Introduction to R

Statistical analysis

Arni Magnusson

Hafro, 8 Nov 2010

Outline

1 Descriptive statistics

Statistical functions

```
rivers

min(rivers)
max(rivers)
range(rivers)
quantile(rivers)

sum(rivers)
mean(rivers)
median(rivers)

sd(rivers)
var(rivers)
```

```
mtcars

cor(mtcars$hp, mtcars$disp)
cor(mtcars)
```

Loess smoother

```
plot(dist ~ speed, data=cars)

lofit <- loess(dist ~ speed, data=cars)$fit

lines(cars$speed, lofit, lwd=2, col="red")
```

2 Significance tests

t.test

```
t.test(x1, x2)

?t.test
```

chisq.test

```
chisq.test(obs, exp)

?chisq.test
```

3 Linear models

Linear regression

```
lm(formula, data)

lm(y ~ x)

lm(y ~ x1+x2)

lm(dist ~ speed, data=cars)

?lm
```

Formula syntax

`~`	is a function of	`y ~ x`
`+`	and	`y ~ x1 + x2`
`:`	interaction term	`y ~ x1 + x2 + x1:x2`
`I`	do not interpret	`y ~ x1 + I(x2+x3)`
`*`	both terms and their interaction	`y ~ x1 * x2`
`-`	but not this term	`y ~ x1 * x2 - x2`
`.`	all terms, or update	`y ~ . + x3`

Fixing the intercept or slope

`lm(y ~ 1)`	estimate intercept only, null model
`lm(y ~ -1 + x)`	estimate slope, fix intercept at 0
`lm(offset(y-3) ~ -1 + x)`	estimate slope, fix intercept at 3
`lm(y ~ offset(3*x))`	estimate intercept, fix slope at 3

`?formula`

aov

`aov(formula, data)`

`?aov`

glm

`glm(formula, data, family, link)`

`?glm`
`?family`

- gaussian
- binomial
- poisson

  ...

Modelling tools

`coef(model)`	coefficient
`predict(model)`	predictions
`fitted(model)`	fitted values
`residuals(model)`	residuals
`summary(model)`	estimates, SE, p values, R^2
`anova(model)`	p values
`AIC(model)`	AIC value
`update(model, formula)`	modify
`add1(model, candidates)`	add one term
`drop1(model, candidates)`	drop one term
`step(model, candidates)`	add and drop iteratively

4 Examples

Chick weights (t.test)

```
chick2 <- split(chickwts$weight,
                chickwts$feed)[c("linseed", "soybean")]
chick2

boxplot(chick2)
```

```
t.test(chick2$linseed, chick2$soybean)
```

- Assume equal variance in both groups? `var.equal=T`
- Don't use functions like black box; do once by hand if possible

Plant growth (aov)

```
PlantGrowth

boxplot(weight ~ group, data=PlantGrowth)
```

```
aov(weight ~ group, data=PlantGrowth)

summary(aov(weight ~ group, data=PlantGrowth))
```

Car stopping distance (simple lm)

```
cars

head(cars)

plot(dist ~ speed, data=cars)
```

```
mylm <- lm(dist ~ speed, data=cars)
abline(mylm)
summary(mylm)

par(mfrow=c(2,2))
plot(mylm)
```

Car stopping distance (simple lm)

Try log-log transformation

```
par(mfrow=c(1,1))
plot(log(dist) ~ log(speed), data=cars)
```

```
mylog <- lm(log(dist) ~ log(speed), data=cars)
abline(mylog)
summary(mylog)
```

Car stopping distance (simple lm)

Model comparison: visualize fit

```
plot(dist ~ speed, data=cars, main="normal")
abline(mylm)

dev.new()

plot(log(dist) ~ log(speed), data=cars,
     main="log-log")
abline(mylog)
```

Car stopping distance (simple lm)

Model comparison: diagnostic plots

```
par(mfrow=c(2,2))
plot(mylm, main="normal")

dev.new()

par(mfrow=c(2,2))
plot(mylog, main="log-log")
```

Car stopping distance (simple lm)

Model comparison: R^2 and AIC

```
summary(mylm)
summary(mylog)

names(summary(mylm))

summary(mylm)$r.s
summary(mylog)$r.s
```

```
AIC(mylm, mylog)
```

Tooth growth (ancova lm)

```
ToothGrowth

head(ToothGrowth)

summary(ToothGrowth)
```

```
boxplot(len ~ supp, data=ToothGrowth)

plot(len ~ dose, data=ToothGrowth)

plot(len ~ log(dose), data=ToothGrowth)
```

Tooth growth (ancova lm)

```
library(lattice)
xyplot(len ~ log(dose)|supp, data=ToothGrowth,
       panel=function(...){panel.xyplot(...);
       panel.lmline(...)})
```

Same line, different intercept, different slope, or both different

```
lm(len ~ log(dose), data=ToothGrowth) # coefs 2

lm(len ~ log(dose)+supp, data=ToothGrowth)  # 3

lm(len ~ log(dose):supp, data=ToothGrowth)  # 3

lm(len ~ log(dose)*supp, data=ToothGrowth)  # 4
```

Tooth growth (ancova lm)

Forward selection

```
add1(lm(len ~ 1, data=ToothGrowth),
     . ~ log(dose)*supp, test="F")


add1(lm(len ~ log(dose), data=ToothGrowth),
     . ~ log(dose)*supp, test="F")


add1(lm(len ~ log(dose)+supp, data=ToothGrowth),
     . ~ log(dose)*supp, test="F")
```

Tooth growth (ancova lm)

Backward selection

```
drop1(lm(len ~ log(dose)*supp,
         data=ToothGrowth), test="F")


anova(lm(len ~ log(dose)*supp,
         data=ToothGrowth))
```

Tooth growth (ancova lm)

Plot model predictions

```
mylm <- lm(len ~ log(dose)*supp,
           data=ToothGrowth)


plot(len ~ log(dose), data=ToothGrowth,
     subset=supp=="OJ", ylim=c(0,35),
     pch=16, col="orange")


points(len ~ log(dose), data=ToothGrowth,
       subset=supp=="VC", pch=16, col="blue")
```

Tooth growth (ancova lm)

Plot model predictions

```
d <- c(0.5, 1, 2)

ojfit <- predict(mylm,
                 data.frame(dose=d, supp=factor("OJ")))

vcfit <- predict(mylm,
                 data.frame(dose=d,
                 supp=factor("VC")))

lines(log(d), ojfit, lwd=2, col="orange")

lines(log(d), vcfit, lwd=2, col="blue")
```

Tooth growth (ancova lm)

Other approaches

```
example(boxplot)

anova(lm(len ~ factor(dose)*supp,
          data=ToothGrowth))
```

Should dose be a linear term or a factor?

The question is whether we're interested only in 0.5/1/2 mg doses, or also in predicting the effect of other doses

Nonlinear models might be more appropriate

Fuel efficiency (multiple lm)

Stepwise selection: starting from null model

```
mylm1 <- step(lm(I(1/mpg) ~ 1, data=mtcars),
                 . ~ cyl+disp+hp+drat+wt+qsec
                 +factor(vs)+factor(am)+gear+carb)
```

Stepwise selection: starting from full model

```
mylm2 <- step(lm(I(1/mpg) ~ cyl+disp+hp+drat+wt
                    +qsec+factor(vs)+factor(am)
                    +gear+carb, data=mtcars))
```

Fuel efficiency (multiple lm)

Model comparison: AIC

```
summary(mylm1)

summary(mylm2)



AIC(mylm1, mylm2)
```

Extra credit

Now repeat the `lm()` examples

using the `linest()` function in Excel

Horse kicks

```
kick <- read.table("c:/shop/kick.txt",
                    header=T)
kick

head(kick)
```

```
xtabs(N ~ Corps+Year, data=kick)

tapply(kick$N, kick$Corps, sum)
```

```
barplot(tapply(kick$N, kick$Corps, sum))
```

Horse kicks

IX is before V, fix that

```
lev <- c("G", as.character(as.roman(c(1:11,14,15))))
```

```
kick$Corps <- ordered(kick$Corps, levels=lev)
```

```
barplot(tapply(kick$N, kick$Corps, sum))
```

Horse kicks (chisq.test)

Does the "deaths-due-to-horse-kicks" rate very between corps?

```
chisq.test(tapply(kick$N, kick$Corps, sum))
```

Does the "deaths-due-to-horse-kicks" rate very between years?

```
barplot(tapply(kick$N, kick$Year, sum))

chisq.test(tapply(kick$N, kick$Year, sum))
```

Horse kicks (glm)

```
par(mfrow=c(2,1))
barplot(tapply(kick$N, kick$Corps, sum),
        main="Deaths by Corps")
barplot(tapply(kick$N, kick$Year, sum),
        main="Deaths by Year")
```

```
kick.0 <- glm(N ~ 1, data=kick, family=poisson)

anova(step(kick.0, . ~ factor(Year)*Corps),
           test="Chisq")
```