

Prosjektoppgave - SOK-1005

Kandidatnr.: 72

05 06 2022

```
library(tidyverse)
library(readr)
library(lubridate)
```

Oppgave 1

Jeg slår sammen de 6 datasettene til ett stort datasett ved hjelp av funksjoner union og merge.

```
df1 <- read_csv("C:/Users/mgmal/Desktop/Prosjekt/AppWichStoreAttributes.csv")
df2 <- read_csv("C:/Users/mgmal/Desktop/Prosjekt/county_crime.csv")
df3 <- read_csv("C:/Users/mgmal/Desktop/Prosjekt/county_demographic.csv")
df4 <- read_csv("C:/Users/mgmal/Desktop/Prosjekt/county_employment.csv")
df5 <- read_csv("C:/Users/mgmal/Desktop/Prosjekt/weekly_sales_10stores.csv")
df6 <- read_csv("C:/Users/mgmal/Desktop/Prosjekt/weekly_weather.csv")

# str(df1) # [10 x 14]
# str(df2) # [6 x 12]
# str(df3) # [6 x 14]
# str(df4) # [6 x 5]
# str(df5) # [79,459 x 17] # weekly_sales er det største datasettet
# str(df6) # [270 x 16]

df1 <- rename(df1, County_Name = Store_County)
df1 <- rename(df1, Store_num = Store_Num)
join1 <- merge(df5, df1) # 79459 obs 30 variables
join2 <- merge(join1, df2) # 79459 obs 41 variables
join3 <- merge(join2, df3) # 79459 obs 54 variables
join4 <- merge(join3, df4) # 79459 obs 58 variables
df6 <- rename(df6, Store_Weather_Station = Weather_Station)
final_df <- union_all(join4, df6)
final_df$Date <- mdy(final_df$Date)
final_df <- final_df %>% filter(!is.na(Description))
# str(final_df) # 79459 obs of 73 variables
```

Det endelige datasettet final_df inneholder 79,459 rader/observasjoner og 73 kolonner/variabler. Datasettet weekly_sales (df5) var det største datasettet og inneholdt 79,459 rader og 17 kolonner. Det ser ut til at sammenslåingen har latt seg gjennomføre. Jeg sammenligner for ordens skyld noen sentrale variabler:

```
# Sammenligner min final_df med det originale weekly_sales datasettet:  
sum(df5$Profit)
```

```
## [1] 3367938
```

```
sum(final_df$Profit, na.rm=T)
```

```
## [1] 3367938
```

```
# 502 forskjellige varer/produkter:  
count(unique(final_df[c("Description"))]))
```

```
##      n  
## 1 502
```

```
count(unique(df5[c("Description"))]))
```

```
## # A tibble: 1 x 1  
##      n  
##   <int>  
## 1 502
```

Alt ser ut til å være OK og jeg går videre med min analyse.

Oppgave 2

En ukentlig salgsrapport kan eksempelvis inneholde følgende elementer:

```
# Funksjonen filter(Date=="2012-04-01") henter ut data for perioden 2012-04-01 tom 2012-04-07.  
# Altså for hele uken.  
# Funksjonen filter(Date=="2012-04-08") henter ut data for perioden 2012-04-08 tom 2012-04-14.  
  
# De mest lønnsomme produktene, utsalgssted 2, periode 2012-04-01 tom 2012-04-07 (1 uke)  
final_df %>% filter(Date=="2012-04-01") %>%  
  filter(Store_num==2) %>%  
  group_by>Description) %>%  
  summariseProfit = sum(Profit)) %>%  
  arrange(desc(Profit))  
  
## # A tibble: 174 x 2  
##   Description          Profit  
##   <chr>                <dbl>  
## 1 REGULAR SAVORY TURKEY      809.  
## 2 REGULAR CHICKEN BACON RANCH    770.  
## 3 REGULAR HONEY MUSTARD HAM     559.  
## 4 MINI SAVORY TURKEY        501.  
## 5 CHIPS                  458.  
## 6 REGULAR CLASSIC TUNA       452.  
## 7 21OZ DRINK                 420.  
## 8 REGULAR BMT                  400  
## 9 REGULAR SPICY ITALIAN      395.  
## 10 REGULAR CHICKEN TERIYAKI    356.  
## # ... with 164 more rows  
  
# De minst lønnsomme produktene  
# Kan evt. ekskludere produkter som inneholder ordene "free" og "reward"  
final_df %>% filter(Date=="2012-04-01") %>%  
  filter(Store_num==2) %>%  
  group_by>Description) %>%  
  summariseProfit = sum(Profit)) %>%  
  arrange(Profit)  
  
## # A tibble: 174 x 2  
##   Description          Profit  
##   <chr>                <dbl>  
## 1 REGULAR SUB OR SALAD, REWARD -539.  
## 2 MINI SUB, REWARDS        -263.  
## 3 BOGO MINI SUB           -205.  
## 4 VAL MEAL 1 MINI MTBALL CHIP -84.2  
## 5 FREE MINI SUB            -58.2  
## 6 FREE REGULAR              -31  
## 7 21 OZ DRINK, REWARDS      -24  
## 8 CHIPS, REWARDS            -23.1  
## 9 VAL MEAL 3 MINI TURKEY CHIP -10.6  
## 10 FREE COOKIE                 -9.25  
## # ... with 164 more rows
```

```
# De mest solgte produktene
head(final_df %>% filter(Date=="2012-04-01") %>%
  group_by(Description) %>%
  summarise(Sold = sum(Sold)) %>%
  arrange(desc(Sold)), 3)
```

```
## # A tibble: 3 x 2
##   Description           Sold
##   <chr>                  <dbl>
## 1 CHIPS                 3911
## 2 21OZ DRINK             2479
## 3 VAL MEAL 1 MINI MTBALL CHIP  2076
```

```
# Produktbeskrivelse og profitt for en spesifikk dato
store_2 <- final_df %>% select>Description, Profit, Date, Store_num) %>%
  filter>Date=="2012-04-01", Store_num==2)
head(store_2)
```

```
##          Description Profit      Date Store_num
## 1 REGULAR FLATBREAD SPICY IT    7.00 2012-04-01      2
## 2 SUNRISE MELT REGULAR FLATBREAD  4.24 2012-04-01      2
## 3                      REGULAR BLT 237.77 2012-04-01      2
## 4      MINI SAVORY TURKEY BT & HAM  98.41 2012-04-01      2
## 5 REGULAR FLATBREAD STEAK CHEESE 24.00 2012-04-01      2
## 6              MINI SPICY ITALIAN 133.51 2012-04-01      2
```

Nøyaktig samme prosedyre kan brukes for andre variabler som f.eks kostnad, margin, pris, osv.

Slik innsikt kan være nyttig hvis selskapet ønsker å (for eksempel) fokusere på de mest lønnsomme produktene, kutte i varesortimentet, kutte kostnader eller se potensielle i andre, mindre populære produkter.

Datasattet inneholder 502 forskjellige produkter, men noen produkter ligner på hverandre og kan gruppertes sammen. Eksempelvis kan alle produkter som inneholder ordet "pizza" i produktbeskrivelsen gruppertes i en kategori/variabel, og alle produkter som inneholder "burger" gruppertes i en annen. På denne måten kan en se hvordan de ulike produktkategoriene "rangerer" i forhold til hverandre.

```
# Binning
pizza <- final_df %>% filter(grepl('PIZZA', Description)) %>%
  select>Description, Profit, Date, Day, Store_num) %>%
  mutate>Description = ifelse(grepl('PIZZA', Description), 'Pizza')

burger <- final_df %>% filter(grepl('BURGER', Description)) %>%
  select>Description, Profit, Date, Day, Store_num) %>%
  mutate>Description = ifelse(grepl('BURGER', Description), 'Burger')

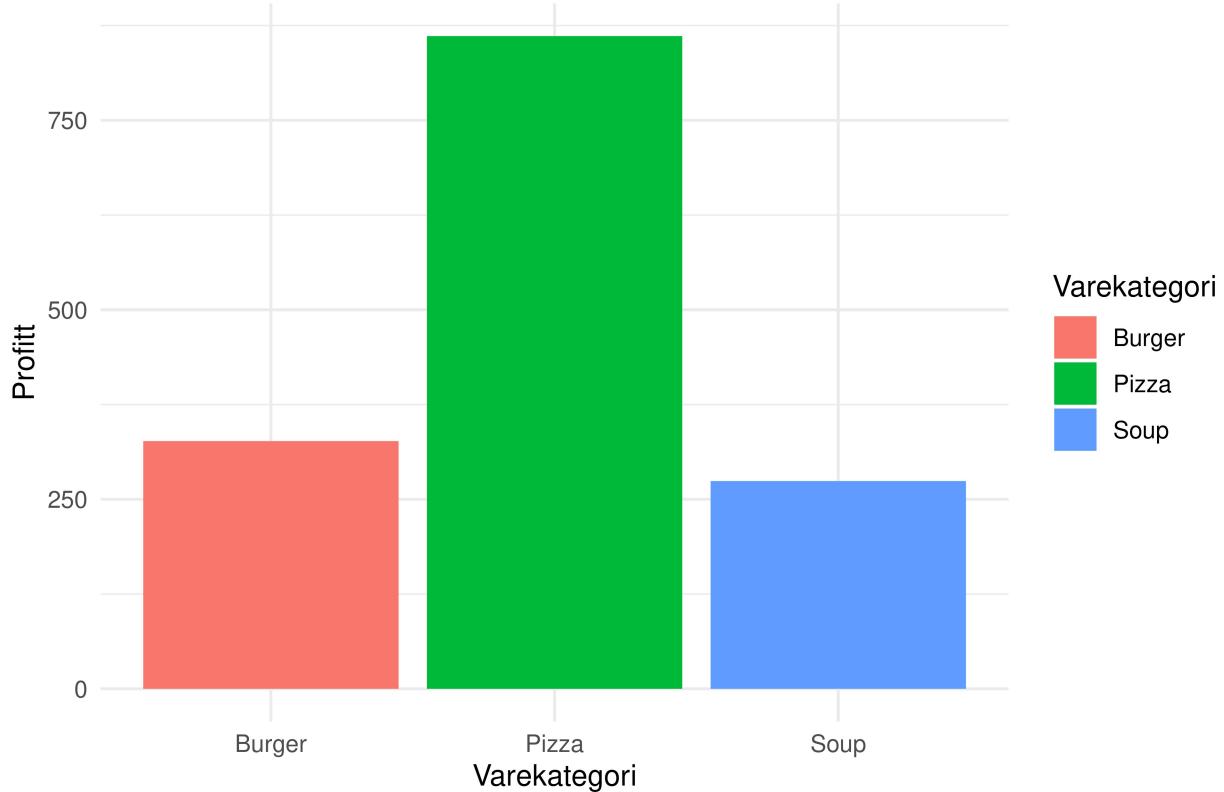
soup <- final_df %>% filter(grepl('SOUP', Description)) %>%
  select>Description, Profit, Date, Day, Store_num) %>%
  mutate>Description = ifelse(grepl('SOUP', Description), 'Soup')
```

```

df_barplot2 <- bind_rows(pizza, burger, soup)
df_barplot2 <- df_barplot2 %>% filter(Date=="2012-04-01")
df_barplot2 <- setNames(aggregate(df_barplot2$Profit, by=list(Varekategori=df_barplot2$Description),
                                    FUN=sum),c("Varekategori", "Profitt"))
ggplot(df_barplot2, aes(x=Varekategori, y=Profitt, fill=Varekategori)) + geom_bar(stat="identity") +
  theme_minimal() + labs(title = "Profitt per varekategori, 1 uke")

```

Profitt per varekategori, 1 uke



For perioden 2012-04-01 til og med 2012-04-07 er kolonnen "dag" lik 1 for hele uken, noe som gjør det umulig å analysere endringer fra dag til dag for en spesifikk uke. Det hadde vært interessant å se hvordan salg og profit utvikler seg over tid (varierer med ukedager), men dette er altså ikke mulig på grunn av måten dataene er organisert/formatert på. Når er det best å markedsføre våre produkter? Selskapet som eier dataene bør se om de kan skaffe oss denne typen informasjon hvis denne typen analyse er av interesse.

Det hadde også vært interessant å se på hvilke produkter blir solgt sammen med andre produkter og hvilke produktkombinasjoner er de mest populære. Gitt denne kunnskapen, kan selskapet forsøke å "bundle" disse produktene for å oppnå større profit. Relaterte og mindre populære produkter kan ha et stort potensiale for selskapet når man bruker denne typen kryssalg. Det viser seg at variablen INV_NUMBER inneholder 1 vare per faktura og dette ser ut til å være standarden. Datasettet kan være ukomplett og oppdragsgiveren bør fremlegge et alternativt datasett, hvis så er tilfellet. Samtidig er det ikke utelukket at feilen ligger hos analytikeren og at datasettet ikke var undersøkt grundig nok. Oppdragsgiveren bør kontaktes for å få oppklart dette.

```
# unique(final_df[c("INV_NUMBER")]) # Alle fakturaer
# unique(final_df %>% filter(INV_NUMBER==40074) %>% select>Description()
# unique(final_df %>% filter(INV_NUMBER==11064) %>% select>Description()
# unique(final_df %>% filter(INV_NUMBER==7134) %>% select>Description())
# unique(final_df %>% filter(INV_NUMBER==14825) %>% select>Description())
# unique(final_df %>% filter(INV_NUMBER==4096) %>% select>Description())
# unique(final_df %>% filter(INV_NUMBER==1909) %>% select>Description())
# unique(final_df %>% filter(INV_NUMBER==4114) %>% select>Description())
# unique(final_df %>% filter(INV_NUMBER==35019) %>% select>Description())
# unique(final_df %>% filter(INV_NUMBER==3642) %>% select>Description())
unique(final_df %>% filter(INV_NUMBER==14391) %>% select>Description())

##          Description
## 1 GARDEN BURGER SALAD
```

Oppgave 3

En månedlig salgsrapport på aggregert nivå kan inneholde følgende:

```
# Utsalgssteder med mest og minst profit, Mai måned, 2012:
final_df %>% filter(Year==2012) %>%
  filter(Month==5) %>%
  group_by(Store_num) %>%
  summarise

```

```
# De mest lønnsomme produktene:
final_df %>% filter(Year==2012) %>%
  filter(Month==5) %>%
  group_by>Description) %>%
  summarise(Profit = sum(Profit)) %>%
  arrange(desc(Profit))
```

```
## # A tibble: 368 x 2
##   Description      Profit
##   <chr>            <dbl>
## 1 REGULAR SAVORY TURKEY    19229.
## 2 REGULAR HONEY MUSTARD HAM 15018.
## 3 REGULAR CHICKEN TERIYAKI  13155.
## 4 REGULAR CHICKEN BACON RANCH 12585.
## 5 CHIPS              11316.
## 6 21OZ DRINK         11221.
## 7 REGULAR BMT          10380.
## 8 MINI SAVORY TURKEY    10042.
## 9 REGULAR COLD CUT MUSHROOM 9834.
## 10 REGULAR ROASTED CHICKEN 8076.
## # ... with 358 more rows
```

```
# De minst lønnsomme produktene:
final_df %>% filter(Year==2012) %>%
  filter(Month==5) %>%
  group_by>Description) %>%
  summarise(Profit = sum(Profit)) %>%
  arrange(Profit)
```

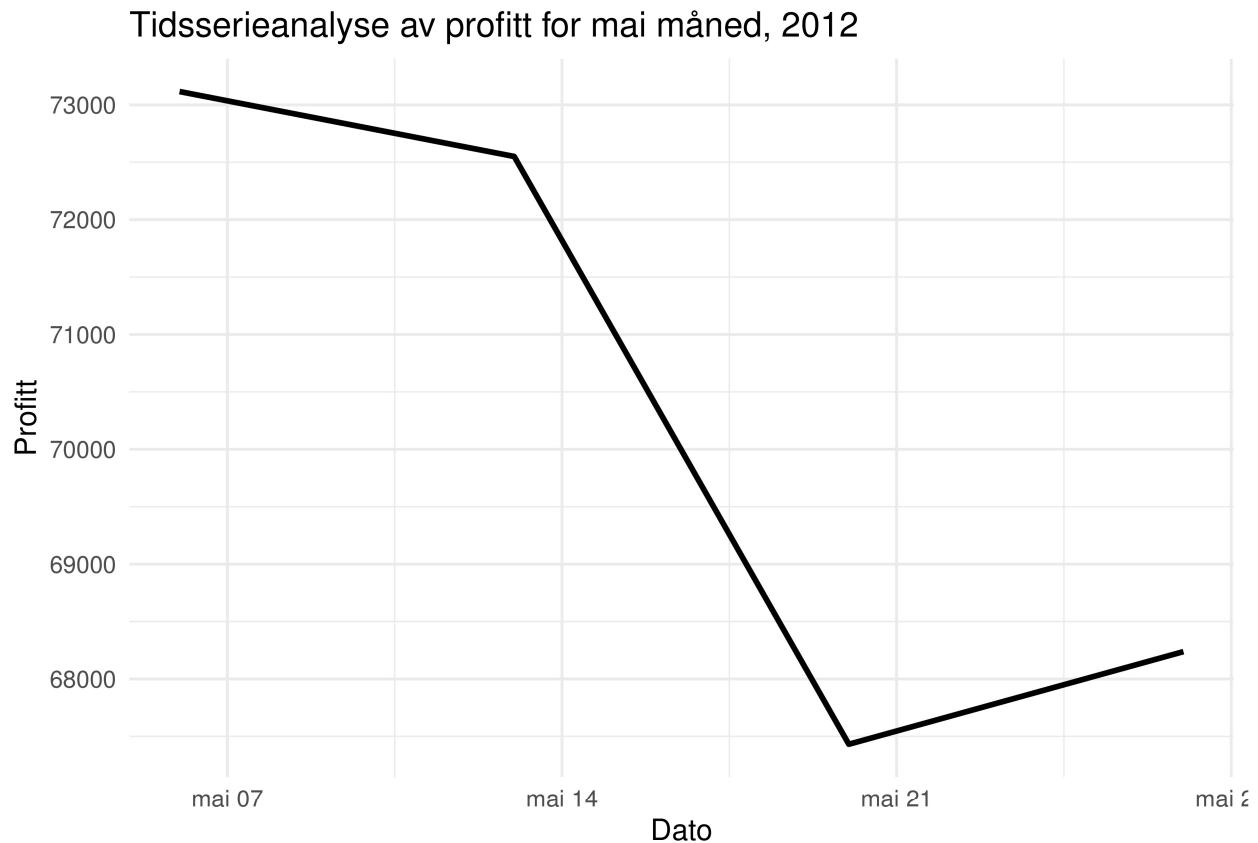
```
## # A tibble: 368 x 2
##   Description      Profit
##   <chr>            <dbl>
## 1 REGULAR SUB OR SALAD, REWARD -11386.
## 2 MINI SUB, REWARDS        -3861.
## 3 VAL MEAL 1 MINI MTBALL CHIP -1824.
## 4 FREE ORDER             -1118.
## 5 21 OZ DRINK, REWARDS     -454.
## 6 CHIPS, REWARDS          -393.
## 7 FREE MINI SUB           -234.
## 8 VAL MEAL 3 MINI TURKEY CHIP -226.
## 9 1 COOKIE, REWARDS       -200
## 10 FREE REGULAR          -151
## # ... with 358 more rows
```

```
# De mest solgte produktene:  
final_df %>% filter(Year==2012) %>%  
  filter(Month==5) %>%  
  group_by>Description() %>%  
  summarise(Sold = sum(Sold)) %>%  
  arrange(desc(Sold))
```

```
## # A tibble: 368 x 2  
##   Description      Sold  
##   <chr>            <dbl>  
## 1 CHIPS             16709  
## 2 21OZ DRINK        10522  
## 3 VAL MEAL 1 MINI MTBALL CHIP  9227  
## 4 1 COOKIE          6141  
## 5 30OZ DRINK         5007  
## 6 REGULAR HONEY MUSTARD HAM  5000  
## 7 REGULAR SAVORY TURKEY    4008  
## 8 40OZ BEVERAGE       3323  
## 9 REGULAR COLD CUT MUSHROOM 3238  
## 10 3 COOKIES         3127  
## # ... with 358 more rows
```

Vi kan se på utviklingen av diverse variabler over tid. For eksempel, for profitt:

```
# Tidsserieanalyse av profitt for mai måned, 2012:  
oppg3plot1 <- final_df %>% filter(Year==2012) %>% filter(Month==5)  
unique(oppg3plot1[c("Date")]) # 4 uker  
  
## Date  
## 1 2012-05-06  
## 148 2012-05-13  
## 286 2012-05-20  
## 431 2012-05-27  
  
oppg3plot1 <- aggregate(oppg3plot1$Profit, by=list(Dato=oppg3plot1$Date), FUN=sum)  
oppg3plot1 <- rename(oppg3plot1, Profitt = x)  
# Jeg kunne ha forandret variabelnavnene i final_df i stedet for å slippe å gjøre det flere ganger  
ggplot(oppg3plot1, aes(x = Dato, y = Profitt)) + theme_minimal() +  
  geom_line(aes(), size = 1) +  
  labs(title = "Tidsserieanalyse av profitt for mai måned, 2012")
```



```

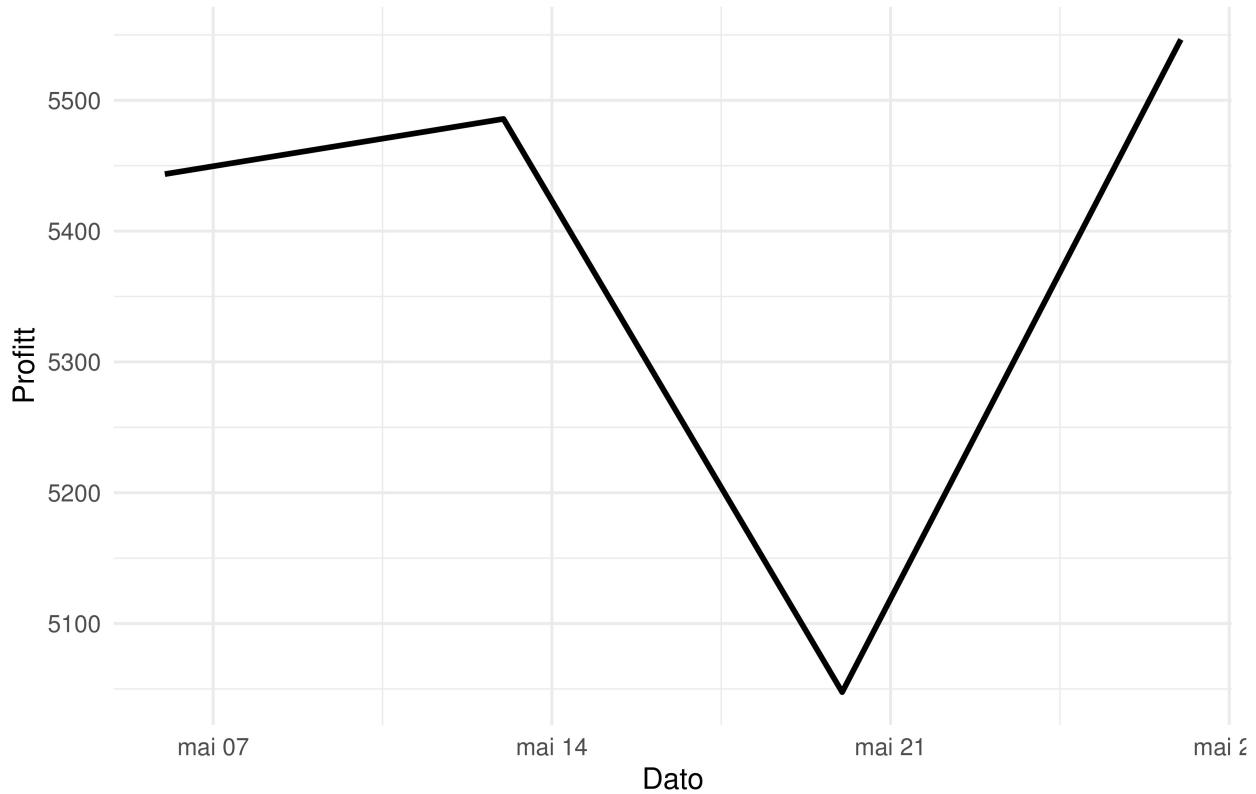
# Tidsserieanalyse av profitt for mai måned, 2012 for utsalgssted 3:
oppg3plot2 <- final_df %>% filter(Year==2012) %>% filter(Month==5) %>% filter(Store_num==5)
unique(oppg3plot2[c("Date")]) # 4 uker

##          Date
## 1 2012-05-06
## 163 2012-05-13
## 334 2012-05-20
## 502 2012-05-27

oppg3plot2 <- aggregate(oppg3plot2$Profit, by=list(Dato=oppg3plot2$Date), FUN=sum)
oppg3plot2 <- rename(oppg3plot2, Profitt = x)
ggplot(oppg3plot2, aes(x = Dato, y = Profitt)) + theme_minimal() +
  geom_line(aes(), size = 1) +
  labs(title = "Tidsserieanalyse av profitt for mai måned 2012 for utsalgssted 3")

```

Tidsserieanalyse av profitt for mai måned 2012 for utsalgssted 3



Vi kan lage prognosenter for neste måned ved hjelp av regresjonsanalyse (justert for sesongvariasjoner). Datasettet inneholder mange variabler som kan brukes som uavhengige variabler i predikasjon av f.eks profitt. Dette kan være variabler som Month, Store_Name samt temperaturdata og supplerende makroøkonomisk data.

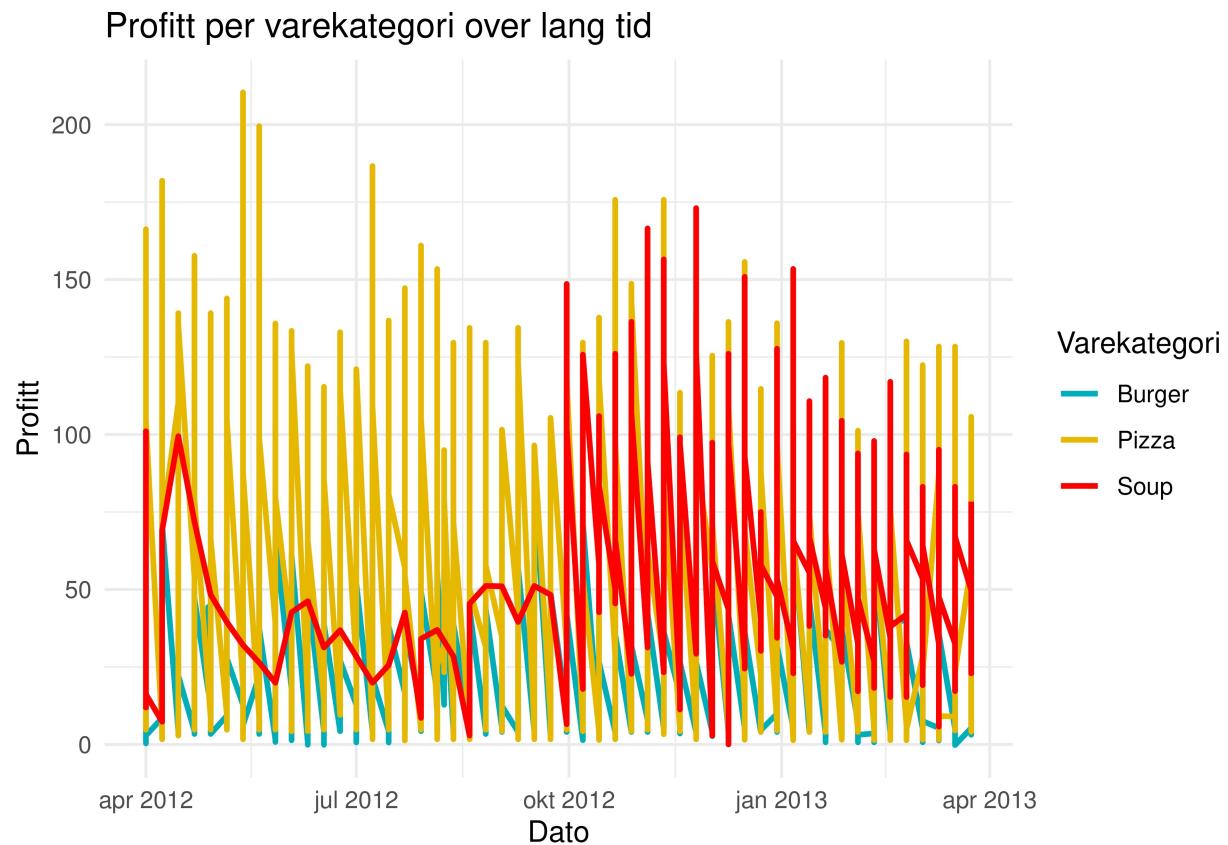
```
lm1 <- lm(final_df$Profit ~ final_df$Month + final_df$Store_num)
summary(lm1)
```

```
##
## Call:
## lm(formula = final_df$Profit ~ final_df$Month + final_df$Store_num)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1331.43   -39.29   -30.06    1.79  1899.23 
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 57.66519  0.94148  61.249 <2e-16 ***
## final_df$Month -0.85972  0.09846 -8.731 <2e-16 ***
## final_df$Store_num -0.74645  0.04619 -16.160 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 94.27 on 79456 degrees of freedom
## Multiple R-squared:  0.004246, Adjusted R-squared:  0.004221 
## F-statistic: 169.4 on 2 and 79456 DF,  p-value: < 2.2e-16
```

Betakoeffisientene fra en regresjonsmodell kan brukes for å predikere fremtidig utvikling i en avhengig variabel. Dette var bare et enkelt eksempel for å illustrere potensialet av predikativ regresjonsanalyse. Med en veldig lav R^2 verdi er akkuraten denne modellen ikke særlig godt egnet for prediksjoner og prognosenter.

Flere forslag og ideer:

```
# Tidsserieanalyse, dato og profitt, utvikling over lang tid
# En ser at kategorien "soup" har hatt en positiv utvikling den siste tiden.
# Har produktet blitt mer populært? Har kostnadene sunket? Bør de satse mer på det?
df_barplot6 <- bind_rows(pizza, burger, soup)
df_barplot6 <- rename(df_barplot6, Dato = Date)
df_barplot6 <- rename(df_barplot6, Profitt = Profit)
df_barplot6 <- rename(df_barplot6, Varekategori = Description)
ggplot(df_barplot6, aes(x = Dato, y = Profitt)) +
  geom_line(aes(color = Varekategori), size = 1) +
  scale_color_manual(values = c("#00AFBB", "#E7B800", 'red')) +
  theme_minimal() + labs(title = "Profitt per varekategori over lang tid")
```

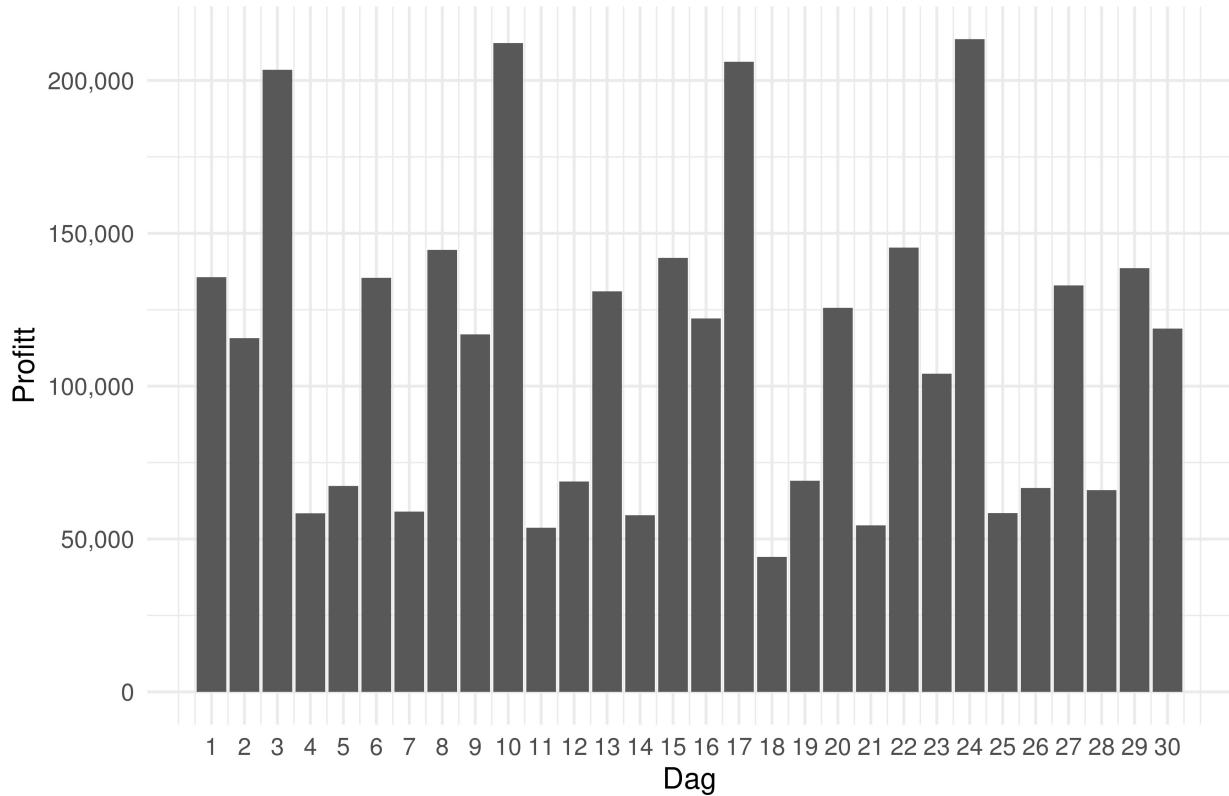


```

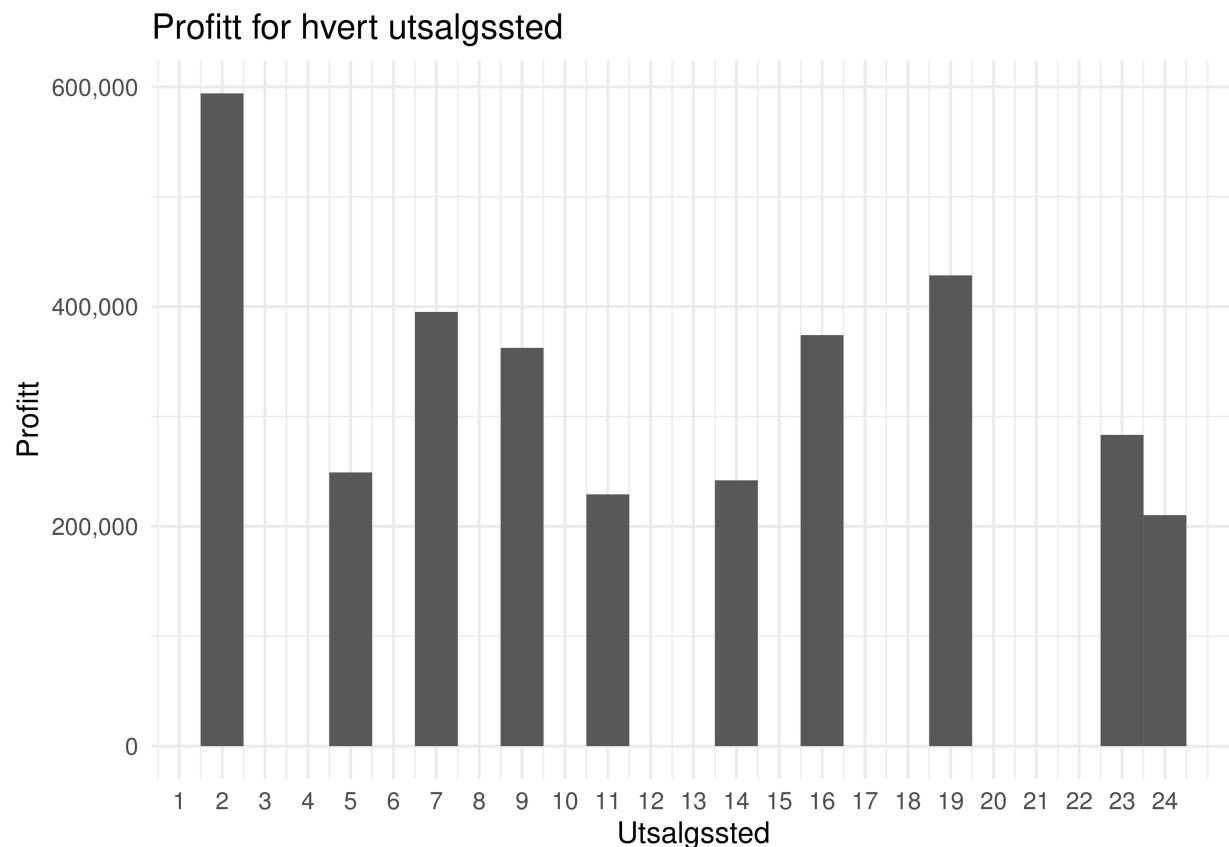
# Best day for sales
# Konsernet tjener mest profitt på søndager
# y-labelen kunne ha vært i enheter av hundre tusen
# Grafen kunne vært penere. De syv ukedagene kunne hatt hver sin farge i diagrammet.
require(scales)
df_barplot3 <- setNames(aggregate(final_df$Profit, by=list(Dag=final_df$Day),
                                    FUN=sum),c("Dag", "Profitt"))
ggplot(df_barplot3, aes(x=Dag, y=Profitt)) + geom_bar(stat="identity") +
  scale_x_continuous("Dag", labels = as.character(1:30), breaks = 1:30) +
  labs(title = "Profitt per dag. Dag 1: fredag") + theme_minimal() +
  scale_y_continuous(labels = comma)

```

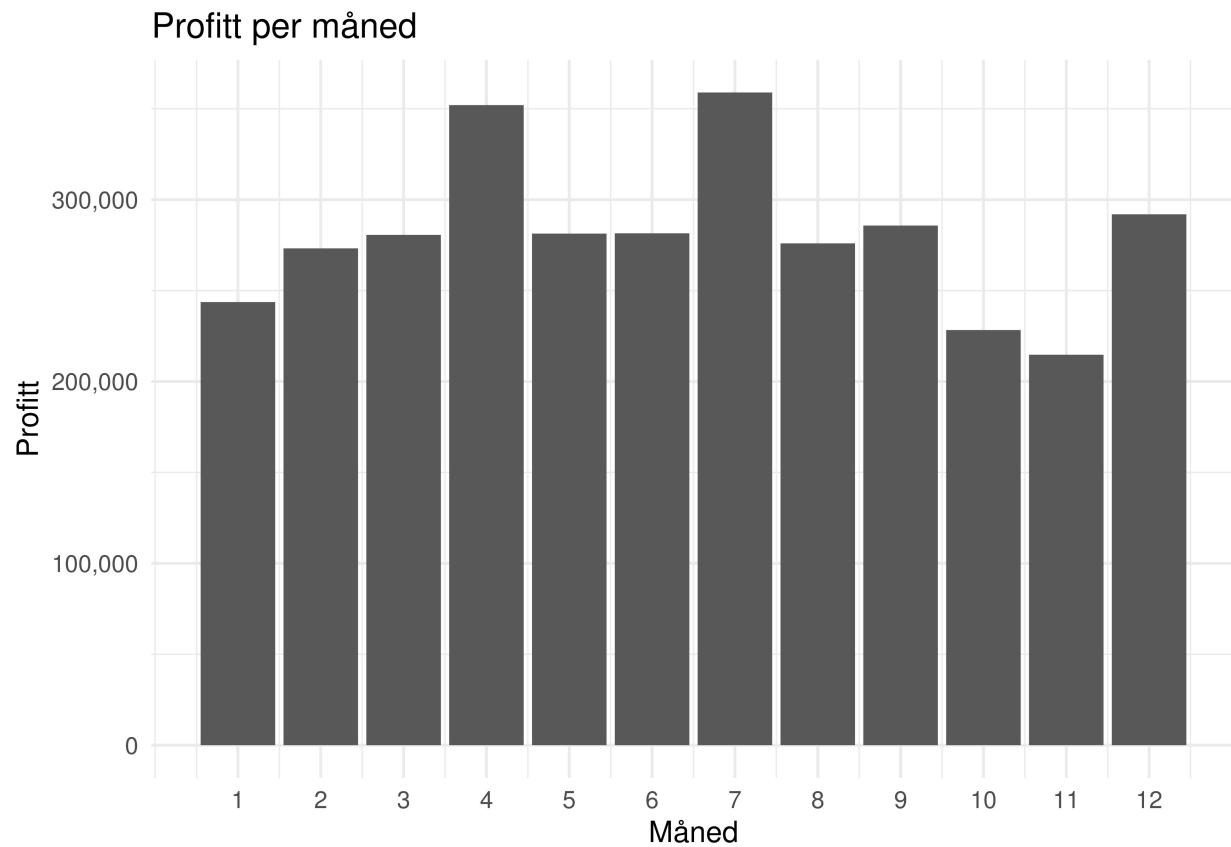
Profitt per dag. Dag 1: fredag



```
# Best store
df_barplot4 <- setNames(aggregate(final_df$Profit, by=list(Utsalgssted=final_df$Store_num),
                                FUN=sum), c("Utsalgssted", "Profitt"))
ggplot(df_barplot4, aes(x=Utsalgssted, y=Profitt)) +
  geom_col(width = 1, position = "dodge") +
  scale_x_continuous("Utsalgssted", labels = as.character(1:24), breaks = 1:24) +
  labs(title = "Profitt for hvert utsalgssted") +
  theme_minimal() + scale_y_continuous(labels = comma)
```



```
# Best month:
df_barplot5 <- setNames(aggregate(final_df$Profit, by=list(Category=final_df$Month),
                                         ,FUN=sum),c("Måned", "Profitt"))
ggplot(df_barplot5, aes(x=Måned, y=Profitt)) + geom_bar(stat="identity") +
  scale_x_continuous("Måned", labels = as.character(1:12), breaks = 1:12) +
  labs(title = "Profitt per måned") + theme_minimal() +
  scale_y_continuous(labels = comma)
```



Oppgave 4

Dataene kan benyttes til å planlegge nye utsalg og dersom konsernledelsen ønsker å etablere et nytt utsalg, kan de benytte disse dataene til å finne den beste lokasjonen. Ettersom utsalgsstedet #2 er det mest populære utsalgsstedet, kan det være lurt å etablere en ny butikk i Free Standing, Power City. Nytten av dette må ses i sammenheng med konkurransen i området, kostnader forbundet med etablering og kanskje til og med kriminalitetsstatistikken. Dette kan være variabler som Store_Competition_Fastfood, Store_Competition_Otherfood, Annual_Rent_Estimate, Cost og County_Total_Crimes.

```
final_df %>%
  group_by(Store_num) %>%
  summarise(Profit = sum(Profit)) %>%
  arrange(desc(Profit))

## # A tibble: 10 x 2
##   Store_num   Profit
##       <dbl>     <dbl>
## 1 1         594086.
## 2 2         428464.
## 3 3         395206.
## 4 4         374082.
## 5 5         362474.
## 6 6         283267.
## 7 7         249082.
## 8 8         241926.
## 9 9         229132.
## 10 10        210219.

head(final_df %>% filter(Store_num==2) %>% select(Store_num, Store_City, Store_Location),1)

##   Store_num Store_City Store_Location
## 1 2          Power City  Free Standing
```