

Prosjektoppgave SOK-1005

Gruppe nr. 4

6/6/2022

Oppgave 1

I denne oppgaven skal vi være analytikere for en sandwich-bedrift i USA. Vår oppgave blir å analysere og se på detaljer innen salg.

I første oppgaven, skal vi sette sammen seks ulike datasett til et stort. Hvor alt av salgsinformasjon og detaljer som påvirker bedriften på ulike måter. Vi starter med å lese inn alle de seks datasettene. Bruker `left_join` for å få alle kolonnene og rader til å høre sammen i det nye datasettet, `left_join` ligger i pakken `Dplyr`. Vi summerer profitten for alle butikkene og finner ut at butikk nr 2 og 3, har tall vi tenker er interessante. Butikk nr 2 er ca på 1.5 mill og butikk nr 3 har 0 i profitt. Dette vil være noe bedriften bør vurdere å se nærmere på i nærmeste fremtid.

```
profit <- with(fullyJoined, sum(Profit[Store_num >'2']))
profit
```

```
## [1] 1500248
```

```
profit2 <- with(fullyJoined, sum(Profit[Store_num =='2']))
profit2
```

```
## [1] 594085.5
```

```
profit3 <- with(fullyJoined, sum(Profit[Store_num == "3"]))
profit3
```

```
## [1] 0
```

Oppgave 2

```
store_num_2 <-
  salesCountyData %>%
  filter(Store_num == "2") %>%
  select("County_Unemployed", "Sales", "Description", "Day", "Date", "Profit")

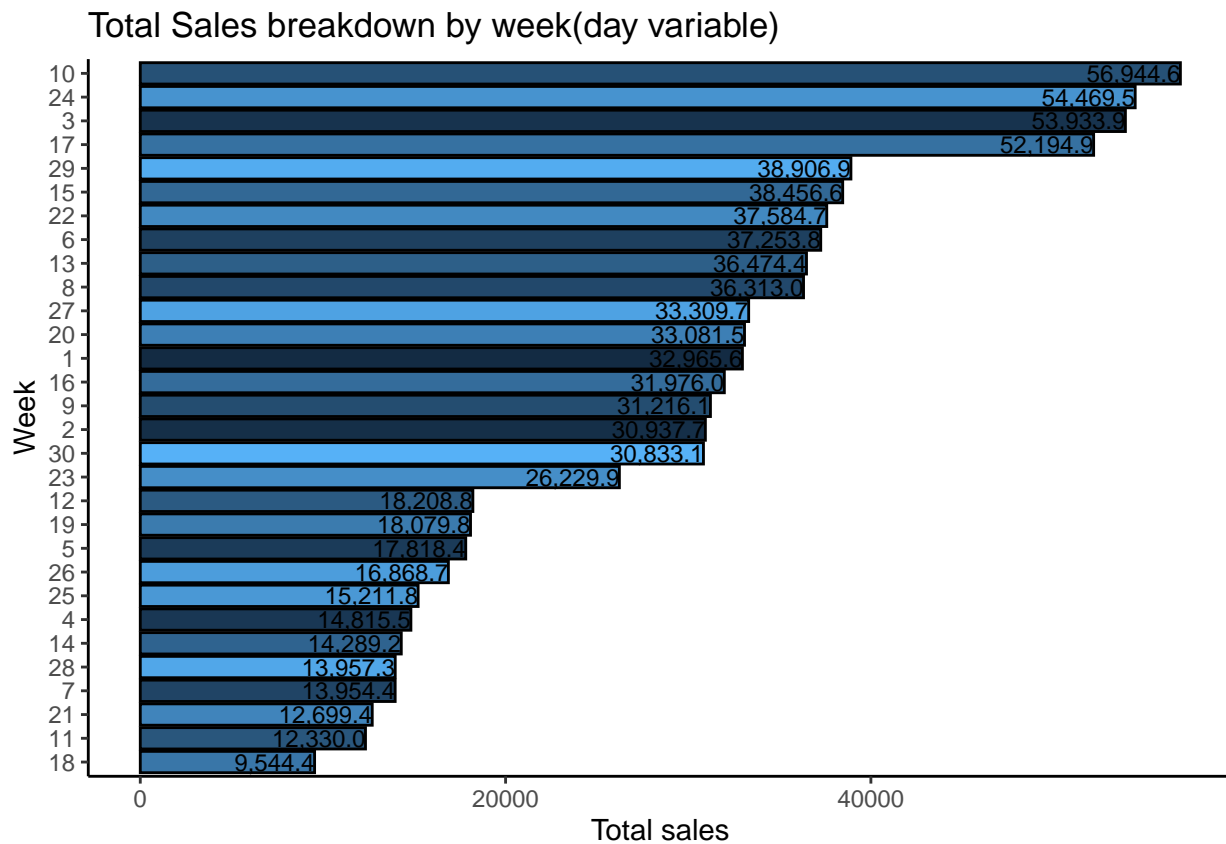
#total sales per week (day variable)
sales_by_day <- store_num_2 %>%
  group_by(Day) %>%
  summarise(Total_Sales=sum(Sales)) %>%
```

```

ungroup

##Visualizing summary data ->
#Ranging Top turnover weeks
sales_by_day %>%
  ggplot(aes(reorder(Day, Total_Sales), Total_Sales, fill=Day)) +
  geom_col(show.legend = FALSE, color="black") +
  geom_text(aes(label=comma(Total_Sales)), size=3, hjust=1, color="black") +
  coord_flip() +
  theme_classic() +
  labs(title = "Total Sales breakdown by week(day variable)", x="Week", y= "Total sales")

```

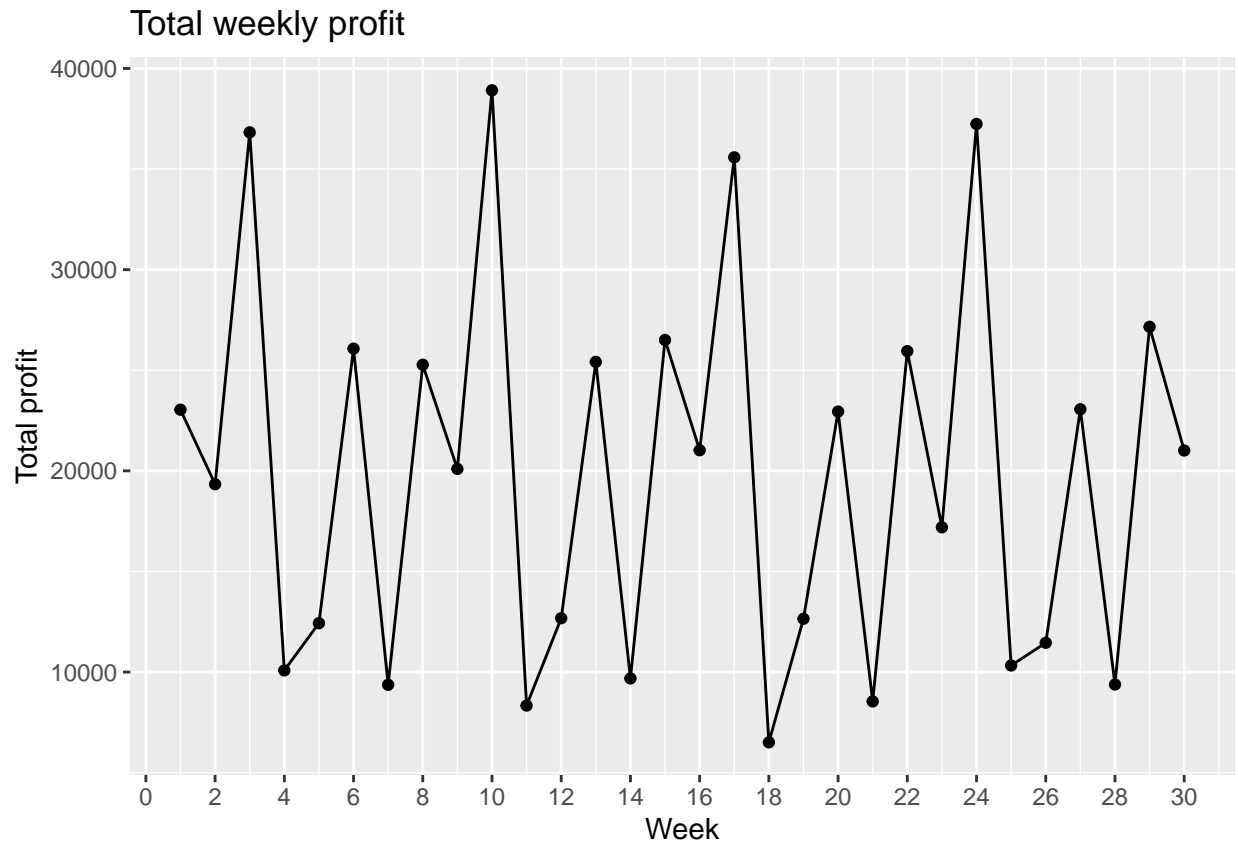


```

#filtering for profit
profit_by_day <- store_num_2 %>%
  group_by(Day) %>%
  summarise(Total_profit=sum(Profit)) %>%
  ungroup

#plotting profit each week
ggplot(data=profit_by_day, aes(x=Day, y=Total_profit, group=1)) +
  geom_line() +
  scale_x_continuous(n.breaks = 16) +
  labs(title = "Total weekly profit", x = "Week", y = "Total profit") +
  geom_point()

```



```
# total weekly turnover
```

```
# best sellers 1 weekly store num 2
```

```
store_1week <- wSales %>%
  filter(Store_num == "2", Month == "10") %>%
  select("Description", "Price", "Sold", "Sales", "Profit", "Date", "Month", "Day")
```

```
#Total turnover week 7
```

```
new_sum <- store_1week %>%
  filter(Day == "7") %>%
  group_by(Day) %>%
  summarise(Total_sales = sum(Sales)) %>%
  ungroup()
new_sum
```

```
## # A tibble: 1 x 2
##   Day Total_sales
##   <int>      <dbl>
## 1     7    13954.
```

```
#total rev. for store number two in week number 7 is 13954 dollar
```

```
## 'summarise()' has grouped output by 'INV_NUMBER', 'Description', 'Price',
## 'Sold', 'Cost', 'Profit'. You can override using the '.groups' argument.
```

```
## # A tibble: 174 x 7
##   INV_NUMBER Description      Price Sold Cost Profit Sales
##   <int> <chr>          <dbl> <int> <dbl> <dbl> <dbl>
## 1    1651 REGULAR SAVORY TURKEY    6.75  171  367.   809. 1176
## 2    1611 REGULAR HONEY MUSTARD HAM    5    185  365.   559. 924.
## 3   35050 REGULAR CHICKEN BACON RANCH  8.25   97    0    770. 770.
## 4    5014 CHIPS                1.1   674  276.   458. 734.
## 5     151 MINI SAVORY TURKEY    4.75  131  143.   501. 644
## 6   11519 21OZ DRINK            1.5   388  199.   420. 619.
## 7    1561 REGULAR CLASSIC TUNA    6.75   85  143.   452. 595
## 8    1501 REGULAR BMT            6.75   82  174   400  574
## 9    1521 REGULAR MEATBALL MARINARA    5    115  299.   269. 568.
## 10   1511 REGULAR COLD CUT MUSHROOM    5    110  214.   335. 550.
## # ... with 164 more rows
```

```
## 'summarise()' has grouped output by 'INV_NUMBER', 'Description', 'Price',
## 'Sold', 'Cost', 'Profit'. You can override using the '.groups' argument.
```

```
## # A tibble: 174 x 7
##   INV_NUMBER Description      Price Sold Cost Profit Sales
##   <int> <chr>          <dbl> <int> <dbl> <dbl> <dbl>
## 1    1651 REGULAR SAVORY TURKEY    6.75  171  367.   809. 1176
## 2   35050 REGULAR CHICKEN BACON RANCH  8.25   97    0    770. 770.
## 3    1611 REGULAR HONEY MUSTARD HAM    5    185  365.   559. 924.
## 4     151 MINI SAVORY TURKEY    4.75  131  143.   501. 644
## 5    5014 CHIPS                1.1   674  276.   458. 734.
## 6    1561 REGULAR CLASSIC TUNA    6.75   85  143.   452. 595
## 7   11519 21OZ DRINK            1.5   388  199.   420. 619.
## 8    1501 REGULAR BMT            6.75   82  174   400  574
## 9   13751 REGULAR SPICY ITALIAN    5.75   68  70.7   395. 466.
## 10  13821 REGULAR CHICKEN TERIYAKI    7.75   64  91.5   356. 448
## # ... with 164 more rows
```

```
## 'summarise()' has grouped output by 'INV_NUMBER', 'Description', 'Price',
## 'Sold', 'Cost', 'Profit'. You can override using the '.groups' argument.
```

```
## # A tibble: 200 x 7
##   INV_NUMBER Description      Price Sold Cost Profit Sales
##   <int> <chr>          <dbl> <int> <dbl> <dbl> <dbl>
## 1    1651 REGULAR SAVORY TURKEY    6.75  202  433.   975. 1408.
## 2    1611 REGULAR HONEY MUSTARD HAM    5    215  424.   651. 1075
## 3   35050 REGULAR CHICKEN BACON RANCH  8.25  121    0    968  968
## 4     151 MINI SAVORY TURKEY    4.75  186  203.   723. 926.
## 5    1611 REGULAR HONEY MUSTARD HAM    5    185  365.   559. 924.
## 6    5014 CHIPS                1.1   773  316.   531. 848.
## 7    1631 REGULAR ROASTED CHICKEN    6.75  101  230.   477. 706.
## 8    1511 REGULAR COLD CUT MUSHROOM    5    128  249.   390. 640.
## 9   11519 21OZ DRINK            1.5   385  198.   418. 616.
## 10   1501 REGULAR BMT            6.75   87  185.   424. 609
## # ... with 190 more rows
```

```
## 'summarise()' has grouped output by 'INV_NUMBER', 'Description', 'Price',
## 'Sold', 'Cost', 'Profit'. You can override using the '.groups' argument.
```

```
## # A tibble: 200 x 7
##   INV_NUMBER Description      Price Sold Cost Profit Sales
##   <int> <chr>          <dbl> <int> <dbl> <dbl> <dbl>
## 1      1651 REGULAR SAVORY TURKEY    6.75  202 433.   975. 1408.
## 2     35050 REGULAR CHICKEN BACON RANCH 8.25  121   0    968   968
## 3       151 MINI SAVORY TURKEY    4.75  186 203.   723.  926.
## 4      1611 REGULAR HONEY MUSTARD HAM    5    215 424.   651. 1075
## 5      1611 REGULAR HONEY MUSTARD HAM    5    185 365.   559.  924.
## 6      5014 CHIPS    1.1    773 316.   531.  848.
## 7      1631 REGULAR ROASTED CHICKEN    6.75  101 230.   477.  706.
## 8     13751 REGULAR SPICY ITALIAN    5.75   76 79.0   448.  527.
## 9      1561 REGULAR CLASSIC TUNA    6.75   84 142.   446.  587.
## 10     1501 REGULAR BMT    6.75   87 185.   424.  609
## # ... with 190 more rows
```

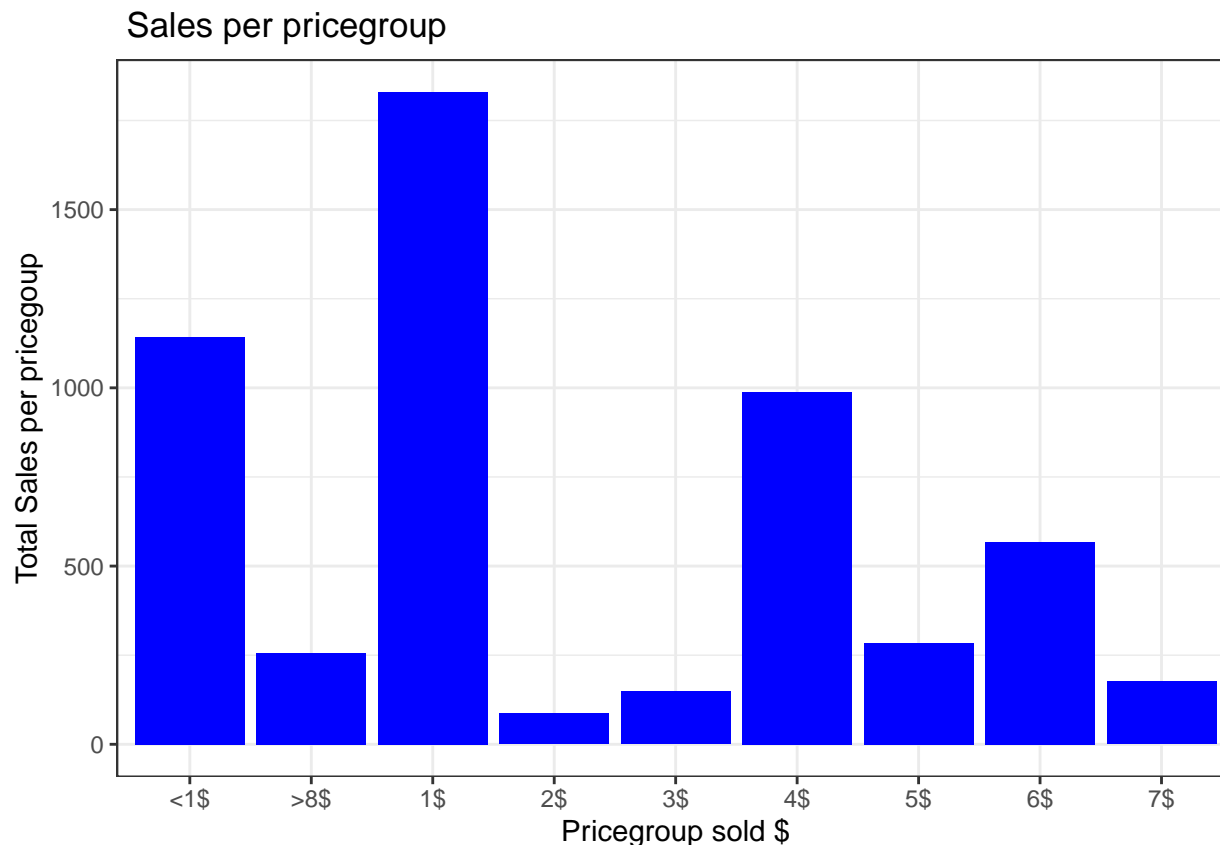
```
## 'summarise()' has grouped output by 'INV_NUMBER', 'Description', 'Price',
## 'Sold', 'Cost', 'Profit'. You can override using the '.groups' argument.
```

```
## # A tibble: 174 x 7
##   INV_NUMBER Description      Price Sold Cost Profit Sales
##   <int> <chr>          <dbl> <int> <dbl> <dbl> <dbl>
## 1      4212 REGULAR SUB OR SALAD, REWARD    0    80   0 -539.  -539.
## 2      4210 MINI SUB, REWARDS    0    51   0 -263.  -263.
## 3      4014 BOGO MINI SUB    0    49   0 -205.  -205.
## 4      9001 VAL MEAL 1 MINI MTBALL CHIP  -0.6   422   0 -84.2  -84.2
## 5      4017 FREE MINI SUB    0    12   0 -58.2  -58.2
## 6      4015 FREE REGULAR    0     5   0 -31    -31
## 7      4208 21 OZ DRINK, REWARDS    0    15   0 -24    -24
## 8      4207 CHIPS, REWARDS    0    21   0 -23.1  -23.1
## 9      9023 VAL MEAL 3 MINI TURKEY CHIP  -0.6    53   0 -10.6  -10.6
## 10     4020 FREE COOKIE    0    10   0 -9.25  -9.25
## # ... with 164 more rows
```

```
## 'summarise()' has grouped output by 'INV_NUMBER', 'Description', 'Price',
## 'Sold', 'Cost', 'Profit'. You can override using the '.groups' argument.
```

```
## # A tibble: 200 x 7
##   INV_NUMBER Description      Price Sold Cost Profit Sales
##   <int> <chr>          <dbl> <int> <dbl> <dbl> <dbl>
## 1      4212 REGULAR SUB OR SALAD, REWARD    0    87   0 -596  -596
## 2      4014 BOGO MINI SUB    0    96   0 -412  -412
## 3      4210 MINI SUB, REWARDS    0    39   0 -185.  -185.
## 4      4015 FREE REGULAR    0    23   0 -144  -144
## 5      9001 VAL MEAL 1 MINI MTBALL CHIP  -0.6   471   0 -94.0  -94.0
## 6      4207 CHIPS, REWARDS    0    21   0 -23.1  -23.1
## 7      4208 21 OZ DRINK, REWARDS    0    12   0 -19.2  -19.2
## 8      4017 FREE MINI SUB    0     4   0 -15.5  -15.5
## 9      9023 VAL MEAL 3 MINI TURKEY CHIP  -0.6    68   0 -13.6  -13.6
## 10     4020 FREE COOKIE    0    12   0  -6    -6
## # ... with 190 more rows
```

```
## 'summarise()' has grouped output by 'Price_per_item'. You can override using the
## '.groups' argument.
```



Vi skal nå lage en ukentlig salgsrapport for bedriften. Dette gjør vi for å kunne se etter spesielle tilfeller salg og resultater, hvor til slutt bedriften kan bruke dataen til å vurdere om de bør forandre på noe. Vårt valg av utsalg nr 2, er grunnet våre funn i oppgave 1. Butikk nr 2 har en høy profitt, som gjør det interessant å se på hvordan bedriften driftes. I en ukentlig rapport vil det være interessant å se profitt, omsetning og varer som har høye salgstall og lave salgstall. I dette datasettet er ukene representert som variabelen «day». Dette gjør at vi ikke har mulighet til å dele opp datasettet for hver dag, noe som gjør at vi ikke kan ta en daglig rapport av salg for alle butikkene. En dagsrapport er essensielt for en ukesrapport, for å kunne få mer detaljert informasjon og nyttig informasjon til ledelsen. Vi vil tro når en bedrift ønsker seg en ukentlig rapport, vil de nok ha muligheten til å kunne gå inn på enkelte dager for finne data. Hver dagsrapport kan hjelpe bedriften til å vite når de skal justere salg og andre tiltak som vil forbedre bedriften.

Datasettet er vrient til tider, derfor vil vi anbefale ledelsen i bedriften og justere hvordan dataen fra de ulike butikkene blir samlet inn. Dette vil gjøre det enklere for å lage gode figurer og tabeller, som gir en bedre oversikt og informasjonsflyt.

Vi rangerer prisen på varer i ulike prisgrupper, dette gjør at vi får en oversikt over hvilken prisgruppe som det blir solgt mest av og hvilken det blir solgt minst av. I butikk nr2, prisgruppen: (1 dollar) er den gruppen som har størst omsetning og den som har minst omsetning er prisgruppen: (2 dollar). Videre kan vi se hvordan konsumentene velger varer basert på prisen deres. Det kan vi se gjennom våre to prisgrupper på: (større enn 1 dollar) og (mindre enn 8 dollar), hvor vi ser at prisgruppen (større enn 1 dollar) blir betydelig konsumert mer enn den med 8 dollar. Det viser oss at konsumentene er interessert i å bruke mer enn 1 dollar, men mindre enn 8 dollar. Vi legger også merke til at det er ingen varer som går over 8 dollar, det kan gi oss et innblikk i hva konsumentene er villige til å betale. Det kan være noe ledelsen bør tenke på i videre prising av varer ved nye utsalg.

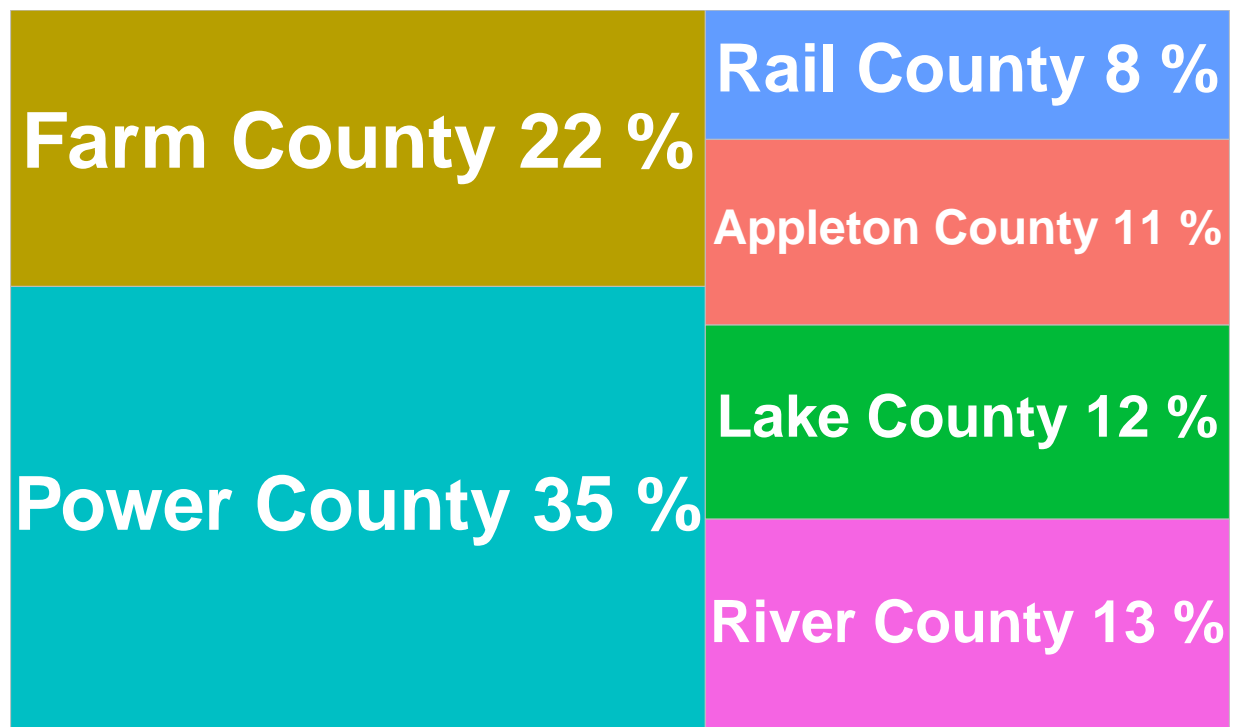
Oppgave 3

```
df_4 <- fullyJoined %>%
  filter(Month == 10, Year == 2012, Price >= 0, Cost >= 0, Sold >= 0) # Filtering for needed data

inntekt_summary <- df_4%>%
  group_by(Store_County) %>%
  summarise( Total_revenue = sum(Sales)) %>% #
  summarise( percent_revenue = Total_revenue/sum(Total_revenue),
             Total_revenue = Total_revenue, Country = Store_County) #Choosing data
inntekt_summary <- inntekt_summary %>%
  mutate(labels = paste(Country, paste(round(percent_revenue,2)*100,'%') , sep = ' '))
#plotting the strongest county
inntekt_summary %>%
  ggplot(aes(area = Total_revenue, fill = Country, label = labels)) +
  geom_treemap() +
  geom_treemap_text(fontface = "bold", colour = "white", place = "centre", grow = TRUE) +
  #Changing estetics
  theme(legend.position = "none") +
  labs(title = 'comparing county sales data') +
  labs(subtitle = 'Shows the most dominant county') +
  theme(plot.title = element_text(size = 20, face = "bold"),
        plot.subtitle = element_text(size = 12))
```

comparing county sales data

Shows the most dominant county



Vi skal nå lage en månedlig salgsrapport for bedriften. Vi sammenligner de ulike butikkene i forhold til hvor de ligger geografisk. Ut ifra dette kan vi se på variabelen «PowerCounty», hvor vi finner ut hvilket County

som har høyest profitt. Derifra kan man se på variablene «Crime» og «Weather» og se på hvordan de er knyttet opp til det County med høyest profitt, om de viser oss en sammenheng mellom lav kriminalitet og godt vær. «PowerCounty» er den variabelen med høyest salgsprosent, som viser oss så mye som 35% av all salgsdataen for hele datasettet. «RailCounty» er det County som har lavest salgsprosent av hele datasettet, på hele 8%.

Oppgave 4

Dataene kan brukes til å bestemme nye lokasjoner til nye utslag. Ved bruk av dataen, kan konsernledelsen se etter sammenhengen mellom omsetning, vær, kriminalitet og velferd. Det er flere variabler som konsernledelsen bør og kan se gjennom, men disse kan være gode indikatorer på hvor nytt utslag bør være. Geografisk sett vil steder med godt vær og lite kriminalitet være gode forutsetninger for nye utslag, dette kan ofte være steder med en høyere velferd enn resterende av de andre kommuner. Vi kan også se at frittstående butikker er mer lønnsomme. Fra oppgave 3 kan vi legge merke til at det er stort utslag og høy profitt i «PowerCounty» som er en sterk indikator på at et nytt utslag her vil også kunne drifte godt.

Ved å se på prisgruppene fra oppgave 2, nevnt lenger opp, kan man tro at det er konsumenter med et lavere budsjett som oftest handler. Det vil vi tro grunnet at prisgruppen: (større enn 1 dollar) er den høyeste av prisgruppene. Det vil være lurt av konsernledelsen å legge et nytt utslag i en av kommunene som også har en høy profitt, men ikke i den høyeste som er «PowerCounty», dette er grunnet unødvendig konkurranse og det vil mye mulig være lettere å starte et nytt utslag i et fylke med mindre konkurranse.

Som en oppsummering bør konsernledelsen og bedriften se på viktige forutsetninger som nevnt over, før de velger et nytt geografisk område for nytt utslag.