Introductory Statistics with R:
Linear models for continuous response (Chapters 6, 7, and 11)

Statistical Packages

STAT 1301 / 2300, Fall 2014

Sungkyu Jung
Department of Statistics
University of Pittsburgh

E-mail: sungkyu@pitt.edu
http://www.stat.pitt.edu/sungkyu/stat1301/

1. Simple linear regression
2. Multiple Regression
3. Analysis of Variance

Section 1

Simple linear regression

`lm` (Linear Model) function

A model for `thuesen` data:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i,\ \epsilon_i \sim N(0, \sigma^2),$$

$$\text{short.velocity}_i = \text{intercept} + \text{slope} \times \text{blood.glucose}_i + \text{error}_i.$$

```
library(ISwR)
plot( thuesen$blood.glucose, thuesen$short.velocity)
lm(short.velocity ~ blood.glucose, data = thuesen)
> attach(thuesen)
> lm(short.velocity ~ blood.glucose)

Call:
lm(formula = short.velocity ~ blood.glucose)

Coefficients:
  (Intercept)  blood.glucose
      1.09781        0.02196
```

Saving and extracting the result of analysis

Model object

- The result of `lm` is a *model object*. Whereas other statistical systems focus on generating printed output that can be controlled by setting options, you get instead the result of a model fit encapsulated in an object.
- The model object is a sort of list.
- From the model object the desired quantities can be obtained using extractor functions.

```
fit<-lm(short.velocity ~ blood.glucose)
summary(fit)
str(fit)
fit$coefficients
fit$fitted.values
plot(short.velocity ~ blood.glucose)
abline(fit)
```

All further analysis is based on the model object (`fit` in this example).

Fitted values $\hat{y}_i$

```
fitted(fit)
fit$fitted.values
```

Residuals $r_i = y_i - \hat{y}_i$

```
resid(fit)
fit$residuals
```

Exercise: Produce a scatterplot of `short.velocity` versus `blood.glucose` with fitted line and residual line segments.

Leverage, Cook's distance and studentized residuals

```
leverages <- hatvalues(fit)
cooks.distance(fit)
rstudent(fit)
```

Residual vs fitted values plot

```
plot(fitted(fit),rstudent(fit))
```

Checking Normality

```
qqnorm(resid(fit))
qqline(resid(fit))
```

Just want everything?

```
plot(fit)
```

Prediction and confidence intervals

`predict` function

Obtain the fitted values with upper and lower bounds on confidence intervals (`interval="confidence"`) or prediction interval (`interval="prediction"`). The arguments can be abbreviated:

```
predict(fit, int = "c")
predict(fit, int = "p")
```

Prediction for new data points

New data points (here `newpts`) arranged as a data frame containing suitable x values (here `blood.glucose`).

```
newpts <- data.frame(blood.glucose = 1:30)
pp <- predict(fit, int="p", newdata=newpts)
pc <- predict(fit, int="c", newdata=newpts)
plot(blood.glucose,short.velocity,
     ylim=range(short.velocity, pp, na.rm=T),
     xlim=range(blood.glucose,newdata,na.rm=T))
matlines(newpts$blood.glucose, pp, lty=c(1,3,3), col="black")
matlines(newpts$blood.glucose, pc, lty=c(1,3,3), col="blue")
```

- `range` function returns a vector of length 2 containing the minimum and maximum values of *all* of its arguments.
- `matlines` function plots the columns of one matrix against the columns of another.

Section 2

Multiple Regression

Scatterplot matrix

```
data(cystfibr)
par(mex = 0.5)
pairs(cystfibr, gap = 0, cex.labels= 0.9)
```

Fitting multiple regression model by `lm` function

```
lm(pemax~age+sex+height+weight+bmp+fev1+rv+frc+tlc,
          data = cystfibr)
a<-lm(pemax~., data = cystfibr)
summary(a)
```

Here, the model formula `pemax~.` stands for `pemax` explained by *all* variables in the data frame.

Model formulas

Suppose we have either continuous (numeric) or categorical (factor) variables as follows.

- `Y` : response variable
- `X`, `Z`, `W` : predictor variables.

Basic formulas

- `Y ~ X`: $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$.
- `Y ~ X+W`: $Y_i = \beta_0 + \beta_1 X_i + \beta_2 W_i + \epsilon_i$.

Examples:

```
lm(pemax ~ age, data = cystfibr)
lm(pemax ~ age + sex, data = cystfibr)
```

Importantly, the use of the "+" symbol in this context is different than its usual meaning.

Model formulas

Symbol	Example	Meaning
`+`	`+X`	include this variable
`-`	`-X`	delete this variable
`:`	`X:Z`	include the interaction between these variables
`*`	`X*Y`	include these variables and the interactions between them
`|`	`X | Z`	conditioning: include x given z
`^`	`(X + Z + W)^3`	include these variables and all interactions up to three way
`I`	`I(X*Z)`	as is: include a new variable consisting of these variables multiplied
`1`	`X - 1`	intercept: delete the intercept (regress through the origin)

In addition, the dot symbol (".") can only be used when the data frame is specified, and means "all variables except the response".

Examples

```
Y ~ X + Z + W + X:Z + X:W + Z:W
Y ~ X * Z * W - X:Z:W
Y ~ (X + Z + W)^2
Y ~ X + Y - 1
Y ~ . - X , data = adataframe # used within lm()
```

Exercises

- Fit a quadratic regression to `thuesen` data

- Fit a Quartic regression to `thuesen` data without the intercept.

- Merge weight and height to create BMI $= wt/ht^2$ and fit a linear regression model `pemax ~BMI` to `cystfibr` data

AIC and BIC for each model

```
fit1 <- lm(pemax~rv+bmp, data = cystfibr)
AIC(fit1)
extractAIC(fit1)
BIC(fit1)
```

Comparing two models

H0: Model 1: `pemax ~rv + bmp` is true. (or the `weight` coefficient $= 0$))
H1: Model 2: `pemax ~rv + bmp + weight` is true. (coefficient $\neq 0$)

To request an F-test for significance of the more complex model, use `anova` function.

```
fit1 <- lm(pemax~rv+bmp, data = cystfibr)
fit2 <- lm(pemax~rv+bmp+weight, data = cystfibr)
anova(fit1,fit2)
```

add1 function: try adding one more predictor to the current model

```
>fit1 <- lm(pemax~age, data = cystfibr)
>add1(fit1, pemax~age+sex+height+weight+bmp+fev1+rv+frc+tlc,
     data = cystfibr, test = 'F')
```

Unfortunately, using `pemax~.` in the second argument of `add1` function does not provide all variables in the data frame, but provides all variables in the current model.

drop1: try removing one predictor from the current model

```
> fit1 <- lm(pemax~., data = cystfibr)
> drop1(fit1,test ='F')
```

Exercise: do forward and backward variable selections by i) F-tests and ii) AIC values.

Automated variable selection by step function

`step(model.object, scope, direction = "backward",test = "F", ...))`

- `model.object` the model object representing the initial model in the stepwise search.
- `scope` a model formula for full model (as used in `add1`)
- Options for `direction` include `"both"`, `"backward"`, `"forward"`.
- Optional argument `test = "F"` is passed to `add1` and `drop1`.

Example:

```
fit.null <- lm(pemax~1, data =cystfibr)
step(fit.null,pemax~age+sex+height+weight+bmp+fev1+rv+frc+tlc,
     data= cystfibr,direction="both")
fit.full <- lm(pemax~., data =cystfibr)
step(fit.full,direction="backward",test = "F")
```

Section 3

Analysis of Variance

red cell folate data

```
> summary(red.cell.folate)
     folate            ventilation
 Min.   :206.0   N2O+O2,24h:8
 1st Qu.:249.5   N2O+O2,op  :9
 Median :274.0   O2,24h     :5
 Mean   :283.2
 3rd Qu.:305.5
 Max.   :392.0
```

juul data

```
> summary(juul[,3:5])
      sex              igf1            tanner
 Min.   :1.000   Min.   : 25.0   Min.   :1.00
 1st Qu.:1.000   1st Qu.:202.2   1st Qu.:1.00
 Median :2.000   Median :313.5   Median :2.00
 Mean   :1.534   Mean   :340.2   Mean   :2.64
 3rd Qu.:2.000   3rd Qu.:462.8   3rd Qu.:5.00
 Max.   :2.000   Max.   :915.0   Max.   :5.00
 NA's   :5       NA's   :321     NA's   :240
```

Exercise: Treat variables sex and tanner as categorical variables.

ANOVA model fit and ANOVA table

aov for balanced data

```
> table(red.cell.folate$ventilation)

N2O+O2,24h  N2O+O2,op     O2,24h
         8          9          5
> aov(folate~ventilation, data = red.cell.folate)
Call:
   aov(formula = folate ~ ventilation, data = red.cell.folate)

Terms:
                ventilation Residuals
Sum of Squares     15515.77  39716.10
Deg. of Freedom           2        19

Residual standard error: 45.72003
Estimated effects may be unbalanced
> summary(aov(folate~ventilation, data = red.cell.folate))
             Df Sum Sq Mean Sq F value Pr(>F)
ventilation   2  15516    7758   3.711 0.0436 *
Residuals    19  39716    2090
---
```

ANOVA model fit and ANOVA table

An ANOVA model is a linear regression model with multiple (binary) predictors.

`lm` to fit, anova for table

```
> lm(folate~ventilation, data =red.cell.folate)

Call:
lm(formula = folate ~ ventilation, data = red.cell.folate)

Coefficients:
          (Intercept)  ventilationN2O+O2,op      ventilationO2,24h
               316.62                -60.18                -38.62

> anova(lm(folate~ventilation, data =red.cell.folate))
Analysis of Variance Table

Response: folate
             Df  Sum Sq Mean Sq F value  Pr(>F)
ventilation   2  15516  7757.9  3.7113 0.04359 *
Residuals    19  39716  2090.3
```

Pairwise comparisons

The p-value in the ANOVA table is used for testing $H_0 : \alpha_1 = \cdots = \alpha_I = 0$. In case where H_0 is rejected (i.e., we have enough evidence for the alternative: there is at least one pair (i, j) such that $\alpha_i \neq \alpha_j$), one often performs the pairwise comparison for all pairs of levels.

`pairwise.t.test` function

```
> attach(red.cell.folate)
> pairwise.t.test(folate,ventilation)

Pairwise comparisons using t tests with pooled SD

data:  folate and ventilation

          N2O+O2,24h N2O+O2,op
N2O+O2,op 0.042      -
O2,24h    0.310      0.408

P value adjustment method: holm
```

Other methods for p-value adjustment can be used:

```
pairwise.t.test(folate,ventilation, p.adj= "bonferroni" )
# try none, holm, fdr for option p.adj
TukeyHSD(aov(folate~ventilation, data =red.cell.folate))
```

Miscellaneous functions for one-way ANOVA

`bartlett.test` for homogeneity of variances

The traditional one-way ANOVA requires an assumption of equal variances for all groups. To check whether this assumption is true:

```
bartlett.test(folate~ventilation)
```

`oneway.test` for unequal variances

There is, however, an alternative procedure that does not require that assumption. It is due to Welch and similar to the unequal-variances t test.

```
oneway.test(folate~ventilation)
```

Kruskal-Wallis test

A nonparametric counterpart of a one-way analysis of variance:

```
kruskal.test(folate~ventilation)
```

Diagnostics

```
plot(model.object)
```

Graphical representation for comparing levels

```
xbar <- tapply(folate, ventilation, mean)
s <- tapply(folate, ventilation, sd)
n <- tapply(folate, ventilation, length)
sem <- s/sqrt(n)
stripchart(folate~ventilation, method="jitter",
              jitter=0.05, pch=16, vert=T)
arrows(1:3,xbar+sem,1:3,xbar-sem,angle=90,code=3,length=.1)
lines(1:3,xbar,pch=4,type="b",cex=2)

boxplot(folate~ventilation)
```

juul data: Preparation

```
detach(juul)
juul$sex <- factor(juul$sex,labels=c("M","F"))
juul$menarche <- factor(juul$menarche,labels=c("No","Yes"))
juul$tanner <- factor(juul$tanner,
                      labels=c("I","II","III","IV","V"))
attach(juul)
```

Main effect model

```
TF <-complete.cases(juul[,2:5])
juul2<-subset(juul,TF)
a<-lm(igf1~ menarche +  tanner, data = juul2)
anova(a)
```

Full model with interaction

```
a<-lm(igf1~ menarche +  tanner + menarche:tanner, data = juul2)
a2<-lm(igf1~ menarche*tanner, data = juul2)
anova(a)
table(menarche,tanner)
```

ToothGrowth data

The response is the length of odontoblasts (teeth) in each of 10 guinea pigs at each of three dose levels of Vitamin C (0.5, 1, and 2 mg) with each of two delivery methods (orange juice or ascorbic acid).

```
TG2<-transform(ToothGrowth, dose = factor(dose))
attach(TG2)
table(supp,dose)
summary(aov(len~ supp*dose))
```

Two-way Interaction plot

```
#interaction.plot(x.factor, trace.factor, response)
interaction.plot(supp,dose,len)
interaction.plot(dose,supp,len)
```