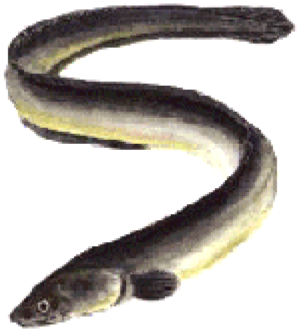

S-Plus workshop

7-9 and 14-16 January

students.washington.edu/arnima/s

Syllabus

Tue 7	**Introduction** Import data, summarize, regression, plots, export graphs
Wed 8	**Basic statistics** Descriptive statistics, significance tests, linear models
Thu 9	**Linear models** Anova, LM, GLM, loess
Tue 14	**Graphics** Types, multipanel, export graphs
Wed 15	**Data manipulation** Data objects, describe, extract, sort, manipulate
Thu 16	**Programming** Functions, import/export, project management, packages

Today: Linear models

1 Object anatomy
lm, summary

2 Regression plots
plot, loess, boxplot, coplot, interaction.plot, diagnostic plots

3 Auxiliary functions
extract elements, build models, predict, diagnose, transform

4 Exercise
weight loss

Fetch data and create models

```
library(MASS)
#R: data(mammals, cabbages)
#S: mammals <- mammals
#S: cabbages <- cabbages
mammals.lm <- lm(log(brain)~log(body), data=mammals)
cabbages.aov <- aov(VitC~Cult+Date, data=cabbages)
cabbages.lm <- lm(VitC~HeadWt, data=cabbages)
cabbages.ancova <- lm(VitC~HeadWt+Cult*Date, data=cabbages)
```

Object anatomy - How they print

`mammals.lm`

```
Call:
lm(formula = log(brain) ~ log(body), data = mammals)

Coefficients:
(Intercept)    log(body)
     2.1348       0.7517

#S: Degrees of freedom: 62 total; 60 residual
#S: Residual standard error: 0.6942947
```

Object anatomy - How they print

```
summary(mammals.lm)
#R: summary(mammals.lm, cor=T)
```

```
Call:
lm(formula = log(brain) ~ log(body), data = mammals)

Residuals:
     Min       1Q   Median       3Q      Max
-1.71550 -0.49228 -0.06162  0.43597  1.94829

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.13479    0.09604   22.23   <2e-16 ***
log(body)    0.75169    0.02846   26.41   <2e-16 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.6943 on 60 degrees of freedom
Multiple R-Squared: 0.9208,     Adjusted R-squared: 0.9195
F-statistic: 697.4 on 1 and 60 DF,  p-value: < 2.2e-16

Correlation of Coefficients:
          (Intercept)
log(body) -0.3964
```

Object anatomy - What’s inside

```
names(mammals.lm)
```

```
  call           # recipe, what we can type to create this model
  coefficients   # parameter estimates
  fitted.values
  residuals
  rank           # number of parameters estimates, df used
  df.residual    # residual degrees of freedom, df left
```

```
mammals.lm$call

mammals.lm$coe

mammals.lm$fit

mammals.lm$res

mammals.lm$rank

mammals.lm$df.res
```

Object anatomy - What’s inside

```
names(summary(mammals.lm))
```

```
  coefficients   # parameter estimates and t test of β=0
  r.squared
  correlation    # between parameter estimates
```

```
summary(mammals.lm)$coe

x <- summary(mammals.lm)

x$coe

x$r.s

x$cor
```

Symbols - Formula notation

~	`# is a function of`	`y ~ x`
+	`# add term`	`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`
.	`# same as before`	`y ~ . + x3`

Symbols - Formula notation

```
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
```

Symbols - ( ) [ ] { }

```
f(x)    # Pass argument x to function f

x[i]    # Extract element i from vector x

{cmd}   # Lump commands together as a block, used when programming
```

Regression plots

Scatterplot and friends

```
plot(log(mammals$body), log(mammals$brain))

abline(mammals.lm)

points(5, 0)

points(5, 0, cex=2)

lines(c(6,4,5), c(0,1,-1))

x.human <- log(mammals$body)[row.names(mammals)=="Human"]

x.human

y.human <- log(mammals$brain)[row.names(mammals)=="Human"]

points(x.human, y.human, pch=3, cex=2)

text(x.human, y.human+0.5, "me")
```

Smoothing with loess

```
#R: library(modreg)

plot(log(mammals$body), log(mammals$brain))

mammals.loess <- loess(log(brain)~log(body), data=mammals)

mammals.loess

summary(mammals.loess)

names(mammals.loess)
  call    # recipe, what we can type to create this model
  fitted

mammals.loess$fit

cbind(log(mammals), mammals.loess$fit)
```

Smoothing with loess

```
points(log(mammals$body), mammals.loess$fit, col=6)

lines(log(mammals$body), mammals.loess$fit, col=6)

x <- log(mammals$body)

y <- mammals.loess$fit

plot(log(mammals$body), log(mammals$brain))

lines(x[order(x)], y[order(x)])
```

Box plot

```
boxplot(cabbages$VitC)

boxplot(split(cabbages$VitC, cabbages$Date))
```

Conditioning plot

```
#R: library(lattice)

coplot(VitC~HeadWt|Cult, data=cabbages)

coplot(VitC~HeadWt|Cult, data=cabbages, panel=panel.smooth)

coplot(VitC~HeadWt|Date, data=cabbages, panel=panel.smooth,
       rows=1)

coplot(VitC~HeadWt|Date*Cult, data=cabbages, panel=panel.smooth)

#S: coplot(VitC~HeadWt|Date*Cult, data=cabbages,
           panel=panel.smooth, span=0.9)
```

Interaction plot

```
interaction.plot(cabbages$Cult, cabbages$Date, cabbages$VitC)
```

Plot influence diagnostics

```
par(mfrow=c(2,3))
#R: par(mfrow=c(2,2))
plot(mammals.lm)
par(mfrow=c(1,1))
plot(mammals.lm$fit, mammals.lm$res)
abline(h=0)
identify(mammals.lm$fit, mammals.lm$res, row.names(mammals))
```

Auxiliary functions

Formal extraction of elements

```
coef(mammals.lm)          # same as mammals.lm$coef
fitted(mammals.lm)        # same as mammals.lm$fitted
residuals(mammals.lm)     # select one of five different kinds of residuals
args(residuals.lm)
deviance(mammals.lm)      # GLM context, for lm this is SSE=sum(mammals.lm$res^2)
```

Model building and selection

```
update(mammals.lm, .~.+I(body^2))

cabbages.0 <- lm(VitC~1, data=cabbages)  # null model, intercept only

cabbages.full <- update(cabbages.0, .~.+HeadWt*Cult*Date)

add1(cabbages.0, cabbages.full)

drop1(cabbages.full)

cabbages.step <- step(cabbages.0, list(lower=cabbages.0,
                      upper=cabbages.full))

cabbages.step

anova(cabbages.full)

cabbages.plain <- update(cabbages.0, .~.+HeadWt+Cult+Date)

AIC(cabbages.0, cabbages.plain, cabbages.full)
```

Predict from new data

```
new.cabbage <- data.frame(Cult="c39", Date="d16", HeadWt=4.0)

predict(cabbages.plain, new.cabbage)

predict(cabbages.full, new.cabbage)

exp(predict(mammals.lm, data.frame(body=100)))
```

Influence diagnostics

```
mammals.diag <- ls.diag(mammals.lm)

#S: mammals.diag <- ls.diag(lsfit(log(mammals$body),
                                  log(mammals$brain)))

plot(mammals.diag$cooks, type="h")

abline(h=0)
```

See slide: Plot influence diagnostics

Transform response variable

```
mammals.plain <- update(mammals.lm, brain~body)
library(MASS)
plot(boxcox(mammals.plain))    # evaluate 1/Y, log(Y), Y, Y^2, ...
plot(logtrans(mammals.plain))  # evaluate log(Y+0.1), log(Y+1), ...
```

Models related to lm and aov

```
?glm
?gam  #R: library(mgcv)
?nls  #R: library(nls)
```

Caveats

I recommend never using attach() on data frames.

Extract residuals from lm and aov objects using the lazy $res, but use formal residuals(x,type="") for other models.

Exercise: weight loss

```
library(MASS)
#R: data(wtloss)
wtloss <- wtloss
```

Analyze the data:

- Fit a model that goes through the data reasonably well
- Paste a table and graph into Word
- Bonus question: one might be interested in predicting the person’s weight after two years at the health clinic