diags: R Tools for Catch per Unit Effort Analysis

Laurence Kell
ICCAT

Abstract

The **diags** package provides methods for plotting and summarising Catch per Unit Effort (CPUE) data used in fitting fish stock assessment methods. Programs for stock assessment are generally implemented as standalone executable programs with their own text files for input and output files. **diags** provides a set of methods for reading these data and the results from fitting them nto R.

Keywords: R, CPUE, diagnsotics, residuals, stock assessment.

Contents

1. introduction

The **diags** package provides methods for plotting and summarising Catch per Unit Effort (CPUE) data used in fitting fish stock assessment methods. Programs for stock assessment are generally implemented as standalone executable programs with their own text files for input and output files. **diags** provides a set of methods for reading these data and the results from fitting them nto R.

The stock assessment package for which there are R methods for reading text files are

- ASPIC a biomass dynamic model fitted by maximising the likelihood
- BSP a Bayesian biomass dynamic model fitted using the SIR algorithm
- VPA Suite Imput file format mainly used by ICES for virtual population analysis
- VPA2Box An age structured model based on virtual population analysis
- Multifan-CL A statistical, length-based, age-structured model
- Stock Synthesis age and size structure assessment model

2. Data

The `readCpue` method reads data from the various stock assessment files into a commom data frame. There is an example data frame in the package

```
> library(diags)
> data(rsdl)
> head(rsdl)
```

```
   year         name     obs     hat    residual residualLag        qqx
12 1967 Japan LL II 0.25266 0.19459  0.2611499   0.2826152  0.6929003
13 1968 Japan LL II 0.25789 0.19440  0.2826152   0.3775346  0.7690553
14 1969 Japan LL II 0.27397 0.18782  0.3775346   0.1159142  1.5705850
15 1970 Japan LL II 0.20649 0.18389  0.1159142   0.1312879  0.2353289
16 1971 Japan LL II 0.20799 0.18240  0.1312879  -0.2008793  0.3572158
17 1972 Japan LL II 0.14379 0.17578 -0.2008793  -0.3022600 -0.6205683
         qqy       qqHat
12  0.2611499  0.2675000
13  0.2826152  0.2941443
14  0.3775346  0.5745746
```

```
15  0.1159142  0.1074100
16  0.1312879  0.1500545
17 -0.2008793 -0.1920418
```

The columns identify the observations (`year,name` and may include other covariates such as age, season, etc.), the original observations (`obs`) and the fitted values and the residuals (`obs,hat`) if `diags` has been used to read in the data, the residuals with a lag of 1 (`residualLag`) and the quanitiles (`qqx,qqy,qqHat`) assumming a normal distribution.

In some assessment packages the data are in a specific file in other cases the data are in a suite of files found in a dir. Therefore the `readCpue` takes either a file or a dir as irs first arguemnt depending on the assessment method e.g. reading in from vpa2box and SS

```
> u2box=readCpue("unisex09.c01","2box")
> uSS  =readCpue("myDir","ss")
```

`readCpue` only reads in the data as input to a stock assessment, `diags` reads the residuals and and covariates as well.

```
> u2box=readCpue("unisex09.c01","2box")
> uSS  =readCpue("myDir","ss")
```

There are also some methods for writing the various input files-

3. Transformations

```
> library(gam)
> gm  =gam(log(obs)~lo(year)+name,data=rsdl)
> rsdl=data.frame(rsdl,gam=predict(gm),gamRsdl=residuals(gm))
> scl =coefficients(gm)[3:9]
> names(scl)=substr(names(scl),5,nchar(names(scl)))
> rsdl=transform(rsdl,scl=scl[as.character(name)])
> rsdl[is.na(rsdl$scl),"scl"]=0
```

4. Plotting

Plotting is done using **ggplot2**, for more details see. A few basic tricks are used i.e.

transform

layers

geoms **geom_point**

geom_line

geom_smooth

facet

theme

5. Exploratory Data Analysis

Time Series Plot the CPUE time series.

Correlations between indices Plot the indices against each other and then

as do indices 4, 6 and 8.

```
> ggplot(rsdl)+ geom_line(aes(year,exp(gam)),col="red")  +
+               geom_smooth(aes(year,obs),se=FALSE)      +
+               geom_point(aes(year,obs,col=name))       +
+               facet_wrap(~name,ncol=1,scale="free_y")  +
+               theme_ms(legend.position="none")         +
+               xlab("Year") + ylab("Index")
```

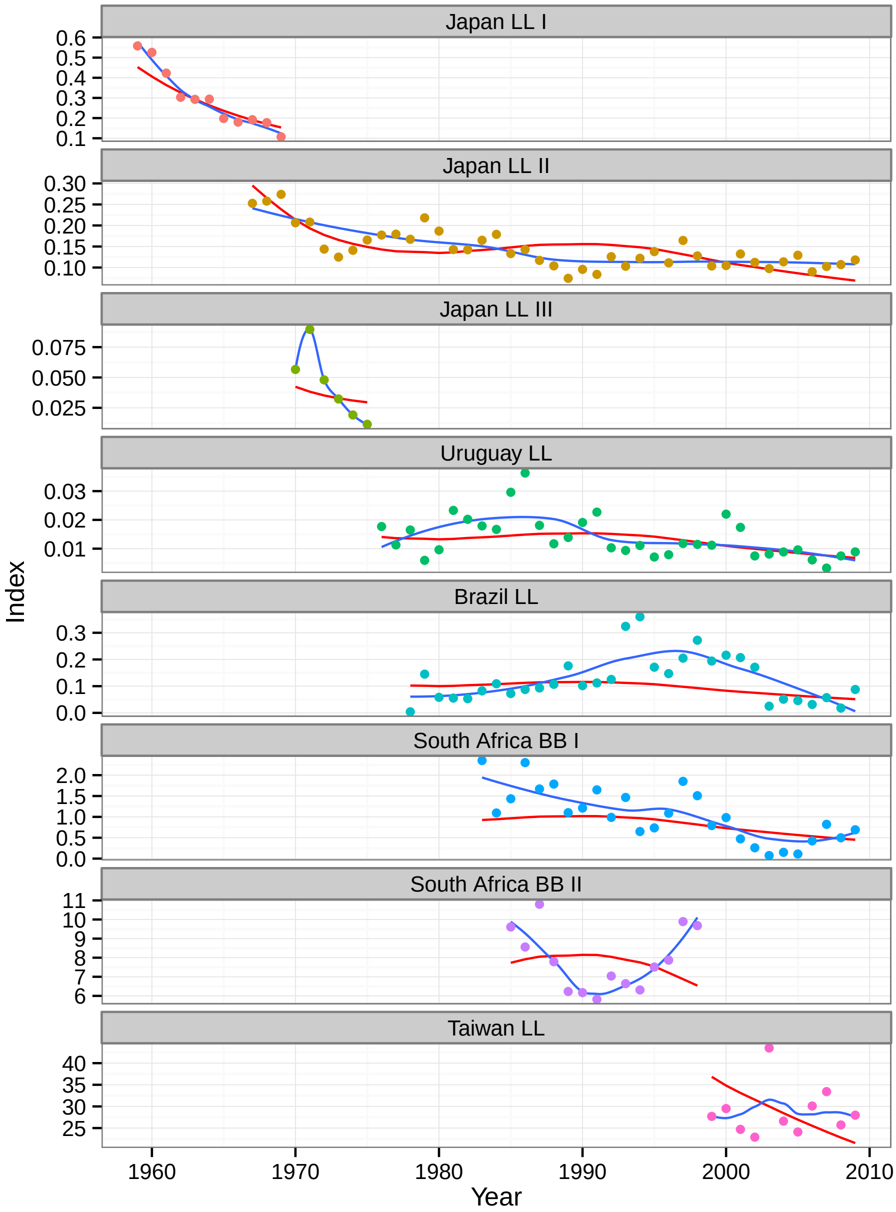

Figure 1: **Plot of indices of abundance, points are the observed index values and the blue a lowes fit to the points by index. The red line is GAM fitted to lo(year) and fleet.**

```
> uMat=ddply(rsdl,.(name),transform, obs=stdz(obs))
> uMat=cast(uMat,year~name,value="obs")
> uMat=uMat[apply(uMat,1,function(x) !all(is.na(x))),]
> pM=plotmatrix(uMat[,-c(1:2,4)])
> pM$layers[[2]]=NULL
> mns=ddply(subset(pM$data,!(is.na(x) & !is.na(y))),.(xvar,yvar), function(x) mean(x$y,na.
> pM+geom_hline(aes(yintercept=V1),data=mns,col="red") +
+     geom_smooth(method="lm",fill="blue", alpha=0.1)   +
+     theme(legend.position="bottom")                    +
+     xlab("Index")+ylab("Index")                        +
+     theme_ms()
```

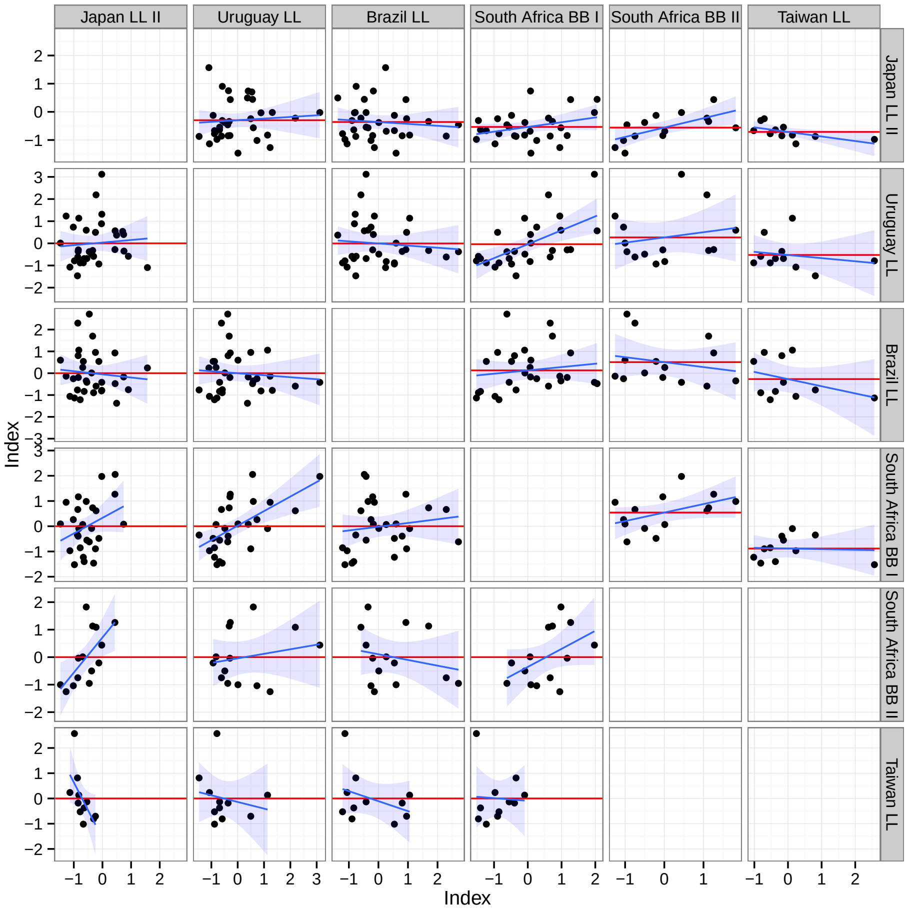

Figure 2: **Pairwise scatter plots of the indices of abundance, blue lines are linear regressions fitted to the points, the shade area is the standard error of predicted means and the red line is the mean of the points on the y-axis.**

```
> cr=cor(uMat[,-1],use="pairwise.complete.obs")
> dimnames(cr)=list(gsub("_"," ",names(uMat)[-1]),gsub("_"," ",names(uMat)[-1]))
> cr[is.na(cr)]=0
> corrplot(cr,diag=F,order="hclust",addrect=2)  +
+                theme(legend.position="bottom")  +
+                theme_ms()
```

```
NULL
```

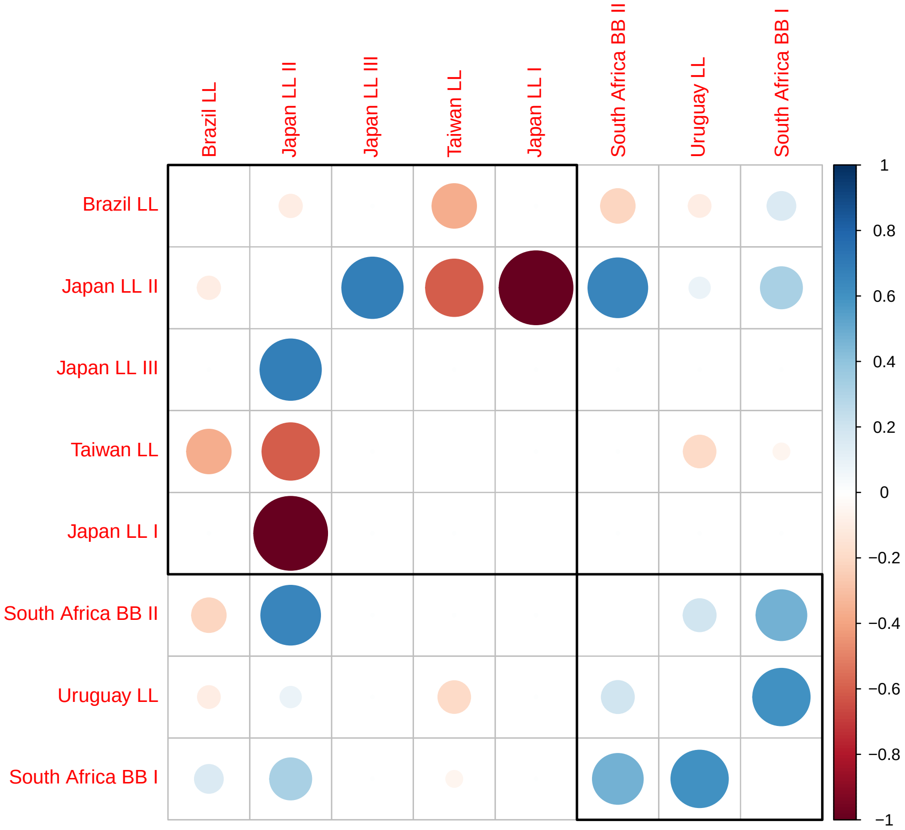

Figure 3: **A plot of the correlation matrix for the indices, blue indicate a positive correlation and red negative. the order of the indices and the rectanglur boxes are chosen based on a hierarchical cluster analysis using a set of dissimilarities for the indices being clustered.**

```
> ggplot(subset(rsdl,name %in% c("Japan LL II","Japan LL III","South Africa BB II")))+
+             geom_point(aes(year,exp((gam+gamRsdl)-scl),col=name))+
+             geom_smooth(aes(year,exp(gam+gamRsdl-scl),group=name,col=name),se=T,fill="b
+             theme_ms(legend.position="bottom") +
+             xlab("Year") +ylab("Index")
```

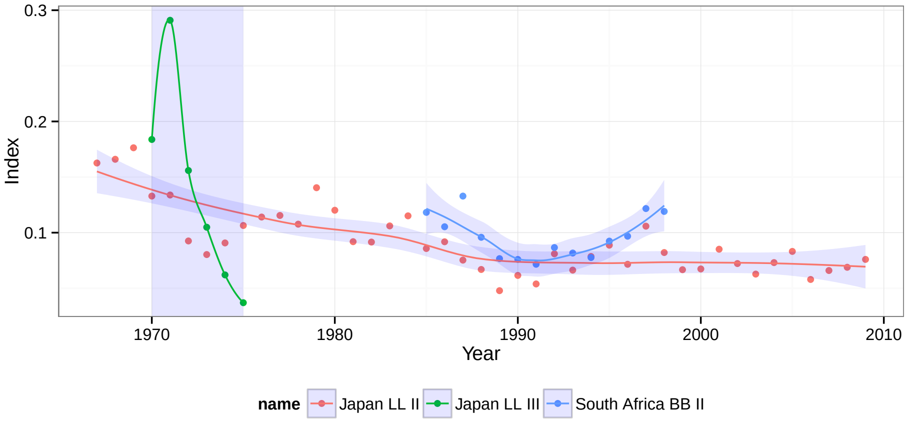

Figure 4: **Plots of Japan LL II, Japan LL III and South Africa BB II indices, lowess regressions and SEs are shown for each series.**

Figure 5: **Plots of South Africa BB I, South Africa BB II and Uruguay LL indices, lowess regressions and SEs are shown for each series.**

```
>     ggplot(subset(rsdl,name %in% c("South Africa BB I","South Africa BB II","Uruguay LL"
+             geom_point(aes(year,exp((gam+gamRsdl)-scl),col=name))+
+              geom_smooth(aes(year,exp(gam+gamRsdl-scl),group=name,col=name),se=T,fill="b
+              theme_ms(legend.position="bottom")  +
+              xlab("Year") +ylab("Index")
```

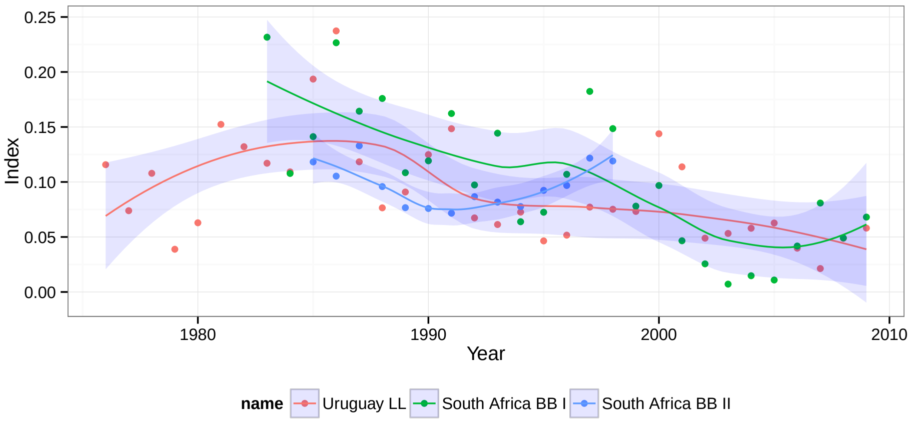

Figure 6: **Observed CPUE verses fitted, blue line is a linear resgression fitted to points, black the y=x line.**

6. Residual Analysis

Affiliation:

Laurence Kell
ICCAT Secretariat
C/Corazón de María, 8.
28002 Madrid
Spain

E-mail: Laurie.Kell@iccat.int

```
> dat=ddply(rsdl, .(name), with, data.frame(obs=stdz(obs),hat=stdz(hat)))
> ggplot(dat) +
+           geom_abline(aes(0,1))                              +
+           geom_point( aes(obs,hat))                          +
+           stat_smooth(aes(obs,hat),method="lm",fill="blue", alpha=0.1)      +
+           facet_wrap(~name,ncol=3)                           +
+           theme_ms(legend.position="bottom")                 +
+           xlab("Fitted") + ylab("Observed")
```

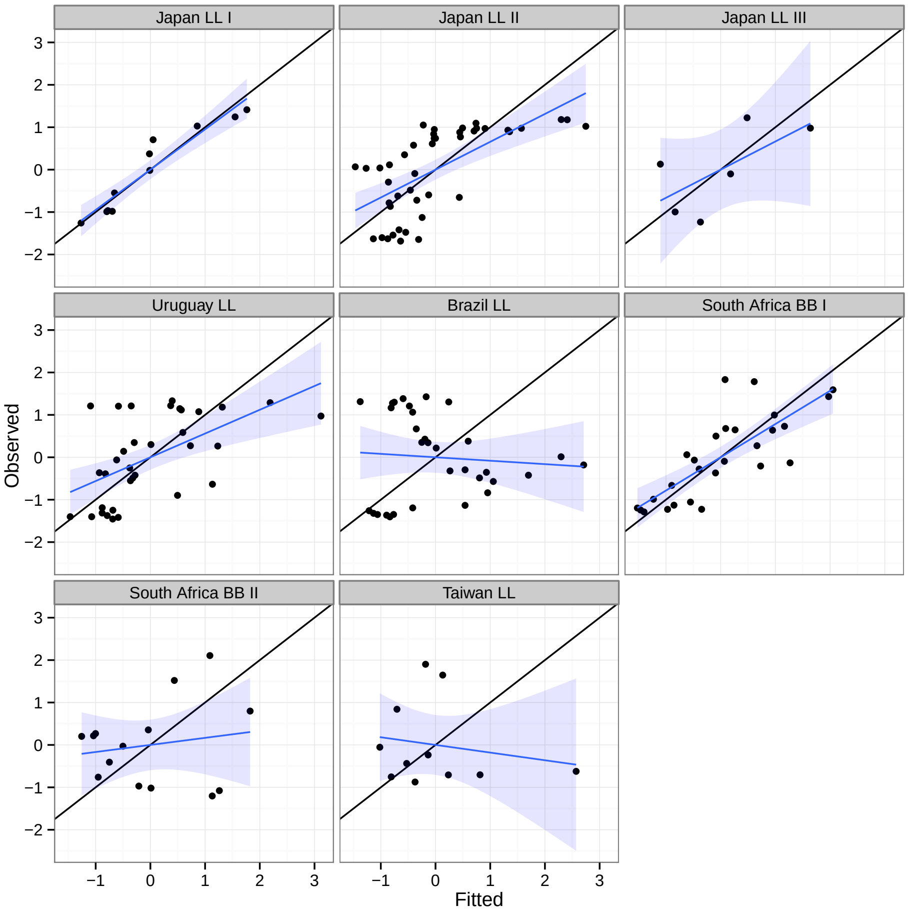

Figure 7: **Observed CPUE verses fitted, blue line is a linear resgression fitted to points, black the y=x line.**

```
> dat=ddply(rsdl, .(name), transform, residual=stdz(residual,na.rm=T))
> ggplot(aes(year,residual),data=dat) +
+   geom_hline(aes(yintercept=0))      +
+   geom_point()                       +
+   stat_smooth(,method="loess",se=T,fill="blue", alpha=0.1)  +
+   facet_wrap(~name,scale="free",ncol=2)    +
+              theme_ms()
```

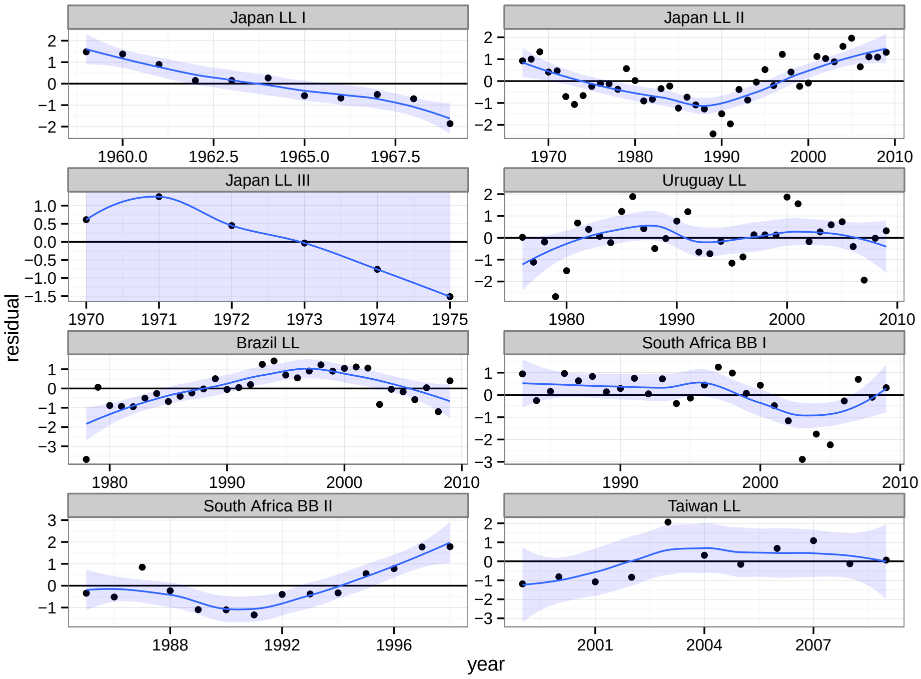

Figure 8: **Residuals by year, with lowess smoother and SEs.**

```
> ggplot(rsdl)                                                              +
+   geom_point( aes(residual,residualLag))                                  +
+   stat_smooth(aes(residual,residualLag),method="lm",se=T,fill="blue", alpha=0.1)       +
+   geom_hline(aes(yintercept=0))                                           +
+   facet_wrap(~name,scale="free",ncol=3)                                   +
+   xlab(expression(Residual[t])) +
+   ylab(expression(Residual[t+1])) +
+   theme_ms(legend.position="bottom")
```

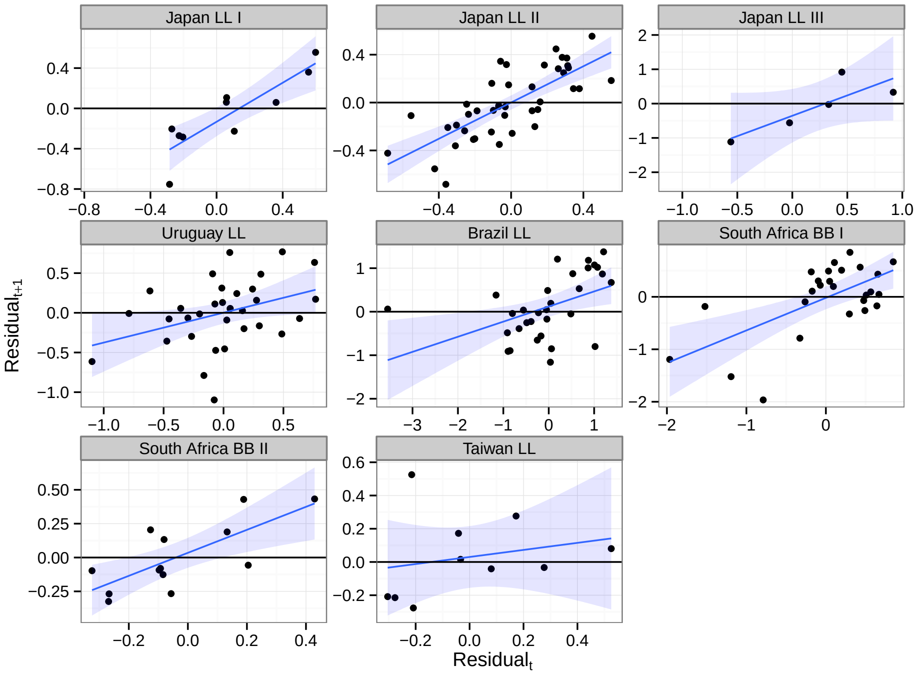

Figure 9: **Plot of autocorrelation, i.e.** $residual_{t+1}$ **verses** $residual_t$**.**

```
> ggplot(rsdl)                                              +
+   geom_point( aes(qqx,qqy))                               +
+   stat_smooth(aes(qqx,qqHat),method="lm",se=T,fill="blue", alpha=0.1)       +
+   facet_wrap(~name)                                       +
+   theme_ms(legend.position="bottom")                      +
+               theme_ms()
```

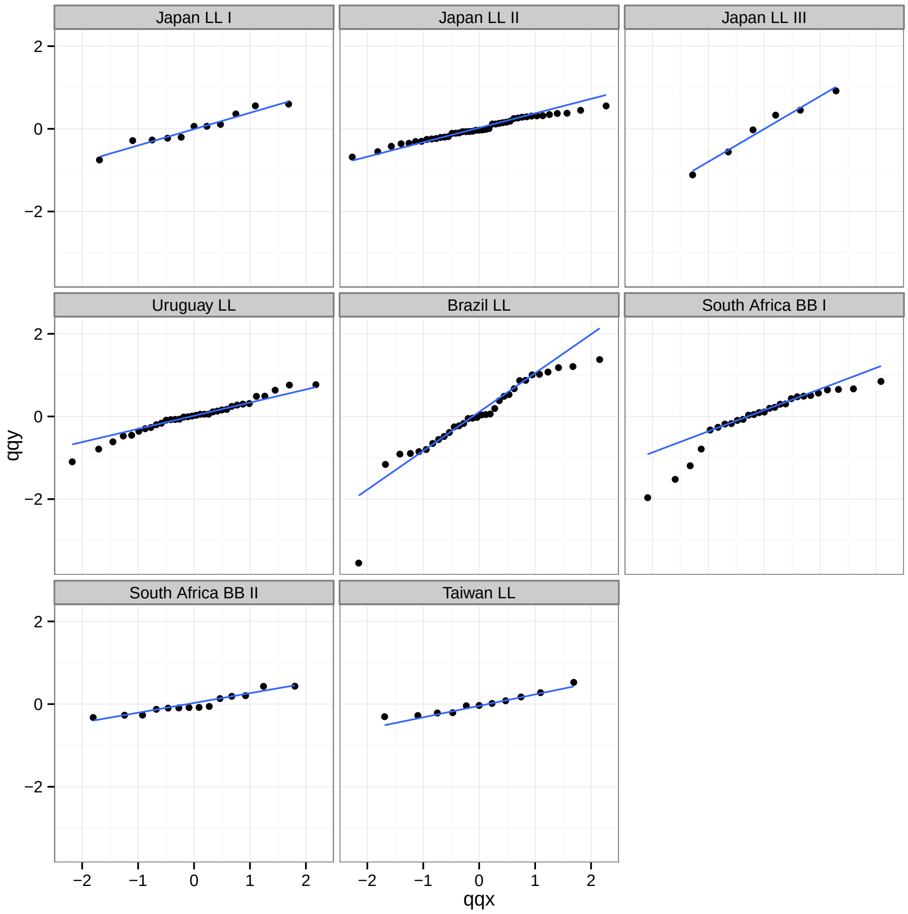

Figure 10: **Quantile-quantile plot to compare residual distribution with the normal distribution.**

```
> ggplot(aes(hat, residual),data=rsdl)    +
+   geom_hline(aes(yintercept=0))          +
+   geom_point()                           +
+   stat_smooth(method="loess",span=.9,fill="blue", alpha=0.1)    +
+   facet_wrap(~name,scale="free",ncol=3) +
+               theme_ms()
```

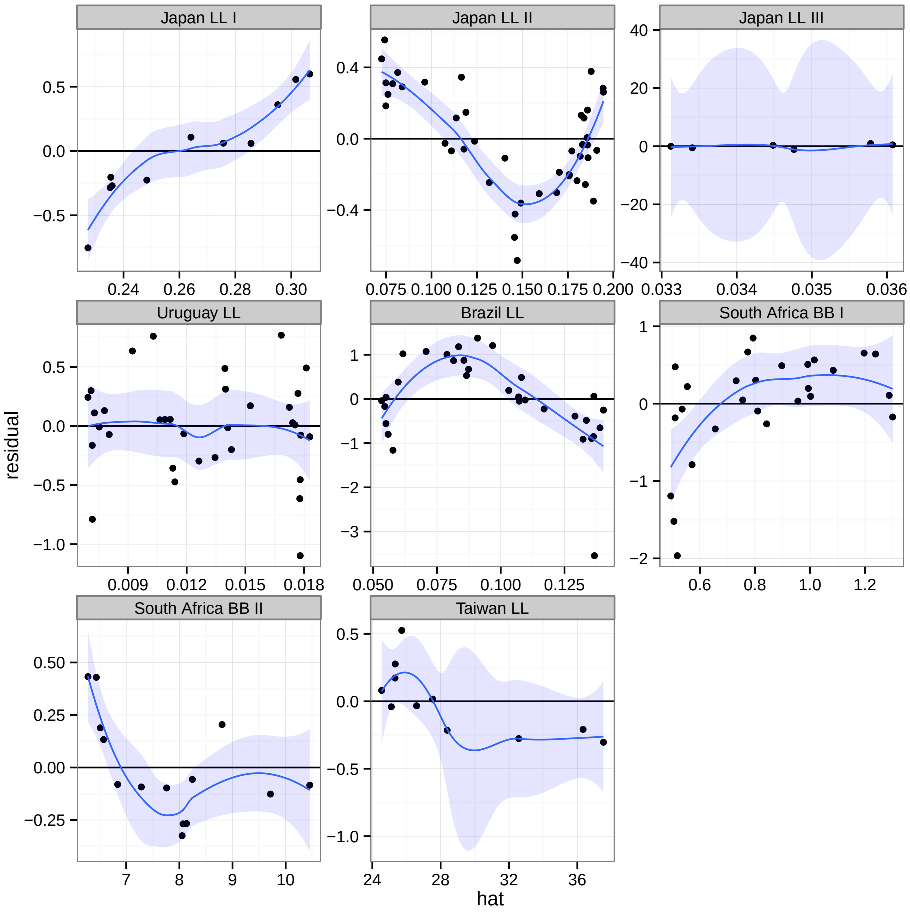

Figure 11: **Plot of residuals against fitted value, to check variance relationship.**