Lab: Canonical Correlation Analysis

2025-03-27

M1 MIDS/MFA/LOGOS
Université Paris Cité
Année 2024
Course Homepage
Moodle

```
Loading required package: testthat
Loading required package: corrr
Loading required package: magrittr
Attaching package: 'magrittr'
The following objects are masked from 'package:testthat':

    equals, is_less_than, not
Loading required package: lobstr
Loading required package: sloop
Loading required package: ggforce
Loading required package: ggplot2
Loading required package: gt
Attaching package: 'gt'
The following object is masked from 'package:testthat':

    matches
Loading required package: glue
Loading required package: skimr
Attaching package: 'skimr'
The following object is masked from 'package:corrr':

    focus
The following object is masked from 'package:testthat':

    matches
Loading required package: patchwork
```

```
Loading required package: tidyverse
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v lubridate 1.9.3     v tibble    3.2.1
v purrr     1.0.4     v tidyr     1.3.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x readr::edition_get()     masks testthat::edition_get()
x magrittr::equals()       masks testthat::equals()
x tidyr::extract()         masks magrittr::extract()
x dplyr::filter()          masks stats::filter()
x magrittr::is_less_than() masks testthat::is_less_than()
x purrr::is_null()         masks testthat::is_null()
x dplyr::lag()             masks stats::lag()
x readr::local_edition()   masks testthat::local_edition()
x dplyr::matches()         masks tidyr::matches(), skimr::matches(), gt::matches(), testth
x magrittr::not()          masks testthat::not()
x purrr::set_names()       masks magrittr::set_names()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
Loading required package: ggfortify
```

```
Loading required package: viridisLite
```

Canonical Correlation Analysis

$$C(X,Y) = \mathbb{E}\left[XY^\top\right]$$

$$\begin{bmatrix} C_{xx} & C_{xy} \\ C_{xy}^\top & C_{yy} \end{bmatrix}$$

The first canonical components are the solution of the next problem

Optimization problem

$$\begin{array}{ll} \text{Maximize} & u^\top C_{xy} v \\ \text{subject to} & u^\top C_{xx} uv = 1 = v^\top C_{yy} v \end{array}$$

Proposition

Let

$$U \times D \times V^\top$$

be a SVD of

$$C_{xx}^{-1/2} \times C_{xy} \times C_{yy}^{-1/2}$$

The solution to the optimization problem above is

$$a = C_{xx}^{-1/2} u_1 \qquad \text{and} \qquad b = S_{yy}^{-1/2} v_1$$

where u_1 and v_1 are the leading left and right singular vectors of $C_{xx}^{-1/2} \times C_{xy} \times C_{yy}^{-1/2}$, that is the first column vectors of U and V.

Proof:

Proposition

A sequence of canonical components of C_{xy} can be obtained from the sequence of (extended) left and right singular vectors of C_{xy} with respect to C_{xx} and C_{yy}

Proof:

Proposition

Let H_X (resp. H_Y) be orthorgonal projection matrix on the linear space spanned by the columns of X (resp. Y).
Canonical correlations $\rho_1 \geq ... \geq \rho_s, ...$ are the positive square roots of the eigenvalues $\lambda_1, ... \geq \lambda_s, ...$ of $H_X \times H_Y$ (which are the same as $H_Y \times H_X$): $\rho_s = \lambda_s$ Vectors $U^1, ..., U^{p_1}$ are the standardized eigenvectors corresponding to the decreasing eigenvalues $\lambda_1 \geq ... \geq \lambda_{p_1}$ of $H_X \times H_Y$
Vectors $V^1, ..., V^{p_2}$ are the standardized eigenvectors corresponding to the decreasing eigenvalues $\lambda_1 \geq ... \geq \lambda_{p_2}$ of $H_X \times H_Y$

Canonical Correlation Analysis (CCA) in `R`

`cancor()` from base package `R`

Function `cancor(x, y, xcenter=T, ycenter=T)` computes the *canonical correlations* between two data matrices `x` and `y`. Henceforth we assume that the columns of `x` and `y` are centered. Matrices `x` and `y` have the same number n of rows. `x` (resp. `y`) has `p1` (resp. `p2`) columns.

> The canonical correlation analysis seeks linear combinations of the `y` variables which are *well explained* by linear combinations of the `x` variables. The relationship is symmetric as *well explained* is measured by correlations.

The result is a list of five components

- `cor` correlations.
- `xcoef` estimated coefficients for the `x` variables.
- `ycoef` estimated coefficients for the `y` variables.

Our assumption above allows us to assume `xcenter` and `ycenter` are zeros.

The next example is taken from the documentation. Use `?LiveCycleSavings` to get more information on the dataset.

```
LifeCycleSavings |>
  as_tibble() |>
  slice_sample(n=5)
```

```
# A tibble: 5 x 5
     sr pop15 pop75   dpi  ddpi
  <dbl> <dbl> <dbl> <dbl> <dbl>
1 18.6   45.2  0.56  138.  5.14
2  4.71  47.2  0.66  243.  5.08
3 11.2   27.8  2.37 1681.  4.32
4 10.8   47.6  1.14  471.  2.8
5  1.27  34.0  3.08 1900.  1.12
```

```
fm1 <- lm(sr ~ pop15 + pop75 + dpi + ddpi, data = LifeCycleSavings)

summary(fm1)
```

```
Call:
lm(formula = sr ~ pop15 + pop75 + dpi + ddpi, data = LifeCycleSavings)

Residuals:
    Min      1Q  Median      3Q     Max
-8.2422 -2.6857 -0.2488  2.4280  9.7509

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 28.5660865  7.3545161   3.884 0.000334 ***
pop15       -0.4611931  0.1446422  -3.189 0.002603 **
pop75       -1.6914977  1.0835989  -1.561 0.125530
dpi         -0.0003369  0.0009311  -0.362 0.719173
ddpi         0.4096949  0.1961971   2.088 0.042471 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.803 on 45 degrees of freedom
Multiple R-squared:  0.3385,    Adjusted R-squared:  0.2797
F-statistic: 5.756 on 4 and 45 DF,  p-value: 0.0007904
```

```
pop <- LifeCycleSavings |>
  dplyr::select(starts_with('pop'))
oec <- LifeCycleSavings |>
  dplyr::select(-starts_with('pop'))

res.cca <- cancor(pop, oec)

res.cca$cor
```

```
[1] 0.8247966 0.3652762
```

This tells us that highest possible linear correlation beween a linear combination of `pop15, pop75` and a linear combination of `sr, dpi, ddpi` is `res.cca$cor[1]`. The coefficients of the corresponding linear combinations can be found on the rows of components `xcoef` and `ycoef`

Question

Check that the different components of the output of `cancor()` satisfy all properties they should satisfy.

Solution

```
cc <- cor(
  as.matrix(pop) %*% as.matrix(res.cca$xcoef),
  as.matrix(oec) %*% as.matrix(res.cca$ycoef)
  )

diag(cc) - res.cca$cor
```

```
[1]  2.220446e-16 -5.551115e-17
```

Question

Design a suite of tests (using `testthat`) that any contender of the implementation provided by package `stats` should pass.

Solution

TODO

Package `CCA`

Abstract of CCA: An R Package to Extend Canonical Correlation Analysis

> Canonical correlations analysis (CCA) is an exploratory statistical method to highlight correlations between two data sets acquired on the same experimental units. The `cancor()` function in `R` (`R` Development Core Team 2007) performs the core of computations but further work was required to provide the user with additional tools to facilitate the interpretation of the results.

As in PCA, CA, MCA, several kinds of graphical representations can be displayed from the results of CCA:

1. a barplot of the squared canonical correlations (which tells us about the low rank approximations of $H_X \times H_Y$)
2. scatter plots for the initial variables X^j and Y^k (ako correlation circles)
3. scatter plots for the individuals (rows)
4. biplots

Applications

Question

1. Load `nutrimouse` dataset from `CCA`.
2. Insert the 4 elements of list `nutrimouse` in the global environment (see `list2env()`)

Solution

```
stopifnot(
  require(CCA)
)

data("nutrimouse")

e <- list2env(nutrimouse, .GlobalEnv)
```

Question

- Compute the cross correlation matrix between `gene` and `lipid`
- Visualize the cross correlation matrix

Solution

```
Y <- as.matrix(gene)   # 40 x 120
X <- as.matrix(lipid)  # 40 x 20
```

```
c_XY = corrr::correlate(cbind(X, Y))
```

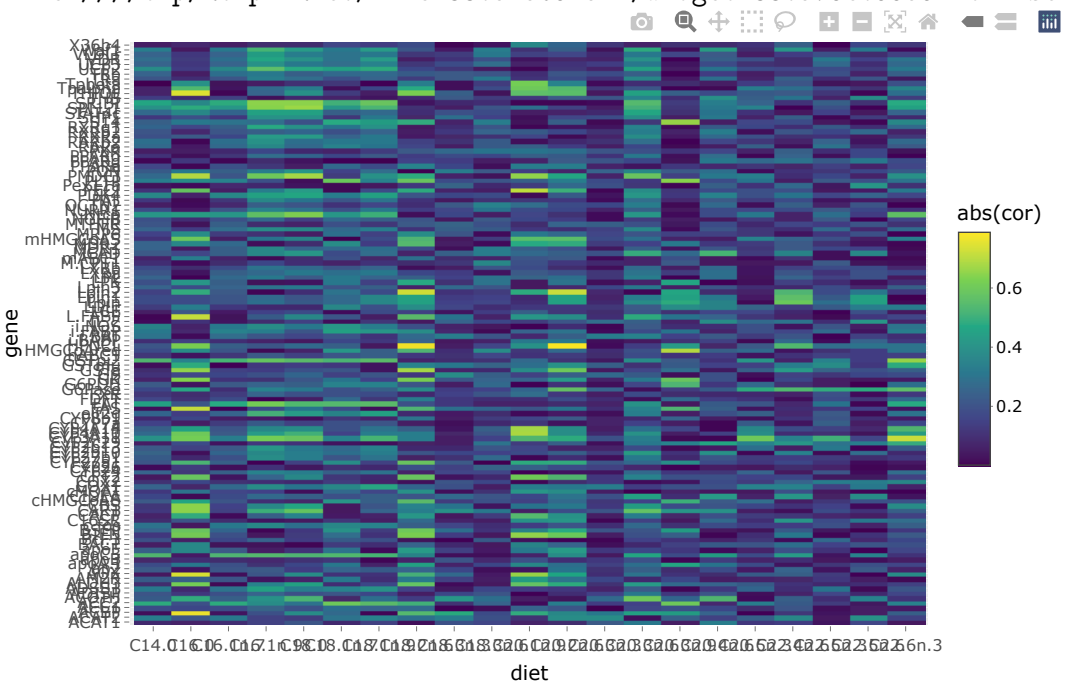
```
Correlation computed with
* Method: 'pearson'
* Missing treated using: 'pairwise.complete.obs'
```

```
c_XY_long <- c_XY |>
  tidyr::pivot_longer(cols=-c(term), names_to="term2", values_to="cor")
```

```
p  <- c_XY_long |>
  dplyr::filter(
    term %in% names(gene),
    term2 %in% names(lipid)
  ) |>
  ggplot() +
  aes(x=term2, y=term, fill=abs(cor)) +
  geom_tile() +
  scale_fill_viridis_c() +
  xlab("diet") +
  ylab("gene")

p |> plotly::ggplotly()
```

```
PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is in
file:////tmp/RtmpNZRDcb/file433ba1a69fa1f/widget433ba75cb99a9.html screenshot complete
```



Question

- Compute the canonical correlations between `gene` and `lipid`, save the result in `res.cca`
- Check the canonical correlations.
- Comment

Solution

```
res.cca <- cancor(as.matrix(gene), as.matrix(lipid))
```

```
sum(res.cca$cor)
```

```
[1] 21
```

$H_X \times H_Y$ has 21 eigenvalues equal to 1. As the subspaces defined by the columns in `gene` and `lipid` have dimensions at most `21` and `40`, $H_X \times H_Y$ equals the projection of $\mathbb{R}^{40}$ over the smallest subspace.

Question

Sample 10 columns from `gene` and `lipid` and repeat the operation

Solution

```
set.seed(42)
n <- 10

ss_gene <- gene |>
  dplyr::select(sample(names(gene),  n)) |>
  scale()

ss_lipid<- lipid |>
  dplyr::select(sample(names(lipid),  n)) |>
  scale()

res.cca <- cancor(as.matrix(ss_gene), as.matrix(ss_lipid))

res.cc <- cc(X=as.matrix(ss_gene), Y=as.matrix(ss_lipid))
```

Question

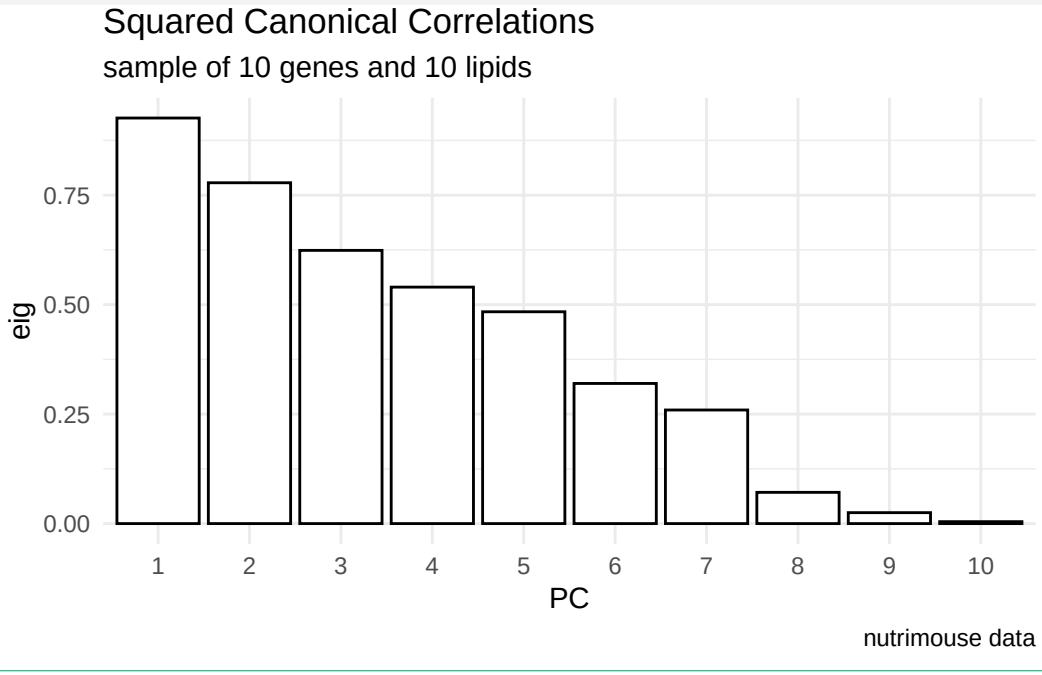
Screeplot

Solution

```
res.cca$cor |>
  as_tibble() |>
  gt::gt() |>
  gt::fmt_scientific() |>
  gt::tab_caption("Canonical correlations between `gene` columns of nutrimouse and  `l
```

value
9.62×10^{-1}
8.82×10^{-1}
7.90×10^{-1}
7.35×10^{-1}
6.96×10^{-1}
5.66×10^{-1}
5.09×10^{-1}
2.67×10^{-1}
1.58×10^{-1}
6.62×10^{-2}

```
res.cca$cor |>
  as_tibble() |>
  mutate(PC=as.factor(1:n), eig=value^2, percent=eig, cumulative=cumsum(eig)) |>
  ggplot() +
    aes(x=PC, y=eig, label=eig) +
    geom_col(fill="white", color="black") +
    theme_minimal() +
    labs(
      title="Squared Canonical Correlations",
      subtitle="sample of 10 genes and 10 lipids",
      caption="nutrimouse data"
    )
```



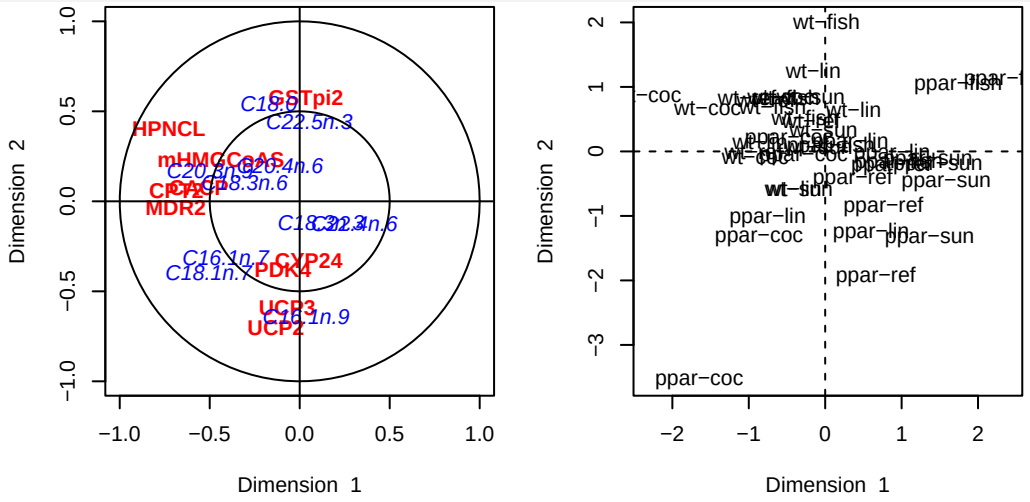
Question

Build a correlation circle

Solution

We use function `plt.cc` from `CCA`.

```
CCA::plt.cc(
  res.cc,
  var.label=T,
  ind.names = str_c(genotype, diet, sep="-")
)
```



References

https://www.jstatsoft.org/article/view/v023i12