# Arbeitsblatt 1

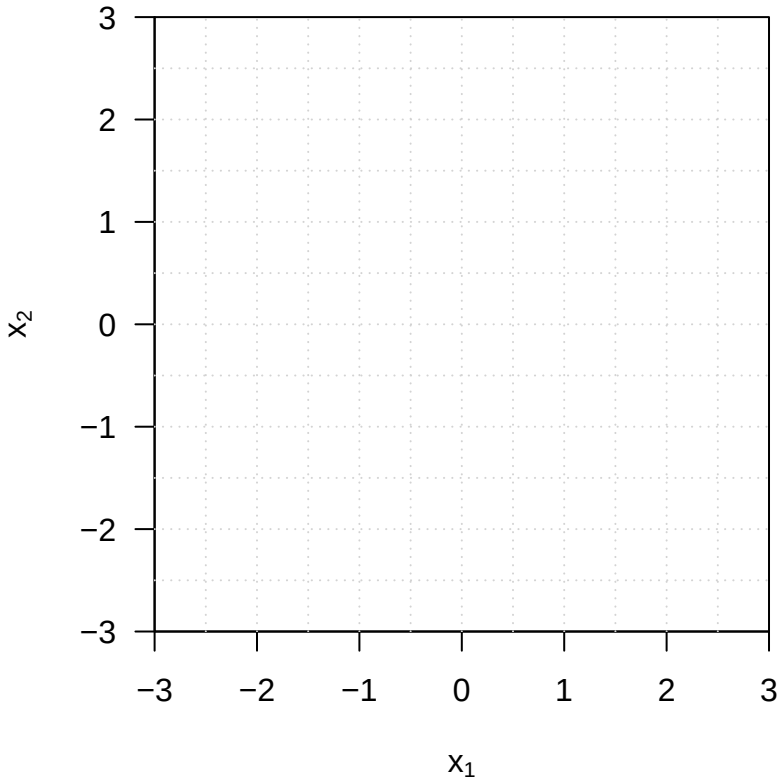
## Aufgabe 1: PCA von Hand

Wir wollen eine PCA mit Lineal und Papier durchführen. Wir haben die Koordinaten von 5 Beobachtungen:

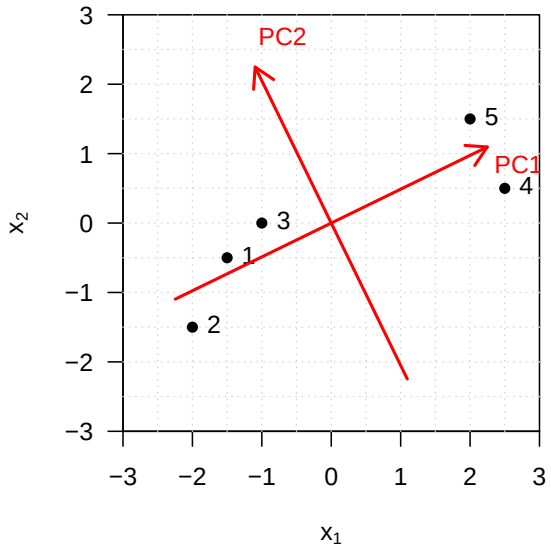
Tabelle 1: Datensatz

| | X_1 | X_2 |
|---|---|---|
| 1 | -1.5 | -0.5 |
| 2 | -2.0 | -1.5 |
| 3 | -1.0 | 0.0 |
| 4 | 2.5 | 0.5 |
| 5 | 2.0 | 1.5 |

a) Zeichnen Sie die 5 Beobachtungen in das vorgedruckte Koordinatensystem (benutzen Sie als Punktbezeichnung die Nummer der Beobachtung). Wenn Ihnen das manuelle Zeichnen zu mühsam ist, können Sie alternativ die Aufgabe auch direkt mit dem shinyApp_PCA_Rotation.R bearbeiten.




b) Zeichnen Sie in das Diagramm ein gedrehtes Koordinatensystem ein, so dass die erste Achse (PC1) die grösste Varianz hat. Die zweite Achse (PC2) steht senkrecht dazu.




c) Welche Koordinaten (Scores) haben die Punkte im neuen Koordinatensystem (grobe Abschätzung genügt).

```
   PC1  PC2
1 -1.6  0.2
2 -2.5 -0.5
3 -0.9  0.4
4  2.5 -0.6
5  2.5  0.5
```

d) Vergleichen Sie Ihre Ergebnisse mit dem R output.

```
         PC1        PC2
1 -1.5673743  0.2081770
2 -2.4551500 -0.4714216
3 -0.8987892  0.4383811
4  2.4661635 -0.6465581
5  2.4551500  0.4714216
```

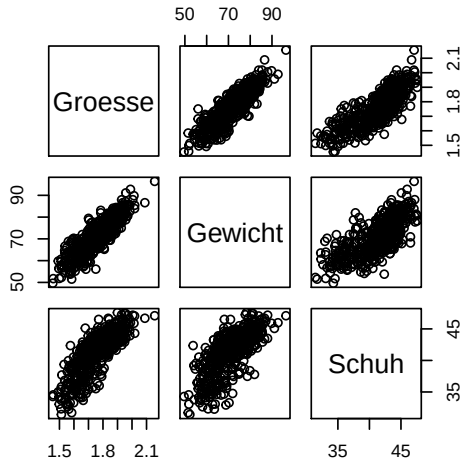
Wir erhalten einen sehr ähnlichen Score-Plot, wenn wir die PCA "von Hand" durchführen, wie wenn wir die PCA mittels R durchführen. }

## Aufgabe 2: Körpergrössen

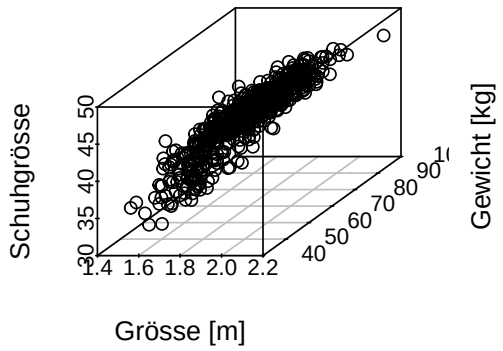
Laden Sie den R-Datensatz *body*. Er enthält die Körpergrösse, das Gewicht und die Schuhgrösse von 600 Personen (simulierte Daten).

a) Visualisieren Sie die Daten. Was sieht man?

```
load(file="Daten/body.Rdata")
# Scatterplotmatrix
pairs(body)
```




```
library(scatterplot3d)
scatterplot3d(x=body$Groesse, y= body$Gewicht, z=body$Schuh, xlab="Grösse [m]",
              ylab="Gewicht [kg]", zlab="Schuhgrösse")
```




Man sieht, dass die Variablen stark korreliert sind.

b) Berechnen Sie die Kovarianz-Matrix der Daten (R-Befehl `cov()`). Berechnen Sie die Summe der Varianzen der 3 Variablen (d.h. Diagonale der Kovarianz-Matrix).

```
cov(body)
```

```
           Groesse    Gewicht      Schuh
Groesse 0.01284146  0.7673536  0.2863454
Gewicht 0.76735357 58.8527013 17.3186532
Schuh   0.28634536 17.3186532  9.7924606
```

```
sum(diag(cov(body)))
```

```
[1] 68.658
```

```
# oder
var(body$Groesse)+var(body$Gewicht)+var(body$Schuh)
```

```
[1] 68.658
```

c) Führen Sie mit der Funktion `prcomp` die Hauptkomponentenanalyse durch. Bei der Hauptkomponentenanalyse arbeiten wir mit zentrierten Daten. Sie können das manuell machen oder bei der Funktion `prcomp` das Argument `center` auf `TRUE` setzen (Defaulteinstellung).

```
pca <- prcomp(body)
pca
```

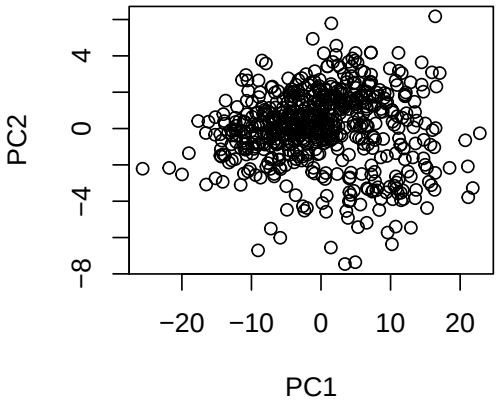
```
Standard deviations (1, .., p=3):
[1] 8.0225106 2.0725037 0.0453369

Rotation (n x k) = (3 x 3):
                PC1          PC2         PC3
Groesse -0.01271152  0.009491687  0.99987415
Gewicht -0.95305114 -0.302668278 -0.00924306
Schuh   -0.30254246  0.953048698 -0.01289344
```

```
pca <- prcomp(body, center=TRUE)
pca
```

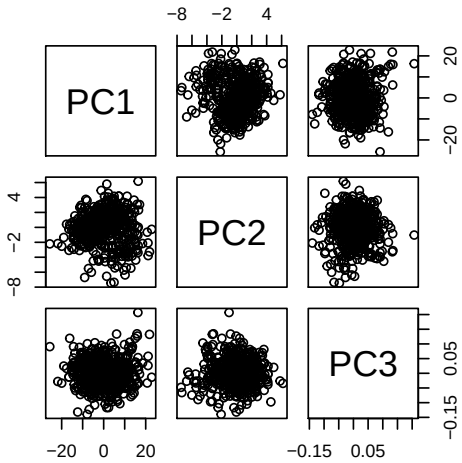
```
Standard deviations (1, .., p=3):
[1] 8.0225106 2.0725037 0.0453369

Rotation (n x k) = (3 x 3):
                PC1          PC2         PC3
Groesse -0.01271152  0.009491687  0.99987415
Gewicht -0.95305114 -0.302668278 -0.00924306
Schuh   -0.30254246  0.953048698 -0.01289344
```

d) Visualisieren Sie die Hauptkomponenten. Auf die Hauptkomponenten können Sie mit dem PCA-Objekt `pca$x` zugreifen.

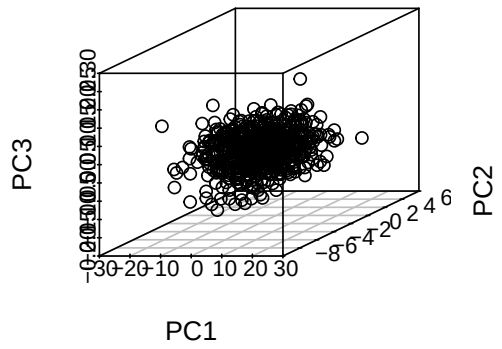
```
plot(PC2~PC1, data=pca$x, xlab="PC1", ylab="PC2")
```



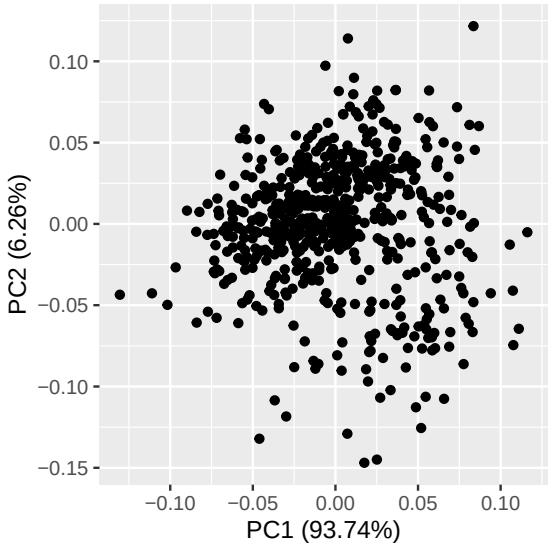

```
pairs(pca$x)
```




```
scatterplot3d(x=pca$x[,"PC1"], y= pca$x[,"PC2"], z=pca$x[,"PC3"],
              xlab="PC1", ylab="PC2", zlab="PC3")
```



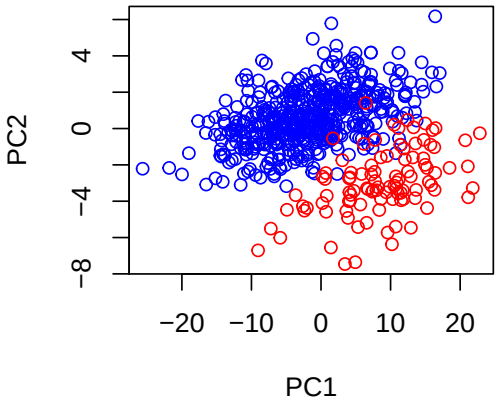

```
library(ggfortify)
autoplot(pca)
```



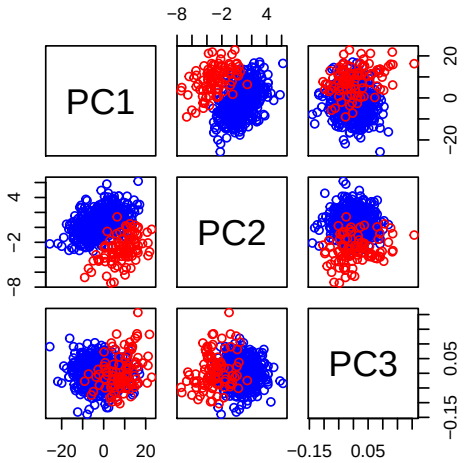

Wenn man die Einheiten der Abbildungen betrachtet, sieht man, dass die Streuung von der ersten zur dritten Hauptkomponente immer kleiner wird. Beim autoplot wird der Anteil der Streuung der einzelnen Hauptkomponenten direkt in der Achsenbeschriftung angegeben.

e) Sie erfahren, dass die letzten 100 Beobachtungen von Frauen stammen, während der Rest der Daten zu jungen Männern gehört. Visualisieren Sie diese Beobachtungen in den Hauptkomponenten mit einer anderen Farbe.

```
plot(PC2~PC1, data=pca$x, xlab="PC1", ylab="PC2",
     col=c(rep("blue", 500), rep("red",100)))
```




```
pairs(pca$x, col=c(rep("blue", 500), rep("red",100)))
```




Insbesondere in den ersten beiden Hauptkomponenten sieht man, dass die Gruppe der Frauen sich wirklich etwas von den Männern unterscheiden.

f) Berechnen Sie die Kovarianzmatrix der Hauptkomponenten. Was sehen Sie? Berechnen Sie zusätzlich die Summe der Varianzen und vergleiche Sie das Ergebnis mit b). Wie gross ist der Anteil der ersten beiden Komponenten an der Gesamtvarianz (d.h. wie gross ist der Anteil an der Streuung, der durch diese beiden Komponenten beschrieben wird).

```
round(cov(pca$x),3)
```

```
       PC1   PC2   PC3
PC1 64.361 0.000 0.000
PC2  0.000 4.295 0.000
PC3  0.000 0.000 0.002
```

```
# Summe der Varianz
sum(diag(cov(pca$x)))
```

```
[1] 68.658
```

```
# Anteil der ersten beiden PCs
sum(diag(cov(pca$x))[1:2])/sum(diag(cov(pca$x)))
```

```
[1] 0.9999701
```

```
# oder direkt
summary(pca)
```

```
Importance of components:
                          PC1     PC2     PC3
Standard deviation     8.0225 2.07250 0.04534
Proportion of Variance 0.9374 0.06256 0.00003
Cumulative Proportion  0.9374 0.99997 1.00000
```

Alle nicht Diagonalelemente der Kovarianzmatrix sind 0. Die Varianz der ersten Hauptkomponente ist klar am grössten, gefolgt von der Zweiten. Die Summe der Varianzen ist identisch zur Varianz der ursprünglichen Daten.
Die ersten beiden Hauptkomponenten beschreiben praktisch 100% der Variation. Bereits die erste Komponente beschreibt ca. 94% der Streuung. Das macht auch Sinn, da die Daten ja hoch korreliert sind.

g) Die ursprünglichen Variablen haben unterschiedliche Einheiten mit unterschiedlichen Grössenordnungen. In solchen Fällen werden die Variablen für die Hauptkomponentenanalyse in der Regel zuerst mit der Standardabweichung **standardisiert**. Standardisieren Sie die ursprünglichen Daten so, dass die transformierten Daten den Mittelwert 0 und eine Standardabweichung von 1 haben (Z-Transformation $Z = \frac{X-\mu}{\sigma}$). D.h. Mittelwert abziehen und durch die Standardabweichung der Daten teilen. Führen Sie die Schritte der Teilaufgabe a-e) danach noch einmal durch. Was ändert sich? (Zum Standardisieren können Sie die Funktion `scale` verwenden oder noch einfacher, bei der Funktion `prcomp` einfach das Argument `scale` auf `TRUE` setzen.)

```
# Standardisierung
body_sta <- (body -matrix(apply(body, 2, FUN=mean), ncol=3,
                          nrow=dim(body)[1],byrow=TRUE))/
  matrix(apply(body, 2, FUN=sd),  ncol=3, nrow=dim(body)[1],
         byrow=TRUE)
# Alternative
body_sta2 <- scale(body, center = TRUE, scale = TRUE)
#Kovarianzmatrix
cov(body_sta)
```

```
          Groesse   Gewicht     Schuh
Groesse 1.0000000 0.8826836 0.8074897
Gewicht 0.8826836 1.0000000 0.7214147
Schuh   0.8074897 0.7214147 1.0000000
```

```
sum(diag(cov(body_sta)))
```

```
[1] 3
```

```
# PCA
(pca_sta <- prcomp(body_sta))
```

```
Standard deviations (1, .., p=3):
[1] 1.6154005 0.5371167 0.3193538

Rotation (n x k) = (3 x 3):
               PC1        PC2        PC3
Groesse -0.5963947  0.1811090 -0.7819929
Gewicht -0.5771265  0.5803497  0.5745599
Schuh   -0.5578874 -0.7939734  0.2415947
```

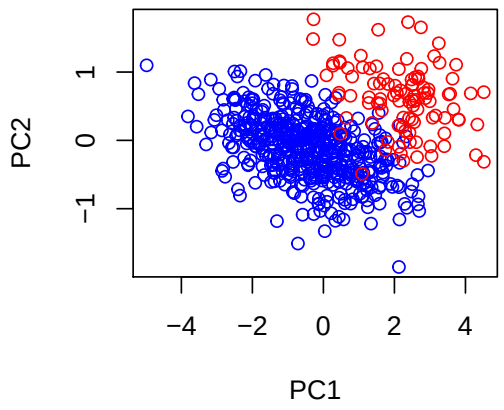
```
#Alternative
(pca_sta2 <- prcomp(body, scale=TRUE))
```

```
Standard deviations (1, .., p=3):
[1] 1.6154005 0.5371167 0.3193538

Rotation (n x k) = (3 x 3):
               PC1        PC2        PC3
Groesse -0.5963947  0.1811090 -0.7819929
Gewicht -0.5771265  0.5803497  0.5745599
Schuh   -0.5578874 -0.7939734  0.2415947
```

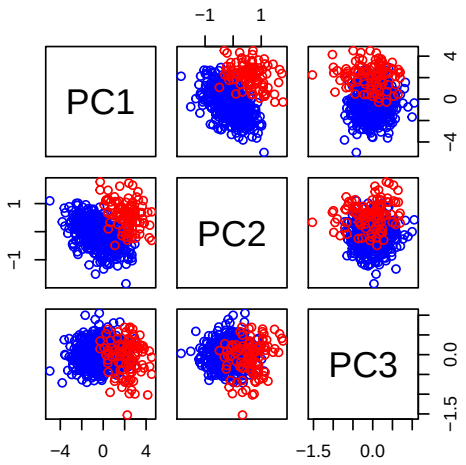
```
summary(pca_sta)
```

```
Importance of components:
                          PC1     PC2    PC3
Standard deviation     1.6154 0.53712 0.3194
Proportion of Variance 0.8698 0.09616 0.0340
Cumulative Proportion  0.8698 0.96600 1.0000
```

```
plot(PC2~PC1, data=pca_sta$x, xlab="PC1", ylab="PC2",
     col=c(rep("blue", 500), rep("red",100)))
```



```
pairs(pca_sta$x, col=c(rep("blue", 500), rep("red",100)))
```




```
# Summe der Varianz
sum(diag(cov(pca_sta$x)))
```

```
[1] 3
```

```
# Anteil der ersten beiden PCs
summary(pca_sta)
```

```
Importance of components:
                          PC1     PC2    PC3
Standard deviation     1.6154 0.53712 0.3194
Proportion of Variance 0.8698 0.09616 0.0340
Cumulative Proportion  0.8698 0.96600 1.0000
```

Die Summe der Varianzen beträgt nun für die ursprünglichen Variablen und für die Hauptkomponenten genau 3 (entspricht der Anzahl Variablen).
Die Gruppe der Kinder kann man nun bereits in den ersten beiden Hauptkomponenten etwas deutlicher erkennen. Die erste Hauptkomponente beschreibt jetzt "nur" noch 87% der Variation.

g) Jede PC ist eine linear Kombination der Original-Variablen und den Koeffizienten der Rotationsmatrix (auch Ladungen genannt). Betrachten Sie die Koeffizienten für den standardisierten und den nicht standardisierten Fall (`...$rotation`). Inwiefern unterscheiden sich diese? Würden Sie hier mit den standardisierten oder den nicht standardisierten Daten arbeiten?

```
pca$rotation
```

```
                PC1          PC2         PC3
Groesse -0.01271152  0.009491687  0.99987415
Gewicht -0.95305114 -0.302668278 -0.00924306
Schuh   -0.30254246  0.953048698 -0.01289344
```

```
pca_sta$rotation
```

```
               PC1        PC2        PC3
Groesse -0.5963947  0.1811090 -0.7819929
Gewicht -0.5771265  0.5803497  0.5745599
Schuh   -0.5578874 -0.7939734  0.2415947
```

Bei der ersten Analyse wird die PC1 hauptsächlich durch das Gewicht gesteuert, da diese Variable die absolut grössten Werte und auch die grösste Streuung hat. Die PC2 wird hauptsächlich durch die Schuhgroesse bestimmt, die PC3 durch die Grösse. Da die Grösse in Meter gemessen wurde, ist die entsprechende Streuung in Meter klein. Würde man die Grösse in Centimeter angeben, würde diese Variable einen viel grösseren Beitrag zur Streuung liefern und entsprechend stärker berücksichtigt.
Bei den standardisierten Grössen werden die unterschiedlichen Einheiten ausgeglichen, so dass alle Variablen einen ähnlichen Einfluss auf die PC1 haben.
Insgesamt macht es hier sicher Sinn mit den standardisierten Grössen zu arbeiten.

## Aufgabe 3: Abstimmungsverhalten der Kantone

Laden Sie den Datensatz *abst.Rdata*. Er enthält die Resultate (Anteil Ja-Stimmen in Prozent) der letzten 64 eidgenössischen Abstimmungen aufgeteilt nach Kantonen. Es handelt sich um aufbereitete Daten von der Webseite: https://www.bfs.admin.ch/bfs/de/home/statistiken/politik/abstimmungen.assetdetail.3362357.html. Machen Sie sich mit dem Datensatz vertraut. Genauere Informationen zu den einzelnen Abstimmungen finden Sie im Netz.

a) Gewinnen Sie zuerst einen Überblick über die Daten. Was fällt Ihnen auf? Kommentieren Sie!

```
load("Daten/abst.Rdata")
colnames(abst)
```

```
 [1] "Verbot.der.Diskriminierung.aufgrund.der.sexuellen.Orientierung"
 [2] "Mehr.bezahlbare.Wohnungen"
 [3] "Steuerreform.und.AHV.Finanzierung"
 [4] "EU.Waffenrichtlinie"
 [5] "Zersiedelungsinitiative"
 [6] "Hornkuh.Initiative"
 [7] "Selbstbestimmungs.Initiative"
 [8] "Überwachung.von.Versicherten"
 [9] "Bundesbeschluss.über.die.Velowege"
[10] "Fair.Food.Initiative"
[11] "Initiative..Für.Ernährungssouveränität."
[12] "Vollgeld.Initiative"
[13] "Geldspielgesetz"
[14] "Neue.Finanzordnung.2021"
[15] "Initiative..Abschaffung.der.Billag.Gebühren."
[16] "Reform.der.Altersvorsorge"
[17] "Zusatzfinanzierung.der.AHV"
[18] "Ernährungssicherheit"
[19] "Energiegesetz"
[20] "Unternehmenssteuerreform.III"
[21] "Fonds.für.die.Nationalstrassen.und.den.Agglomerationsverkehr"
[22] "Erleichterte.Einbürgerung.der.dritten.Ausländergeneration"
[23] "Atomausstiegsinitiative"
[24] "Nachrichtendienstgesetz"
[25] "Initiative..AHVplus..für.eine.starke.AHV."
```

```
[26] "Initiative..Grüne.Wirtschaft."
[27] "Asylgesetz"
[28] "Fortpflanzungsmedizingesetz"
[29] "Initiative..Für.eine.faire.Verkehrsfinanzierung."
[30] "Initiative..Für.ein.bedingungsloses.Grundeinkommen."
[31] "Initiative..Pro.Service.public."
[32] "Sanierung.Gotthard.Strassentunnel"
[33] "Initiative.gegen.die.Spekulation.mit.Nahrungsmitteln"
[34] "Durchsetzungsinitiative"
[35] "Initiative.gegen.die.Heiratsstrafe"
[36] "Radio..und.Fernsehengesetz..Radio..und.Fernsehgehabgabe."
[37] "Initiative..Millionen.Erbschaften.besteuern."
[38] "Stipendieninitiative"
[39] "Fortpflanzungsmedizin.und.Gentechnologie.im.Humanbereich"
[40] "Initiative..Energie..statt.Mehrwertsteuer."
[41] "Initiative..Steuerfreie.Kinderzulagen."
[42] "Gold.Initiative"
[43] "Initiative..Stopp.der.Überbevölkerung..ECOPOP.."
[44] "Initiative.zur.Abschaffung.der.Pauschalbesteuerung"
[45] "Initiative..Für.eine.öffentliche.Krankenkasse."
[46] "Initiative..Schluss.mit.der.MwSt.Diskriminierung.des.Gastgewerbes.."
[47] "Beschaffung.des.Kampfflugzeugs.Gripen"
[48] "Mindestlohn.Initiative"
[49] "Initiative..Pädophile.sollen.nicht.mehr.mit.Kindern.arbeiten.dürfen."
[50] "Medizinische.Grundversorgung"
[51] "Initiative..Gegen.Masseneinwanderung."
[52] "Initiative..Abtreibungsfinanzierung.ist.Privatsache."
[53] "Finanzierung.und.Ausbau.der.Eisenbahninfrastruktur"
[54] "Nationalstrassenabgabegesetz..NSAG."
[55] "Familieninitiative"
[56] "Initiative..1.12...für.gerechte.Löhne."
[57] "Liberalisierung.der.Öffnungszeiten.von.Tankstellenshops"
[58] "Epidemiengesetz"
[59] "Aufhebung.der.Wehrpflicht"
[60] "Asylgesetz.1"
[61] "X.Volkswahl.des.Bundesrates."
[62] "Raumplanungsgesetz"
[63] "Initiative..gegen.die.Abzockerei."
[64] "Familienpolitik"
```

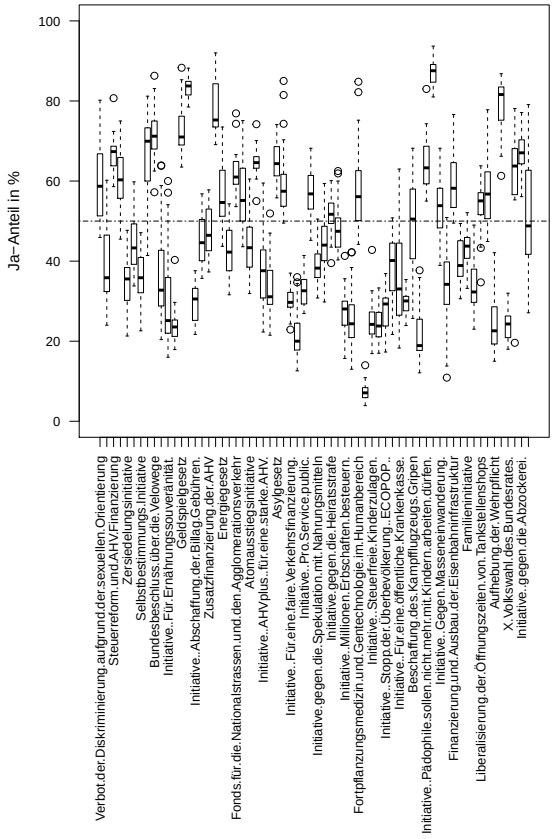
```
summary(apply(abst, MARGIN=2, FUN=median))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   7.10   32.00   43.88   45.95   58.33   87.55
```

```
summary(apply(abst, MARGIN=2, FUN=sd))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.118   5.126   6.803   6.995   8.134  13.213
```

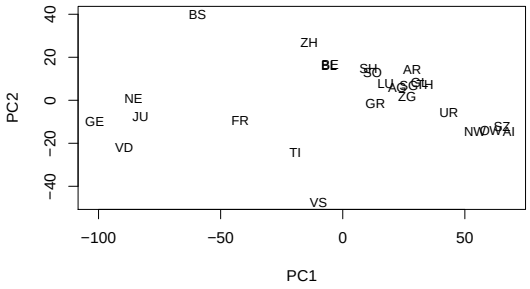
```
par(mar=c(25, 4,2,1))
boxplot(abst, las=2, ylim=c(0,100), ylab="Ja-Anteil in %", cex.axis=0.8)
abline(h=50, lty=6)
```




- Die Lage (Mittelwert oder Median) bei den einzelnen Abstimmungen schwankt zwischen 7 bis 87%, d.h. die Werte liegen fast im ganzen möglichen Wertebereich.
- Die Ergebnisse der einzelnen Abstimmungen in den verschiedenen Kantonen streuen zum Teil beträchtlich. Die Standardabweichung liegt zwischen 2 bis 13%, das ist ein Faktor von mehr als 5.
- Es gibt Ausreisser, aber nur wenige extreme (z.B. Energie statt Mehrwertsteuer oder Medizinische Grundversorgung)
- Die meisten Verteilungen sind recht symmetrisch.

b) Führen Sie eine Hauptkomponenten-Analyse durch (auf welchen Daten?) und stellen Sie die Daten in den ersten beiden Hauptkomponenten dar. Beschreiben Sie die Struktur der Daten in dieser Darstellung. Benutzen Sie als die Kantonsnamen als Label (z.B. `text(..., labels=row.names(...))`).

```
abst.pcS <- prcomp(abst, scale=F)
plot(abst.pcS$x[,1], abst.pcS$x[,2], type="n", xlab="PC1", ylab="PC2")
text(abst.pcS$x[,1], abst.pcS$x[,2], labels=row.names(abst), cex=0.8)
```




Da die Variablen/Merkmale alle in Prozent sind, kann man auch mit den unskalierten Variablen arbeiten. Allerdings werden dann die Variablen/Merkmale mit grosser Streuung die ersten Hauptachsen gegebenenfalls dominieren. Das kann in diesem Fall sogar gewünscht sein, weil Abstimmungen, bei denen sich alle Kantone gleich verhalten, ja nicht wirklich die Unterschiede zwischen den Kantonen aufzeigen.
Stellt man die Kantone aufgrund der Daten aus den Volksabstimmungen in den ersten beiden Hauptkoordinaten dar, so zeigen sich Gruppen. Auch den berüchtigte "Röstigraben" kann man sehen (Westschweiz links, ost- und zentralschweizerischen Kantone rechts). Die Richtung der zweiten Hauptkomponente scheint Kantone mit grösseren Städten von eher ländlichen Gegenden zu unterscheiden.

c) Genügen die beiden ersten Hauptkomponenten, um die Variabilität der Daten sinnvoll zu approximieren?

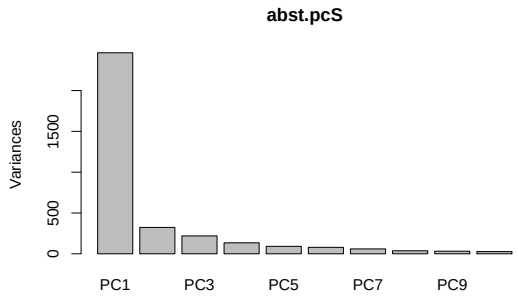
```
summary(abst.pcS)
```

```
Importance of components:
                          PC1      PC2     PC3      PC4    PC5    PC6     PC7    PC8
Standard deviation     49.6288 17.97922 14.8113 11.60676 9.5555 8.8993 7.73649 6.0895
Proportion of Variance  0.6905  0.09062  0.0615  0.03777 0.0256 0.0222 0.01678 0.0104
Cumulative Proportion   0.6905  0.78108  0.8426  0.88034 0.9059 0.9281 0.94492 0.9553
                           PC9    PC10    PC11    PC12    PC13    PC14    PC15    PC16
Standard deviation     5.64053 5.23268 4.76206 3.69454 3.27411 3.02675 2.87752 2.65256
Proportion of Variance 0.00892 0.00768 0.00636 0.00383 0.00301 0.00257 0.00232 0.00197
Cumulative Proportion  0.96423 0.97191 0.97827 0.98209 0.98510 0.98767 0.98999 0.99196
                          PC17    PC18    PC19    PC20    PC21    PC22    PC23    PC24
Standard deviation     2.51194 2.21877 2.01176 1.81206 1.70549 1.60414 1.40844 1.30011
Proportion of Variance 0.00177 0.00138 0.00113 0.00092 0.00082 0.00072 0.00056 0.00047
Cumulative Proportion  0.99373 0.99511 0.99624 0.99717 0.99798 0.99870 0.99926 0.99973
                          PC25      PC26
Standard deviation     0.97825 2.204e-14
Proportion of Variance 0.00027 0.000e+00
Cumulative Proportion  1.00000 1.000e+00
```

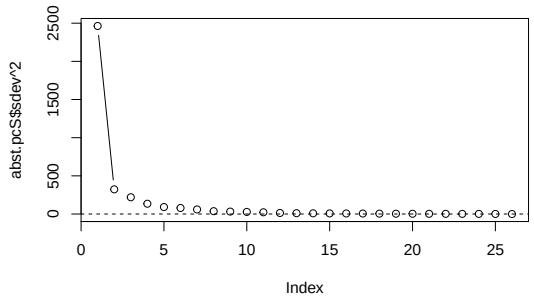
Die ersten beiden Hauptkomponenten erfassen nicht ganz 80% der Varianz. Mit 78% liegt die erklärte Varianz aber nur knapp unter der 80-20-Faustregel.

d) Benutzen Sie für die Beurteilung der Approximation das Scree-Diagramm. Kommen Sie zum gleichen Schluss wie in c)?

```
names(abst.pcS$sdev) <- paste("PC", 1:length(abst.pcS$sdev), sep="")
screeplot(abst.pcS) ## or
```




```
plot(abst.pcS$sdev^2, type="b")
abline(h=0, lty=2)
```




Ein Knick ist allenfalls bei Komponente 5 zu sehen. 5 Komponenten zu verwenden ist auch eine Option.

## Aufgabe 4: Zehnkampf

Die folgende Tabelle zeigt die Resultate vom Zehnkampf der Olympischen Spiele in Rio de Janeiro 2016. Die Daten sind im File **zehnkampf.csv** abgespeichert und enthalten folgende Variablen:

| | | | | | |
|---|---|---|---|---|---|
| `m100` | 100m Lauf (s) | `m400` | 400m Lauf (sek.) | `speer` | Speerwurf (m) |
| `weit` | Weitsprung (m) | `hurd` | 110m Hürdenlauf (s) | `m1500` | 1500m Lauf (s) |
| `kugel` | Kugelstoss (m) | `disc` | Diskuswurf (m) | `punkte` | Totale Punktzahl |
| `hoch` | Hochsprung (cm) | `stab` | Stabhochsprung (cm) | | |

a) Beurteilen Sie anhand von Boxplots, ob sich die Resultate einer PCA, basierend auf unskalierten und auf skalierten Variablen, unterscheiden werden. Begründen Sie die Antwort.

```
load("Daten/zehnkampf.Rdata")
head(zehnkampf)
```

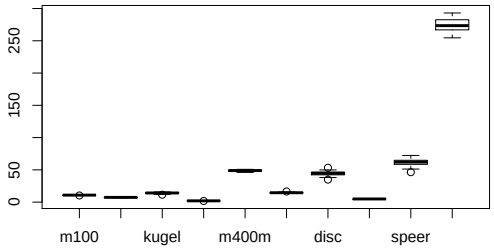
```
  Rang           Name punkte  m100 weit kugel hoch m400m  hurd  disc stab speer  m1500
1    1   Ashton Eaton   8893 10.46 7.94 14.73 2.01 46.07 13.80 45.49  5.2 59.77 263.33
2    2    Kevin Mayer   8834 10.81 7.60 15.76 2.04 48.28 14.02 46.78  5.4 65.04 265.49
```

```
3    3  Damian Warner   8666 10.30 7.67 13.66 2.04 47.35 13.58 44.93  4.7 63.19 264.90
4    4   Kai Kazmirek   8580 10.78 7.69 14.20 2.10 46.75 14.62 43.25  5.0 64.60 271.25
5    5 Larbi Bourrada   8521 10.75 7.52 13.78 2.10 47.98 14.15 42.39  4.6 66.49 254.60
6    6  Leonel Suarez   8460 11.21 7.14 14.27 2.07 48.15 14.48 47.07  4.9 72.32 268.32
```

```r
dim(zehnkampf)
```

```
[1] 23 13
```

```r
boxplot(zehnkampf[,4:13]) # spread ist schlecht zu beurteilen, da Lage sehr verschieden
```




```r
apply(zehnkampf[,4:13],2,sd)  # berechne fuer jede Spalte die Standardabweichung
```

```
      m100       weit      kugel       hoch      m400m       hurd       disc       stab
 0.2483113  0.3160815  1.0485482  0.0934516  1.1780096  0.6330031  4.1391910  0.3043168
     speer      m1500
 6.4208866 10.0432871
```

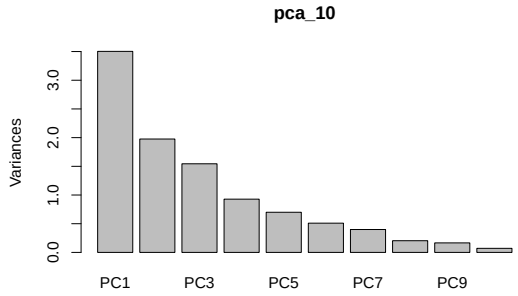
Man sollte hier skalieren, da die Variablen unterschiedliche Einheiten haben. Z.B. wird beim Laufen die Zeit gemessen, die ein Athlet braucht, beim Speerwurf wird aber gemessen, wie weit der Speer geworfen wird. Wenn die Varianzen ähnlich wären, würde die Skalierung am Ergebnis trotzdem nichts ändern. Hier sind die Standardabweichungen aber sehr verschieden.

b) Führen Sie eine PCA durch. Lassen Sie die Variable Punkte dabei weg, da es sich dabei nicht um eine Diziplin handelt. Wie viele Hauptkomponenten sind nötig, um 80% der Varianz zu erklären?

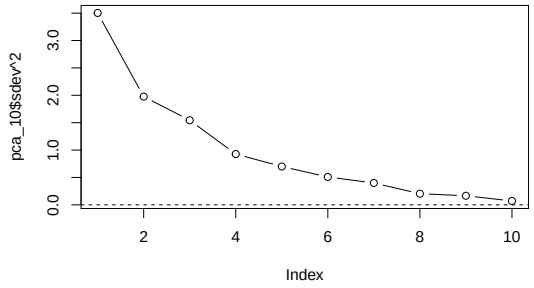
```r
pca_10 <- prcomp(zehnkampf[,4:13],  scale = TRUE)
summary(pca_10)
```

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6    PC7     PC8
Standard deviation     1.8716 1.4056 1.2427 0.96326 0.83646 0.71391 0.6316 0.45152
Proportion of Variance 0.3503 0.1976 0.1544 0.09279 0.06997 0.05097 0.0399 0.02039
Cumulative Proportion  0.3503 0.5479 0.7023 0.79511 0.86507 0.91604 0.9559 0.97632
                           PC9   PC10
Standard deviation     0.40717 0.2664
Proportion of Variance 0.01658 0.0071
Cumulative Proportion  0.99290 1.0000
```

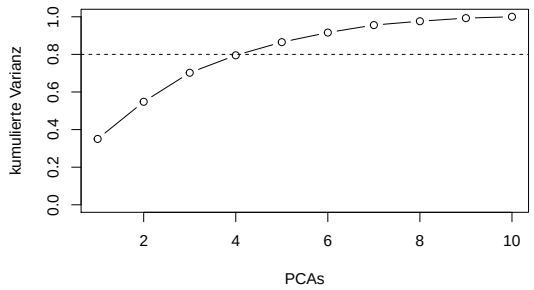
```r
## Screen-Plot
names(pca_10$sdev) <- paste("PC", 1:length(pca_10$sdev), sep="")
screeplot(pca_10)
```




```
## oder
plot(pca_10$sdev^2, type="b")
abline(h=0, lty=2)
```



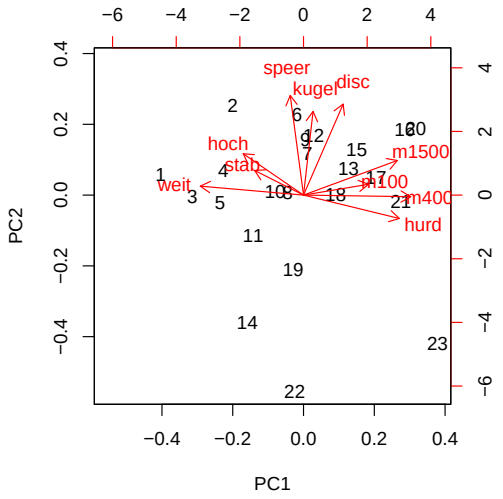

```
## kumulierte Varianz gegen Anzahl PCs
plot(1:length(pca_10$sdev), cumsum(pca_10$sdev^2)/sum(pca_10$sdev^2),
     ylim = c(0,1), ylab="kumulierte Varianz", xlab="PCAs",
     type = "b")
abline(h = 0.8, lty = 2)
```




Mit 4 Hauptkomponenten können wir knapp 80% der Varianz erklären. Bei 4 ist auch ein kleiner Knick zu sehen. Da die ersten beiden Hauptkomponenten zusammen nur 55% der Varianz erklären, gibt der 2D score-Plot, der durch die ersten beiden PCs aufgespannt wird, vermutlich, eine recht verzerrte Darstellung der Konstellation.

c) Stellen Sie die Ergebnisse der PCA in einem Biplot `biplot()` dar. Können Sie die Principal Components (PCs) im Hinblick auf die ursprünglichen Variablen interpretieren? Welche Disziplinen korrelieren miteinander? Welche Disziplinen sind im 2D Score-Plot nicht optimal wiedergegeben. Gibt es Gruppen von Athleten?

```
biplot(pca_10)
```




Im Biplot sind die projizierten ursprünglichen Feature-Vektoren dargestellt. Man sieht dass, die Sprung-, Wurf- und Laufdisziplinen jeweils miteinander korreliert sind. Sprung- und Laufdisziplinen sind eher negativ korreliert. Der Grund dafür ist, dass bei den Sprungdisziplinen grosse Werte gut sind und bei den Laufdisziplinen kleine Zeiten.
Im ursprünglichen hochdimensionalen Raum ist die Länge jedes Feature-Vektors durch die Varianz dieses Features gegeben. Für skalierte Feature sind die Feature-Vektoren im 10D Raum alle gleich lang. Nach Projektion in 2D sind die Feature-Vektoren nicht mehr gleich lang. Z.B. Stab und 100m sind deutlich kürzer. D.h. dass dieses Feature in einer Richtung, die fast orthogonal zur 2PC-Ebene ausgerichtet ist. Daher wird dieses Feature in dieser 2D Darstellung schlechter erfasst und Athleten, die in 2D nahe beieinander liegen können sehr unterschiedliche Leistung in diesen Disziplinen haben.
Die Nummern zeigen die Ränge der Athleten. Die Guten stehen eher oben rechts. Logischerweise sind das die Athleten, welche in den Sprung- und Wurfdisziplinen grosse Werte und bei den Laufdisziplinen kleine Werte haben.

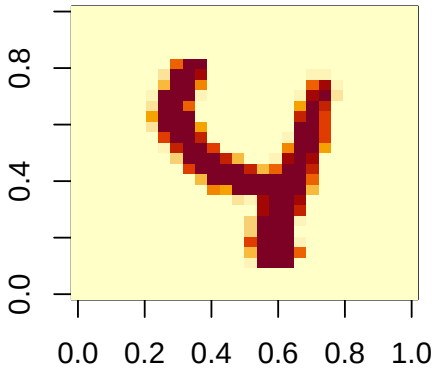
## Aufgabe 5: Ausreisserdetektion mit PCA

Laden Sie den Datensatz `mnist20x4and1000x0.Rdata`. Er erhält zwei Objekte. `x` ist eine Matrix mit Pixel-Grauwert-Daten von handgeschriebenen Zahlen, wobei jede Zeile zu einem "linearisierten" Bild gehört. Im Vektor `xlabel` sind die entsprechenden Labels zu finden. Der Datensatz enthält 1000 Nullen die mit 20 Vierern kontaminiert sind. Das Ziel dieser Aufgabe ist es, die Ausreisser mittels PCA zu identifizieren.

a) Machen Sie sich mit dem Datensatz vertraut. Einzelne Bilder können Sie wie folgt anschauen. Visualisieren Sie einige Elemente.

```
load("Daten/mnist20x4and1000x0.RData")
image_nummer <- 7 # hier können Sie ein Bild von 1 bis 1020 auswählen
```

```
bild <- matrix( x[image_nummer, ], ncol=28, byrow = TRUE)
image(t(bild[28:1,1:28]))
```




b) Führen Sie mit der Funktion `prcomp` eine Hauptkomponentenanalyse auf die Matrix `x` durch. Sollten die Variablen standardisiert werden oder nicht? Warum? Was genau macht der `prcomp` Befehl?

```
pca_klassisch <- prcomp(x)
str(pca_klassisch)
```

```
List of 5
 $ sdev    : num [1:784] 740 646 510 465 354 ...
 $ rotation: num [1:784, 1:784] -8.70e-19 8.33e-17 0.00 4.44e-16 -1.11e-16 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:784] "pixel0" "pixel1" "pixel2" "pixel3" ...
  .. ..$ : chr [1:784] "PC1" "PC2" "PC3" "PC4" ...
 $ center  : Named num [1:784] 0 0 0 0 0 0 0 0 0 0 ...
  ..- attr(*, "names")= chr [1:784] "pixel0" "pixel1" "pixel2" "pixel3" ...
 $ scale   : logi FALSE
 $ x       : num [1:1020, 1:784] -327 -689 -570 -222 -124 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:1020] "4" "4" "4" "4" ...
  .. ..$ : chr [1:784] "PC1" "PC2" "PC3" "PC4" ...
 - attr(*, "class")= chr "prcomp"
```

Alle Pixel sind Intensitätswerte (Graustufen) und haben die gleiche Einheit. Eine Skalierung ist nicht notwendig.

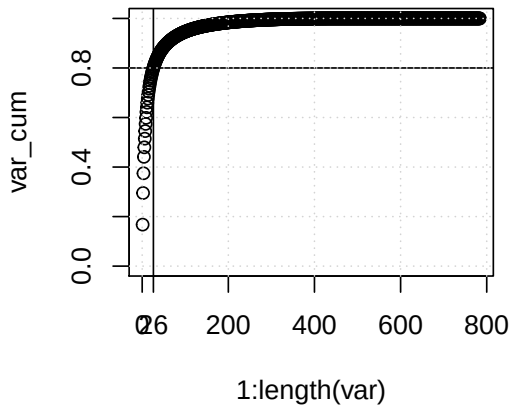
c) Wieviel Hauptkomponenten braucht es, um 80% der Varianz in den Daten zu erklären? Greifen Sie mit `$sdev` auf die Standardabweichung der Hauptkomponenten zu.

```
var <- pca_klassisch$sdev^2
var_cum <-  cumsum(var)/sum(var)
```

```
plot(1:length(var), var_cum,
      ylim=c(0,1))
(pc_number <- which( round(var_cum, 2) == 0.8))
```

```
[1] 26
```

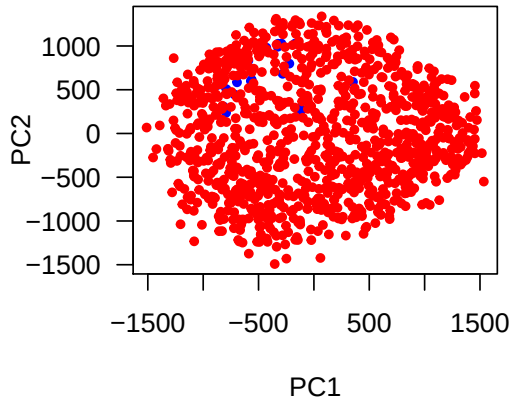
```
abline( h = 0.8, v = pc_number)
grid()
axis(1, at = pc_number, labels = pc_number )
```




26 Komponenten sind notwendig.

d) Erstellen Sie einen Scatterplot der ersten beiden Hauptkomponenten. Wählen Sie andere Farben für 4-er und 0-er Beobachtungen. Können Sie einige Ausreisser identifizieren?

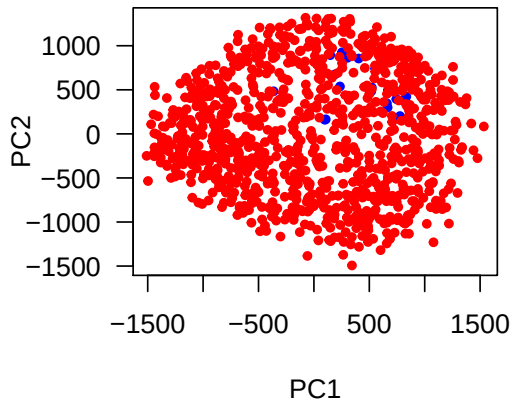
```
plot(pca_klassisch$x[,1],pca_klassisch$x[,2], col = c("red", "blue")[as.factor(xlabel)],
     pch=20, las=1, xlab="PC1", ylab="PC2")
```




Die 20 Beobachtungen, welche zu 4-er Bilder gehören, liegen mitten in den anderen Daten.

e) Laden Sie jetzt die Pakete `MASS` und `rrcov` und führen Sie mit folgendem Befehl eine robuste Hauptkomponentenanalyse durch: `pca.rob = PcaHubert(x, k = 2)`. Wofür steht das Argument k? Schauen Sie sich die Struktur des pca.rob Objekts an. Das Objekt ist von der Klasse S4, d.h. Sie können verschiedene Attributen mit `@` ansprechen, ähnlich zu `$` in einem data.frame Objekte. Plotten Sie auch hier die ersten beiden Hauptkomponenten (Tipp: `plot(pca.rob@scores, ...)`).

```
library("MASS")
library("rrcov")
pca.rob = PcaHubert(x, k = 2)
plot(pca.rob@scores[,1],pca.rob@scores[,2], col = c("red", "blue")[as.factor(xlabel)],
     pch=20, las=1, xlab="PC1", ylab="PC2")
```

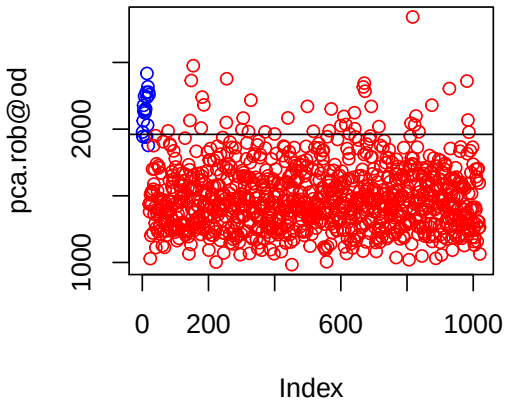



Auch in dieser Abbildung sehen wir die 4-er nicht getrennt. Das Argument $k$ entspricht dem

Anzahl berechneten Hauptkomponenten. Zu dieser Dimension wird auch die orthogonale Distanzen berechnet.

f) Nun plotten Sie die orthogonale Distanzen `pca.rob@od` der Beobachtung zur 2D-Ebene gegen die Beobachtungsnummer. Mit `abline(h = pca.rob@cutoff.od)` können Sie in den Plot noch eine vordefinierte Cutoff-Linie einzeichnen. Wählen Sie wieder andere Farben für 4-er und 0-er Beobachtungen. Welche Beobachtungen sind Ausreiser?

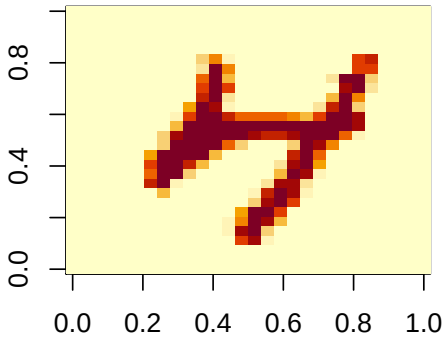
```
plot(pca.rob@od, col = c("red", "blue")[as.factor(xlabel)])
abline(h = pca.rob@cutoff.od)
```



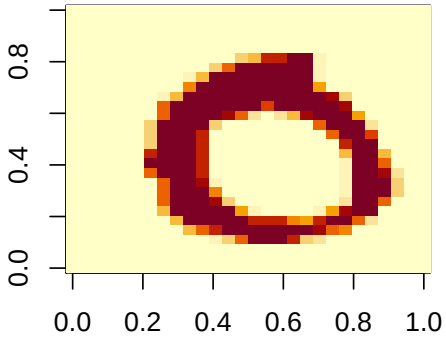

```
# identifiziere Zeilen, die nach diesem Kriterium Ausreisser sind:
(outlier.rows <- which( pca.rob@od >= pca.rob@cutoff.od ))
```

```
  4   4   4   4   4   4   4   4   4   4   4   4   4   4   4   4   0   0   0   0   0   0
  1   3   4   5   6   7   8   9  10  12  14  15  16  17  19  20  78 144 148 154 178 181
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
186 254 255 301 305 321 328 371 400 440 499 517 565 571 600 609 615 626 641 642 668 671
  0   0   0   0   0   0   0   0   0   0   0   0   0
673 692 715 812 817 823 827 836 875 928 981 984 987
```

```
## Nicht erkannte 4
bild <- matrix( x[2, ], ncol=28, byrow = TRUE)
image(t(bild[28:1,1:28]))
```




```
## fälschlicherweise als Ausreisser ausgewiesene 0
bild <- matrix( x[987, ], ncol=28, byrow = TRUE)
image(t(bild[28:1,1:28]))
```




Jetzt sieht man, dass fast alle 4-er Beobachtungen als Ausreisser erkannt werden. Es gibt aber auch andere Ausreisser.

## Aufgabe 6: Leistungsnachweis

Versuchen Sie einen eigenen Datensatz für den Leistungsnachweis zu finden.

Anforderungen an Datensatz

- Mindestens 4 quantitative Variablen
- Mindestens eine qualitative Variable, welche als Zielvariable für ein Klassifikationsproblem verwendet werden kann (nicht zu viele Stufen)

- Mindestens 100 Beobachtungen

Erste Schritte

- Datensatz einlesen
- Kurze Beschreibung des Datensatz
- allenfalls Datenaufbereitung
- Kurze summary-Statistik