal variable og performance-metrics.

```
run_best_subset <- function(features, tidspunkt, max_vars = 20, outcome_var = "tilskuere") {

  train_sub <- train_data |> select(all_of(c(outcome_var, features)))
  test_sub <- test_data |> select(all_of(c(outcome_var, features)))

  X_train_raw <- model.matrix(as.formula(paste(outcome_var, "~ . -1"))), train_sub)
```

```

X_test_raw <- model.matrix(as.formula(paste(outcome_var, "~ . -1")), test_sub)

y_train <- train_sub[[outcome_var]]
y_test <- test_sub[[outcome_var]]

qrX <- qr(X_train_raw)
X_train <- X_train_raw[, qrX$pivot[1:qrX$rank], drop = FALSE]
X_test <- align_matrix(X_test_raw, X_train)

regfit <- regsubsets(x = X_train, y = y_train, nvmax = max_vars, method = "exhaustive", really.big = TRUE)

metrics_list <- lapply(1:max_vars, function(k) {
  beta <- coef(regfit, id = k)
  Xk <- X_test[, names(beta)[-1], drop = FALSE]
  pred <- beta[1] + Xk %*% beta[-1]
  m <- calc_metrics(y_test, pred)
  tibble(
    Model = "Best Subset",
    Tidspunkt = tidspunkt,
    Antal_variabler = k,
    MSE = m$MSE,
    RMSE = m$RMSE,
    R2 = m$R2
  )
})
metrics_df <- bind_rows(metrics_list)

best_row <- metrics_df[which.min(metrics_df$RMSE), ]
return(best_row)
}

```

Vi kører Best Subset Method for alle perioder Vi looper over alle tidspunkter/perioder i features_list. Vi kører run_best_subset for hver periode med de tilhørende features. Til sidst binder vi alle resultater sammen i en tibble via bind_rows.

```

results_subset <- bind_rows(
  lapply(names(features_list), function(tidspunkt) {
    run_best_subset(features_list[[tidspunkt]], tidspunkt)
  })
)

```

Warning in leaps.setup(x, y, wt = weights, nbest = nbest, nvmax = nvmax, : 1
linear dependencies found

Reordering variables and trying again:

Warning in leaps.setup(x, y, wt = weights, nbest = nbest, nvmax = nvmax, : 1
linear dependencies found

Reordering variables and trying again:

Warning in leaps.setup(x, y, wt = weights, nbest = nbest, nvmax = nvmax, : 1
linear dependencies found

Reordering variables and trying again:

```
Warning in leaps.setup(x, y, wt = weights, nbest = nbest, nvmax = nvmax, : 1
linear dependencies found
```

Reordering variables and trying again:

.3.9 Sammenligning af nøgletal

Samling af de forskellige modeller i en tibble

Hjælpefunktion Vi opretter en tibble med kolonnerne over vores værdier (MSE, RMSE og R²) for alle modeller.

```
metrics_to_tibble <- function(metrics, model_name) {
  tibble(
    Model = model_name,
    MSE = metrics$MSE,
    RMSE = metrics$RMSE,
    R2 = metrics$R2
  )
}
```

Lineær regression Vi bruger bind_rows til at danne metrics for hver lm model til en tibble

```
metrics_lm <- bind_rows(
  metrics_to_tibble(metrics_3_dage, "3 dage før"),
  metrics_to_tibble(metrics_7_dage, "7 dage før"),
  metrics_to_tibble(metrics_10_dage, "10 dage før"),
  metrics_to_tibble(metrics_flere_måneder, "Flere måneder")
)
```

Ridge regression Vi bruger transmute til at lave en ny tibble med kun de valgte kolonner. Vi bruger paste("Ridge -") til at tilføje teksten "Ridge" foran perioden. Vi beholder MSE, RMSE og R² som performance-metrics.

```
metrics_ridge <- metrics_ridge_best |>
  transmute(
    Model = paste("Ridge -", Periode),
    MSE,
    RMSE,
    R2
  )
```

Lasso regression Vi bruger transmute til at lave en ny tibble med kun de valgte kolonner. Vi bruger paste("Lasso -") til at tilføje teksten "Lasso" foran perioden. Vi beholder MSE, RMSE og R² som performance-metrics.

```
metrics_lasso <- metrics_lasso_best |>
  transmute(
    Model = paste("Lasso -", Periode),
    MSE,
    RMSE,
    R2
  )
```

Best Subset Vi bruger transmute til at lave en ny tibble med kun de valgte kolonner. Vi bruger paste("Best subset-") til at tilføje teksten "Best subset" foran perioden. Vi beholder MSE, RMSE og R² som performance-metrics.

```
metrics_subset <- results_subset |>
  transmute(
    Model = paste("Best subset -", Tidspunkt),
    MSE,
    RMSE,
    R2
  )
```

Vi samler alle modeller i en tibble Vi bruger bind_rows til at samle alle modeller, og bruger arrange til at sorter efter den bedste (laveste) MSE.

```
metrics_samlet_alle <- bind_rows(
  metrics_lm,
  metrics_ridge,
  metrics_lasso,
  metrics_subset
) |>
  arrange(MSE)
```

Vi ser resultatet

```
#View(metrics_samlet_alle)
```

.3.10 Cross Validation

Vi kører K-fold CV for Lasso regressions modellen, da det er den model med den laveste RMSE

Vi laver en hjælpefunktion Funktionen beregner RMSE mellem de observerede værdier og de forudsagte (predicted) værdier. Vi skriver formlen ind for R², sætter at na.rm = TRUE, så alle NA-værdier bliver fjernet.

```
rmse <- function(obs, pred) {
  sqrt(mean((obs - pred)^2, na.rm = TRUE))
}
```

Vi laver en align_matrix Formålet med vores funktion her, er at tilpasse en matrix (newx) til strukturen i en reference-matrix (refx), så de har samme kolonner. out <- matrix(0, nrow = nrow(newx), ncol = ncol(refx)) Oprettet en ny matrix (out) fyldt med 0'er, med samme antal rækker som newx og samme antal kolonner som refx. colnames(out) <- colnames(refx): Giver out de samme kolonnenavn som reference-matricen refx. common <- intersect(colnames(newx), colnames(refx)): Finder de kolonnenavn, som findes i begge matricer. out[, common] <- newx[, common]: Indsætter værdierne fra newx i de kolonner i out, som de har til fælles. Kolonner, der ikke findes i newx, forbliver 0.

```
align_matrix <- function(newx, refx) {
  out <- matrix(0, nrow = nrow(newx), ncol = ncol(refx))
  colnames(out) <- colnames(refx)
  common <- intersect(colnames(newx), colnames(refx))
  out[, common] <- newx[, common]
  out
}
```

K-fold CV funktion for LASSO (da det er vores bedste model) Vi laver en funktion, som evaluerer, hvor godt vores Lasso-model forudsiger vores y-varibel: Antallet af tilskuere.

```
cv_lasso <- function(features, data, k = 5, lambda_grid) {

  folds <- createFolds(data$tilskuere, k = k)

  mse_vec <- numeric(k)
  rmse_vec <- numeric(k)

  for (i in seq_along(folds)) {

    test_idx <- folds[[i]]
    train_cv <- data[-test_idx, ]
    test_cv <- data[test_idx, ]

    formula <- as.formula(paste("tilskuere ~", paste(features, collapse = "+")))

    X_train <- model.matrix(formula, train_cv)[, -1]
    y_train <- train_cv$tilskuere

    X_test_raw <- model.matrix(formula, test_cv)[, -1]
    X_test <- align_matrix(X_test_raw, X_train)

    cv_fit <- cv.glmnet(
      x = X_train,
      y = y_train,
      alpha = 1,
      lambda = lambda_grid,
      standardize = TRUE
    )

    best_lambda <- cv_fit$lambda.min

    lasso_fit <- glmnet(
      x = X_train,
      y = y_train,
      alpha = 1,
      lambda = best_lambda,
      standardize = TRUE
    )

    pred <- predict(lasso_fit, s = best_lambda, newx = X_test)

    mse_vec[i] <- mean((test_cv$tilskuere - pred)^2)
    rmse_vec[i] <- rmse(test_cv$tilskuere, pred)
  }

  tibble(
    CV_MSE = mean(mse_vec),
    CV_RMSE = mean(rmse_vec)
  )
}
```

Lambda-grid Vi opretter et numerisk sekvens, der starter ved 5 og slutter ved -3 Sekvensen længde indeholder

100 jævnt fordelte værdier. Hver værdi i sekvsensen bruges som en eksponent med grundtal 10, dvs 10^{-3}

```
lambda_grid <- 10^seq(5, -3, length = 100)
```

Vores lambda_grid bliver en vektor med 100 positive tal. Værdierne går fra 100.000 (10^5) ned til 0.001 (10^{-3}). Tallene er logaritmisk fordelte, ikke lineært.

Feature-lister Vi definer de fælles forklarende variabler for de fleste modeller. Vi bruger combine (c) til at sammensætte de forskellige variabler.

```
base_features <- c(
  "temp_dry", "wind_speed", "nedbør_seneste_7_timer",
  "ugedag",
  "placering_lag_hjemme", "placering_lag_udet",
  "form_seneste_3_hjemmekampe", "form_seneste_3_udet",
  "udehold_rang",
  "sæson", "runde", "tidsperiode",
  "ferieperiode",
  "antal_transfers",
  "stadion_kapacitet"
)
```

Features for hvert tidspunkt Vi laver fire modeller med forskellige tidshorisonter. Vi bruger combine (c) til at kombinerer base_features, som vi defineret foroven, samt billetsalg varabel og mål i seneste 3 hjemmekampe. I “flere måneder”-modellen definerer hvilke variabler der skal med via combine (c).

```
features_list <- list(
  "3 dage før" = c(base_features, "d3_tilskuere", "mål_seneste_3_hjemme"),
  "7 dage før" = c(base_features, "d7_tilskuere", "mål_seneste_3_hjemme"),
  "10 dage før" = c(base_features, "d10_tilskuere", "mål_seneste_3_hjemme"),
  "Flere måneder før" = c(
    "ugedag", "sæson", "runde", "tidsperiode",
    "ferieperiode", "antal_transfers", "stadion_kapacitet"
  )
)
```

Vi kører LASSO K-fold CV

Vi bruger set.seed(1), fordi vi er gruppe #1 k <- 5 betyder, at vi anvender 5-fold cross-validation

```
set.seed(1)
k <- 5
```

Vi kører LASSO-regression for hver feature periode. Vi bruger map_dfr til at køre funktionen for hvert navn. Det samler alle resultaterne i en data-frame. Features_list vælger vores aktuelle feature sæt for hver model. Train_data er vores træningsdata. k = k: Dette er 5-fold cross-validation. Lambda_grid er de lambda_værdier, der testes i vores LASSO-model. Resultatet gemmes i res(CV_MSE/RMSE).

```
cv_lasso_results <- map_dfr(names(features_list), function(name) {
  res <- cv_lasso(
    features = features_list[[name]],
    data = train_data,
    k = k,
    lambda_grid = lambda_grid
})
```

```

tibble(
  Model = "LASSO",
  Tidspunkt = name,
  CV_MSE = res$CV_MSE,
  CV_RMSE = res$CV_RMSE
)
})

```

LOOCV - “leave one out cross validation” kunne her foretages, men da vi har mange faktorer med mange niveauer, dette vil medføre at fjernelse af en enkelt observation, at nogle niveauer kun har en enkelt observation i træningsdatasættet, hvilket forhindrer dummy kodning.

Bias variance trade-off LOOCV har lav bias men høj varians, hvorfor hver model er nærmest identisk. 5-fold giver en højere bias, men meget lavere varians, hvilket giver pålidelige vurderinger af modellens generaliserbarhed da vi allerede benytter 5-fold Cross validation på de lineære og lasso/ridge modeller, vil LOOCV ikke give ny og væsentlig information, hvorfor denne undlades.

.3.11 Prædiktion

Vi bruger Lasso-modellen til prædiktion, da det er den model der har den laveste RMSE. Vi kan estimere flere måneder før, men til 10,7 og 3 dage før kampen, er vi nødt til at lave et skøn baseret på best og worst case scenario.

Næste hjemmebanekamp er runde 20 i mod Brøndby IF, som skal spilles søndag den 15/02/2026 klokken 18:00. Deres stadionkapacitet er 9.566 tilskuere.

Vi kan lave en prædiktion for flere måneder. Vi kender ikke variabler som fx vejrdata og billetsalg endnu, men vi kan godt estimere et interval flere måneder ude. RMSE er dog højere ved modellen for flere måneder ude, hvilket betyder, at intervallet vil være større end fx modellen for 3 dage, hvilket er den model med den laveste RMSE ift alle modeller. I teorien giver dette også rigtige god mening, fordi jo flere forklarende signifikante variabler vi har, jo bedre vil vores resultat være.

Klargøring af data Det første vi skal gøre, er at forberede træningsdata til vores Lasso-nodel. Vi starter med at konverterer karakterer/numeriske værdier til faktorer. Hvis vi ikke gør dette, så vil modellen fejlagtigt antage en numerisk orden.

```

train_data <- train_data |>
  mutate(
    ugedag      = as.factor(ugedag),
    tidsperiode = as.factor(tidsperiode),
    sæson       = as.factor(sæson)
  )

```

Vi tjekker levels, og at vi har alt data med, som lige er blevet konverteret til faktorer.

```
levels(train_data$ugedag)
```

```
[1] "Fredag" "Lørdag" "Mandag" "Onsdag" "Søndag" "Tordag"
```

```
levels(train_data$tidsperiode)
```

```
[1] "Fredag_aften"        "Lørdag_aften"        "Lørdag_eftermiddag"
[4] "Mandag_aften"        "Mandag_eftermiddag"  "Mandag_formiddag"
[7] "Onsdag_aften"         "Søndag_aften"        "Søndag_eftermiddag"
[10] "Søndag_formiddag"     "Tordag_aften"        "Tordag_eftermiddag"
```

```

levels(train_data$sæson)

[1] "2003/2004" "2004/2005" "2005/2006" "2006/2007" "2007/2008" "2013/2014"
[7] "2015/2016" "2016/2017" "2021/2022" "2022/2023" "2023/2024" "2024/2025"
[13] "2025/2026"

```

Feature-lister Vi definer de fælles forklarende variabler for de fleste modeller. Vi bruger combine (c) til at sammensætte de forskellige variabler.

```

features_list <- list(
  "3 dage" = c(base_features, "d3_tilskuere", "mål_seneste_3_hjemme"),
  "7 dage" = c(base_features, "d7_tilskuere", "mål_seneste_3_hjemme"),
  "10 dage" = c(base_features, "d10_tilskuere", "mål_seneste_3_hjemme"),
  "flere måneder" = c(
    "ugedag", "sæson", "runde", "tidsperiode",
    "ferieperiode", "antal_transfers", "stadion_kapacitet"
  )
)

```

Vi definer "features" som "flere måneder" fra features_list

```
features <- features_list[["flere måneder"]]
```

Vi laver en Lasso-model på vores træningsdata

Responsevariabel Her definer y_træningsdata som vores responsvariabel. Det bruges som mål i vores Lasso-model.

```
y_train <- train_data$tilskuere
```

Designmatrix for træningsdata Vi designer en matrix, som kan bruges direkte af glmnet. model.matrix omdanner vores forklarende variable til en numerisk designmatrix. Kategoriske variabler (dem vi lige har konverteret til faktorer, bliver dummy- -kodet). [-1]: Fjerner intercept-kolonnen, og glmnet håndterer selv interceptet. drop = FALSE: Sikrer, at output altid er en matrix (også ved en variabel).

```
X_train <- model.matrix(as.formula(paste("~", paste(features, collapse = "+"))), train_data)[, -1, drop = FALSE]
```

Vi definer vores lambda-grid Vi opretter 100 lambda-værdier mellem 10^{-3} og 10^{-3} . Dette bestemmer, hvor meget modellen straffes.

```
lambda_grid <- 10^seq(3, -3, length = 100)
```

Estimering af Lasso-modellen alpha = 1 : Lasso-regressionsmodel Modellen estimeres for alle lambda-værdier vi lige har sat.

```
lasso_fit <- glmnet(X_train, y_train, alpha = 1, lambda = lambda_grid)
```

Cross-validation for bedste lambda Vi finder den bedste balance mellem bias, varians og overfitting.

```
cv_fit <- cv.glmnet(X_train, y_train, alpha = 1, lambda = lambda_grid)
best_lambda <- cv_fit$lambda.min
```

Ny kamp data Vi laver en tibble, hvor vi manuelt skriver værdierne ind for vores model, som skal forudsige tilskuereantallet flere måneder før. De kategoriske variabler bliver sat som "factor", og de numeriske værdier er forsat numeriske værdier.

```
ny_kamp_flere_måneder <- tibble(
  ugedag      = factor("Søndag", levels = levels(train_data$ugedag)),
  sæson       = factor("2025/2026", levels = levels(train_data$sæson)),
  runde        = 20,
  tidsperiode = factor("Søndag_aften", levels = levels(train_data$tidsperiode)),
  ferieperiode = factor("Vinterferie", levels = levels(train_data$ferieperiode)),
  antal_transfers = 0,
  stadion_kapacitet = 9566
)
```

Vi laver en designmatrix for den nye kampe, som matcher med vores oprindelige designmatrix (X-train). model.matrix() omdanner ny_kamp_flere_måneder til en numerisk matrix. paste(features, collapse = "+"): Sikrer at samme features som i træningen bruges. [-1]: Fjerner intercept. drop = FALSE: Bevarer matrix-format (selvom det kun er en række)

```
X_new_raw <- model.matrix(as.formula(paste("~", paste(features, collapse = "+"))), ny_kamp_flere_månede
```

Vi opretter en tom matrix med samme struktur som træningsdata. X_train indeholder alle dummy-variabler, men den nye kamp har kun en konkret kombination af levels. Der vil derfor mangle kolonner. Vi opretter derfor nu en nul-matrix med præcis samme kolonnennavne og rækkefølge som X_train.

```
X_new <- matrix(0, nrow = 1, ncol = ncol(X_train))
colnames(X_new) <- colnames(X_train)
```

Vi indsætter nu de relevante værdier. intersect() finder de dummy-variabler, der findes i begge matricer. Kun disse kolonner fyldes med 1'ere, resten forbliver 0. Hvis vores kolonner ikke matcher, så får vi enten en fejl eller en forkert prædiktion, derfor gør vi dette.

```
common_cols <- intersect(colnames(X_train), colnames(X_new_raw))
X_new[, common_cols] <- X_new_raw[, common_cols]
```

Prediktion med Lasso Vi bruger predict til at beregne outputtet for vores model lasso_fit er den allerede trænede Lasso-model, s = best_lambda vælger den lambda som giver den mindste cross validation (CV) fejl.

```
tilskuere_pred <- predict(lasso_fit, s = best_lambda, newx = X_new)
```

Vi konverterer resultatet til en numerisk værdi.

```
tilskuere_pred <- as.numeric(tilskuere_pred)
```

Vores estimat for kampdagen er 5214 tilskuere. Vi skal huske, at der kan være fejl i begge retninger. Vi bruger nu vores cross validation RMSE fra tidligere, til at finde det rigtige interval for tilskuereantallet for VFFs næste hjemmekamp.

```
tilskuere_pred
```

```
[1] 5128,975
```

Vi vælger hvilken RMSE vi vil bruge. Vi tager CV-RMSE fra cross-validation, som vi har lavet tidligere. Vi bruger filter til at isolere tidspunktet til "Flere måneder før"-modellen. Vi bruger efterfølgende pull-funktionen til at trække CV-RMSE ud.

```
rmse_value <- cv_lasso_results %>%
  filter(Tidspunkt == "Flere måneder før") |>
  pull(CV_RMSE)
```

Vi beregner vores interval

```
lower <- tilskuere_pred - rmse_value
upper <- tilskuere_pred + rmse_value
```

Vi printer resultatet

```
message("Forventet tilskuertal for næste kamp:\n")
```

Forventet tilskuertal for næste kamp:

```
message(round(tilskuere_pred), "+", round(rmse_value), "\n")
```

5129±1406

```
message("Interval: ", round(lower), "til", round(upper), "\n")
```

Interval: 3723til6535

Vi kan se at vores interval ligger mellem 4663 til 7452. Dvs vi kan ud fra vores model konkludere, at for VFFs næste hjemmekamp imod Brøndby IF vil der være et sted mellem 3808 til 6621 tilskuere.

Feature sets for alle tidshorisonter. vi definerer nu de forskellige features til de forskellige tidspunkter før kampdag flere måneder er en langsigtet prediction, hvorfor der er flere ukendte faktorer de 3 andre er mere kortsigtede, hvorfor der også er mere data til rådighed jo nærmere vi er på kampstart.

```
feature_sets <- list(
  "flere_måneder" = c(
    "ugedag", "sæson", "runde", "tidsperiode",
    "ferieperiode", "antal_transfers", "stadion_kapacitet"
  ),
  "10_dage" = c(
    "temp_dry", "wind_speed", "nedbør_seneste_7_timer",
    "ugedag", "placering_lag_hjemme", "placering_lag_udet",
    "form_seneste_3_hjemmehold", "form_seneste_3_udehold",
    "udehold_rang", "sæson", "runde", "tidsperiode",
    "ferieperiode", "stadion_kapacitet",
    "d10_tilskuere", "mål_seneste_3_hjemme"
  ),
  "7_dage" = c(
    "temp_dry", "wind_speed", "nedbør_seneste_7_timer",
    "ugedag", "placering_lag_hjemme", "placering_lag_udet",
    "form_seneste_3_hjemmehold", "form_seneste_3_udehold",
    "udehold_rang", "sæson", "runde", "tidsperiode",
    "tidsperiode"
  )
)
```

```

    "ferieperiode", "stadion_kapacitet",
    "d7_tilskuere", "mål_seneste_3_hjemme"
),
"3_dage" = c(
    "temp_dry", "wind_speed", "nedbør_seneste_7_timer",
    "ugedag", "placering_lag_hjemme", "placering_lag_udet",
    "form_seneste_3_hjemmehold", "form_seneste_3_udehold",
    "udehold_rang", "sæson", "runde", "tidsperiode",
    "ferieperiode", "stadion_kapacitet",
    "d3_tilskuere", "mål_seneste_3_hjemme"
)
)

```

Træn Lasso modeller for alle tidshorisonter.

```

train_lasso_model <- function(train_data, features, lambda_grid) {

  X_train <- model.matrix(
    as.formula(paste("~", paste(features, collapse = "+"))),
    train_data
  )[,-1, drop = FALSE]

  y_train <- train_data$tilskuere

  cv_fit <- cv.glmnet(
    x = X_train,
    y = y_train,
    alpha = 1,
    lambda = lambda_grid,
    standardize = TRUE
  )

  best_lambda <- cv_fit$lambda.min

  lasso_fit <- glmnet(
    x = X_train,
    y = y_train,
    alpha = 1,
    lambda = best_lambda,
    standardize = TRUE
  )

  list(
    model = lasso_fit,
    lambda = best_lambda,
    features = features,
    x_cols = colnames(X_train),
    factor_levels = train_data |>
      select(where(is.factor)) |>
      map(levels)
  )
}

```

Lav ny kamp.

```

lav_ny_kamp <- function(data, kamp_id, tidspunkt, feature_sets, lasso_models) {
  features <- feature_sets[[tidspunkt]]

  # Tag rækken for den kamp, vi vil evaluere
  ny <- data[kamp_id, features, drop = FALSE] # brug korrekt kamp, ikke data[1, ]

  # Behold numeriske features som de er (ingen nulstilling)
  return(ny)
}

```

Predict tilskuere.

```

predict_tilskuere <- function(tidspunkt, ny_kamp, lasso_models) {

  lasso_model <- lasso_models[[tidspunkt]]

  X_new_raw <- model.matrix(
    as.formula(paste("~", paste(lasso_model$features, collapse = "+"))),
    ny_kamp
  )[, -1, drop = FALSE]

  # Juster kolonner så de matcher træningsdata
  X_new <- matrix(0, nrow = 1, ncol = length(lasso_model$x_cols))
  colnames(X_new) <- lasso_model$x_cols
  common_cols <- intersect(colnames(X_new_raw), lasso_model$x_cols)

  if(length(common_cols) > 0){
    X_new[, common_cols] <- X_new_raw[, common_cols]
  }

  pred <- predict(lasso_model$model, s = lasso_model$lambda, newx = X_new)
  as.numeric(pred)
}

```

vi definerer en lambda grid som kontrollerer hvor kraftigt vi straffer store koefficenter for lasso, som bruges til at undgå overfitting.

```
lambda_grid <- 10^seq(3, -3, length = 100)
```

Vi træner en lasso model pr tidshorisont, med de features som passer til den. Lapply sikrer, at vi laver alle modeller uden at gentage kode.

```

lasso_models <- lapply(names(feature_sets), function(tidspunkt) {
  train_lasso_model(train_data = train_data, features = feature_sets[[tidspunkt]], lambda_grid = lambda_grid)
})

```

Vi navngiver modellerne med tidshorisonterne så vi kan bruge den igen senere.

```
names(lasso_models) <- names(feature_sets)
```

3.12 Evaluering af vores model på en allerede spillet kamp (fra datasættet)

Funktion til at evaluere én kamp på alle tidshorisontter. Denne funktion tager kamp_id og returnerer predictions for alle tidshorisontterne for hver tidshorisont oprettes: lav_ny_kamp (opretter en dataframe med features) Predict_tilskuere(laver prediction med korrekte lasso model) Resultat gemmes i tibble.

```

evaluer_kamp <- function(kamp_id, data, lasso_models, feature_sets) {

  map_dfr(names(feature_sets), function(tidspunkt) {
    ny_kamp <- lav_ny_kamp(data, kamp_id, tidspunkt, feature_sets)
    pred <- predict_tilskuere(tidspunkt, ny_kamp, lasso_models)
    tibble(
      Kamp = kamp_id,
      Tidspunkt = tidspunkt,
      Prediktion = pred
    )
  })
}

```

vi kan nu teste vores modeller med forskellige tidshorisonter ved at indtaste kamp id fra en af kampene i datasættet, indtast blot et tal mellem 1-204 og se hvordan de varierer på de forskellige tidspunkter før kampstart.

```

kamp_id <- 120
eval_data <- evaluer_kamp(kamp_id, train_data, lasso_models, feature_sets)

```

```
print(eval_data)
```

	Kamp	Tidspunkt	Prediktion
	<dbl>	<chr>	<dbl>
1	120	flere_måneder	4885.
2	120	10_dage	4183.
3	120	7_dage	3750.
4	120	3_dage	3996.

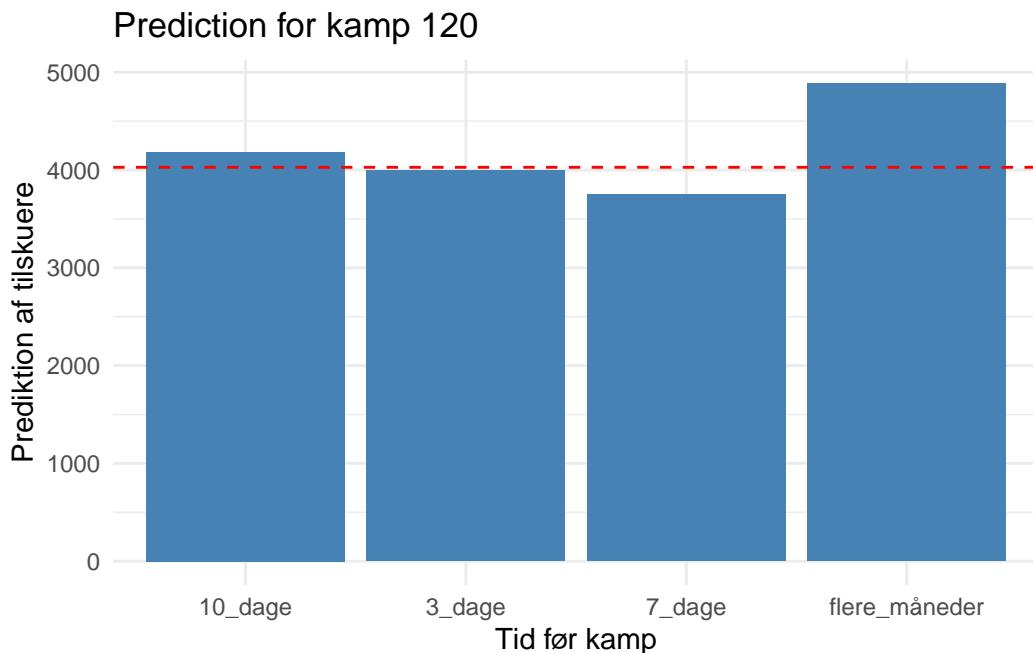
Plot prædiktioner vs. faktisk tilskuertal. vi visualiserer sammenligningen af modellens prædiktioner med det faktiske tilskuertal.

```
faktisk <- train_data$tilskuere[kamp_id]
```

```

ggplot(eval_data, aes(x = Tidspunkt, y = Prediktion)) +
  geom_bar(stat = "identity", fill = "steelblue") + # Prædiktioner som blå søjler
  geom_hline(yintercept = faktisk, color = "red", linetype = "dashed") + # Faktisk tilskuertal rød streg
  labs(title = paste("Prediction for kamp", kamp_id),
       y = "Prædiktion af tilskuere",
       x = "Tid før kamp") +
  theme_minimal()

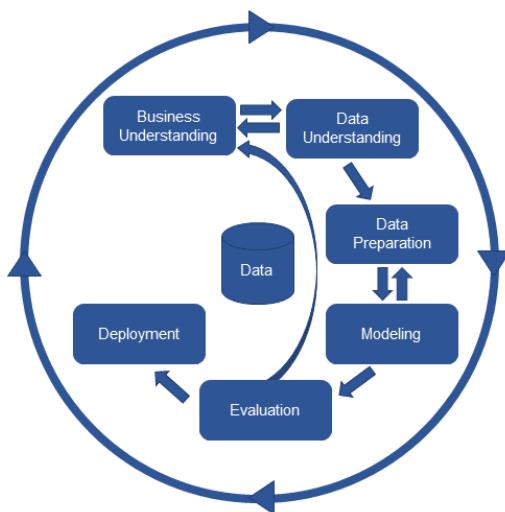
```



.4 Eksempler på markdown-sproget

Her hentes et billede

Sådan henviser/refererer du til en figur: 1



Figur 1: Faserne i CRISP-DM.

Her er et eksempel på anvendelse af en png fil, som skal ligge i dit working directory. Tekst mellem [] er figurteksten, som du ønsker i din opgave. Tekst mellem () er (placering og) navn på figuren. Tekst mellem {} er et henvisningsnavn, som du kan bruge til at henvisse til figuren i din tekst.

Her er et eksempel på en figur, som jeg henviser til 2 . En figur som JEG har lavet længere nede i koden.

Her et eksempel med en tabel 1 , som JEG har lavet længere nede.

Her er noget tekst blabla, og nu vil jeg gerne lave en reference til afsnit 0.1.

Reference fx fra interviews bla.bla..bla... eller en bog (Bang 2024) osv

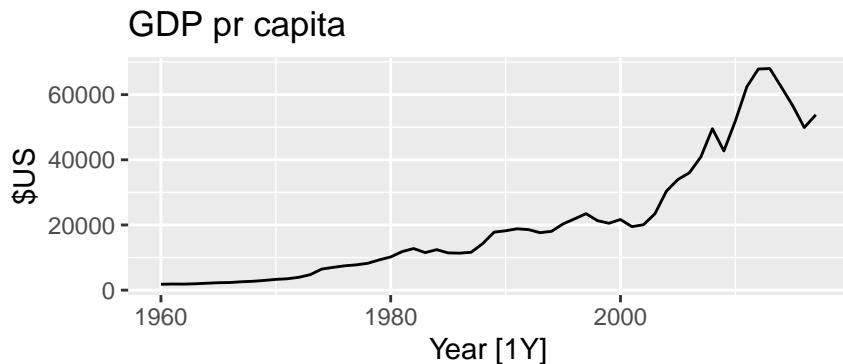
Reference til en ligning: blabla 1 blabla

$$\hat{T}_t = \frac{1}{m} \sum_{j=-k}^k y_{t+j} \quad (1)$$

Jeg kan også referere til fx en værdi, som jeg har udregnet i en kode chunk:

I forbindelse med den simple lineære regression er $R^2 = 0,59$

Lav gerne mange kode chunks i stedet for at have lange chunks - det giver et bedre overblik. Indholdet i en kode chunk skal være sammenhængende. Men lav gerne flere i forbindelse med fx, at du laver en Lasso regression.



Figur 2: Australisk BNP pr capita

Tabel 1: Outliers og indflydelsesrige observationer.

Observation	Outlier	Høj leverage	Indflydelsesrig observation	Reference til plot
16	Ja	Nej	Nej	Residuals vs Fitted, Q-Q Plot, Scale-Location, Boxplot
17	Ja	Nej	Nej	Residuals vs Fitted, Q-Q Plot, Scale-Location
39	Nej	Ja	Nej	Residuals vs Leverage
117	Nej	Ja	Ja	Residuals vs Leverage (Cook's Distance)

.5 Transskriberede interviews og præsentationer

TurboScribe er brugt til at transskribere lydfilerne der blev optaget ved besøget hos Viborg FF (Leif Foged 2025).

.5.1 Daniel

Jeg forsøger at optage det, og jeg skal nok lægge det ud til alle, så der ikke er nogen grund til, at vi alle sammen sidder med det. Dennis, hvad er det, du hedder? Daniel. Daniel fra VFF. Er det i orden, at jeg optager forløbet her? Selvfølgelig er det, og vi vil også gerne gemme det til Bo. Tak skal du have. Men måske skal der være en mere, der gør det? Rigtig juridisk, fordi jeg selv er med, som Bo er. Jeg tror, der er to, der optager nu, men I er også velkomne til at optage lørdag. Vi tager den lige i plenum. Er der nogen, der er meget imod, at der ligger en mikrofon heroppe til at optage lyd? Ingen indsigelser. Jeg tænker, det er godt nok. Er det okay, at jeg også gør det? Ja, det må du også gerne. Godmorgen. Nu har jeg lige fået en lille rundvisning af virksomheden. Mit navn er Dan, og jeg kommer til at holde et lidt længere oplæg, som handler om, hvordan vi arbejder med data i en

fodboldklub. Så kommer der nogle lidt mere specialister ind senere i dag. I kommer bl.a. til at høre fra Palle et par gange. Frem til frokost, så vil der være meget også, der holder nogle oplæg for jer. Efter det, så vil der være noget interview over på stadion. Jeg kommer mere ind på det lige om lidt. Men vi er sindssygt glade for, at I kommer her i dag. Og over på den bænk der, der sidder tre, som har taget den uddannelse, som I sidder og tager lige nu. Og I skal bare vide, at som virksomhed sidder vi sindssygt glade for de kompetencer, I kommer med. Og nu er det jo tredje gang i dag, at jeg kommer på besøg. Så jeg håber, at vi bliver bedre hver gang. Men det må være op til de tre hårdommere at vurdere det. Programmet ser sådan her ud for i dag. Vi er indkommet. Der er en kort rundvisning. Så laver jeg en præsentation på ca. 45 minutter. Det er selvfølgelig helt i orden, at vi optager og bruger det til vores opgaver. Selvfølgelig er det det. I får de slides, jeg har med i dag. Og generelt alle de slides, I bliver præsenteret for. Er der noget, som er forretningssensitivt, så tager jeg det ud. Ellers så får I slidesene i sin helhed. Jeg har prøvet at lægge nogle pauser ind. Både til at man kan tanke lidt op på kaffe osv. Har man nogle spørgsmål under oplægget, så er man meget velkommen til at stille. Så går Olga, Signe og Anja på kl. 10.10. Så er der noget pauseværk igen. Så går Palle på i en halv times tid. Så har jeg lagt en frokostpause. Så går turen til stadion. Det er fordi, vi skal over på stadion i eftermiddag. Jeg kommer til at vise jer rundt på stadion. Og fortælle lidt om nogle af de ting. Eller vise frem, hvad det egentlig betyder. Fordi jeg går ud fra, at der også er nogle, som ikke er helt lige så meget inde i fodbold. Og hvad betyder det her. Så kommer jeg til at snakke lidt om det. Så bliver Palle interviewet. Det bliver sammen med Olga også, om at bringe noget billetdata. Så vil alle os fire fra dataafdelingen være til rådighed for noget interview. Og så runder vi af 14.30. Så skulle der måske lige være, at man sidder tilbage med nogle spørgsmål efterfølgende. Når der kommer noget næste uge, og hvordan var det lige med det der. Så er I meget velkommen til at række ud til en af jeres tre undervisere. I forhold til at sende spørgsmål til os. Så er det meget velkomment. Vi har fået hjælp, hvor vi kommer til at være så transparente, som vi overhovedet kan. Der er ting, vi ikke må sige. Der er ting, vi ikke er helt ude i skoven på. Men vi havde en generalforsamling i går. Hvor der er rigtig mange ting, der bliver lagt på bordet. Så kom egentlig også bare. Men det kan være, at vi henviser til nogle andre. Hvis det ikke lige er overhovedet. Cool. Jeg vil ikke gøre det så langt. Men bare lige sådan, så I ved, hvem det er, der står og præsenterer. Jeg er 29 år. Jeg har tidligere arbejdet sammen med Simon oppe i AaB. Ikke i helt lignende funktioner, som jeg har nu. Men mere som analytiker. Hvor man filmer træninger og kampe. Og hjælper omkring førsteholdet. Jeg har været her i tre år. I nogle forskellige roller. Da jeg startede her, så var jeg en, der rejste med førsteholdet ud. Og filmede kampe. Filmede træninger. Jeg synes, at trænerne havde de bedst mulige forudsætninger for at bruge video live under kampe. Men også under træninger. I dag, så har jeg en titel "Head of Data & Technology". Og det betyder, at jeg varetager rigtig mange forskellige opgaver. Men min hovedopgave, det er: Hvordan vi indsamler, opbevarer og bruger data på tværs af funktioner. Så jeg har en kandidatgrad i økonomi fra SDU. Yes. Og det er lidt af det, jeg kommer til at snakke om. Jeg bliver tit mødt med det der med: Hvad er det egentlig, man arbejder med data i en fodboldklub? Og hvad er fodbold egentlig for noget? Så hvis der er nogen, der har nogle spørgsmål, hvor jeg tager noget for givet. Det kan være, hvad laver en scoutingafdeling? Ræk en hånd op. For der kommer helt sikkert til at være folk, der har de samme spørgsmål. Yes. Det var, hvad jeg var lidt inde på på rundturen. Jeg deler det sådan lidt op i to sektorer. Altså vi har administrationen. Det er alle dem, der varetager opgaver, som er uden for fodbolden. Det vil sige, det er dem, der sørger for, at der er boder på stadion. De har billetter. De sælger sponsorater. Når vi kigger op på stadion. Her er et billede af stadion. Og I kan se, der er solgt nogle sponsorater til nogle virksomheder. Så skal vi have nogle sælgere ud og aktivere de her virksomheder. Som kan sørge for, at der bliver lavet nogle partnerskaber. Det er cirka halvdelen af virksomheden, der gør det. Den anden halvdel af virksomheden tager altså kerneopgaver. Det er det, der har noget at gøre med fodboldspillet. Det har noget at gøre med fodboldbanen. Jeg kommer også lidt ind på, hvordan vi deler det op på data. Der er selvfølgelig nogle trænere. I dag er der nogle mentaltrænere. Altså hvordan trives spillerne. Der er nogle fysioterapeuter og læger, som tager sig af skader og problemstillinger. Fysiske trænere, der måler, hvor langt løber vi. Og så er der analytikere, som laver mange forskellige ting. Men også rapporterer på data. Og så rigtig meget video. Altså hvordan understøtter vi beslutningstagerne med den video, vi skal bruge. Administration først. Jeg opdeler det sådan i fire afdelinger. Vi har en driftsafdeling. Det kommer I til at høre en masse fra i dag. Og Palle. Det er en afdeling af mennesker, som sørger for, at der er mad i boderne. Der bliver solgt billetter. Der er nok personale på stadion. Alle de ting, der skal til for at facilitere en fodboldkamp. Det vil sige, når vi på søndag har en hjemmekamp. Hvordan sørger vi så for, at der er det rigtige i boderne? Hvordan kan vi få aktiveret fans? Hvor mange billetter har vi ude? Hvor mange frivillige betyder det, vi skal? Alle de her ting, der skal til for at kunne facilitere en god ramme omkring en fodboldkamp. Salg. De står for aktivering af sponsorer. Det vil sige, at de taler med virksomheder. I forhold til, hvordan kan vi finde et eller andet partnerskab. Hvor vi kan tilbyde noget. Ofte er det, at vi kan tilbyde en unik platform. I forhold til, at vi er den mest eksponerede virksomhed i kommunen. Selvom vi ikke er nær den

største omsætning. Så er der rigtig mange, der interesserer sig for os. Så når man hænger et skilt op på stadion. Så bliver det set af rigtig mange mennesker. Så vi har en platform, vi kan sælge på. Og så tit er det også noget med tændingserfaring / oplevelse. Som en del af det. Altså den stemning, man kan skabe på stadion. Hvordan kan man tilbyde noget til partnerne, som de ikke kan få ved at købe almindelige billetter. Det står salg for. Det er samarbejde. Når der bliver solgt billetter til sponsoraterne, hvordan bliver det så ført over til drift? Det tror jeg, at Ole kommer mere ind på senere. Så har vi en afdeling, som hedder marketing og kommunikation. Under den paraply sidder der marketing og kommunikation. Der sidder to content creators. Det vil sige, de er ude at filme og optage vores kampe. Så man kan lægge noget godt indhold på sociale medier. Så er der marketing. Så er der en ansvarlig for vores LinkedIn. Det vil sige vores virksomhedskommunikation. Og så er der en generel kommunikationsansvarlig. Der er rigtig mange medier, der godt kunne tænke sig at høre: Hvordan går det med det? Hvordan sørger man så for, at man omlægger sin kommunikation for at have et tydeligt billede. Under det, så er der også events og merchandise. Events er jo sådan lidt en... det er jo det, man selvfølgelig ikke skal gøre, når man præsenterer. Så er det i hvert fald langsomt. Events er en del af det med at... altså når der er noget, som en kunde tager på søndag efter kampen, hvor der bliver solgt brugt sportstøj, så har vi faktisk en medarbejder, der er ansat til at facilitere rammerne. For at vi kan lave andre ting end blot fodboldkampe. Så når vi senere i dag går op og kigger på vores stadion, så har vi fået en ny lounge, som kan hoste små 200 mennesker. Hvis der skal være konferencer og andet. Så kan man komme derop hver dag. Vi har kun 16 hjemmekampe om året. Og på 365 dage, så er det ikke ret mange, hvor vi egentlig har fodboldkampe. Hvordan kan vi få aktiveret vores stadion og lave flere ting, som vi kan sælge? En af mulighederne er at have noget personale, som kan bruge de rammer, vi egentlig har. Så når vi har nogle lounges, skyboxe og så videre, hvordan kan vi så på en eller anden måde skabe nogle rammer, der gør, at virksomheder kan se det som mødelokaler, konferencelokaler. Der skal vi også have nogle til det. Til merchandise, der har vi "torvesalg" / stande, som vi har nede i byen, som er om onsdag og torsdag. Og så på kampdag. Det vil sige, hvordan får vi noget af de trøjer, vi gerne vil have. Nu sørger jeg lige for at bevæge den her, så den ikke går i sort igen. Hvordan får vi lavet nogle trøjer i samarbejde med vores trøjsponsor, som er Capelli. Hvordan sørger vi for at have noget udstyr til alle de fans, som går på tribunen. Hvem er det, der køber handsker og huer osv. Det er merchandise, der står for det. Økonomien. Der er en økonomimedarbejder, som hedder Pia. Hun bor dernede. Og så vores CFO, vores økonomidirektør, Lars. Og jeg kommer lidt nærmere ind på, hvad de laver også. Ja, så Palle. Palle kommer selvfølgelig til at gå lidt mere i dybden med præcis, hvad de laver senere. Og så vil Holger også fortælle lidt om det. Men det, det handler om, det er driften af boder og billetter samt aktiviteter på stadion. Det vil sige, når vi har en fanzone, hvem er det så, der skal sørge for, at den har de rigtige ting i boderne. Hvordan sørger vi for, at der er noget folk kan købe og drikke. Her er der data på boder. Det vil Subino måske afdække lidt mere bagefter. Og så er der nogle data på billetter, som Holger er ekspert på. Selv, der er... det er lidt det her med, at vi har nogle sælgere, der tager ud til virksomhederne og sørger for, at vi er stærkt repræsenteret i det erhvervsmæssige virksomhedsnetværk, der er her i Viborg. Vi kunne godt tænke os at være noget på byen. Og det vi bedst muligt kan tilbyde, det er en platform, hvor man kan blive set og hørt. Og man kan skabe forbindelser på tværs af forskellige virksomheder. Det vil sige, at vi laver forskellige netværk. Hvis man er en tømrervirksomhed, så kan man møde en mæglervirksomhed, så kan man møde en bankvirksomhed. Så vi kan sætte folk sammen. Det er noget af det, vi kan tilbyde. På datasiden, så har vi et CRM-system, hvor vi registrerer aftaler, hvor der er styr på produkter og indstillinger osv. Og en nogle her er, at det her data tit bliver født hos os. Det vil sige, at når der er en ny partner, som starter hos os, så bliver de lagt ind i CRM-systemet. Men vi har et separat billetsystem, som hedder Eventii. Og hvordan sørger vi så for, at de to ting er godt integreret? Men som regel bliver B2B-kunderne lagt i vores CRM-system. Som jeg sagde, så handler det rigtig meget om: Hvordan kan vi lave noget indhold, altså noget content, som vi kan eksponere på sociale medier. Vi har fra sidste måned, eller starten af den her måned, lanceret en ny app, hvor man kan få sine billetter igennem, men også hvor vi kan tilbyde noget videoindhold, som vi ikke kan tilbyde på vores sociale medier. Her har vi en masse data på, hvordan B2B har foldet sig ud på medierne. Vi kan se på vores forskellige platforme, altså hjemmesider, hvad de trykker på, hvad de kigger på, osv. Vi har registreret, hvor mange, når vi laver et opslag på sociale medier, hvor mange der rent faktisk reagerer på det. Jeg kan afsløre, at når vi vinder, så er der rigtig mange, der reagerer, men når vi taber, så er der rigtig mange, som skal fortælle, hvor dårlig vores præstation var, hvor dårlig maden var i boderne. Et sjovt eksempel på det med fodbold og oplevelse er, at vi har en lounge på kampdag, hvor folk kommer op og spiser og så videre. Og så har vi nogle gange spurgt ude i spørgeskemaet: Hvad synes I så om maden? Og hvis vi har vundet, så er maden altid god. Altid god, fantastisk mad. Og hvis vi har tabt, så er det lige meget, hvad vi har lavet. Så er det den dårligste mad, der nogensinde har været lavet, og ideelle klager vælter ind og alle sådan nogle ting. Så vi er også en virksomhed, som er relativt præget af: Hvordan går det egentlig nede på fodboldbanen. Og det er ikke noget, vi kan styre, men det er bare sådan, det er. Jeg var lidt

inde på den app der. Igennem den, så får vi også indsamlet en masse stamdata på fans, som vi kan bruge i et marketing-setup. Det vil sige, at vi bliver meget mere skarpe på: Hvordan er det, vi lige præcis skal målrette vores kommunikation ind imod de enkelte fans. På sigt er planen sådan set, at vi godt kunne tænke os at blive endnu skarpere på det. Så når vi taler med sponsorer, at vi kan være meget tydelige omkring: Okay, du er en bankvirksomhed. Vi har lige nu de her fans, som I kan kommunikere til igennem os. Det er et mål, vi har på. Og så er der data på merchandise. Altså hvad er det for noget, vi sælger, hvornår er det, vi sælger det, hvad er det, vi skal være opmærksomme på. Og I er meget velkomne til at stille spørgsmål om det også. Yes. Økonomi, det er en afdeling, der skal være der. Som I kan se, det er her, vi betaler løn. Det er her, der bliver afregnet fakturaer osv. Jeg vil ikke komme for meget ind i det, det er meget klassisk, men nogle gange så glemmer man, at selv en fodboldklub skal også have en medarbejder, som sørger for, at vi holder styr på de her ting. Men det består primært af vores økonomimedarbejdere, som løser mange ting, og så vores CFO, Lars. Ja, så sporten. Og her kan jeg godt komme til at tage mange ting for givet, så hvis I har nogle spørgsmål omkring, hvad er det, hvad er det, så spørg ind. Den første afdeling, jeg vil komme ind på, det er den, vi kalder performance. I mange fodboldklubber er performance noget med fysisk data, men den kalder vi physical hos os. Der kigger vi på: Vi har en analytiker på førsteholdet, vi har en analytiker ansat på akademiet, som kigger på: Hvad er det egentlig, der afgør fodboldkampe? Hvad er det for nogle KPI'er, vi skal have? Rettet op imod: Hvordan er det, vi gerne vil spille fodbold? Hvad er det for nogle ting, der afgør fodboldkampe? Så kigger vi over tid: Hvordan har vi klaret os? Fodbold er meget sådan en sport, at du kan have en rigtig god kamp, du rammer stolpen fire gange, den går ikke ind, modstanderen løber op og laver et heldigt mål, og så har du altså bare tabt 1-0. Du kan krydse dig tilbage og tænke: Vi har virkelig gjort alt, hvad vi kunne for at vinde, men på dagen var vi der bare ikke. Og hvordan er det så, at vi sørger for, at vi holder øje med, hvordan vi præsterer? Godmorgen. Hvordan har vi klaret os? Så kigger vi lidt mere over tid. De ting, vi godt kunne tænke os at være gode til, er det så rent faktisk de ting, vi er gode til over tid? Det kommer jeg mere ind på. Benchmark vores præstation op mod andre klubber. Der snakker man meget om de ting, der afgør fodboldkampe. Hvor gode er vi til det? Når vi skal ned fra AaB i dag, hvordan er vores styrkeforhold så op mod dem i forhold til præstation? Og hvad for noget data har vi til rådighed? Her har vi fuldstændig vanvittige mængder af data. Vi har nogle dataleverandører i Superligaen, som gør, at vi egentlig har mere data til rådighed, end vi har tid til at kigge på. Så vi har i dag 12-13 millioner rækker data, der kommer bare fra én leverandør. Udover det, så har vi 2-3 andre leverandører, der leverer data til os. Udover det, så har vi nogle ejere, som sidder i London, men som er danskere, som også producerer data til os. Hvis man ikke er fodboldnord: Når du så siger data, hvad er det for noget, vi taler om? Er det antallet af skud på mål? Er det et hurtigt beløb? Det er et rigtig, rigtig godt spørgsmål. Jeg deler det op i to. Det, der sker, det kan for eksempel være skud på mål, måltaklinger og alle sådan nogle ting. Indkast, målspark. Og så er der tracking data, det vil sige, du har XY-koordinaterne på alle spillerne samtidig. Det vil sige, du kan faktisk gøre stort set, hvad du vil. Fordi du ved fuldstændig, hvor alle spillerne er, og du ved, hvor bolden er. Rigtig mange fodboldklubber bruger eventdata, fordi det er lige til, at du har noget færdigt data, som er processeret. Men i virkeligheden burde vi gå mere tracking-data-vejen. Problemet er bare, at det handler om, at du skal have nogle kompetencer i virksomheden, som kan bruge det til lige netop det, du vil. Så vi har egentlig begge ting til rådighed i forhold til præstationsmåling. Vi bruger noget af det her tracking data, men er ikke kommet langt nok med det. Og det handler simpelthen om ren manpower. Og i dag har vi faktisk også skeleton-data, som er, at vi får til alle fodboldkampe 25 punkter på kroppen på fodboldspillerne, som faktisk gør det i stand for os, så det egentlig ligner FIFA eller et spil. Så vi kan se, hvordan står spillerne hele tiden i løbet af fodboldkampen. Vi kan se, hvordan de løber. Alting. Og det har vi sådan set data på. Før det, jeg skal nok komme til, så havde vi egentlig kun XY på spillerne, og så havde vi XY også sat på bolden, så vi vidste, om bolden var oppe i luften. Men i dag har vi faktisk sådan, så vi også kan sige, og det har vi på hele sidste sæson også. Det eneste, vi bruger det til i dag, helt lavpraktisk, det er, at vi kan se situationen på den vinkel, vi gerne vil. Altså: Hvis man forestiller sig et skud på mål i en fodboldkamp, så tit er det eneste, man har til rådighed, det er videobillederne. Men det kan godt være lidt svært at se: Kunne målmanden se skuddet, kunne han ikke se skuddet? I dag kan vi bare gå ind, bruge skeleton-data, dreje det, og så kan vi se, hvad kunne målmanden se i det øjeblik, bolden er sparket. Så det er brugbart, men vi har et meget, meget stort potentiale, som vi ikke bruger i dag. Rigtig fedt. Jeg tror, det er herovre først. Jeg ved ikke... Hvordan bliver alt det der data indtastet og opdateret? Er der noget, eller er det noget, der sker automatisk? Ja, altså... På Second Spectrum, som er dem, der laver tracking, så er der otte kamераer på stadion, som egentlig indsamler det fuldt automatisk. Der er nogle forskellige tjek og sådan nogle ting der, fordi hvis I forestiller jer efterårsvêr, forårsvêr, så kan der godt være noget tåge, og nogle gange kan der også være noget pibe på stadion. Det kan være lidt svært at se, hvor er alle spillerne samtidig. Så der er nogle forskellige tjek, som man kan køre igennem. Og alle klubberne spiller ikke med samme trøjenummer, så der kan godt være nogle trøjenumre, kaméraerne kan have svært ved

at tracke. Er det nr. 17 eller nr. 11? Altså, det er ret svært. Så det kører fuldt automatisk som udgangspunkt. Så har vi eventdataen, vi har lavet igennem tagging-manuals. Så de firmaer, der har det, har en fast procedure: De får nogle videobilleder, og så taster de egentlig ind alt, hvad der sker. Så det sker manuelt hos dem, men vi får udeladt det efterfølgende. Hvad er det for nogle mennesker, hvilken baggrund har dem, der sidder og kigger på alt herinde under kontrol? Rigtig godt spørgsmål. Begge to har trænerbaggrund, så der er det godt, fordi de ved, hvad deres kompetencer er, og hvad det er, data kan hjælpe med. Vores opgave er datainfrastrukturen: at sørge for at sætte det rigtigt op og sådan nogle ting. Deres kompetence er, om data rent praktisk viser det, som de går og tænker. Så for eksempel: Hvordan kunne vi godt tænke os at spille fodbold? Jamen det er dem, der har bedre "tænder" med øjnene, og så hjælper vi dem med data. I forhold til det data, I får ind på jeres spillere... hvor I det også for andre klubber? Ja, altså til vores kampe i Superligaen. Der spiller vi imod Fredericia i aften, og det er altid sådan lidt tricky, fordi de spiller også i Superligaen, vi spiller på et Superliga-stadion. Hvorfor får vi det så ikke på den kamp? Fordi aftalen er, at vi får det til alle vores Superliga-kampe. Til alle vores Superliga-kampe har vi det på både os selv og modstanderen. Og det gør, at vi kan gøre med data, hvad vi har lyst til. Vi ved også, hvor langt hvert hold har løbet osv. Fedt. Det her, det er noget af det, jeg kommer til at tage ud i forhold til det, jeg sender til jer, men det er bare et eksempel på, at tidligere, så har vores KPI'er blevet lavet i PowerPoint og et program, der hedder Keynote, og det har været datainfrastrukturen, der har været kilden. Og så hver eneste gang vores analytikere skulle lave en opsummering af: Hvordan har det faktisk gået de sidste 10 kampe, så havde de skulle tage PowerPoint-dokumenter og fremskrive nede i hånden over i Excel. Alle sådan nogle ting. I dag, der i samarbejde med Olga, har vi gjort det sådan, at nu kan de bare komme og bede om, hvad de har lyst til, og så har vi fået datainfrastrukturen her, og så er det et ret tæt samarbejde med dem om, hvordan skal det her se ud, så det viser de ting, de skal bruge. Så nu kan de bare skrive til os: Jeg skal bruge vores KPI'er på de sidste 10 kampe, så kan vi give det til dem. Hvor tidligere var det en manuel proces, der godt kunne tage en halv dag. Så skulle han åbne alle filerne, måske printe dem ud, skrive over i Word måske, sætte over i Excel, få koderne til at virke i Excel. Det var et større arbejde for dem. Og det er helt skørt, fordi det er forskellige kompetencer. Men det ser sådan ud her i dag. Den grønne er os, den orange det er modstanderen, og forskellige KPI'er, som vi kigger på. "Færdskruer" / scores bliver udregnet af vores ejere, så det er et mål for, hvem var egentlig bedst i den her kamp. Der kunne vi godt tænke os, at den grønne er højere end den orange. Det giver os en større sandsynlighed for at vinde fodboldkampen. Det betyder ikke, at vi har vundet, det betyder, at vi har skabt forudsætninger for at vinde, eller bedre forudsætninger for at vinde. Det her, det er ting, som vi ved statistisk underbygger, at man øger sin sandsynlighed for at vinde. Det betyder ikke, at man vinder, men at være bedre på de her ting her øger vores sandsynlighed for at vinde. De ting, der er her, de er mere bygget op på, hvordan kunne vi godt tænke os at spille. Og hvorfor er de to så ikke de samme, er et rigtig godt spørgsmål. Fordi, hvordan vi godt kunne tænke os at spille, skal jo gerne bygges op på, hvordan vinder man fodboldkampen. Det vil give meget god mening. Der er bare nogle ting, som godt kan have afledte effekter. Det vil sige, hvordan vi godt kunne tænke os at spille fodboldkampen på førsteholdet, kan gøre det nemmere at finde nye fodboldspillere. Det kan gøre det nemmere for hele vores akademiasdeling at bygge et program op: Hvordan man faktisk skal træne. Så der er nogle afledte effekter af det her, som kan ses på tværs af den sportslige sektor. Og det vi viser her er mål for præstationen. Så er der fysik... og jeg har lige taget billeder af en spiller, der i dag spiller AGF, Tobias Beck. Han har en GPS-vest der. Han har sensorer. Det, man ser her, kan godt ligne en sports-bh. Men omme bagved, der sidder en GPS, som sådan set mäter X, Y og alt, hvad han laver. Så det er ligesom, hvis man løber med et løb. Det, vi får ud her, det er over 200 datapunkter: Hvor langt han har løbet, hvor stærkt han har løbet, forskellige zoner, hvor meget acceleration og deceleration han har lavet. Og nu nævnte vi jo det før, i forhold til at vi får data på modstanderen. Det gør vi jo ikke på vores GPS'er, fordi vi udveksler ikke GPS-data med de andre klubber. Det vil sige, vi ved kun, hvad de andre har lavet til kamp. Det ved vi ikke. Vi ved ikke, hvad de har lavet til træning. Nu har vi to forskellige systemer lige pludselig, som begge introducerer fysisk data. Hvordan sørger vi for, at de er sådan nogenlunde ens? Hvad kan vi bruge det her til? Jamen, vi kan tage GPS'erne med overalt. Så om vi træner herude, eller om vi træner på stadion, eller hvis vi skal på træningslejr i Spanien, så kan vi tage dem her med. Og det, vi bruger det til, det er, at vi kigger på, hvordan den enkelte spiller er "loaded". Det I kan se her, det er taget fra en spiller på førsteholdet. Og det er bygget op på forskning. Så det er en mørk grøn her. Det er for: Hvad har han lavet i total distance, altså hvor meget har man løbet i alt på den her uge. Så det er forskellige uger. I kan se, der er nogle uger, der er meget lave. Det kan der være forskellige årsager til. Nogle uger er meget høje. Der har han løbet rigtig langt i løbet af ugen. Den lysegrønne er den uge, man står i, plus de 3 sidste uger. Så det giver et mål for, har han lavet rigtig meget igennem de sidste 3 uger. Forholdet mellem den mørkegrønne og den lysegrønne er bevist i den gule. Så det er en ratio, som man skal ramme for at minimere sandsynligheden for, at spilleren bliver skadet. Og den skal helst ligge i det interval her. Og nu kan I se, den hakker en del. Og vi er ikke

så bekymrede for de her udfald. Altså det der, det er rigtig godt. Dem der er vi ikke så bekymrede for, fordi de skal ikke så meget ud. Det der, det er som regel ikke så godt. Så er der en ret stor sandsynlighed for, at han går hen og bliver skadet. Det er heller ikke super godt. Fordi så vil der helt naturligt ske... det er når man render baglæns, altså de sidste 3 uger sammen med den uge. Så kommer det til at se meget voldsomt ud den her uge. At styre de her ting på tværs af nogle forskellige... og det her, 204 og 205, det er nogle forskellige steder, man løber ind. Yes. Så det er det vigtigste, den fysiske sektor kan lave til os. Det er at kunne monitorere spillerne, så de ikke laver for meget. Fordi en af de dyreste ting for os overhovedet, det er, at vores spillere sidder ude. Så vi skal sørge for, at spillerne ligger til "rette" / tilpas. Det kan vi gøre på bedst mulig vis ved at sørge for, at de holder sig inden for den ratio der, i forhold til det, vi har sat op af den her fysiske koordinator. Hvor længe har man bedt spillerne om at spille rundt i? Og har de udfordret dem i forhold til det? Har man bare accepteret det? Det fede i fodbold med dataindsamling på fodboldspillere, det er, at du kan få fodboldspillere til at gøre stort set altid. Fordi de vil bare spille. Så hvis træneren siger: Vi begynder at spille med GPS fra i morgen, så står de klar til at tage den med GPS. Fordi de skal ikke risikere, at der er noget, der gør, at de ikke skal spille i weekenden. Så det der er aldrig en udfordring. Der kan godt være nogle udfordringer helt separat med at spise og sådan noget. Men sådan de ting der, det er der som aldrig udfordringer med. Der kan være udfordringer i forhold til udstyr, så vi har haft andet udstyr end de her, hvor de har siddet rundt om maven, og det har været lidt mere pulsbaseret. Men i en fodboldkamp, hvis man fortsætter op imod modstandere, så kan man rive lidt i hinanden og rive lidt i toppen. Så har vi oplevet, at spillerne har taget de der ned, og så bliver de irriterede, og så piller de ved dem og smider dem ud. Så vi kan godt have nogle udstyrsværdier, men på det der har vi faktisk ikke haft udfordringer. Og vi har indsamlet det i sådan over tre år, men struktureret på, at vi... altså igen, det der med datamodenhed og sådan nogle ting. Vi har rigtig god data fra januar i år og frem. Der har vi en meget, meget fin datastruktur, og kan faktisk måle lige præcis på det, vi gerne vil. Men vi har over tre års data på det. Hvor tit følger I op på den dag til den næste dag? Det gør vi hver eneste dag. Altså på det her, det er... vi har to ansatte nærmest på fuld tid på førsteholdet, der følger det her hver eneste dag. Så når spiller X skal ud at træne i morgen, hvad skal han så løbe, for at han rammer det rigtige load af ugen? Fordi han skal ikke for højt op for eksempel. Så det bliver fulgt hver eneste dag. Før træning, efter træning, når data kommer ind: Okay. Og vi kan faktisk følge det live. Altså: Uuh, han skulle kun ramme cirka 5.000 meter i dag. Han har ramt 4.900 allerede. Det går bare... Vi skal begynde at tage ham ud, så tager vi ham bare ud af træning. Fordi så har han nået noget af de fysiske mål, og så kan han godt deltage på let hold med noget let, og det kan være, at han får lov til at løbe 100 meter bare for sig selv, så han rammer de mål, han skal nå. Men de bliver fulgt fuldstændig slavisk, før, under og efter træning. Ja, jeg ved ikke, om der var først... Ja, nu ser vi da ikke så meget med fodboldspillerne. Hvilke problemer har I ellers oplevet i forhold til indtag af data? Har der været nogle modstød? Jeg oplever generelt en rigtig stor nysgerrighed på, hvad vi kan få ud af vores data. Jeg synes, der hvor vi oplever udfordringer, det er forståelsen (jeg kommer ind på det lige om lidt også) af, hvor lang tid forskellige dele af arbejdet med data kan tage. Hvor hvis du ikke har en datamæssig baggrund, kan du godt tro, at det tager jo ikke ret lang tid. Og så tager det bare fire uger, uden de oplever, at der kommer noget. Og så lige pludselig oplever de, at de ikke bare får det hele, fordi man har fået styr på data. Det vil jeg sige, det er nok sådan mine primære tanker: At der har været rigtig meget uddannelse internt: Hvad er det, der tager tid, når vi går i gang? Hvad er det så, I kommer til at opleve? Hvornår er det, det ikke virker? Og sådan nogle ting. Det har der været en del af. Fordi vi er en virksomhed, som er meget autodidakt og oplært. Du kan godt blive CEO i en fodboldklub uden at have en lang uddannelse. Altså, du kan få nærmest hvilken som helst position i en fodboldklub uden at have en lang uddannelse. Det vil sige, tit dem, der sidder rundt i roller i en fodboldklub, kan godt have et behov for at skulle lære det her data og sætte det på grund. Nu har du også været i OB. Hvordan ligger Viborg i forhold til OB i forhold til data? De hedder jo ikke " mig ", Simon. Jeg synes, det er et godt spørgsmål. Altså, jeg kender ikke OB, men det er godt sådan, at jeg kender Viborg. Så det er et godt "sæt" / sammenligningspunkt. Jeg oplevede den afdeling som minimum lige så nysgerrig, på grund af, at det er et sæt, som jeg er i. Men jeg kan ikke svare ret meget på det.

.5.2 Praktikanter

Jeg var tre år inden for HR og løn. Efterfølgende – og imellem de her to ting – har jeg faktisk også været en tur på universitetet, fordi jeg skulle finde ud af, hvad jeg egentlig gerne ville uddanne mig indenfor. Jeg tror, jeg havde en idé om, at det skulle være noget med tal, men jeg vidste ikke helt præcis, hvad det skulle være. Og jeg vidste, at jeg ikke skulle læse erhvervsøkonomi i hvert fald. Så havde jeg min serviceøkonom, og pludselig finder jeg ud af, at der er en top-up-uddannelse, man kan tage i dataanalyse. Men jeg opfyldte ikke adgangskravene

til at komme ind. Så jeg tænkte: okay, det er sådan en sats – og så søger jeg ind og kommer til optagelsesprøve ved Bjarne. Jeg skulle tage en matematikprøve først, for at vise, at jeg kunne matematikken. Det lykkedes, og jeg kom ind på uddannelsen og er i praktik her ved Viborg FF. Her har jeg været i tre måneder frem til den 31. oktober. Under praktikken har det været meget projektbaseret. Jeg har haft det her rådata projekt, som jeg nævnte sidst, da vi var ude ved jer og præsentere. Her fik jeg udleveret en masse data fra boderne på stadion – fra deres brugssystemer i mange forskellige formater, både PDF- og Excel-filer og alt muligt. Min opgave var at finde ud af, hvordan vi kunne få struktureret data, så vi kan visualisere den på den ene eller den anden måde. Hovedformålet var egentlig at få den strukturerede data klargjort. Derudover har jeg haft et andet projekt, som jeg bare har kaldt spillerstatistik. Det handler om, at man gerne vil lave benchmarking på køb, salg og leje af spillere i Superligaen – generelt fra 2021 og frem til nu. Det er stadig et igangværende projekt og har primært involveret webscraping fra Wikipedia og fra Transfermarkt (som jeg ikke kendte før), hvor man kan se, hvilke spillere der har været på de forskellige hold i Superligaen. Første step i det er også at skabe struktur i data, så man kan få det output, man ønsker til den sportslige ledelse. Vi har også haft fokus på mappestruktur, organisering og dokumentation af den kode, man laver, og hvordan man gør det bedst muligt, så man kan finde tilbage til: Hvad var det egentlig, jeg kodede? Hvorfor gjorde jeg det sådan? Og hvis jeg skal overdrage mine opgaver, når praktikken slutter, så kan en anden følge med i, hvordan projektet har forløbet. Helt i starten blev vi også præsenteret for Power BI og Power Apps. Jeg har mest brugt Power BI til et forsøg med visualiseringer af den strukturerede data, og jeg ved, at Anja har brugt Power Apps rigtig meget. Opbygningen af praktikken har været en fire-dages arbejdsuge, fordi jeg har et studiejob ved siden af, som jeg er på hver tirsdag. Der har været fleksibilitet i forhold til det. Der kommer en QR-kode senere – jeg håber rigtig meget, at I vil svare på nogle spørgsmål ved at scanne den. Det er til vores bachelorprojekt. Hvad er det for et skridt? Hej, jeg hedder Anja. Kort om min baggrund: Jeg bor i Viborg, er 35 år og har for mange år siden taget en finansøkonom. Jeg har arbejdet som økonomimedarbejder, og jeg har læst erhvervsøkonomi på Aarhus Universitet. Sideløbende har jeg arbejdet som revisorassistent. Hvis jeg tager min tidlige uddannelse med ind i det her, så er jeg rigtig glad for at have haft statistik fra finansøkonomen. På erhvervsøkonomi har jeg også haft organisationsadfærd, som jeg synes er relevant. Jeg har også været hjemmegående og haft ”hjemmefronten” i nogle år, så jeg kommer lidt rusten ind i uddannelsen. Jeg synes, det er mega interessant, men det har krævet en indsats at forstå det hele. I min praktik har jeg arbejdet i Power Apps-platformen, hvor jeg har lavet en app, og jeg vil vise jer noget af den. Jeg har også forsøgt at lave visualiseringer i Power BI – jeg havde ikke arbejdet i nogen af programmerne før; de var helt nye for mig. Jeg har været på kursus én dag i hver under praktikken, hvilket var rigtig fedt. Jeg viser lige nogle outputs fra dashboardet. Jeg er også begyndt på en dataanalyse i R, som jeg kender. Daniel nævnte, at der sidder nogen i den sportslige afdeling og beslutter, hvilke KPI'er der skal kigge på. Jeg har fået nogle GPS-data sammen med andre indsamlede data, og jeg har selv fundet nogle data, jeg har taget ind i mit dataset. Jeg har bevidst prøvet ikke at blive påvirket af, hvad andre tror og mener, men at komme med et nysgerrigt perspektiv. Jeg går ikke vildt meget op i fodbold, men ved da lidt mere nu, og jeg synes det er spændende. Jeg er dog ikke nået i mål med analysen endnu, men det bliver spændende at se, om mine indsigtter peger i samme retning som deres teori – eller noget andet. Jeg viser ikke selve Power BI-rapporten, fordi det er data på fodboldspillere, som ikke bare skal vises til alle. Jeg har taget nogle skærmbilleder, så I kan se, at det her er en side, hvor jeg kan vælge den spiller, jeg vil se. Visualiseringerne bliver vist for hele holdet i den her periode. Når jeg vælger en spiller, kommer der data frem – blandt andet navn, alder osv. Det her er kropssammensætning, og det er i samspil med træneren: Hvad er det, jeg gerne vil se lige nu? Før appen blev data registreret i Excel-ark, og nogle gange på fysisk papir, som bagefter blev overført til Excel. I Excel laver de så selv beregninger for det, de gerne vil se videre på. I appen handler det også om, hvordan data registreres, og hvad man kan bruge det til. Jeg har forsøgt sammen med træneren at få afklaret, hvad de egentlig gerne vil se. For kropssammensætningen kigger træneren på, hvad der har ændret sig fra denne test til sidste test. Man vil gerne kunne zoome ud og se udviklingen over tid – ikke kun ændringen fra sidste gang. Det har jeg leget med. Jeg blev også inspireret af Justine, som er megateknisk dygtig, og hun viste en måde, jeg ville prøve: I stedet for en drop-down til valg af spiller kan man trykke direkte på spilleren, og så vises data. Jeg har også lavet noget tilsvarende med testresultater fra hoppetests. Nu går jeg videre til appen. Appen er lavet til at registrere updates fra både kropssammensætning og nogle fysiske test. Hvis man trykker ”start”, kan man vælge kropssammensætning først: vælge en spiller, indtaste og gemme data. Jeg vil lige nævne, at jeg først lavede et bud på en app ud fra, hvad jeg troede gav mening. Undervejs observerede jeg nogle test og kunne se, at praksis var anderledes. For hoppetests foregår det fx sådan, at en spiller hopper, så kommer en ny spiller, og så hopper den første igen. Derfor giver det mening, at de bruger papir til at skrive forsøg. Appen skulle kunne håndtere, at man ikke skal ind på en side for hver spiller hele tiden – det tager for lang tid i praksis. Træneren fortalte, at de gerne vil lave test i hold/grupper, med op til fem i hver gruppe.

Derfor lavede jeg en ny løsning, så det fungerer i praksis. Jeg har også et eksempel på en test, jeg ikke kunne observere, fordi den ikke blev gennemført i perioden. Til gengæld fik jeg beskrivelser af, hvordan den foregår, og tog udgangspunkt i det. Det understreger vigtigheden af at forstå præcis, hvad det skal bruges til, og hvordan det foregår i praksis. Det er ikke dem, der skal tilpasse sig min løsning – det er mig, der skal lave en løsning, der fungerer for dem. Jeg lavede derfor mulighed for at teste fem spillere ad gangen – enten forskudt eller samtidig – men stadig med mulighed for pauser, advarsler og stop. Hvad betyder det, at man får en advarsel? Det er en løbetest, hvor spillerne skal nå at løbe inden for en bestemt tid. Hvis de ikke når det, får de først en advarsel. Her bliver spilleren testet én ad gangen, hvor de starter med en starthastighed. Når testen startes, går timeren i gang. Når spilleren når et punkt, indtaster man det. Jeg ved ikke præcis alle detaljer i testen, men det viser principippet: man kan registrere stoptidspunkt og evt. pause. Jeg gennemgik det hurtigt, men det var for at vise, hvilket output der kommer ud. Hvor gemmer man data hen? Data, der registreres i appen, kommer ind i et Excel-ark på nuværende tidspunkt. Det er fra Excel-arket, jeg har taget data over i Power BI. Der kan være datarensning i Power BI, og ellers kan man bruge R. I starten brugte jeg R, fordi jeg følte mig tryg i det, men jo bedre jeg lærte Power BI at kende, jo mere begyndte jeg at bruge Power BI. Det var hele rejsen fra dataindsamling til datarensning og output. Mit navn er Olga, og jeg har en baggrund som finansøkonom. Jeg har også været i praktik og blev efterfølgende tilbuddt en stilling som dataanalytiker. [uklart: et par sætninger om tidligere arbejde/uddannelse] Jeg vil vise en app, hvor jeg kan dele noget af alt det data, jeg har fået. Jeg fik et projekt, hvor opgaven handlede om at beregne og visualisere udnyttelsesgrad i forbindelse med VFF's VIP-spisning blandt partnere. Der er forskellige kategorier af partnere og lounge-typer (fx [uklart: "Video Lounge", "Skybox/Lounge-navne"]). Det var spændende at se, om udnyttelsesgraden faktisk fremgår, fordi man bruger ressourcer (fx øl, personale osv.). Det er værdifuldt, hvis man tidligt kan se, at der fx kun bliver brugt to pladser ud af et større antal bookede pladser. Hvis en partner konsekvent har meget lav udnyttelse, kan man begynde at følge op. Jeg skulle præsentere det på en måde, så forretningen – og især dem, der arbejder med partnere – kan bruge det. Datasættet bestod af: partner-data og hvor mange pladser der blev booket, indgangsdata (hvem der fremmødte), reservationer (hvor mange billetter/kuverter der var reserveret). Jeg vil forklare, hvorfor jeg gik tilbage til det, jeg kalder "[2M]-reservationer" ([uklart: intern betegnelse]). Det indeholder partnernavn og billetter. For at koble indgangsdata på, skulle jeg have en fælles nøgle. Det var svært, fordi det ene datasæt havde navne, og det andet ikke havde navne i samme form. I praksis blev det koblet på partnernavn, men det gav problemer, fordi partnernavne kunne staves på mange måder (punktum, mellemrum, endelser osv.). Det var en lang liste, og det blev et manuelt arbejde at standardisere. Derfor tænkte vi: hvordan kan viøre det anderledes? Hvis vi vil lave et stabilt koble mellem datasæt, skal vi have en fælles kolonne, der kan fungere som nøgle. Vi fik derfor hjælp af vores leverandører til at få tilføjet en kolonne, så vi kunne sætte datasæt sammen mere robust. Nu kunne jeg sammenkoble: tilmeldingsdata + indgangsdata, og derefter koble reservationer på, så vi fik et samlet "univers", som kunne bruges til at beregne udnyttelsesgrad. Jeg byggede en løsning og viste den via en Shiny-app (gættet ud fra "ChinaApp"), mest for at demonstrere output og struktur. Appen havde tre visninger: vælge kategori og se de "kort/kuverter/billetter", en partner har haft. Partner-visning: vælge partner og se kampe, billetter/reservationer, fremmøde og udnyttelsesgrad. Kampene var opdelt i A, B og C-grupper (formentlig segmentering). Kamp-visning: vælge en kamp og se alle partnere samt deres fremmøde/udnyttelse. Det gjorde det hurtigt at se, om en partner "er god til at fremmøde", om udviklingen er stigende eller faldende, og hvor der er afvigelser. Jeg fik feedback om, at appen måske ikke var pæn, men den gav overblik, og det var det vigtigste. Senere begyndte vi at arbejde mere i Power BI, fordi det er nemmere at overtage og nemmere at udbrede i organisationen end en Shiny-løsning, som kræver stærk teknisk dokumentation og R-kompetencer. Efter det her startede vi med at finde processerne for, hvor vi ville dele data. Jeg fik data fra bla. Tilmeldingsdata indeholdt stadig tilmeldinger, og så kom et datasæt med plus indgangsdata. Nu havde vi to datasæt. Vi kunne udregne ting uden at sætte det hele sammen manuelt, fordi vi nu havde en kolonne med en unik fællesnøgle – så fungerede det bare. I forhold til billetter kunne vi filtrere på kampdatasættet og sætte det sammen med et full join, og så var vi i mål. Nu laver jeg et datasæt til hver kamp, og det fungerer. Jeg vil ikke gå i dybden med koden, fordi jeg ikke har brugt noget "smart" – det kører bare. Jeg har brugt tid på at sætte data op, transformere, rense osv. Det hele ligger i en R-kode, som er pakket ind sådan, at jeg bare skal tilføje en ny linje, hvor jeg henter data for den nye kamp. Så er det to klik, og så har jeg et færdigt datasæt. Der er også nogle filer, jeg prøver at tilføje. Fx hvis vi skal have en tilværelsesliste/fremmødeliste (gættet) med, men jeg ikke kan se den i kampdatasættet. Så kan jeg sende det videre til Per: "Jeg vil gerne have tilføjet et event/køb-tidspunkt i den kolonne," og så kan det blive fikset i kilden. Hvis jeg vil udvide det, kan jeg gøre det ved at lave en ny fil. Det gælder fx et flow, hvor vi har opgaver, vi gerne vil automatisere. Man kan fx lave en automatisering via Outlook og lade Power BI hente data derfra. Der er også noget med, at det tager tid at opdatere, og så skal man tjekke, at det fungerer – især fordi man let kan lave en fejl (fx oprette en ny sæson eller ændre noget uden at informere

andre). Derfor er det godt med opfølgning og kontrol, når det er komplet. Jeg viser nu rapporten, vi har nu, og hvordan jeg præsenterer det til kollegaerne. Formålet var at få styr på udnyttelsesgrad. Da vi først fik styr på logikken og koblingen, gik det meget bedre, og så kunne vi kopiere logikken til de andre kategorier (fx sæsonkort, samarbejdsafalta osv.). Man kan se partner og kamp, og om det er billetter eller kort. Hvis det fremgår, kan man beregne udnyttelsesgrad. På kampniveau kan man filtrere: vælge kamp(e), vælge om man kun vil se kort, og filtrere på partnertype (fx bronzepartner, erhvervsklub osv.) og sæson. Vi taler også om aktiveringsgrad, fordi en partner kan have høj udnyttelsesgrad på de få kampe, de faktisk bruger – men stadig have mange kampe, hvor de kunne hente billetter, men ikke gør det. Derfor ser vi både på: udnyttelsesgrad (fremmøde ift. udtagne billetter/kuverter), og aktiveringsgrad (hvor ofte partneren aktiverer/bruger sin adgang). Der var også en pointe om, at mange stadig elsker Excel, og at det skal man også forvente. Hvad kan man bruge oplysningerne til – er det vigtigt for andre afdelinger? Kan det bruges til at vurdere, om man skal fastholde partnere? Pointen er ikke at ”fastholde” dem på den måde, men at aktivere dem. Man skal kontakte dem på en god måde og finde ud af, om der er noget, der gør, at de ikke bruger pladserne – og evt. tilbyde andre løsninger. Har det været svært at lære at bruge? Nej, fordi det handler om at trykke på nogle knapper og filtre. Der er også lavet en knap til at nulstille filtre, så man kan starte forfra. Der bliver spurgt ind til Anjas baggrund, og om den kan bruges. Der bliver også spurgt til, om data kan bruges i kostdelen. Svaret er, at klubben gerne vil være bedre på kostdelen, men at de ikke er der endnu – og at ambitionen er der, men det kræver data og modenhed. Der bliver spurgt til fremtiden for dashboardet. Svaret er, at man fortsætter arbejdet, men at man som virksomhed skal tænke på overdragelse: hvis en medarbejder tager ferie eller skifter job, skal en anden kunne overtage. Derfor er dokumentation vigtig – især hvis man laver Shiny-løsninger med R-kode, som kræver høj teknisk kompetence. Power BI er typisk nemmere at overdrage og sprede i organisationen. Der bliver spurgt ind til forecasting på billetter. Svaret er, at man kan trække, hvor mange billetter der er aktiveret lige nu, på tværs af sæsonkort, partnere, medarbejdere osv. Derfor har forecasting ikke været prioriteret højt i hverdagen, fordi man kan lave et ret præcist estimat tæt på kampstart ud fra historiske udnyttelsesprocenter og aktuelle aktivering. En udfordring nævnes: klubben har cirka 16 hjemmekampe pr. sæson, så der er ikke så mange observationer pr. år. For at få et godt dataset til forecasting kræver det mange sæsoner. Der spørges til procentsammensætning af tilskuere på tværs af kategorier (fans, partnere, spisende gæster osv.). Svaret er, at det er bedre at spørge Palle, som har mere præcist overblik. Der nævnes et eksempel med cirka 700 spisende gæster i en lounge, men at det ikke nødvendigvis er ”fans”, men fx sponsor-/partnergæster. Der spørges ind til dataflow mellem systemer (CRM og billetsystem, fx Eventim/Eventia). Der svarer, at man ikke bruger CRM så meget til at cirkulere data lige nu, men mere billetsystemet. Fremadrettet er ønsket, at systemerne bliver bedre integreret, så data ikke skal indtastes manuelt flere steder, og så det opleves som ét samlet setup. En deltager byder ind med, at diskussionen om Shiny vs. Power BI er interessant, og at vedkommende arbejder med et forskningsprojekt om netop værktøjsvalg, ”pro-code” løsninger og datamodenhed: hvornår kan Power BI være nok, og hvornår giver det mening at kigge på mere avancerede løsninger, især hvis man skal have data live. Til sidst afsluttes med, at man holder en pause, og at Palle kommer påbagefter.

.5.3 Palle

Klubben har rykket meget. Vi var på et tidspunkt 3,5 ansatte, så der havde man rigtig mange roller. Og i dag, der synes jeg, vi sagde i går, at vi var 800. Men jeg skal lige sige, at de tre jeg sagde, at det var administration, så i dag er vi 25 stykker. Så det er nogle helt andre roller. Så i dag går min rolle 100 procent på kampafviklingen. Så når vi har kamp inde på stadion, så er det mit overordnet ansvar, at hele rammen omkring kampen fungerer. At folk møder ind til de forskellige jobs, der er i boder og i sikkerheden og det. Hvor mange vi lige skyder på, at når vi åbner dørene, hvor mange vi skyder på, at vi har i gang på stadion. Altså under tribuneafsnittet til at servicere folk og sådan noget. Hvis vi er 6.000, hvor mange er der så i sving nede på stadion? Det er faktisk rigtigt. Det er data, jeg arbejder med, så det er meget fakta baseret. Hvad siger I? 100? Det er der i hvert fald. 400? Ja, 400 er for meget. Hvad siger du? Under 200. Under 200, nej. Vi runder 250 mennesker til sådan en mellemkamp. Så der er rigtig mange i sving for at servicere de områder, der er. Og egentlig, det er mit ansvar, at det kører. Nu ikke sådan, at jeg har berøring med alle dem, så vi har masser, som er leder af de enkelte områder. Men det er sådan for at sige lidt om, hvad skal man sige, sådan kamprammen, den er i dag. Er der nogen af jer, der kommer fast på stadion? Nej? Men det er jo også det der er sjovt ved det job, der er blevet her, fordi da jeg for mange år siden blev ansat, der blev jeg ansat i klubben, fordi det var hjerteblod, og fordi man havde en eller anden indgang til fodbolden, enten som, hvad det hedder, frivillig på en eller anden måde, eller som fodboldtræner, eller som noget med ungdomseliten. Men i dag er det jo, hvad skal man sige, meget mere

specialister, som bliver taget ind på enkelte områder. Og vi begynder jo stille og roligt at have nogle kollegaer, som måske ikke har den store indgang i fodbold, men bare fordi de er pissedygtige til de områder, de har. Og så må man så håbe på, at vi kan få fodbolden med sådan stille og roligt. Men det er jo den udvikling, der er blevet inde. Og så sent som i går, der vælger vi en ny mand ind i bestyrelsen, som efter sigende på bold.dk i dag er Brøndby-fan, men bare har en hel masse og kan bidrage med i vores bestyrelsesarbejde, på strategiarbejde, fordi han har siddet i Arbejderne Landsbank, og forhandlet med store rettigheder og sponsoraftaler med mange klubber. Så han kommer helt sikkert til at bidrage med en masse i vores bestyrelse. Så det er sådan, det er en helt ny verden. Det vi skal snakke lidt om i dag, det er specielt kampafviklingen, og specielt rammen omkring kampafvikling, og nogle af de ting, som vi går ind og arbejder med. Det øverste billede heroppe, det var sådan, vores stadion så ud for lidt før corona og nogle år før der, hvor vi sådan åbnede rammen for 2.000 mennesker, hvis det var en god dag, og skulle klargøre lidt i det. Ned til en ramme i dag, hvor vi spiller mod West Ham, og har stort set været udsolgt på tribunerne. Og vi har en kapacitet på 10.000. Så det er egentlig den udvikling, der er sket på få år, hvor rammen omkring stadion er blevet en helt anden. Og det har vi brugt ekstremt meget tid på i forhold til vores brug af data. Og udviklet hele den elektroniske platform, udviklet brugen af de data, vi opsamler på kampene. Brug fra brugsgrader og omsætningstallene i boderne på mange forskellige fronter. Så der forsøger vi at tage nogen skridt, og har brugt det til at komme videre med os. Hvis vi kigger på tilskuerudviklingen, så kan I se her 2018/2019, i 1. division var vi egentlig oppe at have et tilskuertal på 3.000, så dykker det selvfølgelig lige her lidt med corona. Men ellers så kan I se, hvor stejl en kurve det har gået op ad. De her to år her, de tre år her er rimeligt sammenlignende, fordi der spiller vi i det, man kalder nedrykningsspillet, det vil sige, at i de sidste ti kampe møder man de seks nederste hold. Hvorimod den her ene runde, der var vi heldige at komme i medaljeslutsplillet, samtidig med at vi er ude at spille europæisk. Men det der er vildt nok at se, det er at 24-25 sætter vi faktisk et tilskuerrekord på 6.400 for Superliga-kampene og det. Hvorimod det her over, der er vi selvfølgelig over med, men det er med et slutspil og det er med et europæisk spil, så hvis man tog det europæiske fodboldspil ud, så går den faktisk under. Og vi opnår faktisk et højere tilskuertal i sidste sæson "2024/2025". Det vil sige vi kæmper en lille smule med det nu. Vi ligger på omkring 6.200. Vi er ikke helt peaket, som vi gerne vil, så der er nogle ting, som vi er begyndt at arbejde på og kigger på nogle af de her tal bagved og prøver at dykke lidt ned i, hvad er det for nogle områder, hvor vi lige taber lidt terræn. Så det er sådan nogle ting, vi prøver at bruge det på. Det samme med sæsonkort. I 1819, der var sæsonkort, i 18/19 der var det den traditionelle gamle ting, med nærmest et plastikkort, som man fik udleveret, og man købte et års kort. Ret hurtigt, så begynder der at ske noget med, at man får indført abonnementer på samme måde som Netflix og alle mulige andre. Så det introducerer vi lige der i begyndelsen af, hvor corona det starter. Vi har jo ikke helt den store udvikling på det i starten, fordi som sagt corona kom. Men med det samme efter, hvor corona løsnede op igen, og vi sådan ret kraftig markedsførte os på sæsonkort og abonnementer, der eksploderer det for os. Så i dag er vi 4.100 abonnementer på, hvad det hedder, omkring stadion. Og I kan se der i 18/19, der var vi 504 årlige sæsonkortholdere. Så en helt eksklusiv udvikling. Du køber en plads, akkurat på samme måde, som du køber et års kort, hvor du har en plads. Men her betaler du bare om måneden. Og så fordelen fra, hvad skal man sige, på års kortet, de skal fornyses hver år. Det her er selvfølgelig ikke noget, med jer at gøre, men generelt så er der en del, der f.eks. får købt et fitnesskort, og det kører bare hele tiden, uden man egentligt får brugt fitnesscenteret. Egentlig er det på samme måde her. Når man først har fået købt abonnement, så kører det bare ind, så man siger det op. Har I sat tal på, hvor mange der så bruger det? Ja. Fordi det er jo så det næste. Det kommer lidt senere. Det er hele vores udnyttelsesgrad, fordi vi ønsker heller ikke bare at sælge der hvor, hvad det hedder, stole og det, det står tomt. Så det kommer lidt senere, ja. Men her i, hvad det hedder, 2023, så begynder vi egentlig at være i en situation, at vi har fået så mange abonnementer, eller så års kort solgt ud, at vi er nødt til at begynde at kigge på nogle andre løsninger. Fordi som sagt, når du køber et års kort, eller et sæsonkort om dig, så optager du en plads, og vi har 10.000 pladser. Nogle af pladserne er sat af i 1400 til udebarende afsnitte, der er 1.300 til bespisende gæster, og 1.800 til partnerbilletter. Så begynder vi sådan stille og roligt at blive med i de tal, eller de sæder, man kan sælge. Når vi så har solgt omkring 3.500 sæsonkort inden abonnement eller årsplan, så begynder der at være færre pladser at sælge. Så vi introducerer et billetabonnement, der er kommet herind til. Og billetabonnement, det er på den måde, at du stadigvæk har adgang til kampene, men du skal aktivt ind og hente billetten, når du ønsker at være her. Så ud af de her, vi er på omkring 780 stykker, de har adgang til det, og der er en garantiperiode en uge før kampen, der er du helt 100% sikker på, at du kan få en billet, men hvis du kører ind i den sidste del af garantiperioden, så kan der være udsolgt. Så det er her, at jeg håber, at vi kommer til at udvikle os, fordi dem, der ligger i den her del med sæsonkortabonnementerne, hvis deres udnyttelsesgrader er for lav, så forsøger vi at få dem over på billetabonnement, fordi så får vi ikke nogen tomme pladser. Og dem her, de bliver stille og roligt facet ud, altså vi sælger ikke mere, men dem, der har været med igennem mange år, har har et års kort og sådan nogle ting, så

de skal selvfølgelig have lov til at gøre det, men det er ikke den vej, vi ønsker at gå. Vi forsøger at arbejde meget med målsætninger i forhold til, hvor vi ligger henre på de enkelte kampe. Nu har vi nu her Vejle, den 23/11. og vi har Midtjylland, den 7 i 12. Hvis man kigger på tallene hørnede, med hvad vi ønsker at opnå, så er det faktisk sådan, i forhold til pladserne på stadion, når vi siger 75%, og I er så skarpe, så 75% af 10.000, det kan vi godt regne ud af alle sammen, hvad det er, så det er egentlig næsten udsolgt, når vi siger 7.200, der skal fremmøde, fordi vi er med på, at vi har kun en fremmødeprocent på 75%, hvorimod til Midtjylland, der går vi op og regner med en fremadprocent på 85%, så der vi kan få 8.500, og så har vi nok en overpræstering den dag, der gør, at vi kan komme lidt højere op. Men her kan I se, hvordan vi forsøger at målrette ud i de forskellige afsnit på stadion, med 100 stykkes i vores Skybox, 100 stykkes i vores Scandu's Clowns, 720 stykkes i vores BDO, samlet 1.150 i VIP-afsnittet, 2.000 partner billetter, 2.500 på Øst tribunen, 1.500 på Syd tribunen, og 300 i Udebanesafsnittet. Så det sætter vi op på målsætningsbasis, og det følger vi. Vi følger de tal, vi følger op på, hvordan det ser ud i de enkelte afsnit, og vi prøver at give et lidt på, hvad er det for nogle mennesker, der møder en specielt, herovre på vores Øst tribune og Syd tribune, og i udebaneafsnittet. Så det forsøger vi at følge ret godt. Så var der det her med brugsgrader. Her har vi vores forskellige produkter. Vi har vores billetsalg. Det ligger rigeligt på med 92% benyttelse. Og det er jo selvfølgelig, fordi det er her, du går ind og køber en billet, så selvfølgelig møder du frem, når du har betalt 150 kroner, eller hvad det er. Og hvorfor er den så ikke højere end 92? Det er fordi, vi også laver nogle aktiveringer. For eksempel så laver vi en aula aktivering til alle folkeskolerne, at de kan hente gratis billetter til en af vores kampe, hvor vi ikke forventer så højt et tilskuerantal på. Og der er vi godt klar over, at når vi laver gratis billetter ved Aula, så ligger vi måske nede på 65% i fremmøde. Så det er derfor, at den ikke er højere. Så kan I tage vores erhvervsklub. De har taget et kæmpe hop. Og det er meget tydeligt på, at vi konkret går ind og arbejder med tingene. Hvor her i de første år, altså i min første tid, der var der ikke så stor fokus på, rent faktisk om de produkter, de havde, og om de blev benyttet eller ej. Og så kan man se, at man tager fat i det, og begynder at kommunikere med det, og stiller krav om, hvad man gør. For eksempel erhvervsklubben, de har alle sammen adgang til spisning. Vi er sådan ret nederen, og regner med, at der kommer 800 mennesker, købt mad ind til 800 mennesker, og så kommer der kun 600. Så det er sådan ret tydeligt, at det, man tager fat i, det dur ikke. Og der er jo masser af gode historier med madspild og alt muligt andet, som man kan putte på sådan en kommunikation. Og det gør vi sådan ret tydeligt. Det samme med sæsonkortholderne. Som I ser, altså vi havde lige det her, forår 23, det er altså der, hvor vi spiller medaljeslutsillet, det er Brøndby FCK, Midtjylland, og alle dem der, der er hver eneste søndag. Så derfor er benyttelsen højere, men I kan se, at ved fast at arbejde med det, der kan vi altså flytte nogle ting. Så samlet på eventen, der formår vi at rykke vores benyttelsesgrader rimelig positivt. Og igen, hvis vi så tager det her efterår, vi falder lidt, så det skal vi have kigget på, det skal vi have øget fokus på. Det prøver jeg at komme med nogle eksempler på lige om lidt. Øhm, her der er egentlig bare nogle forskellige visninger af, hvordan tilskuertallene er egentlig at fordele på erhvervskort, puljebilletter, altså nu snakker vi i partnerdelen, hvordan de ligger fordelt på det. Så her har vi partner i hospitality, og herovre har vi partnerbilletterne. Og forskellen, det er enige. Altså, vi har nogle her med menu på, vi har nogle almindelige sponsorbilletter, vi har nogle 3F, og vi har nogle aktiveringsbilletter. Og der kan I sådan se lidt noget om, hvordan vores fordeling, den er på det. Så for eksempel til en, hvad det hedder, en Brøndby kamp, der har vi 397, som har med menu på, 268 med bare en almindelig sponsorbillett, 3F-billetter, 124 aktiveringsbilletter, det vil sige, det er nogle, som partnere er ude at aktivere. Det kunne være Kløverfonden, det kunne være forskellige ting. Og så har vi samarbejdsklubber til sidst. Så I kan se, vi er sådan ret, vi er ret detaljeret med data'en på de enkelte områder og forsøger at kigge på, hvad er tendenserne på det. Hvad det hedder, herovre er det erhvervkortene, og hvordan ligger det fra kamp til kamp, og i forhold til de målsætninger, vi har. Yes. Det var meget med, hvad skal man sige, vores data ind omkring billetter og partnere. Er der nogen, der har spørgsmål til det? Hvor langt tilbage har I det? Så konkret i data er, jeg har næsten lavet den samme måde at opgøre tilskuertallet på alle 20 år. Men dybden i data er blevet en anden efter at vores billetsystem er blevet så detaljeret, som det er nu. Så hvad har vi, fire år? Hvor vi har den her detaljeringsgrad? Ja. Og du sagde, data var ok ikke? Ja, godt. I forhold til abonnenterne, så har det været en ret klar strategi, at vi gerne vil have flere tilskuere på stadion. Vi går rigtig meget på fællesskab i Viborg. Altså, vi går til fodbold, fordi vi ønsker at gå til fodbold sammen i morgen. Og det er sjovere for alle parter, vores partnere, vores spillere, vores organisation, så er det sjovere at have en ramme, der hedder 5, 6, 7.000 mennesker end og ligge og rode alt for langt nede. Så det har været en helt klar strategi i forhold til at få udviklet den her abonnementssdel. Og som I kan se, så har det været eksklusivt. Vi starter med at kigge på og sige, at vi har en klar målsætning om, hvad vi gerne vil nå. Den er vi forbi, så vi har været nødt til at lave en ny. Altså, der var ikke nogen af os, da vi satte målsætningen der efter corona, at vi får 4.100 abonnenter, når vi har 587 års kort. Det var der ikke nogen af os, der så (Havde set komme). Men vi går ind og laver en målsætning om, at vi skal runde 3.000. Og vi går ind og kigger på prisstrukturen omkring det. Så vi går klart ind og siger, okay, køber I abonnement, så

kan I spare nogle penge. Så det første, vi går ind og gør, det er, at vi går ind og rører ved prisen, og så går vi ind benhårdt efter corona og siger, hvis du vil med til fodbold, så køber du et abonnement, ellers kan du ikke komme ind på stadion. Fordi vi har begrænsning på plads. Det er ikke fra min side, så jeg er billetmand, jeg skal sælge ud altså sælge stadion ud. Der er det ikke dårligere kapacitetsmangel. Det er ikke dårligere at have en lavere kapacitet, fordi med en lavere kapacitet, så stiger efterspørgslen efter de pladser, der er, og det er noget nemmere at sætte krav på det. AGF, de er i gang med at bygge lige nu stadion. De er gået fra at have stadion på hvert 24 til at have 12 eller 11.000 eller sådan noget. Og de havde flere kampe, hvor de var oppe og ringe på de 24, så de er lige blevet skudt ned til de her 12.000. De har kunnet tillade sig at bruge nogle år herpå, hvor de mega meget opdrager deres tilskuere eller deres fans. Altså, det er kun billet abonnementer dernede (IKKE HELT KORREKT), og hvis du ikke bruger dine ting, så bliver du bare skubbet en gang tilbage i køen. Partner. Helt firkantet. Henter du seks partnere, og du ikke bruger dem så får du bare ikke adgang til partnere billetter næste gang. Og det er jo fordi, at de har lige nu en langt større efterspørgsel i det antal pladser, de har. Så det vi gjorde i coronatiden, det var, du kan ikke købe en billet. Du kan købe et abonnement. Færdig bum. Og det var vi selvfølgelig med til, fordi hele vores opstigning den startede kraftigt efter corona. Og det var selvfølgelig med til at sparke det hele i gang. Vi er også meget ind på kundesegmentet og har kigget på, okay, vi har vores syd tribune, hvilke grupper er det vi har dernede. Vi har vores øst tribune, hvilke grupper vi har dernede. Og vi har vores øst tribune, hvor vi har forskellige priskategorier. Altså helt midt på med de her bløde sæder, og det er noget dyrere. Hvor hvis du så går ud i siderne, så er det noget andet. Vi gik meget ind og kiggede på, okay, hvor skal vi have faste pladser hen, hvor skal vi have frie pladser hen. I starten var det ret vigtigt, at vi havde nogle frie pladser, fordi at man kan gruppere sig, som man har lyst til. Men jo højere krav vi stiller, jo flere steder fik vi flere faste pladser. I dag er hele vores øst tribune "bestående" af faste pladser. Fordi det vil folk bare gerne have, og de vil gerne betale for det. Så vi har kunnet skrue prisen op på, hvad skal vi sige, når man tilbyder flere og mere. I starten var frie pladser jo at lavere pris. Og så kunne vi skrue på prisen stille og roligt i forhold til, hvordan vi udviklede de kundesegmenter, der er der. Og i dag, noget af det vi forsøger på at lave, det er at putte menuer med ned oveni billetprisen. Så det vil sige, at når du køber en sæsonkort, så har du betalt for billetten, plus du har betalt for stadionplaten. Og det er der, hvor vi forsøger at ændre på det nu. Og så kan vi ikke gøre det her, uden et ekstremt godt samarbejde med vores kommunikationsafdeling. Det er jo dem, der bestemmer budskaberne. Og det har virkelig givet pote. I forhold til hele den her udvikling på datasiden, i forhold til de elektroniske muligheder, så var vi det første stadion i Danmark, der gik over og havde NFC-scanning. Så det vil sige, at alle vores kort, billetter, har i dag NFC-indarbejdet. Og det betyder jo, at når man går ind og skal ind på stadion, så kan man bare lægge sin telefon op på møllen, og så er der kommunikation med den. Jeg siger ikke, at den normale skanning med QR-kode er sådan noget, ikke fungerer. Men det er en kæmpe fordel med NFC-skanningen, fordi mange har nedtonet med lys på sin skærm, eller der er nok også et par i jeres skærm, der er smadret og sådan nogle ting. Men der er der bare ikke noget i forhold til NFC-skanningen. Og vi kan se det ekstremt meget på vores indgangsdata, hvad det hedder, vi har aldrig kø. Og det er sådan noget, jeg aldrig siger, fordi det er bare der, hvor man virkelig falder i, hvis det er sådan, at det er en gang. Men altså kvarter før tid, da der er vores indgangen, der er det ryttet, for det går bare stærkt, altså man kommer hurtigt ind. Så det har været rigtig fedt. Plus, det var da også fedt at være de første. Det plejer at være FCK og Brøndby og dem der, men så er du pludselig Viborg, der er stået frem. Og det var også sådan, at det er jo sådan noget, der skal godkendes. Det her NFC, det er jo et Apple-produkt, så det skulle godkendes af Apple det tog tre uger. Altså fordi vi skulle huske at sætte skilte op det ene og det andet sted, og frem og tilbage, fordi vi var de første. Så det har været meget fedt. Så har vi fået lanceret en ny app nu her. Selvfølgelig, hvor samarbejdet med billetsystemet også er, således at du kan hente dine kort og dine billetter ned i appen der. Og specielt den del der, med at kunne håndtere sine kort deri, så man kan sende det som e-mail, eller man kan frigive pladsen, det forventer vi os helt vildt meget af, og skal have dyrket meget mere nu her. Fordi, jeg siger ikke, at den profil man har, hvor sin sæsonkort ligger inde på eventi-delen, jeg siger ikke, den ikke fungerer, men det er bare en barriere, du skal via en browser ind på en side, og så først finde det. Nu ligger det bare direkte her i appen, og håndterer kort, så kan man sende det som billet, eller man kan frigive. Og det der frigiver der, gør hver eneste gang, vi kan få folk, der ikke kommer til kampene til at frigive, så kan vi sælge pladsen igen. Og det er bare vigtigt. Fordi vi er så højt oppe i abonnementer, at vi har altså nogle gange kapacitetsproblemer i forhold til at sælge, og derfor så er det ærgerligt, hvis det er sådan, at der er nogen, der ikke får frigivet sin plads. Så vi bruger at belønne, til nogle kampe, belønne for at få frigivet, og for eksempel her til Vejle, så er alle, der har en brugsgrad over 65, får lov til at have en ven med ind til kampen, fordi vi mener, der er plads der. Og så den her send billet via e-mail, det er jo sådan, at vores kort, de er ikke personlige. Så I må godt være to herinde, der deler kort. Der er kun en, der kommer ind, men man må gerne dele sin billet, man må gerne sende sin kort videre til en, vi ønsker bare, at pladserne bliver optaget, hvis de ikke er frigivet. Det her,

altså både den del her, og det her med brugsgrader, det kræver jo også bare, at vi har styr på vores kortdata. Altså vi har styr på data'en, vi har styr på scanningerne, og fordi helt tydeligt nu her, inde på deres profiler, så ligger deres brugsgrader. Så Bent her, han er en af dem, der stadigvæk er tilbage med sæsonkort. Han har 100 procent. Så seks kampe her i efteråret, fremmede seks gange, og ti kampe i foråret, så han ligger på 100 procent. Så holder han den her, det gør han, frem til, hvad det hedder, jul, så plejer vi at have sådan en kop, så de kommer ned i kurs, eller hvad det er, vi kommer ned i vores kløvershop og henter, og det er sådan en stor tilgængelighed. Det er ikke stor, men det er et tilløbsstykke, det er noget folk lige poster bagefter. Så er 100% kruset hentet. Samtidig med, så kan man, hvad det hedder, optjene kløverpoint. Det vi bruger kløverpointene til indtil videre, det er for adgang til for eksempel kick-off events, vi har, eller udebane ture, så der skal man optjene nok pointe. Man får selvfølgelig pointe, som man kan se her, han har fået 10 pointe for at benytte sit kort. Hvis man frigiver, får man også pointe. Hvis man deler sine kort, altså sender billet, til en ven, og vennen bruger kortet, så får man også pointe. Så på den måde er der en hel masse opsamlinger af data, og vi forsøger at gøre noget, hvad skal vi sige, synliggøre deres benyttelser, og også forsøge at give dem noget, lidt ejerskab, og lidt bonus for det. Ude i fremtiden, så kunne det være, at man kunne købe i bordene for det. Der er vi ikke endnu. Men det var så nogle steder, hvor vi kunne udvikle på det. Ja. Det var det, jeg havde tænkt mig at sige. Jeg ved ikke engang, hvor lang tid jeg har brugt det. Det passer som altid perfekt på den her. Så jeg ved ikke, om der er nogen her, der har nogle spørgsmål på det. Hvordan personligt ser du det, som en succes, før en kamp? Hvad er dit mål, for din største succes? Mit mål for den største succes, vi har jo nogle KPI-mål. Vi har selvfølgelig vores tilskuertal, vi har noget, vi skal omsætte billetmæssigt, vi har noget, vi skal omsætte i boderne. Det vil jeg selvfølgelig rigtig gerne opnå. Men en rigtig, rigtig god kampe afvikling for mig, det er også, at der er ro på omkring banen. At vi har styr på sikkerheden i de specielt udebaneafsnittet. At vi ikke har nogen issues i forhold til boderne, men med det personale, der er og kontra tilskuerne. Og også, at vores bodpersonale har en god og en hyggelig dag. Vi er i en situation, at de folk, der arbejder hos os, de er som regel nødt til at have to, måske tre jobs. Fordi vi har 16 events om året. Så når folk kommer på arbejde hos os, og du ikke lige er frivillig og du brænder på det den vej der, men du kommer som timelønnet, så skal du helst have det godt. Så det er sådan en blanding af de der ting. Men selvfølgelig vil jeg gerne opnå vores KPI-tal på det. Men det er også lige så vigtigt, at vi har ro på, at vi er godt forberedt, således at det kører. Det er sådan... Det er sådan, at... Jeg betrækker sådan... Lidt vores kampafvikling, som et stort tandhjul, vi har rigtig, rigtig mange små ting, som skal fungere. Og for at det her tandhjul det kan køre rundt, så duer det ikke, at vi har et eller to områder, hvor tænderne er knækket af. Så vi skal hænge sammen hele vejen rundt. Og det er ret vigtigt for mig personligt. Så er det ret vigtigt, at vores fællesskab er i orden. Måler i sådan, hvor meget ro har der været, eller har der været nogen uheld? Hvert år der bliver der lavet i Superligaen, lavet en Superliga-survey, hvor tilskuerne til klubberne bliver spurgt. Hvor man kan måle sig op mod de andre klubber. Så benchmark med dem der. Udover det her, så har vi også kørt nogle tilskuerundersøgelser for to sæsoner siden, hvor vi prøvede at arbejde med det. De der tilskuerundersøgelser er noget, vi gerne vil gøre lidt mere i. Og derudover så har vi også undersøgelser til vores hovedpersonale, som giver feedback tilbage på, hvordan de lærer. Men der er ikke noget fast ... Den der Superliga-survey? Ja, Superliga-survey. Men der er ikke noget fast skema internt? Nej, men det kunne være et fedt punkt at få udviklet. Ja, for der kunne man højst syndes også have fået mere at måle på. Ja. Betyder det, at der er mere salg, hvis der er ro? Jamen der er mere salg, hvis der bliver scoret en masse mål, og der er spænding og sådan noget. Nej, men grunden til det der med roen, det er bare, at for mig, hvis der er ro på stadion, forstået. Altså i min del, jeg snakker ikke, at der er stille på stadion, der skulle gerne være stille. Men når jeg snakker ro, så er det, fordi vi er på forkant. Så løber vi ikke tør for tingene i boderne, altså. Så er vores bord pakket ordentligt. Og når boderne åbner, eller billetsalget åbner, eller de forskellige ting, så er klargøringen i orden. Så møder folk ind til noget, der fungerer. Det er jo egentlig sådan meget sjovt, hvis du tager sådan en dårlig oplevelse i en bod. Hvis man som kunde kommer ind der. Altså, der er jo rigtig mange berøringsflader. Altså, du kan starte med at komme i fanezonen. Så det vil sige, at den åbner tre timer før. Så derude i fanezonen, der har du berøring med de første VFF-ansatte, der er. Det kan være forhåbentlig en god oplevelse. Så går du i shoppen og handler. Igen forhåbentlig en god oplevelse. Du køber måske en billet, hvis du ikke selv har gjort det, eller går ind på nettet og henter en billet. Du er igennem en sikkerhedstjek. Det er ikke altid, at det behøves at være fedt og lige blive tjekket, om man har noget med ind. Men det skal gøres på en god måde, og der skal laves en god kommunikation omkring det. Altså, må jeg visitere dig nu? Ej, det må du ikke. Okay, er det fordi, du hellere vil visiteres dagen, eller vi skal have en pige ind til at visitere dig? Så der er en dialog om det, men du får ikke lov at gå ind, uden at blive visiteret. Så kan du ikke være med i eventen. Men der er en masse berøringsflader med mange mennesker, som gør, at hvis vi skal lykkes med det her, så skal man have nogle af de samme værdier, for at komme ind. For ellers, så kan du edermamme efter en 5-6 berøring med VFF-personale, stå foran boden, og så er de bare nogle spasser, der er inde. Selvom, at der egentlig ikke

rigtig er noget at sætte fingeren på inde ved boden. De tal, I bruger i jeres forecast, eller i at se på, hvor mange I antager, der kommer. Er den fast for hele sæsonen, eller ændres, den løbende? Når det er godt vejr, så vil vi have flere tilskuere? Vi har arbejdet med tre kategorier. ABC-kampe, og så arbejder vi på den måde, at for eksempel, nu møder vi Randers nu her, på søndag. Så går vi ind og kigger på, hvordan var indtil sidste kamp mod Randers. Hvad var vores omsætningstal der, hvad var vores KPI'er der. Så vi har kampniveauerne, og så har vi sidste år. Og så i forhold til sidste år, hvad der var, så bliver der besluttet aktivering. For eksempel Vejle, der har vi kløverklubernes dag, fordi vi ved, at det er en svær kamp. Så der har vi kløverklubernes dag, og i samarbejde med alle samarbejdsklubberne, skal vi gerne have 1000 mennesker ind til den kamp på aktiveringerne. Hvad er en A-kamp, og hvad er en B-kamp? En A-kamp, det er de fire store. Det er Brøndby, det er AGF, det er Midtjylland og det er FC København. En A-kamp, det er meget kategoriseret som noget, hvor vi kun via markedsføring på kampen, kan skabe grundlag for kampen. Hvormod en C-kamp? Det er en kamp, hvor vi er opmærksomme på, at hvis vi kan nå et tilskuertal, som er vores målsætning, så skal der ske aktivering. Nordsjælland for eksempel er en C-kamp. Men det er jo ikke, fordi Nordsjælland ikke spiller med i toppen af dansk fodbold. Der kommer bare ikke nogen folk fra Nordsjælland, og der er ikke nogen, der gider fra Viborg bare. Nordsjælland trækker i sig selv "tilskuere til". Så der er et ramme omkring kampen, som vi skal bygge på. Så der er ikke nogen placeringer at gøre? Nej. Så hvis Nordsjælland vil ligge op nummer et en sæson, så vil det ikke være? Det vil aldrig have været en A-kamp. Så det er bare ren historik i forhold til tilskuere? Ja. Jeg siger ikke, at Aab for eksempel... Aab er... Nu spiller vi ikke lige mod dem nu... Aab er sådan historisk set været en B- eller A-kamp. Men der er det lidt afhængigt af, hvor godt det går Aab. Fordi Aab har potentiale til at udfylde hele udebaneafsnittet. Hvis det går dem godt. Når du siger, at der skal være aktivering ved en C-kamp... Det kunne for eksempel være Auls. Altså den her skoleplatform. Hvor vi sammen med kommunen går ud og siger... Prøv at høre her, vi giver 1000 billetter. Og så har du mulighed for at komme ind og se kampen i Aula. Ja, det er faste rammer. Det er som regel voucher-aktivering. Hvor vi går ind og sætter rammer på, hvor meget du vil have. For eksempel den her kamp, vi har mod Randers. Der har vi åbent et nyt samarbejde med Skive IK som partnerklub. Så der har vi adgang til billetter. Vi har Grønne Bazaar. De har adgang til billetter. Og generelt sammen har vi haft adgang til billetter. Så jo, så har vi 60 kroners tilbud på syd. Som regel, så er det sådan, at når vi har vores stadion. Så har vi jo vores sydtribune, som er vores... Og så har vi vores lange side øst. Over på Tinghallen. De to steder der, der skal du helt betale for at komme ind og se fodbold. Vestsiden, modsat Tinghallen, det er vores partnerklub. Der betaler du helt automatisk, for der har du købt din billet til. Så når vi aktiverer, f.eks. de her gratisbilletter. Så er det en nordtribune nede i E-afsnit, der er dernede. Fordi vi vil ikke ind og prisdifferentiere på de to tribuner. Hvor vi siger, der betaler du for at komme ind og se fodbold. Så håber vi på, at vi kan udvikle. Vi havde en kamp... Fanden okay, det er også ligegeydigt. Jo, vi lavede en Aula-aktivering. Alle dem, der så kom ind på Aula-aktiveringens og mødte frem. De fik et tilbud om at komme til den næste kamp. Hvor de så fik 50% rabat. Så åbnede de for de næste tribuner. Og alle dem, der så brugte det tilbud og mødte frem. De får så efterfølgende et tilbud om abonnement. Så det er den måde, vi forsøger at sluse det ind på. Hvem er konkurrenterne til jer? Fordi nu snakker vi lidt om... Selvfølgelig er der jo konkurrenterne, nu spiller i jo en kamp mod Fredericia i dag og ellers i Superligaen. Men det er jo sådan spillet på banen. Hvem konkurrerer du med Palle? Du siger, det er fedt at være de første. Men det her... Kigger I sådan på de andre og siger... Vi konkurrerer med de andre klubber om tilbud? Eller er det mere lokalt at sige, at det er det er på HK? Eller er det nogle andre tilbud? Det er lokalt. Når du snakker tilskuerne. Kan du huske i den der undersøgelse, hvor mange procent der købte billetter andre steder? Det er jo ikke ret højt. Så det er meget mere det andet. Faktisk er der et ekstremt tæt samarbejde mellem klubberne. I søndags spiller vi i parken. Der er en af mine kollegaer, Alexander, som står for bodesalg på stadion. Han er over og følge deres ansvarlige for Food and Beverage. Og det er helt ned på detaljen. Hvad mål skal de have? Hvad omsætning skal de have på den kamp? Hvad har de i gennemsnitligt salg pr. tilskuere? Hvad er deres prioriteringer inden for boderne? Der er ekstremt stor åbenhed. Fordi jeg tror, at klubberne kigger på, at jo bedre vi alle sammen bliver, desto større mulighed er der for at løfte grundlaget for tilskuere i Danmark. Så på den måde er vi ikke konkurrenter. Men selvfølgelig er der en lokal, der er opgøret mod Midtjylland. Men det er jo vores fangrupperinger og sådan nogle ting. Vi har administrativt rigtig fint samarbejde med dem. Så det er nogen, man klikker bedre med end andre. Så det er underholdnings.."industrien"? Lad os nu sige, at I spiller søndag kl. 18. En A-kamp mod FCK. Og samtidig er der en eller anden stor musiker, der spiller over i tinghallen. Vil det være en konkurrent for jer? Ja, det kan det være på nogle områder. Så, ja... Hvor tit rammer I kapacitetsproblemer? Jamen, som sagt, det tipper lidt for os lige nu. Men for sidste efterår, der havde vi seks ud af otte kampe, hvor vi har udsolgt på Syd tribunen. Så vi er sådan tæt på... Ja... Jeg tror, at det er der to spørgsmål. Nu snakker vi så om gennemsnit. Hvad bruger sådan en typisk tilskuer? Det har man tal på? Hvad de bruger, når de er inde? Ja, vi ligger på 56 kroner plus moms. Plus billet, hvis det er sådan. Billetten regner vi ikke ind med, fordi

der er så mange... Vi har måske... Til de store kampe, der har vi måske 2.000, som er fysisk inde at købe en billet. Og til sådan en kamp, der regner os, der er der... Ej, så vi har nok 2,5. Men mange har jo en indgang. Altså, enten har de abonnement eller en sponsorbillett, men de er jo betalt. Så derfor bruger vi ikke billetprisen så meget. Hvis du tager i udebane-afsnittet, så nærmer det sig 200 kroner. Skal de købe 2.000 samme dage, eller? Nej, altså, vi starter... Det er simpelthen ikke 22 dage, tror jeg, der er. Der starter vi billetsalget, og så kommunikationsmæssigt. Så kører vi stille og roligt op til... Altså, alle de steder, hvor vi har faste pladser, der går folk hurtigere ind og aktiverer, fordi der er køber de pladsen. For eksempel vores syd tribune, hvor der er frie pladser. Det er meget på de sidste dage, vejr og vind, og sådan nogle ting, der er lidt afgørende på det. Du sagde jo, at forhold til vej og vind, og det har du snakket om, hvad der påvirker så for fremmødet. Udover hvis der er musik, eller jeres håndboldklub spiller. Hvad andre ting påvirker fremmødet? Altså, kan I mærke, hvis det går godt, eller hvis der er krig med Rusland og sådan noget? Kan I mærke det? Det ved jeg faktisk ikke, når du sådan specifikt siger, hvordan udviklingen er. Det, jeg synes, vi kan mærke, bruger vi lidt intet nogle flosker, fordi vi var på studietur i Fårup Sommerland. Så tænker i hvad fanden Fårup Sommerland og fodbold, men det er et event begge dele, og det er et spørgsmål om at servicere folk rigtig godt. De fortalte det sådan, at når de løftede deres tilskuertal, så var det fordi, de byggede en ny rutsjebane eller et eller andet. Men den rutsjebane, den kunne maks. holde sådan 2 år, så omkring to år kunne den maks. give, at du holdt det hop, du steg i tilskuerne. Så er det eventen omkring Fårup Sommerland, der gør, at du fastholder det hop, du har taget. Og jeg tror, at det er lidt der, hvor vi er lige nu, at vores bygger-rutsjebanen var først, og komme i Superligaen igen, ud at spille europæisk og med dansk medaljeslutspl. Der peakede vi der byggede vi vores rutsjebane. Og lige nu, der mangler den lige det der hop, og det kan være mange forskellige ting. Det vi arbejder med lige nu, det er, hvordan er det, vi fastholder den der attraktion i det. Fordi en fed fanzone, oplevelser i fanszonene, det er kun med til at understøtte. Altså folk kommer ikke kun for at opleve, at vi lever i fanszonene. Så det er helheden i det. Men vi har klart ønsker og mål om, at selve resultatet på banen, bliver en mindre faktor. At det er fællesskab og oplevelse sammen, det er det, der skal bære det. Så tager vi lige de sidste to hænder. Har jeg skruet mig derover et par gange eller hvad? Nu skal vi se, om vi kan love at sige, at det skulle aftales. Ja, det kan vi. Med ti, syv og tre dage op til kamp, hvad har det betydning for dig, de sidste ti, syv og tre dage? Det er jo en ingen tvivl om, at jo tidligere, at man ved noget om det er, så er der jo hele klargøringen til boderne og sådan nogle ting. Der er jo forskel på at klargøre en kamp, hvor man har en forventning om, at der er 8.500 på lægterne, og så har man klargøringen til, at der er 6.500 eller 5.500. Så det er, at der er nogle tal der, som vi kigger på. Så er der rigtig meget omkring antallet af, hvad det hedder, boede personale. Altså, der er markant forskel på, om man skal servicere 8.500, eller skal servicere 6.500. Så det er så meget den forberedelsestid, der er i det. At det er den, der er vigtig for os. Så sådan er en fejlmarken for dig. Hvad kunne den ligge på? Hvad der er der? Hvis nu vi anslår et eller andet antal, det her begynder at komme, der vil der altså være en usikkerhed. Det er jo tidligt, at du begynder at snakke om det. Hvad er det med dig? Det er dig, der har forsøgt også. Nej, nu skal jeg ikke. Men nogen gange var det en god forskel. Det første del, forresten. Nogen gange kunne vi ramme det. Men vi fandt ud af, at hvis man prøvede at give en view på den forecast, som vi havde lavet, f.eks. hvis man nu havde lavet nogen, hvad hedder det, kvinder eller noget, så kan man arbejde med det. Men hvad sagde du så meget sidste gang? Plus minus ti? Nej, det var jo ikke det. Men jeg har så fandt ud af, at det betyder ingenting. Så er vi derhenne. Men altså, du skal ikke spørge, hvad jeg har lavet. Noget af det, vi oplever, når vi arbejder med det, er, at på en ting har vi rigtig meget data. Men vi mangler nogen gange noget specifikt på en kamp. F.eks. vi har 16 kampe om året. Men vi møder måske kun holdet én gang. Måske to. Så hvis nu, at vi har spillet fire sæsoner, så kan det rent faktisk være, at vi kun, selvom det er fire gange 16, så kan det være, at vi kun har data fra fire gange fire kampe. Og der kan være helt vildt store forskelle på, om det er maj måned, september måned, december måned, om det er klokken to, klokken fire, klokken seks. Så... Men jeg har store forventninger. Vi har dem. Jeg tror, det var det sidste spørgsmål, så runder vi af. Jeg vil gerne høre lidt om et segment. Ja. Ja. Altså demografien med fordelinger. Ja. Eller køn. Køn kan jeg næsten gætte mig til. Hvad gælder du dig selv om? Jeg tænker hovedparten af mænd. Men det er også bare... Det er jo den idee, jeg har. Var det en følelse eller fakta? Begge dele, tror jeg. Jamen det er jo... Vi har haft en voldsom stigning i... I hvad det hedder... I piger og damer. Så det er ret fedt. Der hvor vi sådan helt konkret har oplevet issues med det, det er på toiletterne. Så der har vi været nødt til på syd tribunen at bygge nye dametoiletter. Fordi vi simpelthen ikke har kapacitet til det. Jeg kan ikke sige dig tallene 100%, fordi den måde, vores profiler er bygget op på, så er det sådan, at vi er ikke kommet dertil endnu, at det er et kort én profil. Det er der for eksempel i FCK, og der er der også hvis du køber til Landskampe og sådan noget. Så der ved man, om det er et barn, eller det er, hvad skal man sige... Altså om piger er en dreng, eller om det er en mand eller en dame. Så det eneste vi har, det er på kategorierne. Altså priskategorierne. Barn, voksne, studerende, eller hvad det hedder, pensionister. Fordi der er rigtig mange, så har de fire profiler til at ligge på deres... Altså de har fire forskellige... Altså de

køber for fire personer, og så kan det være, at det er mand eller kone, der bare. Så jeg kan ikke sige fakta på det. Ikke andet, at vi har oplevet... Altså det er ikke bare far med søn længere. Slet, slet ikke. Det er familier, der kommer af sted. Og det er... Det er langt, langt mere piger og damer, som også møder op, og er en del af vores stemningstribune. Og hele det der fællesskab, det tror jeg, det betyder ekstremt meget. Ja. Ja, fordi jeg tænker, at det har en stor betydning i forhold til Merchandise. Ja, ja. Altså, ja. Godt. Så snakker vi over, eller hvad? Skal vi ikke give dem en hånd, og så... Så kan vi lade dem lige se, så man mixer lidt mere.

.5.4 Palle & Olga

Det er blevet lidt mere systematiseret, ja. Men det er altså, hvad skal vi sige, vores billetdata, altså vores billetdata-platform, det er et univers, hvor der er en masse statistikker og sådan nogle ting. Så vi kan godt bare arbejde i billetsystemet, uden at... Men ja, det er meget udtræk. Jeg ved ikke, hvad der sker. Vi har også snakket om, at der er udarbejdet aviki til eventii. Så det er måske noget, som vi kommer til at gøre i fremtiden. Men så beredte har vi besluttet, at det er nogle andre områder, som har brug for vores tid nu. Det er derfor, det er lidt... Men det er en mulighed, som kommer til at fungere, faktisk. Men altså, vi skal bare lige supplere den her. Så er det jo, hvad det hedder... Der er ikke andre klubber. Altså, vi er dem, vi har eventii. Vi har firmaet ude. Der er ikke andre klubber, der gør det. Så Olga og Daniel er de første, som skubber sidder der. Så ja. Ja? Hvis Viborg klare sig godt de sidste to kampe, er der så flere tilskuere? Hvis i så har tabt de sidste to, er der så en sammenhæng med tilskuere. Er der en sammenhæng i det? Jeg tror, at historien er længere, men det er jo logisk, at får man ikke så mange point på kontoen, så er der en effekt af det, men det er sådan længere. Det er jo ikke fordi, at vi har blaest hele ligaen væk. Vi ligger alligevel stadigvæk med 6.200 og var der jo 6.400 sidste år. Så en mindre effekt. Ikke så stor effekt, som vi ser. Nu snakkede du lidt om, at når I kigger på, hvor mange, I antager, der kommer til en kamp, at det bare er dem, vi spiller, imod omsætningstallene. Er der noget, som måske ikke står på papiret som stadigvæk? Det har en indflydelse med, om de tænker på, om det er solskin eller en søndag weekend, så man måske ikke medregner det, men ligesom bare tager et valg ud fra på selve dagen, eller dagene op til. Men vi har undersøgt det der under selve studiet. Vi har undersøgt, om vejr har en indflydelse. Vi fandt ud af, at det ikke har indflydelse. Så det er ikke så stor forskel. Fordi hvis man nu har købt billetten, og hvis det er en vigtig kamp, som man kommer, så er det ikke den største forskel. Og specielt vi, der venter på, at det regner hele tiden. Så det er der lige meget. Men i forhold til de forskellige ting, vi har lavet nogle overblik, da jeg var i praktik. Også i shiny. Men vi har aldrig gået videre med det. Altså jeg tror, der hvor vores udsving er størst, i forhold til tid og vejr, det tror jeg, det er både sandt. Vores volumen, vores økonomi er markant større, en god sommerdag, måske fredag aften, er markant større. Altså jeg tror, da vi har FCK og Sønderjysk, og fredag aften er i starten, da mener jeg, at vi ligger på 67-68 kroner i omsætning per tilskuer. Hvor mod, at vi mod OB er helt nede på 47. Så der er stor forskel. Men det er også meget logisk, fordi køber du en fadøl kontra en kop kaffe, så er der i hvert fald en 20'er i forskel. Så noget betyder det. Har I egentlig prøvet at krydstjekke det med andre arrangementer? Nej, og ja. Hvad for noget er det, så siger hun? Hvis der nu kommer en stor koncert, måske i Horsens, at det kan trække så mange mennesker samtidig? Der er noget stor forskel, hvis du skal sammenligne det. For eksempel i en koncert, der vil langt hovedparten være fysiskude og købe billetten. Hvis vi er 8.500 her, så er det måske maks. 3.000, som fysisk har købt billetten til kampen. De øvrige har købt billetten via partnerskaber eller via abonnementer. Så det er jo ikke nødvendigvis bare lige til den ene kamp. Det er jo til en serie af kampe. Så der er en forskel. Så det vil have en betydning på de billetter, som ikke er solgt i forvejen, hvis der sker noget i området? Ja, det kan det have, hvis vi ryger i en stor event. Jeg forstod det sådan, at man kunne sammenligne i forhold til fremmødet i Horsens og hos os. Nej, nej, nej. Mere om, hvis der er en stor begivenhed, der gør, at folk måske bare vælger... Smukfest, for eksempel. Det er en konkurrent, når det er der. Og det er heller ikke til at få bodpersonale, fordi de er også dernede alle sammen. Er der en grund til, at vi siger køb per tilskuer, eller omsætning per tilskuer, og ikke omsætning per køb? Jamen, det arbejder vi også med. Det arbejder vi også med. Hvad ligger antaget køb på procentvis? Ej, det kan... Det ved jeg simpelthen ikke mere, end at det er 100 og nogen kroner, der er i kassen hver. Men grunden til, at den også er svær, og nok grunden til, at vi ikke arbejder så meget på den, det er, at vi forsøger, som jeg sagde på abonnementerne, der forsøger vi at sælge menuer med ind. Og i en normal kamp, der har vi på vores partnerside og på vores menu side i abonnement, der har vi solgt omkring 1000 menuer. Og når de bliver købt i boden, så er det jo 0 kroners omsætning. Så derfor bruger vi meget mere, vi bruger omsætning, altså at salget et per tilskuer, og så bruger vi antallet, der handler. Og antallet, der handler er antallet af transaktioner. Vi skal helst ligge på omkring 75 procent af de tilskuere, der er, de skal handle. Så det er mere det tal, vi bruger. Den krølle, der er, får vi faktisk på, hvad det hedder, på søndag kommer der en ny, en ny en der leverer kassesystemet til

os, som skal prøves, og så sætter de selvfolgelig et eller to testterminaler til den her gang, og fire til de næste. Og det er meget, fordi vi ikke kan få nok data ud af vores nuværende, at vi ønsker at skifte den. Ja? Vi tænkte på, at nu har du sagt, at når i har events, så I laver aktivering og sådan noget. Registrerer I på nogen steder, jeres data, hvis I har været i gang med at lave aktivering, i forbindelse med en kamp? Prøv lige en gang mere, altså om vi ved, hvad vi har til hver enkelt kamp? Nej, hvis du ved du har en kamp, hvor vi måske forventer, at der ikke kommer særlig mange, så I laver de her aktivering af billetter. Registrerer I det, i jeres data til kamp, hvor I siger, at I har aktiveret de her billetter? Ja, fordi den måde, at vi laver ... Vi går ikke ned i gågaden og deler billetter ud. Nej, det gør vi også ikke. Alt vores aktivering sker via voucher-koder. Så alle, der bruger de her aktivering, opretter en profil, så vi ved, hvem vi snakker med, og så bruger de voucher-koderne. Så vi ved altid via koderne, hvordan tilgangen er til billetter. Jeg tænker mere i forhold til, at man registrerer dit eget til den her kamp, at vi har haft ... Det gør vi også. Det kører på tilskuerfordelingen på kampen. Så ved jeg, at for eksempel sidste gang til Aula, der var der 625 billetter afhentet, og 462 blev benyttet. Er der en variabel, der har overrasket dig mest, efter at I begyndte at få mere data ind? Er der en variabel, der har overrasket mig mest? Noget af det, vi klør lidt efter lige nu at finde ud af, det er jo netop, at vi har egentlig haft en stegl kurve, der er gået, altså en rimelig fin opad. Og nu har jeg som sagt sagt, at den er knækket lidt for os. Og der er vi sådan ved at forsøge at finde ud af, hvad er grundlaget? Udover fodbold, for eksempel. Hvad er grundlaget? Og der må jeg sige, at noget af det, vi tror, vi har analyseret os frem til indtil videre, og jeg er nødt til at sige, at det stadigvæk er lidt en følelse frem for fakta, så det er vi godt klar over, at det ikke er så godt i det her rum. Men det er jo meget den indsats, vi selv gør, så hvis vores fokus bliver flyttet væk fra fremmøde, så kan man også se det. Også der er mange steder, hvor vi i vores kommunikation, og hvor vi i vores opfordring til at frigive pladser, og hvor vi i vores opfordring til at udnytte de pladser, de har. Når vores fokus bliver fjernet en lille smule fra det, så er vi måske mere over på det nye Apple-univers, vi skal have fortalt om, og vi laver meget mere content i forhold til aktiveringens videomæssigt og sådan nogle ting. Så hvis vores fokus der har været flyttet lidt en smule derfra, så tror jeg måske, at det vil være mit bedste svar på, at jeg synes, at vi kan se vores indsats, den kan vi se, hvor meget den giver pote ind på det. Og det tror jeg, at det er en af tingene, vi har tabt en lille smule. Det kan ikke at vi ikke har lavet noget, når der er noget andet, man har haft fokus på. Men jeg har ikke tal på det. Jeg har et spørgsmål i forhold til, hvordan I arbejder med data. Jer og sammen med Olga. Er der nogle platforme, som i foretrækker, som er bedst at bruge? Jeg arbejder i Excel med det. Det Olga arbejder i er jo 100 gange smartere. Så det er jo helt klart den vej, vi skal. Det vi skal, det er, at Olga får bygget alle de her rapporteringsværktøjer. Således at data'en bare bliver puttet i en silo der, så kører det. Så er vi i gang med ret store projekter lige på det. Har du brug for mellemregningerne Palle? Eller har du nogen tal, det er de tre ting. Du har defineret, hvordan det er, at du har adgang til tallene. Eller har du brug for, at du kan... Jeg har brug for mellemregningerne. Jeg vil vide hvad der ligger under. Men det kommer til at blive helt forkert, det her. Det er jeg ikke sikker på, at vores direktør har behov for. Vi har behov for at vide noget mere, end bare det store tal. Du viste de her skemaer overfor, hvor mange billetter der var til forskellige kategorier. Ved forskellige kampkategorier. Hvor fastlåst er sådan nogle tal? Er det noget, du kan gå ind og tweake? Nu viste du en A- og en B-kamp. Jeg kan ikke huske, hvilken hold det var. Men der var der i hvert fald væsentlig forskel på nogle af tingene. Grundlaget er jo nogenlunde ens. Fordi det er jo produkter, der er solgt ind. Hvis vi tager abonnementerne, og de er så rimelige. Det er jo det samme antal, der er. Så er spørgsmålet, hvor mange der trækker billetterne. Hvis man kigger på vores partners side, så er deres adgang til billetter delt op i to kategorier. Der er det, man kalder eventbilletter. Det vil sige, at de har to billetter til alle kampe. To billetter til hver kamp. Så har de også noget, der hedder puljebilletter. Der kan de for eksempel have 20 puljebilletter. Men de bestemmer selv, hvornår de bruger de puljebilletter. Så for eksempel i en A-kamp, så kan de vælge at bruge... Jeg tror, vi har en grænse på 6 indtil videre. Men mindre man lige spørger, må man bruge 10. Så derfor er det egentlig den måde, der kan være forskellen på. Men det er mere, hvordan virksomhederne selv vælger at bruge deres puljebilletter. Eventbilletterne er to til hver eneste kamp, for eksempel. Så en del af rammen er den samme. Så i virkeligheden ligger den her fordeling fastlåst ret tidligt i processen? Ja, men fordelingen er jo, hvordan billet trækkes er. De eneste, der har fast pladser. Altså, som har et sæde på stadion. Det er dem, der har årssæsonkort og abonnementer. Og så er det 250 partnere, som har faste pladser ude i de sorte sæder. Ellers alt andet er billettræk. Det vil sige, du kan godt have fire pladser til kampen, du kan bruge. Hvis du ikke går ind og aktiverer dine pladser, så er pladserne ikke optaget. Så bliver de ikke taget. Måske lidt opfølgende. Du havde en tabel over nogle målsætninger. I forhold til netop også med de her fordelinger. Hvor mange i antog, at der kom til hver? Hvordan er det fastsat? Er det noget, I har et system for? Hvor får I tallene fra? Altså, hvordan målsætningerne udarbejdedes? Ja. Vi starter med at kigge på kapaciteten, som vi har på stadion. Så starter vi med at forholde den struktur, vi laver. Vi har nogle opdelinger. Vi har en Syd tribune stemningsskabene tribune. Der er nogle begrænsninger på de billettyper de har. Vi har en Øst tribune. På den

side derover, som jeg til sæsonkort holder, der er nogle begrænsninger. Så har vi nogle partnerafsnitter. Så det er meget, hvad der er solgt ind til de enkelte områder. For eksempel partnerbilletter. Jeg tror, det er 1.800, når man regner ud på snit. 1.800 billetter er solgt til hver eneste kamp. Så jeg skulle helst tage plads til dem. På den anden side er det også ret vigtigt, at de bruger de pladser. Hvis nu de ikke rigtig aktiverer de billetter, de har, så er vi stående med nogle tomme pladser. Og hvornår kan vi tillade os at åbne det for almindelige billetter? Så det er på baggrund af en struktur og et salg derinde. Og så det vi egentlig arbejder på, det er, hvad skal der så gøres til sidst af billetsalg eller ekstra billetaktivering eller ekstra ind salg til partnerne. Hvordan vurderer I, hvad I skal gøre? Er det noget, du sidder manuelt og gør, eller har I nogle retningslinjer for det? I forhold til de enkelte kampe? Ja, når I når hen til en kamp et par dage op til, så har I sådan nogle retningslinjer. Hvad gør vi, hvis det ser sådan her ud i forhold til vores målsætning? Målsætningen skal laves fire uger før. Hvis det er sådan, at det er rigtig svært at nå at ændre det, så er det helt store. Jeg kan godt hurtigt ændre en pris, det tager ikke altid lang tid. Men hele den kommunikationsmæssige flow, det skal gerne indarbejdes i det. Så derfor skal du gerne være så tæt på, som overhovedet muligt, strukturmæssigt. Men det, vi arbejder mest med, det er fremmøde procenter i forhold til tallene. Og det er jo derfor, I så med 75 og 85. Så det er meget det, vi arbejder med. Og det er jo ud for historiske tal. Hold da op. Nu sidder du jo, Olga med, vi har jo lige hørt, at der også kommer et nyt projekt lige om hjørnet. Men nu er du kommet ud i fuldtid og sidder med deres både nuværende lyder det til, men også nye programmer. Kunne du ikke prøve at sætte et ord på, hvor I er nu? Og hvor er jeres målsætninger at komme hen med alle de her nye systemer? Systemer, du mener er Power BI og? Ja, Power BI og alt det, det data i benytter. Så lige nu er det mange manuelle processer, I er inde i. Er det noget, I gerne vil ud af? Vi prøver til at få strukturer på data. Det er lidt afhængigt, hvilken afdeling vi snakker om. For eksempel, Daniel han er utrolig god til at samle på data og så strukturere. Så alt det der data, som vi ikke selv har, men de ser ens ud. Så kan man tage det og begynde at arbejde med det. Det er også sådan, at jeg er startet med at arbejde på fuld tid. Og det er også meget begrænset til, hvor jeg er her. Så nu har jeg samarbejdet meget med Daniel i forhold til data. Det har også været lidt afhængigt, som jeg har struktureret. Så det er ligesom jeg sagde, at jeg har været under nogen situationer arbejdet i R. Og så er det Power BI. Power BI har også DAX, der også går i spil. For mig måske giver det lidt mere mening til at bruge R, fordi det er der, hvor jeg har styr på, inden vi begynder at tage Dax det også. For det er et spørgsmål, så vi prøver at producere de der rapporter. For eksempel, Daniel siger, okay, nu kommer vi til at arbejde med Klaus. Det er en ny afdeling for mig. Så igen, vi vil bare have et mål, og vi vil kigge på, hvordan det er det ser ud, som Daniel har også vist, om det er faktisk den der struktur, om vi ikke kan tage det, så begynder vi faktisk at lave rapporter. Eller evig nødt til at opdage dem, hvordan man opsamler dem. Eller måske hjælpe dem med nye programmer. Måske kan vi have sådan en power-app, for at gøre det automatisk, og med drop-down-menu, så man kan skrive på forskellige måder. Så vi prøver at tage det uden at sørge det. Så det vil sige, at det er meget forskelligt, hvor kigger jeg. Men det er bare at få et struktur på data, og så er det bare bagefter, og måske endnu mere spændende ting, som database house. Det er sådan en drøm, ikke? Så måske ligger det der, som Daniel siger. Hvert fald ikke endnu? Så nu, hvis jeg skal udvide det, så er det bare at få et struktur for de rapporter, så mennesker kan se det, og få et overblik. Det er der, hvor vi er nu. Fremtiden. Når alle sammen har rapporter, som de har brug for, så er det nok ledelsen, at vi bestemmer, hvad er vores næste strategiske step. Det er det, som jeg vil sige til det. Okay. Jeg tror, at hvis jeg understøtter lidt af det, så er det sådan, at det vi virkelig, virkelig gerne på kortbanen, så vil jeg gerne have nogle flere dashboards at arbejde ud fra, med overordnet kopimål. Altså, vi vil virkelig gerne have til at hænge den skærm, der er hverhen, sat opude i vores kontor. At for eksempel, når jeg er oppe til kampen, der er der dashboards i forhold til, hvor vi er henne, med det der tilskuerantal og hvad vi skal opnå. Simpelthen for, om vi kan få lavet et mere øget fokus på det, der er det vigtige. Altså, alle har sgu nok at lave hos os. Altså, det er ikke det. Men man kommer hurtigt til at løbe ud af en tangent på et eller andet, fordi, at der kan være en anden efterspørgsel, som måske gør, at man fjerner sig fra noget af det, der kan være svært. Og det skulle vi gerne bruge de data og de her forhold til. Det skulle vi i høj grad gerne bruge det til at træffe beslutninger på baggrunden af data. Et lille eksempel. Vi har jo vores fanzone. Der er vi åbent tre timer før, og så er vi selvfølgelig åbent i pausen. Og vi har egentlig brugt det samme mængde personale i pausen, fordi vi skulle være åbent. Nu er det faktisk, fordi vi har haft en praktikant, som har gået alle transaktionerne igennem. Så viser det sig fandme, at vi har kun har gennemsnitligt otte transaktioner i halvlegen. Det kan godt være, at vi skal ændre lidt på personalet. Sådan ville vi gerne bruge det. Når I kigger på tilskuerantallet og hvad I forventer, er det vigtigere for jer at kigge på det overordnede antal tilskuere? Grunden til, at vi har delt det op i kategorier for at skabe fællesskab omkring det område af tilskuerantal. Fordi vores tilskuerantal er mega vigtigt. Fodbold er for fællesskab. Fællesskaber er sjovere, når vi er mangle. Det vil vi rigtig gerne. Vi har så mange afledte parametre af, at det lykkes os at have et højt tilskuerantal. Derfor er det vigtigt, at alle afdelinger på en eller anden måde har en fornemmelse af, at de bidrager. Og så kan du sige om kommunikation og sådan nogle ting. Hvordan er det

I mäler, at de bidrager? Det er vi bare helt målrettet at forsøge at sige. For eksempel er vores kommunikation ud til partnere derfor, at de skal nå 720 i BDO-loungen. Hvis de ikke når 720 i BDO-loungen, så må vi jo kigge på, hvorfor opnår vi det ikke. Har vi ikke solgt pladserne? Jo, det havde vi faktisk. Vi kunne godt have nået 720, fordi pladserne var med. Så er der noget med fremmødet. Er fremmødet ikke godt nok? Nej, det er ligger skidt. Hvorfor er det ikke? Hvorfor har vi ikke fortalt historien godt nok om, hvad det er for en kamp, hvad det er for en event, hvad der kan vente. Så begynder vi at gå over og se på de der markedsføringsmæssige ting. Hvordan får de fortalt om det? Hvornår får de fortalt om det? Og det er derfor, vi har delt op. Så er det sådan, at man ikke kan sige, at det er opdelt i de her kategorier for, for at få et overordnet tilskuertal? Jamen også fordi strukturen på stadion, altså at vi har låst afsnit til de forskellige ting. Okay, det er klart. Hvad er det for en gruppe, der har den største udnyttelse, eller hvad er det for en gruppe, der har den mindste udnyttelse? Dem, der ligger højest, det er, hvad skal man sige, det er... Jamen prøv at høre, den er egentlig. Den er så simpel. Jo mere du betaler for dit produkt, jo mere kommer du til. Men I har også begrænset plads på de forskellige grupper? Ja, men jo, vi har udsolgt nogle andre. Men jo mere du betaler, jo mere værdi der er på dit produkt, desto højere er dit fremmøde. Så derfor er dem, der har bedst fremmøde, det er dem, der har købt hospitality. Det er omkring 1250 kroner hver billet koster. Så dem, der har højest fremmøde på. Og så er det de sæsonkortholdere, som sidder midt på over i øst, på de dyrere pladser. Så der er forskel på, om du betaler 39 kroner for en barnebillett på syd, og så du betaler 189 for en voksenbillett, med menu over på øst. Så bruger du altså den der til 189 og mere. Jeg kunne godt tænke mig at høre det der med sponsoraftalerne, hvor strategisk vigtigt det er for jer, den data I kigger på. Og det jeg mener med det er, hvis I nu ser en virksomhed, der har haft flere lavt fremmødeprocent, er det sådan, I også kan have enighed om, hvad det betyder, Bruger i det så til at stoppe sponsorrattet til næste sæson? Eller bruger I det data til det? Ja. Jamen det er øvrigt godt. Det jo det vi i høj grad gør, fordi... Lad os nu bare sige, at der er en virksomhed, som har 10 billetter til hospitalet til hver eneste gang. På det tidspunkt har de lavet en god, stor aftale, og det er egentlig fedt. Vi får de penge i kassen, der skal. Problemet er jo bare, at hvis vi gang på gang kan se, at de her 10 billetter ikke bliver udnyttet, så får de jo ikke det udbytte ud af netværket, som de skal have. Så der skal vi ud og have fat i virksomheden. Hvad er det altså? Er det ikke fedt nok, at vi har en forkert approach på, eller skal vi have flyttet? I skyboxen skal vi have lavet om, at I har to skyboxer hver eneste gang. Så det skal vi ud og have, og vi skal gøre det før, at vi sidder ude til kundemødet, for vi forlænger sgu ikke sponsorerrattet, fordi vi bruger det ikke. Der skal vi have fat i det. Det var egentlig det første, du lavede. Det var hele sådan, at vores sælgere har mulighed for at følge det løbende. Okay. Hvis nu vi laver sådan en fin dashboard til dig, og du står og kigger på det, hvad er det så, du gerne vil se? Hvad jeg gerne vil se? Ja. For at hjælpe dig i din hverdag. Jeg vil egentlig gerne dele den og sige, at på et dashboard skal det være meget, hvad der er relevant for kontoret. For kontoret. Det vil sige, at det er meget op på vores KPI-mål, hvor vi er henne i forhold til vores KPI-mål, som er det overordnede. Jeg tror, vi havde en snak, Olga, i går med, hvad det hedder, i forhold til Clock to Mejser, hvor vi sådan lidt ligesom siger, at Olga har lavet rigtig, rigtig mange opsummeringer på forskellige produkter, der er herude på stadion. Men det er i bund og grund at vi gerne vil have dem på fire, hvad det hedder, nøgletal. Og så vil Pernille, som er min kollega, der sidder bare på hospitalet i stedet for, hun skal have, altså hun skal kunne se nede på, hvor mange tæpper, altså er der solgt for eksempel ude på Bane 3 tæpper og sådan nogle ting. Men det er der ikke så mange andre, der behøver at se. Så vi skal have en klik-funktion, der gør, at man kan komme et niveau længere ned. Så du har ikke brug for at vide, hvor mange skal jeg lige nu have af bemanding på den her scene eller tribune? Det skulle jeg jo gerne gøre ved at gå et niveau længere ned. Men for eksempel de der mål, jeg satte op derpå. Det kunne være mega fedt, at vi kunne vise det, for eksempel op til en Midtjylland kamp, hvor vi sætter så benhårde mål på de forskellige ting. Der kunne det være fedt, at vi kunne se forløbet derop til. Nu snakker du om, at I ligger på, hvad var det, 6200 tilskuere typisk for kampe pt. Hvor tætpakket skal stadion være, før det er en fed oplevelse? Jeg tænker, at der må være en kritisk masse for, hvornår det ser dødt og kedeligt ud, fordi der er for få folk. Hvornår rammer man den grænse? Lige nu ligger vi jo så op i sjovt nok på de der 62 procent. Hvis I siger, at jeg tror, at jeg rammer 5.000, så er det en ret fed stadion. Så det skal være halvfyldt over det hele? Ja. Kan du så nævne årsagerne til at der kun er det her 65%'s fremmøde? Fremmøde? Ja. For eksempel Aula aktivering, hvor du ikke betaler noget for billetten. Det ved du selv, hvis du har adgang til at hente en billet. Jeg tror fandme også, at min lilleøster og ens kæreste og deres to børn, de vil nok også godt med. Fordi du har adgang, det koster da ikke noget. Så booker du 8 billetter i stedet for 4. Men din lilleøster, hun gider sgu ikke med alligevel. Så bliver jo fremmødet jo i de mindre. Og det er derfor, at aktiveringerne og det der. Vi snakker ekstremt meget om, at vi bare skal putte et lille, lille beløb på. Det kan være, at alle aktiveringerne i fremtiden godt kunne være, at når du har hentet billetten, så får du en fransk hotdog eller en sodavand. Således at man stadigvæk i principippet betaler 0 kroner, men du betaler noget for billetten. Fordi har vi bare haft pungen op, så har vi altså en helt anden motivation til at bruge den. Jeg tænker, har I sådan en

eller anden type stjerne-spiller på holdet, som du tiltrækker tilskuere eller sådan noget? Min far snakker meget om fodbold, og så tænker jeg, at ham der Cho fra Midtjylland. Han siger, at... Cho, vi kender sgu jo ikke nogen fra Midtjylland. Han siger, at fordi han er sydkoreaner så tiltrækker han sydkoreanere, så kommer vi til at se ham, når han stiller, fordi vi vil se ham. Har I sådan noget, I Viborg? Er det noget, I tænker over? Nej, altså, jeg tror, at det er noget, der betyder rigtig meget. Det er lokalforandring. Så når vi har sådan en Anton Gaai og sådan nogle ting, som folk kan forene sig med, det synes de, det er fedt. Men jeg er heller ikke i tvivl om, altså... Får vi virkelig sådan en stjernespiller, vi kan høre ind Lige nu synes jeg, at vi har en Thomas Jørgensen, der er på vej. Ham der bliver stor, så altså... Ham vil folk gerne ind og se, bare for ham. Ellers havde jeg ikke sådan lige... Ja, altså enig. Der er jo noget med at det er naboenes dreng, som løbet rundt og spillet fodbold med ens egen søn. Altså, det er der noget over. Så er der noget over de der meget spændende, dygtige spillere, hvor Thomas er et godt eksempel. Så er der nogen, der bare godt kan lide at se nogen, der kan noget flair, og drible.

.5.5 Daniel & praktikanterne

Der er ingen problemer med tiden til det. Som udgangspunkt tager jeg jer bagefter lige med rundt kort og viser stadion, og så runder vi dagen af efter det. I er velkomne til at skyde med alle de spørgsmål, I har. Jeg står lige op, fordi jeg ikke har sovet så meget i nat, bare for at holde mig i gang. Men lad os endelig tage alle de spørgsmål, I har. Vi har tænkt os at høre, om I har nogle fastdefinerede roller inden for alt det her data. Hvem er det, der står for at definere den måde, man snakker om tingene på? Hvem sørger for, at alt bliver testet korrekt? Hvem sætter reglerne for at lave de fælles standarder? Det er et vanvittigt godt spørgsmål, og der tror jeg ikke helt, at vi er der, hvor vi fast kan sige, at det er sådan her. Jeg tror, det man kan sige er, at hvis dataafdelingen er involveret i et projekt, så er det også os, der står for det. Men vi har jo også projekter, som dataafdelingen gerne vil understøtte, men ikke kan i dag, fordi vi ikke har bemandingen til det. Grundlæggende, hvis vi er inde over et projekt, for eksempel sammen med Palle, så tager vi ansvaret. Og hvis vi ikke kan stå for indtastningsarbejdet, af den ene eller anden grund, for det kan der jo godt være nogle gange, at vi ikke sidder som dem, der manuelt tester tingene, så sørger vi for, at dem, der sidder med indtastningsarbejdet, får den nødvendige viden. Så vi sikrer os, at det bliver gjort på den måde, som vi aftaler. Det tror jeg er den bedste måde at forklare det på. Når der er nogle andre afdelinger, der skal bruge noget specifikt data, ved de så altid, hvem det er, de skal gå til? Er det altid bare jer? Der tror jeg, det er vigtigt, at alle svarer. For jeg kan jo godt sige ja, men jeg tror også, at folk ved, hvem det er, der sidder med for eksempel billeddata. Men det må I gerne supplere med. Ud fra det, jeg har set, så handler det meget om, at jeg faktisk ikke har oplevet nogen, der kommer og spørger efter data, medmindre det har været i forbindelse med et projekt. Altså at det ligesom er et projekt sammen mellem slutbruger og dataafdeling. Og så ved de jo, hvem de skal gå til – nemlig den, der sidder med projektet. Det er den person, man går til. Jeg ved ikke, om der nogensinde er nogen, der bare kommer og spørger efter data, der ikke er i forbindelse med et projekt. Dataafdelingen er heller ikke så stor, så det er ikke så svært at finde ud af, hvem man skal gå til, hvis man har brug for data. Det er også meget det, jeg forholder mig til. Det er meget projektbaseret arbejde. Vi er i en afdeling, så begynder vi at arbejde med noget, og hvis de har brug for aktuelle data eller har nye ønsker, så går de ikke til marketingafdelingen eller nogen andre, men til Daniel og Maja. Vi kan godt lide, at det er struktureret arbejde. Daniel nævnte også, at vi mødes med dem. Vi er til stede og involverede. Vi har også flere opgaver, som vi synes er spændende, men også nødvendige at tage fat i for at løfte vores datamål. Det betyder, at vi godt kan lide, at det bliver struktureret. Hvis de har lyst til at snakke om noget, opfordrer vi dem til at booke et møde, hvor vi forklarer, hvordan det fungerer. Det kan være, at vi optimerer Daniels PowerPoint, som han plejer at vise, og så forklarer vi det. Vi forklarer også, at hvis de ændrer noget dagen efter, så er det måske ikke en god idé, fordi vi så er nødt til at starte forfra. Så vi er nødt til at forklare dem, at det tager tid, og at de også skal være forberedte på det. Vi holder først strukturen. Hvis I så kommer med noget nyt, for eksempel nogle idéer, tager I så selv fat i de relevante personer og siger, at der er nogen, der skal høre på den nye struktur? Hvordan gør I det i dag? Der indkalder vi den afdeling, vi samarbejder med. For eksempel kan de sige, at de gerne vil have et dashboard, der skal se sådan og sådan ud, og så kigger vi i datasættene. Vi siger også meget tydeligt, at det her data kommer aldrig nogensinde til at kunne bruges på den måde, de måske forestiller sig. Det kræver, at vi gør lige præcis de her ting. Det har vi valgt at implementere, og det kommer til at påvirke jeres arbejde på den og den måde. Så I skal gøre sådan her og sådan her. Og så laver vi en guide til dem. Det er typisk en PowerPoint, hvor der står: Du plejer at gøre sådan her, nu skal du gøre sådan her. Sådan så vi er enige om det. Og vi sørger altid for at sende det på mail, så vi har dokumentation for, hvad vi har aftalt. Hvis der så opstår fejl, kan vi se, at det var fordi, der blev tastet noget andet end det, vi havde aftalt. Hele afdelingen er

stakeholder på det. Det forsøger vi i hvert fald. Og ja, det er også os, der ejer arbejdsprocesserne og opdaterer dem. Hvordan er kulturen hos jer? Er det jer som dataafdeling, der beslutter, at nu sætter vi noget i gang, eller er det efterspørgselsdrevet fra de enkelte afdelinger, eller er det en blanding? I starten var det meget sådan, at vi selv prøvede at fremlægge nogle problemstillinger i forhold til medarbejdere og forretningsgange. Vi startede i én afdeling og sagde: Hvad kan vi gøre her? Men i takt med at slutbrugere som Palle begynder at se, hvad man faktisk kan, så opstår der en efterspørgsel på mere og mere, og så spreder det sig. Når Palle har et værktøj, som nogen i marketing synes er fedt, så vil de også gerne have noget lignende. Så i starten var det meget drevet af dataafdelingen. I dag handler det mere om, at vi prøver at følge med og nå de projekter, der kommer ind, fordi der er rigeligt at lave. Folk oplever, at noget, de før brugte fire timer på hver dag, nu kan løses på to minutter. Det vil man gerne have. Og oven i købet ser det pænere ud. Der er en enorm stor nysgerrighed omkring data. Data er dog også et lidt diffust begreb. Mange siger bare "arbejder du med data?", uden helt at vide, hvad det betyder. Den dashboard- og rapporteringsvej, vi er på nu, giver rigtig god mening i forhold til, hvor vi er. Det handler om at få struktur på den data, vi allerede har. Det spreder sig som ringe i vandet. Det kommer både fra medarbejderne og fra ledelsen. Der er interesse hele vejen rundt. I forhold til bemanding mangler vi folk med de samme kompetencer som dem, der sidder her. Folk der kan arbejde med datastruktur og rettighedsdata, og som samtidig kan arbejde tæt sammen med stakeholders. Hvis vi var fuldstændig statiske i to år og ikke voksede, så tror jeg, at fem fuldtidsansatte kunne understøtte behovet. Det er ledelsen klar over. Hvis man skulle vurdere virksomhedens datamodenhed fra 1 til 10, er det svært, fordi det er meget afdelingsbaseret. Nogle afdelinger ligger lavt, andre ligger højt. Teknologi- og sportsdelen er generelt meget bevidst om data. Administrativt virker der også til at være stor åbenhed og vilje til forandringer. Hvis man sammenligner med de bedste IT-virksomheder, så er vi langt fra. Der er vi nok under 5. Men sammenlignet med mange konkurrenter i fodboldverdenen er vi længere fremme. Derfor lander vi måske realistisk på 7-8, afhængigt af sammenligningsgrundlaget. Modstand opstår primært i forbindelse med forandringer. "Vi plejer at gøre sådan her" eller "vi kan ikke se værdien". Det tekniske forsøger vi at fjerne som barriere, så modstanden ikke kan forklares med systemerne. Det handler i stedet om at forklare værdien. Vi prøver at være meget åbne omkring, hvor lang tid ting tager. Hvis nogen siger, at de gerne vil have noget klar til søndag, så siger vi ærligt, at det ikke er realistisk. Der er stor lyst til, at ting skal være færdige i går, men vi prøver at tage folk med på rejsen. Der er forskel på afdelinger. I nogle afdelinger kan man hurtigt gå i gang med analysemodeller, fordi værdien er tydelig. I andre afdelinger kræver det mere arbejde med datakultur og forståelse, før man kan komme videre. Ledelsen spiller også en vigtig rolle. De går i høj grad forest, og det er vigtigt for at skabe opbakning. Meget af arbejdet handler om at visualisere problemer: at vise, at tre medarbejdere bruger fire timer om dagen på noget, der burde være automatiseret. Når vi sætter grænser for, hvad vi kan nå, oplever vi generelt forståelse. Det skyldes, at vi har taget folk med og forklaret dem processen. Vi viser dem konkret: Det her er det, I beder om, det kræver det her, og det tager tid. Hvis I ændrer noget undervejs, skal vi starte forfra. Vi forventer, at kommunikationen bliver nemmere over tid, men det er en kontinuerlig proces. Når nye medarbejdere starter, sørger vi for, at de bliver taget ordentligt med. Vi arbejder meget projektorienteret. Prioriteringen ligger i sidste ende hos direktionen, men dataafdelingen har et stort ansvar og beslutningsrum. Udviklingsprojekter prioriteres, mens driftsprojekter søges automatiseret. Et projekt som billetdata er i dag primært drift. Data lagres på forskellige måder afhængigt af systemerne. Vi arbejder blandt andet med API-løsninger og Power BI. Ambitionen er i højere grad at øje data selv og ikke være afhængig af leverandørers rapporter. Hverdagen består af både drift, udvikling og stakeholder management. Vi vurderer altid, om det kan betale sig at automatisere en opgave, eller om den er hurtigere at udføre manuelt. Der findes ikke generelle retningslinjer for, hvordan alle medarbejdere arbejder med data. Hver afdeling har sin egen datakultur, ofte præget af lederens fokus. Over tid håber vi at skabe mere fælles struktur. På kampdage kan nogle registreringer godt blive nedprioriteret, fordi driften er vigtigst. Det vigtigste er, at der står personale i boderne og kan sælge. Adgang til rapporter gives via ledelsesgodkendelse. Ikke alle har licenser. Nogle skal kunne arbejde med data, andre skal kun kunne se rapporter. Der har været udfordringer med licenser, deling og prøveperioder. Teams bruges primært til møder. Intern kommunikation foregår mest via Outlook, SMS og WhatsApp. Dataafdelingen er rådgivende i forhold til systemvalg, men bestemmer ikke, hvilke systemer afdelingerne bruger. Mange systemer øger kompleksiteten, især fordi mange medarbejdere er autodidakte og har deres egne løsninger. Organisationen har oplevet stor vækst over de seneste år, hvilket har gjort behovet for en dataafdeling tydeligt. Tidligere kunne ting klares uformelt, men det kan man ikke længere. Fremadrettet forventes fortsat vækst, men i et mere moderat tempo. Til sidst vises der rundt på stadion, og optagelsen afsluttes.