Introduction to Data Science - Winter 2019

End-term assignment

Mads Emil Marker Jungersen, studienummer: 201906249

Dec 05, 2019

Introduction

In this report we will analyse the market concentration af Aldi stores in certain areas in Denmark. The main focus will be a description of the dataset and the explanatory variables with the use of descriptive tools as well as graphically illustrations showing how the different variables are distributed in different areas of Denmark. Furthermore the report will contain a description of the linear relationship between the explanatory variables in the dataset using correlation-matrices and scatter plots. In the end the report touches areas of linear regression trying to build a simple linear model that best explains/fits our data. Doing that we will explain terms as the R-squarred and the adjusted R-squarred. We will later use that linear model to see how well it predicts the number of Aldi-stores in a given area, for instance by calculating the average model error/residual. Finally we will split up our dataset in what we call a "Training dataset" and a "Test dataset" finding the best linear model on our Training dataset and use that for predictions on the Test dataset (out of sample). To test how well our linear model do, we will manually calculate the R-squarred.

Data description

Distribution of Aldi Stores in Denmark

First of all we are going to load the dataset 'ALDI.rds' which we made in week 8-9 and contains all Aldi Stores in Denmark. We made that dataset using the 'google_maps' api and the 'google_places()' function.

```
Aldi <- readRDS(file = "Input_for_report/ALDI.rds")
Aldi %>%
  head() %>%
  kable("latex", booktabs = T, digits = 3,
        col.names = c("Name", "Address", "Lat", "Lng"))%>%
  kable_styling(latex_options = c("striped"))
```

Name	Address	Lat	Lng
ALDI	1, Dannebrogsgade 58, 9000 Aalborg, Denmark	57.054	9.906
ALDI	4, Assensvej 2, 5600 Faaborg, Denmark	55.102	10.232
ALDI	Akacietorvet 2, 3520 Farum, Denmark	55.809	12.358
ALDI	Alte Bahnhofstraße 23, 23769 Fehmarn, Germany	54.479	11.073
ALDI	Apenrader Str. 111, 24939 Flensburg, Germany	54.809	9.421
ALDI	Apotekergade 2, 4840 Nørre Alslev, Denmark	54.899	11.878

As we see the dataset contains some stores in Germany, as we are not intereseted in. Because all addresses contains the country of the store, we can use the 'string_detect()' togehter with the 'filter()' function to select only stores in Denmark.

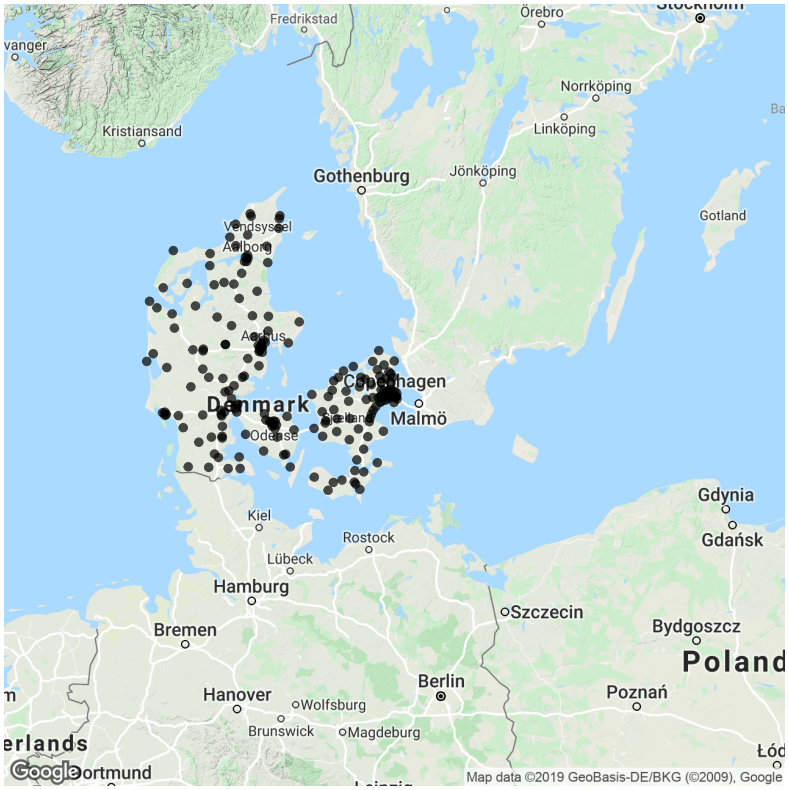
```r
Aldi <- Aldi %>%
  filter(str_detect(address, 'Denmark'))

Aldi %>%
  head() %>%
  kable("latex", booktabs = T, digits = 3,
        col.names = c("Name", "Address", "Lat", "Lng"))%>%
  kable_styling(latex_options = c("striped"))
```

Name	Address	Lat	Lng
ALDI	1, Dannebrogsgade 58, 9000 Aalborg, Denmark	57.054	9.906
ALDI	4, Assensvej 2, 5600 Faaborg, Denmark	55.102	10.232
ALDI	Akacietorvet 2, 3520 Farum, Denmark	55.809	12.358
ALDI	Apotekergade 2, 4840 Nørre Alslev, Denmark	54.899	11.878
ALDI	Bakkenborgvej 2B, 4230 Skælskør, Denmark	55.254	11.296
ALDI	Baltorpvej 22, 2750 Ballerup, Denmark	55.729	12.350

Now that we have a dataset only containing Aldi stores in Denmark, we are going to look how the Aldi stores are distributed across Denmark. To do that, we create a plot using the 'get_map()' function to get a view of Denmark, and then plot the seperate Aldi stores on the map. Furthermore we are going to select a Aldi-store as centrum for our map.

```r
register_google(key = keyD)

# Use store in Gothersgade Copenhagen as centrum
use <- which(str_detect(Aldi$address, "Gothersgade"))

# Generate points of the Aldi-stores to plot on our map.
spec_point    <- Aldi[use,]
other_points  <- Aldi[-use,]

# Save our map with centrum in Gothersgade.
spec_map <- get_map(location = c(lon = Aldi$lng[use], lat = Aldi$lat[use]), zoom = 6,
                    scale = 4, maptype ='terrain', color = 'color')

# Plotting the stores on the map, and save it in the variable 'mapPoints'.
mapPoints <- ggmap(spec_map) +
  geom_point(aes(x = lng, y = lat), data=other_points, alpha=0.7, size = 1) +
  geom_point(aes(x = lng, y = lat), data=spec_point, alpha=0.7, size = 2) +
  labs (title = "Aldi Stores in Denmark")+
  theme(axis.title.x=element_blank(), axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.title.y=element_blank(), axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        plot.title = element_text(hjust = 0.5),
        legend.position="right",
        axis.title=element_text(size=18,face="bold")
  ) +
  NULL

plot(mapPoints)
```

Aldi Stores in Denmark



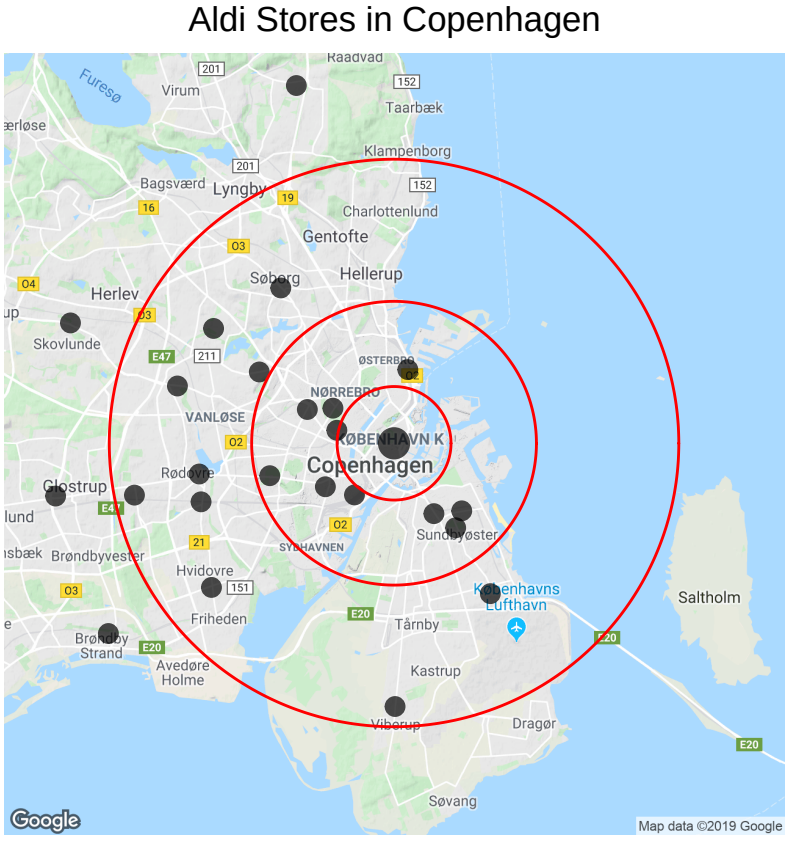
As we see the Aldi stores are more or less evenly distributed across Denmark, with some clusters around the greather cities as Aarhus and Copenhagen and in "Trekantsområdet". We are now going to take a closer look to how the Aldi stores are distributed in Aarhus and Copenhagen. We already have Copenhagen as centrum for our map, so we'll start here.

Distribution of Aldi Stores in Copenhagen

We are more or less going to reuse the previous plot, where we change the zoom attribute. Furthermore we are going to create 3 circles with the 'make_circles()' function, which we have defined in '/input_for_report/Functions/my_functions_v01.R'. The function takes in the Address, Latitude and Longitude as well as the desired Radius as input. The centrum in our case is the Aldi store in Gothersgade and we are going to use 2,5,10 km as radiuses.

```
# Store in Gothersgade as centrum
use <- which(str_detect(Aldi$address, "Gothersgade"))

# Create circles with radius 2, 5 and 10 km
myCircles5  <- make_circles(Aldi[,c(2:4)], 5)
myCircles10 <- make_circles(Aldi[,c(2:4)], 10)
myCircles2 <- make_circles(Aldi[,c(2:4)], 2)

# Create circles with centrum in the store in Gothersgade
spec_circle5  <- myCircles5  %>% filter(ID==Aldi$address[use])
spec_circle10 <- myCircles10 %>% filter(ID==Aldi$address[use])
spec_circle2 <- myCircles2 %>% filter(ID==Aldi$address[use])

# Save our map with centrum in Gothersgade.
```

```
spec_map_cph <- get_map(location = c(lon = Aldi$lng[use], lat = Aldi$lat[use]), zoom = 11,
                    scale = 4, maptype ='terrain', color = 'color')

# Plotting the stores on the map, and save it in the variable 'mapPoints_cph'.
mapPoints_cph <- ggmap(spec_map_cph) +
  geom_point(aes(x = lng, y = lat), data=other_points, alpha=0.7, size = 3) +
  geom_point(aes(x = lng, y = lat), data=spec_point, alpha=0.7, size = 5) +
  geom_polygon(data = spec_circle5, aes(lon, lat, group = ID), color = "red", alpha = 0) +
  geom_polygon(data = spec_circle10, aes(lon, lat, group = ID), color = "red", alpha = 0) +
  geom_polygon(data = spec_circle2, aes(lon, lat, group = ID), color = "red", alpha = 0) +
  labs (title = "Aldi Stores in Copenhagen")+
  theme(axis.title.x=element_blank(), axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.title.y=element_blank(), axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        plot.title = element_text(hjust = 0.5),
        legend.position="right",
        axis.title=element_text(size=18,face="bold")
  ) +
  NULL

plot(mapPoints_cph)
```



As we can see from the plot, the Aldi-store in Gothersgade is the only Aldi store within a radius of 2 km. Furthermore there is 10 other Aldi stores within a radius of 5 km from the Aldi store in Gothersgade. We also see that almost every Aldi store in the Copenhagen area (except 4 stores) is within the radius of 10 km

of the store in Gothersgade. In total we have 25 Aldi-stores in the Copenhagen Area, whereas $\frac{11}{25} \approx 44\%$ are within the radius of 5 km of the one in Gothersgade.

Because there is a limited number of Aldi-stores in Copenhagen (compared to for instance Netto) we were able to count the Stores within the different radiuses, but in the case where the number of stores gets to high, we can generate a tibble containing the crow flight distances from the store we use as centrum to all the other stores. Here we are going to use the 'distm()' function in a for loop:

```
# Creating an empty list with nrow(Aldi)-elements.
All_distance <- vector("list", nrow(Aldi))

# Creating a for loop, looping through all coordinates for the Aldi stores
for (p in 1:nrow(Aldi)){
  temp_res <- distm(c(Aldi$lng[use], Aldi$lat[use]), c(Aldi$lng[p],Aldi$lat[p]))/1000

  # Save results in a tibble
  res <- tibble(
    Centrum = Aldi$address[use],
    Destination = Aldi$address[p],
    Distance = temp_res[1]

  )
  # Save our tibble in our list
  All_distance[[p]] <- res
}


## We then combine the entries of our list into one tibble,
## by iteratively taking the union with the next entry of the list

Final_distances_cph <- All_distance[1][[1]]


for (p in 2:length(All_distance)){
  Final_distances_cph <- union(Final_distances_cph,All_distance[p][[1]])
}

## What does our tibble look like
Final_distances_cph %>%
  head() %>%
  kable("latex", booktabs = T)%>%
  kable_styling(latex_options = c("striped"))
```

Centrum	Destination	Distance
Gothersgade 52, 1123 København, Denmark	1, Dannebrogsgade 58, 9000 Aalborg, Denmark	225.08082
Gothersgade 52, 1123 København, Denmark	4, Assensvej 2, 5600 Faaborg, Denmark	162.37421
Gothersgade 52, 1123 København, Denmark	Akacietorvet 2, 3520 Farum, Denmark	19.88918
Gothersgade 52, 1123 København, Denmark	Apotekergade 2, 4840 Nørre Alslev, Denmark	98.03985
Gothersgade 52, 1123 København, Denmark	Bakkenborgvej 2B, 4230 Skælskør, Denmark	94.32119
Gothersgade 52, 1123 København, Denmark	Baltorpvej 22, 2750 Ballerup, Denmark	15.50790

With that dataset we can simply use the 'filter()' function and the 'nrow()' function to find the amount of stores within a 10 km radius of centrum:

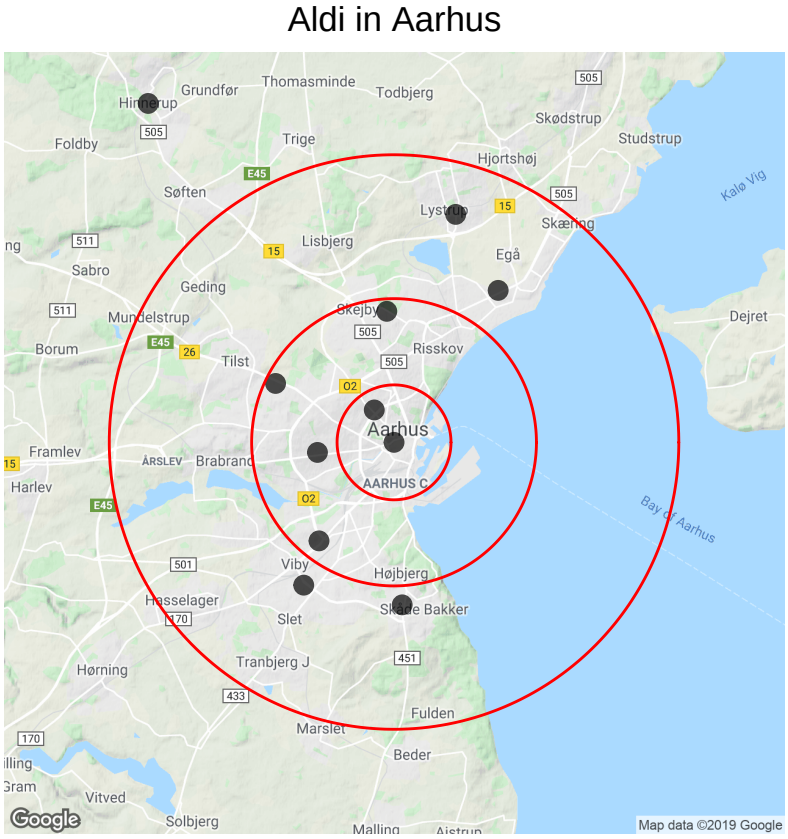
```
Final_distances_cph %>%
  filter(Distance <= 10) %>%
  nrow()
```

```
## [1] 21
```

So there is 21 Aldi stores within a 10km radius (crow flight) of our centrum. The above function might not feel necessary, but becomes very handy and easily to change to, for instance, finding the distance to different nearest competitors, which is (unfortunately) not within the scope of this assignment. As for now, we are going to take a closer look on how the Aldi stores are distributed in Aarhus.

Distribution of Aldi Stores in Aarhus

We are going to reuse the previous plot for Copenhagen only changing the centrum Store to the Aldi store in 'Grønnegade' in Aarhus, so in that sense we are not going to show the code generating the plot:



As we se on the plot there are a total of 11 Aldi Stores in Aarhus, whereas 1 is within a radius of 2 km of our centrum store, 4 more Aldi stores are within a radius of 5 km of our centrum store and the last 5 stores are more than 5 km in crow flight distance away from our centrum store. Out of the total amount of 11 Aldi stores we have $\frac{6}{11} \approx 55\%$ of them within a radius of 5 km to our centrum store, which is around 10%-points more than in Copenhagen. Because of the relatively small amount of Aldi-strores in Aarhus, it would be a litle to comprehensive to implement the function callculating the distances as we did for the Copenhagen case.

Number of Aldi Stores pr. 10.000 capita in different areas

We are now going to evaluate the number of Aldi stores pr. Capita in the 80 areas of Denmark defined in the Dataset 'Discount_Concentration.rds'. We made that Dataset by extracting data from 'Danmarks Statistik'

(DST) using their API, and adding it to the dataset containing our discount stores. The R-scripts doing so are in the R_Scripts folder within this zip-file, where the R-script called '3_Add_DST_data_v01.r' are the one merging all the different datasets to the desired dataset 'Discount_Concentration.rds'. We are now going to load in the dataset 'Discount_Concentration.rds'.

```
Disc_conc <- readRDS(file = "Input_for_report/Discount_Concentration.rds")
```

Because we are only interested in the Aldi stores, we are going to sort out the other discount stores, using the 'select()' function. Furthermore in this case we are only intereseted in the number of stores per capita in the different areas, so we select the columns of interest and using the 'Mutate()' function to generete the number of Aldi-stores pr 10.000 capita.

```
Disc_conc_Aldi_Count <- Disc_conc %>%
  mutate(store_pr_cap = (ALDI/Total)*10000)%>%
  select(c(ALDI, Område, Region, Total, store_pr_cap))%>%
  arrange(desc(store_pr_cap))


Disc_conc_Aldi_Count %>%
  head()%>%
  kable(col.names = c("Nr. Aldi Stores", "Area", "Region",
                      "Population", "Stores pr. Cap."),
        digits = 3, "latex", booktabs = T)%>%
  kable_styling(latex_options = c("striped"))
```

Nr. Aldi Stores	Area	Region	Population	Stores pr. Cap.
6	Brøndby	Hovedstaden	35104	1.709
7	Ikast-Brande	Midtjylland	41664	1.680
5	Sorø	Sjælland	29860	1.674
5	Rebild	Nordjylland	30073	1.663
6	Jammerbugt	Nordjylland	38440	1.561
4	Allerød	Hovedstaden	25729	1.555

It is import to mention that our dataset only contains areas in Denmark where all of the 4 discount stores Netto, Fakta, Aldi, Rema 1000 and Lidl are present. That being said, the above table show the top 6 areas of our dataset containing the most Aldi Stores per 10.000 capita. Brøndby is the city containg the most stores per capita, with around 1.7 Aldi stores pr 10.000 capita. The below table show the 6 areas in dataset containg least Aldi stores pr. capita. It shows that Randers with around 0.1 Aldi stores pr. 10.000 capita is the city in our dataset with least Aldi stores pr. capita.

Nr. Aldi Stores	Area	Region	Population	Stores pr. Cap.
2	Roskilde	Sjælland	87588	0.228
1	Skive	Midtjylland	46151	0.217
1	Varde	Syddanmark	50237	0.199
1	Lyngby-Taarbæk	Hovedstaden	56088	0.178
1	Gentofte	Hovedstaden	75055	0.133
1	Randers	Midtjylland	98009	0.102

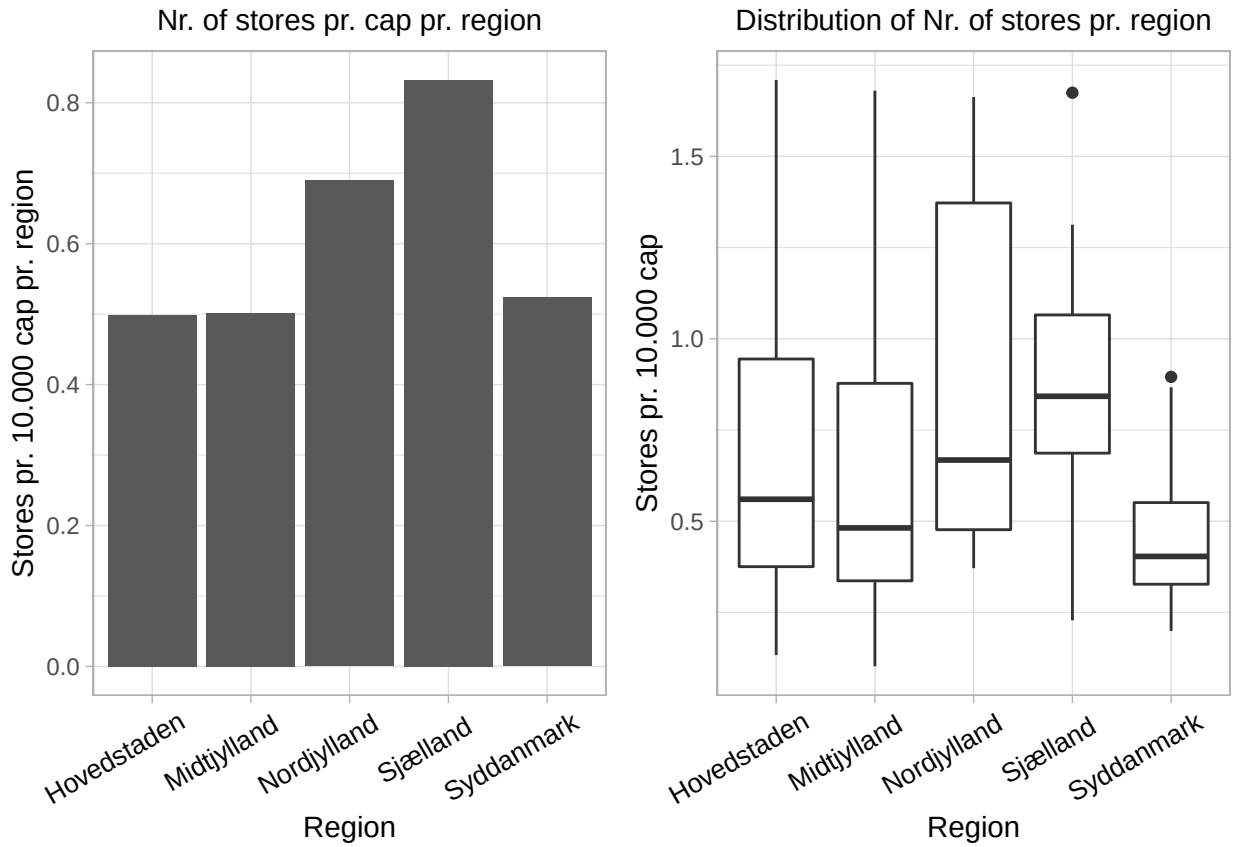
To get a feeling of how the number of Aldi stores pr 10.000 capita is distributed across all 80 areas in our dataset, we can call the 'summary()' function:

```
Store_pr_cap_sum <- t(as.data.frame.complex(summary(Disc_conc_Aldi_Count$store_pr_cap)))

rownames(Store_pr_cap_sum)<- c("Store pr. 10.000 capita")
Store_pr_cap_sum %>%
  kable(digits = 3, "latex", booktabs = T)%>%
  kable_styling()
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Store pr. 10.000 capita	0.102	0.386	0.587	0.705	0.887	1.709

To shortly conclude on the table, it gives us the minimum amount of Aldi stores pr. 10.000 capita (In Randers), the maximum amount (in Brøndby), but also the 1st and 3rd quantile and the mean and median of the number of Aldi stores pr. 10.000 capita. Across the 80 areas there are around 0.7 Aldi stores pr. 10.000 capita on average. To change our scope a bit, we can also see which of the 5 regions in our dataset has the highest amount of Aldi stores pr. cap and how the distribution is. Below is bar-plot and a box-plot containg this information.

```
Count_bar <- Disc_conc_Aldi_Count %>%
  group_by(Region) %>%
  summarise(Store_pr_cap_reg = sum(ALDI)*10000/sum(Total)) %>%
  ggplot(mapping = aes(x= Region, y= Store_pr_cap_reg))+
  geom_col(stat="identity")+
  labs(y = "Stores pr. 10.000 cap pr. region",
       title = "Nr. of stores pr. cap pr. region")+
  theme_light()+
  theme(axis.text.x=element_text(color = "black", size=10, angle=30, vjust=.8, hjust=0.8),
        plot.title = element_text(size=11, hjust = 0.5))



Count_box <- Disc_conc_Aldi_Count %>%
  ggplot(mapping = aes(x= Region, y= store_pr_cap))+
  geom_boxplot()+
  labs(y = "Stores pr. 10.000 cap",
       title = "Distribution of Nr. of stores pr. region")+
  theme_light()+
  theme(axis.text.x=element_text(color = "black", size=10, angle=30, vjust=.8, hjust=0.8),
        plot.title = element_text(size=11, hjust = 0.5))

plot_grid(Count_bar,Count_box, nrow = 1)
```



The above bar plot shows that pr. region, Region Sjælland has the higest number of stores pr 10.000 capita with around 0.8 stores pr. 10.000 capita. Region Nordjylland has the second highest number of Aldi stores pr. capita with around 0.7 Aldi stoers pr. capita. The three remaining regions have around 0.5 aldi stores pr. 10.000 capita.

The boxplot however reveals that within each region there is some variation on the distribution of Aldi stores pr. capita, especially in Region Hovedstaden and Mdtjylland ranging from around 0.12 to 1.7 stores pr. 10.000 capita. Region Syddanmark is the region with the lowest variation ranging from around 0.20 to around 0.80 stores pr. 10.000 capita.

Description of explanatory variables in the dataset and how they varies in the different areas

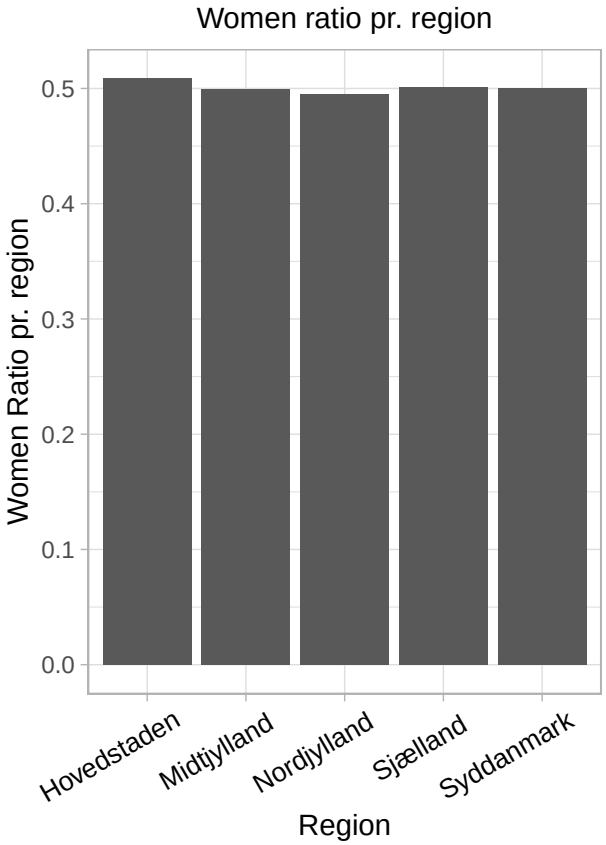
This section will contain a short description of each of the explanatory variables in the dataset, with plot and tables to see how the variables are distributed across the regions/areas.

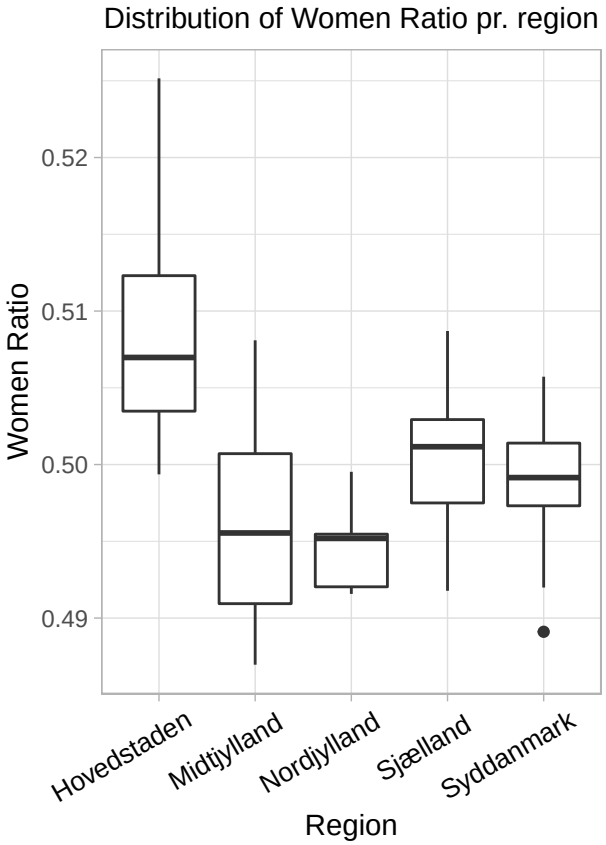
Kvinder

The explanatory variable 'Kvinder' contains the number of women in the area in the 3rd quarter of 2019. This explanatory variable is collected in the '2_Get_population_FOLK1A_v01.r' R-script using the DST-Api more precisely the 'dst_meta()' and 'dst_get_data()'functions. The amount of women dosen't say much, and it is probably more insightfull to look at the ratio of women, so we create a new column containg the ratio of women, therenext finding the 5 areas with the highest women ratio and the 5 areas with the lowest women ratio, aswell as a bar-plot and a boxplot containing the women ratio for the 5 regions and the distribution of the women ratio in the 5 regions. In the end we use 'summary()' function to see the distribution of the women ratio across all 80 areas. The tables and plot is done in the same way as we did in the analysis of the number of stores pr. cap, so in that sense we will not print the code:

Nr. Aldi Stores	Area	Region	Population	Women Ratio
4	Frederiksberg	Hovedstaden	103725	0.525
1	Gentofte	Hovedstaden	75055	0.523
1	Herlev	Hovedstaden	29000	0.515
4	Fredensborg	Hovedstaden	40867	0.515
2	Helsingør	Hovedstaden	62664	0.514
3	Rudersdal	Hovedstaden	56556	0.513

Nr. Aldi Stores	Area	Region	Population	Women Ratio
4	Ringkøbing-Skjern	Midtjylland	56883	0.491
6	Hedensted	Midtjylland	46726	0.490
3	Lemvig	Midtjylland	19938	0.490
1	Vejen	Syddanmark	42776	0.489
7	Ikast-Brande	Midtjylland	41664	0.488
2	Norddjurs	Midtjylland	37463	0.487





	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Women Ratio	0.487	0.496	0.501	0.501	0.505	0.525

As the above tables and plotes reveal the women ratio seems quite evenly distributed across the 80 areas. The lowest women ratio of around 48.7% is in Norddjurs, while the highest women ratio of around 52.5% is

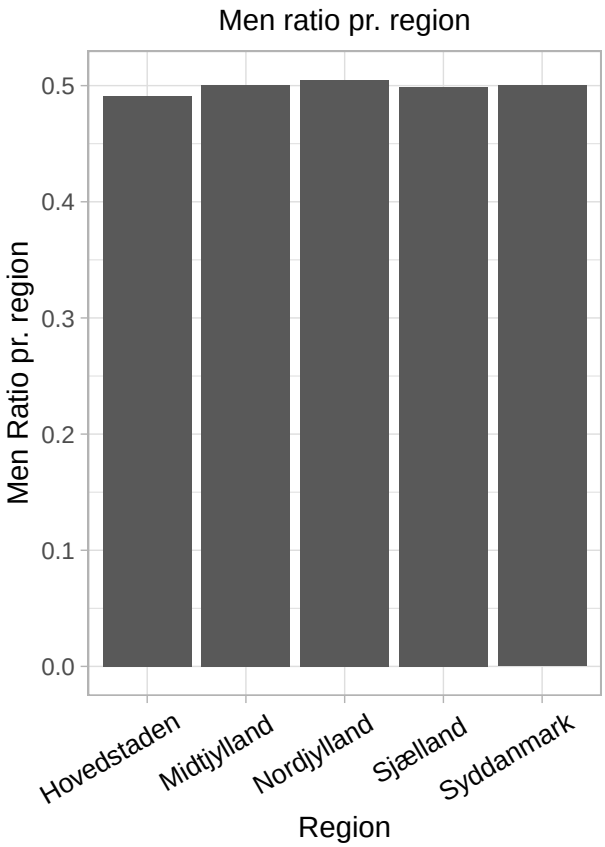
in Frederiksberg. The overall median and mean is around 50%. Eventhough the scatterplot might show some variation within each region, the scale on the y-axis is low, so actually the variation is quite low as well.

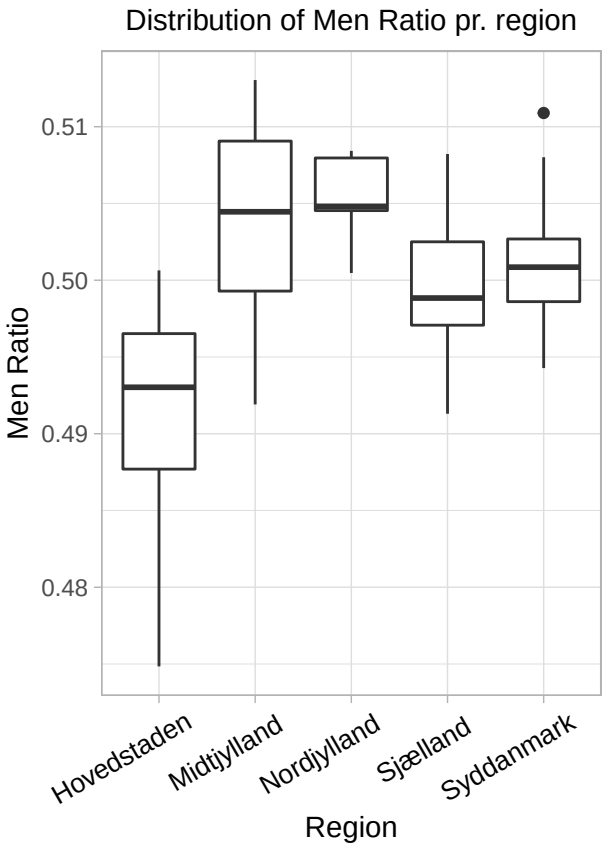
Mænd

The explanatory variable 'Mænd' contains the number of men in the area in the 3rd quarter of 2019. This explanatory variable is collected in the '2_Get_population_FOLK1A_v01.r' R-script using the DST-Api more precisely the 'dst_meta()' and 'dst_get_data()'functions. We chose again to look at the ratio of men. We create a new column containg the ratio of men, therenext finding the 5 areas with the highest men ratio and the 5 areas with the lowest men ratio, aswell as a bar-plot and a boxplot containing the men ratio for the 5 regions and the distribution of the men ratio in the 5 regions. In the end we use 'summary()' function to see the distribution of the men ratio across all 80 areas:

Nr. Aldi Stores	Area	Region	Population	Men Ratio
2	Norddjurs	Midtjylland	37463	0.513
7	Ikast-Brande	Midtjylland	41664	0.512
1	Vejen	Syddanmark	42776	0.511
3	Lemvig	Midtjylland	19938	0.510
6	Hedensted	Midtjylland	46726	0.510
4	Ringkøbing-Skjern	Midtjylland	56883	0.509

Nr. Aldi Stores	Area	Region	Population	Men Ratio
3	Rudersdal	Hovedstaden	56556	0.487
2	Helsingør	Hovedstaden	62664	0.486
4	Fredensborg	Hovedstaden	40867	0.485
1	Herlev	Hovedstaden	29000	0.485
1	Gentofte	Hovedstaden	75055	0.477
4	Frederiksberg	Hovedstaden	103725	0.475





	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Men Ratio	0.475	0.495	0.499	0.499	0.504	0.513

The above plot and tables show more or less the same as the Women Ratio, which makes sense because the number of men and the number of women represent the total amount of people living in the are.

Danmark

The variable 'Danmark' is the total number of danish citizens living in the area in the 3rd quarter of 2019. It is collected in the '2_Get_population_FOLK1B_v01.r' using the same method as the previous two variables. As we did before we are going to find the ratio of danish citizens in the area, but because the variable is chosen to be less important also due to the next variable, we are only doing a 'summary()' to get a distribution of the ratio across the 80 areas:

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Danish Citizen Ratio	0.78	0.907	0.926	0.917	0.939	0.952

Looking at the summary table it seems that there is one or two outliers where the the Danish Citizen ratio is around 78%, while in the other areas it seems to be more stable around $92 - 95\%$.

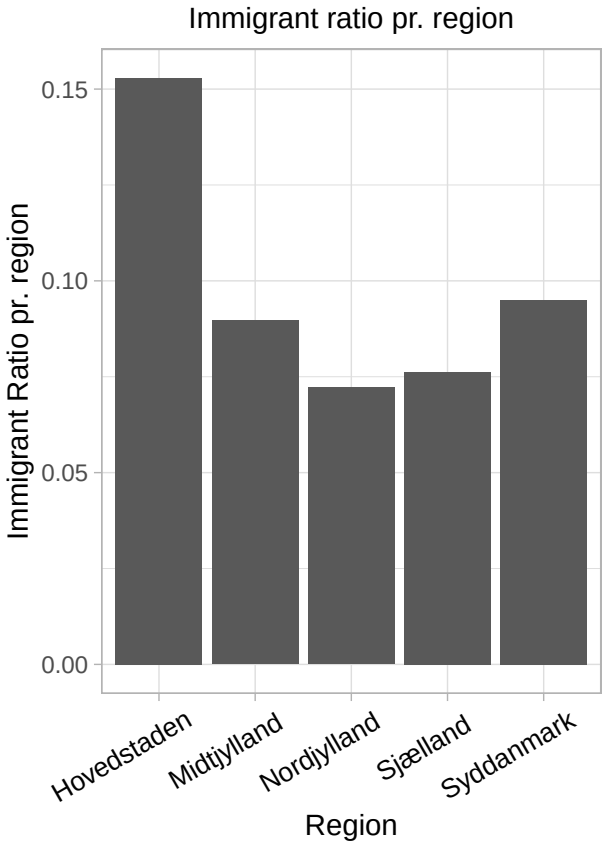
Indvandrere

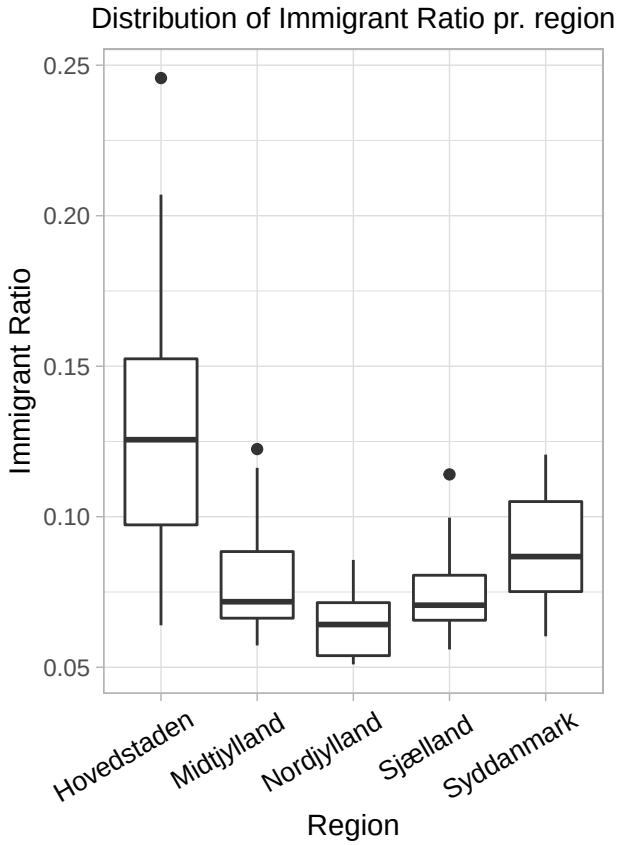
The explanatory variable 'Indvandrere' is collected in the '2_Get_population_FOLK1C_v01.r' R-script using the same method as the previous variables. It describes the number of immigrants living in the area in

the 3rd quarter of 2019. As before we first calculate the immigrant ratio, then we show the 5 areas having the highest immigrant ratio and the 5 areas having the lowest immigrant ratio. We will also display the immigrant ratio pr. region and the dsitribution of the immigrant ratio pr. region by a bar- and boxplot:

Nr. Aldi Stores	Area	Region	Population	Immigrant Ratio
2	Ishøj	Hovedstaden	22988	0.246
6	Brøndby	Hovedstaden	35104	0.207
3	Høje-Taastrup	Hovedstaden	50853	0.198
18	København	Hovedstaden	626508	0.196
4	Albertslund	Hovedstaden	27750	0.180
2	Gladsaxe	Hovedstaden	69489	0.160

Nr. Aldi Stores	Area	Region	Population	Immigrant Ratio
6	Kalundborg	Sjælland	48564	0.058
3	Skanderborg	Midtjylland	62286	0.057
5	Sorø	Sjælland	29860	0.056
6	Jammerbugt	Nordjylland	38440	0.054
1	Morsø	Nordjylland	20366	0.053
5	Rebild	Nordjylland	30073	0.051





	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Immigrant Ratio	0.051	0.067	0.082	0.094	0.111	0.246

From the above plots and tables we see, that the highest Immigration Ratio on around 25% is in Ishøj whereas the lowest immigration ratio is achived in Rebild, where the immigration ratio is around 5%. The overall mean in the 80 areas is around 10%, where region Hovedstaden on average have the highest immigration ratio. The boxplot also reveals that Region Hovedstaden has the highest variation of immigration ratio. While the distribution in the other regions are less spread out.

Dansk_oprindelse

The explanatory variable 'Dansk_oprindelse' is collected in the same R-script as the variable 'Indvadrere'. It describes the number of people with danish origin, living in the area in the 3rd quarter of 2019. We will again calculate the ratio of people with danish origin and use the 'summary()' function to see the distribution of the ratio across the 80 areas.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Danish Origin	0.593	0.86	0.895	0.876	0.917	0.941

Looking at the above summary table, it seems that (which was also the case in Danish Citizen Ratio) there is one or two outliers with low Danish Origin Ratio, while in the other cases it seems to be rather stable around give or take 90%.

Alder

The explanatory variable 'Alder' is collected in the '2_Get_population_FOLK1A_alder_v01.r' and describes the average age in the area in the 3rd quarter of 2019. The distribution of the average age accross the 80 araes is again displayed using the 'summary()' function.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Avg. Age	35.533	40.983	42.108	42.31	43.795	47.723

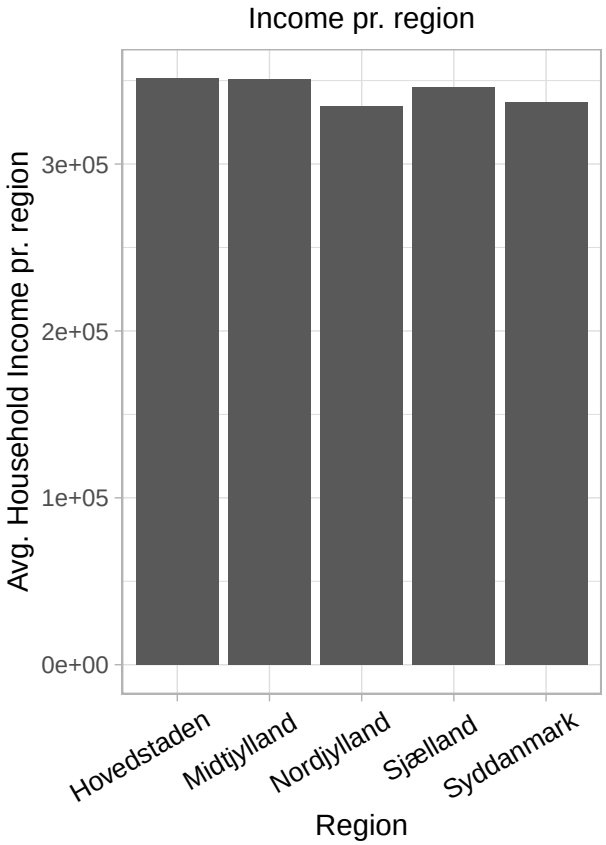
As we see in the above summary table the Avg. age across all 80 areas takes the minimum value of 35.5 years and the maximum value of around 47.7 years. The overall mean of the avg. Age is around 42 years.

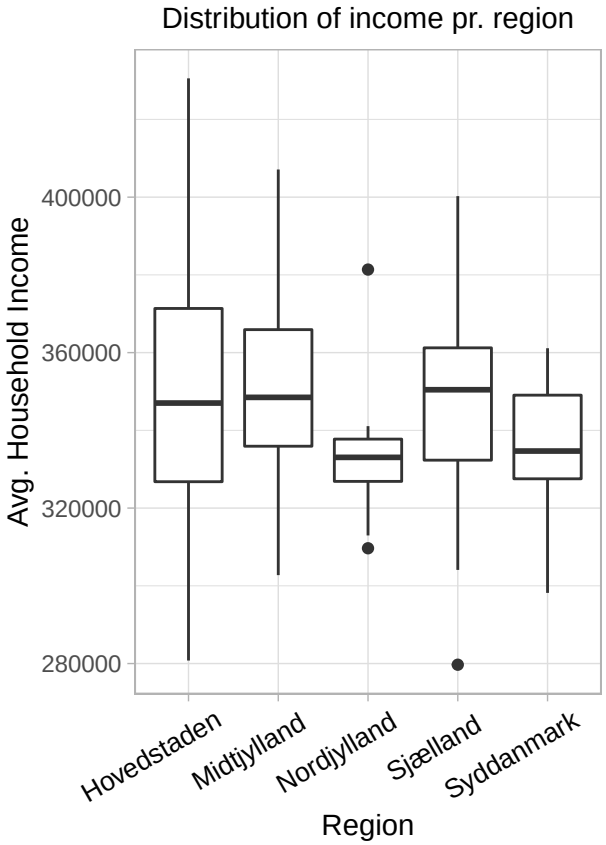
Indkomst

The explanatory variable 'Indkomst' is collected in the '2_Get_indkomst_v01.r' script, and it describes the average household income in 2018 in the Area. We are going to display the top 5 areas with the highest average household income as well as the 5 areas with the lowest average household income. We will also make a bar plot containing the average household income in each region, and also a boxplot displaying the distribution of the average household income in each region. We will also show the distribution across all 80 ares using the 'summary()' function:

Nr. Aldi Stores	Area	Region	Population	Avg. Household Income
3	Egedal	Hovedstaden	43383	430561.1
4	Allerød	Hovedstaden	25729	429340.7
3	Skanderborg	Midtjylland	62286	407111.3
2	Lejre	Sjælland	27995	400262.3
2	Favrskov	Midtjylland	48373	395026.5
3	Greve	Sjælland	50289	392927.0

Nr. Aldi Stores	Area	Region	Population	Avg. Household Income
8	Aalborg	Nordjylland	215510	309665.7
7	Guldborgsund	Sjælland	60872	304094.9
14	Aarhus	Midtjylland	345635	302749.3
9	Odense	Syddanmark	204120	298181.6
18	København	Hovedstaden	626508	280779.9
3	Lolland	Sjælland	41475	279704.1





	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Avg. Household Income	279704.1	327542.7	344436.7	345683.2	358441.4	430561.1

It is shown from the above tables and plots that the minimum average household income on around 280.000 is in Lolland, where the area with the highest average household income is Egedal with around 430.000. On

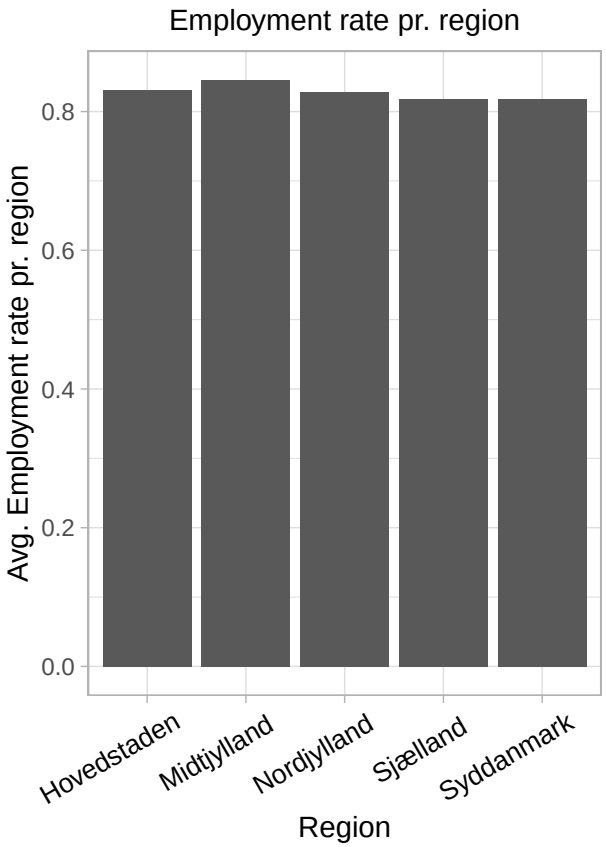
average by region the household income seems to be evenly distributed, but within each region there is some kind of variation, especially in Region Hovedstaden.

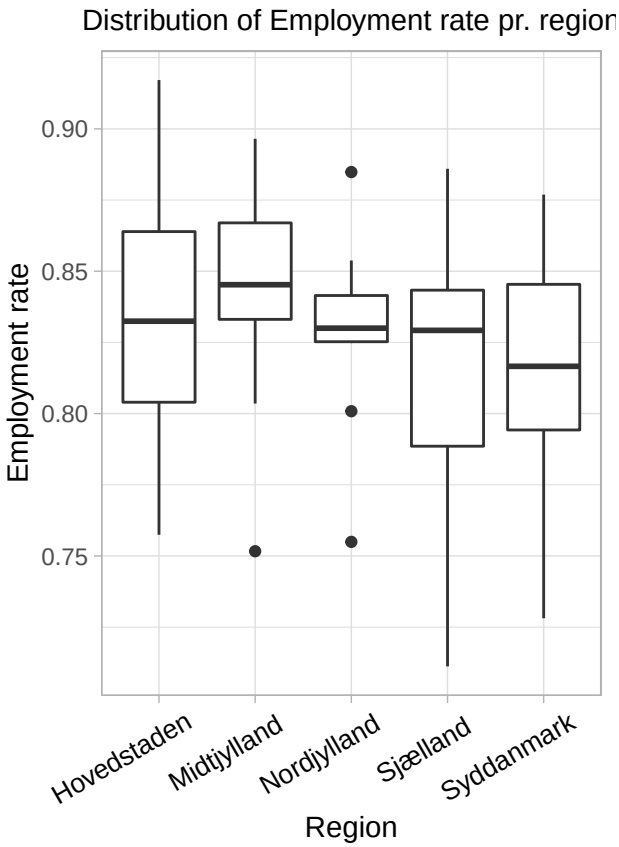
Besk

The explanatory variable 'Besk' describes the employment rate in an area in 2018. The variable is collected in the '2_Get_beskæft_v01.r' script. We are now going to display the 5 ares with the highest employment rate, and the 5 ares with the lowest employment rate. Furthermore we will make a bar plot describing the average employment rate for each of the 5 regions, and a boxplot showing the distribution of the employment rate within each region. We will also display the overall distribution of the employment rate for all areas using the 'summary()' function:

Nr. Aldi Stores	Area	Region	Population	Employment rate
4	Allerød	Hovedstaden	25729	0.917
3	Egedal	Hovedstaden	43383	0.903
2	Favrskov	Midtjylland	48373	0.897
6	Hedensted	Midtjylland	46726	0.893
3	Skanderborg	Midtjylland	62286	0.893
2	Lejre	Sjælland	27995	0.886

Nr. Aldi Stores	Area	Region	Population	Employment rate
18	København	Hovedstaden	626508	0.758
6	Brøndby	Hovedstaden	35104	0.757
8	Aalborg	Nordjylland	215510	0.755
14	Aarhus	Midtjylland	345635	0.752
9	Odense	Syddanmark	204120	0.728
3	Lolland	Sjælland	41475	0.711





	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Employment Rate	0.711	0.804	0.833	0.829	0.86	0.917

Looking at the above tables and plots it is clear, that on average by region the Employment Rate seems to be evenly distributed on around 80%. However the boxplot reveals some variation within the regions, especially Region Sjælland, Syddanmark and Hovedstaden, wheras Region Midtjylland and Nordjylland seems more similar, but with some outliers. The employment rate overall goes from around 70 in Lolland to around 90% in Allerød, where the overall mean is around 82%.

ArbStrk

The variable 'ArbStrk' describes the labor force in each area in the 3rd quarter of 2019. It is defined as people from age 18-65. It is collected in the '2_Get_population_FOLK1A_under18_v01.r' script. We will first calculate the labor force ratio and then use the 'summary()' function the display the distribution across the 80 areas.

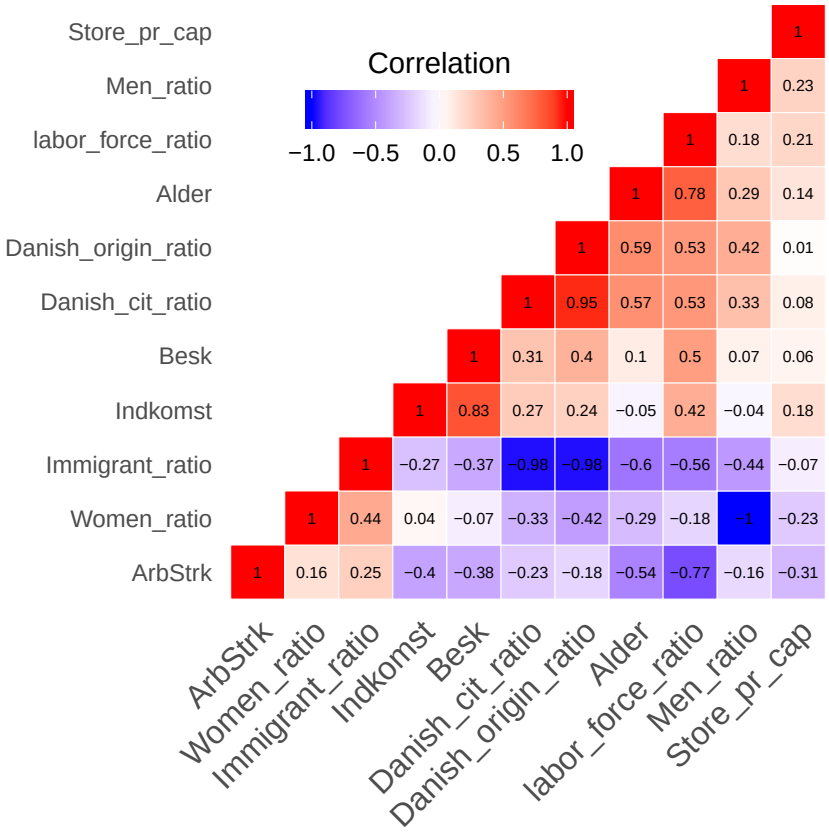
	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Labor Force	0.272	0.396	0.411	0.406	0.423	0.445

The above summary table reveals that the overall mean of the Labor Force is around 40% ranging from around 27% as minimum value and 45% as maximum value.

Describe the correlation between the number of Aldi stores pr. capita and the other explanatory variables

Correlation matrix

In the following section we will create a table containg the correlation coefficient from the Aldi store pr. 10.000 capita to all the other explanatory variables. The correlation shows the strength and the direction of the linear relationship between the variables, compared to for instance the covariance which only gives the direction. The correlation coefficient ranges from -1 to 1, where the absolute value corresponds to the strength of the linear relationship i.e. the closer the coefficient is to -1 or 1 the stronger the linear relationship is. If the correlation coefficient is negative, it means that the two variables are negative correlated i.e if number of Aldi stores increases the other variable decreases. If the correlation coefficient is positive it shows a positive linear relationship i.e. when number of aldi stores increases the other variable increases as well. First we are going to show a graphical illustration of the correlation coefficient between all the variables. The below chart is a slightly changed version of the one found here: http://www.sthda.com/english/wiki/ggplot2-quick-correlation-matrix-heatmap-r-software-and-data-visualization, changed to fit our dataset.



The above chart shows how each varibale (icluding the number of Aldi stores pr. 10.000 capita) is correlated to each other. In this specific assignment we are mostly interested in how the number of Aldi stores pr. 10.000 capita is correlated with the other explanatory variables. However the mutual correlation between the explanatory variables may play a role when finding the best linear model to predict the data, but because the main scope of this assignment dosen't include deep knowledge of the linear regression, we will not dive into it here. For now we will collect the correlation coefficients of interest (between the number of Aldi stores to the other explanatory variables) and put them in a table/data frame.

```r
Disc_conc$Store_pr_cap <- (Disc_conc$ALDI/Disc_conc$Total)*10000
Aldi_corr <- Disc_conc %>%
  select(Alder:Store_pr_cap)

as.data.frame(round(cor(Aldi_corr),3)[11,])%>%
  kable("latex", booktabs = T, col.names = c("Correlation from Store pr. cap"))%>%
  kable_styling()
```

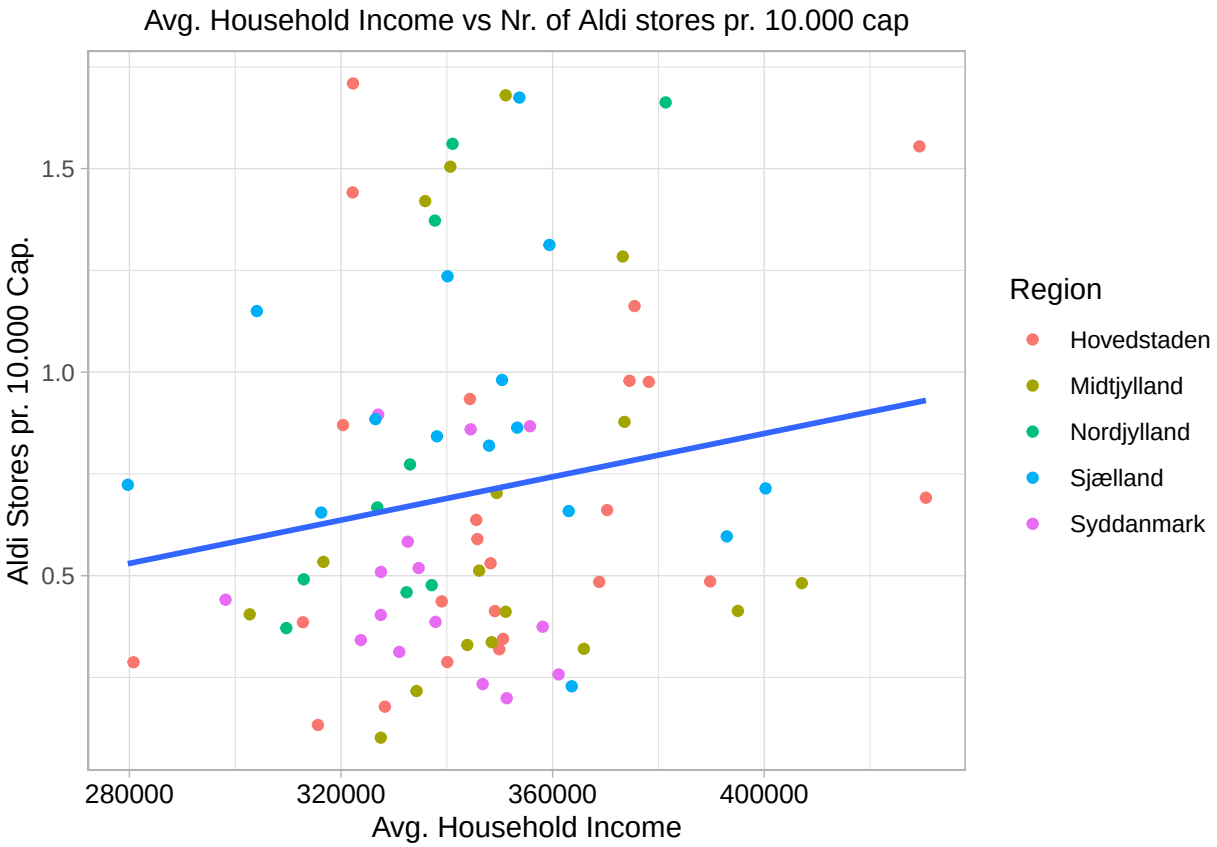
	Correlation from Store pr. cap
Alder	0.137
Indkomst	0.176
Besk	0.063
ArbStrk	-0.309
Women_ratio	-0.226
Men_ratio	0.226
Danish_cit_ratio	0.084
Immigrant_ratio	-0.066
Danish_origin_ratio	0.006
labor_force_ratio	0.206
Store_pr_cap	1.000

Due to the fact that we will take a closer look on 4 chosen explanatory variables later in this assignment, the following section will create scatter plots for these specific variables displaying any correlation more graphically.

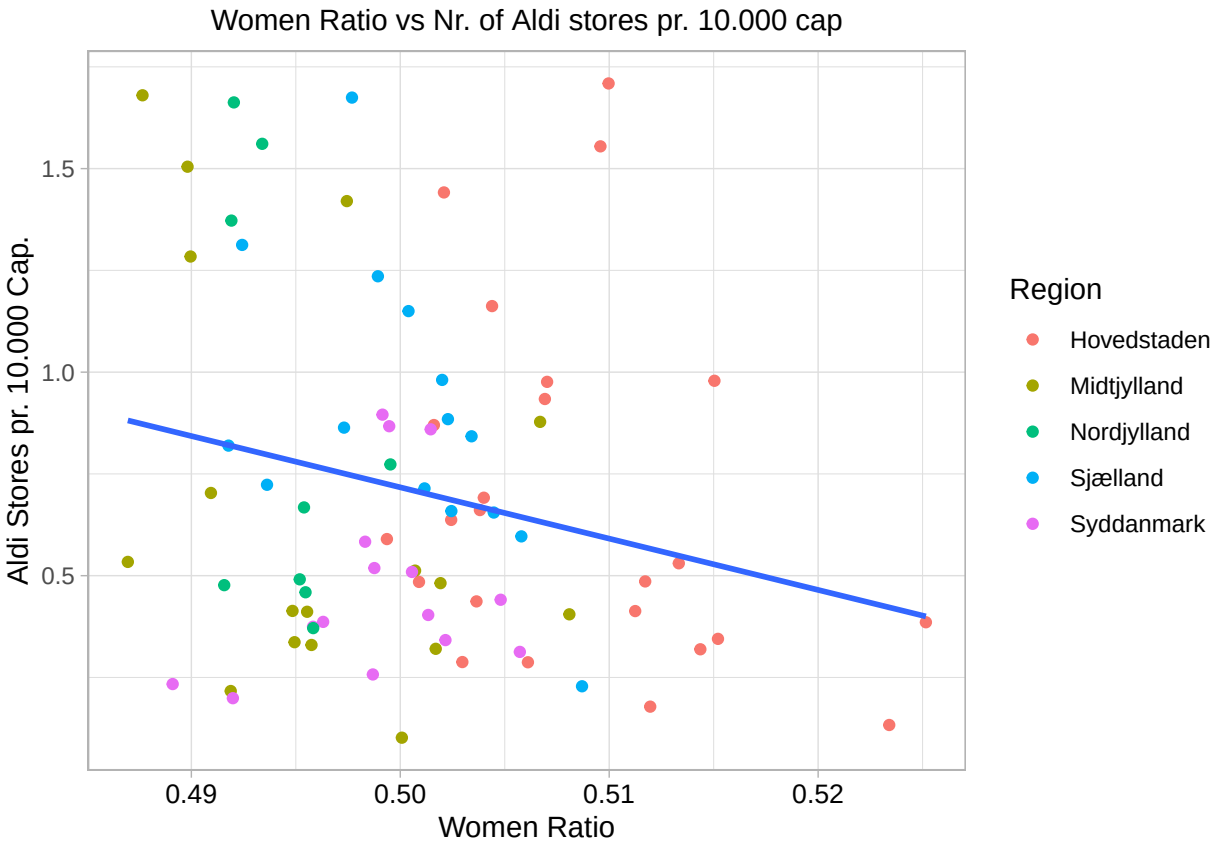
Scatter Plots

In this section we are going to make 4 scatter plots showing the correlation between the 4 explanatory variables Women Ratio, Immigrant Ratio, Income and Employment Rate and the number of Aldi stores. The code generating the plots are very similar, just changing the x-variable, so in that sense we'll only display the R code generating the first scatter plot. Each scatter plot contains a linear regrssion line to better visualise the correlation.

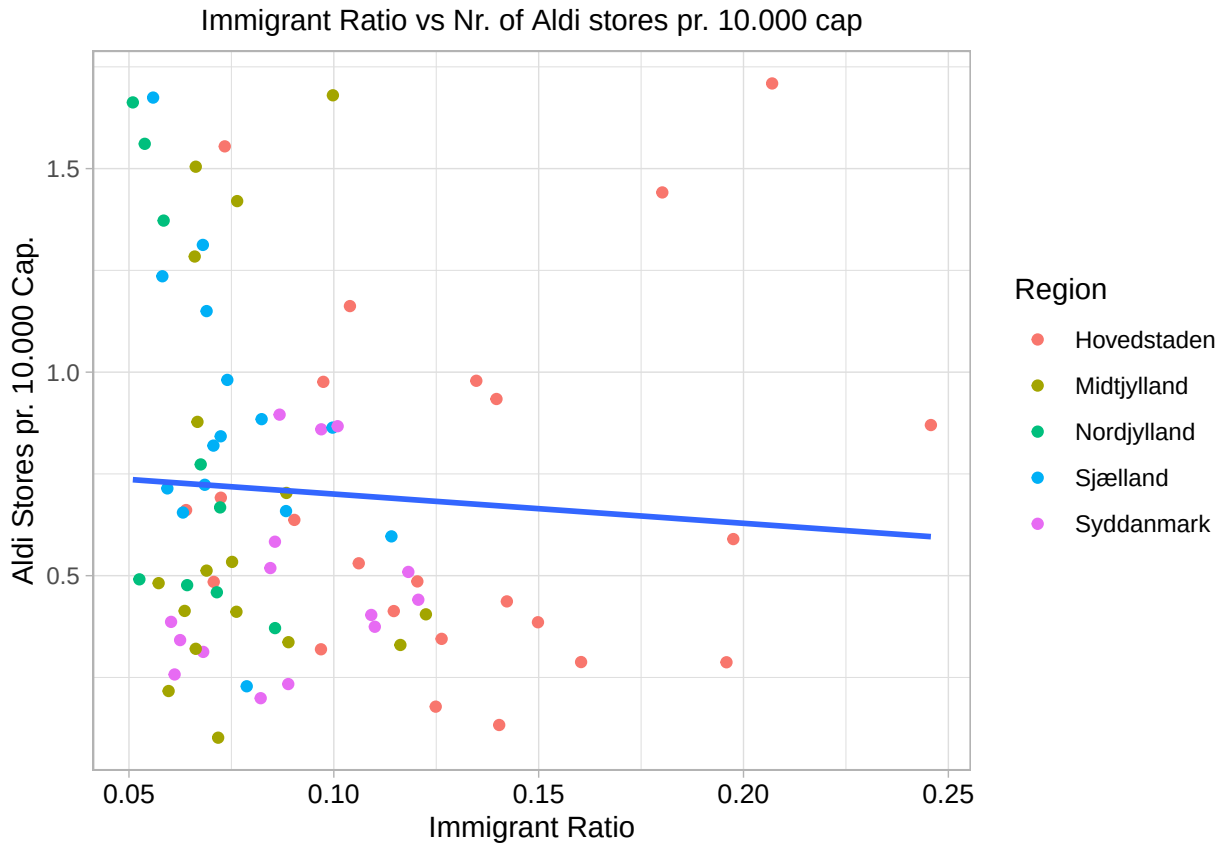
```r
Disc_conc %>%
  ggplot(mapping=aes(x = Indkomst, y = Store_pr_cap))+
  geom_point(mapping = aes(color=Region))+
  geom_smooth(method=lm, se=F)+
  labs(title = "Avg. Household Income vs Nr. of Aldi stores pr. 10.000 cap",
       y= "Aldi Stores pr. 10.000 Cap.",
       x= "Avg. Household Income")+
  theme_light()+
  theme(axis.text.x=element_text(color = "black", size=10),
        plot.title = element_text(size=11, hjust = 0.5))+
  NULL
```



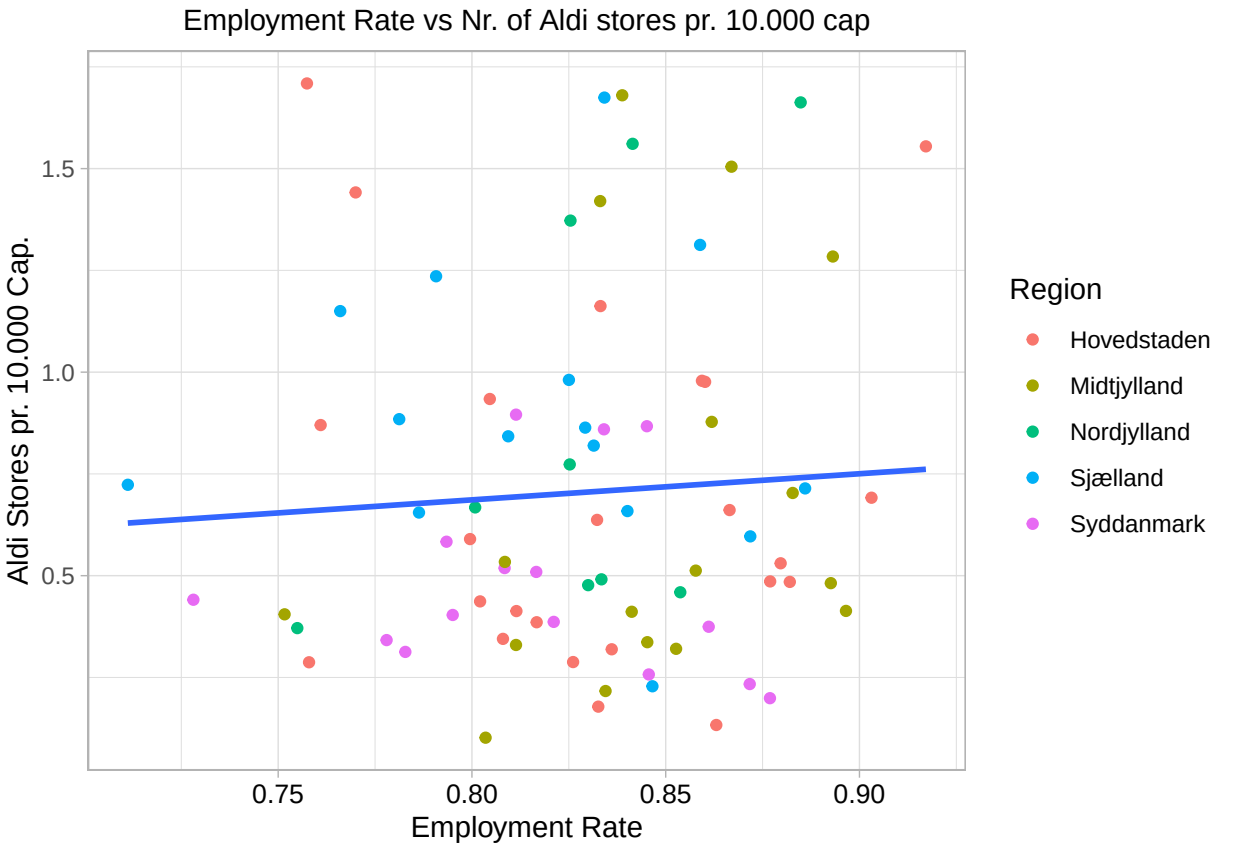
As we can see from the above scatter plot, there is a slightly positive correlation between the average household income and the number of Aldi stores, as we also saw i the previous correlation matrix. However our data seems to be quite spread out, and the linear regression line dosent fit/explain our data very well. We will return to that later on, when we compute the R-squared value for the different linear regression models.



Looking at the above scatter-plot it is clear that there is a negative correlation between the women ratio and the Aldi stores, that is when the number of Aldi stores increase the women ratio decreases and virsa versa. Again, the linear regression line dosen't seem to fit/explain our data very well.



By looking at the above plot, we see a weak negative correlation between the immigration ratio and the number of Aldi stores. As in the prevous cases the linear regression model doesent seem to fit/explain our data very well.



The last scatter-plot reveals a small positive correlation between the employment rate and the number of Aldi stores. Like in all of the previous scatter-plots the data is quite spread out, and the linear regression line therefore dosent fit our data to well.

Models for prediction (Nr of Aldi stores pr. 10.000 capita)

In this section we are going to build a linear model to predict the number of Aldi Stores pr. 10.000 capita using the following 4 explanatory variables:

- Women Ratio
- Immigrant Ratio
- Avg. household income
- Employment Rate

Computing the best model based on Adjusted R Squared

We are going to make 15 different linear models trying out different combinations of the 4 explanatory variables. We will be using the Adjusted R-Squared to determine wich of the models best describes our data. The R-squarred value usually goes from 0 to 1 telling how well the linear model describes/explains our data. If the R-Squarred value is one, then the linear model is able to explain 100% of our data. When adding more explanatory variables to our model, the R-Squarred value will increase, but does that mean that our model gets better? Not necessarily, when adding more explanatory variables to our model, the model becomes less "flexible" due to loss of degrees of freedom. The Adjusted R-Squarred value takes the loss of degrees of freedom into account, which is why all else equal that the Adjusted R-squarred value in this case would be better to determine the best model.

Below we have made the first model, which is simply just what we saw in the first scatterplot. The coefficints for the linear model and the R-squarred as well as the Adjusted R-Squarred values are saved in a list:

```
y <- Disc_conc$Store_pr_cap
x1 <- Disc_conc$Women_ratio
x2 <- Disc_conc$Immigrant_ratio
x3 <- Disc_conc$Indkomst
x4 <- Disc_conc$Besk

res <- summary(mod <- lm(y ~ x1))
m1  <- list(coef=mod$coefficients,R2=res$r.squared,adj.R2=res$adj.r.squared )
```

The rest of the models are made in the same way, in the end we can combine the results for each model in a data.frame:

Computing linear models for full Dataset

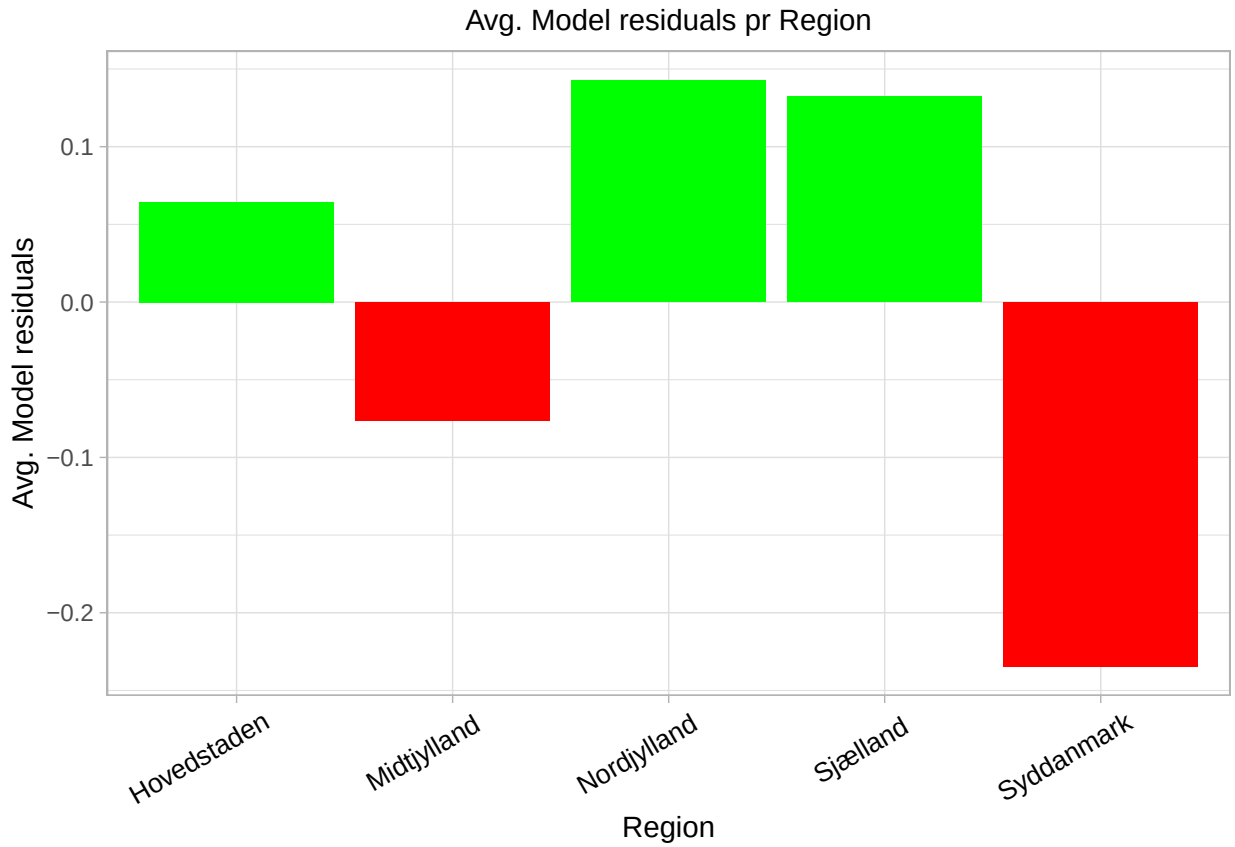
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	**M13**	M14	M15
Intercept	7.018	0.772	-0.214	0.173	7.513	7.513	6.521	-0.167	0.377	0.852	7.492	7.111	**8.552**	1.132	9.173
Women Ratio	-12.602	NA	NA	NA	-13.678	-13.678	-12.424	NA	NA	NA	-16.091	-14.186	**-14.811**	NA	-16.654
Immigrant Ratio	NA	-0.718	NA	NA	0.473	NA	NA	-0.211	-0.537	NA	1.336	0.817	**NA**	-0.707	0.851
Income	NA	NA	0.000	NA	NA	0.473	NA	0.000	NA	0.000	0.000	NA	**0.000**	0.000	0.000
Employment Rate	NA	NA	NA	0.641	NA	NA	0.492	NA	0.457	-2.635	NA	0.752	**-3.454**	-2.937	-3.193
R-squared	0.051	0.004	0.031	0.004	0.053	0.085	0.053	0.031	0.006	0.052	0.096	0.057	**0.121**	0.056	0.125
adj.R-squared	0.039	-0.008	0.019	-0.009	0.028	0.062	0.029	0.006	-0.020	0.028	0.060	0.020	**0.086**	0.019	0.078

Based on the above table, we choose Model 13 as our best model, using the 3 explanatory variables Women Ratio, Income and Employment Rate. Eventhough model 15 has a higher R-Squarred value (using all 4 explanatory variables) we see that the variable Immigration Ratio dosent add significant value to our model based on loss of degrees of freedom expressed in the Adj. R-Squarred value.

Calculate the mean model-error for the Regions

In this section we are going to calculate the mean model-error for each region, displaying how well our model explains/predicts the number of Aldi stores in each region.

```
# Best model (Model 13)
Best_model <- lm(y ~ x1+x3+x4)

# Create columns computing predictions and residuals
Disc_conc$Model_prediction <- predict(Best_model)
Disc_conc$Model_residual <- residuals(Best_model)
Disc_conc$Model_residual1 <- (Disc_conc$Store_pr_cap - Disc_conc$Model_prediction)

#Create a plot displaying mean model residuals grouped by region
Disc_conc %>%
  group_by(Region)%>%
  summarise(Avg_model_residual = mean(Model_residual))%>%
  mutate(Negative = Avg_model_residual < 0)%>%
  ggplot(mapping = aes(x= Region, y=Avg_model_residual, fill=Negative))+
  geom_col(stat="identity")+
  labs(y = "Avg. Model residuals",
       title = "Avg. Model residuals pr Region")+
  scale_fill_manual(values = c("green","red"))+
  theme_light()+
  theme(axis.text.x=element_text(color = "black", size=10, angle=30, vjust=.8, hjust=0.8),
        plot.title = element_text(size=11, hjust = 0.5),
        legend.position = "none")
```



We made the above plot by first creating 3 new columns in our dataset, where the column Model_prediction is the predicted number of Aldi stores pr 10.000 capita based on our best model. The second two columns show the same thing using to different methods. The first column 'Model_residual' is computed using the residuals() function, while the second column 'Model_residual1' is calculated based on our observed number of aldi stores pr. 10.000 capita -(minus) the predicted value. The two columns is identical. The plot then group or dataset by region and calculate the mean model error for each region. If the mean model error is negative it has the color red, while positive model errors are colored green.

The above plot reveals that Region Syddanmark obtains the highest mean residual on around -0.25. That is, on average, our model predict 0.25 Aldi Stores pr. 10.000 capita more than what we observed in Region Syddanmark. Our model also overestimate the number of Aldi Stores in Region Midtjylland, while in the other regions our model underestimate the observed number of Aldi stores pr. 10.000 capita. Our model seems to best predict the number of Aldi stores in Region Hovedstanden where the mean residual is around 0.06. The exact Avg. Model residual for each region is displayed in the below table.

Region	Avg_model_residual
Hovedstaden	0.064
Midtjylland	-0.076
Nordjylland	0.143
Sjælland	0.132
Syddanmark	-0.235

Repeating the analasis without Region Syddanmark

The below analysis finding the best model for the dataset without Region syddanmark follows the exactly same procedure as before. In this case we first split up our full dataset in a dataset called 'Training_data' where we have filtered Region Syddanmark out, and 'Test_data' wich is a dataset only containing values from Region Syddanmark.

```
Training_data <- Disc_conc %>%
  filter(Region != "Syddanmark")

Test_data <- Disc_conc %>%
  filter(Region == "Syddanmark")

y_1 <- Training_data$Store_pr_cap
x1_1 <- Training_data$Women_ratio
x2_1 <- Training_data$Immigrant_ratio
x3_1 <- Training_data$Indkomst
x4_1 <- Training_data$Besk
```

We then combine all the different models in a dataframe:

Computing linear models for Dataset without Region Syddanmark

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	**M13**	M14	M15
Intercept	9.101	0.863	-0.059	0.348	9.609	9.609	8.699	0.104	0.831	1.137	9.821	9.380	**10.366**	1.803	10.854
Women Ratio	-16.640	NA	NA	NA	-17.741	-17.741	-16.627	NA	NA	NA	-20.562	-18.779	**-18.106**	NA	-19.703
Immigrant Ratio	NA	-1.114	NA	NA	0.463	NA	NA	-0.694	-1.099	NA	1.366	0.909	**NA**	-1.399	0.684
Income	NA	NA	0.000	NA	NA	0.463	NA	0.000	NA	0.000	0.000	NA	**0.000**	0.000	0.000
Employment Rate	NA	NA	NA	0.492	NA	NA	0.475	NA	0.036	-2.800	NA	0.850	**-3.377**	-3.593	-3.041
R-squared	0.093	0.011	0.025	0.002	0.095	0.125	0.096	0.029	0.011	0.048	0.136	0.100	**0.157**	0.062	0.160
adj.R-squared	0.079	-0.004	0.010	-0.014	0.066	0.096	0.066	-0.002	-0.021	0.017	0.094	0.056	**0.116**	0.016	0.104

Based on the above table we see that Model 13 is still the best model, because it has the highest value for the adjusted R-Squarred. We now use the new model 13 to predict the number of Aldi stores pr. 10.000 capita in our test_dataset i.e. for Region syddanmark. The predicted value is here calculated by multiplying the coefficients with the observed data for the relevant explanatory variable and then add the intercept coefficient.

```
res <- summary(mod <- lm(y_1 ~ x1_1+x3_1+x4_1))
New_m13  <- list(coef=mod$coefficients,R2=res$r.squared,adj.R2=res$adj.r.squared )

Test_data$Test_Prediction <- New_m13$coef[2]*Test_data$Women_ratio +
  New_m13$coef[3]*Test_data$Indkomst + New_m13$coef[4]*Test_data$Besk + New_m13$coef[1]

Test_data$Test_Residual <- (Test_data$Store_pr_cap - Test_data$Test_Prediction)

Test_data %>%
  select(c("Område", "Store_pr_cap", "Test_Prediction", "Test_Residual"))%>%
  kable(digits = 3, "latex", booktabs = T)%>%
  kable_styling(latex_options = c("striped"))
```

Område	Store_pr_cap	Test_Prediction	Test_Residual
Billund	0.375	0.827	-0.453
Esbjerg	0.519	0.798	-0.280
Fredericia	0.583	0.844	-0.260
Faaborg-Midtfyn	0.387	0.821	-0.434
Haderslev	0.896	0.731	0.164
Kolding	0.860	0.728	0.132
Middelfart	0.257	0.848	-0.590
Nyborg	0.313	0.735	-0.422
Odense	0.441	0.721	-0.280
Svendborg	0.342	0.768	-0.426
Sønderborg	0.403	0.750	-0.347
Varde	0.199	0.799	-0.600
Vejen	0.234	0.839	-0.605
Vejle	0.867	0.800	0.068
Aabenraa	0.509	0.692	-0.182

Looking at the above table it is obvious that our model in most cases overestimate the number of Aldi stores pr. 10.000 capita. Also we can calculate the mean model error to be around -0.301. It is not surprising that our average model error in this case is higher than what we observed when we fitted a model on the full dataset (including Region Syddanmark), because there the data from Region Syddanmark had an influence on the coefficients in the model.

Calculate the R-squarred using our New Model on the test dataset

We are now going to quantify the how well the new model fit/explains our Test_data, i.e. the dataset containing only Region Syddanmark. To do that we are manually calculating R^2 given by the formula:

$$R^2 = 1 - \frac{R_{SS}}{T_{SS}},$$

where $R_{SS} = \sum_{i=1}^{n} \hat{\varepsilon}_i^2$ is the residual variation (Residual sum of Squares) and $T_{SS} = \sum_{i=1}^{n}(y_i - \overline{y})^2$ is the total variation. So the ratio $\frac{R_{SS}}{T_{SS}}$ tells us how much of the variation in the dataset our model fail to explain, where $1 - \frac{R_{SS}}{T_{SS}}$ tells us how much we are able to explain. We calculate these values and collect them in a dataframe:

```
y <- Test_data$Store_pr_cap
Rss <- sum(Test_data$Test_Residual^2)
Tss <- sum((y-mean(y))^2)
R2 <- 1- (Rss/Tss)

performance_model <- data.frame(
  Model13 = c(Rss, Tss, R2)
)

rownames(performance_model) <- c("Rss", "Tss", "R-squared")

performance_model %>%
  kable(digits = 3, "latex", booktabs = T)%>%
  kable_styling(latex_options = c("striped"))
```

	Model13
Rss	2.255
Tss	0.743
R-squared	-2.034

As we can see in the above table our R^2 is negative. That means that the total variation in the data T_{SS} is less than the variation of the residuals R_{SS}, which indicates that our model predicts/explains our date very poorly, actually going against the trend of our data. That also means, that a horizontal line given by the mean of the number of Aldi stores pr. capita better explains our data. When using the mean of the number of Aldi stores (here in Region Syddanmark) as a prediction for the number of Aldi Stores in different areas in Region Syddanmark, we achive an $R^2 = 0$, which is better than our New_model_13. This is also illustrated in the following Table:

```
# Mean as prediction

Test_data$Residual_w_mean <- (Test_data$Store_pr_cap - mean(y))

Test_data %>%
  select(c("Område","Test_Residual","Residual_w_mean"))%>%
  kable(digits = 3, "latex", booktabs = T)%>%
  kable_styling(latex_options = c("striped"))%>%
  row_spec(c(5,6,14), bold = T)
```

Område	Test_Residual	Residual_w_mean
Billund	-0.453	-0.104
Esbjerg	-0.280	0.040
Fredericia	-0.260	0.105
Faaborg-Midtfyn	-0.434	-0.092
Haderslev	**0.164**	**0.417**
Kolding	**0.132**	**0.381**
Middelfart	-0.590	-0.222
Nyborg	-0.422	-0.166
Odense	-0.280	-0.038
Svendborg	-0.426	-0.137
Sønderborg	-0.347	-0.075
Varde	-0.600	-0.280
Vejen	-0.605	-0.245
Vejle	**0.068**	**0.388**
Aabenraa	-0.182	0.030

In the above table we see that the mean as a predicter has a lower residual/error in all the areas except in Haderslev, Kolding and Vejle. So overall the mean as a predicter is actually better than using our model as a predicter.

Conclusion

To shortly conclude on what we have observed in the analysis, we can say that the Aldi stores are more or less evenly distributed across Denmark, but where the concentration of Aldi stores are higher in the bigger cities as Aarhus and Copenhagen. Based on our centrum-stores in Copenhagen and Aarhus, Copenhagen archived the highest concentration (with 11 Aldi stores) when looking at a radius of 5 km to our centrum, but relative to the total amount of Aldi stores in Copenhagen and Aarhus we saw that around 55% was placed within a radius of 5 km in Aarhus, while the percentage in Copenhagen was around 44%. We also saw that across all ares in our dataset, the average number of Aldi stores pr. 10.000 capita is around 0.7, where pr. region Region Sjælland has the highest number of aldi stores pr. 10.000 capita. In the description of our explanatory variables we saw that some variables such as the Men and women Ratio were quite evenly distributed across all areas, whereas some variables such as Income and Immigration ratio contained more variation pr region/area.

Describing the correlation between the number of Aldi stores and the other explanatory variables revealed no significant correlation with the correlation coefficient from number of Aldi stores and the Labor Force of around -0.3 as the most significant. We also noticed in the scatter-plots that the linear regression line didnt explain/fit our data very well, which is also expressed in the R-Squarred/Adj. R-Squarred values in the table computing the best linear model. We otained the highest Adj. R-Squarred model of around 0.08 which means that the model is able to xplain around 8% of the variation of our data. The more or less poor performance of our model is also expressed in the predictions giving relatively high mean model error/residual for the regions. When predicting the number of Aldi stores in Region Syddanmark using the model 'trained' on our Training_data, we also do quite poorly, actually achieving a negative R-Squarred value, meaning our model goes against the trend of our data, and that our data is better explained with a horizontal line at the average. A factor playing a role in the poor performance of our model might be the fact of a limiting amount of Aldi stores in Denmark.

Repoduce the work

To be able to reproduce my work in this assignment, and to make sure people know wich R environment I have used to do my analysis, i run the following R code as the final remark.

```
sessionInfo()
```

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 18362)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United Kingdom.1252
## [2] LC_CTYPE=English_United Kingdom.1252
## [3] LC_MONETARY=English_United Kingdom.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United Kingdom.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] reshape2_1.4.3   cowplot_1.0.0    kableExtra_1.1.0 knitr_1.25
##  [5] geosphere_1.5-10 ggmap_3.0.0      googleway_2.7.1  rstudioapi_0.10
##  [9] forcats_0.4.0    stringr_1.4.0    dplyr_0.8.3      purrr_0.3.3
## [13] readr_1.3.1      tidyr_1.0.0      tibble_2.1.3     ggplot2_3.2.1
## [17] tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.1         jsonlite_1.6       viridisLite_0.3.0
##  [4] modelr_0.1.5       shiny_1.4.0        assertthat_0.2.1
##  [7] sp_1.3-1           cellranger_1.1.0   yaml_2.2.0
## [10] pillar_1.4.2       backports_1.1.4    lattice_0.20-38
## [13] glue_1.3.1         digest_0.6.21      promises_1.1.0
## [16] rvest_0.3.5        colorspace_1.4-1   htmltools_0.4.0
## [19] httpuv_1.5.2       plyr_1.8.4         pkgconfig_2.0.3
## [22] broom_0.5.2        haven_2.2.0        xtable_1.8-4
## [25] scales_1.0.0       webshot_0.5.2      jpeg_0.1-8
## [28] later_1.0.0        generics_0.0.2     withr_2.1.2
## [31] lazyeval_0.2.2     cli_1.1.0          magrittr_1.5
## [34] crayon_1.3.4       readxl_1.3.1       mime_0.7
## [37] evaluate_0.14      fs_1.3.1           nlme_3.1-140
## [40] xml2_1.2.2         tools_3.6.1        hms_0.5.2
## [43] RgoogleMaps_1.4.4 lifecycle_0.1.0    munsell_0.5.0
## [46] reprex_0.3.0       compiler_3.6.1     rlang_0.4.2
## [49] grid_3.6.1         rjson_0.2.20       htmlwidgets_1.5.1
## [52] bitops_1.0-6       labeling_0.3       rmarkdown_1.15
## [55] gtable_0.3.0       DBI_1.0.0          curl_4.2
## [58] R6_2.4.0           lubridate_1.7.4    fastmap_1.0.1
## [61] zeallot_0.1.0      stringi_1.4.3      Rcpp_1.0.2
## [64] vctrs_0.2.0        png_0.1-7          dbplyr_1.4.2
## [67] tidyselect_0.2.5   xfun_0.9
```