Introduction to R

We need to use R as our main statistical programming language and as well as a software for MySQL data.

1 First steps

For the first exercize retrieve data from Vizier (2MASS) for 1 deg area in csv format, file `twomass.csv`.

1. start R:

```
R version 2.9.0 (2009-04-17)
Copyright (C) 2009 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

2. load data from the file

```
> t = read.csv('twomass.csv', header=TRUE,sep="|")
```

Note, that `sep` is used only if you have non-standart separator. The file should be adjusted (all leading strings with format should be removed, but leave a string with headers).

```
_r|_RAJ2000|_DEJ2000|RAJ2000|DEJ2000|2MASS|Jmag|e_Jmag|Hmag|e_Hmag|Kmag|e_Kmag|Qflg|Rflg|Bflg|Cflg|X
0.003535|020.000195|+29.996469|020.000195|+29.996469|01200004+2959472 |15.650| 0.068|15.295| 0.120|1
```

t is what is called a data frame. A data frame is pretty much like a table with named columns and the names of the columns can be printed out with the names function.

```
> names(t)
 [1] "X_r"       "X_RAJ2000" "X_DEJ2000" "RAJ2000"   "DEJ2000"   "X2MASS"
 [7] "Jmag"      "e_Jmag"    "Hmag"      "e_Hmag"    "Kmag"      "e_Kmag"
[13] "Qflg"      "Rflg"      "Bflg"      "Cflg"      "Xflg"      "Aflg"
```

3. The next is to calculate some of the statistical quantities we saw in the lecture and to do that we need to access the individual columns of t:

```
> mean(t$RAJ2000)
[1] 19.99512
> median(t$RAJ2000)
[1] 19.98377
> sd(t$RAJ2000)
[1] 0.5732393
```

This will go wrong sometimes:

```
> mean(t$e_Jmag)
[1] NA
```

- there are NA (not available) values in this column. It is possible to remove them:

```
> mean(t$e_Jmag,na.rm=TRUE)
[1] 0.07895541
```

NB: data frames are usiful analog oftables but sometimes you need just arrays:

```
>x<-array(1:20,dim=c(1,20))
```

array of 20 elements of $1x20$ dimensions.

```
>x<-c(1,2,3,4,5,6,7,8,9,10)
```

vector of 10 elements

```
>df=data.frame(x,0.5*x)
```

and a new data frame.

2 Visualization

Simple plots:

```
>plot(t$RAJ2000,t$DEJ2000)
>plot(t$RAJ2000,t$DEJ2000,xlab="RA, deg",ylab="DEC, deg")
>plot(t$RAJ2000,t$DEJ2000,main="2MASS",xlab="RA, deg",ylab="DEC, deg")
>plot(t$RAJ2000,t$DEJ2000,main="2MASS",xlab="RA, deg",ylab="DEC, deg",pch=20,col="red")
```

Additional abilities come with the lattice library

```
>library(lattice)
>xyplot(t$Jmag ~ t$Jmag-t$Hmag)
>xyplot(t$Jmag ~ t$Jmag-t$Hmag)
>xyplot(t$Jmag ~ t$Jmag-t$Hmag, xlim=c(-2,4), ylim=c(18,5), xlab="J-H",ylab="J")
>histogram(t$Jmag)
>densityplot(t$Jmag)
```

Now let us have some fun:

```
xyplot(t$Jmag ~ t$Jmag-t$Hmag | Qflg, data=t)
```

And now seriosly:

```
>xyplot(t$Jmag[t$Qflg=='AAA'] ~ t$Jmag[t$Qflg=='AAA']-t$Hmag[t$Qflg=='AAA'], xlim=c(-2,4), ylim=c(18,5),
```

Compare this plot to the previous CMD, without selection.

3 Task - [Fe/H] distribution

1. Repeat step 2 from Werkcollege 3, but this time select all objects with there galactic coordinates and metallicities (drop all objects without metallicities), export the result in csv file and load in R
2. Build histograms for distribution of objects in galactic latitude for different intervals of metallicity
3. Build coordinate plot for the distribution of objects on coordinate plane,use different symbols and colors for different metallicity bins.

4 Kernel smoothing

1. Connect to your MySQL database from R
2. select J magnitudes from 2MASS table

```
>jmag <- dbGetQuery(con, "select t1.JMAG from TWOMASS t1")
```

3. find brightest and faintest stars in the sample

```
> min(jmag$JMAG);max(jmag$JMAG)
[1] 4.9
[1] 18.657
```

4. plot a histogram for J magnitude
5. find a density estimation with epanechnikov kernel and bandwidth 0.1 - note, input is a vector not data frame

```
rd<-density(jmag$JMAG,kernel=c("epanechnikov"),bw=0.1)
```

6. plot the resulting distribution

```
>plot(rd)
```

7. find the completeness limit

```
> print(rd$x[rd$y==max(rd$y)])
[1] 16.54075
```

5 Task - Completeness limis in original and cross-identified data

1. Repeat previous tasks for all magnitudes from original and cross-identified catalogs (remember to write a correct SQL statement)
2. Plot all density functions
3. Write a table of completeness limits

6 Dummy PCA

1. generate a sample of the random data

```
>x=sample(100)-50.0+rnorm(100)
>y=-0.23*x+4.0+0.2*rnorm(100)
```

2. create a data frame

```
>xy<-data.frame(x,y)
```

3. perfome PCA with R function prcomp. Note, that the averages will be deduced automatically

```
> rp<-prcomp(xy)
```

Compare deviations with original values:

```
> print(rp)
```

4. Use PCs to compute new coordinates and plot them.

7 Dummy Linear Descrimination

1. Generate dummy dataset:

```
>x<-5.0+0.5*rnorm(50)
>y<-0.0+0.5*rnorm(50)
```

   Create an empty classificator and group all in data frame

```
>cl<-rnorm(50)
>d<-data.frame(cl,x1,x2)
>names(d)<-c("cl","x1","x2")
```

2. "Classify" it according to some combination of (x,y)

```
>d$cl[d$x1 > d$x2]=0
>d$cl[d$x1 <= d$x2]=1
```

   Plot 2 "samples"

```
>plot(d$x1[d$cl==0],d$x2[d$cl==0],col="blue")
>points(d$x1[d$cl==1],d$x2[d$cl==1])
```

3. Performe lda

```
>library(MASS)
>g<-lda(y~x1+x2,data=d)
```

4. Find a line which divide 2 samples:

```
>gmean<-g$prior%*%g$means
>const<-as.numeric(gmean%*%g$scaling)
>a<- -g$scaling[1]/g$scaling[2]
>b<-const/g$scaling[2]
```

   Overplot this line

```
>xp<-c(-100,100)
>yp<-a*xp+b
>lines(xp,yp,col="red")
```

5. You can use g for classification of other samples

```
>pr<-predict(g,inp_sample)
```