Obligatorisk læringsaktivitet i introuger Flow 1, Dataanalyse

Udarbejdet af:

Emil Gustav Haagensen

Indholdsfortegnelse

[Opgave 1 – Data Science Modellen 3](#_bookmark0)

[Opgave 1.2 - Vælg 3](#_bookmark1)

[Opgave 1.3 – Beskriv variabler 5](#_bookmark2)

[Opgave 1.4 – Research goal 7](#_bookmark3)

[Opgave 2 – Korrelation og simple lineær regression 8](#_bookmark6)

[Opgave 2.1 – Beskrivende statistik 8](#_bookmark7)

[Opgave 2.2 - Korrelation 9](#_bookmark8)

[Opgave 2.3 – Simple regressioner 10](#_bookmark10)

[Opgave 2.4 – Korrelation og lineær regression 12](#_bookmark11)

[Opgave 3 – Tilfældigheder og terninger 13](#_bookmark14)

[Opgave 3.1 – Funktion til terninger 13](#_bookmark15)

[Opgave 3.2 – Plot I 14](#_bookmark16)

[Opgave 3.3 – Plot II 15](#_bookmark17)

[Opgave 3.4 – Lav dine egne data 16](#_bookmark18)

[Opgave 4 – Danskernes forhold til alkohol 17](#_bookmark19)

[Opgave 4.1 – Hent data 17](#_bookmark20)

[Opgave 4.2 – Korrelation 20](#_bookmark21)

[Opgave 4.3 – Kritisk tænkning 21](#_bookmark22)

[Opgave 5 – Dataframes 22](#_bookmark23)

[Opgave 5.1 – Månedlige observationer 22](#_bookmark24)

[Opgave 5.2 – Kvartalsvise observationer 23](#_bookmark25)

[Opgave 5.3 - Pivot 25](#_bookmark26)

# 

# Metode

## Til denne OLA er følgende pakker brugt udover BaseR:

* **Tidyverse**

Tidyverse er en større samling af flere R-pakker. Tidyverse er i denne opgave til en mere effektiv data manipulation, visualisering/plotting, feature engineering og generel rensning af data. Mere specifikt er der gjort brugt af underpakkerne: Dplyr, ggplot2, Tidyr.

* **Corrplot**

Coorplot-pakken er i denne opgave blevet brugt til at visualere korrelationsmatricerne på en effektiv måde, så plottet er ”easy to communicate”.

* **Skimr**

Skim-r pakken er blevet brugt til et mere dybdegående summary på datasættet fra boligsiden, herunder både for at danne er overblik over fordelingen af NA-værdier såvel som spredningen i datasættets variabler.

* **DKstat**

DKstat-pakken er Statistikbankens API, som lader os indtrække data gennem API’en fremfor gennem hjemmesiden, download, omformatering til CSV, oprydning mv.

API løsning gør det altså dermed lettere at tilgå dataen, såvel som at opsætte og analysere det.

# Opgave 1 – Data Science Modellen

## Opgave 1.1 – find data

**Data på ejendomme til salg i opgave 1 er fremkommet via webscapping af boligsiden.dk. Der ligger en csv- fil i mappen med OLA-opgaven. Her er to billeder fra boligsiden.dk. Find de to rækker i csv-filen, som matcher de to huse.**

Til følgende opgave arbejdes i BaseR.

Datasættet til opgaven er blevet uddelt i form af CSV-filen `boligsiden.csv`. Datasættet er bestående af webscrapet data fra hjemmesiden bilbasen.dk.

For at indlæse filen benytter vi os af funktionen `read.csv`, som lader os indhente data fra en csv-fil ind i vores R-miljø.

**Et billede, der indeholder Font/skrifttype, tekst, skærmbillede, Grafik

Automatisk genereret beskrivelse**

Det indhentede data bliver indhentet i format af en dataframe. De første 16 rækker af dataframen kan ses nedenfor.

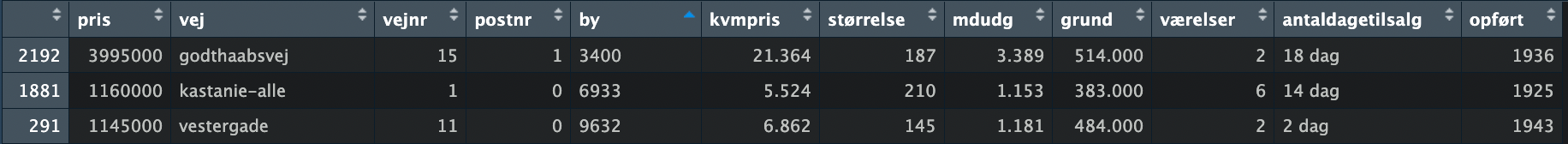
Dataframen indeholder 2552 rækker (ergo 2552 forskellige boliger til salg, med undtagelse af rækker fyldt med NA-værdier).

**Et billede, der indeholder tekst, skærmbillede, software

Automatisk genereret beskrivelse**

Som det fremstår af udsnittet af vores dataframe, så er der både enkelte observationer, kolonner og i nogle tilfælde, hele rækker som indeholder fejlagtig data, herunder kan der påpeges kolonnen, antaldagetilsalg, som har en stor mængde NA-værdier. Kolonnen, grund, som har et ustruktureret format af observationerne, hvor punktummet i værdien nogle gange agerer som tusindetalsseperator, og i andre tilfælde som et komma.

Yderligere, så er der tilfælde hvor dataene kommer ind i forkerte kolonner. F.eks., så er der tre rækker, hvor postnummeret er blevet indhentet til at være i by-kolonnen. Dette kan ses nedenfor.

****

Grunden til at dataen kommer ind på denne måde er på grund af, at dataen er blevet webscrapet.

Webscraping er en meget rå form for data-retrieval, hvor man gennem HTTP-requests, såsom GET og POST, udtrækker information direkte fra hjemmesiders HTML-struktur. De data man scraper, er typisk ikke tilegnet at skulle udtrækkes, og er derfor ikke nødvendigvis opsat i et systematisk format. Det betyder altså derfor, at de data man udtrækker er 100% afhængig af hjemmesidens HTML-struktur, og det kan derfor resulterer i data som indeholder fejl, er ustruktureret mv.

I betragtning af, at vi ikke selv har webscrapet dataen, er det svært at kunne redegøre samt undersøge hvorfor fejlene opstår, da der mangler dokumentation for tilgangen til webscrapingen.

For at de to rækker i dataframen som matcher de to billeder benytter vi os af nogle unikke værdier, som sørger for, at vi kan identificere rækkerne nemt. Som belyst på nedenstående billede kan vi se 4 unikke værdier som vi bruger til at finde rækkerne.

**Et billede, der indeholder tekst, Website, Webside, træ

Automatisk genereret beskrivelse**

Som det fremstår, så finder vi begge boliger ved hjælp af nedenstående kode.

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, linje/række

Automatisk genereret beskrivelse

Rækkerne ser således ud:

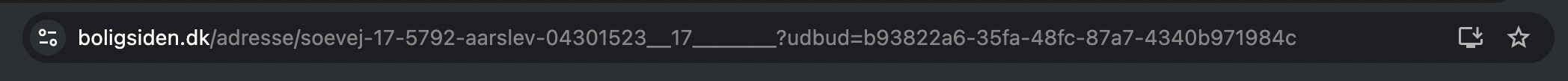


## Opgave 1.2 - Vælg

### Udvælg 2 ejendomme fra CSV-filen og find dem på boligsiden.dk.

Ved at søge på et par af adresserne fra CSV-filen på boligsiden.dk, bliver det hurtigt belyst, at det er et ældre datasæt, da mange af ejendomme ikke kan findes på Boligsiden længere.

Ved hjælp af webscraping ville vi på normalvis være i stand til at skabe links og søge efter boligerne på boligsiden.dk. – Men alle de data som indgår i URL’en til boligerne på hjemmesiden er ikke blevet scrapet og det er dermed ikke muligt alligevel.

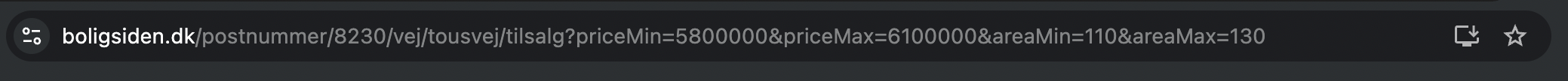


Som det fremstår af ovenstående URL-eksempel er URL’en opbygget med forskellige informationer.

URL’en består af en adresse, herunder vej og vejnr, som er data vi har adgang til. Efter dette findes et unikt bolig-ID (i dette eksempel er ID’et: ”04301523\_\_17\_\_\_\_\_\_?”). Uden dette boligID har vi ikke mulighed for at lave enten en funktion eller et loop, som dynamisk opsætter links udfra vores data i dataframen.

Der er dog mulighed for en anden mere lavpraktisk metode, som kræver en hæftig mængde GET-requests, tid og som formentlig ender med at returnere statuskoden 202 relativt hurtigt, grundet mængden af GET-requests fra samme IP.

Når man søger efter boliger kan man nemlig på boligsiden specificere parametre/filtre for ens søgning, hvorefter der kan foretages egen webscraping for at udtrække de fremfundne boliger, hvis tilgængelige.



I ovenstående URL-eksempel er der opstillet et URL for søgning efter en af boligerne som skulle findes i opgave 1.1. Som det fremstår af URL’en er der kun opstillet parametre, som er tilgængelige gennem vores data i dataframen. Ved hjælp af disse parametre ville det være muligt at afgrænse søgning til kun at indeholde få boliger og i tilfælde af, at boligen stadigvæk er til salg og tilgængelig på hjemmesiden, ville boligen være iblandt.

Problemet med denne fremgangsmåde er omfanget af opgaven. Der ville skulle skabes et link for hver række i dataframen (2.552), hvorefter der ville skulle føres adskillige GET-requests for hver URL.

Ved at lave så mange GET-requests fra samme IP ville medføre at søgningen efter tilgængelige boliger, enten ville meget lang tid, da man ville være nødsaget til at inkorpere tidsintervaller med pauser, så data-udtrækningen ikke fremstår som en robot der vil scrape al den data den kan, som ville medføre at dette ville tage meget lang tid. Og i det andet tilfælde ville man hurtigt støde på en statuskode som udelukker en, så det ikke længere er muligt at indtrække data fra IP-adressen man er på.

Derfor udtrækkes to boliger denne gang manuelt, ved at søge efter boligerne på hjemmesiden, og se om de er tilgængelige.

Det skal desuden påpeges, at hvis der ved scraping i første omgang, også scrapes efter boligernes URL eller deres unikke-ID, ville ovenstående være meget mere gennemførligt.

I tilfælde af, at der ville være et link tilgængeligt for hver bolig, ville der plot skulle skabes et loop, som looper gennem hvert link og returnerer en statuskode. Hvis statuskoden er 404, betyder det at siden ikke er tilgængelige længere som enten betyder at boligen enten er blevet solgt eller af anden grund, ikke længere er tilgængelig på hjemmesiden, derimod hvis der returneres en statuskode 200, ville det betyde, at siden eksisterer og vi har adgang til siden.

De to manuelt udtrukkede boliger har følgende links:

* [https://www.boligsiden.dk/adresse/aadalsvej-19-9210-aalborg-soe-](https://www.boligsiden.dk/adresse/aadalsvej-19-9210-aalborg-soe-08519762__19_______?udbud=3e3e4c00-a3fa-4b94-a5fa-e44ce995ea6c) [08519762 19 ?udbud=3e3e4c00-a3fa-4b94-a5fa-e44ce995ea6c](https://www.boligsiden.dk/adresse/aadalsvej-19-9210-aalborg-soe-08519762__19_______?udbud=3e3e4c00-a3fa-4b94-a5fa-e44ce995ea6c)
* [https://www.boligsiden.dk/adresse/gylfesvej-4-3060-espergaerde-](https://www.boligsiden.dk/adresse/gylfesvej-4-3060-espergaerde-02172715___4_______?udbud=7463ac9a-9d3d-46c2-b7bb-c1ec4c8e51e3) [02172715 4 ?udbud=7463ac9a-9d3d-46c2-b7bb-c1ec4c8e51e3](https://www.boligsiden.dk/adresse/gylfesvej-4-3060-espergaerde-02172715___4_______?udbud=7463ac9a-9d3d-46c2-b7bb-c1ec4c8e51e3)

## Opgave 1.3 – Beskriv variabler

### Forklar NA-værdierne i CSV-filen ud fra, hvad I har observeret i opgave 1.1 og 1.2. Derudover gør rede for variable I mener mangler i CSV-filen sammenlignet med boligsiden.dk.

### En NA-værdi betyder at der mangler data i den pågældende celle/observation. NA-værdier kan opstå på mange forskellige måder.

### NA-værdier kan opstå under selve webscrapingen, hvis hjemmesiden ikke følger samme statiske HTML-struktur, og at en værdi f.eks. er gemt under et bestemt css-tag for den ene side, hvorimod den måske er gemt under et andet tag på en anden side. Dette kan introducere NA-værdier i rækkerne.

### Samtidigt kan det også være, at i dette tilfælde, hvis nogle af informationerne er tilgængelige for nogle af husene, er de ikke nødvendigvis tilgængelige for nogle andre huse, som også bliver webscrapet. I og med, at webscrapingen ikke ser den værdi man søger efter, introduceres derfor en NA-værdi i observationen i stedet for.

### NA-værdier kan på samme tid opstå under data-sorteringen og data-manipulationen.

### Eftersom webscraping er en meget rå måde at indhente data på, resulterer det typisk i en dataframe som er meget rodet, og som kræver en form for oprydning, før dataene er læselige.

### Hvis der ikke tages hensyn, kan det typisk resultere i fejl, som kan ende ud i at omformatere værdier forkert, så de bliver til NA-værdier. Ligeledes kan det også give forkerte værdier – et eksempel for dette kunne være i kolonnen `by` hvor Kongens Lnygby er blevet omformateret til blot at hedde `Kongens`. Dette er sket i et forsøg på at aggregere postnumre i storbyer sammen, så byen ikke længere hedder eks. København K, men derimod blot København.

### Yderligere er der også opstået et problem ift. Postnr/by – hvor postnummeret, for få rækker, er blevet overført til rækken by i stedet for.

### I forhold til hvilke variabler der mangler i data, mangler der primært to vigtige former for data.

### En scrapedate

### En form for unikt ID (enten link, Unikke-ID i bolig-URL el.l.)

### I tilfælde af, at man vil inkorporere en tidsdimension – og eventuelt kigge på om data er forældet, om prisen på gældende huse er ændret fra tidligere scrapedate til nu mv. – vil en scrapedate, gøre dette muligt.

### Samtidigt er det nødvendigt at have en form for unikt ID med i scrapingen. Der er ingen måde af validere eller eftertjekke dataen. Vi har blot fået en CSV-fil og fået at vide ”dette er webscrapet”.

### I dette tilfælde af mange af husene ikke længere tilgængelige på hjemmesiden, men det kræver manuelt arbejde at enkeltvis skulle slå husene op, i stedet for at have et link til hver af husene.

### Samtidigt ved at have et link, kunne man vha. Feature Engineering oprette en ny kolonne ”solgt” – på baggrund af om boligens link returnerer statuskode 200 eller 404. (hvis 404 er huset solgt) – Naturligvis kan det ikke eftertjekkes om huset blot er blevet fjernet eller solgt, men definitionen solgt her, betyder bare at boligen ikke længere er tilgængelig/til salg på hjemmesiden.

### 

## Opgave 1.4 – Research goal

### Med udgangspunkt i Data Science Modellen skal I gøre rede for de skridt, der er blevet taget for at nå frem til CSV-filen. I skal komme med et bud på et “research goal” som kunne have optimeret processen med at fremskaffe data fra boligsiden.

### Et billede, der indeholder skærmbillede, tekst, Website, software Automatisk genereret beskrivelse

### Som det fremstår af CSV-filen, og beskrivelsen af den, er det data som er blevet webscrapet fra hjemmesiden, boligsiden.dk.

### Selve webscrapingen af boligerne ser ud til at kun tage højde for at udtrække informationer som direkte har noget med boligen at gøre, og der fremstår ikke noget klart ’mål’ med dataen, i form af, at der ikke direkte kan dannes en ide om, hvad der skal undersøges på baggrund af de data man har.

### Det kan derfor argumenteres, at der forud for data-retrieval punktet i data-science-modellen ikke nødvendigvis er sat et reserach goal, som styrer de data, som skal udtrækkes, og i tilfælde af, at der var sat et research goal, så har det ikke været et specifikt nok research goal, i og med, at der stadig mangler informationer fra de udtrukkede data, jf. tidligere punkter i opgaven.

### Ved at have prædefineret et konkret research goal, ville det kunne strømline data-retrieval processen ved et klargjort mål for analysen, det ville være nemmere at vide hvilke informationer er vigtige at tage højde for, hvilke er intetsigende/ligegyldige, såvel som sikrer, at alle de nødvendige informationer bliver udtrukket.

### Et eksempel på et research-goal for kunne være:

### Der ønkses at undersøge hvilke faktorer der spiller en rolle i boligens salgspris, såvel som i hvilke boliger som bliver solgt.

### Trods at det er et meget bredt research goal, medfølger research goalet også, at der vil blive udtrukket nær al information som kan trækkes gennem webscraping, herunder også boligernes enkelte link eller boligID så der kan tjekkes hvilke boliger der bliver solgt og hvilke der er til salg i lang tid.

# Opgave 2 – Korrelation og simple lineær regression

## Opgave 2.1 – Beskrivende statistik

### Lav beskrivende statistik for data på boliger til salg via boligsiden.dk fra opgave 1. Vær opmærksom på, hvilke variable I mener kan betragtes som x’er og y i en række af simpel lineære regressioner.

### Som tidligere nævnt, så er CSV-filen bestående af data webscrapet fra boligsiden.dk. Dataene er meget rå, og er derfor endnu ikke klar.

### Vi begynder derfor på 3. step i data-science modellen: Dataforberedelse

### Mere specifikt er det underpunkterne datacleaning og datatransformation, der er fokus på.

#### **Opgave 2.1.1 Klargøring af datasæt.**

### Det første vi lægger mærke til, er at priskolonnen ikke er numerisk, samtidigt med at kolonnen indeholder en mærkelig minimums- og maksimumsværdi

### Priskolonnen har en bolig med en salgspris på 195.000 kr., såvel som en bolig til salg for 51.000.000 kr.

### En maksimumsværdi på 51.000.000 kr. ville ikke i sig selv være underligt, hvis der blev undersøgt f.eks., luksuriøse kæmpevillaer i klampenborg – men meanværdien for priskolonnen er 3.358.327. Det vil altså sige, at i vores eksempel, er en salgspris på 51.000.000 en enorm outlier fra middelværdierne.

### Vi transformerer dermed priskolonnen om til numerisk, såvel som at vi afgrænser dataframen til kun at indeholde rækker hvor pris er mellem 1.500.000 mil. – 15.000.000 mil.

### Fjerner ”kr.” fra rækkerne i priskolonnen og omdanner kolonnen til numerisk.

### 

### Afgrænser dataframen til et interval af salgspriser [1.500.000:15.000.000]

### 

### Vi oplever samtidigt det samme problem med kvmpris kolonnen. Vi ser for denne kolonne en minimumsværdi på 1.412 kr. og en maksimumsværdi på 825.000 kr. Meanværdien er 14.430 kr.

### Vi skaber dermed en afgrænsning for at holde kvmpriserne i et interval, som forsøger både at tage højde for kvmpriser i både billige demografiske områder, såvel som dyre demografiske områder (eksempelvis en lille by i Jylland kontra Klampenborg.)

### Afgrænser dataframen til et interval af kvmpriser [3.000:100.000]

### 

### Yderligere findes der også outliers i Kolonnerne

### Størrelse

### Mdudg

### Grund

### For kolonnen, størrelse, ses en minimumsværdi på 10 kvm., og en maksimumsværdi på 558 kvm., såvel som en meanværdi på 157,3 kvm.

### Vi afgrænser derfor dataframen til et interval af boligstørrelse i kvm. [50:300]

### 

### For kolonnen, mdufg (månedlige udgifter i forbindelse med boligen), ses en minimumsværdi på 1.008 kr., og en maksimumsværdi på 995.000 kr., såvel som en meanværdi på 6.583 kr.

### Vi afgrænser derfor dataframen til at månedlige udgifter i kr. maksimalt må være 20.000 kr.

### 

### Værdierne i kolonnen for grund var ødelagte. Årsagen vides ikke, men det kan observeres, at i kolonnen er det gældende at at punktummet nogle steder agerer som tusindetalsseperator og i andre tilfælde som et komma til decimal-tal. Vi er derfor nødsaget til at skabe en ifelse funktion, som kan opfange et pattern, der kan adskille tilfældende fra hinanden.

### Funktionen ser således ud:

### Et billede, der indeholder tekst, Font/skrifttype, skærmbillede Automatisk genereret beskrivelse

### Funktionen bygger på formodningen om, at hvis der er en talrække på enten 2 & 3, før punktummet, fungerer punktummet som kommaseperator til decimal til, ellers så fungerer punktummet som tusindetalsseperator.

### Eksempler på transformationen af data kan forklares således:

### Disse to boliger har følgende værdi i grund-kolonnen før funktionen: 1 kvm. & 657.000 kvm.

### 

### Efter funktionen har boligerne følgende værdier i grund-kolonnen: 1000 kvm. & 657 kvm.

### 

I min dataframe står grunden som 657.000 i stedet for 657, men bliver transformeret om til 657 hvis jeg overfører rækken til en anden dataframe for at illustrere det her i opgaven. Dette hæmmer forståelsen, men værdierne i den indhentede CSV-fil står som eksempelvis: 657.000, i stedet for det viste 657.

Efter at vi har renset grund-kolonnen afgrænser jeg værdier til et interval af [100:4000] kvm.

****

Grunden til at intervallet er meget bredt, og at der er sat en høj maksimalværdi er, så der er også er taget højde for boliger ude på landet, hvor der typisk medhører store mængder af landareal (ergo stort grund-areal).

I dataframen ser vi også at nogle af postnr værdierne er blevet forskudt over i by-kolonnen, vi indsætter de rigtige postnr. Værdier og indsætter den tilhørende by for postnummeret.



Afslutningsvis i datarensningen fjerner vi rækker med NA-værdier i kolonner, hvor der absolut ikke må være NA-værdier. Disse kolonner er følgende: Pris, Vej, Postnr, Størrelse, Grund, kvmpris.

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede

Automatisk genereret beskrivelse

Afslutningsvis ser vores nye, rensede dataframe således ud:

Et billede, der indeholder tekst, skærmbillede

Automatisk genereret beskrivelse

På baggrund af vores afgrænsning og data-cleaning har vi skåret vores tidligere dataframe på 2552 rækker ned til kun have bestå af 1482 rækker (boliger) nu.

For at drage flere indsigter i datasættet, vælger vi, vha. Feature Engineering, at oprette nogle nye kategori-variabler.

Vi laver følgende kategori-variabler

* Område (demografisk opdeling efter postnumre)
* Husalder (opdeling af husene på baggrund af året huset er opført i)
* Priskategori (opdeling af husene i intervaller af totalprisen)

Udover kategori-variablerne, laver vi også endnu en variabel, husalderiår, som omregner deres opførselsår til husets alder.

#### **Opgave 2.1.2 – Beskrivende statistik**

Følgende beskrivende statistik bliver lavet ved brug af summary() funktionen såvel som sd() funktionen.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Summary(boligsiden.clean) | | | | | | | |
| Variabel | **Min.** | **Q1** | **Median** | **Mean** | **Q3** | **Max.** | **SD** |
| Pris | 1.525.000 Kr. | 2.156.250 kr. | 2.996.500 kr. | 3.787.141 kr. | 4.495.000 kr. | 14.995.000 kr. | 2.352.389 kr. |
| Kvmpris | 3.000 kr. | 12.745 kr. | 17.842 kr. | 21.553 kr. | 26.898 kr. | 87.500 kr. | 13.065,82 kr. |
| Størrelse | 50 | 129 | 153 | 157,2 | 180 | 295 | 40,1 |
| Mdudg | 1.038 kr. | 2.239 kr. | 2.789 kr. | 3.144 kr. | 3.658 kr. | 11.614 kr. | 1.364 kr. |
| Grund | 111 | 717,2 | 835 | 944 | 1021 | 3.862 | 487 |
| Dage til salg | 1 | 6 | 11 | 10,3 | 14 | 20 | 5.7 |

Udover ovenstående tabel over summary() funktionen i R, kan vi yderligere kigge på datasættets fordeling vha. de nyoprettede kategori-variabler.

Et billede, der indeholder linje/række, diagram, Kurve, skærmbillede

Automatisk genereret beskrivelseVi kan se at størstedelen af boligerne i dataframen er boliger som har en salgspris mellem 0 – 3.000.000 kr.

Vi kan ligeledes se, at størstedelen af boligerne til salg er huse som er opført mellem 1901-1970.

Et billede, der indeholder skærmbillede, linje/række, diagram, Kurve

Automatisk genereret beskrivelse

I forhold til demografiske forhold, så kan vi se udfra nedenstående barplot, at boliger på Sjælland og øerne er det sted i Danmark, som har de dyreste huse.

Et billede, der indeholder software, tekst, Multimediesoftware, skærmbillede

Automatisk genereret beskrivelse

Vi kan samtidigt se, at helt nybyggede boliger er den dyreste kategori af boliger, hvorimod at gamle huse er de billigste.

Et billede, der indeholder tekst, diagram, software, Farverigt

Automatisk genereret beskrivelse

**Hvilke uafhængige variabler kan anvendes til en simpel lineære regression?**

Når man skal vurdere hvilke variabler som kan betragtes som uafhængige variabler, er der nogle vigtige kriterier som skal opfyldes.

Først og fremmest, så skal variablen være på en numerisk skala. Det vil sige, at variabler såsom ”vej” eller ”by” ikke kan bruges – og samtidigt skal variablen følge en kontinuerlig skala/udvikling, hvilket hverken ”vej” eller ”by” gør.

Et andet kriterie som skal opfyldes er, at variablen ikke må være en kategori-variabel. Både vej og by er også kategori-variabler. Andre variabler såsom postnummer og antal værelser, kan der argumenteres for at de følger en numerisk kontinuerlig skala, men i og med, at de er kategori-variabler, frasorteres disse.

Samtidigt skal der være en lineær sammenhæng mellem variablerne, hvis variablerne ikke har en lineær sammenhæng sammen, kan det oftes tyde på at en simpel lineære regression ikke er den korrekte model at bruge i tilfældet, og at der skal bruges en anden, og typisk, mere avanceret model.

Samtidigt skal der også kunne skælnes mellem afhængige og uafhængige variabler. Altså hvilke variabler er forklarende og hvilke bliver forklaret, så at sige. Eksempelvis kvmpris og pris er to afhængige variabler, i og med, at de andre variabler er med til at forklare hvorfor prisen stiger ogfalder. F.eks. vil der højst sandsynligvis være en klar sammenhæng mellem pris ~ størrelse. Pris kan dermed ikke bruges som uafhængig variabel.

Afslutnigsvis skal der også være en meningsfuld logisk sammenhæng mellem variablerne. Et eksempel, hvor der ville kunne inddrages en variabel som ikke vil have en logisk sammenhæng kunne være, hvis der i datasættet også var en scraptedate. Der ville ikke være en logisk sammenhæng mellem kvmpris ~ scrapedate, da de ikke har noget med hinanden at gøre.

## Opgave 2.2 - Korrelation

### Hvad er korrelationen mellem m2 og prisen for boliger lagt på boligsiden.dk? Giv en forklaring på begrebet korrelation.

Begrebet korrelation belyser hvor stærk en lineære sammenhæng der er mellem to variable, og hvorvidt at ændringen på en variabel kan forudsige ændringen på den anden variabel. Yderligere viser den retningen af den lineære sammenhæng mellem variablerne. Ved at beregne korrelationen, kan man ud fra korrelationskoefficienten vise hvorvidt der er en positiv lineær sammenhæng (0:1), ingen lineær sammenhæng eller en negativ lineær sammenhæng (0:-1).

For at belyse korrelationen mellem kvm og pris for boliger ud fra det givne datasæt, bruges funktionen cor() som beregner korrelationen mellem to numeriske variabler.

Et billede, der indeholder tekst, skærmbillede, software, Operationssystem

Automatisk genereret beskrivelse

Den beregnede korrelation giver os en værdi på 0.24 (mere specifikt: 0.2374569) – Denne korrelationskoefficient indikerer dermed en svag lineær sammenhæng mellem kvm og pris.

Koefficienten er belyst i korrelationsmatricen.

## Opgave 2.3 – Simple regressioner

### Lav minimum 5 simple regressioner mellem pris pr. m2 og 5 andre variable i CSV-filen fra opgave 1. Giv en forklaring på, hvilken af de fem modeller, der bedst forklarer pris pr. m2. Derudover skal I understøtte jeres 5 modeller med et bud på om, der er sammenhæng (korrelation) mellem jeres 5 udvalgte variable.

### Jeg har valgt at lave 5 simple lineære regressioner mellem kvmpris og følgende, på baggrund af tidligere nævnte kriterier:

### Husalderiår (boligens alder)

### Størrelse (boligens størrelse i kvm)

### Grund (grundens størrelse i kvm)

### Mdudg (de månedlige udgifter i forbindelse med boligen)

### Antaldagetilsalg (liggetid, hvor lang tid har boligen været til salg)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Simple regressioner mellem pris pr. m2 + variabel** | | | | | |
|  | **Kvmpris ~ husalderiår** | **Kvmpris ~ størrelse** | **Kvmpris ~grund** | **Kvmpris ~ mdudg** | **Kvmpris ~ antaldagetilsalg** |
| **Interceptet / afhægig**  **variabel** | 21.821,618 | 26.091,622 | 25.650,89 | -2.089,49 | 18.723,19 |
| **Koefficient** | -4,015 | -28,877 | -4,3421 | 7,5188 | 274,95 |
| **P-værdi for modellen** | 0,6059 | 0,0006336 | 0,0000000003 | 0,000000000002 | 0,0000033 |
| **R^2 for modellen** | -0,0004 | 0,007 | 0,025 | 0,615 | 0,013 |

Vi kan ud fra tabellen se, at alle modeller pånær kvmpris ~ husladeriår er signifikante. Det skal dog påpeges, at de forklarer så lidt af variationen, som kan understreges vha. r^2 værdien for modellerne. Den enestestående undtagelse er modellen for Kvmpris ~ mdudg, hvor vi ser en model som både er signifikant og forklarer en stor del af variationen (ca. 61,5%).

Det skal samtidigt påpeges at nogle af koefficienterne i modellerne er negative. Det vil altså sige, at ved negative koefficienter, så vil modellen påvise, at ved en stigning i den forklarende variabel, så vil Kvmprisen falde.

For at undersøge om der er korrelation mellem de udvalgte variabler (multikollinaritet), udarbejdes en korrelationsmatrice over variablerne.

Et billede, der indeholder tekst, skærmbillede, diagram, Parallel

Automatisk genereret beskrivelse

Som det fremstår af korrelationsmatricen, er der ingen umiddelbar stor korrelation mellem variablerne – men samtidigt belyses korrelationen mellem kvmpris ~ mdudg, hvor vi ser en korrelation på 0,78 (tilsvarende 78%) – Denne høje korrelation belyser igen den høje forklaringsgrad mellem variablerne.

## Opgave 2.4 – Korrelation og lineær regression

### Forklar den teoretiske sammenhæng mellem korrelation og simple lineær regression.

### Korrelation er et begreb som går hånd i hånd med lineære regression, og i særdeleshed simple lineære regressioner.

### En korrelation mellem to variabler beskriver det lineære forhold mellem de to variabler i et interval mellem [-1:1], hvor 1 belyser et korrelationsforhold med en perfekt lineære sammenhæng, 0 belyser et forhold med ingen lineære sammenhæng og -1 belyser en perfekt negativ sammenhæng.

### Sammenhængen mellem korrelation og simple lineære regressioner træder først rigtigt i kraft, når man indblander r^2 som også kaldes for forklaringsgraden.

### Den matematiske betegnelse for korrelationen er `r`, hvor den matematiske betegnelse for forklaringsgraden er `r^2`. Det vil altså sige at korrelationen gangen med hinanden (r \* r) er lig med forklaringsgraden.

### Forklaringsgraden forklarer hvor meget af variationen i y-variablen kan beskrives af x-variablen, hvorimod korrelationen forklarer hældningen. Det vil altså sige at hvis man har en korrelation nær 0, vil koefficienten være enormt lav, hvis man laver en regressionsmodel for de to variabler.

### 

# Opgave 3 – Tilfældigheder og terninger

## Opgave 3.1 – Funktion til terninger

### Lav et script (en funktion) i R-studio, der kan slå med 25.000 terninger. Hvor mange 5’ere har jeres terning slået? Hvad er sandsynligheden, givet jeres resultatet med de 25.000 terninger, for, at jeres script slår en 5’er?

### Nedenstående funktion skaber et simuleret terningekast med et funktionsargument som angiver hvor mange kast man vil lave.

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede

Automatisk genereret beskrivelse

Ved 25.000 kast (terning(25000)), fremkommer en af sum 5 4.111 gange. Det svarer til 16.4% af de 25.000 terningekast.

Et billede, der indeholder tekst, skærmbillede, nummer/tal, Font/skrifttype

Automatisk genereret beskrivelse

Vi kan samtidigt se, at fordelingen af summer skaber en uniform fordeling. Altså en fordeling, hvor der er lige stor sandsynlighed/frekvens af de forskellige resultater, i dette tilfælde, summer.

Et billede, der indeholder tekst, skærmbillede, Rektangel, diagram

Automatisk genereret beskrivelse

### 

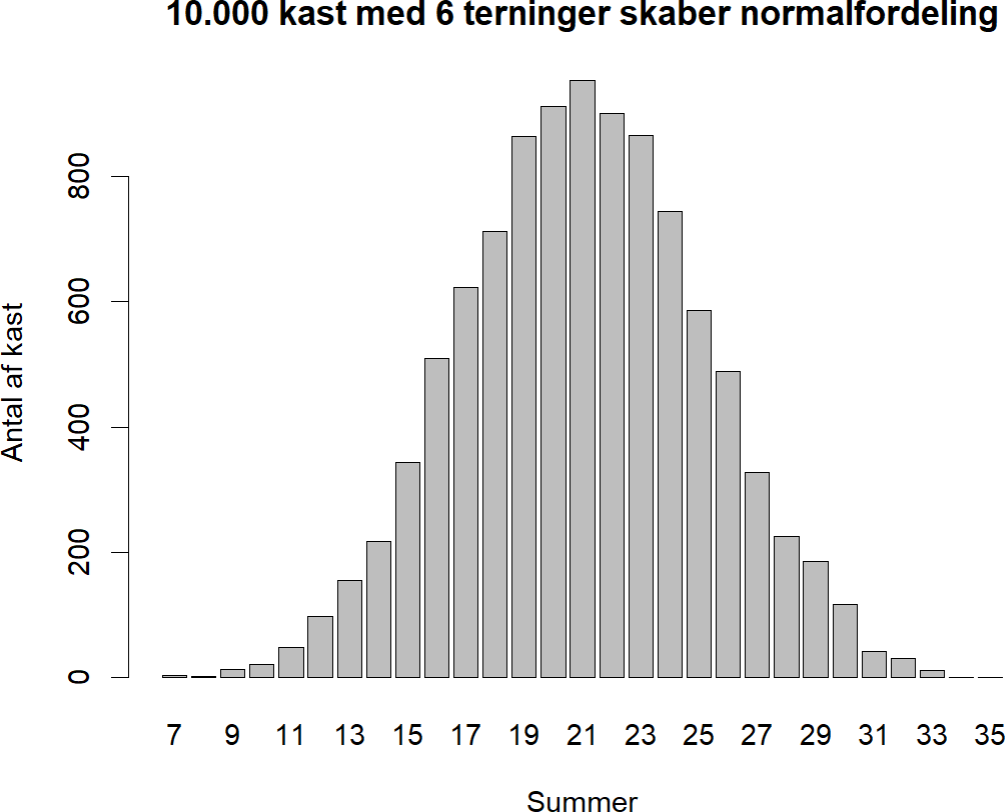
## Opgave 3.2 – Plot I

### Lav et script (en funktion) i R-studio, der kan slå med 6 terninger og vise summen. Slå nu 10.000 gange med de 6 terninger og lav et barplot af jeres resultat. Forklar om jeres resultat giver mening i forhold til jeres funktion.

### Nedenstående funktion laver en simuleret kørsel af kast af 6 terninger ad gangen. Funktions-argumentet definerer hvor mange gange der skal kastes med 6 terninger.

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede

Automatisk genereret beskrivelse



10.000 kast med 6 terninger skaber en normalfordeling, hvorimod 25.000 kast med 1 terning skaber en uniformfordeling.

Grunden til, at 10.000 kast med 6 terninger danner en normalfordeling er, at der er flere kombinationsmuligheder for middelværdierne, kontra de ydre værdier. Så trods af, at der ikke er blevet introduceret nogen form for sandsynlighed, opstår konceptet, kombinatorisk sandsynlighed.

Hvis man derimod kigger på 25.000 terningekast med kun 1 terning, fremstår der en uniformfordeling (lige sandsynlighed for alle summer), og det er på grund af, at der ikke opstår en kombinatorisk sandsynlighed.

### 

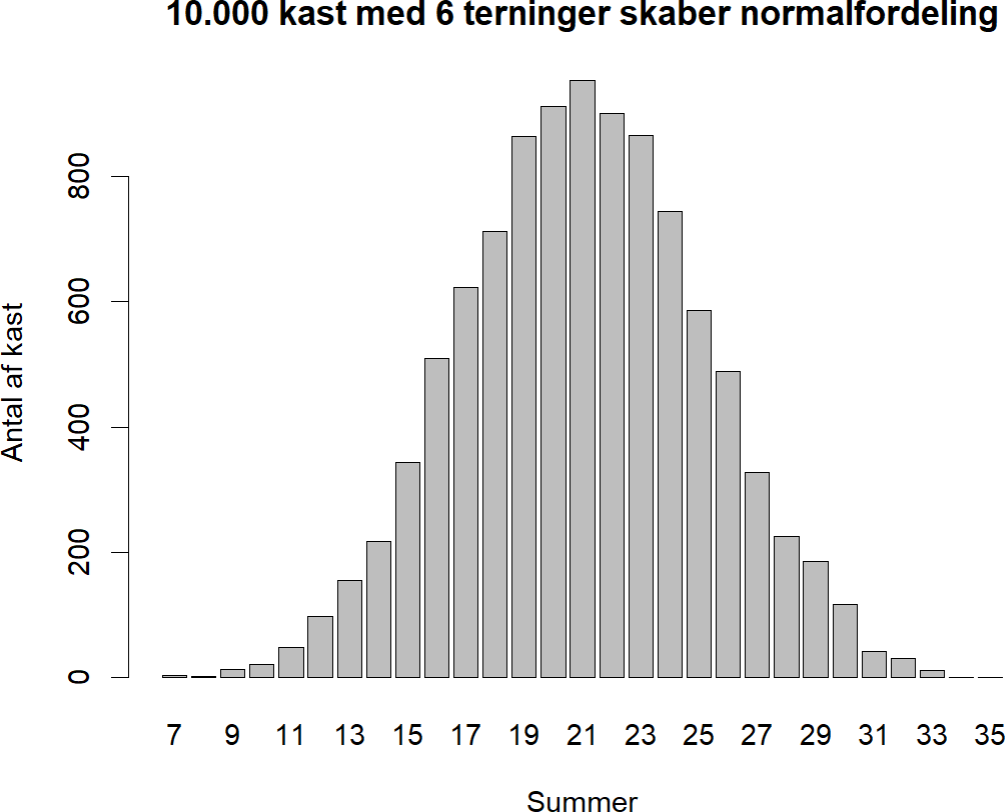
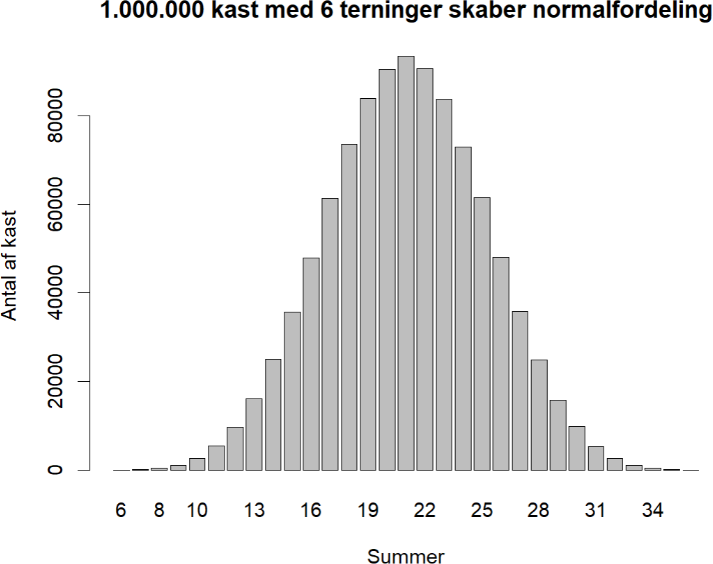
## Opgave 3.3 – Plot II

### Brug jeres script fra 3.2 og slå nu 1.000.000 gange med de 6 terninger. Lav igen et barplot og sammenlign med jeres plot fra 3.2.

Nedenstående kode skaber et barplot over frekvensen ved 1.000.000 kast af 6 terninger.

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelse



Det samme koncept, kombinatorisk sandsynlighed, er gældende ved 1.000.000 kast som ved 10.000 kast. Den eneste forskel er frekvensen af kast, men fordelingen forbliver den samme.

## Opgave 3.4 – Lav dine egne data

### Lav et script i R-studio, der viser en tilfældigt opstillet række af tallene 1, 2, 3, 5, 6. Lav en matrix med to kolonner og fem rækker, hvor den første kolonne skal være tallene 2 til 6 og den anden kolonne skal være jeres tilfældige række af tallene 1, 2, 3, 5, 6.

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelse

Et billede, der indeholder skærmbillede, tekst, nummer/tal, Font/skrifttype

Automatisk genereret beskrivelse

# Opgave 4 – Danskernes forhold til alkohol

## Opgave 4.1 – Hent data

### Hent data fra tabel FU02, alle forbrugsgrupper under 02.1 (alkoholiske drikkevarer) i faste priser for perioden 2000 til 2022 og indlæs i R. Illustrer udviklingen i de enkelte grupper.

### Vi indhenter dataen gennem DKstat API’en.

### Et billede, der indeholder tekst, skærmbillede, Font/skrifttype Automatisk genereret beskrivelse

### Den indhentede dataframe, forbrugdata, indeholder alle komsumgrupper, og ser således ud:

### Et billede, der indeholder tekst, skærmbillede, software Automatisk genereret beskrivelse

### Dataframen sorteres nu, så den kun indeholder konsumgrupperne under 0.2. Samtidigt renses dataframen og dataen bliver klargjort.

### Dataframen ser nu således ud:

### Et billede, der indeholder skærmbillede, tekst Automatisk genereret beskrivelse

### I dataframen har vi følgende overgrupper med tilsvarende undergrupper:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Gruppe | Spiritus og læskedrikke | Øl | Vin | Alternativ til alkohol |
| Undergruppe 1 | Spiritus og likør | Pilsner øl, guldøl | Vin af druer | Vinbaserede drikkevarer og alkoholfri vin |
| Undergruppe 2 | Alkoholiske læskedrikke | Andre alkoholholdige øl | Vin af andre frugter | Øl med lavt alkoholindhold og alkoholfri øl |
| Undergruppe 3 | NA | Øl-baserede drikkevarer | Hedvin | NA |

### Fordelingen af forbruget ser ud som nedenstående plot. Der ses, at vin er den mest populære type alkohol, og udviklingen forholder sig på et nogenlunde stabilt niveau gennem perioden. I modsætning til vin, så er øl noget som danskerne drikker færre og færre af, og ender faktisk mod 2020-2022 at have et forbrug på samme niveau som spiritus og likør, trods at øl førhen var virkelig populært og spiritus og likør førhen ikke blev købt så meget.

### Et billede, der indeholder tekst, diagram, linje/række, Kurve Automatisk genereret beskrivelse

### 

### For at beskrive udviklingen af hver af undergrupperne i de forskellige grupper, dykker vi os nu ned i grupperingerne, for at se hvordan undergrupperne er fordelt i deres respektive grupper.

### Et billede, der indeholder tekst, linje/række, Kurve, diagram Automatisk genereret beskrivelse

### Som det fremstår, så er vin af druer den dominerede vintype, og udgør langt størstedelen af vinforbruget. De to andre vingrupper, vin af andre frugter og hedvin, udgør så lille en del af vinforbruget, at vin af druer næsten er den enestående forbrugsgruppe i sig selv.

### Et billede, der indeholder tekst, diagram, linje/række, Kurve Automatisk genereret beskrivelse

### Som det fremstår, er Pilsnerøl, øl den dominerede øltype og udgør langt størstedelen af forbruget i øl-gruppen. Trods dette, ser vi en kontinuerlig trend i, at forbruget af øl er jævnt faldende gennem hele perioden. Det er samtidigt værd at nævne, at det kan observeres i perioden 2014-2017, at andre alhkolholdige øl overtog en del af forbruget fra pilsnerøl, guldøl kategorien, da man kan se, at de følger direkte modsat udvikling, hvor den ene forbrugsgrupper stiger, mens den anden falder. Det var dog ikke en vedvarende trend i og med, at Pilsnerøl, guldøl hurtigt steg igen, hvorimod andre alkoholholdige øl faldt igen.

### Spiritus og læskedrikke fulgte en meget stabilt udviklende udvikling, jf. 1. figur., gennem hele perioden, men hvis man kigger på underkategorierne, kan man se, at spiritus og likør er den dominerende underkategori, og at alkoholiske læskedrikke udgør en insignifikant del af forbrugsgruppen.

### Et billede, der indeholder tekst, diagram, Kurve, linje/række Automatisk genereret beskrivelse

### I forhold til kategorien, Alternativer til alkohol, opdages en anormal udvikling. Det kan ses, at vinbaserede drikkevarer og alkoholfri vin oplevede en stor stigning i perioden 2010-2016, hvorefter at det faldt ned til kurvens normale niveau igen. Derimod tydes en stigende tendens for forbruget af Øl med lavt alkoholindhold og alkoholfri øl frem mod slutperioden, som kan perspektiveres til stigende forbrugertrends om sundhed i nutidens Danmark.

### Et billede, der indeholder linje/række, Kurve, diagram, tekst Automatisk genereret beskrivelse

## Opgave 4.2 – Korrelation

### Lav en korrelationsmatrix over forbrugsgrupperne under 02.1 og konkludér på resultaterne.

### For at tage et kig på korrelationen mellem grupperne, og herunder også undergrupperne udarbejdes to korrelationsmatricer. Den første korrelationsmatrice er for overgrupperne, hvorefter at den anden korrelationsmatrice er for alle undergrupperne samlet.

### Et billede, der indeholder tekst, skærmbillede, diagram, Rektangel Automatisk genereret beskrivelse

Ifølge ovenstående korrelationsmatrice, fremstår en betydelig negativ korrelation på -0.49 mellem spiritus og læskedrikke vs. Øl. Tolkningen af sådan en korrelationskoefficient indikerer et forhold, hvor en stigning i en af grupperne vil resultere i et fald i den anden. Ligeledes ses samme korrelationsforrhold mellem Øl og alternativer til alkohol, hvor der er en korrelationskoefficient på -0.53. Udover dette opstår der kun korrelationsforhold med så små koefficienter at de ikke erklæres signifikante.

For at undersøge forholdene dybere, tages også et kig på korrelationerne mellem undergrupperne.

*Et billede, der indeholder tekst, skærmbillede, diagram

Automatisk genereret beskrivelse*

Tager vi udgangspunkt i det negative korrelationsforhold mellem spiritus og læskedrikke vs. Øl, påpeges der ifølge ovenstående korrelationsmatrice, at det ikke er alle undergrupperne som er negativt korreleret med hinanden, derimod opleves der faktisk et let positivt korrelationsforhold mellem undergrupperne for øl, Andre alkoholholdige øl & øl-baserede drikkevarer og spiritus og likør & alkoholiske læskedrikke . Det negative korrelationsforhold bliver derimod belyst af undegruppen, Pilsnerøl, guldøl, hvor der igen fremstår et stærkt negativt korrelationsforhold.

Det skal her medtænkes at Pilsnerøl, guldøl er den dominerede undergruppe i gruppen, Øl, og at det derfor samtidigt er denne undergruppe som er den drivende faktor i det negative korrelationsforhold, da de andre undergrupper i gruppen, Øl, bliver nærmest ubetydeliggjort grundet forbrugsforskellen.

Det andet negative korrelationsforhold vi opdatede i den første korrelationsmatrice, bliver igen belyst gennem korrelationsmatricen for undergrupperne. Der opleves nøjagtig det samme forhold mellem undergrupperne, som ved forrige undersøgte korrelationsforhold. Det vil altså sige, at ud af undergrupperne for øl, er det kun undergruppen, Pilsnerøl, guldøl, som har en negativ korrelation med undergrupperne for alternativer til alkohol, hvorimod de øvrige undergrupper i gruppen, Øl, faktisk har et positivt korrelationsforhold.

Samtidigt er det vigtigt at påpege, at der også opleves negative korrelationsforhold mellem undergrupperne i den samme gruppe. Et eksempel på dette, ville være de negative korrelationsforhold mellem Pilsnerøl, guldøl og de øvrige undergrupper i gruppen, Øl. Dette korrelationsforhold indikerer at forbrugerne ikke køber begge slags øl samtidigt, men derimod vælger en af dem, da det negative korrelationsforhold påpeger at ved stigning af en af undergrupperne, så falder de øvrige.

## Opgave 4.3 – Kritisk tænkning

### Forhold jer kritisk til de indlæste data når I reflekter over jeres resultater.

De indlæste data er indtrukket fra Statistikbankens database, gennem deres API, DKstat.

Det medfører dermed, at det er ekstern data, som skaber nødvendigheden for at forholde sig kritisk til dataene, i og med, at vi ikke selv har indhentet det.

Der tages udgangspunkt i Danmarks Statistiks egen dokumentation for dataindhentningen.

Undersøgelsen/dataindhentingen er en stikprøveundersøgelse, hvor der bliver adspurgt et hvis antal husstande ud af Danmarks i alt 2.800.000 husstande.

Ud af de adspurgte husstande har kun 13 pct. Af husstandene valgt at deltage, dette svarer til i alt 2.301 husstande. – Dette skaber naturligvis en usikkerhed omkring dataene, og i særdeleshed for detaljerede forbrugsgrupper (såsom undergrupperne for alkohol).

Udover en lille stikprøve, så dokumenterer Danmarks Statistik selv, at man skal være forsigtig og tage højde for dataens pålidelighed, da de adspurgte husstande selv kan ’pynte’ på deres tal. Eksempelvis så er ikke særlig positivt set på hvis man har et kæmpe forbrug af alkohol, og de kan dermed selv dokumentere, at de ikke bruger lige så meget, som de egentlig gør.

Yderligere så blandt de 13 pct. Som deltog i undersøgelsen, så var en større andel etnisk danske, en større andel var pensionister, folk på arbejdsmarkedet, lønmodtagere grund niveau end i befolkningen. Der er en større andel husstande med personer fra 60 år og op, en mindre andel af unge og husstande hvor hovedindkomstmodtageren har en grundskole.

Det vil altså sige, at der kan forekomme en form for mispræcision i datasættet, da der ikke er en ligevægtet fordeling af husstandstyper, hvor der eksempelvel er en stor overvægt at pensionister kontra unge husstande.

# Opgave 5 – Dataframes

## Opgave 5.1 – Månedlige observationer

### Lav en 36 x 3 dataframe med kolonnenavne ”Klasse”,”Uge”,”Score”. Første kolonne skal fyldes med A,B,C,D så der startes med 9 A’er, derpå 9 B’er osv. Anden kolonne skal fyldes med tallene 1 til 9, der gentages for hvert bogstav. Sidste kolonne skal fyldes med observationer. Det er op til jer, hvilke værdier I vil putte i framen.

### Nedenstående kodee skaber en dataframe med opgavebskrivelsesnskriterier. ”Uge” er blevet lavet om til ”Måned”, grundet en `typo` i opgaven. Opgave 5.1 beder os nemlig om at udtrække hver 3. række for at omdanne dataframen til kvartalsvis-basis (ergo burde det være måned fremfor uge).

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelse

Den producerede dataframe ser ud som følgende:

Et billede, der indeholder tekst, skærmbillede

Automatisk genereret beskrivelse

## Opgave 5.2 – Kvartalsvise observationer

### I skal lave en ny dataframe, der er 9x3 og bygger på den dataframe I lavede opgave 5.1. I skal tage udgangspunkt i framen fra opgave 5.1 og loope igennem. I loopet skal I hver tredje gang lave en dataframe 1x3 dataframe med samme navne som i 5.1.

### Indholdet skal være som følger: 1 element og 2 element henter I fra 5.1-framen. Det sidste element skal være gennemsnittet af de forrige tre observationer.

### Nedenstående kode skaber en ny dataframe, hvor hver 3. række fra den oprindelige dataframe i 5.1, indsættes. For kolonnen `score` indsættes ikke bare hver 3. række, men derimod gennemsnittet for hver 3. række (ergo række 1,2,3 sammen, række 4,5,6 sammen osv).

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelse

Den producerede dataframe ser ud some følgende: Vi ser her, at vi trækker hver 3. række fra den oprindelige dataframe (jf. rownames)

Et billede, der indeholder tekst, skærmbillede, nummer/tal

Automatisk genereret beskrivelse

## Opgave 5.3 - Pivot

### I skal nu konvertere denne nye dataframe til en ny dataframe som har følgende navne på kolonnerne: ”Uge”,”A”,”B”,”C”,D” og rækkerne indeholder de gennemsnit som I har beregnet.

### For at omformatere dataframen til et wide-format, bruges funktionen `Pivot\_wider` fra pakken `Tidyr`.

### Tidyr er en pakke, som har indbyggede funktioner, som streamliner og gør det nemmere at sortere, manipulere og generelt rydde op i dataen (hence the name the tidy).

### Pivot\_wider er en funktion som transformere dataen, og bruges i typiske tilfælde, hvis man har en kategori-variabel i en kolonne (i vores eksempel, klasse), hvor der er gentagende observationer for hver unikke observation. Ved at bruge Pivot\_wider omdanner man pr. unikke observation om til kolonnerne, hvor de tilsvarende værdier, som specificeres i funktionens argument, indsættes i kolonnen for den unikke observation/kategori, jf. nedenstående billeder.

**

Den transformerede dataframe ser således ud:

*Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, nummer/tal

Automatisk genereret beskrivelse*