PCA II: Swiss fertility data

2024-09-05

```
stopifnot(
  require(broom),
  require(DT),
  require(GGally),
  require(ggforce),
  require(ggfortify),
  require(ggvoronoi),
  require(glue),
  require(httr),
  require(magrittr),
  require(patchwork),
  require(skimr),
  require(tidymodels),
  require(tidyverse)
)

old_theme <- theme_set(theme_minimal())
```

- **M1 MIDS/MFA**
- **Université Paris Cité**
- Année 2024-2025
- Course Homepage

- Moodle

> **Objectives**

Swiss fertility data

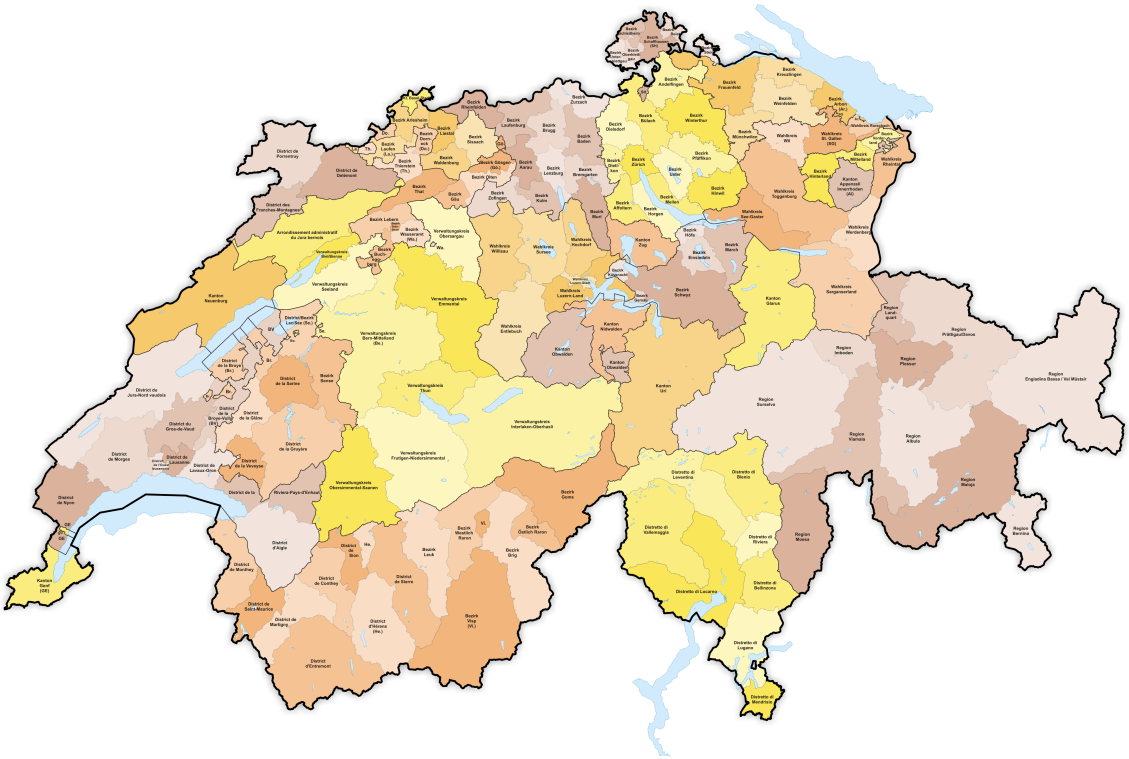
Dataset `swiss` from `datasets::swiss` connect fertility and social, economic data within 47 French-speaking districts in Switzerland.

- `Fertility` : fertility index
- `Agriculture` : jobs in agricultural sector
- `Examination` : literacy index (military examination)
- `Education` : proportion of people with successful secondary education
- `Catholic` : proportion of Catholics
- `Infant.Mortality` : mortality quotient at age 0

Fertility index (`Fertility`) is considered as the *response variable*

The social and economic variables are *covariates* (*explanatory* variables).

See European Fertility Project for more on this dataset.



PCA (Principal Component Analysis) is concerned with covariates.

```
data("swiss")

swiss %>%
  glimpse(50)
```

```
Rows: 47
Columns: 6
$ Fertility        <dbl> 80.2, 83.1, 92.5, 85.8,~
$ Agriculture      <dbl> 17.0, 45.1, 39.7, 36.5,~
$ Examination      <int> 15, 6, 5, 12, 17, 9, 16~
$ Education        <int> 12, 9, 5, 7, 15, 7, 7, ~
$ Catholic         <dbl> 9.96, 84.84, 93.40, 33.~
$ Infant.Mortality <dbl> 22.2, 22.2, 20.2, 20.3,~
```

Have a look at the documentation of the dataset

Describe the dataset

- Compute summary for each variable

solution

It is enough to call `summary()` on each column of `swiss`. This can be done in a functional programming style using package `purrr`. The collections of summaries can be rearranged so as to build a dataframe that is fit for reporting.

```
tt <- map_dfr(swiss, summary, .id = "var") %>%
  mutate(across(where(is.numeric), ~ round(.x, digits=1)))
```

```
tt %>%
  knitr::kable()
```

var	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Fertility	35.0	64.7	70.4	70.1	78.4	92.5
Agriculture	1.2	35.9	54.1	50.7	67.7	89.7
Examination	3.0	12.0	16.0	16.5	22.0	37.0
Education	1.0	6.0	8.0	11.0	12.0	53.0
Catholic	2.1	5.2	15.1	41.1	93.1	100.0
Infant.Mortality	10.8	18.1	20.0	19.9	21.7	26.6

Function `skim` from `skimr` delivers all univariate summaries in proper form.

```
foo <- swiss %>%
  select(-Fertility) %>%
  skim()
```

```
foobar <- foo %>%
  filter(skim_type=="numeric") %>%
  rename(variable=skim_variable)  %>%
    mutate(across(where(is.numeric), ~ round(.x, digits=1)))
```

```
foobar %>%
  knitr::kable()
```

skim_type	variable	n_missing	complete_rate	numeric.mean	numeric.sd	numeric.p0	numeric.p25	numeric.p50	numeric.p75	numeric.p100	numeric.hist
numeric	Agriculture	0	1	50.7	22.7	1.2	35.9	54.1	67.7	89.7	
numeric	Examination	0	1	16.5	8.0	3.0	12.0	16.0	22.0	37.0	
numeric	Education	0	1	11.0	9.6	1.0	6.0	8.0	12.0	53.0	
numeric	Catholic	0	1	41.1	41.7	2.1	5.2	15.1	93.1	100.0	
numeric	Infant.Mortality	0	1	19.9	2.9	10.8	18.1	20.0	21.7	26.6	

- Display graphic summary for each variable.

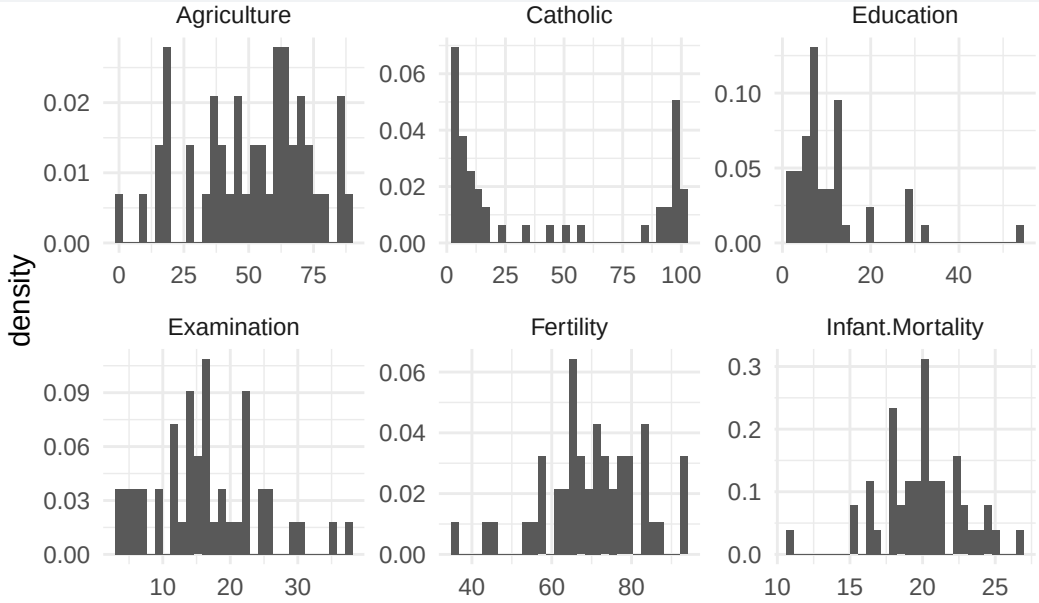
solution

We have to pick some graphical summary of the data. Boxplots and violine plots could be used if we look for concision.
We use histograms to get more details about each column.
Not that covariates have different meanings: `Agriculture`, `Catholic`, `Examination`, and `Education` are percentages with values between 0 and 100.
We have no details about the standardized fertility index `Fertility`
`Infant.Mortality` is also a rate:

> Infant mortality is the death of an infant before his or her first birthday. The infant mortality rate is the number of infant deaths for every 1,000 live births. In addition to giving us key information about maternal and infant health, the infant mortality rate is an important marker of the overall health of a society.

see Center for Desease Control
We reuse the function we have already developped during previous sessions.

```
make_biotifoul(swiss, .f = is.numeric)
```



Histograms reveal that our covariates have very different distributions.
Religious affiliation (`Catholic`) tells us that there two types of districts, which is reminiscent of the old principle *Cujus regio, ejus religio* , see Old Swiss Confederacy.
`Agriculture` shows that in most districts, agriculture was still a very important activity.
`Education` reveals that in all but a few districts, most children did not receive secondary education. `Examination` shows that some districts lag behind the bulk of districts. Even less exhibit a superior performance.
The two demographic variables `Fertility` and `Infant.Mortality` look roughly unimodal with a few extreme districts.

Investigate correlations

Compute, display and comment the sample correlation matrix.

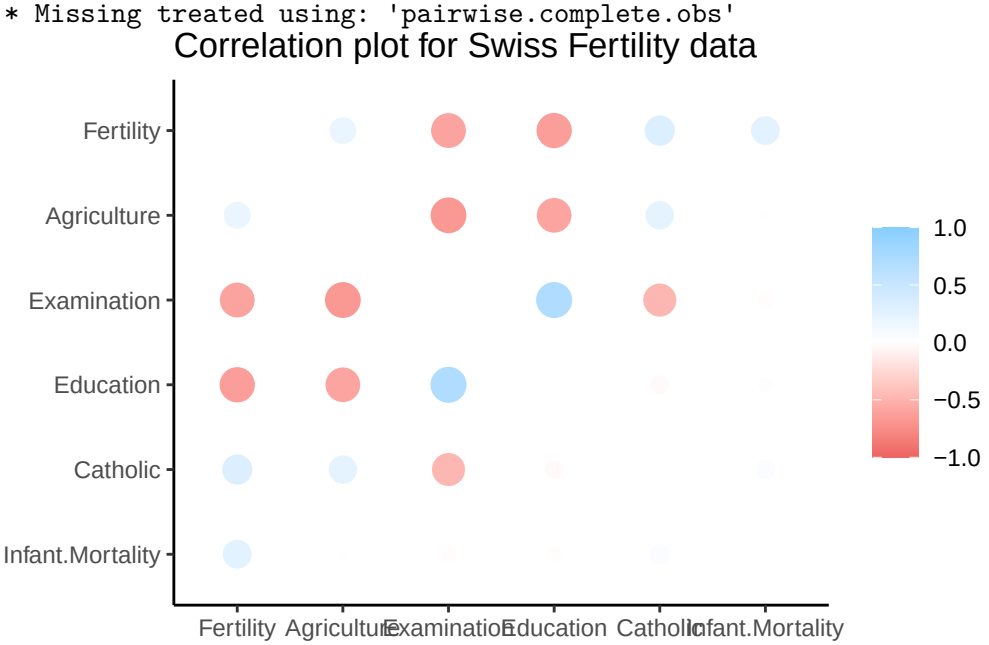
Display jointplots for each pair of variables.

solution

Package `corrr`, functions `correlate` and `rplot` provide a conveniemt tool.
Note that `rplot()` creates a graphical object of class `ggplot`. We can endow it with more layers.

```
corrr::correlate(swiss) %>%
  corrr::rplot() %>% +
  ggtitle("Correlation plot for Swiss Fertility data")
```

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



The high positive linear correlation between `Education` and `Examination` is moderately surprising. The negative correlation between the proportion of people involved in `Agriculture` and `Education` and `Examination`is also not too surprising. Secondary schooling required pupils from rural areas to move to cities.
A more intriguing observation concerns the pairs `Catholic` and `Examination` (negative correlation) and `Catholic` and `Education` (little correlation).
The response variable `Fertility` looks negatively correlated with `Examination` an `Education`. These correlations are worth being further explored. In Demography, the decline of Fertility is often associated with the the rise of women education. Note that `Examination` is about males, and that `Education` does not give details about the way women complete primary education.

Perform PCA on covariates

Pairwise analysis did not provide us with a clear and simple picture of the French-speaking districts.

Play with centering and scaling

We first call `prcomp()` with the default arguments for centering and scaling, that is, we center columns and do not attempt to standardize columns.

solution

```
pco <- swiss %>%
  select(-Fertility) %>%
  prcomp()
```

The result

solution

Hand-made centering of the dataframe

```
X <- select(swiss, -Fertility)

n <- nrow(X)

Y <-  (X - matrix(1, nrow = n, ncol=1) %*%  rep(1/n,n) %*% as.matrix(X))

Y <- as.matrix(Y)
```

```
tibble(var=names(X), mX=colMeans(X), mY=colMeans(Y))  %>%
  mutate(across(where(is.numeric), ~ round(.x, digits=2))) %>%
 knitr::kable()
```

var	mX	mY
Agriculture	50.66	0
Examination	16.49	0
Education	10.98	0
Catholic	41.14	0
Infant.Mortality	19.94	0

Function `scale(X, scale=F)` from base `R` does the job.

solution

```
svd_Y <-  svd(Y)

svd_Y %$%
  (as.matrix(Y) - u %*% diag(d) %*% t(v)) %>%
  norm(type = "F")                                                    ①
```

```
[1] 2.054251e-13
```

```
norm( diag(1, ncol(Y)) -
  (svd_Y %$% (t(v) %*% v)), 'F')                                      ②
```

```
[1] 1.261261e-15
```

Note that we used the exposing pipe `%$%` from `magrittr` to unpack `svd_Y` which is a list with class `svd` and members named `u`, `d` and `v`.
We could have used `with(,)` from base `R`.

solution

The matrix $1/nY^T \times Y$ is the covariance matrix of the covariates. The spectral decomposition of the symmetric Semi Definite Positive (SDP) matrix $1/nY^T \times Y$ is related with the SVD factorization of Y.
The spectral decomposition of $Y^T \times Y$ can be obtained using `eigen`.

```
(t(eigen(t(Y) %*% Y )$vectors) %*% svd_Y$v ) %>%
  round(digits=2)
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    0    0    0
[2,]    0   -1    0    0    0
[3,]    0    0    1    0    0
[4,]    0    0    0    1    0
[5,]    0    0    0    0    1
```

Here, the eigenvectors of $Y^T \times Y$ coincide with the right singular vectors of Y corresponding to non-zero singular values. Up to sign changes, it is always true when the non-zero singular values are pairwise distinct.

Now we check that `prcomp` is indeed a wrapper for `svd`.

```
(Y - pco$x %*% t(pco$rotation )) %>%
  round(digits = 2)  %>%
  head()
```

```
             Agriculture Examination Education Catholic Infant.Mortality
Courtelary             0           0         0        0                 0
Delemont               0           0         0        0                 0
Franches-Mnt           0           0         0        0                 0
Moutier                0           0         0        0                 0
Neuveville             0           0         0        0                 0
Porrentruy             0           0         0        0                 0
```

```
(svd_Y$v %*% t(pco$rotation )) %>%
  round(2)
```

```
     Agriculture Examination Education Catholic Infant.Mortality
[1,]           1           0         0        0                 0
[2,]           0           1         0        0                 0
[3,]           0           0         1        0                 0
[4,]           0           0         0        1                 0
[5,]           0           0         0        0                 1
```

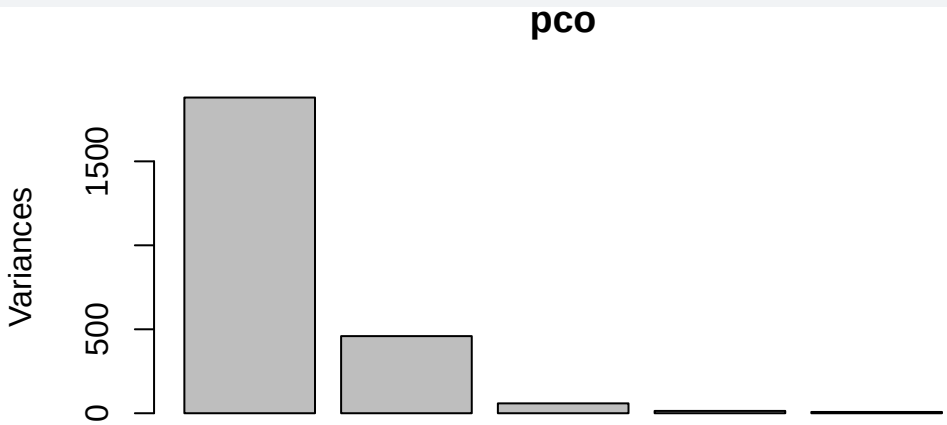
```
(t(pco$x) %*% pco$x) %>%
  round(2)
```

```
          PC1      PC2     PC3    PC4    PC5
PC1 86484.49     0.00    0.00   0.00   0.00
PC2     0.00 21127.44    0.00   0.00   0.00
PC3     0.00     0.00 2706.14   0.00   0.00
PC4     0.00     0.00    0.00 639.22   0.00
PC5     0.00     0.00    0.00   0.00 348.01
```

solution

Objects of class `pca` can be handled by generic functions like `plot`.

```
plot(pco)
```



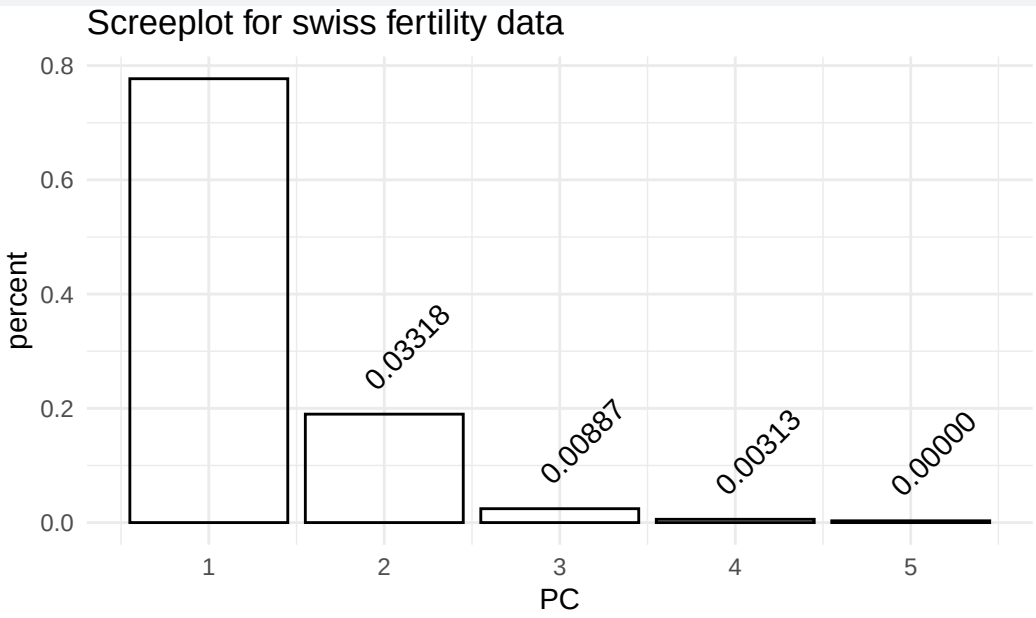
The displayed plot is the so-called *screeplot*, it provides information about the approximation of the decomposedmatrix by its truncated SVDs.

solution

```
p_screeplot <- . %>%                                                    ①
  tidy(matrix="pcs") %>% {
  ggplot(.) +
  aes(x=PC, y=percent, label=format(1.-cumulative,2)) +                 ③
  geom_text(angle=45, vjust=-1, hjust=-0.1) +
  geom_col(fill=NA, colour="black")                                     ②
  }
```

① Define a pipeline for building a screeplot
② Mind the braces on the right side of the first pipe
③ `1- percent` tell the reader about the relative Frobenious error achieved by keeping the first components of the SVD expansion.

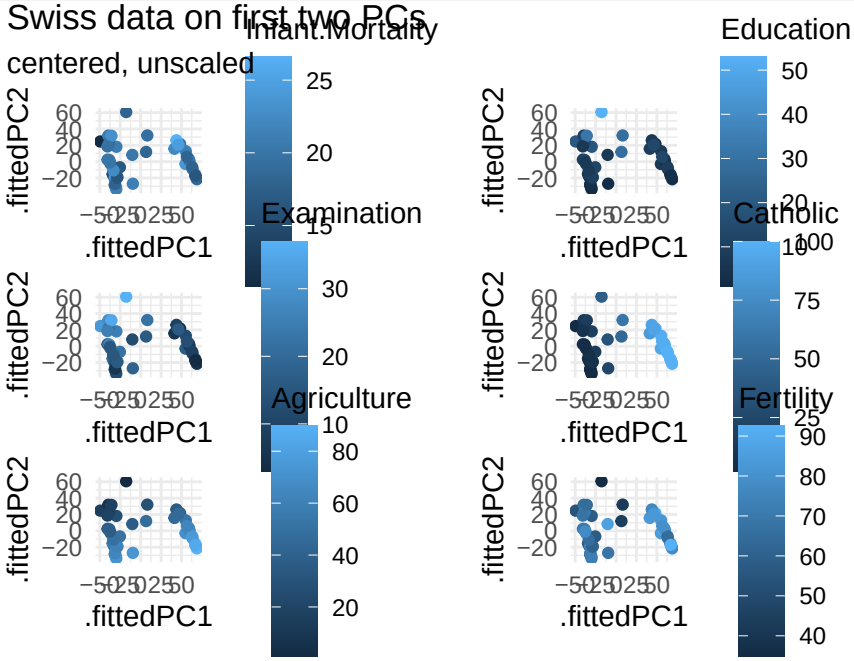
```
pco %>%
  p_screeplot() +
  labs(title="Screeplot for swiss fertility data",
  caption="Keeping the first two components is enough to achieve relative Froebenius r
```



Project the dataset on the first two principal components (perform dimension reduction) and build a scatterplot. Colour the points according to the value of original covariates.

solution

```
p <-  pco %>%                                                    ①
  augment(swiss) %>%
  ggplot() +
  aes(x=.fittedPC1, y=.fittedPC2, label=.rownames) +
  geom_point() +
  coord_fixed() +                                                ②
  ggrepel::geom_text_repel()

(p +
  aes(color=Infant.Mortality)) +
(p +
   aes(color=Education)) +
(p +
   aes(color=Examination)) +
(p +
   aes(color=Catholic)) +
(p +
   aes(color=Agriculture)) +
(p +
   aes(color=Fertility)) +
plot_layout(ncol = 2) +
plot_annotation(title="Swiss data on first two PCs" ,
                subtitle = "centered, unscaled")
```



solution

We can extract factor V from the SVD factorization using generic function `tidy` from package `broom`

```
pco %>%
  tidy(matrix="v") %>%
  glimpse()
```

```
Rows: 25
Columns: 3
$ column <chr> "Agriculture", "Agriculture", "Agriculture", "Agriculture", "Ag~
$ PC     <dbl> 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, ~
$ value  <dbl> 0.28151505, -0.88377692, -0.36961938, -0.02652821, -0.04863543,~
```

The result is a tibble in *long form*. It is worth pivoting the dataframe

```
pco %>%
  tidy(matrix="v") %>%
  pivot_wider(id_cols =column,
              names_from = PC,
              values_from = value)
```

```
# A tibble: 5 x 6
  column               `1`     `2`      `3`     `4`     `5`
  <chr>              <dbl>   <dbl>    <dbl>   <dbl>   <dbl>
1 Agriculture        0.282  -0.884  -0.370   -0.0265 -0.0486
2 Examination       -0.121   0.174  -0.450   -0.867   0.0332
3 Education         -0.0584  0.311  -0.807    0.485  -0.117
4 Catholic           0.950   0.303   0.00166 -0.0715  0.0223
5 Infant.Mortality   0.0105  0.0193  0.0985  -0.0867 -0.991
```

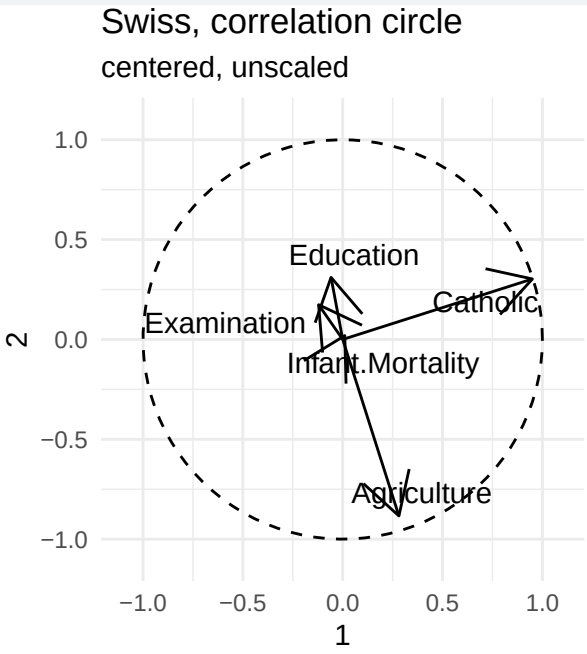
Think of the rows of `swiss` as vectors. Then matrix `v` In wide form, we readily access to the decomposition of the or

solution

```
prep_co_circle <-  . %>%
  tidy(matrix="v") %>%
  pivot_wider(id_cols =column,
              names_from = PC,
              values_from = value)
```

```
co_circle_ppl <-  (
    pco %>%
    prep_co_circle() %>%
    filter(F)
    ) %>%
  ggplot() +
  aes(x=`1`, y=`2`, label=column) +
  geom_segment(aes(xend=0, yend=0), arrow = grid::arrow(ends = "first")) +
  ggrepel::geom_text_repel() +
  coord_fixed() +
  xlim(c(-1.1, 1.1)) + ylim(c(-1.1, 1.1))  +
  annotate("path",
   x=0+1*cos(seq(0,2*pi,length.out=100)),
   y=0+1*sin(seq(0,2*pi,length.out=100)), linetype="dashed")
```

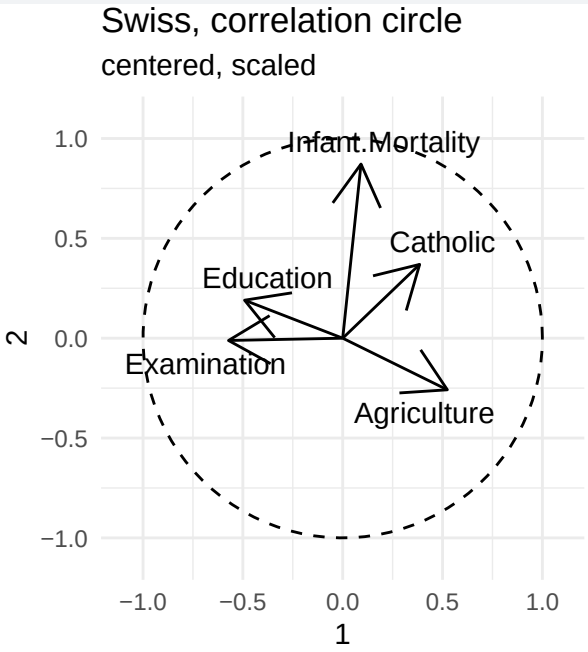
```
co_circle_ppl %+% (
  pco %>%
  prep_co_circle()
  )  +
  ggtitle("Swiss, correlation circle",
          subtitle = "centered, unscaled")
```




```
# pco %$% {
#   ifelse(!is.null(center), "centered", "not centered") ;
#   ifelse(!is.null(scale), "scaled", "not scaled")
# }
```

solution

```
pco2 <- select(swiss, -Fertility) %>%
 prcomp(scale. = T)

co_circle_ppl %+% (
  pco2 %>%
  prep_co_circle()
  )  +
  ggtitle("Swiss, correlation circle",
          subtitle = "centered, scaled")
```



Sanity checks

- X : data matrix after column centering (use `scale(., center=T, scale-F)`)

$$X$$

solution

```
X <-  as.matrix(select(swiss, -Fertility)) |>
  scale(center = T, scale=F)

# check centering, spot the difference in variances
X |>
  as_tibble() |>
  summarise(across(everything(), c(var, mean)))
```

```
# A tibble: 1 x 10
  Agriculture_1 Agriculture_2 Examination_1 Examination_2 Education_1
          <dbl>         <dbl>         <dbl>         <dbl>       <dbl>
1          516.      2.64e-15          63.6     -1.51e-16        92.5
# i 5 more variables: Education_2 <dbl>, Catholic_1 <dbl>, Catholic_2 <dbl>,
#   Infant.Mortality_1 <dbl>, Infant.Mortality_2 <dbl>
```

```
# should be 0
norm(X  %*% pco$rotation - pco$x)
```

```
[1] 0
```

```
# check the left singular vectors
pco$x %*% diag((pco$sdev)^(-1)) |>
  as_tibble() |>
  summarise(across(everything(), c(mean,var)))
```

```
Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if
`.name_repair` is omitted as of tibble 2.0.0.
i Using compatibility `.name_repair`.
# A tibble: 1 x 10
      V1_1  V1_2      V2_1  V2_2      V3_1  V3_2      V4_1  V4_2     V5_1  V5_2
     <dbl> <dbl>     <dbl> <dbl>     <dbl> <dbl>     <dbl> <dbl>    <dbl> <dbl>
1 7.44e-17     1 -1.05e-16     1 -8.24e-17  1.00 -6.84e-17     1 5.56e-16     1
```

```
#
pco$rotation %*% (diag((pco$sdev)^(-2)) %*% t(pco$x) %*% X)
```

```
                  Agriculture   Examination     Education      Catholic
Agriculture      4.600000e+01  6.994405e-15  9.325873e-15 -2.192690e-14
Examination      1.346007e-13  4.600000e+01  3.042011e-14  3.273354e-13
Education        1.090239e-13  2.825518e-14  4.600000e+01 -3.185507e-13
Catholic         1.054712e-15 -2.102485e-15 -4.982126e-15  4.600000e+01
Infant.Mortality 1.172396e-13 -2.442491e-14 -7.194245e-14 -1.971756e-13
                 Infant.Mortality
Agriculture         -5.329071e-15
Examination          4.440892e-16
Education           -1.598721e-14
Catholic             4.440892e-16
Infant.Mortality     4.600000e+01
```

solution

```
pco |>
  tidy(matrix="v") |>
  pivot_wider(id_cols =column,
              names_from = PC,
              values_from = value) |>
  rowwise() |>
  summarise(column, l2=sum((c_across(where(is.numeric)))^2))
```

```
# A tibble: 5 x 2
  column              l2
  <chr>            <dbl>
1 Agriculture       1.00
2 Examination       1.00
3 Education         1
4 Catholic          1
5 Infant.Mortality  1.00
```

Checking Orthogonality of V

solution

```
# checking that pco$rotation is an orthogonal matrix
t(pco$rotation) %*% pco$rotation
```

```
               PC1           PC2           PC3           PC4           PC5
PC1  1.000000e+00 -4.341417e-16 -7.220786e-17  2.710505e-18  3.469447e-18
PC2 -4.341417e-16  1.000000e+00  3.649425e-16 -8.001412e-17  6.938894e-17
PC3 -7.220786e-17  3.649425e-16  1.000000e+00  3.642919e-17 -1.387779e-17
PC4  2.710505e-18 -8.001412e-17  3.642919e-17  1.000000e+00  2.498002e-16
PC5  3.469447e-18  6.938894e-17 -1.387779e-17  2.498002e-16  1.000000e+00
```

```
pco$rotation %*% t(pco$rotation)
```

```
                 Agriculture   Examination     Education      Catholic
Agriculture      1.000000e+00  6.223320e-17  2.177078e-16  3.248270e-16
Examination      6.223320e-17  1.000000e+00 -5.316927e-16  1.517883e-17
Education        2.177078e-16 -5.316927e-16  1.000000e+00 -2.059984e-16
Catholic         3.248270e-16  1.517883e-17 -2.059984e-16  1.000000e+00
Infant.Mortality 6.245005e-17  2.983724e-16 -1.249001e-16 -1.734723e-17
                 Infant.Mortality
Agriculture          6.245005e-17
Examination          2.983724e-16
Education           -1.249001e-16
Catholic            -1.734723e-17
Infant.Mortality     1.000000e+00
```

Compare standardized and non-standardized PCA

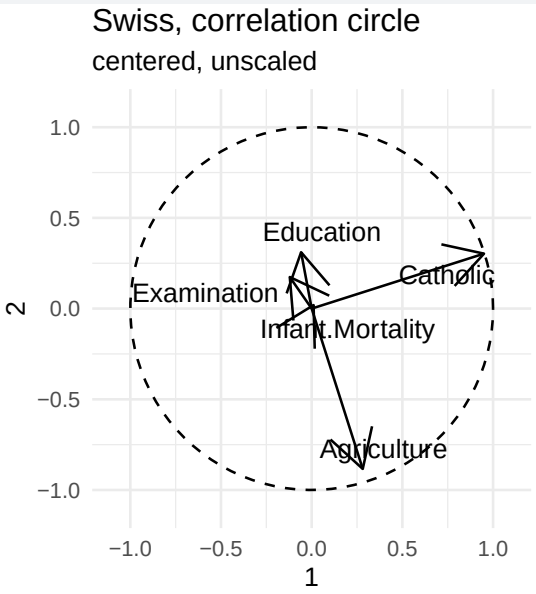
Pay attention to the correlation circles.

1. How well are variables represented?
2. Which variables contribute to the first axis?

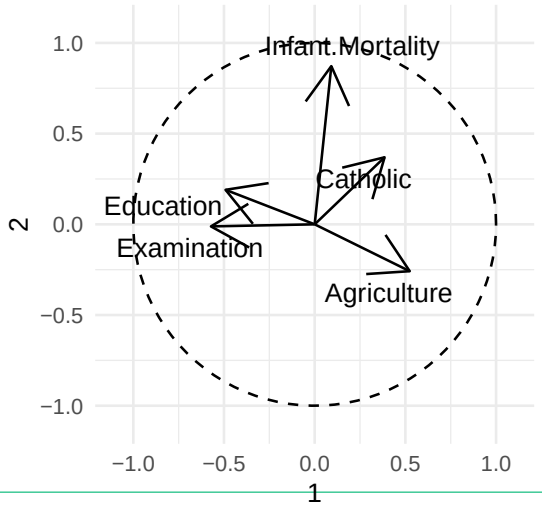
solution

```
pco_c <- swiss %>%
  select(-Fertility) %>%
  prcomp()

pco_cs <- swiss %>%
  select(-Fertility) %>%
  prcomp(scale.=T, center=T)
```

```
(co_circle_ppl %+% (pco_c %>%
  prep_co_circle())  +
  ggtitle("Swiss, correlation circle",
          subtitle = "centered, unscaled"))
```




```
(
  co_circle_ppl %+% (pco_cs %>%
  prep_co_circle())  +
  ggtitle("Swiss, correlation circle",
          subtitle = "centered, scaled")
)
```



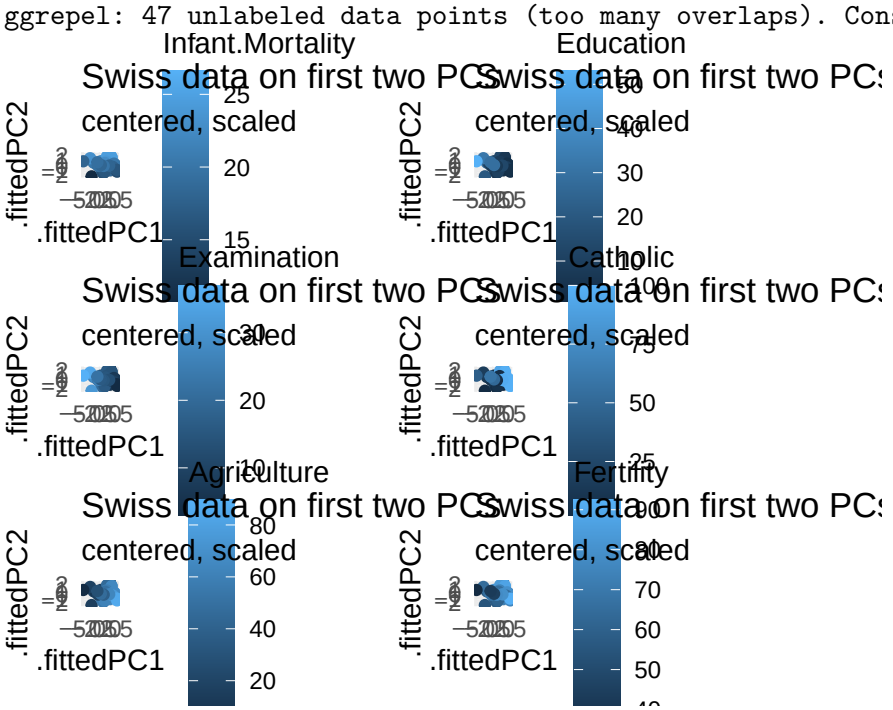
Explain the contrast between the two correlation circles.

In the sequel we focus on standardized PCA.

solution

```
q <-  p %+% (pco_cs %>%
  augment(swiss)) +
  ggtitle("Swiss data on first two PCs", subtitle = "centered, scaled")

(q +
  aes(color=Infant.Mortality)) +
(q +
   aes(color=Education)) +
(q +
   aes(color=Examination)) +
(q +
   aes(color=Catholic)) +
(q +
   aes(color=Agriculture)) +
(q +
   aes(color=Fertility)) +
plot_layout(ncol = 2)
```

```
Warning: ggrepel: 47 unlabeled data points (too many overlaps). Consider increasing ma
ggrepel: 47 unlabeled data points (too many overlaps). Consider increasing max.overlap
ggrepel: 47 unlabeled data points (too many overlaps). Consider increasing max.overlap
ggrepel: 47 unlabeled data points (too many overlaps). Consider increasing max.overlap
ggrepel: 47 unlabeled data points (too many overlaps). Consider increasing max.overlap
```

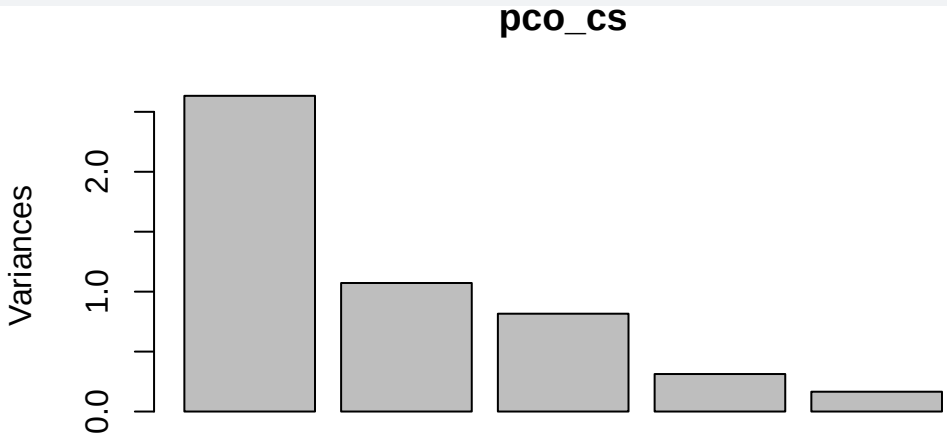


Investigate eigenvalues of covariance matrix

How many axes should we keep?

solution

```
p_screeplot %+% (pco_cs%>%
  tidy(matrix="pcs"))

plot(pco_cs)
```



Elbow rule: keep the first three PCs

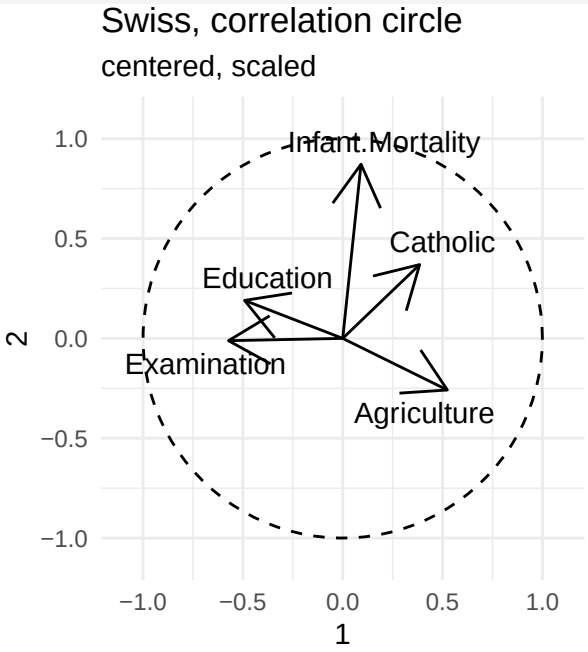
Provide an interpretation of the first two principal axes

1. Which variables contribute to the two first principal axes?

solution

This comes from the correlation circle. We rely on function `prep_co_circle` and on the graphical pipeline `co_circle_ppl`.

```
(
  co_circle_ppl %+%
    prep_co_circle(pco_cs) +
    ggtitle("Swiss, correlation circle",
            subtitle = "centered, scaled")
)
```

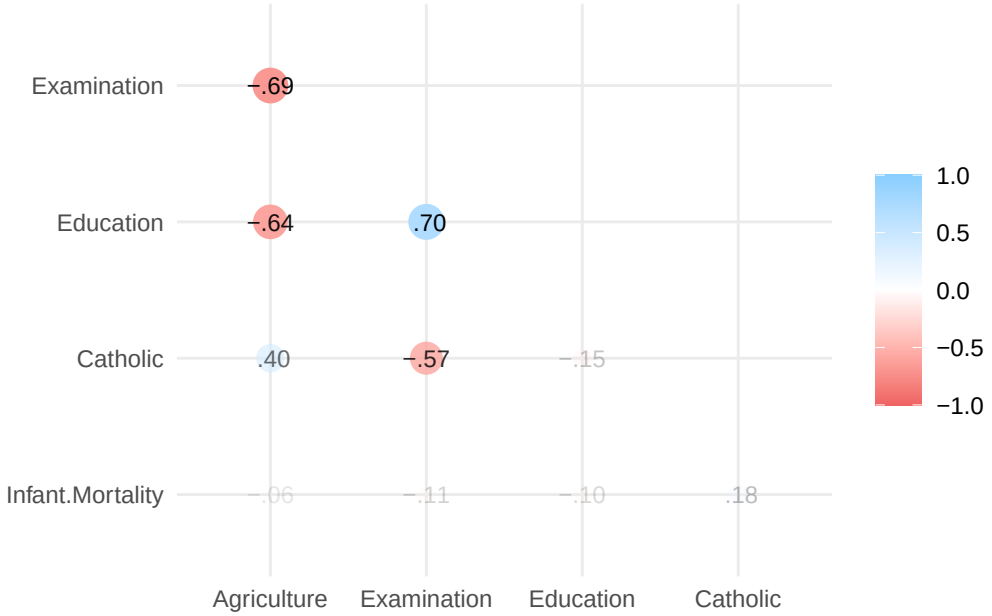


2. Analyze the signs of correlations between variables and axes?

solution

```
swiss |>
  select(-Fertility) |>
  corrr::correlate() |>
  corrr::shave() |>
  corrr::rplot(print_cor = T) +
  theme_minimal()
```

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



Add the `Fertility` variable

Plot again the correlation circle using the same principal axes as before, but add the `Fertility` variable. How does `Fertility` relate with covariates? with principal axes?

solution

```
U <-  pco_cs %$%    # exposition pipe
  (1/sqrt(nrow(x)-1) *x %*% diag((sdev)^(-1)))

Uprime <- with(pco_cs,
  1/sqrt(nrow(x)-1) *x %*% diag((sdev)^(-1)))

t(U) %*% U
```

```
              [,1]          [,2]          [,3]          [,4]          [,5]
[1,]  1.000000e+00 -1.717376e-16 1.110223e-16 -3.008119e-16  6.210310e-16
[2,] -1.717376e-16  1.000000e+00 2.498002e-16 -1.970266e-16  3.816392e-17
[3,]  1.110223e-16  2.498002e-16 1.000000e+00  4.523508e-15  5.828671e-16
[4,] -3.008119e-16 -1.970266e-16 4.523508e-15  1.000000e+00 -6.432029e-16
[5,]  6.210310e-16  3.816392e-17 5.828671e-16 -6.432029e-16  1.000000e+00
```

```
t(Uprime) %*% Uprime
```

```
              [,1]          [,2]          [,3]          [,4]          [,5]
[1,]  1.000000e+00 -1.717376e-16 1.110223e-16 -3.008119e-16  6.210310e-16
[2,] -1.717376e-16  1.000000e+00 2.498002e-16 -1.970266e-16  3.816392e-17
[3,]  1.110223e-16  2.498002e-16 1.000000e+00  4.523508e-15  5.828671e-16
[4,] -3.008119e-16 -1.970266e-16 4.523508e-15  1.000000e+00 -6.432029e-16
[5,]  6.210310e-16  3.816392e-17 5.828671e-16 -6.432029e-16  1.000000e+00
```

```
(norm(U,type = "F"))^2
```

```
[1] 5
```

Display individuals (districts)

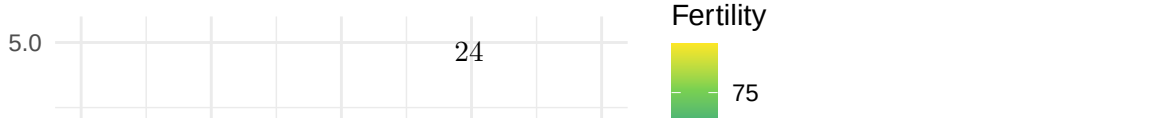
Comment

Biplot

solution

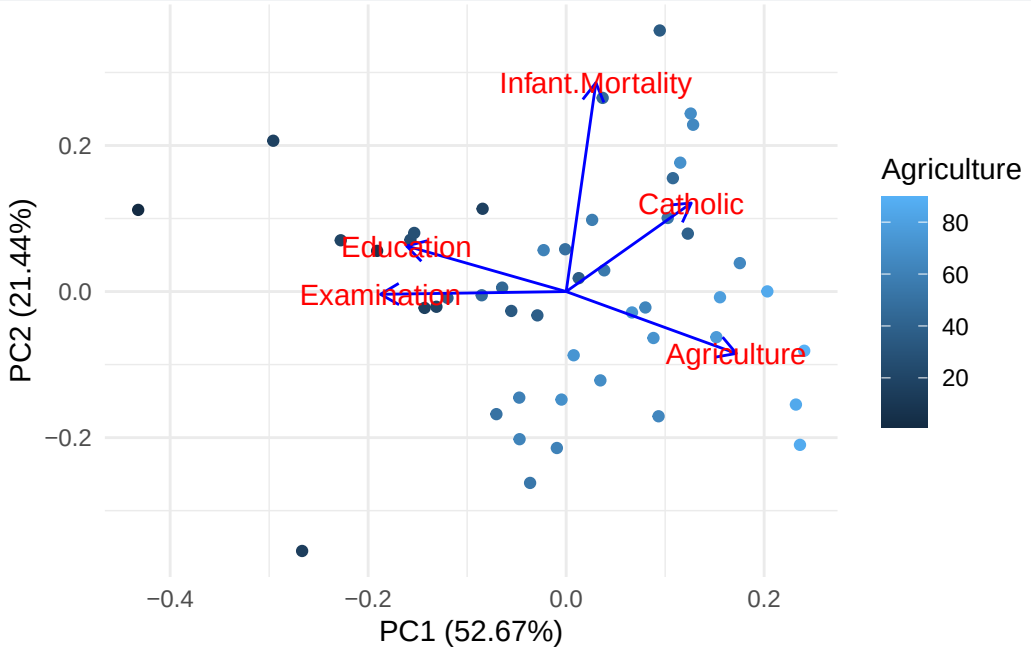
```
pco <- swiss %>%
  select(-Fertility) %>%
  prcomp(scale.=T)

df_cocirc <- pco %>%
  tidy(matrix="v") %>%
  pivot_wider(id_cols =column,
              names_from = PC,
              values_from = value)

augment(pco, data=swiss) %>%
  ggplot() +
  geom_point(aes(x=.fittedPC1,
                 y=.fittedPC2,
                 color=Fertility, label=.rownames)) +
  coord_fixed() +
  ggrepel::geom_text_repel(aes(x=.fittedPC1,
                               y=.fittedPC2,
                               color=Infant.Mortality,
                               label=.rownames)) +
  geom_segment(data=df_cocirc,
               mapping=aes(x= 4* `1`,
                           y= 4 * `2`,
                           linetype=factor(column),
                           label=column,
                           xend=0,
                           yend=0),
               arrow = grid::arrow(ends = "first",
                                   unit(.1, "inches")
                                  )) +
  scale_color_viridis_c() +
  xlim(c(-5,5)) +
  ylim(c(-5,5)) #+
```

```
Warning in geom_point(aes(x = .fittedPC1, y = .fittedPC2, color = Fertility, :
Ignoring unknown aesthetics: label
Warning in geom_segment(data = df_cocirc, mapping = aes(x = 4 * `1`, y = 4 * :
Ignoring unknown aesthetics: label
Warning: ggrepel: 37 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



solution

```
autoplot(pco,
         data=swiss,
         color="Agriculture",
         loadings = TRUE,
         loadings.colour = 'blue',
         loadings.label = TRUE)
```



References

https://scholar.google.com/citations?user=xbCKOYMAAAAJ&hl=fr&oi=ao