11 Principal Components

If two regressors are highly correlated, we can typically drop one of the regressors because it mostly contains the same information.

The idea of principal component regression is to exploit the correlations among the regressors to reduce their number while retaining as much of the original information as possible.

11.1 Principal Components

The principal components (PC) are linear combinations of the regressor variables that capture as much of the variation in the original variables as possible.

Principal Components

Let $\boldsymbol{X}_i$ be a k-variate vector of regressor variables.

The **first principal component** is $P_{i1} = \boldsymbol{w}_1' \boldsymbol{X}_i$, where $\boldsymbol{w}_1$ satisfies

$$\boldsymbol{w}_1 = \text{argmax}_{\boldsymbol{w}'\boldsymbol{w}=1} \; Var[\boldsymbol{w}'\boldsymbol{X}_i]$$

The **second principal component** is $P_{i2} = \boldsymbol{w}_2' \boldsymbol{X}_i$, where $\boldsymbol{w}_2$ satisfies

$$\boldsymbol{w}_2 = \text{argmax}_{\substack{\boldsymbol{w}'\boldsymbol{w}=1 \\ \boldsymbol{w}'\boldsymbol{w}_1=0}} \; Var[\boldsymbol{w}'\boldsymbol{X}_i]$$

The l**-th principal component** is $P_{il} = \boldsymbol{w}_l' \boldsymbol{X}_i$, where $\boldsymbol{w}_l$ satisfies

$$\boldsymbol{w}_l = \text{argmax}_{\substack{\boldsymbol{w}'\boldsymbol{w}=1 \\ \boldsymbol{w}'\boldsymbol{w}_1=\ldots=\boldsymbol{w}'\boldsymbol{w}_{l-1}=0}} \; Var[\boldsymbol{w}'\boldsymbol{X}_i]$$

A k-variate regressor vector $\boldsymbol{X}_i$ has k principal components $P_{i1}, \ldots, P_{ik}$ and k corresponding **weights** or **principal component loadings** $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_k$.

By definition, the principal components are descendingly ordered by their variance:

$$Var[P_{i1}] \geq Var[P_{i2}] \geq \ldots \geq Var[P_{ik}] \geq 0$$

The principal component weights are orthonormal:

$$\boldsymbol{w}_i' \boldsymbol{w}_j = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

Moreover, $\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_k$ form an orthonormal basis for the k-dimensional vector space $\mathbb{R}^k$. The regressor vector admits the following decomposition into its principal components:

$$\boldsymbol{X}_i = \sum_{l=1}^{k} P_{il} \boldsymbol{w}_l \tag{11.1}$$

The decomposition of a dataset into its principal components is called **principal component analysis (PCA)**.

11.2 Analytical PCA Solution

In this subsection, we will use some matrix calculus and eigenvalue theory. To recap the relevant matrix algebra, the following resources will be useful:

- Eigenvalues and Eigenvectors: https://matrix.svenotto.com/04_furtherconcepts.html
- Derivative rules for vectors: https://matrix.svenotto.com/05_calculus.html

The maximization problem for the first principal component is

$$\max_{\boldsymbol{w}} \; Var[\boldsymbol{w}'\boldsymbol{X}_i] \quad \text{subject to } \boldsymbol{w}'\boldsymbol{w} = 1. \tag{11.2}$$

The variance of interest can be rewritten as

$$\begin{aligned} Var[\boldsymbol{w}'\boldsymbol{X}_i] &= E[(\boldsymbol{w}'(\boldsymbol{X}_i - E[\boldsymbol{X}_i]))^2] \\ &= E[(\boldsymbol{w}'(\boldsymbol{X}_i - E[\boldsymbol{X}_i]))((\boldsymbol{X}_i - E[\boldsymbol{X}_i])'\boldsymbol{w})] \\ &= \boldsymbol{w}'E[(\boldsymbol{X}_i - E[\boldsymbol{X}_i])(\boldsymbol{X}_i - E[\boldsymbol{X}_i])']\boldsymbol{w} \\ &= \boldsymbol{w}'\Sigma\boldsymbol{w} \end{aligned}$$

where $\Sigma = Var[\boldsymbol{X}_i]$ is the population covariance matrix of $\boldsymbol{X}_i$. Thus, the constrained maximization problem Equation 11.2 has the Lagrangian

$$\mathcal{L}(\boldsymbol{w}, \lambda) = \boldsymbol{w}'\Sigma\boldsymbol{w} - \lambda(\boldsymbol{w}'\boldsymbol{w} - 1),$$

where λ is a Lagrange multiplier.

Recall the derivative rules for vectors: If $\boldsymbol{A}$ is a symmetric matrix, then the derivative of $\boldsymbol{a}'\boldsymbol{A}\boldsymbol{a}$ with respect to $\boldsymbol{a}$ is $2\boldsymbol{A}\boldsymbol{a}$. Therefore, the first order condition with respect to $\boldsymbol{w}$ is

$$\Sigma\boldsymbol{w} = \lambda\boldsymbol{w}. \tag{11.3}$$

The pair $(\lambda, \boldsymbol{w})$ must satisfy the eigenequation Equation 11.3, which is precisely the eigenequation which defines an eigenvalue-eigenvector pair. The Lagrange multiplier λ must be an eigenvalue of Σ and the weight vector $\boldsymbol{w}$ must be a corresponding eigenvector.

By the first order condition with respect to λ,

$$\boldsymbol{w}'\boldsymbol{w} = 1,$$

the eigenvector is normalized to length 1.

Therefore, the variance of interest is

$$Var[\boldsymbol{w}'\boldsymbol{X}_i] = \boldsymbol{w}'\Sigma\boldsymbol{w} = \boldsymbol{w}'(\lambda\boldsymbol{w}) = \lambda. \tag{11.4}$$

Consequently, $Var[\boldsymbol{w}'\boldsymbol{X}_i]$ must be an eigenvalue of Σ and $\boldsymbol{w}$ is a corresponding normalized eigenvector.

The expression $Var[\boldsymbol{w}'\boldsymbol{X}_i] = \lambda$ is maximized if we use the largest eigenvalue $\lambda = \lambda_1$. Consequently, the variance of the first principal component P_{i1} is equal to the largest eigenvalue λ_1 of Σ, and the first principal component weight $\boldsymbol{w}_1$ is a normalized eigenvector corresponding to the eigenvalue λ_1.

Analogously, the second principal component weight $\boldsymbol{w}_2$ must also be a normalized eigenvector of Σ with the additional restriction that it is orthogonal to $\boldsymbol{w}_1$. Therefore, it cannot be an eigenvector corresponding to the first eigenvalue, and we use the second largest eigenvalue $\lambda = \lambda_2$ to maximize Equation 11.4.

The variance of the second principal component P_{i2} is equal to the second largest eigenvalue λ_2 of Σ, and the second principal component weight $\boldsymbol{w}_2$ is a corresponding normalized eigenvector.

To continue this pattern, the variance of the l-th principal component P_{il} is equal to the l-th largest eigenvalue λ_l of Σ, and the l-th principal component weight $\boldsymbol{w}_l$ is a corresponding normalized eigenvector.

Principal Components Solution

Let Σ be the covariance matrix of the k-variate vector of regressor variables $\boldsymbol{X}_i$, let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k \geq 0$ be the eigenvalues ordered in descending order of Σ, and let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ be corresponding orthonormal eigenvectors.

- The principal component weights are $\boldsymbol{w}_l = \boldsymbol{v}_l$ for $l = 1, \ldots, k$
- The principal components are $P_{il} = \boldsymbol{v}_l'\boldsymbol{X}_i$, and they have the properties

$$Var[P_{il}] = \lambda_l, \quad Cov(P_{il}, P_{im}) = 0, \quad l \neq m.$$

Principal components are uncorrelated because

$$\begin{aligned} Cov(P_{im}, P_{il}) &= E[\boldsymbol{w}_m'(\boldsymbol{X}_i - E[\boldsymbol{X}_i])(\boldsymbol{X}_i - E[\boldsymbol{X}_i])'\boldsymbol{w}_l] \\ &= \boldsymbol{w}_m'\Sigma\boldsymbol{w}_l = \lambda_m\boldsymbol{w}_m'\boldsymbol{w}_l, \end{aligned}$$

where $\boldsymbol{w}_m'\boldsymbol{w}_l = 1$ if $m = l$ and $\boldsymbol{w}_m'\boldsymbol{w}_l = 0$ if $m \neq l$

11.3 Sample principal components

The covariance matrix $\Sigma = Var[\boldsymbol{X}_i]$ is unknown in practice. Instead, we estimate it from the sample $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$:

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{X}_i - \overline{\boldsymbol{X}})(\boldsymbol{X}_i - \overline{\boldsymbol{X}})'.$$

Let $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \dots, \hat{\lambda}_k \geq 0$ be the eigenvalues of $\widehat{\boldsymbol{\Sigma}}$ and let $\hat{\boldsymbol{v}}_1, \dots, \hat{\boldsymbol{v}}_k$ be corresponding orthonormal eigenvectors. Then,

- The l-th sample principal component for observation i is

$$\widehat{P}_{il} = \widehat{\boldsymbol{w}}_l' \boldsymbol{X}_i$$

- The l-th sample principal component weight vector is

$$\widehat{\boldsymbol{w}}_l = \hat{\boldsymbol{v}}_l$$

- The (adjusted) sample variance of the l-th sample principal components series $\widehat{P}_{1l}, \dots, \widehat{P}_{nl}$ is $\hat{\lambda}_l$, and the sample covariances of different principal components series are zero.

11.4 PCA in R

Let's compute the sample principal components of the `mtcars` dataset:

```r
pca = prcomp(mtcars)
## the principal components are arranged by columns
## first few rows of principal components:
pca$x |> head()
```

```
                          PC1       PC2       PC3        PC4        PC5
Mazda RX4          -79.596425  2.132241 -2.153336 -2.7073437 -0.7023522
Mazda RX4 Wag      -79.598570  2.147487 -2.215124 -2.1782888 -0.8843859
Datsun 710        -133.894096 -5.057570 -2.137950  0.3460330  1.1061111
Hornet 4 Drive       8.516559 44.985630  1.233763  0.8273631  0.4240145
Hornet Sportabout  128.686342 30.817402  3.343421 -0.5211000  0.7365801
Valiant            -23.220146 35.106518 -3.259562  1.4005360  0.8029768
                          PC6          PC7         PC8        PC9        PC10
Mazda RX4         -0.31486106 -0.098695018  0.07789812 -0.2000092 -0.29008191
Mazda RX4 Wag     -0.45343873 -0.003554594  0.09566630 -0.3533243 -0.19283553
Datsun 710         1.17298584  0.005755581 -0.13624782 -0.1976423  0.07634353
```

```
Hornet 4 Drive    -0.05789705 -0.024307168 -0.22120800  0.3559844 -0.09057039
Hornet Sportabout -0.33290957  0.106304777  0.05301719  0.1532714 -0.18862217
Valiant           -0.08837864  0.238946304 -0.42390551  0.1012944 -0.03769010
                        PC11
Mazda RX4         -0.1057706
Mazda RX4 Wag     -0.1069047
Datsun 710        -0.2668713
Hornet 4 Drive    -0.2088354
Hornet Sportabout  0.1092563
Valiant           -0.2757693
```

```
## the principal components weights
pca$rotation |> head()
```

```
              PC1          PC2         PC3          PC4         PC5
mpg  -0.038118199  0.009184847  0.98207085  0.047634784 -0.08832843
cyl   0.012035150 -0.003372487 -0.06348394 -0.227991962  0.23872590
disp  0.899568146  0.435372320  0.03144266 -0.005086826 -0.01073597
hp    0.434784387 -0.899307303  0.02509305  0.035715638  0.01655194
drat -0.002660077 -0.003900205  0.03972493 -0.057129357 -0.13332765
wt    0.006239405  0.004861023 -0.08491026  0.127962867 -0.24354296
              PC6          PC7           PC8          PC9         PC10
mpg  -0.143790084 -0.039239174 -2.271040e-02 -0.002790139  0.030630361
cyl  -0.793818050  0.425011021  1.890403e-01  0.042677206  0.131718534
disp  0.007424138  0.000582398  5.841464e-04  0.003532713 -0.005399132
hp    0.001653685 -0.002212538 -4.748087e-06 -0.003734085  0.001862554
drat  0.227229260  0.034847411  9.385817e-01 -0.014131110  0.184102094
wt   -0.127142296 -0.186558915 -1.561907e-01 -0.390600261  0.829886844
              PC11
mpg   0.0158569365
cyl  -0.1454453628
disp -0.0009420262
hp    0.0021526102
drat  0.0973818815
wt    0.0198581635
```

```
## the standard deviations of the principal components
## are the square roots of the sample eigenvalues
pca$sdev
```

```
 [1] 136.5330479  38.1480776   3.0710166   1.3066508   0.9064862   0.6635411
 [7]   0.3085791   0.2859604   0.2506973   0.2106519   0.1984238
```

Principal components are sensitive to the scaling of the data. Consequently, it is recommended to first scale each variable in the dataset to have mean zero and unit variance: `scale(mtcars)`. In this case, Σ is the correlation matrix.

```r
pca = mtcars |> scale() |> prcomp()
pca$x |> head()
```

```
                          PC1        PC2        PC3         PC4        PC5
Mazda RX4         -0.64686274  1.7081142 -0.5917309  0.11370221  0.9455234
Mazda RX4 Wag     -0.61948315  1.5256219 -0.3763013  0.19912121  1.0166807
Datsun 710        -2.73562427 -0.1441501 -0.2374391 -0.24521545 -0.3987623
Hornet 4 Drive    -0.30686063 -2.3258038 -0.1336213 -0.50380035 -0.5492089
Hornet Sportabout  1.94339268 -0.7425211 -1.1165366  0.07446196 -0.2075157
Valiant           -0.05525342 -2.7421229  0.1612456 -0.97516743 -0.2116654
                          PC6         PC7          PC8         PC9        PC10
Mazda RX4         -0.01698737 -0.42648652  0.009631217 -0.14642303  0.06670350
Mazda RX4 Wag     -0.24172464 -0.41620046  0.084520213 -0.07452829  0.12692766
Datsun 710        -0.34876781 -0.60884146 -0.585255765  0.13122859 -0.04573787
Hornet 4 Drive     0.01929700 -0.04036075  0.049583029 -0.22021812  0.06039981
Hornet Sportabout  0.14919276  0.38350816  0.160297757  0.02117623  0.05983003
Valiant           -0.24383585 -0.29464160 -0.256612420  0.03222907  0.20165466
                         PC11
Mazda RX4          0.17969357
Mazda RX4 Wag      0.08864426
Datsun 710        -0.09463291
Hornet 4 Drive     0.14761127
Hornet Sportabout  0.14640690
Valiant            0.01954506
```

```r
pca$rotation |> head()
```

```
            PC1         PC2         PC3          PC4         PC5         PC6
mpg  -0.3625305  0.01612440 -0.22574419 -0.022540255 -0.10284468 -0.10879743
cyl   0.3739160  0.04374371 -0.17531118 -0.002591838 -0.05848381  0.16855369
disp  0.3681852 -0.04932413 -0.06148414  0.256607885 -0.39399530 -0.33616451
hp    0.3300569  0.24878402  0.14001476 -0.067676157 -0.54004744  0.07143563
drat -0.2941514  0.27469408  0.16118879  0.854828743 -0.07732727  0.24449705
wt    0.3461033 -0.14303825  0.34181851  0.245899314  0.07502912 -0.46493964
             PC7         PC8         PC9        PC10        PC11
mpg  0.367723810  0.754091423 -0.23570162 -0.13928524 -0.12489563
cyl  0.057277736  0.230824925 -0.05403527  0.84641949 -0.14069544
```

```
disp  0.214303077 -0.001142134 -0.19842785 -0.04937979  0.66060648
hp   -0.001495989  0.222358441  0.57583007 -0.24782351 -0.25649206
drat  0.021119857 -0.032193501  0.04690123  0.10149369 -0.03953025
wt   -0.020668302  0.008571929 -0.35949825 -0.09439426 -0.56744870
```

```r
pca$sdev
```

```
 [1] 2.5706809 1.6280258 0.7919579 0.5192277 0.4727061 0.4599958 0.3677798
 [8] 0.3505730 0.2775728 0.2281128 0.1484736
```

11.5 Variance of principal components

Since the sample principal components are uncorrelated, the total variation in the data is

$$Var\left[\sum_{m=1}^{k} \widehat{P}_{im}\right] = \sum_{m=1}^{k} Var[\widehat{P}_{im}] = \sum_{m=1}^{k} \hat{\lambda}_l.$$

The proportion of variance explained by the l-th principal component is

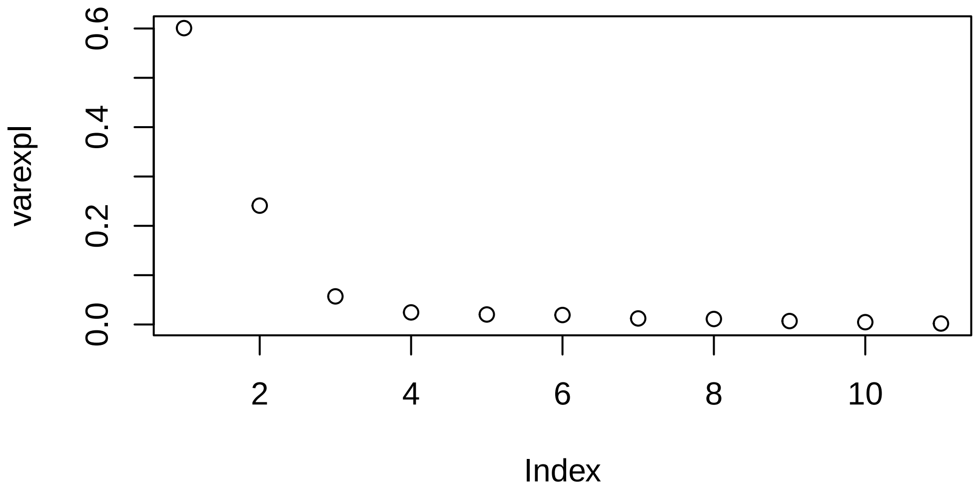
$$\frac{Var[\widehat{P}_{il}]}{Var[\sum_{m=1}^{k} \widehat{P}_{im}]} = \frac{\hat{\lambda}_l}{\sum_{m=1}^{k} \hat{\lambda}_m}$$

A scree plot is useful to see how much each principal component contributes to the total variation:

```r
pcvar = pca$sdev^2
varexpl = pcvar/sum(pcvar)
varexpl
```

```
 [1] 0.600763659 0.240951627 0.057017934 0.024508858 0.020313737 0.019236011
 [7] 0.012296544 0.011172858 0.007004241 0.004730495 0.002004037
```

```r
plot(varexpl)
```




```
cumsum(varexpl)
```

```
 [1] 0.6007637 0.8417153 0.8987332 0.9232421 0.9435558 0.9627918 0.9750884
 [8] 0.9862612 0.9932655 0.9979960 1.0000000
```

The first principal component explains more than 60% of the variation, the first four explain more than 90% of the variation, the first 6 more than 95%, and the first 9 principal components more than 99% of the variation.

11.6 Linear regression with principal components

Principal components can be used to estimate the high-dimensional (large k) linear regression model

$$Y_i = \boldsymbol{X}_i'\boldsymbol{\beta} + u_i, \quad i = 1, \dots, n.$$

While ridge and lasso shrink coefficients to prevent overfitting, PCA reduces dimensionality by transforming variables into orthogonal components before estimation.

Since the principal component weights $\boldsymbol{w}_1, \dots, \boldsymbol{w}_k$ form a basis of $\mathbb{R}^k$, the regressors have the basis representation given by Equation 11.1. Similarly, we can represent the coefficient vector in terms of the principal component basis:

$$\boldsymbol{\beta} = \sum_{l=1}^{k} \theta_l \boldsymbol{w}_l, \quad \theta_l = \boldsymbol{w}_l'\boldsymbol{\beta}. \tag{11.5}$$

Inserting in the regression function gives

$$\boldsymbol{X}_i'\boldsymbol{\beta} = \sum_{l=1}^{k} \underbrace{\boldsymbol{X}_i'\boldsymbol{w}_l}_{=P_{il}} \theta_l,$$

and the regression equation becomes

$$Y_i = \sum_{l=1}^{k} P_{il}\theta_l + u_i. \tag{11.6}$$

This regression equation is convenient because the regressors P_{il} are uncorrelated, and OLS estimates for θ_l can be inserted back into Equation 11.5 to get an estimate for $\boldsymbol{\beta}$.

When k is large, this approach is still prone to overfitting. The k principal components of $\boldsymbol{X}_i$ explain 100% of its variance, but it may be reasonable to select a smaller number of principal components $p < k$ that explain 95% or 99% of the variance.

The remaining $k - p$ principal components explain only 5% or 1% of the variance. The idea is that we truncate the model by assuming that the remaining principal components contain only noise that is uncorrelated with Y_i.

Assumption (PC): $E[P_{im}Y_i] = 0$ for all $m = p + 1, \dots, k$.

This assumption implies that the components with indices larger than p contribute no systematic predictive power for Y_i, and hence only introduce noise.

Because the principal components are uncorrelated, we have $\theta_l = E[Y_i P_{il}]/E[P_{il}^2]$, and, therefore $\theta_m = 0$ for $m = p + 1, \dots, k$. Consequently,

$$\boldsymbol{\beta} = \sum_{l=1}^{p} \theta_l \boldsymbol{w}_l, \tag{11.7}$$

and Equation 11.6 becomes a factor model with p factors:

$$Y_i = \sum_{l=1}^{p} \theta_l P_{il} + u_i = \boldsymbol{P}_i'\boldsymbol{\theta} + u_i,$$

where $\boldsymbol{P}_i = (P_{i1}, \dots, P_{ip})'$ and $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p)'$. The least squares estimator of $\boldsymbol{\theta}$ using the regressors $\boldsymbol{P}_i$, $i = 1, \dots n$ can then be inserted to Equation 11.7 to obtain an estimate for $\boldsymbol{\beta}$.

In practice, the principal components are unknown and must be replaced by the first p sample principal components

$$\widehat{\boldsymbol{P}}_i = (\widehat{P}_{i1}, \dots, \widehat{P}_{ip})', \quad \widehat{P}_{il} = \widehat{\boldsymbol{w}}_l'\boldsymbol{X}_i.$$

The feasible least squares estimator for θ is

$$\widehat{\boldsymbol{\theta}} = (\widehat{\theta}_1, \dots, \widehat{\theta}_p)' = \bigg(\sum_{i=1}^{n} \widehat{\boldsymbol{P}}_i \widehat{\boldsymbol{P}}_i'\bigg)^{-1} \sum_{i=1}^{n} \widehat{\boldsymbol{P}}_i Y_i,$$

and the principal components estimator for $\boldsymbol{\beta}$ is

$$\widehat{\boldsymbol{\beta}}_{pc} = \sum_{l=1}^{p} \widehat{\theta}_l \widehat{\boldsymbol{w}}_l.$$

11.7 Selecting the number of factors

To select the number of principal components, one practical approach is to choose those that explain a pre-specified percentage (90-99%) of the total variance.

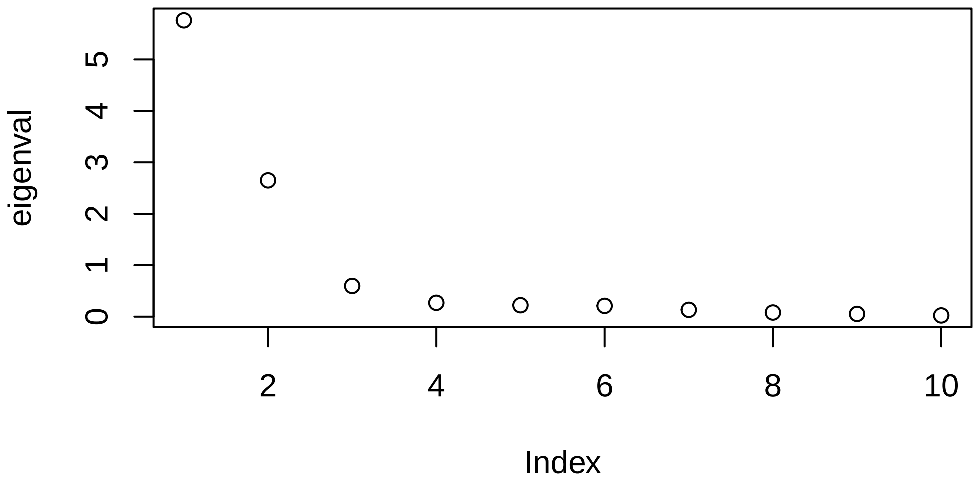
```
Y = mtcars$mpg
X = model.matrix(mpg ~., data = mtcars)[,-1] |> scale()
## principal component analysis
pca = prcomp(X)
P = pca$x #full matrix of all principal components
## variance explained
eigenval = pca$sdev^2
varexpl = eigenval/sum(eigenval)
cumsum(varexpl)
```

```
 [1] 0.5760217 0.8409861 0.9007075 0.9276582 0.9498832 0.9708950 0.9841870
 [8] 0.9922551 0.9976204 1.0000000
```

The first four principal components explain more than 92% of the variance, and the first seven more than 98%.

Another method involves creating a scree plot to display the eigenvalues (variances) for each principal component and identifying the point where the eigenvalues sharply drop (elbow point).

```
plot(eigenval)
```



We find an elbow at four principal components.

Selecting the number of principal components, similar to shrinkage estimation, involves balancing variance and bias. If the Assumption (PC) holds, the PC estimator is unbiased; if it doesn't, a small bias is introduced. Increasing the number of components p reduces bias but increases variance, while decreasing p reduces variance but increases bias.

Similarly to the shrinkage parameter in ridge and lasso estimation, the number of factors p can be treated as a tuning parameter. We can use m-fold cross validation to select p such that the MSE is minimized.

The `caret` package in R provides a convenient way to perform cross-validation and select the optimal number of principal components.

```
set.seed(111)
## PCR 10-fold cross-validation
library(caret)
```

```
Lade nötiges Paket: ggplot2
```

```
Lade nötiges Paket: lattice
```

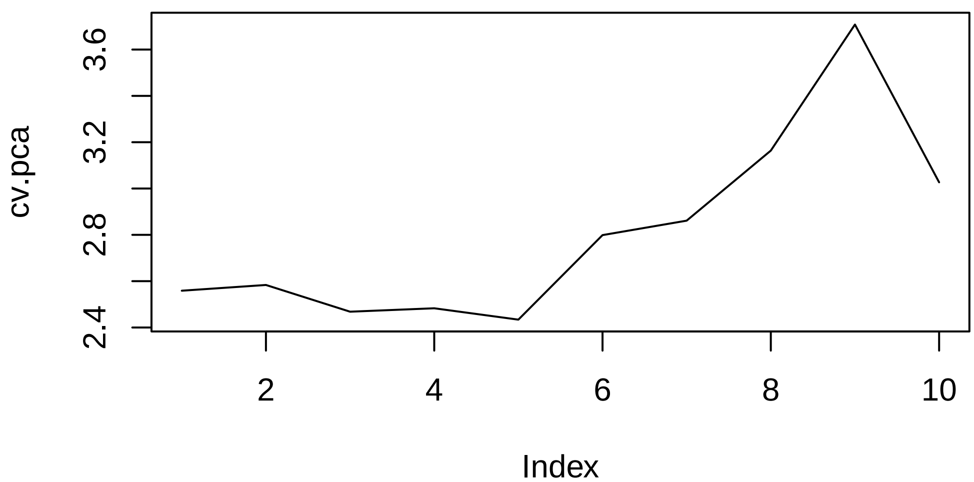
```
myfunc.cvpca = function(p){
  data_pca = data.frame(Y, P[,1:p])
  cv = train(
    Y ~ ., data = data_pca,
    method = "lm",
    metric = "RMSE",
    trControl = trainControl(method = "cv", number = 10)
```

```
  )
  return(cv$results$RMSE)
}
# Iterate function crossval over ncomp = 1, ..., maxcomp
maxcomp = 10 # select not more than number of variables (for data_small select <=4)
cv.pca = sapply(1:maxcomp, myfunc.cvpca) # sapply is useful for iterating over function argu

# Find the number of components with the lowest RMSPE
which.min(cv.pca)
```

```
[1] 5
```

```
plot(cv.pca, type="l")
```



The 10-fold cross validation method suggests to use 5 principal components.

11.8 R-codes

metrics-sec11.R