Political Data Science

Lektion 12

Usuperviseret læring

Opgave 1

1. Indlæs pakkerne `rio`, `dplyr`, `ggplot2` og `purrr`.
2. Indlæs de to datasæt, `holdninger.rdata` og `partivalg.rdata` fra GitHub. De to datasæt indeholder henholdsvis holdninger på 7 dimensioner samt partivalg for omtrent 2000 respondenter. Data stammer fra valgundersøgelsen 2015.

Indlæs eventuelt data med koden:

```
holdninger <- import("https://github.com/jespersvejgaard/PDS/raw/master/data/holdninger.rdata")
partivalg <- import("https://github.com/jespersvejgaard/PDS/raw/master/data/partivalg.rdata")
```

Opgave 2: PCA

1. Undersøg gennemsnittet og standardafvigelsen for variablene i `holdninger`. Brug eventuelt funktionen `map` fra pakken `purrr`. Bør vi centrere og skalere data, før vi laver PCA?
2. Beregn PCA med funktionen `prcomp()` og gem outputtet i et nyt objekt. Hvad består outputtet af?
3. Lav et biplot med de to første principal components. Brug funktionen `biplot()`, hvor det første argument skal være objektet fra ovenfor, og det andet argument skal være `scale = 0`. Hvordan fortolker du resultatet?
4. Undersøg proportion of variance explained ved at kalde `summary()` på objektet fra pkt. 2. Kommenter på resultatet.
5. Lav en ny dataframe, hvor du kobler datasættet `partivalg` med elementet `x` i dit PCA-objekt fra pkt. 2 ovenfor. Det kan fx se ud som herunder:

```
ny_df <- data.frame(holdninger_pca$x,
                    partivalg)
```

6. Lav et scatterplot med `ggplot`, hvor de to akser henholdsvist er PC1 og PC2, de først to principal components (`geom_point()`). Farvelæg punkterne efter partivalg, og brug `facet_wrap()` til at lave et plot for hvert partivalg. Hvordan tolker du resultatet?

Opgave 3: K-Means clustering

1. Start med at normalisere datasættet `holdninger` med funktionen `scale()` og gem det som et nyt data frame-objekt, fx som herunder.

```
holdnigner_norm <- scale(holdninger) %>% as.data.frame()
```

2. Brug `map` til at tjekke gennemsnittet og standardafvigelsen ud for variablene i det normaliserede datasæt. Hvad er de omtrentligt?
3. Cluster det normaliserede datasæt og lav et nyt clusters-objekt, der indeholder de resulterende clusters. Brug funktionen `kmeans()` og sæt værdier for argumenterne `center` og `nstart`.

4. Lav en ny dataframe, hvor du nu kobler datasættet `partivalg` med elementet `x` i dit PCA-objekt fra før, dvs. fra opgave 2 pkt. 2, samt med cluster-objektet fra pkt. 3 ovenfor. Det kan fx se ud som følger:

```
ny_df <- data.frame(holdninger_pca$x,
                    partivalg,
                    cluster_km = cluster_objekt$cluster)  # cluster_objekt fra pkt. 3
```

5. Lav et scatterplot med `ggplot`, hvor de to akser henholdsvist er PC1 og PC2, de først to principal components (`geom_point()`). Eksperimenter med at lade `shape` og `color` for geomet `geom_point()` være dine clusters og partivalg. (Bemærk, at kun factors kan bruges til at angive shapes i ggplot. Du kan dog ændre charactors til factors ved at putte as.factor() rundt om din character vector).

Bonus-opgave: Hierarkisk clustering

1. Brug funktionen `hclust()` til at lave hierarkisk clustering (HC). Det første argument i `hclust()` skal være dit normaliserede datasæt fra opgave 3 pkt. 1, nu skal det dog puttes ind i funktionen `dist()` for at beregne de parvise afstande. Lave forskellige HC-objekter, hvor du afprøver forskellige typer linkage ved at sætte `method = "complete"` mv. Det kan se ud som herunder:

```
objekt_hc <- hclust(dist(df_norm), method = "average")
```

2. Plot dendrogrammerne med `plot()`. Kommenter på forskellene.
3. Beskær dendrogrammerne med funktionen `cutree()`, og gem de resulterende clusters i en ny vektor. Kobl vektoren sammen med den tidligere vektor med K-Means clusters. I hvor stort omfang overlapper de to?