Lineare Modellwahl III

Partial Least Squares Regression

Büsra Karaoglan (754331)

Contents

Einleitung

Partial Least Squares oder PLS ist ein Sammelbegriff für rechnergestützte statistische Verfahren, die einen Kompromiss zwischen "Strukturenentdeckend" (zum Beispiel Clusteranalyse) und "Datenreduzierend" (zum Beispiel Faktorenanalyse) bilden.

Bei der Faktorenanalyse (Hauptkomponentenanalyse (engl. für Principal Component Analysis, PCA)) erfolgt eine Reduzierung der Dimension des durch die *unabhängigen Variablen* X aufgespannten Raumes in eine kleinere Anzahl latenter Variablen, die jedoch einen Großteil der in den Ausgangsdaten enthaltenen Varianz erklären sollten. Ziel der Faktoranalyse ist also, die Zahl der unabhängigen Variablen X zu verkleinern, ohne dass die Daten wesentlich an "Aussagekraft" verlieren. Diese "Optimierung" findet aber **ausschließlich im Raum der unabhängigen Variablen, X** statt.

Möchte man nun anschließend eine Regression dieses optimierten Raumes unabhängiger Variablen auf einen Satz *abhängiger Variablen* Y durchführen, so steht man vor dem Problem, dass unter Umständen einige der ursprünglichen unabhängigen Variablen X einen besseren Zusammenhang mit den abhängigen Variablen Y ergeben hätten als die neu erzeugten, anzahlmäßig weniger latenten Variablen. Man läuft also Gefahr, Zusammenhangsinformation zwischen X und Y zu verlieren, da man ungeachtet Y bereits im Raum X vollendete Tatsachen geschaffen hat.

Partial Least Squares Verfahren tragen diesem Sachverhalt dadurch Rechnung, dass in beiden Räumen X und Y "gleichzeitig optimiert" wird. Die Zerlegung der Variablenräume Y und X erfolgt unter der Nebenbedingung maximaler Kovarianzen zwischen den (neu erzeugten, latenten) unabhängigen Variablen X und den abhängigen Variablen Y. Die Datenreduktion, oder besser gesagt, die Reduktion der Dimensionalität findet in beiden Räumen, X und Y, gleichzeitig statt. Die Faktoren werden also unter Zuhilfenahme der Varianz-Kovarianzmatrix zwischen X und Y bestimmt.

Der wesentliche Unterschied zwischen der PLS-Regression und der PCR (Hauptkomponentenregression (engl. für Principal Component Regression)) liegt also darin, dass die PLS bei der Findung der PLS-Komponenten für die X-Daten bereits die Struktur der Y-Daten benutzt. Damit wird häufig erreicht, dass weniger PLS-Komponenten nötig werden und diese außerdem leichter zu interpretieren sind.

Es gibt zwei Ansätze der PLS-Regression. Der erste einfachere Ansatz ist der PCR ähnlich und bestimmt den Zusammenhang zwischen einer einzigen Zielgröße y (zum Beispiel die Oktanzahl) und vielen Messgrößen X (zum Beispiel Spektren). Dieser PLS-Ansatz wird PLS1 genannt. Es ist aber auch möglich, ein gemeinsames Modell für viele Zielgrößen Y (zum Beispiel Zusatzstoff 1, Zusatzstoff 2, usw.) und viele Messgrößen X zu errechnen. Man nennt diese PLS-Methode PLS2. Eigentlich ist die PLS1-Methode im PLS2-Ansatz als Sonderfall enthalten. In Abbildung soll die Idee und die beteiligten Matrizen für den allgemeinen Fall der PLS2 vorgestellt werden.

Allgemeines Vorgehen

Ausgangspunkt ist die Datenmatrix $\mathbf{X}$ der Dimension $(N \times M)$, mit N Objekten und M gemessenen Eigenschaften zum Beispiel den M Spektrenwerten. Zu jedem Objekt i wird eine Zielgröße y_i gemessen $(i = 1, \ldots, N)$, die den Vektor $\mathbf{y}$ bildet. Werden zu jedem Objekt mehrere y_{ij}-Werte gemessen, so ergeben die verschiedenen y_j-Vektoren die Matrix $\mathbf{Y}$ mit der Dimension $(N \times K)$, wobei K die Anzahl der y_j-Zielgrößen ist $(j = 1, \ldots, K)$. Die Idee der PLS ist es, sowohl mit den X-Daten eine PCA zu machen als auch mit den Y-Daten, wobei aber beide voneinander wissen. In Abbildung ist dieser Informationsaustausch zwischen der X- und der Y-Seite als Pfeil angedeutet, wobei die PCA der X-Daten Information aus den Y-Daten erhält und die PCA der Y-Daten von den X-Daten beeinflusst wird.

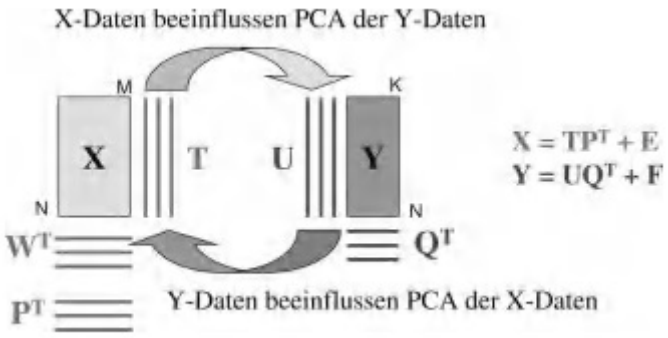


Figure 1: Schematische Darstellung der PLS und der beteiligten Matrizen

Auf die mathematische Herleitung der PLS wird hier nicht näher eingegangen und kann im Anhang nachgelesen werden. Hier folgt nun eine kurze Erklärung der in Abbildung **??** dargestellten Matrizen. Die Methode der PLS kombiniert Funktionen der Hauptkomponentenanalyse und der multiplen Regression. Hierfür wird eine Regression von vielen unabhängigen x-Variablen auf eine oder mehrere y-Variablen berechnet. Der Unterschied zur Multilinearen Regression ist der, dass die x-Variablen hoch korreliert und interkorreliert sein dürfen, dass es viel mehr x-Variable als Objekte geben darf und trotzdem die Regression gerechnet werden kann.
Auch bei der PLS Regression werden die $\mathbf{X}$-Daten in die Matrizen $\mathbf{T}$ und $\mathbf{P}$ zerlegt, wie bei der PCA. Allerdings wird bei der Zerlegung in die PLS-Komponenten für die $\mathbf{X}$-Daten die Zielgröße y schon mit einbezogen. Als Zwischenschritt ist hier bei der PLS die $\mathbf{W}$-Matrix nötig. In der $\mathbf{W}$-Matrix steckt die Verbindung zu den Y-Daten.
Man hat auf der einen Seite die Datenmatrix $\mathbf{X}$, die mit Hilfe der PCA in die beiden Matrizen $\mathbf{P}$ (Faktormatrix bzw. Loadingsmatrix) und $\mathbf{T}$ (Scoremetrix) plus einer Fehlermatrix $\mathbf{E}$ zerlegt wird: $\mathbf{X} = \mathbf{TP^T} + \mathbf{E}$
Auf der anderen Seite hat man die Zielgrößenmatrix $\mathbf{Y}$, die auch nur aus einem einzigen Vektor bestehen kann. Hat diese $\mathbf{Y}$-Matrix mehr als 1 Vektor, so macht man auch hier eine PCA und erhält die Faktormatrix $\mathbf{Q}$ mit der zugehörigen Scorematrix $\mathbf{U}$ und den Fehlerterm $\mathbf{F}$: $\mathbf{Y} = \mathbf{UQ^T} + \mathbf{F}$

Diese zwei Gleichungen nennt man die "äußeren Beziehungen". Das Ziel von PLS ist es, die Norm von **F** zu minimieren und gleichzeitig eine Korrelation zwischen **X** und **Y** zu erhalten, in dem die Matrizen **U** und **T** in Beziehung zueinander gesetzt werden: $\mathbf{U} = \mathbf{BT}$. Diese Gleichung nennt man auch die "innere Beziehung".

Predictive Ability

Wie bei der Hauptkomponenten-Regression wird auch bei PLS-Regression vorzugsweise beim Vorliegen vieler Regressoren eingesetzt, etwa um dem Multikollinearitätsproblem zu begegnen.
Natürlich stellt sich dann die Frage, wie viele Komponente bestimmt und für die Regressionsbeziehung berücksichtigen werden sollte. Eine praktikable Vorgehensweise basiert auf der *Kreuzvalidierung*. Dazu werden wiederholt Beobachtungen aus dem Datensatz entfernt; mit den restlichen wird die PLS-Regression durchgeführt. Für jede Anzahl von Komponenten wird jeweils der Wert eines Zielkriteriums bestimmt. Die Anzahl wird dann gewählt, bei der das zusammengefasste Kriterium minimal ist.
Bei vielen Verfahren der multivariaten Statistik lässt sich die Zahl der Freiheitsgrade nicht angeben, wodurch die Berechnung des Standardfehlers unmöglich wird. Um dennoch ein Maß für die Größe des Vorhersagefehlers zu bekommen, greift man auf das quadratische Mittel des Fehlers, RMSEP, zurück. **RMSEP** steht für **R**oot **M**ean **S**quared **E**rror of **P**rediction und **PRESS** steht für **PR**edictive **E**rror **S**um of **S**quares.
Für das PRESS-Kriterium gilt bei K Durchgängen, wobei im k-ten Durchgang m Beobachtungen mit den Indizes $i_{1,k}, \ldots, i_{m,k}$ ausgeschlossen sind, ist

$$PRESS = \sum_{k=1}^{K} \sum_{l=1}^{m} \left(y_{i_{l,k}} - \hat{y}_{i_{l,k}} \right)^2$$

RMSEP wird durch Summation aller quadrierten Vorhersagefehler während einer Kreuzvalidierung berechnet und ist ein Maß für die Güte eines Modells. Ein niedriger RMSEP-Wert deutet auf ein gutes Vorhersagemodell. RMSEP misst also die Vorhersagefähigkeit (englisch predictive ability) eines Modells.

$$RMSEP = \sqrt{\frac{PRESS}{n}}$$

PRESS bzw. RMSEP können dazu verwendet werden, die optimale Zahl an Variablen durch einen schrittweisen Variablenselektionsvorgang zu finden. Das "beste" Modell besteht aus möglichst wenigen unabhängigen Variablen und zeigt dabei den niedrigsten (oder nahezu niedrigsten) PRESS-Wert.

Anwendung

Kalibrierung von NIR-Spektren

Die Partial Least Square Regression hat in den letzten Jahren sehr stark an Bedeutung gewonnen und ist zum fast ausschließlich verwendeten Regressionsalgorithmus für die multivariate Regression in der Chemie geworden. Vor allem in der Spektroskopie wird die PLS zur Kalibrierung von Eigenschaften aus Spektren verwendet. Multivariat beschreibt dabei den Umstand, das zu einer gesuchten Konzentration nicht nur *ein* Messwert, sondern eine *Vielzahl* von Messwerten (im Fall eines Spektrums die Intensitäten I_l bei den Wellenzahl v_l) vorliegt. Da nicht jeder Messwert die gleiche Information trägt - manche Spektralbereiche tragen nur wenig oder keine Information über die gesuchte Substanz, andere sind korreliert und liefern redundante Werte-, dienen diese Verfahren vor allen Dingen zur Reduktion der großen Menge an Daten auf die für die Konzentrationsbestimmung notwendigen.

Um die Fähigkeit der PLS-Regression zu demonstrieren, soll ein spektroskopisches Beispiel gewählt werden, denn hier ist die Kollinearität zwischen den einzelnen Spektrenwerten besonders hoch und man misst wie oben erwähnt in der Regel viel mehr X-Variablen (Wellenlängen) als man Kalibrierproben zur Verfügung hat. Eine Kalibrierung mit der Multiple Lineare Regression (kurz MLR) wäre also nur möglich, wenn man sich auf einige wenige einzelne Wellenlängen einschränkt, was prinzipiell möglich wäre, aber natürlich gleich die Frage aufwirft, welche Wellenlängen man wählt. Die Lösung bietet die PLS. Es können das gesamte Spektrum verwenden werden und man erfährt anhand der Regressionskoeffizienten sozusagen als Zugabe, welche Wellenlängen für die Kalibrierung wichtig sind.

Es folgt also ein Beispiel aus der Spektroskopie und eine kurze Zusammenfassung der *gasoline* Daten. Es handelt sich um NIR-Spektren im Wellenlängenbereich 900 bis 1700 nm und Oktanzahl von 60 Benzinproben. Die NIR-Spektren wurden in 2-nm-Intervallen gemessen, was 401 Wellenlängen ergab. Später will man nur die Spektren messen und aus den Spektren den Oktanzahl bestimmen.
NIR: Als nahes Infrarot (NIR) wird der Bereich des elektromagnetischen Spektrums bezeichnet, der sich in Richtung größerer Wellenlänge an das sichtbare Licht anschließt.
Okatanzahl: Die Oktanzahl ist ein Maß für die Klopffestigkeit von Benzinen. Je höher die Oktanzahl, desto höher ist auch die Leistungsfähigkeit des Benzins und damit des Motors.
Die Messung der Spektren ist relativ einfach und kostengünstig im Gegensatz zu der traditionelle Labormethoden für die Bestimmung der Oktanzahl. Es wird also angestrebt, einen funktionalen Zusammenhang mit dem der Oktanzahl aus der gemessenen Spektren berechnet werden kann.
Als Datenbasis für die PLS-Regression sind zum einen die X-Werte (hier *NIR*) und zum anderen die Y-Werte, die in der Regel aufwendig zu bestimmende Referenzwerte (hier *octane*).
NIR: Das NIR-Spektrum ist eine Matrix mit 401 Spalten. *octane*: Die Oktanzahl ist ein numerischer Vektor.

```
library(pls)
data(gasoline)

gasTrain<- gasoline[1:50,]
gasTest<- gasoline[51:60,]
```

Die X-Variablen sind zum Teil stark korreliert, wie aus der Korrelationsmatrix zu erkennen ist.

```
#View(gasoline)
df<-cbind(gasoline$octane, as.matrix(gasoline$NIR))
View(df)
corm<-cor(df)
View(corm)
```

Die mit R durchgeführte PLS-Regression führt nun zu dem nachstehenden Ergebnis, wenn erst einmal keine Beschränkung bei den Faktoren formuliert wird.

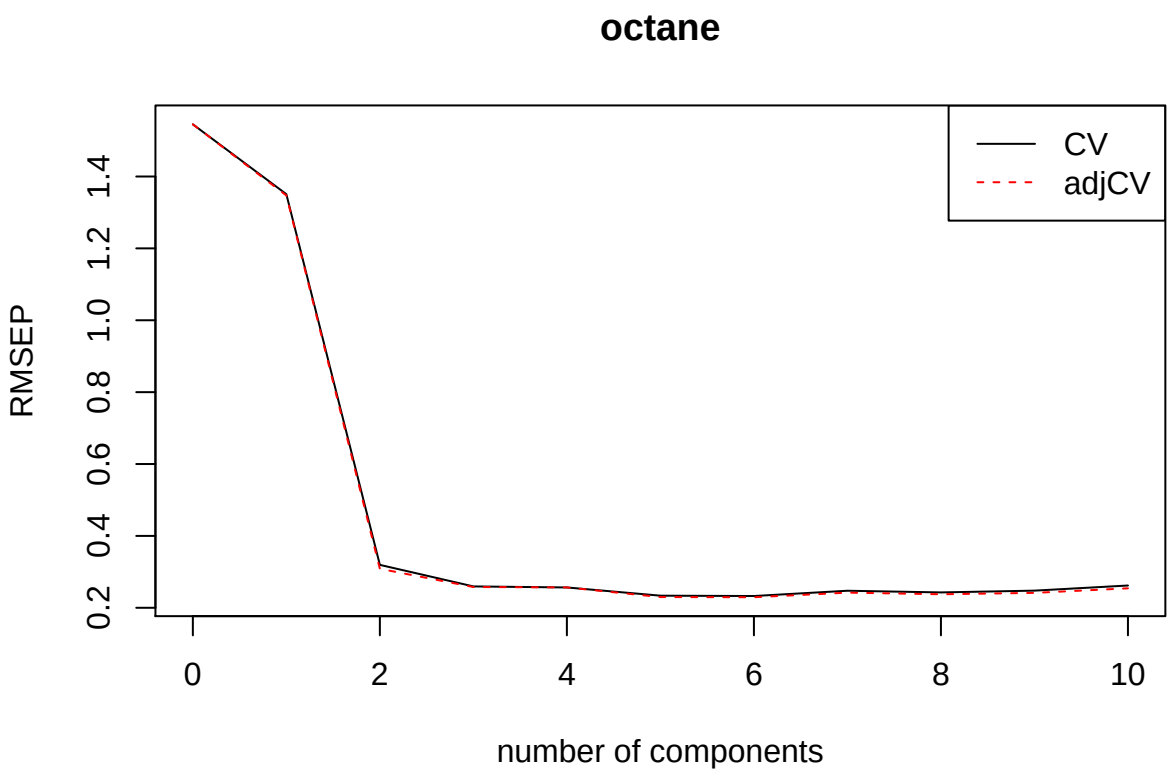
```
gas1<- plsr(octane ~ NIR, ncomp = 10, data = gasTrain, validation = "CV")
summary(gas1)
```

```
## Data:    X dimension: 50 401
##  Y dimension: 50 1
## Fit method: kernelpls
## Number of components considered: 10
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           1.545    1.351   0.3194   0.2594   0.2566   0.2337   0.2328
## adjCV        1.545    1.347   0.3085   0.2578   0.2567   0.2298   0.2294
##        7 comps  8 comps  9 comps  10 comps
## CV      0.2474   0.2428   0.2477    0.2620
## adjCV   0.2422   0.2377   0.2415    0.2542
##
## TRAINING: % variance explained
##         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X         78.17    85.58    93.41    96.06    96.94    97.89    98.38
## octane    29.39    96.85    97.89    98.26    98.86    98.96    99.09
##         8 comps  9 comps  10 comps
## X         98.85    99.02     99.19
## octane    99.16    99.28     99.39
```

Die vorhergesagten Oktanzahlen werden mit zwei PLS-Komponenten bereits zu 96,85% erklärt, weitere PLS-Komponenten verbessern diesen Wert nur unbedeutend. Hier würde man also bei Zugrundelegung des Anteils der erklärten Varianz wohl zwei PLS-Komponenten verwenden, um die abhängige Variable (*octane*) zu erklären.

Zudem ist in diesem Beispiel der RMSEP=0,2996, wenn zwei PLS-Komponenten verwendet werden und die Kreuzvalidierung angewandt wird. Damit ist die optimale Anzahl an PLS-Komponenten gefunden. Um die RMSEPs viel einfacher zu beurteilen werden diese zusätzlich grafisch dargestellt.

```
plot(RMSEP(gas1), legendpos = "topright")
```

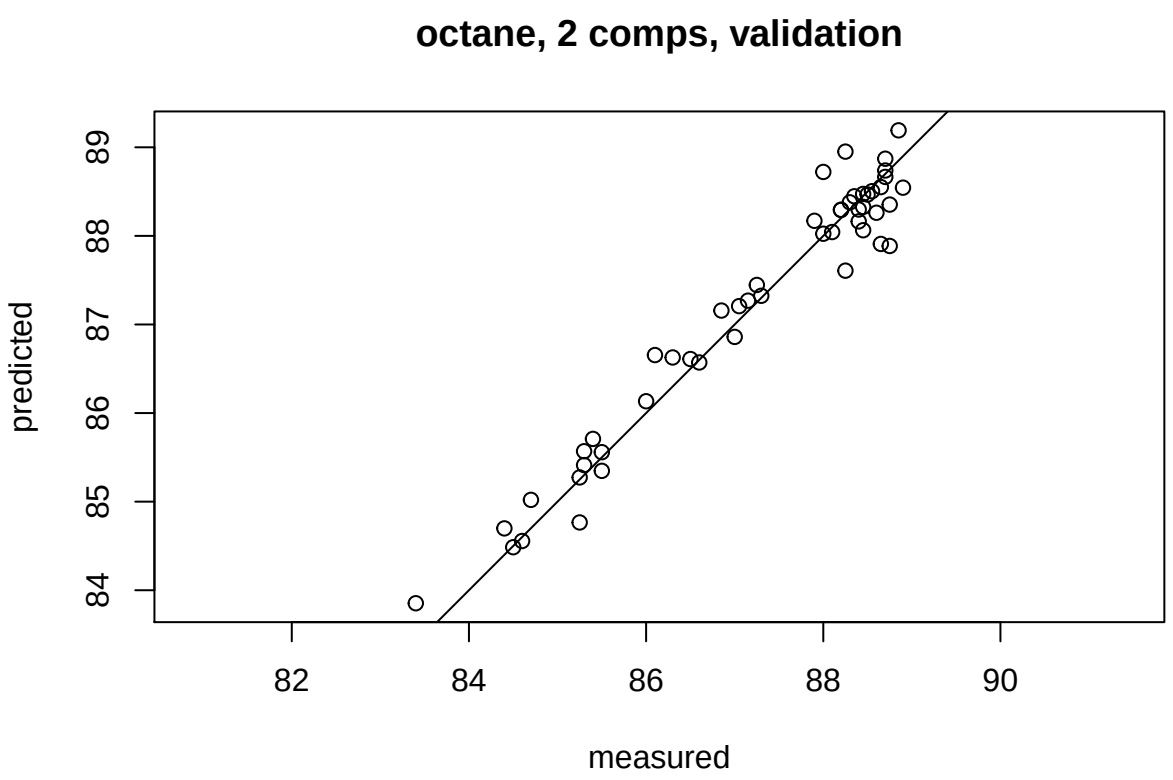


Eine wichtige Möglichkeit die Kalibrationsgüte oder Vorhersagegüte zu überprüfen, ist die grafische Darstellung der vorhergesagten Werte im Vergleich zu den gemessenen Werten. Dazu werden die aus der Kalibriergleichung berechneten $\hat{y}$-Werte gegen die Referenzwerte y aufgetragen. Der Referenzwert wird üblicherweise auf der x-Achse aufgetragen, während der aus der Regressionsgleichung berechnete Y-Wert „Predicted Y“ genannt wird und auf der y-Achse aufgetragen wird.

Anhand des Diagramms kann man Besonderheiten in den Kalibrierdaten erkennen. Werte mit großem Abstand von der Geraden werden schlecht durch die Kalibriergleichung beschrieben. Man erkennt, ob die Vorhersagegenauigkeit für kleine und große Y-Werte gleich gut ist und auch Abweichungen von der Linearität lassen sich an dieser Grafik bereits erkennen.

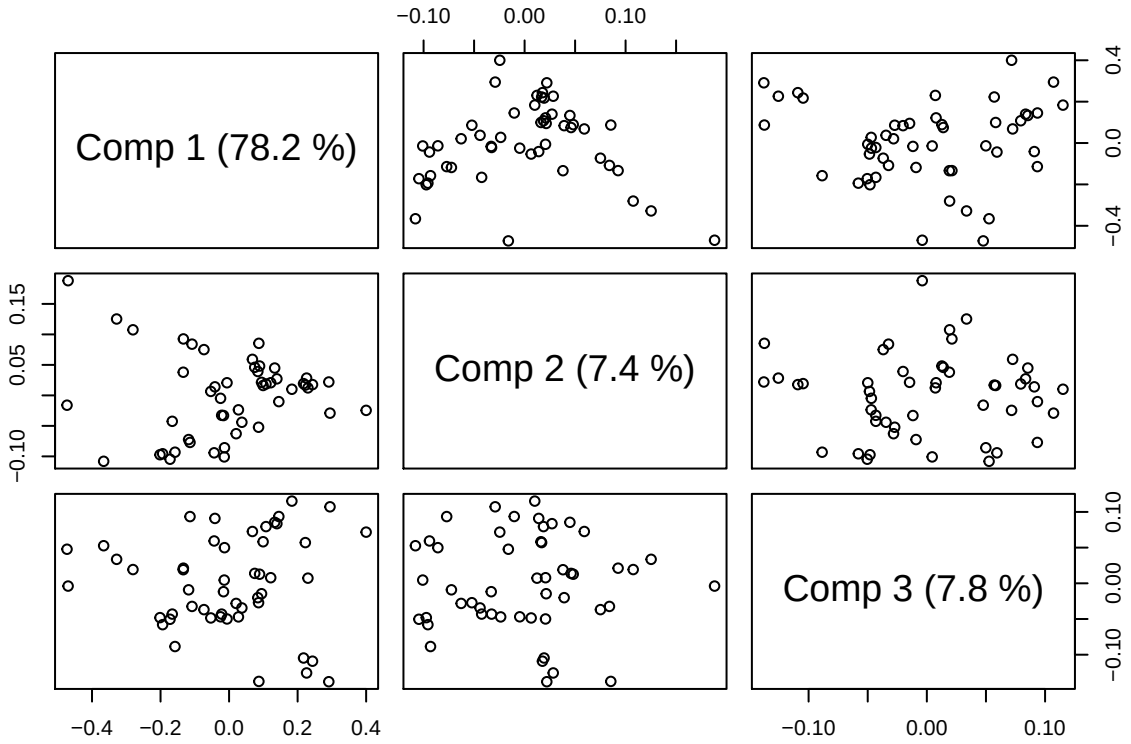
Das untere Bild zeigt nun die mit dem Regressionsmodell vorhergesagten Werte im Vergleich zu den gemessenen Werten berechnet mit zwei PLS-Komponenten.

```
plot(gas1, ncomp = 2, asp = 1, line = TRUE)
```



Der Scoreplot liefert Information über die Objekte bezogen auf die Hauptkomponenten.

```
plot(gas1, plottype = "scores", comps = 1:3)
```



Die erste Hauptkomponente erklärt 78,2% der Varianz in den spektralen Daten, die zweite Hauptkomponente trägt mit weiteren 7,4% bei. Da hier nur PLS1 Modell verwendet wurde, erfolgte demnach auch nur in X-Daten ein PCR. Daher beziehen sich hier die Erklärungsanteile auf X-Daten und konnten schon oben in der Ausgabe für $summary(gas1)$ in der X-Zeile abgelesen werden.

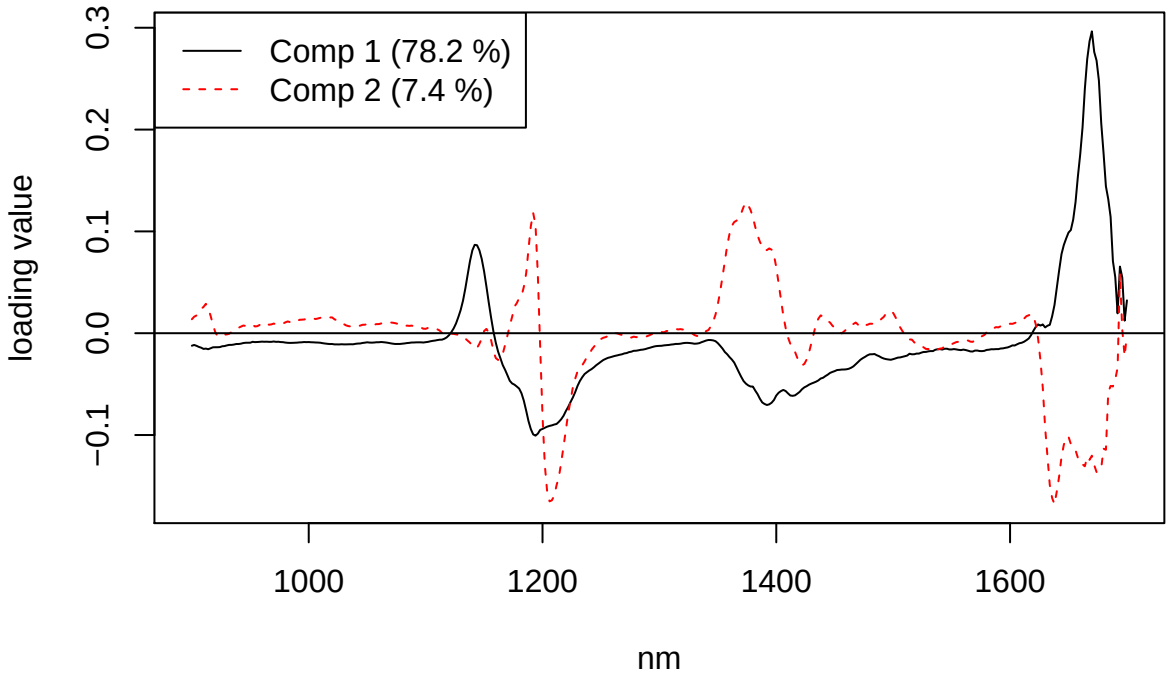
Zusätzlich können diese Erklärungsanteile mit folgendem Befehl angezeigt werden.

```
explvar(gas1)
```

```
##     Comp 1     Comp 2     Comp 3     Comp 4     Comp 5     Comp 6
## 78.1707683  7.4122245  7.8241556  2.6577773  0.8768214  0.9466384
##     Comp 7     Comp 8     Comp 9    Comp 10
##  0.4921537  0.4723207  0.1688272  0.1693770
```

Der Loadingsplot zeigt die Zusammenhänge der einzelnen Variablen zu den Hauptkomponenten. Er wird entweder als zweidimensionaler Plot der Loadings von Comp 1 gegen die von Comp 2 dargestellt oder im Fall von spektralen Daten als Linienplot.

```
plot(gas1, "loadings", comps = 1:2, legendpos = "topleft", labels = "numbers", xlab = "nm")
abline(h = 0)
```




```
predict(gas1, ncomp = 2, newdata = gasTest)
```

```
## , , 2 comps
##
##       octane
## 51 87.94125
## 52 87.25242
## 53 88.15832
## 54 84.96913
## 55 85.15396
## 56 84.51415
## 57 87.56190
## 58 86.84622
## 59 89.18925
## 60 87.09116
```

```
RMSEP(gas1, newdata = gasTest)
```

```
## (Intercept)      1 comps      2 comps      3 comps      4 comps
##      1.5369       1.1696       0.2445       0.2341       0.3287
##     5 comps      6 comps      7 comps      8 comps      9 comps
##      0.2780       0.2703       0.3301       0.3571       0.4090
##    10 comps
##      0.6116
```

```
sqrt(sum((predict(gas1, ncomp = 2, newdata = gasTest)-gasTest[,1])^2)/nrow(gasTest))
```

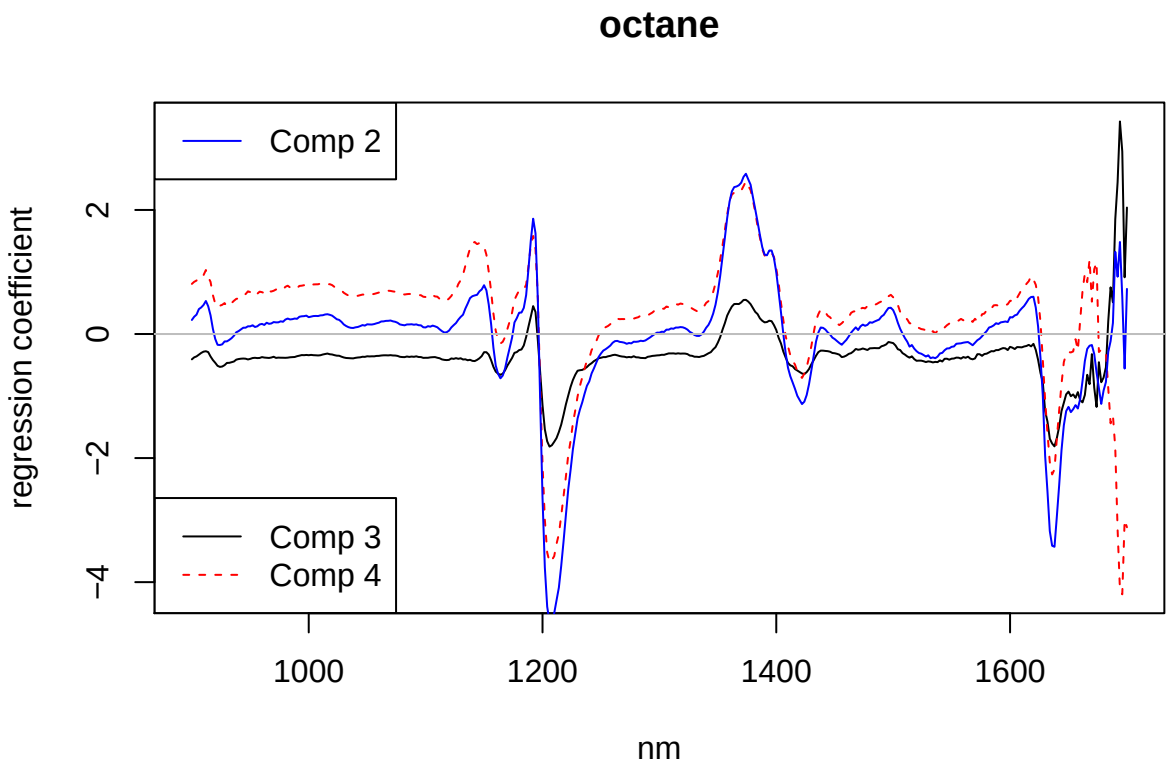
```
## [1] 0.2444825
```

Es folgt nun abschließend ein Vergleich der Validierungsergebnisse des PCR- und PLS-Modells zur Vorhersage der Oktanzahl aus NIR-Spektren. Hierfür werden die Regressionskoeffizienten der PLS und PCR betrachtet. Die folgende Abbildung zeigt die Regressionskoeffizienten für alle Wellenlängen unter Berücksichtigung von zwei PLS-Komponenten und drei bzw. vier PCR-Komponenten.

```
gas.pcr <- pcr(octane ~ NIR, ncomp = 10, data = gasoline, validation = "CV")
gas.pls <- plsr(octane ~ NIR, ncomp = 10, data = gasoline, validation = "CV")


plot(gas.pcr, "coefficients", comps = 3:4, legendpos = "bottomleft",
     labels = "numbers", xlab = "nm")

plot(gas.pls, "coefficients", comps = 2, add = TRUE, col = "blue",
     legendpos = "topleft", labels = "numbers", xlab = "nm")
```



Die Regressionskoeffizienten werden als Linienplot dargestellt, um den Zusammenhang mit den Wellenlängen zu verdeutlichen. Der erste Regressionskoeffizient b1 gehört zu 900 nm (erste gemessene Wellenlänge im Spektrum). Der zweite dann zu 902 nm usw., da alle 2 nm ein Messwert aufgenommen wurde. Der letzte Regressionskoeffizient gehört damit zu der letzten gemessenen Wellenlänge im Spektrum (bei 1700 nm). Die Regressionskoeffizienten für zwei PLS- und vier PCR-Komponenten unterscheiden sich nur sehr wenig und vorwiegend auf den Wellenlängen, die sowieso nicht viel beitragen, also betragsmäßig kleine Werte haben (zum Beispiel zwischen 900 und 1100 nm). In zwei PLS-Komponenten ist also die gleiche Information enthalten wie in vier PCR-Komponenten.

Nun gibt es zwischen 1150 und 1400 und zusätzlich zwischen 1600 und 1700 Wellenzahlen Bereiche der Regressionskoeffizienten, in dem starke Ausschläge mit wenig kleineren überlagerten Ausschlägen vorkommen. Wir wissen, dass betragsmäßig große Regressionskoeffizienten einen großen Beitrag zur Zielgröße leisten, also ist dieser Bereich für die Vorhersage der Oktanzahl besonders wichtig. Um die Vorhersage robuster zu machen, kann man zudem den Wellenzahlbereich beschränken und erneut eine PLS-Regression rechnen.

Hitters-Daten

Die Methode der Partial Least Square kann wie schon erläutert für Variablen verwendet werden, die stark korrelieren. Da die PLS-Methode als Kombination der Hauptkomponentenanalyse (PCA) und der Multiplen Regression betrachtet wird, können die für die PCA verwendeten Daten auch mit der PLS-Methode bearbeitet werden.

```
library(ISLR)
data(Hitters, package = "ISLR")
Hitters <- na.omit(Hitters)

x <- model.matrix(Salary~.,Hitters)[,-1]
y <- Hitters$Salary

set.seed(1)
train <- sample(1:nrow(x), nrow(x)/2)
test <- (-train)
y.test <- y[test]
set.seed(1)
```

Zunächst wird die Korrelationsmatrix ausgegeben.

```
# Multi-collinearity

    Hitters_num<-Hitters[, c(-14,-15,-19, -20)]
    cor_table<-round(cor(Hitters_num, use="complete.obs", method="pearson")*100,1)
    cor_table
```

```
##         AtBat  Hits HmRun  Runs   RBI Walks Years CAtBat CHits CHmRun
## AtBat   100.0  96.4  55.5  90.0  79.6  62.4   1.3   20.7  22.5   21.2
## Hits     96.4 100.0  53.1  91.1  78.8  58.7   1.9   20.7  23.6   18.9
## HmRun    55.5  53.1 100.0  63.1  84.9  44.0  11.3   21.7  21.7   49.3
## Runs     90.0  91.1  63.1 100.0  77.9  69.7  -1.2   17.2  19.1   23.0
## RBI      79.6  78.8  84.9  77.9 100.0  57.0  13.0   27.8  29.2   44.2
## Walks    62.4  58.7  44.0  69.7  57.0 100.0  13.5   26.9  27.1   35.0
## Years     1.3   1.9  11.3  -1.2  13.0  13.5 100.0   91.6  89.8   72.2
## CAtBat   20.7  20.7  21.7  17.2  27.8  26.9  91.6  100.0  99.5   80.2
## CHits    22.5  23.6  21.7  19.1  29.2  27.1  89.8   99.5 100.0   78.7
## CHmRun   21.2  18.9  49.3  23.0  44.2  35.0  72.2   80.2  78.7  100.0
## CRuns    23.7  23.9  25.8  23.8  30.7  33.3  87.7   98.3  98.5   82.6
## CRBI     22.1  21.9  35.0  20.2  38.8  31.3  86.4   95.1  94.7   92.8
## CWalks   13.3  12.3  22.7  16.4  23.4  42.9  83.8   90.7  89.1   81.1
## PutOuts  31.0  30.0  25.1  27.1  31.2  28.1  -2.0    5.3   6.7    9.4
## Assists  34.2  30.4 -16.2  17.9   6.3  10.3  -8.5   -0.8  -1.3  -18.9
## Errors   32.6  28.0  -1.0  19.3  15.0   8.2 -15.7   -7.0  -6.8  -16.5
##         CRuns  CRBI CWalks PutOuts Assists Errors
## AtBat    23.7  22.1   13.3    31.0    34.2   32.6
## Hits     23.9  21.9   12.3    30.0    30.4   28.0
## HmRun    25.8  35.0   22.7    25.1   -16.2   -1.0
```

```
## Runs     23.8  20.2   16.4    27.1    17.9   19.3
## RBI      30.7  38.8   23.4    31.2     6.3   15.0
## Walks    33.3  31.3   42.9    28.1    10.3    8.2
## Years    87.7  86.4   83.8    -2.0    -8.5  -15.7
## CAtBat   98.3  95.1   90.7     5.3    -0.8   -7.0
## CHits    98.5  94.7   89.1     6.7    -1.3   -6.8
## CHmRun   82.6  92.8   81.1     9.4   -18.9  -16.5
## CRuns   100.0  94.6   92.8     5.9    -3.9   -9.4
## CRBI     94.6 100.0   88.9     9.5    -9.7  -11.5
## CWalks   92.8  88.9  100.0     5.8    -6.6  -13.0
## PutOuts   5.9   9.5    5.8   100.0    -4.3    7.5
## Assists  -3.9  -9.7   -6.6    -4.3   100.0   70.4
## Errors   -9.4 -11.5  -13.0     7.5    70.4  100.0
```

Mit Hilfe von *R* werden nun Haupkomponenten- und PLS-Regression durchgeführt.

```
pcr.fit <- pcr(Salary~., data=Hitters , subset=train, scale=TRUE, validation="CV")
summary(pcr.fit)
```

```
## Data:    X dimension: 131 19
##  Y dimension: 131 1
## Fit method: svdpc
## Number of components considered: 19
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           464.6    406.1    397.1    399.1    398.6    395.2    386.9
## adjCV        464.6    405.2    396.3    398.1    397.4    394.5    384.5
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       384.8    386.5    394.1     406.1     406.5     412.3     407.7
## adjCV    383.3    384.8    392.0     403.4     403.7     409.3     404.6
##        14 comps  15 comps  16 comps  17 comps  18 comps  19 comps
## CV        406.2     417.8     417.6     413.0     407.0     410.2
## adjCV     402.8     413.9     413.5     408.3     402.4     405.5
##
## TRAINING: % variance explained
##         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X         38.89    60.25    70.85    79.06    84.01    88.51    92.61
## Salary    28.44    31.33    32.53    33.69    36.64    40.28    40.41
##         8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X         95.20    96.78     97.63     98.27     98.89     99.27     99.56
## Salary    41.07    41.25     41.27     41.41     41.44     43.20     44.24
##         15 comps  16 comps  17 comps  18 comps  19 comps
## X          99.78     99.91     99.97    100.00    100.00
## Salary     44.30     45.50     49.66     51.13     51.18
```

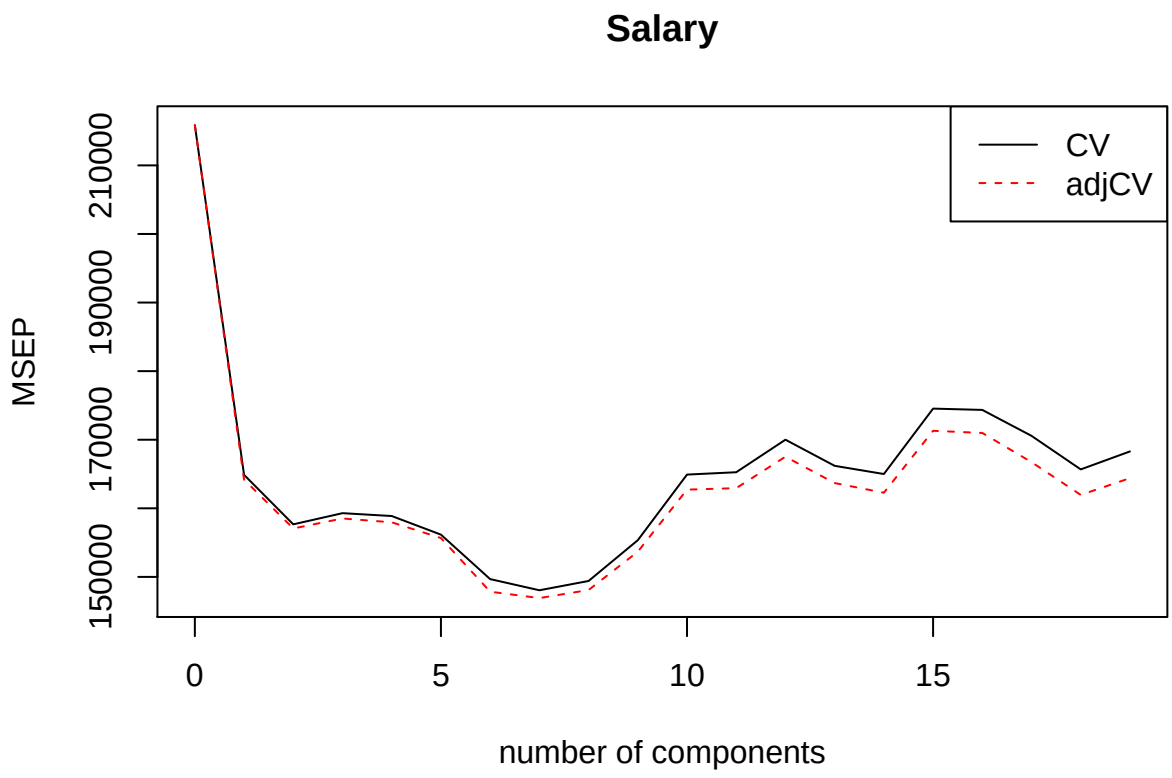
```
pls.fit <- plsr(Salary~., data=Hitters, subset=train, scale=TRUE, validation="CV")
summary(pls.fit)
```

```
## Data:    X dimension: 131 19
##  Y dimension: 131 1
## Fit method: kernelpls
## Number of components considered: 19
##
## VALIDATION: RMSEP
```

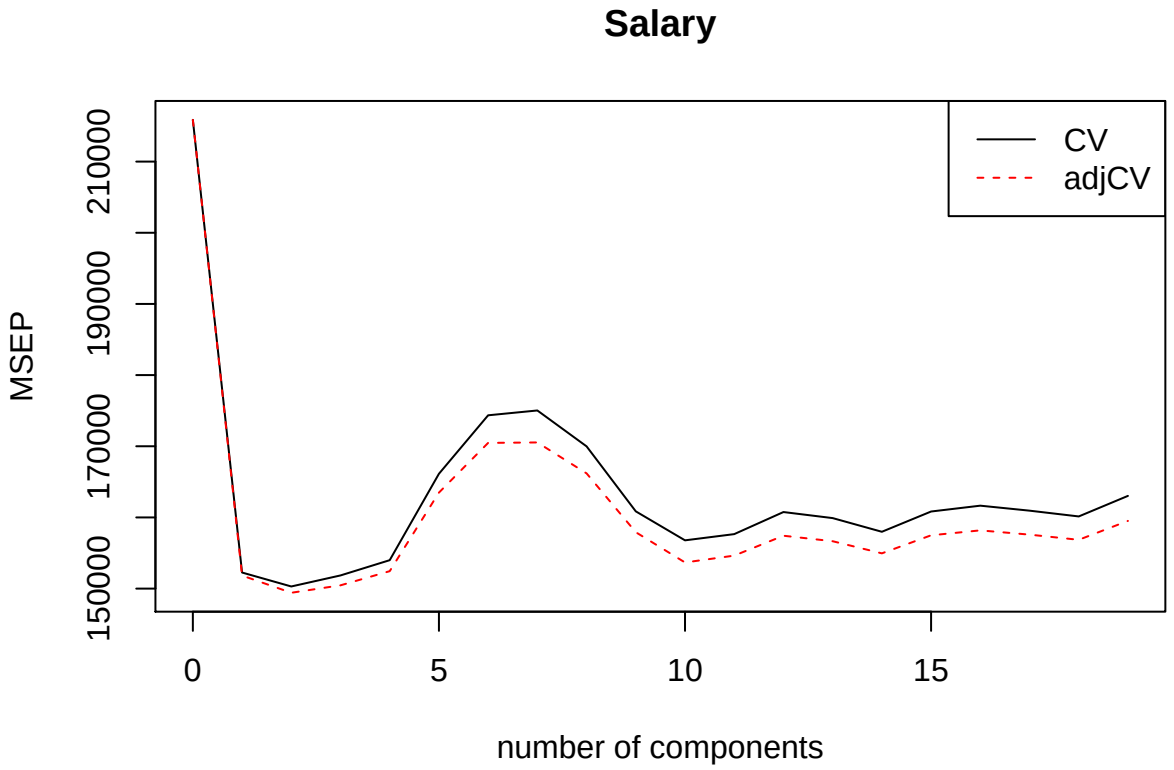
```
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           464.6    390.2    387.7    389.7    392.4    407.6    417.6
## adjCV        464.6    389.7    386.5    387.9    390.4    404.3    412.9
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       418.4    412.3    401.1       396     397.1     400.9     399.9
## adjCV    413.0    407.7    397.4       392     393.2     396.8     395.8
##        14 comps  15 comps  16 comps  17 comps  18 comps  19 comps
## CV        397.5     401.0     402.1     401.2     400.2     403.8
## adjCV     393.6     396.8     397.7     397.0     396.1     399.4
##
## TRAINING: % variance explained
##         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X         38.12    53.46    66.05    74.49    79.33    84.56    87.09
## Salary    33.58    38.96    41.57    42.43    44.04    45.59    47.05
##         8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X         90.74    92.55     93.94     97.23     97.88     98.35     98.85
## Salary    47.53    48.42     49.68     50.04     50.54     50.78     50.92
##         15 comps  16 comps  17 comps  18 comps  19 comps
## X          99.11     99.43     99.78     99.99    100.00
## Salary     51.04     51.11     51.15     51.16     51.18
```

Man kann anstelle des RMSE-Kriteriums das MSE-Kriterium betrachten.

```
validationplot(pcr.fit, val.type = "MSEP", legendpos="topright")
```



```
validationplot(pls.fit, val.type = "MSEP", legendpos="topright")
```



Hier werden zwei PLS-Komponenten berücksichtigt. Ein Vergleich mit den Ergebnissen der Hauptkomponentenregression (hier sieben PCA-Komponenten) lässt erkennen, dass tatsächlich mit viel weniger PLS-Komponenten ein genau so großer Teil der Varianz der abhängigen Variablen erklärt wird, wie es bei Hauptkomponentenregression der Fall ist.

```
pcr.pred <- predict(pcr.fit,x[test,],ncomp=7)
mean((pcr.pred-y.test)^2)
```

```
## [1] 96556.22
```

```
pls.pred <- predict(pls.fit, x[test,], ncomp=2)
mean((pls.pred-y.test)^2)
```

```
## [1] 101417.5
```

```
pcr.fit <- pcr(Salary~., data=Hitters, scale=TRUE, ncomp=7)
summary(pcr.fit)
```

```
## Data:    X dimension: 263 19
##  Y dimension: 263 1
## Fit method: svdpc
## Number of components considered: 7
## TRAINING: % variance explained
##         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X         38.31    60.16    70.84    79.03    84.29    88.63    92.26
## Salary    40.63    41.58    42.17    43.22    44.90    46.48    46.69
```

```r
pls.fit <- plsr(Salary~., data=Hitters, scale=TRUE, ncomp=2)
summary(pls.fit)
```

```
## Data:    X dimension: 263 19
##  Y dimension: 263 1
## Fit method: kernelpls
## Number of components considered: 2
## TRAINING: % variance explained
##         1 comps  2 comps
## X         38.08    51.03
## Salary    43.05    46.40
```

Zuletzt sollte man noch beachten, dass die PLS-Regression mit zwei PLS-Komponenten 46,40% und der PCA-Regression 46,69% der Varianz von *Salary* erklärt. Dies geschieht nur, weil die PCA in X-Daten und PLS in X- und Y-Daten optimiert. Desweiteren hat die PCA-Regression ein MSE-Wert von 96556 und die PLS-Regression von 101417.