Using the `telemetr` package

August 4, 2012

Testing With Random Data

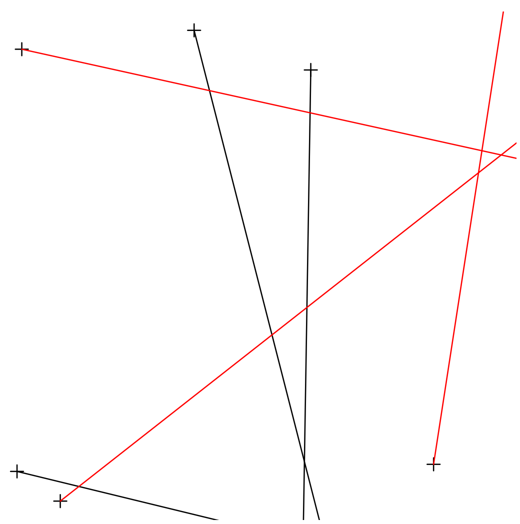
First we'll generate some random data. For each of two animals we'll get three bearings on a single day.

```
> set.seed(999)
> library(telemetr)
> library(ggplot2)
> m = telemetr:::makeMoreTriData(ntowers=3,animals=letters[1:2],dates=Sys.Date())
> summary(m)
```

```
Object of class SpatialPointsDataFrame
Coordinates:
     min   max
x 0.0947 0.787
y 0.0700 0.853
Is projected: NA
proj4string : [NA]
Number of points: 6
Data attributes:
 animal      date               thetaTrue          bearing            theta
 a:3    Min.   :2012-08-04   Min.   :-1.538   Min.   :  8.76   Min.   :0.666
 b:3    1st Qu.:2012-08-04   1st Qu.:-1.029   1st Qu.: 64.51   1st Qu.:2.237
        Median :2012-08-04   Median :-0.152   Median :103.06   Median :4.829
        Mean   :2012-08-04   Mean   :-0.192   Mean   :102.22   Mean   :3.975
        3rd Qu.:2012-08-04   3rd Qu.: 0.469   3rd Qu.:150.16   3rd Qu.:5.774
        Max.   :2012-08-04   Max.   : 1.336   Max.   :180.93   Max.   :6.066
```

We'll plot the bearings, colouring by animal:

```
> plot(m)
> telemetr:::drawVectors(~bearing|animal,m)
```

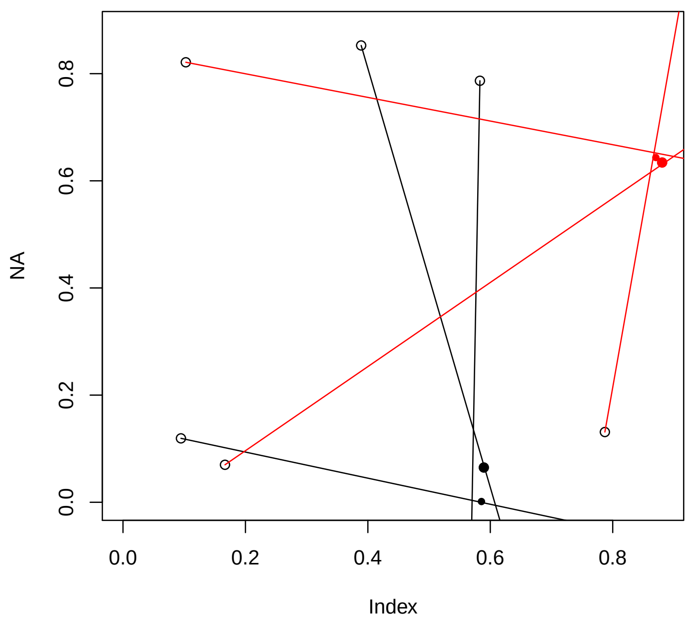


Now we locate using two methods:

```
> trmr = triang(~bearing|animal,m,method="rmr")
> tmle = triang(~bearing|animal,m,method="mle")
```

Next we plot the points and the location estimates:

```
> plot(NA,xlim=range(coordinates(m)[,1],coordinates(trmr[,1]),coordinates(tmle[,1])),
+      ylim=range(coordinates(m)[,2],coordinates(trmr[,2]),coordinates(tmle[,2])))
> points(m)
> telemetr:::drawVectors(~bearing|animal,m)
> points(trmr,pch=19,col=1:2)
> points(tmle,pch=20,col=1:2)
>
```



Testing with Lenth Data

The sample data from Lenth's Technometrics article was typed in and is available the `samples` folder. We read it in. One of the readings was used to test the robustness of the estimators, so we will add a flag column to the data. This table should be close to Lenth's table 1.

```
> lenth = read.table(system.file("samples","lenth.dat",package="telemetr"))
> coordinates(lenth)=~x+y
> lenth$ok = 1
> lenth$ok[6]=0
> lenth
```

```
   coordinates bearing ok
1     (9, 3.2)     234  1
2   (9.4, 5.5)     215  1
3 (9.95, 9.05)     196  1
4   (8.7, 8.7)     193  1
5  (8.07, 9.7)     188  1
6     (5.6, 9)     250  0
7   (4.4, 8.9)     160  1
8 (1.85, 5.25)     118  1
```

Now we loop over the methods and also whether to include the outlier or not. The resulting table should duplicate Lenth's table 2 once all the methods and error estimates are working:

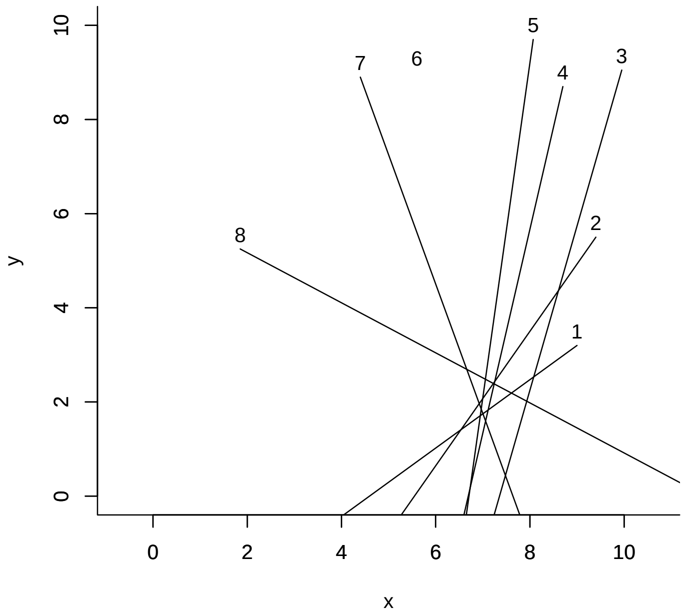
```
> results = data.frame(method=NULL,x=NULL,y=NULL)
> for(subset in list(lenth$ok==1,TRUE)){
+   for(method in c("mle","hub","and","rmr")){
+     tri = triang(~bearing,lenth,method=method,subset=subset)
+     results=rbind(results,data.frame(
+       npts=tri$npts,method=method,
+       x=coordinates(tri)[,1],y=coordinates(tri)[,2]))
+   }
+ }
> options(digits=3)
>   results
```

```
   npts method    x    y
x     7    mle 7.23 1.98
x1    7    hub 7.21 1.97
x2    7    and 7.20 1.96
x3    7    rmr 7.09 1.79
x4    8    mle 5.87 1.13
x5    8    hub 7.11 1.90
x6    8    and 7.20 1.96
x7    8    rmr 7.07 1.82
```

Currently the point estimates are exact, or close to 1dp for most cases.

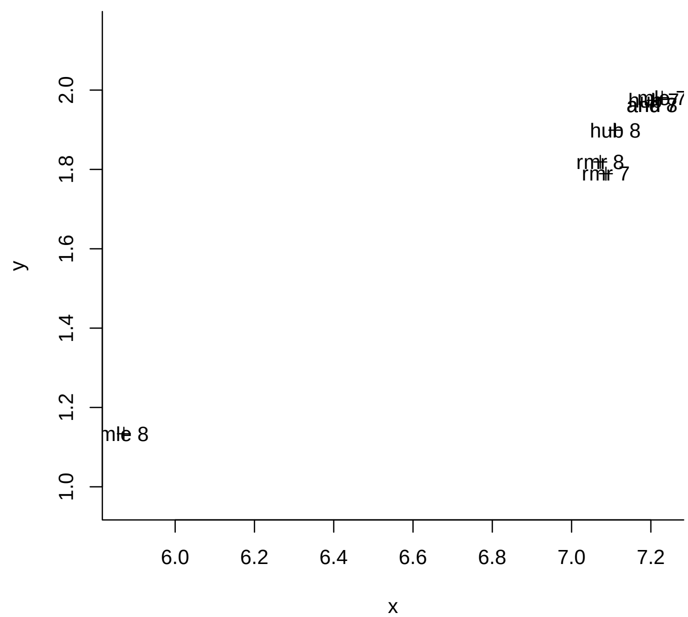
The following plot emulates Lenth's figure 1:

```
> plot(coordinates(lenth),xlim=c(0,10),ylim=c(0,10),bty="l",type="n",asp=1)
> text(coordinates(lenth)[,1],coordinates(lenth)[,2]+.3,as.character(1:nrow(lenth)))
> axis(1)
> axis(2)
> drawVectors(~bearing,lenth,subset=lenth$ok==1)
>
```



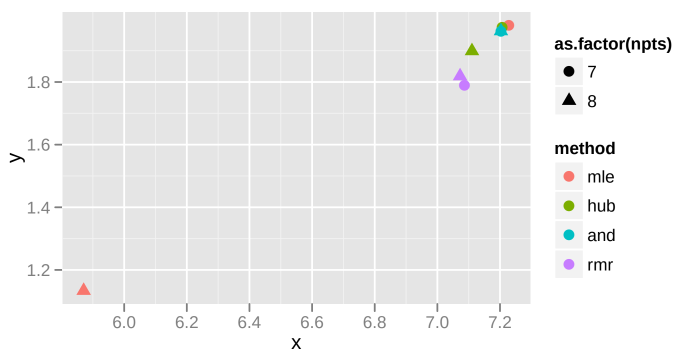
We can try and emulate Lenth's figure 3 but the labelling is problematic.

```
> coordinates(results)=~x+y
> plot(coordinates(results),type="n",asp=1,bty="l")
> plot(results,add=TRUE)
> text(coordinates(results),paste(results$method,results$npts))
```



Another option is to use `ggplot` to style the points:

```
> print(
+       ggplot(as.data.frame(results))+
+    geom_point(aes(x=x,y=y,colour=method,pch=as.factor(npts)),cex=3)+coord_fixed()
+       )
```

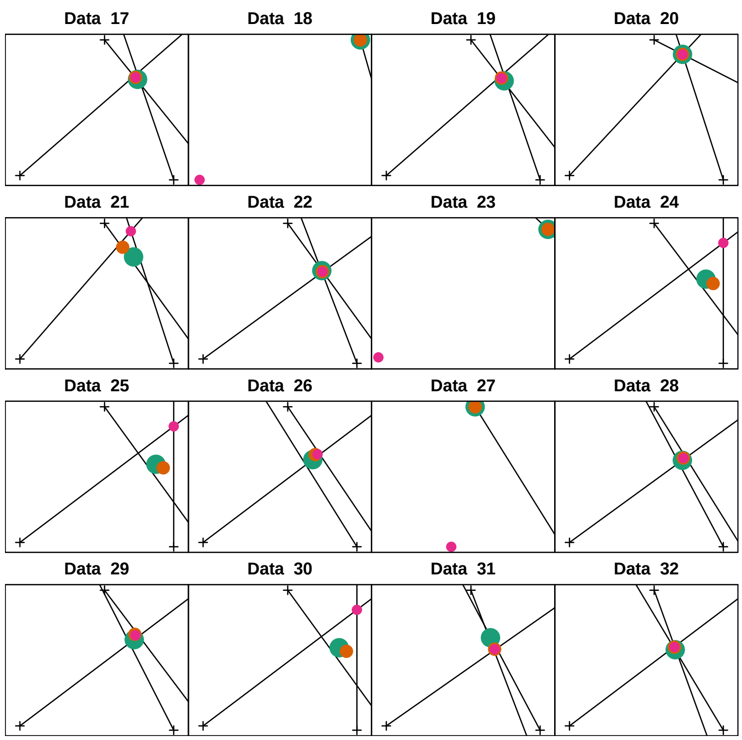


Demo Data

Another batch of demo data was supplied with the Fortran source code. This consisted of a number of azimuths from three towers (although a tower location file with 8 towers was supplied, only three seem to be used in the data). This was converted into a spatial points data frame as `azr` in the package.

```
> data(azr)
> mlefits = triang(~bearing|animal,azr,method="mle")
> rmrfits = triang(~bearing|animal,azr,method="rmr")
> andfits = triang(~bearing|animal,azr,method="and")
> bb = bbox(rbind(coordinates(azr),
+   coordinates(mlefits),
+   coordinates(rmrfits),
+   coordinates(andfits)))
> par(mfrow=c(4,4))
> par(mar=c(0,0,2,0))
> colours=c("#1B9E77", "#D95F02", "#7570B3", "#E7298A")
> for(i in 17:32){
+   bb = bbox(rbind(coordinates(azr),coordinates(rmrfits[i,]),
+     coordinates(mlefits[i,]),coordinates(andfits[i,])))
```

```
+    plot(SpatialPoints(t(bb)),cex=0)
+    plot(azr[azr$animal==i,],add=TRUE)
+    drawVectors(~bearing|animal,azr,subset=azr$animal==i)
+    points(rmrfits[i,],pch=19,col=colours[1],cex=3)
+    points(mlefits[i,],pch=19,col=colours[2],cex=2)
+    points(andfits[i,],pch=19,col=colours[4],cex=1.5)
+    box()
+    title(paste("Data ",i))
+ }
>
```



Appendix

On Finding the Source of a Signal Author(s): Russell V. Lenth Reviewed work(s): Source: Technometrics, Vol. 23, No. 2 (May, 1981), pp. 149-154