Regression Diagnostics Using R

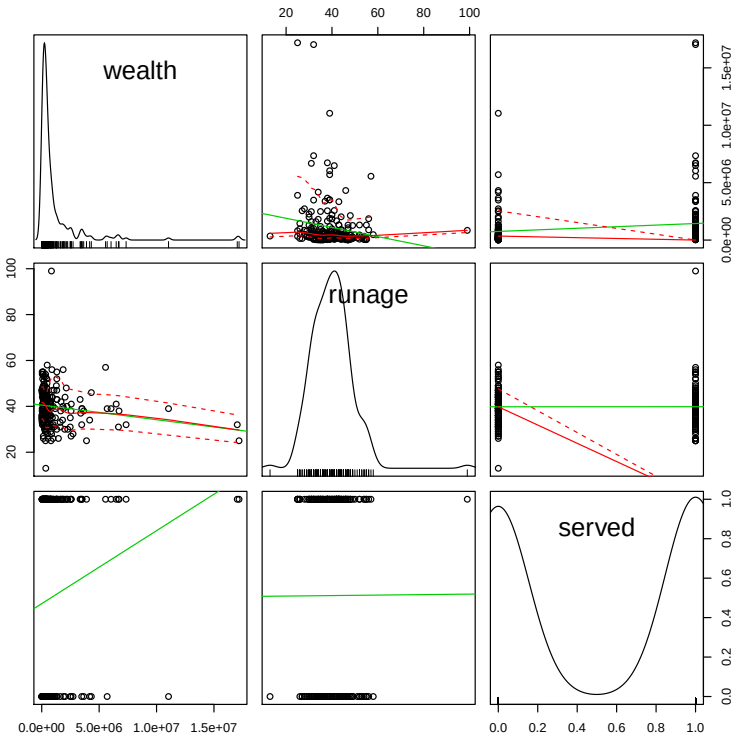
Yiqing Xu

Based on Jens Hainmueller's MIT Lecture 17800.10-1/2
Data used in this handout is based on Eggers-Hainmueller (2009)

1 Overview

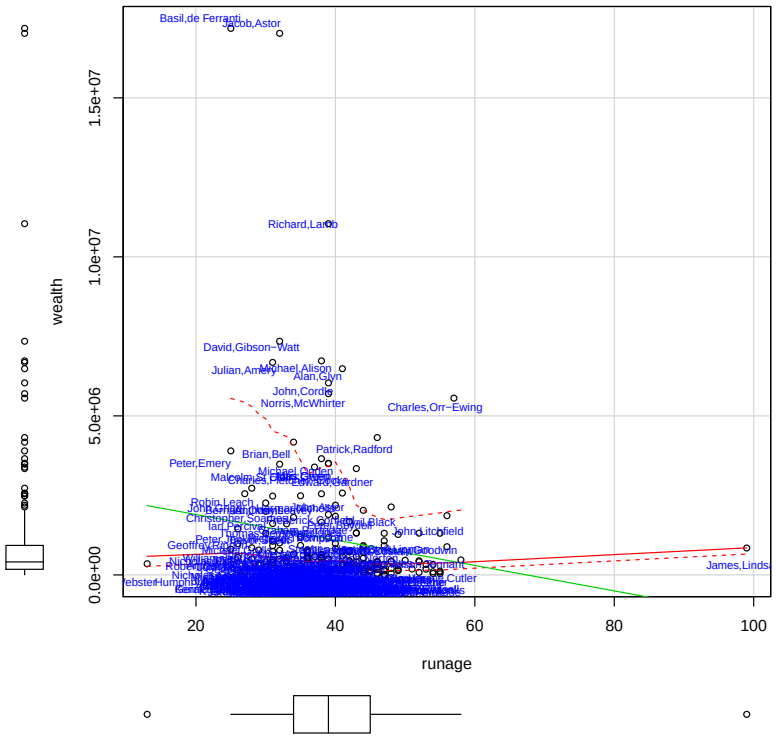
1. Scatterplot: `car` and `lattice`
2. Regression Diagnostics
   - Hat-values: identifying leverage points
   - Studentized residuals: identifying outliers
   - QQ plot: evaluating model fit and normality
   - DFBetas: evaluating influence for each coefficient
   - Cook's distance: summarizing influence across coefficients
   - Automatic regression diagnosis
3. Standard Error Adjustment
   - Breusch-Pagan test
   - Robust standard errors

2 Scatterplot



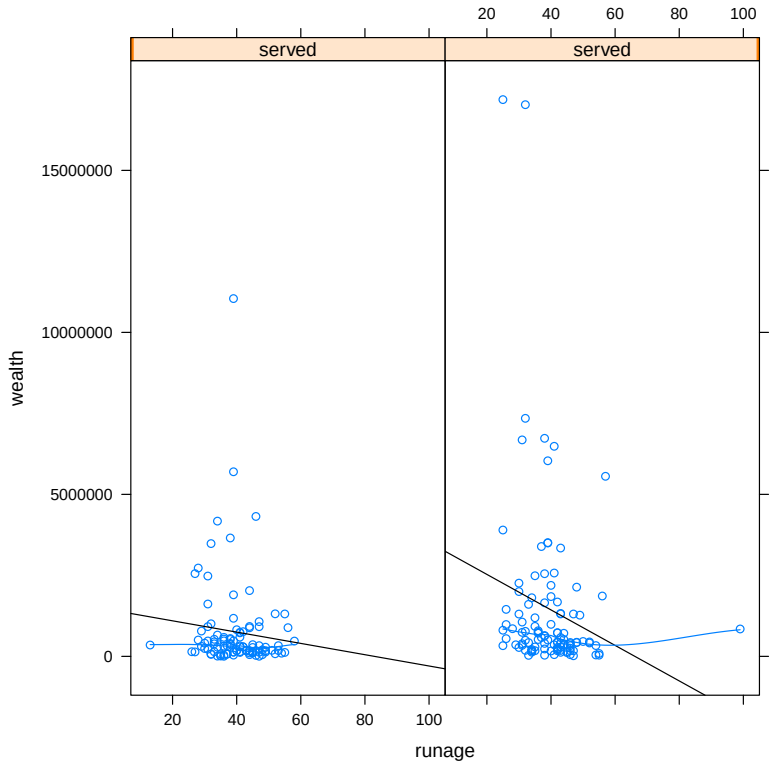
R Code

```
library(foreign)
d<-read.dta("bp.dta") # loading the data
library(car)
scatterplotMatrix(d[,c("wealth","runage","served")])
```



R Code

```
scatterplot(d$wealth~d$runage,xlab="runage",ylab="wealth")
text(y=d$wealth,x=d$runage,labels=d$name,pos=3,cex=.6,col=4)
```



R Code

```
library(lattice)
mypanel<-function(x,y,...) {
  panel.xyplot(x,y,...)
  panel.lmline(x,y)
  panel.loess(x,y)
}
xyplot(wealth~runage|served,data=d,panel=mypanel)
```

3 Regression Diagnostic

3.1 Testing Two Models

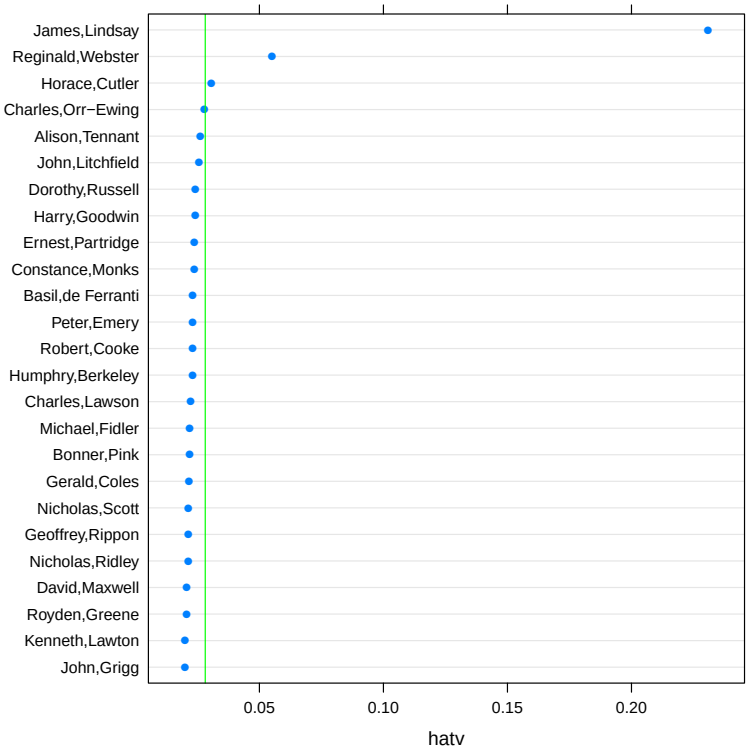
```
------------------------------------------------------------
t test of coefficients:
            Estimate Std. Error t value  Pr(>|t|)
(Intercept)  2349006     702638  3.3431 0.0009808 ***
runage        -40120      16862 -2.3794 0.0182380 *
served        691895     290422  2.3824 0.0180933 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
------------------------------------------------------------


------------------------------------------------------------
t test of coefficients:
             Estimate Std. Error t value  Pr(>|t|)
(Intercept) 13.599589   0.451991 30.0882 < 2.2e-16 ***
runage      -0.025181   0.010847 -2.3215 0.0212179 *
served       0.698336   0.186822  3.7380 0.0002392 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
------------------------------------------------------------
```

R Code

```
library(lmtest)
d<-na.omit(d)
mod1<-lm(wealth~runage+served,data=d)
mod2<-lm(log(wealth)~runage+served,data=d)
coeftest(mod1)
coeftest(mod2)
```

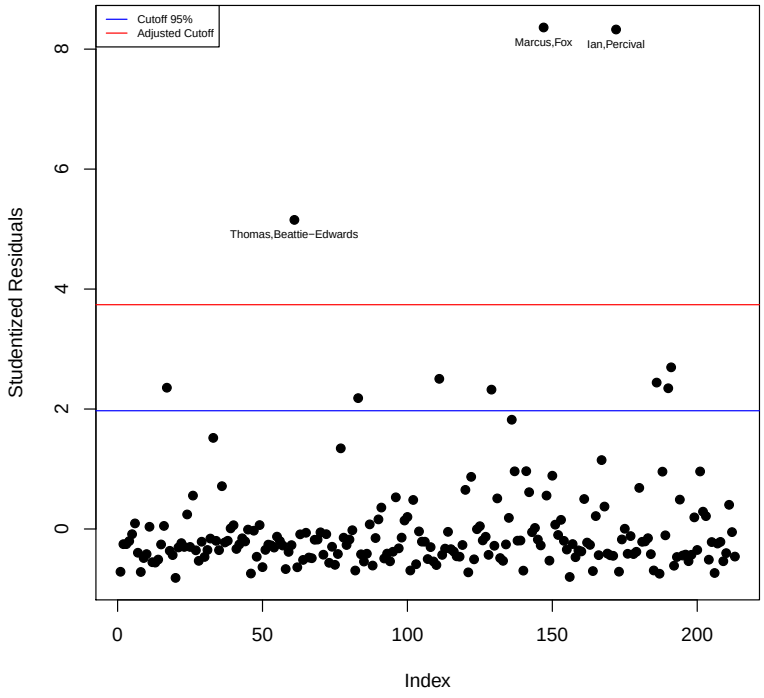
3.2 Hat-values



R Code

```
d$hatv <- hatvalues(mod1)
d <- d[order(d$hatv),]
d$name <- factor(d$name,levels=d$name,ordered=T)
n <- mod1$df.residual + mod1$rank  # num of obs
k <- mod1$rank                     # num of regressors
cutoffhatv <- 2*k/n
mypanel = function(x,y,...){
  panel.dotplot(x,y,...)
  panel.abline(v=cutoffhatv,col="green")
}
dotplot(name~hatv,data=d[d$hatv>.02,],panel=mypanel)
```

3.3 Studentized Residuals

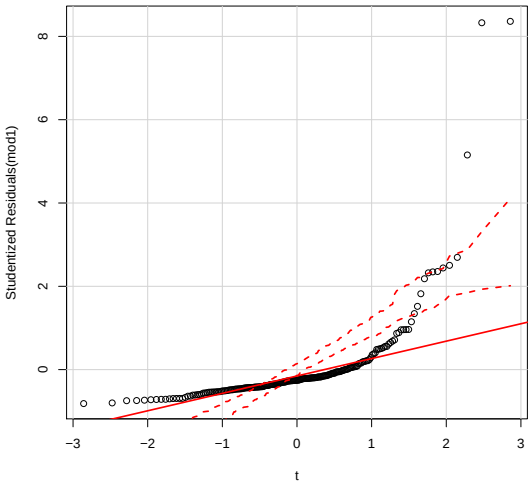


R Code

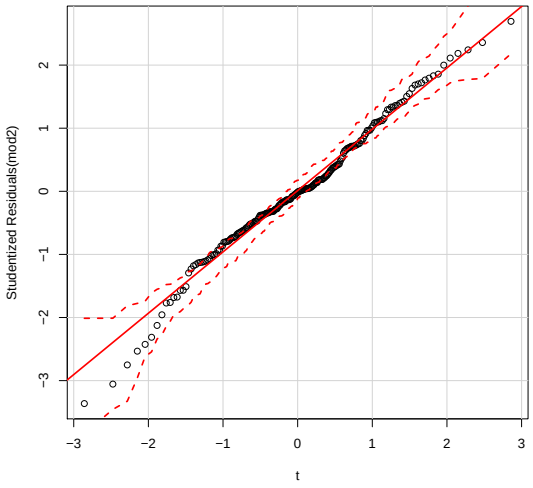
```
d$studresid <- rstudent(mod1)
cutoffstud    <- qt(.025, n-k, lower.tail=F)
cutoffstudadj <- qt(.025/(n-k), n-k, lower.tail=F)

plot(d$studresid, ylab="Studentized Residuals", pch=19)
abline(h=cutoffstud, col="blue")
abline(h=cutoffstudadj, col="red")
legend("topleft", legend=c("Cutoff 95%","Adjusted Cutoff"),
       lty=1, col=c("blue","red"), cex=.6)
text(y=d$studresid[d$studresid>cutoffstudadj],
     x=(1:length(mod1$fitted.values))[d$studresid>cutoffstudadj],
     label=d$name[d$studresid>cutoffstudadj], pos=1, cex=.6)
graphics.off()
```

3.4 Q-Q Plot



(a) Model 1: Raw data

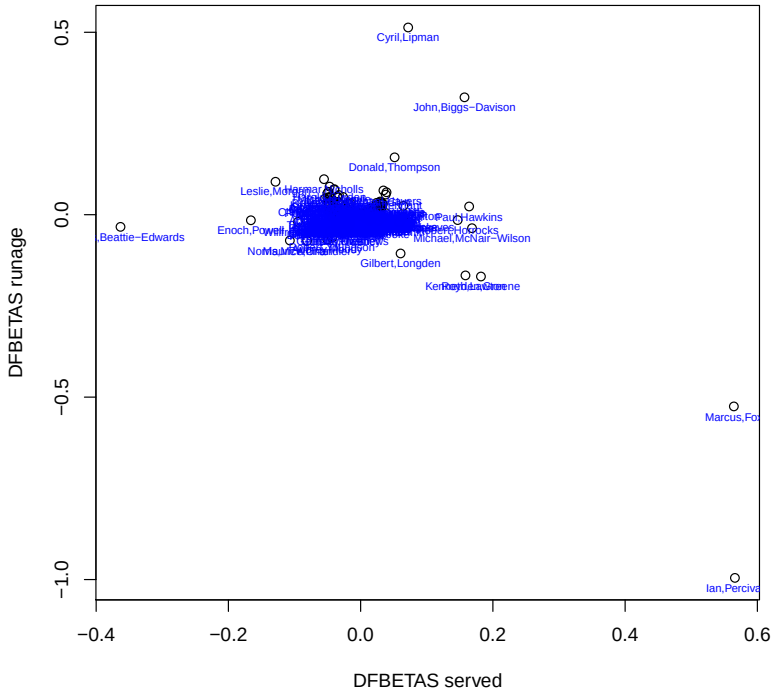


(b) Model 2: After log transformation

R Code

```
qqPlot(mod1,"t",envelope=TRUE)
qqPlot(mod2,"t",envelope=TRUE)
```

3.5 DFBetas

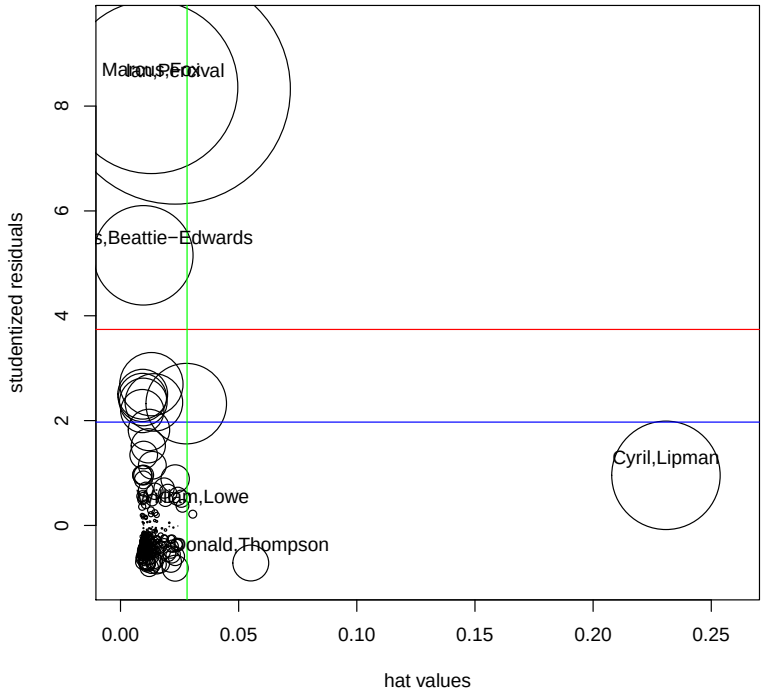


R Code

```
dfbetas <- dfbetas(mod1)
2/sqrt(n)

plot(dfbetas[,3],dfbetas[,2],xlab="DFBETAS served",ylab="DFBETAS runage")
text(dfbetas[,3],dfbetas[,2],label=d$name,post=1,cex=.6,col=4)
```

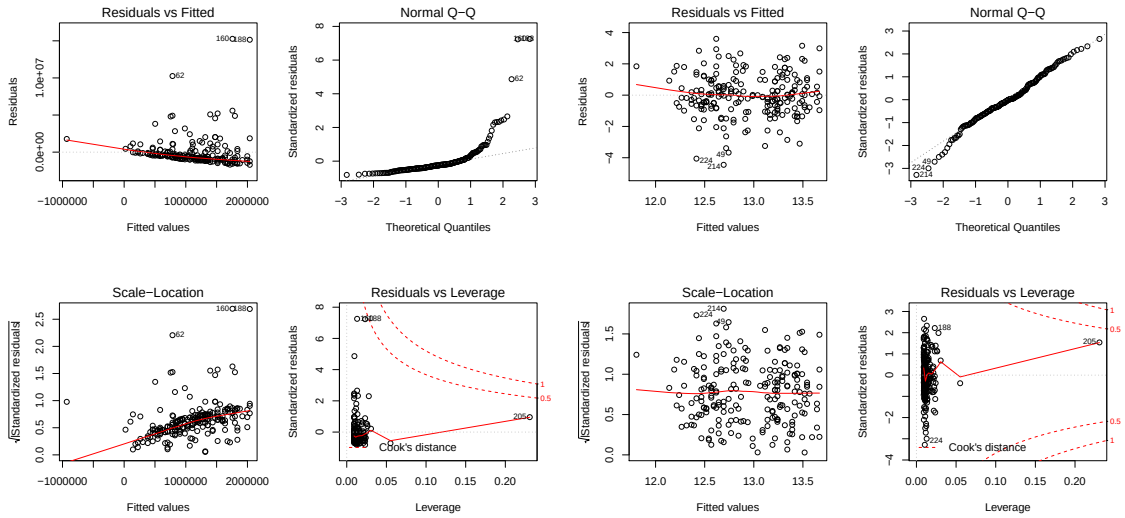
3.6 Influence Plots



R Code

```
symbols(y=rstudent(mod1), x=hatvalues(mod1), circles=sqrt(cookd(mod1)),
        ylab="studentized residuals",xlab="hat values",
        ylim=c(-1,9.5),xlim=c(0,.26))
abline(h=cutoffstud,col="blue")
abline(h=cutoffstudadj,col="red")
abline(v=cutoffhatv,col="green")
filter <- rstudent(mod1) > cutoffstudadj |  hatvalues(mod1) > cutoffhatv
text(y=rstudent(mod1)[filter], x=hatvalues(mod1)[filter], label=d$name[filter],pos=3)
```

3.7 Automatic Regression Diagnostics



(c) Model 1: Raw data

(d) Model 2: After log transformation

R Code

```
par(mfrow=c(2,2))
plot(mod1)
par(mfrow=c(2,2))
plot(mod2)
```

4 SE Adjustment

4.1 Breusch-Pagan Test

```
------------------------------------------------------------
Breusch-Pagan test

data:  mod1
BP = 83.4522, df = 2, p-value < 2.2e-16
------------------------------------------------------------
```

R Code

```
library(lmtest)
bptest(mod1, studentize = FALSE)
```

4.2 Robust SE

```
------------------------------------------------------------
t test of coefficients:
            Estimate Std. Error t value  Pr(>|t|)
(Intercept)  2349006     702638  3.3431 0.0009808 ***
runage        -40120      16862 -2.3794 0.0182380 *
served        691895     290422  2.3824 0.0180933 *
------------------------------------------------------------
t test of coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2349006     813865  2.8862 0.004306 **
runage        -40120      19900 -2.0161 0.045065 *
served        691895     285120  2.4267 0.016081 *
------------------------------------------------------------
t test of coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2349006     819658  2.8658 0.004582 **
runage        -40120      20042 -2.0018 0.046589 *
served        691895     287149  2.4095 0.016835 *
------------------------------------------------------------
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

R Code

```
library(sandwich)
library(lmtest)
coeftest(mod1)                                # homoskedasticity
coeftest(mod1,vcov=vcovHC(mod1,type="HC0")) # classic White
coeftest(mod1,vcov=vcovHC(mod1,type="HC1")) # small sample correction
```