

# Starting R: Session III

Danny Kaplan and Bob delMas and Andy Zieffler

March 12, 2009

# Topics for Session III

- ▶ A Menagerie of Models
- ▶ Bootstrapping
- ▶ Recording Your Commands
- ▶ Graphics
- ▶ Panel Data — organizing data for longitudinal studies

# Some Variations on Linear Models

- ▶ Transformations of Explanatory Variables
- ▶ Transformations of Response Variables
- ▶ Weighted Least Squares
- ▶ Robust Regression

## Wage Data

Data on 93 skilled, entry-level clerical workers hired by the Harris Trust and Savings Bank from 1969 to 1977. The data were made public and are described further in, Roberts, H. V., (1979). *Harris Trust and Savings Bank: An analysis of employee compensation*. Report 7946, Center for Mathematical Studies in Business and Economics, University of Chicago Graduate School of Business.

The data set includes the following variables,

- ▶ `salary`: Starting salary for the employee, in dollars
- ▶ `sex`: Sex of the employee (MALE or FEMALE)
- ▶ `senior`: Seniority of the employee, months since employee was hired
- ▶ `age`: Age of the employee at the time of hire, in months
- ▶ `educ`: Amount of education of the employee at the time of hire, in months
- ▶ `exper`: Amount of prior experience, in months

Unfortunately, the coding for `sex` includes blank characters.

## Reading in the Data I

The data were stored in a SAV file from SPSS. This can be read using the `read.spss` operator in the `foreign` library:

```
> require(foreign)
> wages = read.spss("/Users/kaplan/kaplanfiles/Projects/R-U
  to.data.frame = TRUE)
```

## The t-test approach I

```
> t.test(salary ~ sex, data = wages)
```

```
Welch Two Sample t-test
```

```
data: salary by sex
```

```
t = -5.83, df = 51.329, p-value = 3.71e-07
```

```
alternative hypothesis: true difference in means is not equal to 0
```

```
95 percent confidence interval:
```

```
-1099.6693 -536.3758
```

```
sample estimates:
```

```
mean in group FEMALE
```

```
5138.852
```

```
mean in group MALE
```

```
5956.875
```

But there are covariates!

## A Modeling Approach I

```
> mod1 = lm(salary ~ sex, data = wages)
> mod2 = lm(salary ~ sex + exper + educ +
            senior, data = wages)
> anova(mod1, mod2)
```

Analysis of Variance Table

Model 1: salary ~ sex

Model 2: salary ~ sex + exper + educ + senior

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	91	32278107				
2	88	22657469	3	9620638	12.455	7.3e-07 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

The new terms soak up a lot of the residual variance.

## A Modeling Approach II

```
> require(xtable)
> xtable(summary(mod2)$coef)
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	5845.71	526.37	11.11	0.00
sexMALE	722.30	117.82	6.13	0.00
exper	1.27	0.59	2.16	0.03
educ	90.02	24.69	3.65	0.00
senior	-23.43	5.20	-4.51	0.00

## Transformation of the Response Variable I

The salary data is slightly skew to the right. This is to be expected in this sort of data. Pay raises are often proportional, stated in percent. Considering the logarithm of the wages both fixes the skewness and handles the proportionate nature of raises.

```
> m1 = lm(log10(salary) ~ sex + exper +  
          educ + senior, data = wages)
```

The proportionate increase in salary associated with sex is

```
> 10^coef(m1)[["sexMALE      "]]
```

```
[1] 1.138323
```

Sorry about the crazy level names for the sex variable. How would you recode them?

## Robust Estimation I

Insofar as you are concerned about the effects of outliers, you can use robust estimation techniques. The `rlm` function is contained in the “MASS” library.

```
> require(MASS)
> m2 = rlm(salary ~ sex + exper + educ +
           senior, data = wages)
> summary(m2)
```

```
Call: rlm(formula = salary ~ sex + exper + educ + senior, data = wages)
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-1218.08	-345.05	-30.28	276.43	1625.44

```
Coefficients:
```

	Value	Std. Error	t value
(Intercept)	5879.3760	543.5261	10.8171
sexMALE	702.7066	121.6654	5.7757

## Robust Estimation II

exper	1.2451	0.6063	2.0538
educ	83.0568	25.4982	3.2574
senior	-22.7923	5.3696	-4.2447

Residual standard error: 511.6 on 88 degrees of freedom

## Transformation of the Explanatory Variable I

Perhaps there is a nonlinear dependence on education? The `poly` operator will construct a series of orthogonal polynomials based on the variable. Looking at the regression report can indicate what order polynomial might be appropriate

```
> m2 = lm(log10(salary) ~ sex + poly(exper,  
  3) + educ + senior, data = wages)
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	3.7549	0.0374	100.39	0.0000
sexMALE	0.0617	0.0085	7.27	0.0000
poly(exper, 3)1	0.0985	0.0365	2.70	0.0084
poly(exper, 3)2	-0.1617	0.0386	-4.19	0.0001
poly(exper, 3)3	0.0768	0.0366	2.10	0.0389
educ	0.0055	0.0018	3.04	0.0031
senior	-0.0014	0.0004	-3.57	0.0006

## Transformation of the Explanatory Variable II

No indication of significance in the second- and third-order polynomial terms.

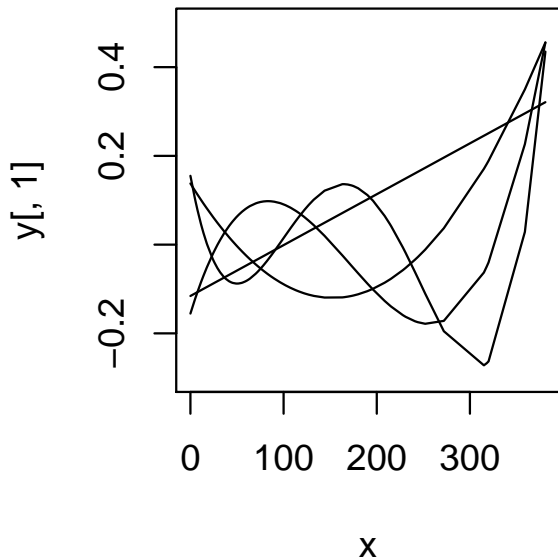
Also see the basis spline function (`bs`) in the "spline" library.

# Opening the Black Box: What are these nonlinear basis functions? I

## Polynomials

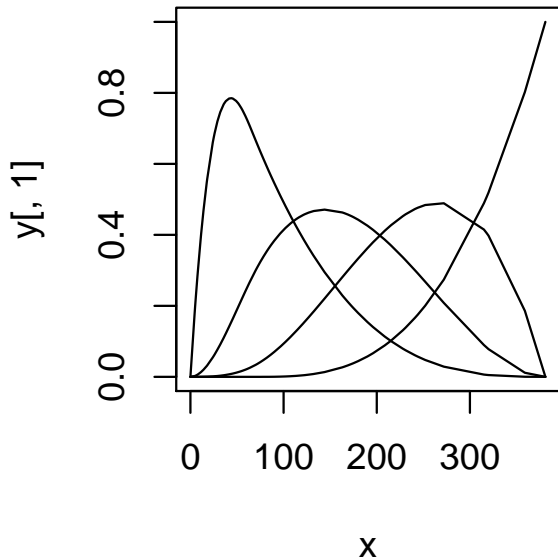
```
> x = sort(wages$exper)
> y = poly(x, 4)
> plot(x, y[, 1], ylim = c(-0.3, 0.5), type = "l")
> lines(x, y[, 2])
> lines(x, y[, 3])
> lines(x, y[, 4])
```

## Opening the Black Box: What are these nonlinear basis functions? II



## Splines

```
> require(splines)
> y = bs(x, 4)
> plot(x, y[, 1], ylim = c(0, 1), type = "l")
> lines(x, y[, 2])
> lines(x, y[, 3])
> lines(x, y[, 4])
```



# Weighted Least Squares I

When the uncertainty in the response variable differs from case to case, weighted least squares is appropriate.

- ▶ The weight should be inversely proportional to the variance for each case.
- ▶ For an average, the weight should be proportional to the number of instances that go into each average.

In BST data, is the math score in each district associated with the fraction of students on free and reduced lunch?

- ▶ Without weighting

```
> m1 = lm(mathavg ~ freelunch, data = bst)
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	651.9637	1.7176	379.57	0.0000
freelunch	-0.6561	0.0521	-12.60	0.0000

- ▶ Weighting by the number of kids who took the test

## Weighted Least Squares II

```
> m2 = lm(mathavg ~ freelunch, data = bst,  
          weights = mtested)
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	653.2056	0.9484	688.76	0.0000
freelunch	-0.7430	0.0291	-25.53	0.0000

# Exercises

- ▶ In the `bst` data, is there a relationship between `pct600m` and `pctmigrant`. Try modeling it both with and without using `mtested` as weights.
- ▶ What happens when you add in `freelunch` and `lep` as covariates?

## Exercise Answers I

```
> m1 = lm(pct600m ~ pctmigrant, data = bst)
> m1w = lm(pct600m ~ pctmigrant, data = bst,
           weights = mtested)
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	75.7822	0.6760	112.10	0.0000
pctmigrant	-0.8859	0.5193	-1.71	0.0891

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	75.2879	0.6951	108.31	0.0000
pctmigrant	-1.9001	0.8745	-2.17	0.0306

```
> m2 = lm(pct600m ~ pctmigrant + lep + freelunch,
           data = bst)
> m2w = lm(pct600m ~ pctmigrant + lep +
           freelunch, data = bst, weights = mtested)
```

## Exercise Answers II

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	89.1165	1.2755	69.87	0.0000	
pctmigrant	-0.2840	0.4407	-0.64	0.5198	and
lep	-0.2536	0.1237	-2.05	0.0412	
freelunch	-0.4350	0.0397	-10.95	0.0000	

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	89.3321	0.6677	133.79	0.0000	
pctmigrant	-0.3354	0.4300	-0.78	0.4360	
lep	-0.2753	0.0542	-5.08	0.0000	
freelunch	-0.4649	0.0280	-16.63	0.0000	

## Logistic Regression I

In the General Social Survey data (`gss.csv`), is there a relationship between age and acceptance of the birth-control pill?

- ▶ Make a binary variable that indicates acceptance of the pill  
> `foo = gss$PILLOK %in% c("AGREE", "STRONGLY AGREE")`
- ▶ Regress this on AGE

```
> m1 = glm(foo ~ AGE, data = gss, family = "binomial")
```

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-0.1779	0.0983	-1.81	0.0704
AGE34-49	-0.1346	0.1313	-1.03	0.3053
AGE50 and older	-0.4603	0.1336	-3.45	0.0006

- ▶ How about Family Type?

```
> m2 = glm(foo ~ FAMTYPE, data = gss, family = "binomial")
```

## Logistic Regression II

	Estimate	Std. Error	z value	P
(Intercept)	-0.6592	0.1060	-6.22	
FAMTYPECOHABS	0.8770	0.2450	3.58	
FAMTYPENUCLEAR	0.2605	0.1498	1.74	
FAMTYPESINGLE ADULT	0.3838	0.1482	2.59	
FAMTYPESINGLE PARENT	0.2203	0.2117	1.04	

Age 50 and older seem to be against the pill in a statistically significant way.

- ▶ Now adjust for the family type:

```
> m3 = glm(foo ~ AGE + FAMTYPE, data = gss,  
           family = "binomial")
```

## Logistic Regression III

	Estimate	Std. Error	z value	Pr
(Intercept)	-0.3888	0.1507	-2.58	
AGE34-49	-0.0720	0.1428	-0.50	
AGE50 and older	-0.4615	0.1548	-2.98	
FAMTYPECOHABS	0.6931	0.2528	2.74	
FAMTYPENUCLEAR	0.0778	0.1614	0.48	
FAMTYPESINGLE ADULT	0.3833	0.1490	2.57	
FAMTYPESINGLE PARENT	0.0451	0.2189	0.21	

## Time Series I

Imagine some very simple data:  $n = 10$  independent points:

```
> x = 1:10
```

```
> y = runif(10)
```

We want the mean and standard error on  $y$ . Easy:

```
> mean(y)
```

```
[1] 0.3724545
```

```
> sd(y)/sqrt(length(y))
```

```
[1] 0.1030414
```

Or, done as a model

```
> m1 = lm(y ~ 1)
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.3725	0.1030	3.61	0.0056

## A Time Series Situation I

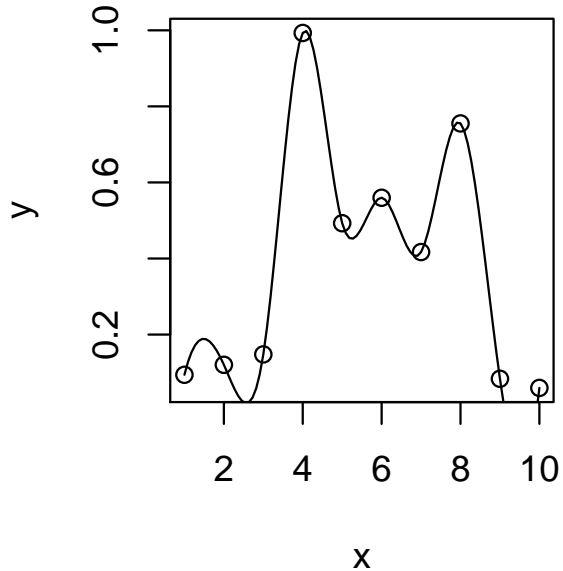
Suppose the 10 points above were a subset of a very fast time-series recording. I'll simulate that by collecting new data that is very finely sampled from a spline through the original data points:

```
> f = splinefun(x, y)
> xx = seq(1, 10, length = 100)
> yy = f(xx)
```

The finely sampled data are obviously correlated with one another:

```
> plot(x, y)
> lines(xx, yy)
```

## A Time Series Situation II



# Time Series Modeling I

We need to take the correlations into account when modeling the data.

```
> library(nlme)
> m3 = gls(model = yy ~ 1, correlation = corAR1(),
           method = "ML")
> summary(m3)
```

Generalized least squares fit by maximum likelihood

Model:  $yy \sim 1$

Data: NULL

AIC	BIC	logLik
-331.5473	-323.7318	168.7736

Correlation Structure: AR(1)

Formula:  $\sim 1$

Parameter estimate(s):

Phi

## Time Series Modeling II

0.9877499

Coefficients:

	Value	Std.Error	t-value	p-value
(Intercept)	0.1995444	0.2229583	0.8949854	0.373

Standardized residuals:

	Min	Q1	Med	Q3
	-1.2163065	-0.2206085	0.9037707	1.3853378
	Max			
	2.8367380			

Residual standard error: 0.2814945

Degrees of freedom: 100 total; 99 residual

# Multinomial Logit Models I

For situations where there are more than two classes.

```
> library(nnet)
> gss$pill = factor(gss$PILLOK, levels = c("STRONGLY DISAGRE",
      "DISAGREE", "DK", "AGREE", "STRONGLY AGREE"),
      ordered = FALSE)
> gss$pill = relevel(gss$pill, "DK")
> mm = multinom(pill ~ FAMTYPE, data = gss)

# weights:  30 (20 variable)
initial value 1499.996134
iter  10 value 1343.429063
iter  20 value 1339.436569
final   value 1339.222822
converged

> mm
```

## Multinomial Logit Models II

Call:

```
multinom(formula = pill ~ FAMTYPE, data = gss)
```

Coefficients:

	(Intercept)	FAMTYPECOHABS
STRONGLY DISAGREE	2.918574	-0.9705652
DISAGREE	2.603454	-0.5218774
AGREE	2.819177	0.3182740
STRONGLY AGREE	2.833995	0.3035611

	FAMTYPENUCLEAR
STRONGLY DISAGREE	-1.4245478
DISAGREE	-1.0694254
AGREE	-0.8349509
STRONGLY AGREE	-0.9976776

	FAMTYPESINGLE ADULT
STRONGLY DISAGREE	-0.9724632
DISAGREE	-0.7879815

## Multinomial Logit Models III

AGREE	-0.2762389
STRONGLY AGREE	-0.4621973
	FAMTYPESINGLE PARENT
STRONGLY DISAGREE	-0.97254307
DISAGREE	-0.58848060
AGREE	-0.67906265
STRONGLY AGREE	-0.06134015

Residual Deviance: 2678.446

AIC: 2718.446

Notice the graded response of COHABS and single adults.

# Recording Your Commands I

Basic concepts:

- ▶ A **script file** is a file containing R commands organized in a way that R can directly execute the commands, as if you were typing them directly into the interpreter.
- ▶ A **function** is a way of organizing operations to allow you to use them over and over again without duplicating the commands themselves. Functions also help in preventing different parts of the software from interfering with one another.

## Recording Your Commands II

- ▶ A **documentation file** is a word-processing file containing information (commentary, summaries, tables, figures) about the work you are doing or the results from that work. The information is usually intended to be read by a human.
  - ▶ Script files can also be read by humans, and sometimes they are a proxy for documentation files. But they are not suitable for presenting results.
  - ▶ Ideally, a documentation file should give enough information about the related scripts that it's possible to figure out the connection between the two and even to revise the documentation when the script is revised.

## Recording Your Commands III

- ▶ Store your sequence of commands in a **script** file. This records them and provides the foundation for documentation, debugging, revision, and validation.
- ▶ Package commands in a **function** to enable them to be re-used in similar contexts. Functions are created with a command, so they are often part of script files.
  - ▶ Functions are the fundamental structure of “modern” programming. Almost all of R consists of functions and the kinds of objects they work on.
  - ▶ For a beginner, a function is appropriate when you are doing the same operation to several different things.

## Recording Your Commands IV

- ▶ A **script** is stored in a file, which is created with an **editor**. You can use any editor you like, but some are better than others:
  - ▶ Some editors know about R syntax and help in matching parentheses, etc.
  - ▶ Some editors provide facilities for automatically executing the script in R. The alternative, is to give an R command to read in the script file — equally effective, but tedious.
  - ▶ Scripts can also be integrated into documentation files. This is an advanced topic and one we won't go into here. These slides were produced using a system (Sweave) that integrates the script with the documentation.

# Editors for Use with R

- ▶ Built in GUI editor
- ▶ Tinn-R
- ▶ Eclipse
- ▶ EMACS