

Chapter 7: Yield Stability Analysis

Jixiang Wu
Associate Professor of Quantitative Genetics/Biostatistics
AHPS Department
South Dakota State University, Brookings, SD 57007
Phone: (605)688-5947
Email: jixiang.wu@sdstate.edu

Introduction

There are several ways to approach yield stability. In this document, we will focus on using two R packages to run regression based yield stability analysis and AMMI stability analysis. Please install three packages: agricolae, minque, and qgtools if you haven't done yet.

Finlay and Wilkinson's Regression Based Analysis

The first data set we will use are from a cotton yield trial from China conducted in 1989 and 1990. With this data set, we will conduct the following data analyses: GGE analysis, FW analysis, and FW analysis with bootstrapping.

With the following R codes, we will load the data named cotchina890.txt, which is available in D2L (under Data Sets).

```
cot<-read.table("cotchina890.txt",header=TRUE)
#cot
cot$Env=paste(cot$Year,cot$Loc,sep=":") ## combine Years and Locations as new
environments

cot=cot[order(cot$Gen,cot$Env),]

ok=complete.cases(cot) # check rows without missing value(s)
cot1=cot[ok,] ##delete missing rows and save it to cot1
```

GGE Analysis

With GGE analysis, we will need load the R package minque. Make sure you install this package before using it. The results will be saved to res1 as a list. It must be pointed out that this analysis is based on MINQUE approach with 10-group randomized jackknife method. This part of analysis is a little overlapped with the previous chapter.

```
rm(list=ls())
```

```
require(minque)

## Loading required package: minque

## Warning: package 'minque' was built under R version 3.4.4

## Loading required package: klaR

## Loading required package: MASS

## Loading required package: Matrix

res1=lmm.jack(Yld~1|Year*Loc+Year*Gen+Loc*Gen,data=cot1)[[1]]
```

Variance components

Both estimated variance components and proportional variance components are provided.

```
data.frame(res1$Var[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
V(Year)	4.42	1.576	0.079	-1.147	9.98
V(Loc)	6.91	4.687	0.519	-9.650	23.46
V(Gen)	27.84	1.213	0.000	23.555	32.12
V(Year:Loc)	125.62	7.327	0.000	99.741	151.50
V(Year:Gen)	0.77	0.445	0.385	-0.803	2.34
V(Loc:Gen)	8.07	1.794	0.006	1.731	14.40
V(e)	36.08	2.255	0.000	28.115	44.04

```
data.frame(res1$PVar[[1]])
```

	Estimate	SE	PValue	X2.5.LL	X97.5.UL
V(Year)/VP	0.021	0.008	0.080	-0.006	0.048
V(Loc)/VP	0.033	0.022	0.514	-0.045	0.111
V(Gen)/VP	0.133	0.005	0.000	0.116	0.150
V(Year:Loc)/VP	0.599	0.035	0.000	0.476	0.722
V(Year:Gen)/VP	0.004	0.002	0.388	-0.004	0.011
V(Loc:Gen)/VP	0.038	0.008	0.005	0.009	0.068
V(e)/VP	0.172	0.011	0.000	0.132	0.212

```
#
```

Predicted Effects

```
PE=data.frame(res1$RandomEffect[[1]])
## only extract genotypic effect
PE[c(27:34),]
```

	Pre	SE	PValue	X2.5.LL	X97.5.UL
Gen(1)	-9.915	0.293	0.000	-10.949	-8.881
Gen(2)	-1.891	0.205	0.000	-2.616	-1.167
Gen(3)	4.294	0.323	0.000	3.152	5.435
Gen(4)	0.305	0.242	0.638	-0.548	1.159
Gen(5)	-0.961	0.351	0.089	-2.202	0.281
Gen(6)	6.493	0.415	0.000	5.027	7.960
Gen(7)	-2.984	0.321	0.000	-4.118	-1.849
Gen(8)	4.658	0.412	0.000	3.202	6.114

FW Analysis

FW analysis without bootstrapping

```

EI<-tapply(cot$Yld,cot$Env,mean) ##Calculate environmental index

EI.data<-as.data.frame.table(EI) # Covert EI into a dataframe
colnames(EI.data)=c("Env","EI")

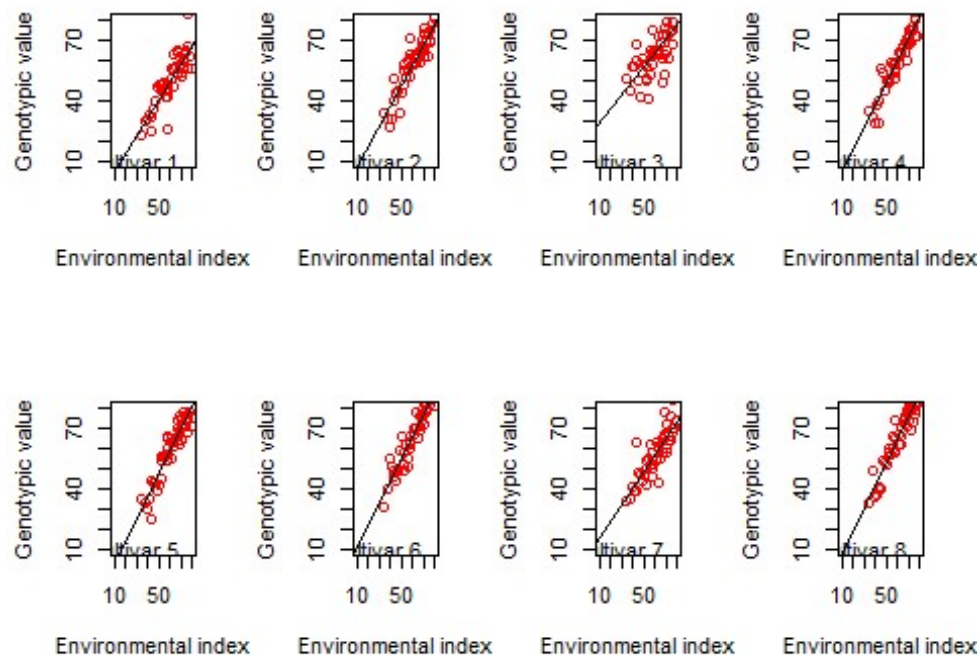
gn=length(unique(cot$Gen))
r2=numeric(gn)
b1=numeric(gn)
par(mfrow=c(2,4))
for(i in 1:8){
  #i=1
  id=which(cot$Gen==i)
  dat1=data.frame(cot[id,],EI)
  #reg=lm(Yld~EI,data=dat1)
  #summary(a)

  plot(Yld~EI,col="red",xlim=c(10,80),ylim=c(10,80),xlab="Environmental
index",ylab="Genotypic value",data=dat1)
  str<-paste("Cultivar",i,sep=" ")
  text(80,10,str,pos=2)

  ##legend(locator(1),"Genotype 1",pch=1,col=2)
  reg<-lm(Yld~EI,data=dat1)
  ##summary(reg)
  r2[i]<-summary(reg)$r.squared
  b1[i]<-coef(reg)[2]

  abline(reg)
}

```



```
b0=b1
r0=r2
data.frame(b0,r0)
```

b0	r0
0.905	0.732
1.023	0.842
0.700	0.487
1.109	0.901
1.131	0.880
1.137	0.877
0.839	0.720
1.155	0.889

FW analysis with bootstrapping

```
B=1000
R2=matrix(0,B,gn)
B1=matrix(0,B,gn)
for(i in 1:gn){
  id=which(cot$Gen==i)
  dat0=data.frame(cot[id,],EI)
  for(k in 1:B){
    id1=sample(length(id),replace=T)
```

```

    y=dat0$Yld[id1]
    x=dat0$EI[id1]
    dat1=data.frame(y,x)
    reg<-lm(y~x,data=dat1)

    R2[k,i]=summary(reg)$r.squared
    B1[k,i]<-coef(reg)[2]
  }
}

CI1=matrix(0,gn,2)
m1=numeric(gn)
for(i in 1:gn){
  v=R2[,i]
  CI1[i,]=quantile(v,p=c(0.025,0.975))
  m1[i]=mean(v)
}

CI2=matrix(0,gn,2)
m2=numeric(gn)
for(i in 1:gn){
  v=B1[,i]
  CI2[i,]=quantile(v,p=c(0.025,0.975))
  m2[i]=mean(v)
}

result.r=data.frame(r0,m1,CI1)
colnames(result.r)=c("Orig","Boot","LL","UL")
rownames(result.r)=paste("Cultivar",1:gn,sep="")

result.b=data.frame(b0,m2,CI2)
colnames(result.b)=c("Orig","Boot","LL","UL")
rownames(result.b)=paste("Cultivar",1:gn,sep="")

data.frame(result.r)

```

	Orig	Boot	LL	UL
Cultivar1	0.732	0.735	0.598	0.860
Cultivar2	0.842	0.838	0.726	0.915
Cultivar3	0.487	0.492	0.289	0.679
Cultivar4	0.901	0.904	0.834	0.948
Cultivar5	0.880	0.882	0.822	0.928
Cultivar6	0.877	0.876	0.806	0.932
Cultivar7	0.720	0.719	0.539	0.859
Cultivar8	0.889	0.889	0.826	0.937

data.frame(result.b)

	Orig	Boot	LL	UL
Cultivar1	0.905	0.904	0.756	1.061
Cultivar2	1.023	1.022	0.882	1.158
Cultivar3	0.700	0.704	0.480	0.930
Cultivar4	1.109	1.110	0.976	1.242
Cultivar5	1.131	1.131	1.011	1.270
Cultivar6	1.137	1.137	0.989	1.276
Cultivar7	0.839	0.840	0.688	0.988
Cultivar8	1.155	1.156	1.026	1.283

We may also use the R function from the package qgtools. The following R codes demonstrate the use of this R function.

```
``{r}
```

```
library(qgtools) y=y=cot1$Yld Gen=paste("G",cot1$Gen,sep="") Env=cot1$Env
data.frame(y,Gen,Env)[1:10,]
```

```
res=stab.fw(y,Gen=Gen,Env=Env,times=1000, Rep=F) res
```

```
library(qgtools) data(maize) #names(maize)
```

```
Geno=as.vector(maizeCultivar)Env = paste(maizeLocation,maize$Year,sep=":")
y=maize$Yld
```

```
res=stab.fw(y,Gen=Geno,Env=Env,times=10,Rep=TRUE) res
```

```
````
```

## AMMI Analysis

The data we will use for analysis are from the R package agricolae. The following code can obtain the data set plrv.

```
library(agricolae)
data(plrv)
head(plrv)
```

| Genotype | Locality | Rep | WeightPlant | WeightPlot | Yield |
|----------|----------|-----|-------------|------------|-------|
| 102.18   | Ayac     | 1   | 0.510       | 5.10       | 18.9  |
| 104.22   | Ayac     | 1   | 0.345       | 2.76       | 12.8  |
| 121.31   | Ayac     | 1   | 0.542       | 4.34       | 20.1  |

|        |      |   |       |      |      |
|--------|------|---|-------|------|------|
| 141.28 | Ayac | 1 | 0.989 | 8.90 | 36.6 |
| 157.26 | Ayac | 1 | 0.625 | 5.00 | 23.1 |
| 163.9  | Ayac | 1 | 0.512 | 2.56 | 19.0 |

## GGE Analysis

Again, first we will run GGE model analysis. By doing so, you will need to use the functions from the R package minque. GGE analysis could provide useful information to reveal some stability results.

### Variance components and proportional variance components

```
ok=complete.cases(plrv)
dat=plrv[ok,] ##delete missing data
res1=lmm.jack(Yield~1|Locality*Genotype+Rep:Locality,data=dat)[[1]]
Variance components for yield
data.frame(res1$Var[[1]])
```

|                      | Estimate | SE    | PValue | X2.5.LL | X97.5.UL |
|----------------------|----------|-------|--------|---------|----------|
| V(Locality)          | 288.12   | 3.974 | 0.000  | 274.086 | 302.16   |
| V(Genotype)          | 26.21    | 1.512 | 0.000  | 20.874  | 31.55    |
| V(Locality:Genotype) | 46.38    | 2.982 | 0.000  | 35.842  | 56.91    |
| V(Locality:Rep)      | 2.03     | 0.542 | 0.018  | 0.115   | 3.94     |
| V(e)                 | 36.95    | 2.216 | 0.000  | 29.124  | 44.78    |

```
Proportional variance components for yield
data.frame(res1$PVar[[1]])
```

|                         | Estimate | SE    | PValue | X2.5.LL | X97.5.UL |
|-------------------------|----------|-------|--------|---------|----------|
| V(Locality)/VP          | 0.721    | 0.008 | 0.000  | 0.692   | 0.749    |
| V(Genotype)/VP          | 0.066    | 0.004 | 0.000  | 0.053   | 0.079    |
| V(Locality:Genotype)/VP | 0.116    | 0.007 | 0.000  | 0.090   | 0.142    |
| V(Locality:Rep)/VP      | 0.005    | 0.001 | 0.015  | 0.000   | 0.010    |
| V(e)/VP                 | 0.092    | 0.005 | 0.000  | 0.074   | 0.111    |

You might see that locality effects made the largest contribution (72%) to the total variance in yield while genotypes with 6.5% contribution. Genotype-by-locality interaction effects made 11.6% contribution, almost doubling comparing to genotype effects. The results suggested that the locations selected for these yield trials were very environmentally diverse while genotypes selected for yield trial were more similar comparing locations. It is common that selected genotypes are similar regarding yield.

The following predicted effects could reveal individual location effects, genotype effects, and their interaction effects.

### Location effects

```
n1=length(unique(dat$Locality))
PE=data.frame(res1$RandomEffect[[1]])
LE=PE[1:n1,]
LE
```

|                  | Pre    | SE    | PValue | X2.5.LL | X97.5.UL |
|------------------|--------|-------|--------|---------|----------|
| Locality(Ayac)   | -6.93  | 0.311 | 0      | -8.03   | -5.83    |
| Locality(LM-02)  | 3.95   | 0.159 | 0      | 3.39    | 4.51     |
| Locality(SR-02)  | -15.64 | 0.223 | 0      | -16.43  | -14.85   |
| Locality(Hyo-02) | 14.99  | 0.226 | 0      | 14.19   | 15.79    |
| Locality(LM-03)  | 23.04  | 0.294 | 0      | 22.00   | 24.08    |
| Locality(SR-03)  | -19.41 | 0.142 | 0      | -19.91  | -18.91   |

### Genotype effects

```
ng=length(unique(dat$Genotype))
e1=n1+1
e2=n1+ng
GE=PE[e1:e2,]
GE
```

|                  | Pre    | SE    | PValue | X2.5.LL | X97.5.UL |
|------------------|--------|-------|--------|---------|----------|
| Genotype(102.18) | -3.723 | 0.416 | 0.000  | -5.194  | -2.253   |
| Genotype(104.22) | 0.546  | 0.436 | 0.644  | -0.994  | 2.086    |
| Genotype(121.31) | -0.514 | 0.194 | 0.102  | -1.201  | 0.173    |
| Genotype(141.28) | 7.742  | 0.692 | 0.000  | 5.299   | 10.185   |
| Genotype(157.26) | 5.337  | 0.444 | 0.000  | 3.770   | 6.904    |
| Genotype(163.9)  | -7.824 | 0.507 | 0.000  | -9.615  | -6.032   |
| Genotype(221.19) | -6.532 | 0.406 | 0.000  | -7.966  | -5.098   |
| Genotype(233.11) | -1.726 | 0.388 | 0.006  | -3.096  | -0.356   |
| Genotype(235.6)  | 6.781  | 0.489 | 0.000  | 5.055   | 8.507    |
| Genotype(241.2)  | -3.661 | 0.252 | 0.000  | -4.552  | -2.770   |
| Genotype(255.7)  | -0.068 | 0.266 | 0.984  | -1.006  | 0.870    |
| Genotype(314.12) | -2.112 | 0.538 | 0.014  | -4.013  | -0.212   |
| Genotype(317.6)  | 4.012  | 0.509 | 0.000  | 2.212   | 5.811    |
| Genotype(319.20) | 6.802  | 0.846 | 0.000  | 3.814   | 9.789    |
| Genotype(320.16) | -3.686 | 0.375 | 0.000  | -5.011  | -2.361   |
| Genotype(342.15) | -3.981 | 0.413 | 0.000  | -5.441  | -2.520   |
| Genotype(346.2)  | -5.818 | 0.454 | 0.000  | -7.421  | -4.215   |



|                   |         |       |       |         |        |
|-------------------|---------|-------|-------|---------|--------|
| Genotype(351.26)  | 4.653   | 0.490 | 0.000 | 2.921   | 6.384  |
| Genotype(364.21)  | 2.883   | 0.287 | 0.000 | 1.868   | 3.897  |
| Genotype(402.7)   | -2.720  | 0.397 | 0.000 | -4.124  | -1.317 |
| Genotype(405.2)   | -1.371  | 0.611 | 0.189 | -3.531  | 0.788  |
| Genotype(406.12)  | 1.691   | 0.515 | 0.037 | -0.128  | 3.510  |
| Genotype(427.7)   | 4.631   | 0.543 | 0.000 | 2.712   | 6.550  |
| Genotype(450.3)   | 4.705   | 0.767 | 0.001 | 1.995   | 7.415  |
| Genotype(506.2)   | 2.230   | 0.567 | 0.014 | 0.228   | 4.233  |
| Genotype(Canchan) | -3.078  | 0.395 | 0.000 | -4.474  | -1.681 |
| Genotype(Desiree) | -12.307 | 0.782 | 0.000 | -15.069 | -9.545 |
| Genotype(Unica)   | 7.110   | 0.515 | 0.000 | 5.292   | 8.927  |

### Genotype-by-location interaction effects

``{r}

e1=e2+1 e2=e2+ng\*nl GLE=PE[e1:e2,] GLE

...

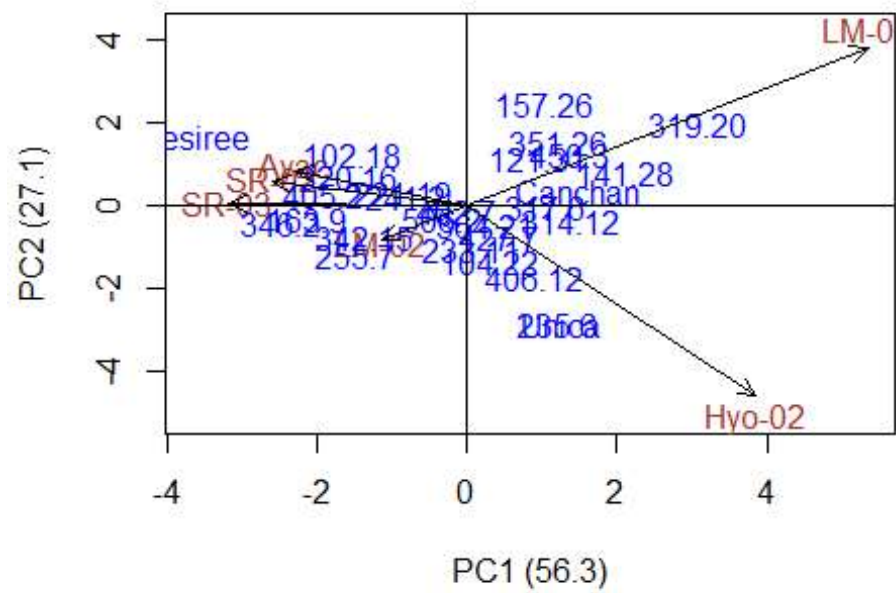
### AMMI Analysis

Since there are only five six locations in this yield trial. Using the regression based method to measure the yield stability for each genotype may not be very appropriate. AMMI approach could be a better way to determine the stability for each genotype.

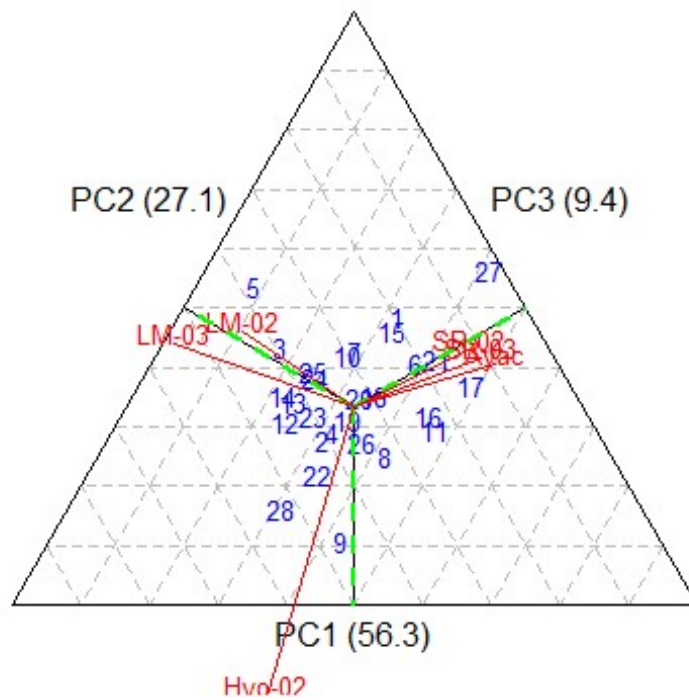
```
library(agricolae)
Example 1
model<- AMMI(dat$Locality, dat$Genotype, datRep, datYield, console=FALSE)
model$ANOVA
```

|           | Df  | Sum Sq | Mean Sq | F value | Pr(>F) |
|-----------|-----|--------|---------|---------|--------|
| ENV       | 5   | 122284 | 24456.9 | 257.04  | 0.000  |
| REP(ENV)  | 12  | 1142   | 95.1    | 2.57    | 0.003  |
| GEN       | 27  | 17533  | 649.4   | 17.54   | 0.000  |
| ENV:GEN   | 135 | 23762  | 176.0   | 4.75    | 0.000  |
| Residuals | 324 | 11998  | 37.0    | NA      | NA     |

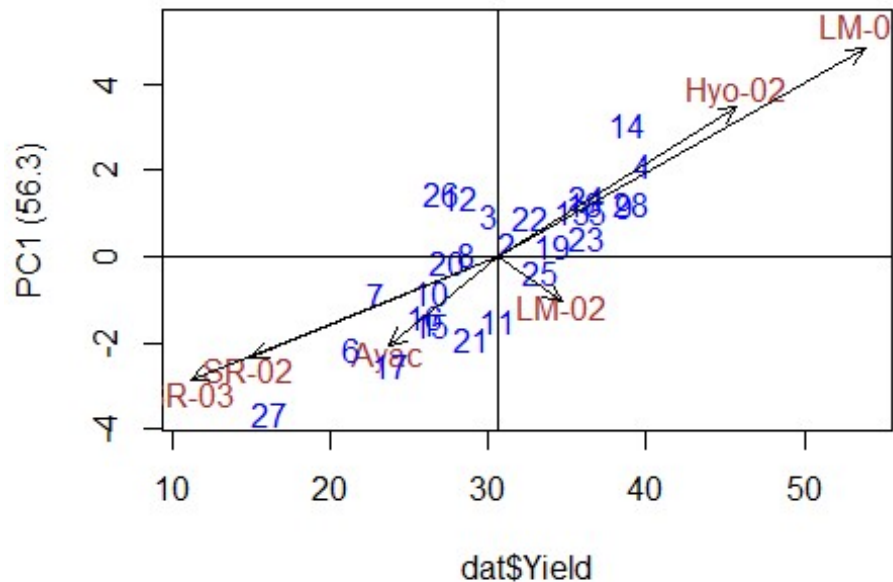
```
biplot
plot(model)
```



```
triplot PC 1,2,3
plot(model, type=2, number=TRUE)
```



```
biplot PC1 vs Yield
plot(model, first=0,second=1, number=TRUE)
```



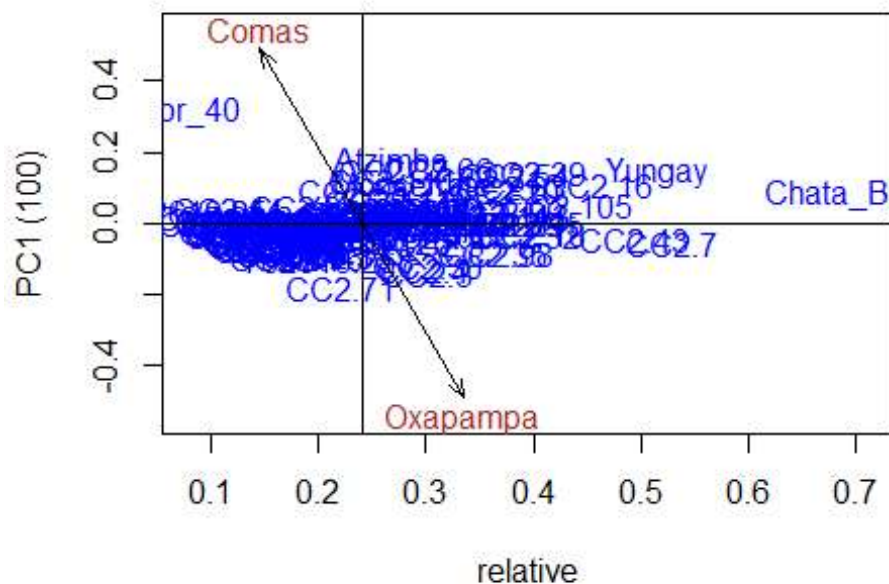
```
Example 2
data(CIC)
data1<-CIC$comas[,c(1,6,7,17,18)]
data2<-CIC$oxapampa[,c(1,6,7,19,20)]
cic <- rbind(data1,data2)
attach(cic)
model<-AMMI(Locality, Genotype, Rep, relative)

##
Warning. The analysis AMMI is not possible.
The number of environments and number of genotypes must be greater than 2

detach(cic)
model$ANOVA
```

|           | Df  | Sum Sq | Mean Sq | F value | Pr(>F) |
|-----------|-----|--------|---------|---------|--------|
| ENV       | 1   | 6.184  | 6.184   | 184.85  | 0      |
| REP(ENV)  | 4   | 0.134  | 0.033   | 8.47    | 0      |
| GEN       | 126 | 9.673  | 0.077   | 19.44   | 0      |
| ENV:GEN   | 108 | 1.348  | 0.012   | 3.16    | 0      |
| Residuals | 492 | 1.943  | 0.004   | NA      | NA     |

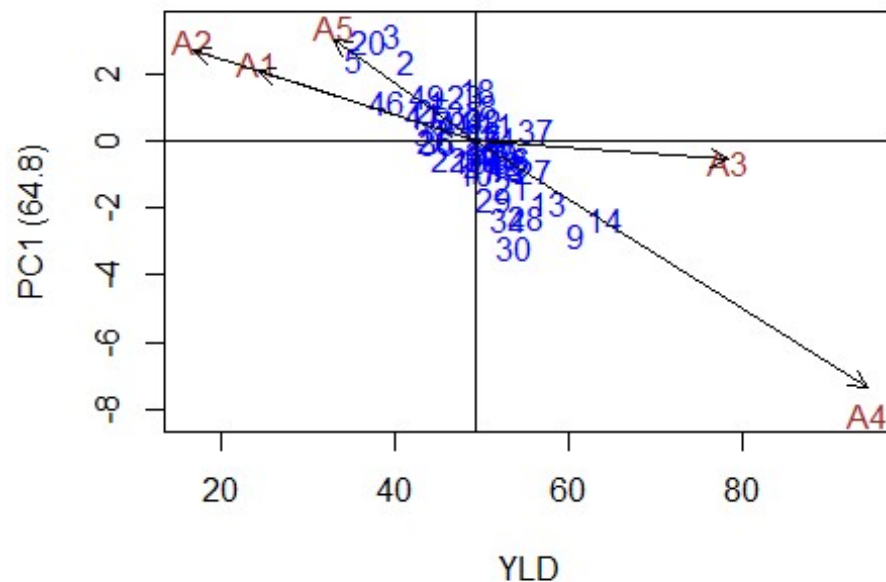
```
plot(model,0,1,angle=20,ecol="brown")
```



```
Example 3
Only means. Mean square error is well-known.
data(sinRepAmmi)
attach(sinRepAmmi)
REP <- 3
MSerror <- 93.24224
#startgraph
model<-AMMI(ENV, GEN, REP, YLD, MSerror,PC=TRUE)
detach(sinRepAmmi)
print anova
print(model$ANOVA,na.print = "")

Df Sum Sq Mean Sq F value Pr(>F)
ENV 4 728767 182191.6
REP(ENV) 10
GEN 49 23489 479.4 5.14 0e+00
ENV:GEN 196 40728 207.8 2.23 1e-12
Residuals 490 45689 93.2

Biplot with the one restored observed.
plot(model,0,1,type=1)
```



```
with principal components model$PC is class "princomp"
```

```
pc<- model$PC
```

```
pc$loadings
```

```
##
```

```
Loadings:
```

```
Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
```

```
A1 -0.243 0.122 0.811 0.262 -0.447
```

```
A2 -0.312 0.120 -0.824 -0.447
```

```
A3 -0.876 -0.131 0.107 -0.447
```

```
A4 0.846 0.287 -0.447
```

```
A5 -0.352 0.348 -0.563 0.489 -0.447
```

```
##
```

```
Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
```

```
SS loadings 1.0 1.0 1.0 1.0 1.0
```

```
Proportion Var 0.2 0.2 0.2 0.2 0.2
```

```
Cumulative Var 0.2 0.4 0.6 0.8 1.0
```

```
summary(pc)
```

```
Importance of components:
```

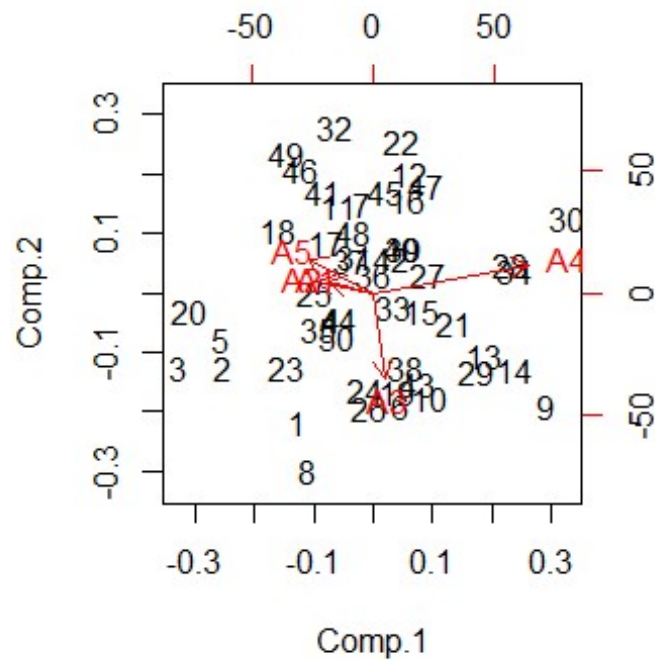
```
Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
```

```
Standard deviation 13.266 7.103 6.054 2.9056 1.83e-08
```

```
Proportion of Variance 0.648 0.186 0.135 0.0311 1.24e-18
```

```
Cumulative Proportion 0.648 0.834 0.969 1.0000 1.00e+00
```

```
biplot(pc)
```



```
Principal components by means of the covariance similar AMMI
It is to compare results with AMMI
cova<-cov(model$genXenv)
values<-eigen(cova)
total<-sum(values$values)
round(values$values*100/total,2)

[1] 64.81 18.58 13.50 3.11 0.00

AMMI: 64.81 18.58 13.50 3.11 0.00
```