Contents

IEMS 304: Statistical Methods for Data Mining, aka Stats II

Spring Quarter, 2017

Tuesday and Thursdays 12:30–1:50, Tech L251

Instructor: Professor Edward C. Malthouse
Office: Fisk 304D, 1845 Sheridan Road
Phone: 847-467-3376
email: ecm@northwestern.edu
Office Hours: after class or by appointment
Lab sessions: Wednesday, 12-12:50 or 1-1:50 in C135
TA: Cheolmin Kim CheolminKim2019@u.northwestern.edu

Objectives: You will learn how to build, interpret and apply predictive models. I will expose you to many of the issues that arise in building such models, e.g., exploratory vs. confirmatory studies, multicollinearity, heteroscedasticity, nonlinearity, interactions, model selection, regularization, bias-variance tradeoff, extrapolation, and the curse of dimensionality. You will learn how some parametric and nonparametric methods address these issues. You will understand how and why different methods work, when you should be concerned about various issues, which methods are well-suited for certain situations, and what conclusions that can be drawn from various models.

Prerequisites: A previous course in statistics at the level of IEMS 303 plus a course in matrix algebra.

Course Materials

1. Course packet.
2. Handouts, homework assignments, data sets, announcements, etc. will be posted to Canvas.
3. (suggested) *An Introduction to Statistical Learning with Applications in R*, James, Witten, Hastie and Tibshirani, Springer, labeled **JWHT**. You can get access to an e-version for free here. Click here for video lectures from the authors.
4. (optional) *Applied Linear Regression Models*, any Edition by M. H. Kutner, C. J. Nachtsheim, and J. Neter, McGraw-Hill, labled **KNN**.

There is no perfect book for this course, which covers both linear regression and machine learning methods. JWHT does a great job on the latter, but does not cover traditional regression in as much depth as I would like. KNN covers regression very well, but barely covers machine learning. Of the two, JWHT is the best single book, and it's free. There are dozens of regression books that you can read from. I have placed some on reserve at the library.

Useful References: (if you want to dig deeper or see alternative presentations of material)

- *Statistics and Data Analysis: from Elementary to Intermediate*, Tamhane and Dunlop.
- *Introduction to Linear Regression Analysis*, D. Montgomery, E. Peck, and G. Vining.
- *Regression Analysis: Theory, Methods and Applications*, Sen and Srivastava.

Pretest. IEMS has implemented pretesting in this course in order to ensure that student prerequisite knowledge is sufficient for success in the course. Data from these pretests will also be used to identify potential areas for improvement in the IEMS curriculum and as support for future accreditation evaluations. It is extremely important that you take these pretests seriously. They will have a significant impact on future accreditation of our program.

You must achieve a passing score of 70% or higher by Monday, April 10 at 11:59 p.m. Otherwise, your final course grade will be reduced by one letter. You may take the pretest as many times as you wish before then in order to achieve this passing score; however after 15 attempts you must request additional attempts from Prof. Wilson (students rarely, if ever, need this many). Feedback will be provided at the end of each attempt pointing out concepts that you should review before a subsequent attempt. Note that informational questions (such as "Are you currently declared as an IE major?") do not count towards your score on the pretest. **Again, failure to attain a passing score by April 10 will result in a one-letter-grade reduction in the final course grade. This will be firmly enforced.**

It is in your best interest to achieve a passing score on the pretest by the end of the add/drop period (May 5) so that you can drop this course if necessary and add a replacement course.

To take the pretest, go to https://assessments.mccormick.northwestern.edu and log in with your netid and password. If you encounter any problems, contact Prof. Wilson immediately. You are strongly encourage to complete this requirement early, so that you can get help with any technical glitches that arise. Tech support will only be provided from the software developers during business hours. Technical issues will not be considered a valid excuse for not passing the pretest by the deadline and the one-letter-grade reduction will still apply.

Office hours. I have office hours after class in Tech C113. Please send me an email if you want to meet after class. Our TA, Cheolmin Kim, has office hours on Wednesdays 10–12 in Tech C229

Grades. I expect "B" students to be able to work problems similar to those in the assigned homework from the text. "A" students should be able to solve problems that are not exactly like the assigned problems from the text and combine concepts from multiple chapters to solve a problem. Your course grade will be determined as follows:

1. *Midterm:* 25%. Currently scheduled for **Thursday, Feb 5** during class. **Do not miss this class**. The exam will cover all material through January 29. Exams will be returned to you during the TA session on Wednesday, May 3, prior to the course drop date.
2. *Final exam:* 30%. **Monday, June 5 from 3:00 until 5:00 pm.**
3. *Homework:* 25%
4. *Project:* 20%
5. *Class participation bonus.* If you say nothing in class, after class, or during office hours your "class participation" will not affect your course grade. If you contribute to class by consistently asking constructive questions or making insightful comments, I may increase your course grade by one third of a letter (e.g., if the numbers indicate your are a B+ student, I may assign you a course grade of A−).

Software: I will teach and use R/S in class. R will differentiate you in the job market and help you in other classes (See *NYT* article). You can download a copy for free from CRAN. Minitab,

SPSS and Stata are easy to learn and have the functionality to do assignments until about week 7. You could also use SAS. Some of the assignments can be completed in Excel, but its functionality is limited.

Exam policies:

- I do not give makeup exams. Please attend the exam at the scheduled time.
- You may use one 8.5 by 11 inch sheet of notes on the midterm and two sheets of this size on the final. I will provide a normal table if necessary.
- I will distribute practice exams given during previous years.
- You must show your work on the exam to receive credit. I will give full credit if you get the right answer and show your work.
- When answering essay questions, be concise and get to the point. **Answer the question and nothing else.** Lead with the headline. Good answers prioritize what is most important. Think before you start to write and don't do a "brain dump." You should get into these habits in your professional life too.

Honesty, Plagiarism, and Cheating. The code of student conduct for all Northwestern University students is contained in the Student Handbook. All students should have received a copy of this booklet. If you have not, please notify me immediately. Questions of academic dishonesty, cheating, plagiarism and other violations, and the terms and conditions are all listed in the handbook.

How to be successful in this class. Practice daily! Start with the problems assigned in the packet and textbook, and then do the problems in the practice exams in the back of the course packet. The only way to learn this material is to practice.

Course Outline:

This is my intended schedule, although we may not get to all of the topics.

- Week 1
  - Familiarize yourself with R
  - Mar 28: Review of simple linear regression: model assumptions, interpretation, inference, prediction intervals, sums of squares, measures of fit (JWHT §3.1; KNN Chs. 1–3, dropping §2.11, 3.10)
  - Mar 6 TA session: Introduction to R (JWHT §2.1)
  - Mar 7: Multiple linear regression model, interpretation, derivation of ordinary least-squares (OLS) estimates, and basic inference (JWHT §3.2; KNN Ch. 6)
  - New R: `read.csv`, `names`, `summary`, `cor`, `plot`, `lm`, `confint`, `predict`, `abline`
- Week 2: Sums of squares and multicollinearity
  - Apr 4: Newfood case, Multicollinearity (JWHT §3.3.3.6; KNN §7.6, 10.5)
  - Apr 5 TA session: Regression in R (JWHT §3.6.1–3.6.3)

  - Apr 6: Quality control case; Extra and partial sums of squares, F test (KNN §7.1–5)
  - New R: `anova`, `drop1`, `deviance`, `install.packages`, `library`, `vif`

- Week 3: Diagnostics and transformations
  - Apr 11: Residuals, QQ plots and outlier detection; variance stabilizing transformations; Tukey's ladder of transformations (JWHT §3.3.2, 3.6.5; KNN §8.1)
  - Apr 12 TA session: `anova/drop1` and diagnostics in R
  - Apr 13: Polynomial and multiplicative models; purify and business failure cases
  - New R: `plot`, formulas with `log`, `sqrt` or `I`

- Week 4: Categorical predictors and interactions
  - Apr 18: Dummy variables and the F test revisited, missing values (JWHT §3.3.1, 3.6.6; KNN §8.3–4)
  - Apr 19 TA session: Transformations and Dummies in R (JWHT §3.6.5, 3.6.6)
  - Apr 20: Interactions (JWHT §3.6.4; KNN §8.2, 5, 6)
  - New R: `is.na`, `factor`, formulas with `as.factor`, `*`, `:`

- Week 5
  - Apr 25: Logistic regression (JWHT §4.1–3; KNN §14.1-14.9)
  - Apr 26 TA session: Midterm Review, Interactions in R (JWHT §3.6.4)
  - Apr 27: **Midterm** (covers linear regression, JWHT ch. 3, not logistic)
  - New R: `glm`, `table`

- Week 6:
  - May 2: Maximum likelihood estimation, likelihood ratio test, ROC curves
  - May 3 TA session: Return midterms, logistic regression in R (JWHT §4.6.2)
  - May 4: Model validation: penalized and out-of-sample methods (JWHT §5.1)
  - New R: `AIC`, `logLik`, `for` loops, `lm`/`glm` with `subset`
  - **Drop deadline on May 5**

- Week 7: Model selection and regularization
  - May 9: Automated variable selection, bias-variance tradeoff (JWHT §6.1; KNN §9.1, 4, 5).
  - May 10 TA session: Cross validation in R (JWHT §5.3.1–3)
  - May 11: Ridge regression and the lasso (JWHT §6.2; KNN 11.2).
  - New R: `step`, `lm.ridge`, `glmnet`

- Week 8: Smoothing and generalized additive models
  - May 16: Bin smoothers and splines (JWHT §7.1–6)

  - May 17 TA session: Selection and shrinkage in R (JWHT §6.5, 6.6)
  - May 18: Generalized additive models (JWHT §7.7)
  - New R: `cut`, `gam`, `plot.gam`, formulas with `bs` and `s`

- Week 9: Tree-based models
  - May 23: Recursive partitioning and CART (JWHT §8.1)
  - May 24 TA session: Smoothing in R (JWHT §7.8)
  - May 25: Bagging, random forests and boosting (JWHT §8.2)
  - New R: `tree`, `cv.tree`, `randomForest`, `importance`, `varImpPlot`, `gbm`

- Week 10
  - May 30: Teams present projects
  - May 31 TA session: Trees in R (JWHT §8.3)
  - Jun 1: Introduction to recommender systems

- Final exam: Monday, June 5, 3:00–5:00pm

Project: Please complete a project of your choice in which you develop and document a regression model. You may work in self-selected groups of 1–5 students and I will help you find a group if necessary. More information to come. **Read Siegel Ch. 13** available on Canvas. You will be evaluated on the following criteria: (1) difficulty of the project and the quality of your analysis; (2) quality of the your written report (the report should follow the outline given in Siegel Ch. 13); and (3) Quality of your presentation. The presentation should be 10 minutes long.

The project should analyze a real data set. You may work on a data set from your own research, or from kdnuggets or the UCI machine learning repository. I may also have some projects for real companies, and will update you in week 4. **The project should address a substantive issue; projects addressing trivial or non-existant issues are not allowed.** As a general rule of thumb, the data set should have hundreds or thousands of cases and at least five variables, although problems of exceptional substantive interest could be smaller. Unless you have strong computation skills, you should probably avoid data requiring substantial pre-processing (e.g., multiple tables from a relational database), or very large data sets with hundreds of thousands or cases and/or hundreds of predictors. You should find a project and get my approval by the end of week 5. Email me a list of the team members and a brief description of the data set: (1) what problem is being studied; (2) why is it important; (3) how many variables and cases; (4) where are you getting the data? Meet with me at least once to discuss an analysis plan.

R Basics

- Install R from CRAN. Use Quick-R for excellent reference. Here is *NYT* article.
- R is case sensitive, unlike SAS and SPSS, e.g., `MEAN` $\neq$ `mean` $\neq$ `MeAn`
- You can get help on any command by typing ? at the command prompt, e.g., `?mean`
- Specify paths with a forward slash instead of a backslash. (Backslash is used to indicate special characters, e.g., as in C and Python, `\t` is a tab and `\n` is a new line.)
- You can bring back previous commands with the up arrow
- To quit type `q()`
- Use `#` to specify a comment
- `T` and `F` are constants for true and false
- `NA` is a constant indicating a missing value
- `==` is a logical equals, while `=` is an assignment equals (`!=` is not equals, like C)
- The first element of an array is numbered 1

Basic Data Structures in R

- R offers many data structures including `list`, `data.frame`, `matrix`, and `vector`
- Use `as.`function to cast and `is.`function to test type, e.g., `as.integer`, `as.double`, `as.numeric`, `as.vector`, `as.matrix`, etc. For example, `as.character(5)` returns the number 5 as a string
- Use `c` to specify a vector constant, e.g.,

```
> x = c(1,2,3,4,5)
> x
[1] 1 2 3 4 5
> is.vector(x)
[1] TRUE
> as.character(x)
[1] "1" "2" "3" "4" "5"
```

- Equivalently, you can use the `:` to specify a range

```
> x = 0:5
> x
[1] 0 1 2 3 4 5
> x[2:4]   # reference the second through fourth elements
[1] 1 2 3
> x[-1]    # drop the first element
[1] 1 2 3 4 5
```

- Use `matrix` to specify a matrix

```
> x = matrix(c(1,2,3,4), nrow=2)
> x
     [,1] [,2]
[1,]    1    3
[2,]    2    4
> is.matrix(x)
[1] TRUE
```

Reading Data into R

- Manage giant data sets outside of R, e.g., in a relational database system.
- Do complicated preprocessing elsewhere, e.g., with Python.
- Sample down to tens of thousands of cases and export a delimited file, e.g., `employee.csv`:

```
empnum,salary,female,ageyrs,expyrs,trainlev
1,32368,1,42,3,2
2,53174,0,54,10,2
3,52722,0,47,10,1
4,53423,0,47,1,2
5,50602,0,44,5,2
6,49033,0,42,10,1
7,24395,0,30,5,1
8,24395,1,52,6,1
9,43124,0,48,8,1
...
```

- Reading data: use `read.table` or one of its cousins such as `read.csv`. Both return a `data.frame`.

```
> employee = read.table("~ecm/teach/data/employee.csv", header=T, sep=",")
> employee = read.csv("~ecm/teach/data/employee.csv") # this is equivalent
> ?read.csv     # shows you how the options are set
```

  For help type `?read.table`, e.g., `header=T` indicates a header line and `sep=","` specifies comma-separated values.

- R cannot handle commas or dollar signs within a number without special packages—avoid them when exporting data!

Interacting with Data Frames

- To show the variables in a `data.frame`, use `names`

```
> names(employee)
[1] "empnum" "salary" "female" "ageyrs" "expyrs" "trainlev"
```

- Use `dim`, `nrow`, and `ncol` to show the dimensions.

```
> dim(employee)
[1] 71  6
> nrow(employee)
[1] 71
> ncol(employee)
[1] 6
```

- Use `summary` to show basic summary statistics

```
> summary(employee)
     empnum          salary          female           ageyrs
 Min.   : 1.0   Min.   :23975   Min.   :0.0000   Min.   :30.00
 1st Qu.:18.5   1st Qu.:36792   1st Qu.:0.0000   1st Qu.:40.50
 Median :36.0   Median :49033   Median :0.0000   Median :46.00
 Mean   :36.0   Mean   :45127   Mean   :0.3944   Mean   :45.54
 3rd Qu.:53.5   3rd Qu.:53174   3rd Qu.:1.0000   3rd Qu.:51.00
 Max.   :71.0   Max.   :62530   Max.   :1.0000   Max.   :61.00
...
```

- To show a particular column, use a dollar sign or the column number, e.g.,

```
> employee$salary[1:10]
 [1] 32368 53174 52722 53423 50602 49033 24395 24395 43124 23975
> employee[1:10, 2]    # print rows 1-10 of second column
 [1] 32368 53174 52722 53423 50602 49033 24395 24395 43124 23975

> employee[2,]    # print second row
  empnum salary female ageyrs expyrs trainlev
2      2  53174      0     54     10        2
```

- Question: How could we print rows 4–8 in columns 2–3?

Managing Data Objects

- The `ls()` function lists objects in the working folder.

```
> ls()
[1] "default"  "employee"

> x = 4
> x
[1] 4

> ls()
[1] "default"  "employee" "x"
```

- Use `rm` to remove an object

```
> rm(x)
> ls()
[1] "default"  "employee"
```

- Use `search` to display the search path, e.g., `package:stats` has all the statistical functions—type `ls(pos=2)`

```
> search()
[1] ".GlobalEnv"        "package:stats"     "package:graphics"
[4] "package:grDevices" "package:utils"     "package:datasets"
[7] "package:methods"   "Autoloads"         "package:base"
```

- Objects can be added or dropped from the search path with `attach` and `detach`

```
> attach(employee)
> search()
 [1] ".GlobalEnv"        "employee"          "package:stats"
 [4] "package:graphics"  "package:grDevices" "package:utils"
 [7] "package:datasets"  "package:methods"   "Autoloads"
[10] "package:base"
> salary[1:10]
 [1] 32368 53174 52722 53423 50602 49033 24395 24395 43124 23975
```

R: Operators and Mathematical Functions

- Arithmetic: Use `+`, `-`, `*`, `/` and `^` (exponentiation)
- For integer division and the mod function use `%/%` and `%%`

```
> 0:10
 [1]  0  1  2  3  4  5  6  7  8  9 10
> 0:10 %% 4
 [1] 0 1 2 3 0 1 2 3 0 1 2
> 0:10 %/% 4
 [1] 0 0 0 0 1 1 1 1 2 2 2
```

- R offers usual math functions, e.g., `sqrt()`, `log()`, `round()`, `exp()`, `sin()`, `cos()`, and `tan()`, etc.

```
> round(log(1:5), 4)   # round to 4 decimal places
[1] 0.0000 0.6931 1.0986 1.3863 1.6094
```

- Question: how could we find base-10 logs?
- Functions for basic summary statistics:

```
> length(employee$salary)
[1] 71

> mean(employee$salary)
[1] 45127.42

> sqrt(var(employee$salary))   # same as sd(employee$salary)
[1] 10816.88

> min(employee$salary)
[1] 23975

> max(employee$salary)
[1] 62530

> quantile(employee$salary)
   0%   25%   50%   75%  100%
23975 36792 49033 53174 62530
```

Factors

- *Factors* are categorical variables, which automatically receive special treatment by certain functions in R, e.g., `summary` displays a frequency distribution and the linear model function will create dummies.
- You can assign value labels with the `labels` option:

```
> table(employee$trainlev)

 1  2  3
38 24  9
> employee$trainlev = factor(employee$trainlev, labels=c("Low", "Med", "High"))
> table(employee$trainlev)

 Low  Med High
  38   24    9
```

- If values are not numbered consecutively from 1 then use `levels` option:

```
> employee$female = factor(employee$female, levels=0:1, labels=c("male", "female"))
> summary(employee[,3:6])
    female       ageyrs          expyrs       trainlev
 male  :43   Min.   :30.00   Min.   : 0.000   Low :38
 female:28   1st Qu.:40.50   1st Qu.: 3.000   Med :24
             Median :46.00   Median : 5.000   High: 9
             Mean   :45.54   Mean   : 5.746
             3rd Qu.:51.00   3rd Qu.: 8.500
             Max.   :61.00   Max.   :10.000
```

Dates

- Variables of class `Date` record dates as the number of days that have elapsed since Jan 1, 1970.
- Use `as.Date` to cast a variable as a Date. The default

```
> as.Date("1970-01-02")
[1] "1970-01-02"
> as.numeric(as.Date("1970-01-02"))
[1] 1
> as.Date("02JAN1970", "%d%b%Y")   # big Y means 4 digit year
[1] "1970-01-02"
> as.Date("01/02/1970", "%m/%d/%Y")
[1] "1970-01-02"
> as.Date("02/01/1970", "%d/%m/%Y")
[1] "1970-01-02"
> as.Date("02/01/70", "%d/%m/%Y")
[1] "0070-01-02"
> as.Date("02/01/70", "%d/%m/%y")   # little y means 2 digit year
[1] "1970-01-02"
```

- Functions `weekdays`, `months` and `quarters` extract parts of dates. Subtraction, greater-than, etc. can be used.

```
> x=as.Date("1970-01-02")
> months(x)
[1] "January"
> as.Date("1970-01-10")-x
Time difference of 8 days
> as.Date("1970-01-10")>x
[1] TRUE
> as.Date("1970-01-10")<x
[1] FALSE
```

Merges in R

```
> one = data.frame(id=1:4, x=1:4)
> two = data.frame(id=c(1,2,3,5), y=c(1,2,3,5))

> merge(one, two, by="id")    # inner join
  id x y
1  1 1 1
2  2 2 2
3  3 3 3

> merge(one, two, by="id", all.x=T)  # left join
  id x  y
1  1 1  1
2  2 2  2
3  3 3  3
4  4 4 NA

> merge(one, two, by="id", all.y=T)  # right join
  id  x y
1  1  1 1
2  2  2 2
3  3  3 3
4  5 NA 5

> merge(one, two, by="id", all=T)  # full outer join
  id  x  y
1  1  1  1
2  2  2  2
3  3  3  3
4  4  4 NA
5  5 NA  5

> cbind(one, two)   # bind the columns without a key
  id x id y
1  1 1  1 1
2  2 2  2 2
3  3 3  3 3
4  4 4  5 5
```

R: Membership Case

The data set default is from a company that provides a service to its customers. The service is targeted at **adults**. Customers join and may cancel at any time. The company would like to understand who is likely to default. Customers pay an initial down payment and then monthly payments over three years. You have a sample of customers who joined over a three-year period of time.

- `enrolldt`: date of enrollment (e.g., 2007/05/01)
- `price`: price of the membership
- `downpmt`: down payment
- `monthdue`: monthly dues
- `pmttype`: method of paying monthly dues. The four types of payment that we want to study are 1=Book, 3=Statement, 4=Checking EFT, and 5=Credit Card EFT.
- `use`: number of times the customer used the service during the first month
- The `default` variable in the data set takes the value 1 if customers has defaulted within the first month of their membership and 0 otherwise
- `age` and `gender` of the member

```
default = read.csv("/Users/ecm/teach/data/defaultsmall.csv")
default$gender = factor(default$gender, labels=c("Male", "Female"))
default$pmttype = factor(default$pmttype, levels=c(1,3:5),
  labels=c("Book", "Statement", "Check EFT", "Credit EFT"))
```

- Your turn: Read the data set into R and prepare to analyze it.

Introduction to Regression

- *Objective:* to quantify the relationship between an interval-level *response* variable and one or more *predictor* variables.
- *Response variable* (also called *dependent*, *criterion*, or *output* (Y) variable): a variable that we wish to study and is causally dependent on other variables
- Note: we will also study categorical dependent variables, but this will be called the *classification problem*
- *Predictor variables* (also called *independent* (X) variables, *covariates*, or *inputs*): variables that are (causally) related to the response variable
  - *Factors* refer to categorical variables
  - *Covariates* refer to numerical variables
- *Why?*
  - *Prediction*: primarily interested in estimating response variable, i.e., $\hat{y}$ (pronounced "y-hat") values
  - Causal attribution: primarily interested in how predictor variables affect response variable, e.g., "β_j" values. This is more *prescriptive*

  Causal attribution is much more difficult, with many more considerations.

Linear Regression

Assume that

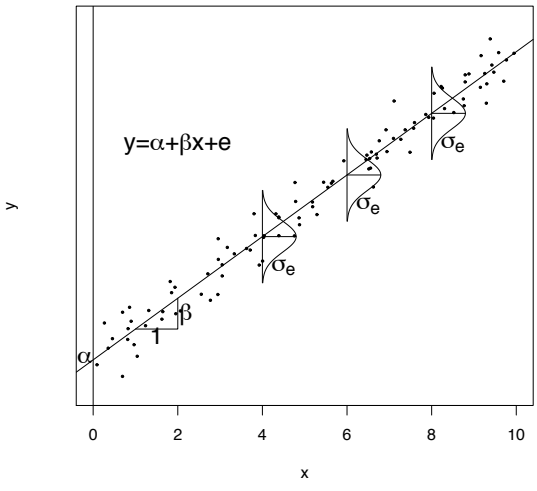
$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i,$$

where **error** $e_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$, and ϵ_i independent of ϵ_j for $i \neq j$. This implies that

$$\mu_{Y|x} = E(Y|x) = \beta_0 + x\beta_1.$$

This is called the **regression of y on x**.

- β_0: the **intercept**. On average $Y = \beta_0$ when $x = 0$.
- β_1: the **slope**. Every unit increase in x is associated with an increase in Y of β_1, *on the average*
- σ_ϵ: **standard deviation of the errors** (σ_ϵ^2 called **error variance**). If errors are normal, empirical rule tells us for any x, 68% of points will fall within one σ_ϵ of the mean $(\beta_0 + \beta_1 x)$.



Estimating the Regression Model

- In practice, we do not know value of *parameters* β_0, β_1, and σ_ϵ
- Estimate β_0, β_1, and σ_ϵ with b_0, b_1, and S_ϵ
- The estimate of $\mu_{Y_i|x_i}$ is denoted by **fitted**, **predicted** or **y-hat value** $\hat{y}_i = b_0 + b_1 x_i$
- The **residual** for observation i is

$$\hat{e}_i = y_i - \hat{y}_i = y_i - a - bx_i$$

- We choose b_0 and b_1 so that they minimize the **least-squares criterion** (RSS means **residual sum of squares**):

$$\text{RSS} = \sum_{i=1}^{n} \hat{e}_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - b_0 - b_1 x_i)^2,$$

- Theorem: the **ordinary least-squares estimates** (OLS) are

$$b_1 = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sum (x_i - \overline{x})^2} = r\frac{S_y}{S_x} \quad \text{and} \quad b_0 = \bar{y} - b_1 \bar{x}$$

  Note, for every SD change in x, there is a change of r SD's in y, on average.

- Estimate σ_ϵ with the **residual standard error**

$$S_\epsilon = \sqrt{\frac{\text{RSS}}{n-2}} = S_y \sqrt{(1 - r^2)\frac{n-1}{n-2}}$$

Sampling Distribution of Parameters

- Theorem: *standard errors* are given by

$$S_{b_1} = \frac{S_\epsilon}{S_x\sqrt{n-1}} \quad \text{and} \quad S_{b_0} = S_\epsilon\sqrt{\frac{1}{n} + \frac{\bar{x}^2}{S_x^2(n-1)}}$$

- Theorem:
  - $E(b_1) = \beta_1$ and $E(b_0) = \beta_0$
  - b_1 and b_0 have normal distributions and the following have t distributions with $n-2$ degrees of freedom:

$$\frac{b_1 - \beta_1}{S_{b_1}} \quad \text{and} \quad \frac{b_0 - \beta_0}{S_{b_0}}$$

- Definition/Theorem: The **coefficient of determination** (aka R^2) is the square of the correlation and tells you the percentage of the variability of y that is "explained" by x

$$R^2 = r^2 = 1 - \frac{\text{RSS}}{\text{TSS}},$$

  where $\text{TSS} = \sum(y_i - \bar{y})^2 = (n-1)S_y^2$ is the **total sum of squares**.

- Gauss-Markov Theorem: b_1 has a variance that is "smaller" than that of any other linear, unbiased estimator and is called the *best linear unbiased estimator* ("BLUE").

Glossary: Regression Terms and Symbols

Term	**Symbol**
Error	e or ϵ
Residual or estimated error	$\hat{e}$ or $\hat{\epsilon}$
Intercept parameter Intercept estimate	β_0 or α $\hat{\beta}_0$, b_0, or a
Slope parameter Slope estimate	β_1 or β $\hat{\beta}_1$, b_1, or b
Standard deviation of the errors	$\sigma_\epsilon = \sigma$
Error variance Variance of the errors	$\sigma_\epsilon^2 = \sigma^2$
Sum of squared errors Residual sum of squares	SSE = RSS
Total sum of squares	SST = TSS
Regression sum of squares	SSR
Residual standard error Root mean squared error Standard error of the estimate	$S_\epsilon = \hat{\sigma}$ = RMSE

Click Ball Point Pens Example

y_i Sales in territory $i = 1, \ldots, 40$

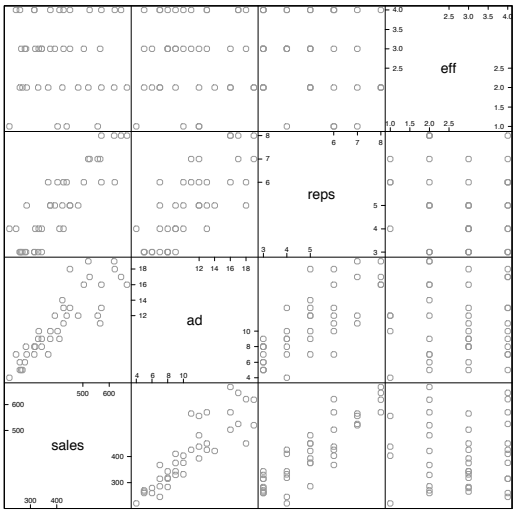
x_{i1} Advertising (number of TV spots) in territory i

x_{i2} Number of sales reps in territory i

x_{i3} Wholesaler efficiency index in territory i (4=outstanding, 3=good, 2=average, 1=poor)

```
click = data.frame(sales=c(260.3,286.1,279.4,410.8,438.2,315.3,565.1,570.0,426.1,315.0,
    403.6,220.5,343.6,644.6,520.4,329.5,426.0,343.2,450.4,421.8,245.6,503.3,375.7,265.5,
    620.6,450.5,270.1,368.0,556.1,570.0,318.5,260.2,667.0,618.3,525.3,332.2,393.2,283.5,
    376.2,481.8), ad=c(5,7,6,9,12,8,11,16,13,7,10,4,9,17,19,9,11,8,13,14,7,16,9,5,18,18,
    5,7,12,13,8,6,16,19,17,10,12,8,10,12), reps=c(3,5,3,4,6,3,7,8,4,3,6,4,4,8,7,3,6,3,5,
    5,4,6,5,3,6,5,3,6,7,6,4,3,8,8,7,4,5,3,5,5), eff=c(4,2,3,4,1,4,3,2,3,4,1,1,3,4,2,2,4,
    3,4,2,4,3,3,3,4,3,2,2,1,4,3,2,2,2,4,3,3,3,4,2))
```

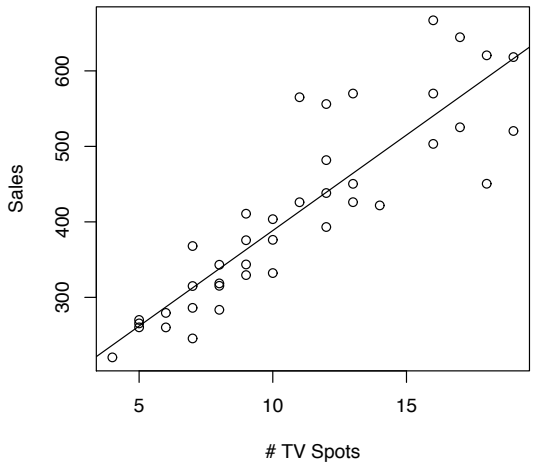
```
> round(cor(click), 4)
       sales     ad    reps     eff
sales 1.0000 0.8802  0.8818  0.0019
ad    0.8802 1.0000  0.7763  0.0321
reps  0.8818 0.7763  1.0000 -0.1896
eff   0.0019 0.0321 -0.1896  1.0000

> plot(click)
```



This is a **scatterplot matrix**

Estimates From Click Ball Point Pens

```
> fit = lm(sales ~ ad, click)
> plot(click$ad, click$sales,
    xlab="# TV Spots", ylab="Sales")
> abline(fit)
> summary(fit)
> plot(fit)   # gives diagnostic plots
```




```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  135.434     25.907   5.228 6.50e-06 ***
ad            25.308      2.214  11.430 7.33e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 59.56 on 38 degrees of freedom
Multiple R-squared: 0.7747,Adjusted R-squared: 0.7687
F-statistic: 130.6 on 1 and 38 DF,  p-value: 7.327e-14
```

- Residual standard error: $S_\epsilon = 59.56$, standard error of estimate
- R-square = 0.7747: fraction of variation explained by model
- Adj R-sq: 0.7687 adjusted for number of parameters

Interpretation of Output

- The estimated regression model is

$$\hat{y} = 135 + 25.3\texttt{ad}$$

  What does 25.3 tell us? What about 135?

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  135.434     25.907   5.228 6.50e-06 ***
ad            25.308      2.214  11.430 7.33e-14 ***
```

- Standard errors are $S_{b_0} = 25.91$ and $S_{b_1} = 2.21$
- A 95% CI for β_1: $25.3 \pm 2.02 \times 2.21 \approx (20.8, 29.8)$

```
> confint(fit)
                2.5 %    97.5 %
(Intercept) 82.98862 187.87857
ad          20.82538  29.79001
```

- To test the hypotheses (with Type I error rate .05)

$$H_0 : \beta_1 = 0 \quad \text{versus} \quad H_1 : \beta_1 \neq 0,$$

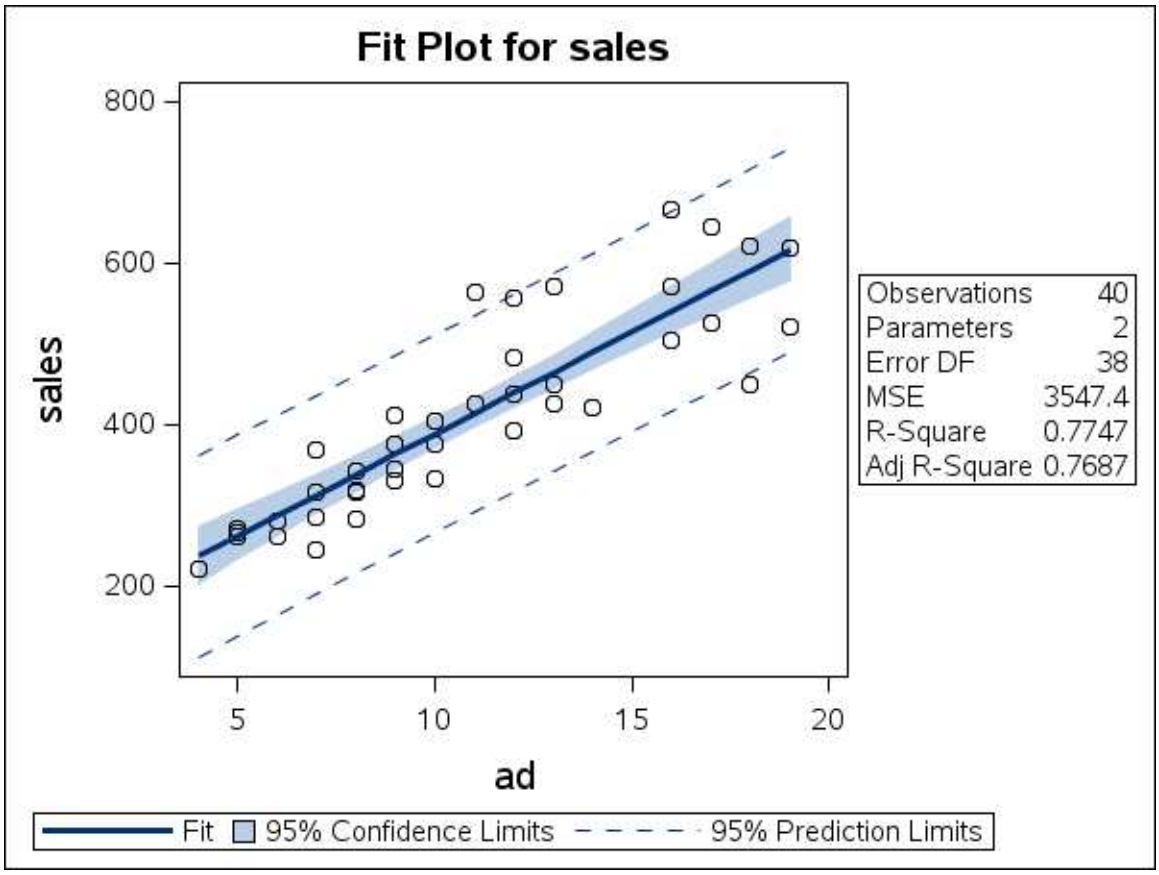
  P-value$= 7.33 \times 10^{-14} < .05$, so reject H_0 and conclude $\beta_1 \neq 0$.

- The expected sales when advertising is 5 (spots) is

$$\hat{y} = 135 + 25.3 \times 5 = 261.97$$

```
> predict(fit, data.frame(ad=5))
261.9721
```

Prediction and Confidence Intervals



- *Confidence interval for mean prediction* (shaded blue area): 95% confidence interval for $\hat{y} = b_0 + b_1 x$, i.e., indicates sampling variation of predicted values $E(Y|x)$.
- *Prediction interval* (dashed lines): indicates where the middle 95% of the distibution of Y for a given x_0 falls. If we knew parameters, it would be $(\beta_0 + \beta_1 x_0) \pm 1.96\sigma_\epsilon$.

Additional Results

- The standard error of a new observation Y given x_0:

$$S_{Y|x_0} = \sqrt{S_\epsilon^2 \left(1 + \frac{1}{n}\right) + S_{b_1}^2 (x_0 - \bar{x})^2}$$

Example: find a 95% prediction interval for the mean sales when there are 5 ads. Hint: the 97.5 percentile of a t distribution with $40 - 2 = 38$ degrees of freedom is 2.024.

```
predict(fit, data.frame(ad=5), interval="prediction")
```

$$S_{Y|x_0} = \sqrt{3547 \left(1 + \frac{1}{40}\right) + 2.214^2 (5 - 10.90)^2} = 61.70$$

$$262.0 \pm 2.024 \times 61.699 = (137.1, 386.9)$$

- The standard error of a predicted value $\hat{Y}$ given x_0

$$S_{\hat{Y}|x_0} = \sqrt{\frac{S_\epsilon^2}{n} + S_{b_1}^2 (x_0 - \bar{x})^2}$$

- Example: find a confidence interval for the mean sales when there are 5 ads.

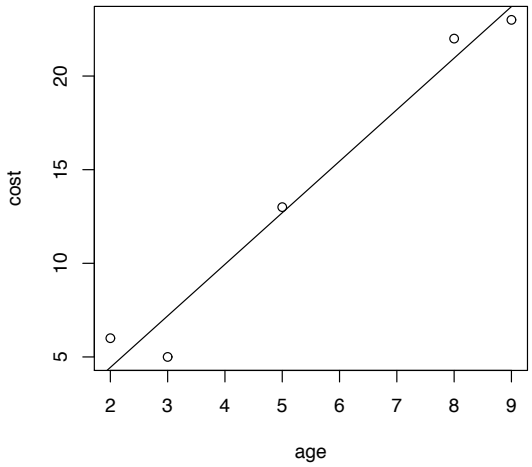
```
predict(fit, data.frame(ad=5), interval="confidence")
```

$$S_{\hat{Y}|x_0} = \sqrt{\frac{3547}{40} + 2.214^2 (5 - 10.90)^2} = 16.104$$

$$262.0 \pm 2.024 \times 16.104 = (229.4, 294.6)$$

Your Turn: Machine Example Revisited

```
> machine = data.frame(age=c(2,5,9,3,8),
  cost=c(6,13,23,5,22))
> fit = lm(cost ~ age, machine)
> plot(machine)
> abline(fit)
```




```
> summary(fit)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.0645     1.7108  -0.622  0.57788
age           2.7527     0.2828   9.734  0.00230 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.725 on 3 degrees of freedom
Multiple R-squared: 0.9693,Adjusted R-squared: 0.9591
F-statistic: 94.75 on 1 and 3 DF,  p-value: 0.002303
```

Your Turn

The amounts of a chemical compound y, which was dissolved in 100 grams of water at various temperatures, x° C, were recorded:

```
dat = data.frame(
  x=c(0,0,0, 15,15,15, 30,30,30,  45,45,45,  60,60,60,  75,75,75),
  y=c(8,6,8, 12,10,14, 25,21,24,  31,33,28,  44,39,42,  48,51,44))
```

1. Find the equation of the regression line.
2. Graph the line on a scatterplot.
3. Compute and interpret the standard error of the estimate.
4. Compute and interpret the coefficient of determination.
5. Find a 99% CI for β_0
6. Find a 99% CI for β_1
7. Test at the 1% level if the slope differs from 0.
8. Estimate the mean amount of chemical that will dissolve in 100 grams of water at 50° C.
9. Find a 99% CI for the mean amount that will dissolve at 50° C.
10. Find a 99% PI for the mean amount that will dissolve at 50° C.
11. Examine the residual and Q-Q plots. What are you looking for and why?

```
dat=data.frame(
circ = c(2081995,1374858,1284613,1057536,970051,963069,828236,779259,768288,
  691771,663693,657015,645623,533384,528777,514702,492002,486426,
  443592,349182),
linerate=c(37.65,18.48,14.50,14.61,16.47,16.07,13.82,13.05,13.78,12.25,10.53,
```

```
  14.18,12.83,7.81,5.17,11.08,6.58,8.77,6.03,6.77),
row.names=c("WSJ","NY Daily News","USA Today","LA Times","NYT", "NY Post",
  "Philadelphia","Chi Tribune","Wash Post","SF Chronicle","Chi Sun Times",
  "Detroit News","Detroit Free Press","Long Island Newsday","KC Times",
  "Miami Herald","Cleveland","Milwaukee","Houston","Baltimore"))
dat$milline=1000000*dat$linerate/dat$circ
```

Multiple Linear Regression

Multiple linear regression allows us to study the relationship between a response variable and *multiple* predictor variables.

- x_{ij}: value of the j^{th} predictor variable on observation i ($i = 1, \ldots, n$ and $j = 1, \ldots, p$). $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)'$ is an $n \times p$ matrix with rows $\mathbf{x}_i = (1, x_{i1}, \ldots, x_{i,p})'$
- y_i: value of dependent variable. $\mathbf{y} = (y_1, \ldots, y_n)'$.
- β_0: the *intercept*. When $x = 0$, on average $Y = \beta_0$.
- β_j: the *slope coefficient for variable* j. A unit increase in x_j is associated with an increase in y of β_j, *on average*.
- Model:

  $$y_i = \beta_0 + x_{i1}\beta_1 + \cdots + x_{ip}\beta_p + e_i = \mathbf{x}_i'\boldsymbol{\beta} + e_i$$

  where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)'$ and e_i is normally distributed with mean 0 and standard deviation σ_e. This implies

  $$E(Y|x_i) = \beta_0 + x_{i1}\beta_1 + \cdots + x_{ip}\beta_p$$

- σ_e: *standard deviation of the errors*.
- σ_e^2 is called the *error variance*.

Estimating the Regression Model

- We don't know values of *parameters* β_0, $\beta_1, \ldots, \beta_p$, and σ_e.
- Estimates denoted by $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$, and S_e
- $E(Y_i|x_i)$ estimates called *fitted*, *predicted*, or *"y-hat" values*:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip}$$

- The *residual* for observation i is

$$\hat{e}_i = y_i - \hat{y}_i = y_i - (\hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_{ij})$$

- We choose $\mathbf{b}$ to minimize the *least-squares criterion* (RSS means *residual sum of squares*):

$$\text{RSS} = \sum_{i=1}^{n} \hat{e}_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})'(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})$$

- The *ordinary least squares* (OLS) estimates of $\boldsymbol{\beta}$:

$$\frac{\partial \text{RSS}}{\partial \hat{\boldsymbol{\beta}}} = -2\mathbf{X}'(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) \quad \Longrightarrow \quad \hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

- Predicted values are given by

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{H}\mathbf{y}$$

where $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ is called the *hat matrix*

Some Properties of OLS Estimates

- $\hat{\boldsymbol{\beta}}$ is unbiased

$$E(\hat{\boldsymbol{\beta}}) = E((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}) = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'E(\mathbf{X}\boldsymbol{\beta} + \mathbf{e}) = \boldsymbol{\beta}$$

- $V(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$

$$V((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}) = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'V(\mathbf{y})\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$$

- Gauss-Markov Theorem: $\hat{\boldsymbol{\beta}}$ has a covariance matrix that is "smaller" than that of any other linear estimator and is called the *best linear unbiased estimator* ("BLUE").

- Estimate σ_e^2 with the *mean squared error*

$$S_e^2 = \text{MSE} = \frac{\text{RSS}}{n - p - 1}$$

and σ_e with the *residual standard error* (a.k.a. *root mean squared error*) $S_e = \sqrt{S_e^2}$.

- If $\mathbf{e} \sim N(0, \sigma^2\mathbf{I})$ and the linear model is correct, then $\hat{\boldsymbol{\beta}}$ has a multivariate normal distribution because it is a linear transformation of normally distributed $\mathbf{e}$:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}) = \boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{e}$$

- $S_{\hat{\beta}_j} = S_e\sqrt{v_j}$ is called the *standard error* of $\hat{\beta}_j$, where v_j is the j^{th} diagonal element of $(\mathbf{X}'\mathbf{X})^{-1}$, and $(b_j - \beta_j)/S_{b_j}$ has a t distribution.

- A $(1-\alpha)\%$ confidence region for $\boldsymbol{\beta}$ is

$$(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})'\mathbf{X}'\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \leq \hat{\sigma}^2\chi_{p+1}^{2(1-\alpha)} \tag{3.15}$$

Process for Building a Regression Model

Suppose that the model is correct (i.e., the dependent variable y is really a linear function of the specified x variables plus additive, normal, independent, homoscedastic errors)

1. Inspect your data for outliers, typos, missing values, etc.
   - Generate n, means, mins, and maxs of each variable
   - Generate boxplots or histograms of each variable
   - Generate a scatterplot matrix for small data sets
   - Generate correlation matrix to assess correlations between predictor variables and pairwise correlations with DV
2. Estimate model and check normality and shape of residuals
3. Test *overall significance of the model*

   $H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$

   H_1 : at least one $\beta_j \neq 0$
4. If you can reject H_0 in Step 3, interpret model and test significance of individual coefficients. If you cannot reject H_0 in Step 2, don't try to test individual coefficients.

In practice you usually will not know the correct model, which complicates the process substantially.

Estimates from Click Ball Point Pens

```
> fit = lm(sales ~ ad + reps + eff, click)
> summary(fit)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   31.150     34.175   0.911    0.368
ad            12.968      2.737   4.738 3.34e-05 ***
reps          41.246      7.280   5.666 1.95e-06 ***
eff           11.524      7.691   1.498    0.143
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 44.42 on 36 degrees of freedom
Multiple R-squared: 0.8812,Adjusted R-squared: 0.8714
F-statistic: 89.05 on 3 and 36 DF,  p-value: < 2.2e-16
```

- Is the regression significant? *Solution:*

  $H_0 : \beta_1 = \beta_2 = \beta_3 = 0$ versus H_1: at least one $\beta_j \neq 0$.
  $P = 2.2e-16 < .05$, so reject H_0 and conclude that at least one of the predictors is predictive.

- State estimated regression equation. *Solution:*

$$\hat{y} = 31.15 + 12.97\texttt{ad} + 41.25\texttt{reps} + 11.52\texttt{eff}$$

- Interpret the coefficient for `reps` (41.25).

Estimates from Click Ball Point Pens (Continued)

- Construct at 95% CI for `reps`.
  Solution: $41.25 \pm 2.028 \times 7.28 = [26.5, 56.0]$

```
> confint(fit)
                 2.5 %    97.5 %
(Intercept) -38.159815 100.46059
ad            7.416798  18.51953
reps         26.480882  56.01037
eff          -4.074175  27.12268
```

- Is `eff` different from zero (use .05 level)?

  $H_0 : \beta_3 = 0$ versus $H_1 : \beta_3 \neq 0$.

$$\begin{aligned} P(b_3 > 11.52) &= P\left(T_{36} > \frac{11.52 - 0}{7.6912}\right) \\ &= P(T_{36} > 1.498) = 0.0714 \end{aligned}$$

  The P value is thus `2*(1-pt(1.498, 36)) = 0.1429`. We cannot reject H_0 because $0.1428 > 0.05$ and we cannot conclude that wholesaler efficiency affects sales.

- Predict sales for `ad`=4, `reps`=3, `eff`=1. *Solution:*

$$\hat{y} = 31.15 + 12.97 \times 4 + 41.25 \times 3 + 11.52 \times 1 = 218.3$$

```
> predict(fit, data.frame(ad=4, reps=3, eff=1))
       1
218.2842
```

Your Turn

1. A commercial real estate company evaluates vacancy rates, square footage, rental rates, and operating expenses for commercial properties in a large metropolitan area in order to provide clients with quantitative information upon which to make rental decisions. The data in `commercial.txt` are taken from 81 suburban commercial properties that are the newest, best located, most attractive, and expensive for five specific geographic areas. It includes their age (`x1`), operating expenses and taxes (`x2`), vacancy rates (`x3`), total square footage (`x4`), and rental rates (`y`).

   (a) Read the commercial data set into R and run basic descriptive statistics (counts, mins, maxs, means). Do the descriptives make sense? Hint:

   ```
   > comm = read.table("c:/teach/304/data/commercial.txt", header=T)
   > summary(comm)
   ```

   (b) Produce a scatterplot matrix and correlation matrix. Discuss the relationships between the variables. Hint: use `plot(comm)` and `cor(comm)`.

   (c) Regress rental rates on the four predictor variables. State the estimated regression equation. Hint:

   ```
   > fit = lm(y ~ x1 + x2 + x3 + x4, comm)
   > summary(fit)
   ```

   (d) Obtain the residual plot study the distribution of residuals. Does the distribution to be fairly symmetrical? Hint: `plot(fit, which=1)`

   (e) Test whether the overall model is significant. State the null and alternative, P-value and decision. Hint: `summary(fit)`

   (f) What fraction of variation in rental rates is explained by these predictor variables?

   (g) If the overall regression model is significant in the previous part then test whether each of the individual regression coefficients equals 0 at the 5% level against a 2-sided alternative, i.e., $H_0 : \beta_j = 0$ for $j = 1, 2, 3, 4$. For each predictor, state the P-value and your decision.

   (h) Assume that the regression model you have estimated is appropriate. Three properties with the following characteristics did not have any rental information available.

	Property 1	Property 2	Property 3
`x1`	4	6	12
`x2`	10	11.5	12.5
`x3`	0.1	0	0.32
`x4`	80,000	120,000	340,000

   Predict the rental rate and compute separate prediction intervals for the rental rates using 95% confidence. **Briefly tell what the prediction interval tells you.** Hint:

```
> newx = data.frame(x1=c(4,6,12), x2=c(10,11.5,12.5), x3=c(.1,0,.32),
  x4=10000*c(8,12,34))
> predict(fit, newx, interval="prediction")
```

2. In a small-scale experimental study of the relation between degree of brand liking (`y`) and the moisture content (`moisture`) and sweetness (`sweetness`) of the product, the results in `brand.txt` were obtained from the experiment based on a completely randomized design.

   (a) Read the brand data set into R and run basic descriptive statistics (counts, mins, maxs, means). Do the descriptives make sense?

   (b) Produce a scatterplot matrix and correlation matrix. Discuss the relationships between the variables.

   (c) Regress liking on the two predictor variables. State the estimated regression equation.

   (d) Obtain the residual plot study the distribution of residuals. Does the distribution to be fairly symmetrical?

   (e) What fraction of variation in rental rates is explained by these predictor variables?

   (f) Test whether the overall model is significant. State the null and alternative, P-value and decision.

   (g) If the overall regression model is significant in the previous part then test whether each of the individual regression coefficients equals 0 at the 5% level against a 2-sided alternative, i.e., $H_0 : \beta_j = 0$ for $j = 1, 2$. For each predictor, state the P-value and your decision.

   (h) Assume that the regression model you have estimated is appropriate. Predict the liking when `moisture=5` and `sweetness=4`. Find a prediction interval and separately a confidence interval for the estimated mean value using 99% confidence.

Answers

1. (a) The summary statistics make sense. Age ranges from 0 to 20, which is reasonable for real estate properties. Operating expenses and taxes are positive. The vacancy rate is between 0 and 1. Square footage looks reasonable, as do rental rates. (b) The first thing to note is the correlations with y. The older the property, the lower the rent. The higher the expenses, the higher the rent. Vacancy rate has a positive correlation, but it is very weak. Square footage has a positive correlation with rent. There are also correlations among the predictor variables, especially between age and expenses (positive, size and expenses (positive), and expenses and vacancy rate (negative). (c) $\hat{y} = 12.2$ -0.142 `x1` $+0.282$ `x2` $+0.619$ `x3` $+0.00000792$ `x4`. (d) The residual plot shows no patterns. (e) $H_0 : \beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$ versus H_1 : at least one $\beta_j \neq 0$. $P = 7.27 \times 10^{-14} < .05$ so reject H_0. At least one predictor is related to rental rate. (f) $R^2 = .5847$. (g) $H_0 : \beta_j = 0$ versus $H_1 : \beta_j \neq 0$. `x1`, `x2`, and `x4` all have $P < .05$. `x3` has $P = .57$ and we cannot reject the null or conclude that vacancy rate has an effect on the rental rate. (h) Property 1: 15.15, 12.85, 17.44; Property 2: 15.54, 13.25, 17.84; Property 3: 16.91, 14.53, 19.29.

2. (a) They make sense. All variables are positive. (b) Moisture has a stronger positive correlation with liking than sweetness. There is no correlation between sweetness and moisture because the data are from an experiment with an orthogonal design. (c) $\hat{y} = 37.650 + 4.425\texttt{moisture} + 4.375\texttt{sweetness}$. (d) There may be an inverted-U shaped relationship, but the fit is not bad. (e) $R^2 = .9521$. (f) $H_0 : \beta_1 = \beta_2 = 0$ versus H_1 : at least one $\beta_j \neq 0$. $P = 2.658 \times 10^{-9} < .05$ so reject H_0. At least one predictor is related to liking. (g) $H_0 : \beta_j = 0$ versus $H_1 : \beta_j \neq 0$. `moisture` and `sweetness` have $P < .05$. (h) Use `predict(fit, data.frame(moisture=5, sweetness=4), interval="conf", level=.99)` to find 77.275, 73.88, 80.67. Use `predict(fit, data.frame(moisture=5, sweetness=4), interval="pred", level=.99)` to find 77.275, 68.48, 86.07.

Comparing Regression Coefficients

- *Question:* Which of the variables is more "important" in explaining sales?
- *Answer:* The coefficients are not directly comparable because of differences in units of measurement.
- *Ideal solution:* convert to commensurate units, e.g., dollars.
- *Possible solution:* Use *standardized regression coefficients* (all variable standardized before the analysis to have mean 0 and variance 1). The "unit" of measurement is now the standard deviation

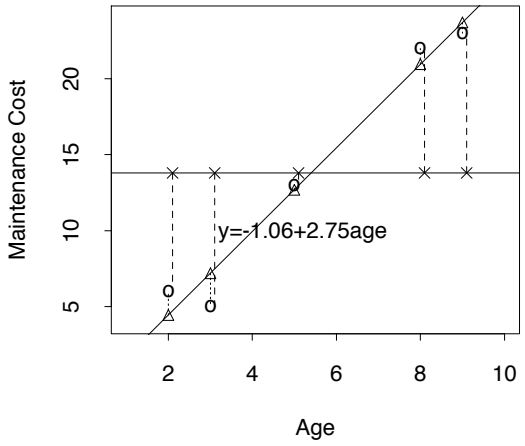
```
> Zclick = as.data.frame(scale(click[,1:4]))
> fit = lm(sales ~ ad + reps + eff -1, Zclick)  # -1 drops intercept
> summary(fit)
Coefficients:
     Estimate Std. Error t value Pr(>|t|)
ad    0.45101    0.09390   4.803 2.59e-05 ***
reps  0.54902    0.09559   5.744 1.40e-06 ***
eff   0.09157    0.06028   1.519    0.137
```

- *Possible solution:* compare t scores ($t = b_j/S_{b_j}$, reported in the fifth column of output)
- See Bring (1994) for discussion

Standardized Regression Coefficients

- Theorem: the standardized regression coefficient is $b_j S_{x_j}/S_y$, where S_{x_j} and S_y are the standard deviations of x_j and y, respectively, and b_j is the (unstandardized) regression estimate for variable j.
- Standardized regression coefficients are sometimes called "*beta*" coefficients
- Interpretation: a standard deviation increase in x_j is associated with "beta" standard deviations in y — the units of measurement are now the standard deviation.
- In simple linear regression, "beta" is the correlation r between x and y.
- Do not use "beta" coefficients blindly:
  - If x_j is a 0-1 variable, then the standard deviation is $\sqrt{\bar{x}_j(1-\bar{x}_j)}$
  - If you are analyzing a designed experiment, you select the x_j values and thus the standard deviation
  - "Beta" values are still a function of other variables in the model
  - "Beta" values do not consider costs

Machine example



- Regression model

$$y_i = \beta_0 + \beta x_i + e_i,$$

where e_i has mean 0 and variance σ^2. Note that the variance of e_i does not depend on i; this is called *homoscedastic error variance*. We also assume that e_i is independent of e_j for $i \neq j$. If you intend to do hypothesis tests or compute confidence intervals, either the distribution of e_i must be normal or the sample size must be large.

- Null model (overall mean of y) (horizontal line in figure)

$$\bar{y} = \frac{69}{5} = 13.8$$

The $\times$ symbols on the figure show the "fitted values" for the null model.

- Estimated regression equation (diagonal line in figure)

$$\hat{y}_i = -1.06 + 2.75 x_i$$

the triangle symbols on the plot show the fitted values $\hat{y}_i$.

	`age`	`cost`		RSS	MSS	TSS
i	x_i	y_i	$\hat{y}_i$	$(y_i - \hat{y}_i)^2$	$(\hat{y}_i - \bar{y})^2$	$(y_i - \bar{y})^2$
1	2	6	4.44	2.43	87.59	60.84
2	5	13	12.70	0.09	1.21	0.64
3	9	23	23.71	0.50	98.20	84.64
4	3	5	7.19	4.80	43.65	77.44
5	8	22	20.96	1.08	51.22	67.24
Sum	27	69	69	8.92	281.88	290.8

- Sum of squared errors:

$$\text{RSS} = \sum_{i=1}^{5}(y_i - \hat{y}_i)^2 = 8.9247$$

  The dotted lines show the errors $y_i - \hat{y}_i$.

- Regression/Model sum of squares

$$\text{MSS} = \sum_{i=1}^{5}(\hat{y}_i - \bar{y})^2 = 281.88$$

- Total sum of squares

$$\text{TSS} = \sum_{i=1}^{5}(y_i - \bar{y})^2 = 290.8$$

  The dashed lines show the errors $y_i - \bar{y}$ under the null model.

- Note that the ANOVA equality holds:

$$\text{RSS} + \text{MSS} = \text{TSS} \Longrightarrow 8.92 + 281.88 = 290.8$$

- The sums of squares are converted into variance estimates by dividing by the appropriate degrees of freedom. The degrees of freedom for the model (regression) equal p, the number of predictor variables in the model, where the intercept is not included. The error degrees of freedom equal $n - p - 1$.

	Df	Sum of Sq	Mean Sq	F Value	Pr(F)
Model	1	281.8753	281.8753	94.75	0.0023
Error	3	8.9247	2.9749		
Total	4	290.8000			

- The *root mean squared error* (or *standard error of the regression*) is $\sqrt{2.9749} = 1.72479$, which estimates σ. The mean squared error (2.9749) estimates σ^2.

- The *coefficient of determination*, or R^2, gives the fraction of variation explained by the model:

$$R^2 = \frac{\text{MSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}} = \frac{281.8753}{290.8} = 96.93\%$$

- The F statistic is computed as follows

$$F = \frac{281.8753}{2.9749} = 94.75$$

  The corresponding P-value (.0023) tests "the overall significance of the model," i.e., H_0 : all $\beta = 0$ versus H_1 : at least one $\beta \neq 0$. The larger F, the more evidence against H_0. If you cannot reject H_0, you cannot conclude that the regression model does an better than the null model.

IEMS 304: Homework 1
Due: April 6, 12:30 pm
Professor Malthouse

You may work in self-selected groups of at most four. Turn in one copy per group, with all names on it.

1. JWHT problem 8a,b on pages 121–2 (Hint: see §2.3.4 on page 48–49.) Click here for data. Note: omit part c for now.

2. JWHT problem 9(a)–(d) on page 122

3. Consider the regression model $y_i = \alpha + \beta x_i + e_i$, where e_i are independent random variables with $E(e_i) = 0$ and $V(e_i) = \sigma^2$ for all i.

   (a) What is the implication for the regression function if $\beta = 0$, so that the model is $y_i = \alpha + e_i$? How would the regression function plot on a graph?

   (b) Derive the least square estimator a of α for the model above (with $\beta = 0$) and show that it equals the sample mean $a = \bar{y}$.

   (c) Prove that the estimate a in the previous part is an unbiased estimator of α.

   (d) What is the variance of your estimate a?

   (e) Discuss why your estimates are (at least approximately) normally distributed.

   (f) The Gauss-Markov theorem states that OLS estimates are best linear unbiased estimates ("BLUE"), i.e., among all linear, unbiased estimates, the OLS estimates have the smallest variance. Show that your esimate from part (b) is BLUE. Hints: Let $\hat{\alpha} = \sum_{i=1}^{n} c_i y_i$ be another linear (it is a linear combination of y_i) unbiased estimate, where c_i are constants. Let $d_i = c_i - 1/n$ be the difference between the constants of the new estimator and those from OLS $(1/n)$. Show that $d_i = 0$ for all i, otherwise the variance will be greater than that of $\bar{y}$ from part (d). When $d_i = 0$ the new estimates are the same as the OLS ones.

      i. What does the unbiased assumption imply about the sum of c_i?
      ii. Show $\sum_i d_i/n = 0$.
      iii. Evaluate $V(\hat{\alpha})$ in terms of d_i and find when it is minimized over the d_i values.
      iv. Or you can think geometrically.

Newfood case

Mr. Conrad Ulcer, newly appointed New Products Marketing Director for Concorn Kitchens, was considering the possibility of marketing a new highly nutritional food product with widely varied uses. This product could be used as a snack, a camping food, or as a diet food. The product was to be generically labeled Newfood.

Because of this wide range of possible uses, the company had great difficulty in defining the market. The product was viewed as having no direct competitors. Early product and concept tests were very encouraging. These tests led Mr. Ulcer to believe that the product could easily sell 2 million cases (24 packages in a case) under the proposed marketing proposal involving a 24-cent package price and an advertising program involving $3 million in expenditures per year. There were no capital expenditures required to go national, since manufacturing was to be done on a contract-pack basis.

Because there was considerable uncertainty among Concorn Management as to either probable first-year and subsequent-year sales, or the best introductory campaign, Ulcer decided that a six-month market test would be conducted. The objectives of the test were to:

- Better estimate first-year sales.
- Study certain marketing variables to determine an optimal — or at least better — introductory plan.
- Estimate the long-run potential of the product

These objectives were accomplished through the controlled introduction of the product into four markets. Conditions were experimentally varied within the grocery stores in each of the four markets. Sales were measured with a store audit of a panel of stores. Preliminary results had been obtained. Now it was up to Mr. Ulcer to understand their implications on the introduction of Newfood.

Design of Experimental Study

The three variables included in the experimental design were price, advertising expenditures, and location of the product within the store. Three prices were tested (24 cents, 29 cents, and 34 cents), two levels of advertising (a simulation of a $3 million introduction and a $6 million plan), and two locations (placing the product in the bread section versus the instant breakfast section). Prices and location were varied across stores within cities while advertising was varied across cities. The advertising was all in the form of TV spots. The levels were selected so that they would stimulate on a local basis the impact that could be achieved from national introduction programs at the $3 million and $6 million expenditure levels. Due to differential costs between markets and differential costs between spot and network (to be used in national introduction), an attempt was made to

equate (and measure) advertising inputs of gross advertising impressions generated, normalized for market size. Unfortunately, it was not possible to achieve exactly the desired levels. This was due to the problem of non-availabilities of spots in some markets and discrepancies between estimates of TV audiences made at the time the test was being planned and the actual audiences reached at the time the commercials were actually run.

In the selection of cities and stores for the tests, attempts were made to match stores on such variables as store size, number of checkout counters, and characteristics of the trading area. Because it was not certain that adequate matches had been achieved, Ulcer decided to obtain measurements on some of these variables for possible use in adjusting for differences in cell characteristics. He also felt that it might be possible to learn something about the relationships between these variables and sales, and that this information would be of assistance in planning the product introduction into other markets.

```
newfood = data.frame(
  sales=c(225,323,424,268,224,331,254,492,167,226,210,289,204,288,245,161,161,
    246,128,154,163,151,180,150),
  price=c(24,24,24,24,24,24,24,24,29,29,29,29,29,29,29,29,34,34,34,34,34,34,34,34),
  ad=c(0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1),
  loc=c(0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1),
  income=c(7.3,8.3,6.9,6.5,7.3,8.3,6.9,6.5,6.5,8.4,6.5,6.2,6.5,8.4,6.5,6.2,
    7.2,8.1,6.6,6.1,7.2,8.1,6.6,6.1),
  volume=c(34,41,32,28,34,41,23,37,33,39,30,27,37,43,30,19,32,42,29,24,32,36,29,24),
  city=c(3,4,1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,4,1,2))
```

Questions for class discussion

1. Compute the correlation matrix. How do you explain the 0 correlations (e.g., between location and advertising)?

2. Run a regression of `sales` (the first two months sale) on `price` alone. Next, on `price` and `ad`. Finally on `price`, `ad`, and `loc`. Thus, you will have three regressions. What happens to the coefficients of `price` in the three regressions? What happens to the coefficients of `ad` in the two regressions? Explain.

3. Run a regression of `sales` against `price`, `ad`, `loc`, and `volume`. What happens to the coefficients of `price`, `ad`, and `loc` which you found in the third regression in question 2 above? Which coefficient changes the most with the introduction of store size? Why does this happen?

4. Finally, run a regression of `sales` against `price`, `ad`, `loc`, `volume`, and `income`. What changes do you observe between these results and that of the fourth regression? Explain.

5. What additional regression runs, if any, should be made to complete the analysis of these data?

Effects of Model Misspecification: Omitting Relevant Predictor

- Suppose we fit the model

$$y = \beta_0 + x\beta_1 + e$$

- But the true model is

$$y = \beta_0 + x\beta_1 + z\beta_2 + e$$

- Theorem: If $\mathbf{x}_1$ and $\mathbf{x}_2$ are correlated, b_1 will be biased, with the direction of the bias depending on the sign of the correlation between $\mathbf{x}_1$ and $\mathbf{x}_2$ and the sign of β_2.

$$E(b_1) = \beta_1 + \beta_2 r_{xz} \frac{S_z}{S_x}$$

where S_x and S_z are the sample standard deviations of x and z

Direction of Bias in b_1

Sign of correlation between x and z	if $\beta_2 > 0$	if $\beta_2 < 0$
$\text{corr}(x, z) = r_{xz} > 0$	Upward bias	Downward bias
$\text{corr}(x, z) = r_{xz} = 0$	No change	No change
$\text{corr}(x, z) = r_{xz} < 0$	Downward bias	Upward bias

- Note that if x and z are uncorrelated (orthogonal design), we can add or drop variables without changing the other coefficients.

1. Definition and Effects

- Definition: predictor variables are highly correlated with one another (Note: Y is not mentioned here). When one or more X's move together, it is difficult to sort out their separate effects.
- What are some effects? You cannot sort out what is doing what.
  - *Unstable coefficients.* High estimated standard errors for one or more slope coefficients.
    - Implication: Low t-ratio, so sometimes we cannot reject the null hypothesis that $\beta = 0$. At the extreme, the model as a whole may be significant, while none of the individual slope parameters are!
    - Coefficients can change wildly when variables are added or dropped from the model.
  - *Incorrect signs.* Slope estimates can have signs that are not consistent with intuition.
- Note: predicted values not affected directly

Predicted Values Unaffected

- Suppose the true model is

$$y = 2x_1 + x_2$$

- We have $n = 3$ observations:

x_1	x_2	y	e
1	2	4	0
2	4	8	0
3	6	12	0

  Note that $x_2 = 2x_1$, so that the two columns are perfectly correlated

- The true model fits perfectly, but so do many others, e.g.,

$$y = 4x_1 + 0x_2 \Longrightarrow x_2 \text{ no effect}$$

$$y = 0x_1 + 2x_2 \Longrightarrow x_1 \text{ no effect}$$

$$y = 6x_1 - x_2 \Longrightarrow x_2 \text{ negative effect}$$

  Predictions correct for all these choices of parameter estimates, but substantive interpretation completely different!

- Which (b_1, b_2) is correct? We can't tell from these data.

- What happens if we use our fitted model to *extrapolate*? e.g., estimate y for $x_1 = 1$ and $x_2 = 1$. The correct answer is 3, but the other models gives estimates 4, 2, and 5, respectively. **Extrapolation is especially questionable when using a model estimated from multicollinear data.**

2. Detecting Multicollinearity

- Compute a correlation matrix of the predictor variables and possibly a scatterplot matrix. Large correlations indicate multicollinearity could be a problem.
- Unstable coefficients or incorrect signs
- *Tolerance* is $1 - R^2$ from this regression (fraction of variance *unexplained* by the model)
- *Variance inflation factor* (VIF) is 1/tolerance.

```
> install.packages("car")    # do this only once
> library(car)    # you must first download it from CRAN
> fit = lm(sales~price+ad+loc+volume+income, newfood)
> vif(fit)
   price       ad      loc   volume   income
1.079882 2.697664 1.005447 3.447143 3.367158


> # where does VIF for ad come from?
> summary(lm(ad ~ price + loc+volume+income, newfood))

...
Multiple R-squared:  0.6293,Adjusted R-squared:  0.5513

> 1/(1-.6293)
[1] 2.697599
```

Living with Multicollinearity

1. If the objective is primarily to make good *predictions*, then you could do nothing, although you may be better off using forward/backward/stepwise regression, ridge regression, principal components regression (PCR), or other related techniques. Data mining applications usually fall into this category.
2. If the objective is to *interpret* regression coefficients, then
   (a) If possible, avoid multicollinearity with an *orthogonal* design, where predictor variables are uncorrelated
   (b) Understand why you have multicollinearity
      - Predictors causally related, e.g., $x_1 \rightarrow x_2$. Fit correct model.
      - Predictors manifestations of common, underlying latent construct, e.g., $w \rightarrow x_1$ and $w \rightarrow x_2$. Often, estimate w and use it instead of x_1 and x_2.
      - Correlated decision/causal variables. Include controls, perhaps use shrinkage.
   (c) Use *shrinkage* estimation
      - Principal components regression (PCR), partial least squares (PLS), using factor scores as predictors
      - Ridge regression

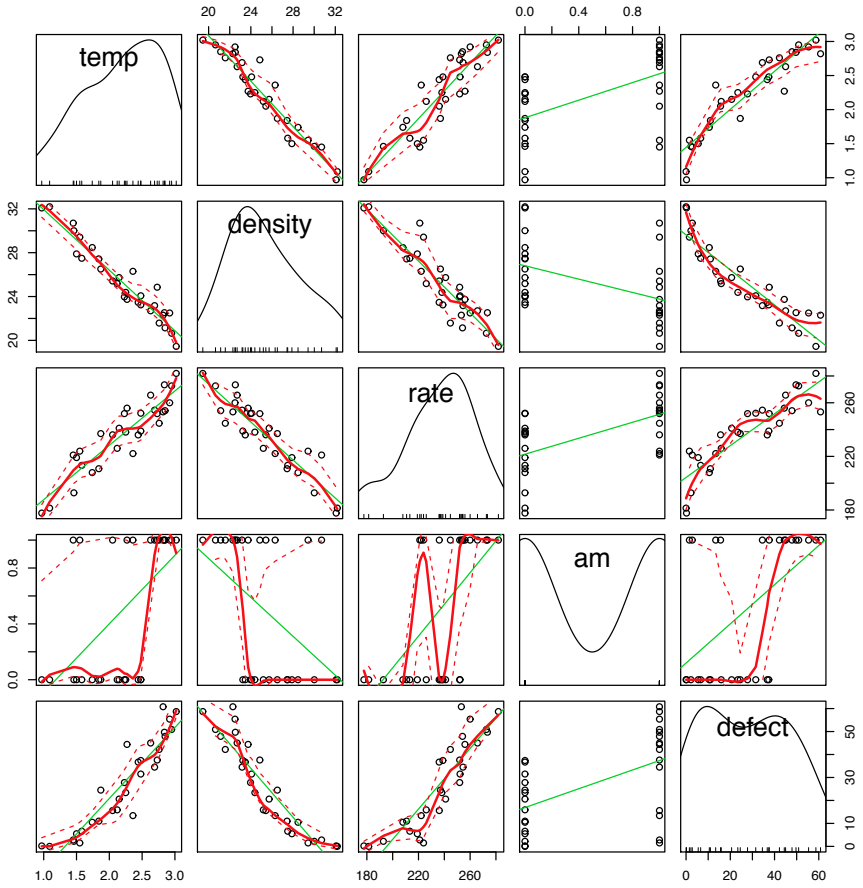
Model Specification Issues

- For successful use of multiple regression, sufficient knowledge *about the subject domain* is required to identify relevant predictor variables and their functional relationship with the dependent variable.
- If the only goal is predictive accuracy, it should make sense for a variable to be in the model, e.g., does it make sense to have customer ID number as a predictor variable? The variable should be a "cause" of the dependent variable.
- If the goal is interpretation, include a variable if
  - The variable is important in making a managerial decision, e.g., type of display in a promotional response model for a retailer.
  - The variable helps to control for important causal factors, e.g., seasonality or competitive actions.

Scatterplots with CAR

The car package offers improved scatterplot matricies:

```
> library(car) # do this if you haven't ready done so
> scatterplotMatrix(~temp+density+rate+am+defect, quality)
```



- The red lines are *smoothers*, which trace the middle of the distribution of the vertial variable conditional on the horizontal variable. Green lines are robust regression lines.
- The ticks are called a *rug* and show individual observations.
- The graphs on the diagonal are density estimates (like a histogram).

Quality Control Case

Everybody seems to disagree about just why so many parts have to be fixed or thrown away after they are produced. Some say that it's the temperature of the production process, which needs to be held constant (within a reasonable range). Others claim that it's clearly the density of the product, and that if we could only produce a heavier material, the problems would disappear. Then there is Ole, who has been warning everyone forever to take care not to push the equipment beyond its limits. This problem would be the easiest to fix, simply by slowing down the production rate; however, this would increase costs. Interestingly, many of the workers on the morning shift think that the problem is "those inexperienced workers in the afternoon," who, curiously, feel the same way about the morning workers.

Ever since the factory was automated, with computer network communication and bar code readers at each station, data have been piling up. You've finally decided to have a look. After your assistant aggregated the data by 4-hour blocks and then typed in the AM/PM variable, you found the following description of the variables:

- `temperature`: measures the temperature variability as a standard deviation during the time of measurement
- `density`: indicates the density of the final product
- `rate`: rate of production
- `am`: 1 indicates morning and 0 afternoon
- `defect`: average number of defects per 1000 produced

Discussion Questions

1. Generate a scatterplot matrix and a correlation matrix. Interpret the correlations. What "obvious conclusions" can you draw?
2. Run a multiple regression predicting defect rate from the other four variables. Is the overall model significant? Which predictors, if any, are significant? Compute and interpret variance inflation factors. What "obvious conclusions" can you draw?
3. Predict defect from each of the predictor variables separately, e.g., `defect` from `temp`, `defect` from `density`, `defect` from `rate`, etc. Which of the predictors are significant in the simple linear regressions?
4. Would it be appropriate to use stepwise regression?
5. Would it be appropriate to use principal components regression?

6. Would it be appropriate to regress `defect` on the first component?

7. Perform further analysis as needed. What action do you recommend? Why?

8. To compare the two shifts, would it be appropriate to perform an independent-sample t-test (i.e., H_0 : AM defect rate = PM defect rate)? How is this different from the multiple regression approach? Which is preferred? Discuss.

9. How would you present your findings to a client?

```
quality = data.frame(
  temp=c(.97,2.85,2.95,2.84,1.84,2.05,1.5,2.48,2.23,3.02,2.69,2.63,1.58,2.48,2.25,
    2.76,2.36,1.09,2.15,2.12,2.27,2.73,1.46,1.55,2.92,2.44,1.87,1.45,2.82,1.74),
  density=c(32.08,21.14,20.65,22.53,27.43,25.42,27.89,23.24,23.97,19.45,23.17,
    22.7,27.49,24.07,24.38,21.58,26.3,32.19,25.73,25.18,23.74,24.85,30.01,
    29.42,22.5,23.47,26.51,30.7,22.3,28.47),
  rate=c(177.7,254.1,272.6,273.4,210.8,236.1,219.1,238.9,251.9,281.9,254.5,265.7,
    213.3,252.2,238.1,244.7,222.1,181.4,241,226,256,251.9,192.8,223.9,260.0,236,
    237.0,221,253.2,207.9),
  am=c(0,1,1,1,0,1,0,0,0,1,1,1,0,0,0,1,1,0,0,0,1,1,0,1,1,0,0,1,1,0),
  defect=c(.2,47.9,50.9,49.7,11,15.6,5.5,37.4,27.8,58.7,34.5,45,6.6,31.5,23.4,
    42.2,13.4,0,20.6,15.9,44.4,37.6,2.2,1.5,55.4,36.7,24.5,2.8,60.8,10.5)
)
```

Measuring model performance and variable importance

- The *full model* is $y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + e$.
- The variation left unexplained by the full model is given by the *residual sum of squares* or *deviance*:

$$\text{RSS} = \sum_{i=1}^{n} \hat{e}_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- When $\beta_1 = \cdots = \beta_p = 0$ we call it the *intercept* or *null* model, and it turns out $\hat{y}_i = \bar{y}$, the mean of y. The variation left unexplained by the null model is the *total sum of squares*:

$$\text{TSS} = \sum_{i=1}^{n} (y_i - \bar{y})^2 = (n-1)S_y^2$$

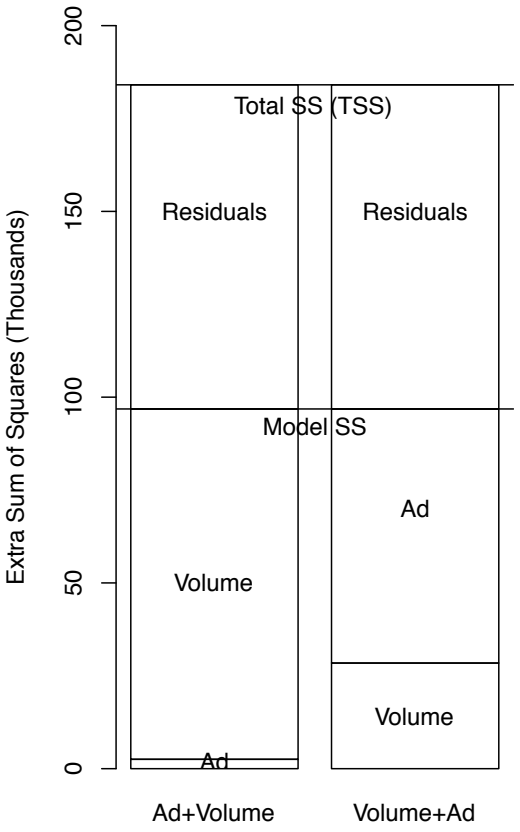
- Usually the full model explains more than the null model

```
> deviance(lm(sales~1, newfood))  # null model leaves 184K unexplained
[1] 184066

> deviance(lm(sales~ad, newfood))  # model with ads leaves 182K unexplained
[1] 181544.5

> deviance(lm(sales~ad+volume, newfood)) # model with both leaves 87K unexplained
[1] 87246.89
```

- We can say that `ad` explains $184066 - 181544.5 = 2,521.5$.
- `ad` and `volume` together explain $184066 - 87247 = 96,819$.

The anova and drop1 commands

```
> attach(newfood)
> var(sales)*(nrow(newfood)-1) # TSS
[1] 184066
> sum((sales-mean(sales))^2) # TSS
[1] 184066

> fit = lm(sales ~ ad + volume, newfood)
> anova(fit)  # extra SS
Analysis of Variance Table
Response: sales
          Df Sum Sq Mean Sq F value Pr(>F)
ad         1   2521    2521  0.6069 0.4446
volume     1  94298   94298 22.6971 0.0001
Residuals 21  87247    4155

> drop1(fit)  # partial SS
Single term deletions
Model: sales ~ ad + volume
       Df Sum of Sq    RSS
<none>               87247
ad      1     68391 155638
volume  1     94298 181544


> fit2 = lm(sales ~ volume+ad, newfood)

> anova(fit2)  # extra SS
Analysis of Variance Table
Response: sales
          Df Sum Sq Mean Sq F value Pr(>F)
volume     1  28428   28428  6.8426 0.0161
ad         1  68391   68391 16.4614 0.0006
Residuals 21  87247    4155

> drop1(fit2)  # partial SS
Single term deletions
Model: sales ~ volume + ad
       Df Sum of Sq    RSS
<none>               87247
volume  1     94298 181544
ad      1     68391 155638
```



- TSS = RSS for intercept model = sum of extra SS
- Extra SS (anova): change in SS if term added
- Partial SS (drop1): change in SS if term dropped
- For last term in, Extra SS = Partial SS

The F test of "overall significance"

- Recall how to test the following hypothesis:

  $H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$

  H_1 : at least one $\beta_j \neq 0$

```
> summary(fit)
Call: lm(formula = sales ~ ad + volume, data = newfood)

            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -324.31     116.95  -2.773 0.011396 *
ad            159.25      39.25   4.057 0.000567 ***
volume         14.87       3.12   4.764 0.000105 ***

Residual standard error: 64.46 on 21 degrees of freedom
Multiple R-squared:  0.526,Adjusted R-squared:  0.4809
F-statistic: 11.65 on 2 and 21 DF,  p-value: 0.0003941
```

- This test compares the full model (H_1) with the null model

$$F = \frac{\frac{\text{TSS}-\text{RSS}}{p}}{\frac{\text{RSS}}{n-p-1}} = \frac{\frac{184066-87246.89}{2}}{\frac{87246.89}{21}} = 11.652 = \left(\frac{\frac{\Delta \text{RSS}}{\Delta df}}{S_e^2}\right)$$

- The P-value can be found in R:

```
> 1 - pf(11.652, 2, 21)
[1] 0.0003941411
```

- This foreshadows an important application of the F test

The F test for a single predictor

- Now consider testing $H_0 : \beta_2 = 0$ versus $H_1 : \beta_2 \neq 0$
- Assuming the null is true ($\beta_2 = 0$) we get $y = \beta_0 + \beta_1 x_1 + e$, which will be called the *reduced* model
- The `summary` output give a t test and shows $P = .000105$.
- We can equivalently perform an F test, which will have more general uses later in the course:

```
> anova(fit)
Analysis of Variance Table
Response: sales
          Df Sum Sq Mean Sq F value    Pr(>F)
ad         1   2521    2521  0.6069 0.4446424
volume     1  94298   94298 22.6971 0.0001048 ***
Residuals 21  87247    4155

> drop1(fit, test="F")
Single term deletions
Model: sales ~ ad + volume
       Df Sum of Sq    RSS F value    Pr(>F)
<none>               87247
ad      1     68391 155638  16.461 0.0005666 ***
volume  1     94298 181544  22.697 0.0001048 ***
```

- The following F has 1, 21 df

$$F = \frac{\frac{\Delta \text{RSS}}{\Delta df}}{S_e^2} = \frac{\frac{94298}{1}}{\frac{87247}{21}} = \frac{\frac{94298}{1}}{4155} = 22.6971$$

```
> 1-pf(22.6971, 1, 21)
[1] 0.0001047984
```

- Why is the `ad` not significant in the `anova` output?

Summary of key points

- Model selection involves picking a model between the null and full model
- We use RSS to measure what is unexplained by a model
- The `anova` command generates *extra sum of squares*, telling how much RSS is reduced as we add terms to a model one at a time. They depend on the order of the terms.
- The `drop1` command generates *partial sum of squares*, telling how much RSS is reduced when we drop each term. They do not depend on order.
- The F test allows for hypothesis testing between the full and reduced models

Your Turn

1. Consider the click ballpoint pens data given on page 23

   (a) How much variation is left unexplained by the intercept model? (this will be called the *null deviance*)

   (b) How much variation is explained by adding `ad` to the intercept model?

   (c) How much additional variation is explained by adding `reps` to a model that already has `ad` in it?

   (d) How much additional variation is explained by adding `eff` to a model that already has `ad` and `eff` in it?

   (e) How much variation is unexplained by a model having all three predictors?

   (f) How much less variation is explained if we drop `ad` from a model with all three predictors in it?

   (g) Compute R^2 for the three-predictor model "by hand" using only the numbers you have found above. Confirm your answer by having R compute it.

   (h) Compute adjusted R^2 by hand and confirm it.

   (i) Compute the F statistic for the overall test of significance by hand.

   (j) Compute the F statistic to test $H_0 : \beta_1 = 0$ by hand.

2. Consider the commercial properties data.

   (a) Obtain the analysis of variance table that decomposes the regression sum of squares into extra sums of squares associated with X_4; with X_1 given X_4; with X_2 given X_1 and X_4; and with X_3 given X_1, X_2 and X_4. Hint use the `lm` and `anova` functions.

   (b) Test whether X_3 can be dropped from the regression model given that X_1, X_2 and X_4 are retained. Use the F test statistic and level of significance of .01. State the null and alternative hypotheses, test statistic, P-value and decision.

   (c) Test whether both X_2 and X_3 can be dropped from the regression model given that X_1 and X_4 are retained; use $\alpha = 0.01$. State the null and alternative hypotheses, test statistic, P-value and decision. Hint: use the `pf` function to find P values.

   (d) Find the variance inflation factors for the full model with all four predictors in the model. What do they tell you?

3. Consider the brand problem.

   (a) Find the variance inflation factors for the full model with both predictors in the model. What do they tell you?

   (b) Obtain the analysis of variance table that decomposes the regression sum of squares into extra sums of squares associated with `moisture`; and with `sweetness` given `moisture`?

(c) Obtain the analysis of variance table that decomposes the regression sum of squares into extra sums of squares associated with `sweetness`; and `moisture` given `sweetness`. What do you notice?

(d) Regress liking on moisture content only. How does the estimate of β_1 in the previous part compare with the estimate in the model with both predictors?

Answers

1. (a) TSS = 598253; (b) 463451, Hint: `fit = lm(sales ~ ad + reps + eff, click)` and then `anova(fit0)`; (c) 59327; (d) 4431; (e) 71044; (f) 68391, Hint: `drop1(fit, test="F")`; (g) $1 - 71044/598253 = .8812$, Hint: `summary(fit)`; (h) $1 - (71044/36)/(598253/39) = .8714$; (i) $((598253 - 71044)/3)/(71044/36) = 89.05$; (j) $(44295/1)/(71044/36) = 22.45$, Hint: see `drop1` output.

2. (a) Hint: `fit = lm(y ~ x4+x1+x2+x3, comm)` then `anova(fit)`; (b) $H_0 : \beta_3 = 0$ versus $H_1 : \beta_3 \neq 0$, $P = 0.5704$, we cannot reject H_0. (c) $H_0 : \beta_2 = \beta_3 = 0$ versus $H_1 : \beta_2 \neq 0$ or $\beta_3 \neq 0$. From the output below the P-value is less than 0 and so we reject H_0.

```
> fit2 = lm(y ~ x1+x4, comm)
> 1-pf(((deviance(fit2)-deviance(fit))/2) / (deviance(fit)/76), 2, 76)
[1] 6.682136e-05
```

(d) Hint: `vif(fit)`. The VIF values are 1.41, 1.24, 1.65, and 1.32. All are fairly close to 1 indicating that multicollinearity is not a serious problem.

3. (a) The VIFs are both 1 indicating uncorrelated predictors; (b) The extra SS for moisture are 1566 and for sweeness 306; (c) They are the same as in the previous part; (d) The coefficient is the same.

Multiple Linear Regression

Multiple linear regression allows us to study the relationship between a response variable and *multiple* predictor variables.

- x_{ij}: value of the j^{th} predictor variable on observation i ($i = 1, \ldots, n$ and $j = 1, \ldots, p$). $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)'$ is an $n \times p$ matrix with rows $\mathbf{x}_i = (1, x_{i1}, \ldots, x_{i,p})'$
- y_i: value of dependent variable. $\mathbf{y} = (y_1, \ldots, y_n)'$.
- β_0: the *intercept*. When $x = 0$, on average $Y = \beta_0$.
- β_j: the *slope coefficient for variable* j. A unit increase in x_j is associated with an increase in y of β_j, *on average*.
- Model:

$$y_i = \beta_0 + x_{i1}\beta_1 + \cdots + x_{ip}\beta_p + e_i = \mathbf{x}_i'\boldsymbol{\beta} + e_i$$

  where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)'$ and e_i is normally distributed with mean 0 and standard deviation σ_e. This implies

$$E(Y|x_i) = \beta_0 + x_{i1}\beta_1 + \cdots + x_{ip}\beta_p$$

- σ_e: *standard deviation of the errors*.
- σ_e^2 is called the *error variance*.

Estimating the Regression Model

- We don't know values of *parameters* $\beta_0, \beta_1, \ldots, \beta_p$, and σ_e.
- Estimates denoted by $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$, and S_e
- $E(Y_i|x_i)$ estimates called *fitted*, *predicted*, or *"y-hat" values*:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip}$$

- The *residual* for observation i is

$$\hat{e}_i = y_i - \hat{y}_i = y_i - (\hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_{ij})$$

- We choose $\mathbf{b}$ to minimize the *least-squares criterion* (RSS means *residual sum of squares*):

$$\text{RSS} = \sum_{i=1}^{n} \hat{e}_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})'(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})$$

- The *ordinary least squares* (OLS) estimates of $\boldsymbol{\beta}$:

$$\frac{\partial \text{RSS}}{\partial \hat{\boldsymbol{\beta}}} = -2\mathbf{X}'(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) \quad \Longrightarrow \quad \hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

- Predicted values are given by

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{H}\mathbf{y}$$

where $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ is called the *hat matrix*

Some Properties of OLS Estimates

- $\hat{\boldsymbol{\beta}}$ is unbiased

$$E(\hat{\boldsymbol{\beta}}) = E((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}) = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'E(\mathbf{X}\boldsymbol{\beta} + \mathbf{e}) = \boldsymbol{\beta}$$

- $V(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$

$$V((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}) = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'V(\mathbf{y})\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$$

- Gauss-Markov Theorem: $\hat{\boldsymbol{\beta}}$ has a covariance matrix that is "smaller" than that of any other linear estimator and is called the *best linear unbiased estimator* ("BLUE").

- Estimate σ_e^2 with the *mean squared error*

$$S_e^2 = \text{MSE} = \frac{\text{RSS}}{n - p - 1}$$

and σ_e with the *residual standard error* (a.k.a. *root mean squared error*) $S_e = \sqrt{S_e^2}$.

- If $\mathbf{e} \sim N(0, \sigma^2\mathbf{I})$ and the linear model is correct, then $\hat{\boldsymbol{\beta}}$ has a multivariate normal distribution because it is a linear transformation of normally distributed $\mathbf{e}$:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}) = \boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{e}$$

- $S_{\hat{\beta}_j} = S_e\sqrt{v_j}$ is called the *standard error* of $\hat{\beta}_j$, where v_j is the j^{th} diagonal element of $(\mathbf{X}'\mathbf{X})^{-1}$, and $(b_j - \beta_j)/S_{b_j}$ has a t distribution.

- A $(1-\alpha)\%$ confidence region for $\boldsymbol{\beta}$ is

$$(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})'\mathbf{X}'\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \leq \hat{\sigma}^2\chi_{p+1}^{2(1-\alpha)} \tag{3.15}$$

Process for Building a Regression Model

Suppose that the model is correct (i.e., the dependent variable y is really a linear function of the specified x variables plus additive, normal, independent, homoscedastic errors)

1. Inspect your data for outliers, typos, missing values, etc.
   - Generate n, means, mins, and maxs of each variable
   - Generate boxplots or histograms of each variable
   - Generate a scatterplot matrix for small data sets
   - Generate correlation matrix to assess correlations between predictor variables and pairwise correlations with DV
2. Estimate model and check normality and shape of residuals
3. Test *overall significance of the model*

   $H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$

   H_1 : at least one $\beta_j \neq 0$
4. If you can reject H_0 in Step 3, interpret model and test significance of individual coefficients. If you cannot reject H_0 in Step 2, don't try to test individual coefficients.

In practice you usually will not know the correct model, which complicates the process substantially.

Estimates from Click Ball Point Pens

```
> fit = lm(sales ~ ad + reps + eff, click)
> summary(fit)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   31.150     34.175   0.911    0.368
ad            12.968      2.737   4.738 3.34e-05 ***
reps          41.246      7.280   5.666 1.95e-06 ***
eff           11.524      7.691   1.498    0.143
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 44.42 on 36 degrees of freedom
Multiple R-squared: 0.8812,Adjusted R-squared: 0.8714
F-statistic: 89.05 on 3 and 36 DF,  p-value: < 2.2e-16
```

- Is the regression significant? *Solution:*

  $H_0 : \beta_1 = \beta_2 = \beta_3 = 0$ versus H_1: at least one $\beta_j \neq 0$.
  $P = 2.2e-16 < .05$, so reject H_0 and conclude that at least one of the predictors is predictive.

- State estimated regression equation. *Solution:*

$$\hat{y} = 31.15 + 12.97\texttt{ad} + 41.25\texttt{reps} + 11.52\texttt{eff}$$

- Interpret the coefficient for `reps` (41.25).

Estimates from Click Ball Point Pens (Continued)

- Construct at 95% CI for `reps`.
  Solution: $41.25 \pm 2.028 \times 7.28 = [26.5, 56.0]$

```
> confint(fit)
                 2.5 %    97.5 %
(Intercept) -38.159815 100.46059
ad            7.416798  18.51953
reps         26.480882  56.01037
eff          -4.074175  27.12268
```

- Is `eff` different from zero (use .05 level)?

  $H_0 : \beta_3 = 0$ versus $H_1 : \beta_3 \neq 0$.

$$\begin{aligned} P(b_3 > 11.52) &= P\left(T_{36} > \frac{11.52 - 0}{7.6912}\right) \\ &= P(T_{36} > 1.498) = 0.0714 \end{aligned}$$

  The P value is thus `2*(1-pt(1.498, 36)) = 0.1429`. We cannot reject H_0 because $0.1428 > 0.05$ and we cannot conclude that wholesaler efficiency affects sales.

- Predict sales for `ad`=4, `reps`=3, `eff`=1. *Solution:*

$$\hat{y} = 31.15 + 12.97 \times 4 + 41.25 \times 3 + 11.52 \times 1 = 218.3$$

```
> predict(fit, data.frame(ad=4, reps=3, eff=1))
       1
218.2842
```

Your Turn

1. A commercial real estate company evaluates vacancy rates, square footage, rental rates, and operating expenses for commercial properties in a large metropolitan area in order to provide clients with quantitative information upon which to make rental decisions. The data in `commercial.txt` are taken from 81 suburban commercial properties that are the newest, best located, most attractive, and expensive for five specific geographic areas. It includes their age (`x1`), operating expenses and taxes (`x2`), vacancy rates (`x3`), total square footage (`x4`), and rental rates (`y`).

   (a) Read the commercial data set into R and run basic descriptive statistics (counts, mins, maxs, means). Do the descriptives make sense? Hint:

   ```
   > comm = read.table("c:/teach/304/data/commercial.txt", header=T)
   > summary(comm)
   ```

   (b) Produce a scatterplot matrix and correlation matrix. Discuss the relationships between the variables. Hint: use `plot(comm)` and `cor(comm)`.

   (c) Regress rental rates on the four predictor variables. State the estimated regression equation. Hint:

   ```
   > fit = lm(y ~ x1 + x2 + x3 + x4, comm)
   > summary(fit)
   ```

   (d) Obtain the residual plot study the distribution of residuals. Does the distribution to be fairly symmetrical? Hint: `plot(fit, which=1)`

   (e) Test whether the overall model is significant. State the null and alternative, P-value and decision. Hint: `summary(fit)`

   (f) What fraction of variation in rental rates is explained by these predictor variables?

   (g) If the overall regression model is significant in the previous part then test whether each of the individual regression coefficients equals 0 at the 5% level against a 2-sided alternative, i.e., $H_0 : \beta_j = 0$ for $j = 1, 2, 3, 4$. For each predictor, state the P-value and your decision.

   (h) Assume that the regression model you have estimated is appropriate. Three properties with the following characteristics did not have any rental information available.

	Property 1	Property 2	Property 3
`x1`	4	6	12
`x2`	10	11.5	12.5
`x3`	0.1	0	0.32
`x4`	80,000	120,000	340,000

   Predict the rental rate and compute separate prediction intervals for the rental rates using 95% confidence. **Briefly tell what the prediction interval tells you.** Hint:

```
> newx = data.frame(x1=c(4,6,12), x2=c(10,11.5,12.5), x3=c(.1,0,.32),
  x4=10000*c(8,12,34))
> predict(fit, newx, interval="prediction")
```

2. In a small-scale experimental study of the relation between degree of brand liking (`y`) and the moisture content (`moisture`) and sweetness (`sweetness`) of the product, the results in `brand.txt` were obtained from the experiment based on a completely randomized design.

   (a) Read the brand data set into R and run basic descriptive statistics (counts, mins, maxs, means). Do the descriptives make sense?

   (b) Produce a scatterplot matrix and correlation matrix. Discuss the relationships between the variables.

   (c) Regress liking on the two predictor variables. State the estimated regression equation.

   (d) Obtain the residual plot study the distribution of residuals. Does the distribution to be fairly symmetrical?

   (e) What fraction of variation in rental rates is explained by these predictor variables?

   (f) Test whether the overall model is significant. State the null and alternative, P-value and decision.

   (g) If the overall regression model is significant in the previous part then test whether each of the individual regression coefficients equals 0 at the 5% level against a 2-sided alternative, i.e., $H_0 : \beta_j = 0$ for $j = 1, 2$. For each predictor, state the P-value and your decision.

   (h) Assume that the regression model you have estimated is appropriate. Predict the liking when `moisture=5` and `sweetness=4`. Find a prediction interval and separately a confidence interval for the estimated mean value using 99% confidence.

Answers

1. (a) The summary statistics make sense. Age ranges from 0 to 20, which is reasonable for real estate properties. Operating expenses and taxes are positive. The vacancy rate is between 0 and 1. Square footage looks reasonable, as do rental rates. (b) The first thing to note is the correlations with y. The older the property, the lower the rent. The higher the expenses, the higher the rent. Vacancy rate has a positive correlation, but it is very weak. Square footage has a positive correlation with rent. There are also correlations among the predictor variables, especially between age and expenses (positive, size and expenses (positive), and expenses and vacancy rate (negative). (c) $\hat{y} = 12.2 - 0.142$ `x1` $+0.282$ `x2` $+0.619$ `x3` $+0.00000792$ `x4`. (d) The residual plot shows no patterns. (e) $H_0 : \beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$ versus H_1 : at least one $\beta_j \neq 0$. $P = 7.27 \times 10^{-14} < .05$ so reject H_0. At least one predictor is related to rental rate. (f) $R^2 = .5847$. (g) $H_0 : \beta_j = 0$ versus $H_1 : \beta_j \neq 0$. `x1`, `x2`, and `x4` all have $P < .05$. `x3` has $P = .57$ and we cannot reject the null or conclude that vacancy rate has an effect on the rental rate. (h) Property 1: 15.15, 12.85, 17.44; Property 2: 15.54, 13.25, 17.84; Property 3: 16.91, 14.53, 19.29.

2. (a) They make sense. All variables are positive. (b) Moisture has a stronger positive correlation with liking than sweetness. There is no correlation between sweetness and moisture because the data are from an experiment with an orthogonal design. (c) $\hat{y} = 37.650 + 4.425\texttt{moisture} + 4.375\texttt{sweetness}$. (d) There may be an inverted-U shaped relationship, but the fit is not bad. (e) $R^2 = .9521$. (f) $H_0 : \beta_1 = \beta_2 = 0$ versus H_1 : at least one $\beta_j \neq 0$. $P = 2.658 \times 10^{-9} < .05$ so reject H_0. At least one predictor is related to liking. (g) $H_0 : \beta_j = 0$ versus $H_1 : \beta_j \neq 0$. `moisture` and `sweetness` have $P < .05$. (h) Use `predict(fit, data.frame(moisture=5, sweetness=4), interval="conf", level=.99)` to find 77.275, 73.88, 80.67. Use `predict(fit, data.frame(moisture=5, sweetness=4), interval="pred", level=.99)` to find 77.275, 68.48, 86.07.

Comparing Regression Coefficients

- *Question:* Which of the variables is more "important" in explaining sales?
- *Answer:* The coefficients are not directly comparable because of differences in units of measurement.
- *Ideal solution:* convert to commensurate units, e.g., dollars.
- *Possible solution:* Use *standardized regression coefficients* (all variable standardized before the analysis to have mean 0 and variance 1). The "unit" of measurement is now the standard deviation

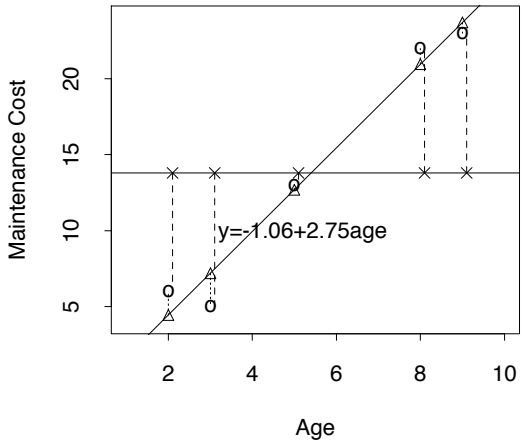
```
> Zclick = as.data.frame(scale(click[,1:4]))
> fit = lm(sales ~ ad + reps + eff -1, Zclick)  # -1 drops intercept
> summary(fit)
Coefficients:
     Estimate Std. Error t value Pr(>|t|)
ad    0.45101    0.09390   4.803 2.59e-05 ***
reps  0.54902    0.09559   5.744 1.40e-06 ***
eff   0.09157    0.06028   1.519    0.137
```

- *Possible solution:* compare t scores ($t = b_j/S_{b_j}$, reported in the fifth column of output)
- See Bring (1994) for discussion

Standardized Regression Coefficients

- Theorem: the standardized regression coefficient is $b_j S_{x_j}/S_y$, where S_{x_j} and S_y are the standard deviations of x_j and y, respectively, and b_j is the (unstandardized) regression estimate for variable j.
- Standardized regression coefficients are sometimes called "*beta*" coefficients
- Interpretation: a standard deviation increase in x_j is associated with "beta" standard deviations in y — the units of measurement are now the standard deviation.
- In simple linear regression, "beta" is the correlation r between x and y.
- Do not use "beta" coefficients blindly:
  - If x_j is a 0-1 variable, then the standard deviation is $\sqrt{\bar{x}_j(1-\bar{x}_j)}$
  - If you are analyzing a designed experiment, you select the x_j values and thus the standard deviation
  - "Beta" values are still a function of other variables in the model
  - "Beta" values do not consider costs

Machine example



- Regression model

$$y_i = \beta_0 + \beta x_i + e_i,$$

where e_i has mean 0 and variance σ^2. Note that the variance of e_i does not depend on i; this is called *homoscedastic error variance*. We also assume that e_i is independent of e_j for $i \neq j$. If you intend to do hypothesis tests or compute confidence intervals, either the distribution of e_i must be normal or the sample size must be large.

- Null model (overall mean of y) (horizontal line in figure)

$$\bar{y} = \frac{69}{5} = 13.8$$

The $\times$ symbols on the figure show the "fitted values" for the null model.

- Estimated regression equation (diagonal line in figure)

$$\hat{y}_i = -1.06 + 2.75 x_i$$

the triangle symbols on the plot show the fitted values $\hat{y}_i$.

	`age`	`cost`		RSS	MSS	TSS
i	x_i	y_i	$\hat{y}_i$	$(y_i - \hat{y}_i)^2$	$(\hat{y}_i - \bar{y})^2$	$(y_i - \bar{y})^2$
1	2	6	4.44	2.43	87.59	60.84
2	5	13	12.70	0.09	1.21	0.64
3	9	23	23.71	0.50	98.20	84.64
4	3	5	7.19	4.80	43.65	77.44
5	8	22	20.96	1.08	51.22	67.24
Sum	27	69	69	8.92	281.88	290.8

- Sum of squared errors:

$$\text{RSS} = \sum_{i=1}^{5}(y_i - \hat{y}_i)^2 = 8.9247$$

  The dotted lines show the errors $y_i - \hat{y}_i$.

- Regression/Model sum of squares

$$\text{MSS} = \sum_{i=1}^{5}(\hat{y}_i - \bar{y})^2 = 281.88$$

- Total sum of squares

$$\text{TSS} = \sum_{i=1}^{5}(y_i - \bar{y})^2 = 290.8$$

  The dashed lines show the errors $y_i - \bar{y}$ under the null model.

- Note that the ANOVA equality holds:

$$\text{RSS} + \text{MSS} = \text{TSS} \Longrightarrow 8.92 + 281.88 = 290.8$$

- The sums of squares are converted into variance estimates by dividing by the appropriate degrees of freedom. The degrees of freedom for the model (regression) equal p, the number of predictor variables in the model, where the intercept is not included. The error degrees of freedom equal $n - p - 1$.

	Df	Sum of Sq	Mean Sq	F Value	Pr(F)
Model	1	281.8753	281.8753	94.75	0.0023
Error	3	8.9247	2.9749		
Total	4	290.8000			

- The *root mean squared error* (or *standard error of the regression*) is $\sqrt{2.9749} = 1.72479$, which estimates σ. The mean squared error (2.9749) estimates σ^2.

- The *coefficient of determination*, or R^2, gives the fraction of variation explained by the model:

$$R^2 = \frac{\text{MSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}} = \frac{281.8753}{290.8} = 96.93\%$$

- The F statistic is computed as follows

$$F = \frac{281.8753}{2.9749} = 94.75$$

  The corresponding P-value (.0023) tests "the overall significance of the model," i.e., H_0 : all $\beta = 0$ versus H_1 : at least one $\beta \neq 0$. The larger F, the more evidence against H_0. If you cannot reject H_0, you cannot conclude that the regression model does an better than the null model.

IEMS 304: Homework 2
Due: April 13
Professor Malthouse

You may work in self-selected group of at most 5 students. Turn in one copy of your answers. All group members must put their name on the homework.

1. Use the auto data set from JWHT problem 3.9 on page 122.

   (a) Regress `mpg` on `cylinders`, `displacement`, `weight`, and `year`. Comment on the signs of the estimated coefficients and note which are significantly different from 0. What is value of R^2?

   (b) Compute the variance inflation factors. What do they tell you?

   (c) Drop `weight` from the model. What happens to the parameter estimates and R^2?

   (d) Drop `weight` and `displacement` from the model. What happens to the parameter estimates and R^2?

2. JWHT problem 3.14a–f on page 125. For part (c)–(e), are the parameters "covered" by the 95% confidence intervals?

3. Suppose that there are two predictor variables, x_1 and x_2, but we fit the straight line model $y = \beta_0 + \beta_1 x_1 + e$ omitting x_2. If, in fact, the true model is $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + e$, show that

$$E(b_1) = \beta_1 + \beta_2 \sum_{i=1}^{n} c_i x_{i2} = \beta_1 + \frac{\beta_2}{S_{11}} \sum_{i=1}^{n} (x_{i1} - \bar{x}_1)(x_{i2} - \bar{x}_2) = \beta_1 + \beta_2 r \frac{s_2}{s_1},$$

   where $c_i = (x_{i1} - \bar{x}_1)/S_{11}$, $S_{11} = \sum_{i=1}^{n} (x_{i1} - \bar{x}_1)^2$, r is the sample correlation coefficient between x_1 and x_2, and s_1, s_2 are the sample SD's of x_1, x_2, respectively. Thus b_1 is biased with the bias given by the second term in the above expression. Under what condition is this bias zero? Discuss how this result applies to JWHT problem 3.14. Hint: See JWHT equation (3.4).

4. In class I mentioned that one of the effects of multicollinearity is to increase the variance of the slope estimates. This problem will show you why this is the case. Suppose that $y = \beta_1 z_1 + \beta_2 z_2 + e$, where all variables have been standardized prior to estimation making the intercept unnecessary. You have observed (z_{i1}, z_{i2}, y_i), for $i = 1, \ldots, n$, where the sample means are 0 and the sample variances are 1, i.e.,

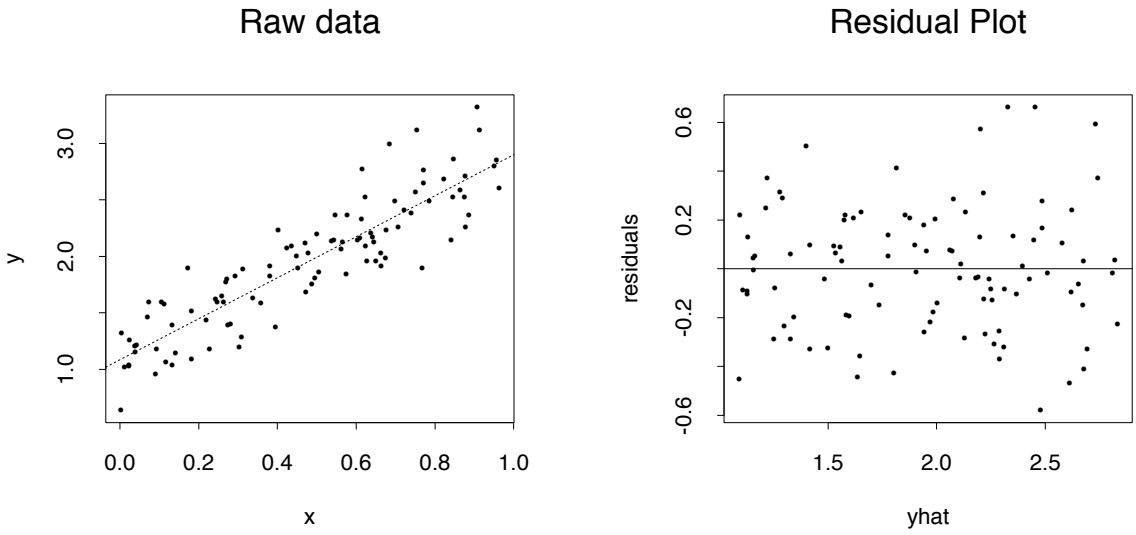
$$\bar{z}_j = \frac{1}{n} \sum_{i=1}^{n} z_{ij} = 0, \quad S_j^2 = \frac{1}{n-1} \sum_{i=1}^{n} (z_{ij} - 0)^2 = 1, \quad \text{and} \quad r = \frac{1}{n-1} \sum_{i=1}^{n} z_{i1} z_{i2},$$

   where r is sample correlation between z_1 and z_2. How does correlation r affect the variance of the slope estimates? Hint: recall that $V(\mathbf{b}) = \sigma^2 (\mathbf{Z}'\mathbf{Z})^{-1}$ and

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

Ideal Regression Model

True model: $y = 1 + 2x + e$



Least-squares fit: $\hat{y} = 1.09 + 1.81x$

- Definition of residuals

$$\hat{e} = y - \hat{y}$$

- Residual plots ($\hat{e}$ against $\hat{y}$) help us to understand how well the model fits the data. Use `plot(fit, which=1)` in R.
- Here, residuals do not follow any pattern
- Variance of residuals does not depend on $\hat{y}$ (homoscedasticity)
- This is an ideal residual plot

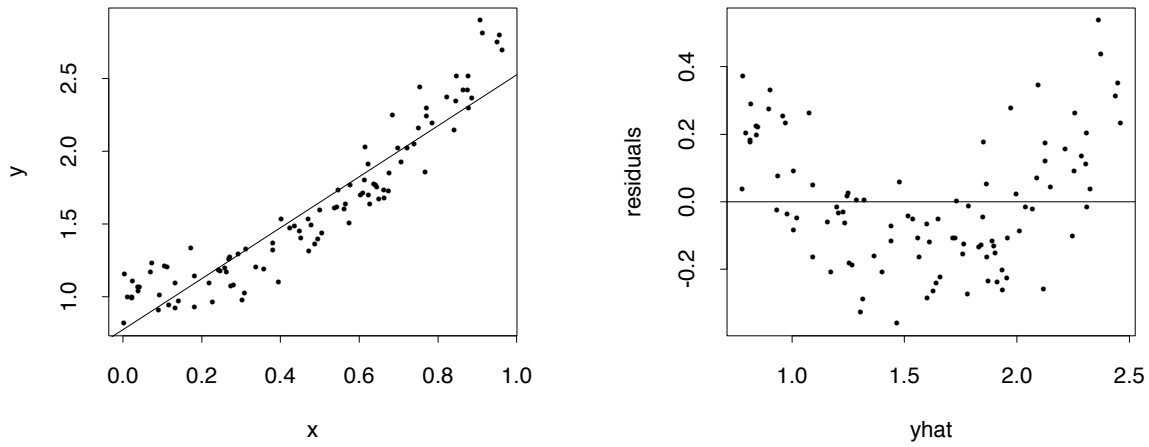
Model misspecification I

- Suppose the true model is

$$y = 1 + 2x^2 + e$$

- We estimate the *incorrect* (misspecified) model:

$$y = \beta_0 + \beta_1 x + e$$



- True relationship is nonlinear (curved)
- Fitted line does not describe the data well
- Pattern in residual plot indicates model misspecification
- But, Variance of residuals does *not* depend on $\hat{y}$ (error variance still homoscedastic)

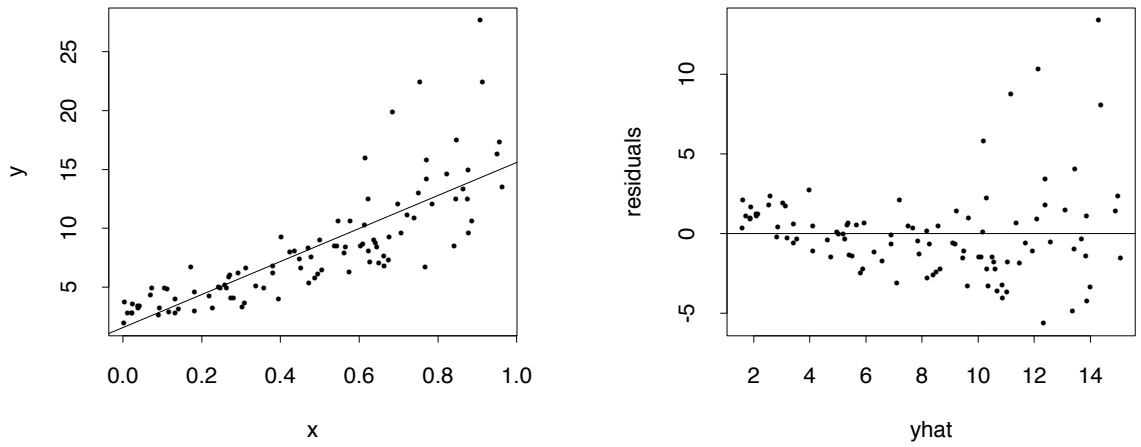
Model misspecification II

- Suppose the true model is

$$y = \exp(1 + 2x + e)$$

- We estimate the *incorrect* (misspecified) model:

$$y = \beta_0 + \beta_1 x + e$$

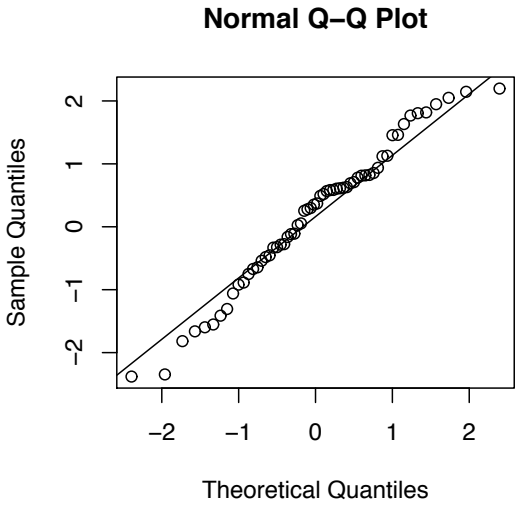


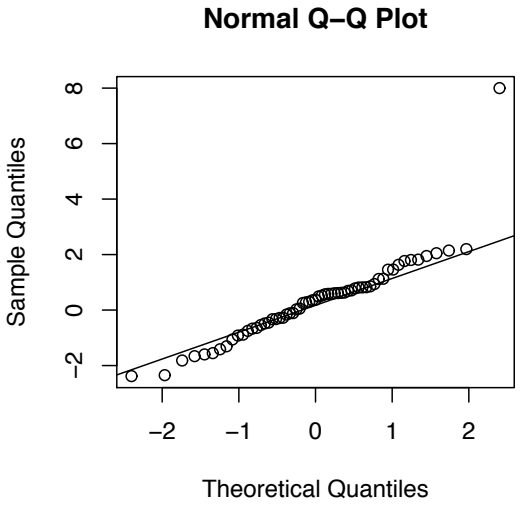
- Residuals show a pattern: first they are mostly positive, then mostly negative, then roughly centered at 0.
- Variance of the residuals increases with $\hat{y}$ indicating heteroscedasticity
- Note that

$$\log(y) = 1 + 2x + e$$

Evaluating Normality

- *When the CLT has not "converged"* then the population distribution of residuals must be normal for you to use the t distribution.
- Evaluate normality using a *normal probability plots*, which plot the observed quantile against normal quantiles ("Q-Q plot").
- Points falling on a line indicate normality.

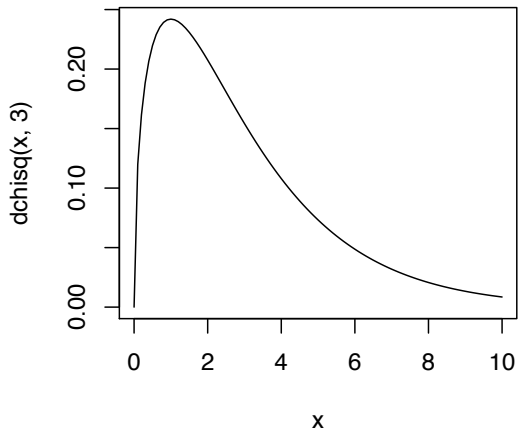




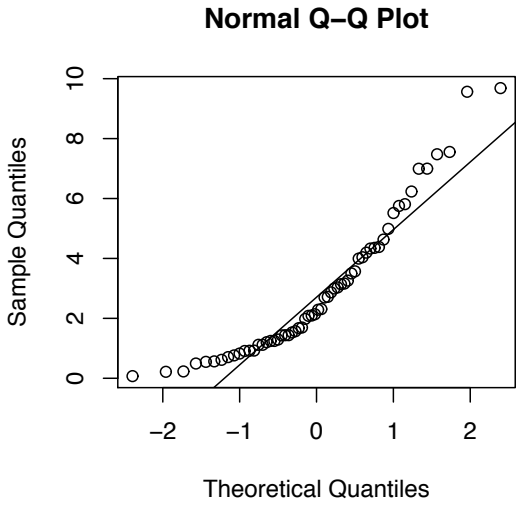

```
> set.seed(12345)
> z = rnorm(60)

> # data from normal distribution
> qqnorm(z); qqline(z)

> # now we add an outlier
> qqnorm(c(z, 8)); qqline(c(z,8))
```

Evaluating Normality

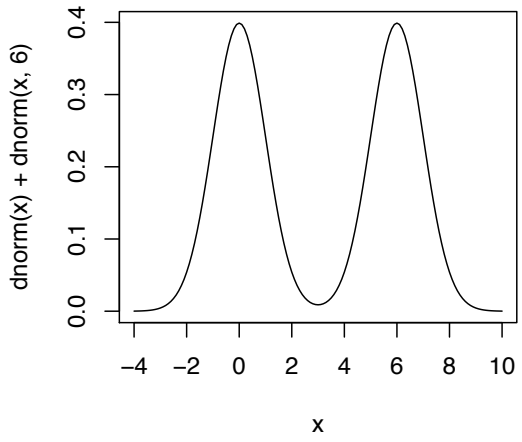
```
> x = seq(0, 10, .1)
> plot(x, dchisq(x, 3), type="l")
```



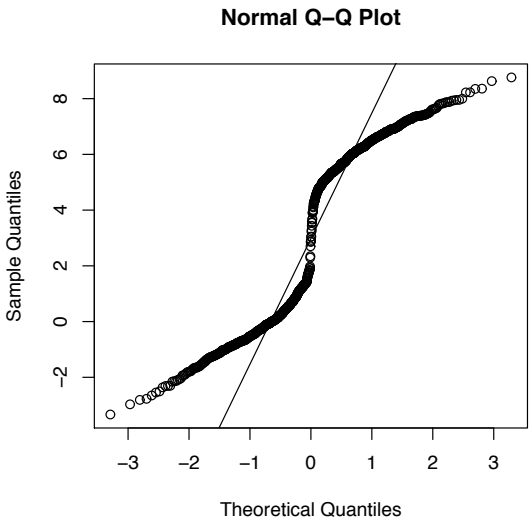

```
> rx = rchisq(60, 3)
> qqnorm(rx); qqline(rx)
```




```
> x = seq(-4, 10, .1)
> plot(x, dnorm(x)+dnorm(x,6), type="l")
```



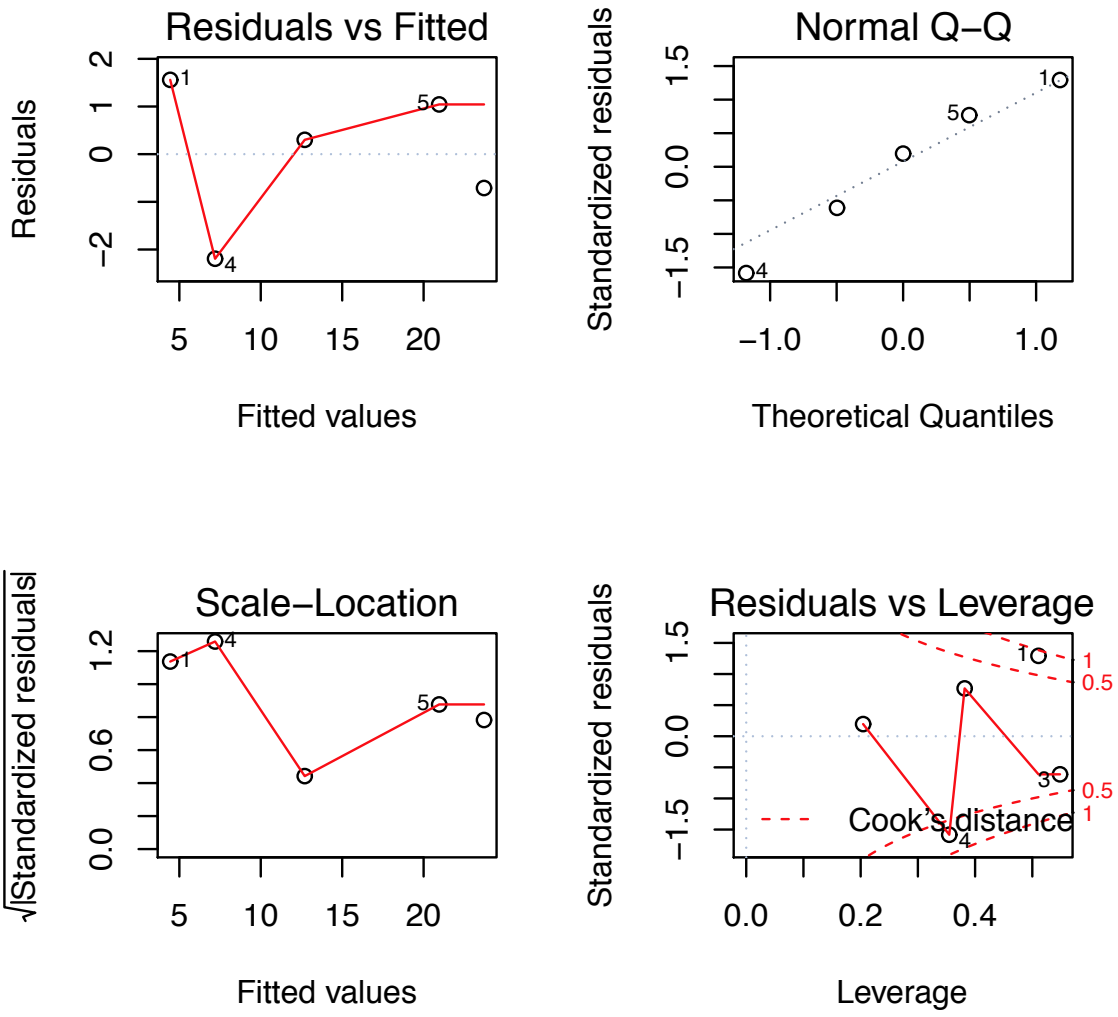

```
> x = c(rnorm(100),rnorm(100,6))
> qqnorm(x); qqline(x)
```



Residual and QQ Plots in R

- `plot(fit)` gives diagnostic plots of `lm` objects.
- Alternatively, `plot(fit, which=1)` gives residuals `plot(fit, which=2)` gives QQ plots, etc. See `?plot.lm`

```
> machine = data.frame(age=c(2,5,9,3,8), cost=c(6,13,23,5,22))
> fit = lm(cost ~ age, machine)
> par(mfrow=c(2,2)) # show 2*2 grid of plots. Use c(1,1) for one plot per page
> plot(fit)
```



Review of Key Results

- Multiple linear regression model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}, \quad E(\mathbf{e}) = \mathbf{0}, \quad \text{and} \qquad V(\mathbf{e}) = \sigma^2\mathbf{I} = V(\mathbf{y})$$

- We estimate $\boldsymbol{\beta}$ with OLS estimates $\mathbf{b}$ and $\mathbf{e}$ with $\hat{\mathbf{e}}$

$$\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \quad \text{and} \quad S_e^2 = \sum e_i^2/(n-p)$$

- Estimates of predicted values and residuals are given by

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{b} = \mathbf{H}\mathbf{y} \quad \text{and} \quad \hat{\mathbf{e}} = \mathbf{y} - \mathbf{X}\mathbf{b}$$

  where $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ is the *hat matrix*

- Theorem: $V(\hat{\mathbf{y}}) = \sigma^2\mathbf{H}$ and $V(\hat{\mathbf{e}}) = \sigma^2(\mathbf{I} - \mathbf{H})$

- The *leverage* of observation i is h_{ii}, where

$$0 \le h_{ii} \le 1 \quad \text{and} \quad \sum_{i=1}^{n} h_{ii} = p$$

  As a rule of thumb, leverages greater than twice their average, i.e., $h_{ii} > 2p/n$, are considered large

- For simple linear regression

$$h_{ii} = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{j=1}^{n}(x_j - \bar{x})^2}$$

Standardized Residuals

- Problem 1: estimated residuals $\hat{\mathbf{e}}$ do not have constant variance (even when the model, which assumes homoscedastic errors, is true).
- The *standardized residual*, sometimes called the *studentized residual* is obtained by dividing $\hat{e}$ by its estimated standard deviation

$$r_i = \frac{\hat{e}_i}{S_e\sqrt{1-h_{ii}}}$$

  Note: these are sometimes called *internal standardized residuals*.
- Problem 2: if there are outliners, S is inflated, which deflates all r_i. One solution is to omit observation i and reestimate the model giving prediction $\hat{y}_{(i)}$ and MSE $S^2_{(i)}$.
- The *deleted* (or *external*) residual and *studentized deleted/external residual* are

$$d_i = y_i - \hat{y}_{(i)} \quad \text{and} \quad \frac{d_i}{S_{(i)}\sqrt{1-h_{ii}}}$$

- Cook's distance

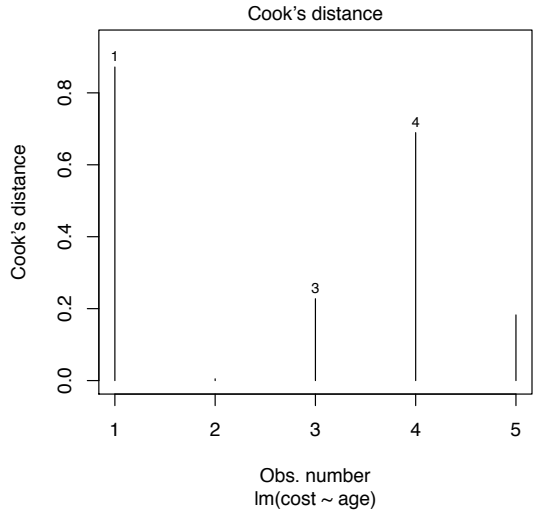
$$D_i = \frac{\sum_j (\hat{y}_j - \hat{y}_{j(i)})^2}{pS_e^2}$$

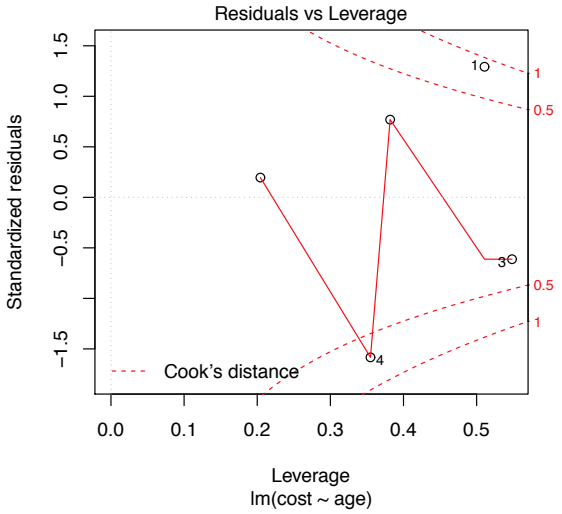
  As a rule of thumb, **values greater than 1 are considered large.**

Leverage in R: Machine Example

```
> fit = lm(cost~age, machine)
> plot(fit, which=c(4,5))

> # use lm.influence function to get leverages in R
> lm.influence(fit)$hat
        1         2         3         4         5
0.5107527 0.2043011 0.5483871 0.3548387 0.3817204
> sum(lm.influence(fit)$hat)
[1] 2

> # check our work with matrix inversion
> X=cbind(1,machine$age)
> diag(X %*% solve(t(X) %*% X) %*% t(X))
[1] 0.5107527 0.2043011 0.5483871 0.3548387 0.3817204

> # check our work with simple formula
> xbar = mean(machine$age)
> 1/5+(machine$age-xbar)^2/sum((machine$age-xbar)^2)
[1] 0.5107527 0.2043011 0.5483871 0.3548387 0.3817204
```

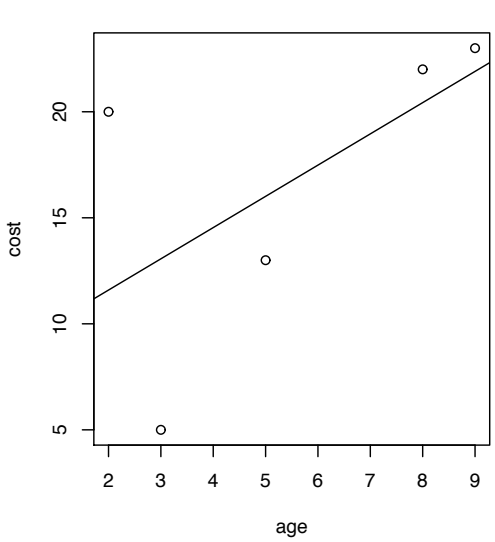
None are considered large since all are less than $2(2/5) = 0.8$

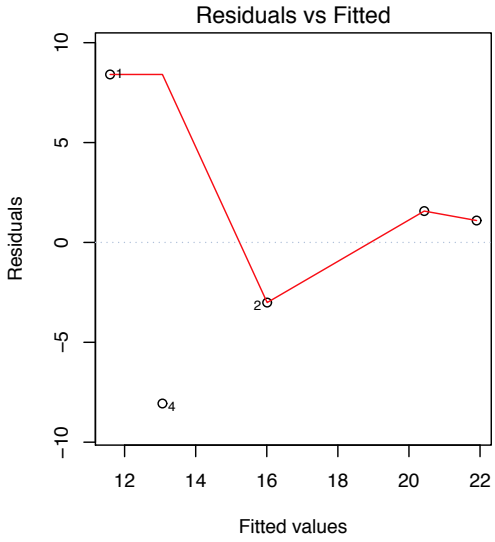


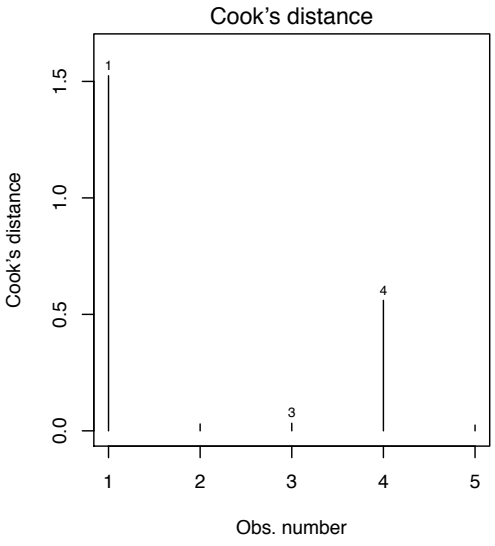


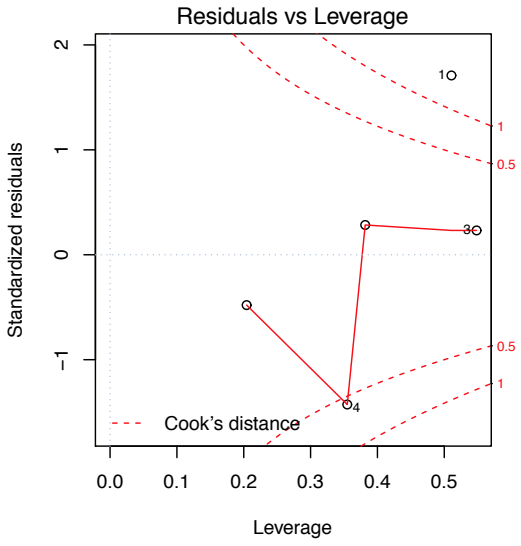
Modified Machine Example

```
> machine2 = data.frame(age=c(2,5,9,3,8), cost=c(20,13,23,5,22)) # change y1=20 from 6
> fit = lm(cost~age, machine2)
> plot(machine2); abline(fit)
> plot(fit)
```









Transformations

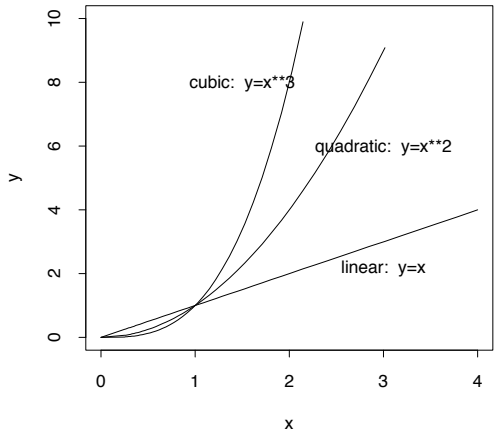
- Transformations change the functional relationship between dependent and predictor variables
- Two reasons for transformations:
  - Heteroscedasticity / non-symmetric error distributions ($E(e_i) = 0$ but Skewness$(e_i) \neq 0$): transform the *dependent* variable
  - Underlying relationship nonlinear: Transform the *predictor* variable(s)
- Outline of transformation lecture
  1. Transformations of dependent variable
  2. Identifying nonlinear relationships
     (a) Scatterplots
     (b) Compare R^2 values for models using various transformations (e.g., Tukey's ladder of re-expressions)
  3. Other transformations based on combinations of variables

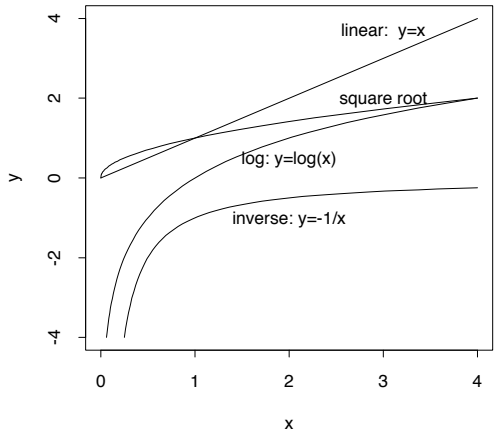
Heteroscedasticity

- Slide 63 gave assumptions of regression, including homoscedasticity
- If a model has heteroscedastic error variance, the least-squares estimates will still be unbiased, but will not be BLUE.
- Modeling heteroscedastic data:
  - Use variance-stabilizing transformations
  - Use *weighted least squares*
  - Use a different model, e.g., Poisson or logistic regression
- For count and amount dependent variables, use the logarithm or square root as a variance-stabilizing transformation. As shown in class, take logs when the **standard deviation** of the errors is proportional to the mean, and square roots when the **variance** is proportional to the mean.

Tukey's Ladder of Transformations

Consider models of the form $y = \beta x^k$ over $[0, \infty)$





Note: "Returns" refer to the first derivative (slope)

k	Function	Slope	Nature of "Returns"
3	$y = \beta x^3$	$dy/dx = 3\beta x^2$	Increasing
2	$y = \beta x^2$	$dy/dx = 2\beta x^1$	Increasing
1	$y = \beta x^1$	$dy/dx = \beta x^0$	Constant
$1/2$	$y = \beta\sqrt{x}$	$dy/dx = \beta x^{-1/2}/2$	Decreasing
0	$y = \beta \log x$	$dy/dx = \beta x^{-1}$	Decreasing
-1	$y = \beta x^{-1}$	$dy/dx = -\beta x^{-2}$	Decreasing

- For $k < 1$ the slope approaches 0, but never changes sign
- You may need to shift variables, e.g, $\log(x + 1)$ or $1/(x + 1)$

Tukey's First Aid Re-Expressions[1]

"Choosing exactly the right re-expression for a particular quantity may not be easy. To try to do a good job, we may have to (1) sense rather weak indications from the data in hand, (2) draw on experience with other bodies of data, or (3) lean on subject-matter knowledge. Even all three may not suffice. Both because we may not be prepared to try hard to choose our re-expression, or because we have too little information for anyone to choose reliably, we need rules of thumb that can provide "first aid," that can lead us to re-expressions that are almost always not bad — and usually pretty good.

Four rules will deal quite effectively with most of our needs, namely:

1. Take logs of an amount or count (if there are zeros or infinities, we may need to deal with them; see the next section)
2. Take logits or folded logs of fractions or percents; use some multiple of

$$\log\left(\frac{p}{1-p}\right)$$

...

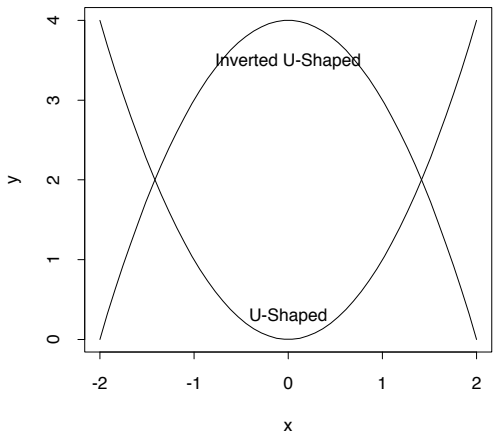
These rules are not supposed to be a final answer—just as first aid for the injured is no substitute for a physician—but they offer a safe beginning."

Also see discussion of "Tukey's ladder of re-expressions" in Tukey (1977), *EDA*, pp. 90–1.

[1] Mosteller and Tukey (1977), *Data Analysis and Regression*, p. 109

Polynomial Transformations

Consider models the form $y = \beta_0 + \beta_1 x + \beta_2 x^2$



- The min/max value equals $-\beta_1/2\beta_2$
- U-shaped when $\beta_2 > 0$
- Higher-order polynomials can also be used, e.g., cubic $\beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$, but they are problematic
  - Curvature can change several times — few theories postulate this.
  - x, x^2, x^3 highly correlated — mean center or standardize x's before fitting model.
  - Interpretation complicated — don't interpret estimates of individual terms. Use F-tests to gauge significance and plots to interpret effects.

Purification case

The "techies" (scientists) in the laboratory have been lobbying you, and management in general, to include just one more laboratory step. They think it's a good idea, although you have some doubt because one of them is known to be good friends with the founder of the start-up biotechnology company that makes the reagent used in the reaction. But if adding this step works as expected, it could help immensely in reducing production costs. The trouble is, the test results just came back and they don't look so good. Discussion at the upcoming meeting between the technical staff and management will be spirited, so you've decided to take a look at the data.

Your firm is anticipating government approval from the Food and Drug Administration (FDA) to market a new medical diagnostic test made possible by monoclonal antibody technology, and you are part of the team in charge of production. Naturally, the team has been investigating ways to increase production yields or lower costs.

The proposed improvement is to insert yet another reaction as an intermediate purifying procedure. This is good because it focuses resources down the line on the particular product you want to produce. But it shares the problem of any additional step in the laboratory: one more manipulation, one more intervention, one more way for something to go wrong. In this particular case, it has been suggested that, while small amounts of the reagent may be helpful, trying to purify too will will actually decrease the yield and increase costs.

The design of the test was to have a series of test production runs, each with a different amount of purifier, including one test run with the purification step omitted entirely (i.e., 0 purifier). The order of the tests was randomized so that any time trends would not be mistakenly interpreted as being due to purification.

```
purify = data.frame(x = 0:10,
  y=c(13.39,11.86,27.93,35.83,28.52,41.21,37.07,51.07,51.69,31.37,21.26))
```

1. Regress `yield` on `amount`. Is the regression significant? Based on this test alone, do you recommend including a purifying step in the process?
2. Generate a scatterplot of `yield` against `amount`. Comment.
3. Modify your regression model as appropriate. Based on the revised analysis, do you recommend including a purifying step in the process?

Video solution

Multiplicative Models

- Multiplicative models have the form

$$y = \beta_0 x_1^{\beta_1} x_2^{\beta_2} \epsilon$$

- Taking natural logs of both sides we get

$$\log y = \log \beta_0 + \beta_1 \log x_1 + \beta_2 \log x_2 + \log \epsilon,$$

  which has the form of a multiple linear regression model, regressing $\log y$ on $\log x_1$ and $\log x_2$. We estimate this model under the assumption that $\log \epsilon$ is normal with $V(\log \epsilon) = \sigma^2$ constant.

- Predictions: we estimate (untransformed) y with

$$\exp(b_0 + b_1 \log x_1 + b_2 \log x_2 + S_\epsilon^2/2)$$

- The coefficient of variation of e (rather than the variance) is

$$\frac{\sqrt{V(y)}}{E(y)} = \sqrt{\exp(\sigma^2) - 1},$$

  which doesn't depend on i.

- Even without the assumption that ϵ is log normal, we will see that if $\sigma_y \propto \mu_y$ then logging y "stabilizes" the error variance, making it constant. Logging y is called a *variance stabilizing transformation*.

Interpreting β_1

$$y = \beta_0 x^{\beta_1} \qquad \frac{dy}{dx} = \beta_0 \beta_1 x^{\beta_1 - 1} \qquad \frac{d^2y}{dx^2} = \beta_0 \beta_1 (\beta_1 - 1) x^{\beta_1 - 2}$$

- Interpretation (ignoring error term, and assuming $\beta_0 > 0$ and $x > 0$)
  - $\beta_1 = 1 \Longrightarrow y \propto x$ (linear, proportional returns)
  - $0 < \beta_1 < 1 \Longrightarrow$ changes in y *decrease* as x increases (concave downward)
  - $\beta_1 > 1 \Longrightarrow$ changes in y *increase* as x increases (concave upward)
- In economic applications, β_j is the *elasticity* of y with respect to x_j — the expected percentage change in y of a 1% change in x_j, all else being equal. Let dx be an infinitesimal change in x and dy be the corresponding change in y. Then the elasticity is

$$\frac{dy/y}{dx/x} = \frac{dy}{dx} \cdot \frac{x}{y} = \beta_0 \beta_1 x^{\beta_1 - 1} \cdot \frac{x}{\beta_0 x^{\beta_1}} = \beta_1$$

- We estimate β_1 by logging both sides. The base of the logarithm does not matter, but natural logs are usually used.

Background: Lognormal Distribution

- If y has a normal distribution with mean μ and variance σ^2, then $y^* = \exp(y)$ has a *log-normal distribution*, i.e., $\log y^* = y$ is normal.
- Theorem: the mean and variance of y^* are

$$E(y^*) = \exp(\mu + \sigma^2/2)$$

$$V(y^*) = \exp(2\mu + \sigma^2)(\exp(\sigma^2) - 1)$$

- The coefficient of variation of y^* is

$$\begin{aligned} \frac{\sqrt{V(y^*)}}{E(y^*)} &= \frac{\sqrt{\exp(2\mu + \sigma^2)(\exp(\sigma^2) - 1)}}{\exp(\mu + \sigma^2/2)} \\ &= \sqrt{\exp(\sigma^2) - 1}, \end{aligned}$$

  which does not depend on μ

Business Failure Case

Consider the slightly scary topic of business failures. This problem analyzes data from each stat on the number of failed businesses and the population in thousands for each of the 50 states and the District of Columbia (51 observations).

```
busfail = data.frame(
  row.names=c("AL","AK","AZ","AR","CA","CO","CT","DE","DC","FL","GA","HI","ID",
    "IL","IN","IA","KS","KY","LA","ME","MD","MA","MI","MN","MS","MO","MT","NE",
    "NV","NH","NJ","NM","NY","NC","ND","OH","OK","OR","PA","RI","SC","SD","TN",
    "TX","UT","VT","VA","WA","WV","WI","WY"),
  pop=c(4187,599,3936,2424,31211,3566,3277,700,578,13679,6917,1172,1099,11697,
    5713,2814,2531,3789,4295,1239,4965,6012,9478,4517,2643,5234,839,1607,1389,
    1125,7879,1616,18197,6945,635,11091,3231,3032,12048,1000,3643,715,5099,
    18031,1860,576,6491,5255,1820,5038,470),
  fail=c(841,108,2064,186,19695,1542,1093,137,200,5088,2350,305,350,2094,1091,507,
    1069,841,664,383,1540,2720,2546,921,322,1230,173,399,568,617,2843,448,6916,1194,
    145,2127,1440,969,3124,344,392,175,1209,7096,351,173,1738,2025,315,1224,90)
)
```

1. Make a scatterplot of business failures against population and superimpose a regression line. Describe the relationship. Comment on whether the linear model appears to hold.
2. Make a scatterplot of the log of business failures against the log of population. Superimpose a regression line. Does the linear model hold better with the logged data?
3. Regress the log of failures on the log of population. State estimated regression equation.
4. Test at the 5% level to see whether there is a significant relationship between the logs of failure and population. Explain.
5. Test whether the population slope for the logs is significantly different from 1 or not. What does this tell you?
6. Illinois had 2,094 failures with a population of 11,697 (thousand). Find the predicted log failures for Illinois.
7. Estimate the unlogged failures and identify Illinois on your scatterplot. (Hint: use `identify` function in R.)

Your Turn

1. The table below shows the level of investment and the results obtained by the important players in fiber-optics cable for long-distance communications.

```
fiber = data.frame(invest=c(1300,500,130,2000,1200,110,40,60,57,500,90,90),
  miles=c(1700,650,110,1200,2400,165,72,45,85,650,50,87))
```

   (a) Find the regression equation predicting circuit miles from investment.

   (b) Draw the scatterplot and residuals. Discuss whether the linear model holds.

   (c) Examine Cook's distance. Are there influential observations, as indicated by having Cook's distance greater than 1?

   (d) Regress the log of circuit miles on the log of investment. State the estimated equation.

   (e) Draw the scatterplot and residuals for the log model. Discuss whether the linear model holds.

   (f) Do firms that spend more achieve significantly more circuit miles? State the null and alternative, P-value and decision using the 5% level of significance.

   (g) Test at the 5% level whether the coefficient for log(investment) is different from 1, which indicates that investment is proportional to miles. What do values not equal to 1 indicate in terms of economies of scale?

   (h) Predict log(miles) from an investment of $1000.

   (i) Predict unlogged miles from an investment of $1000.

2. A research analyst for an oil company wants to develop a model to predict miles per gallon based on highway speed. An experiment is designed in which a test car is driven at speeds ranging from 10 miles per hour to 75 miles per hour in increments of 5 MPH. Two replicates were observed for each speed:

```
speed = data.frame(mph=rep(seq(10,75,by=5), 2),
  mpg=c(4.8,8.6,9.8,13.7,18.2,19.9,22.4,21.3,20.5,18.6,14.4,12.1,10.1,8.4,
    5.7,7.3,11.2,12.4,16.8,19,23.5,22,19.7,19.3,13.7,13,9.4,7.6))
```

   (a) Make a scatterplot of MPG against MPH. Based on the plot, do you suggest transforming the data? If, so which transformation do you suggest?

   (b) Regress MPG on MPH (do not include transformations yet). Report (i) the estimated regression equation, (ii) the P-value testing the overall significance of the model, and (iii) a residual plot of residuals versus predicted values.

   (c) Perform a lack-of-fit test on the model from the previous part. Report the P-value and your decision.

   (d) Add appropriate transformations to your model. Report (i) the estimated regression equation, (ii) the P-value testing the overall significance of the model, and (iii) a residual plot with comments about the fit of your model.

(e) Perform a lack-of-fit test on the model from the previous part.

(f) Using the model you developed in the previous part, estimate gas milage for a car traveling at 62 miles per hour.

(g) At what speed is gas milage maximized?

(h) Produce a scatterplot with the three fits superimposed (linear, quadratic and "full" model for the LOF test where a separate mean estimate is made for each unique value of MPH).

Answers

1. (a) miles(hat) = 101.813 + 0.986 invest; (b) The linear model does not hold because the variance of the residuals increases with the mean of miles. (c) Yes, observation 4 is influential, and 5 is nearly "influential." (d) log(miles) = 0.06803 + 1.00735 log(miles). (e) The residuals have more constant variance and no obvious pattern. (f) $H_0 : \beta \leq 0$ versus $H_1 : \beta > 0$, $P = 1.02 \times 10^{-6}/2 < .05$, so we reject H_0. (g) $H_0 : \beta = 1$ versus $H_1 : \beta \neq 1$. A 95% confidence interval for the slope is $1 \in [0.79, 1.22]$. Since 1 is in this interval, we cannot reject the null hypothesis that the slope is 1. It is plausible that miles are proportional to investment. Recall that investment $= e^a \times$ invest^b. (h) predict(fit, data.frame(invest=1000)) = 7.026552 (i) exp(7.026552 + 0.1938/2), where 0.1938 is the MSE from the ANOVA table.

2. (a) The scatterplot shows an inverted-U shaped relationship, but no heteroscedasticity. We should add a quadratic term for speed. (b) mpg = 12.75 + 0.039 speed. $P = 0.468 > 5\%$ so we cannot reject $H_0 : \beta = 0$. The residuals show a strong pattern indicating that the model is misspecified. (c) $P = 2.436 \times 10^{-12} < 5\%$, so we reject the null hypothesis and conclude that the model does not fit adequately. (d) Add a quadratic term: $\texttt{mpg} = -7.56 + 1.27\texttt{speed} - 0.0145\texttt{speed}^2$. $P = 2.338 \times 10^{-14} < 5\%$, so we reject $H_0 : \beta_1 = \beta_2 = 0$. The residuals are not perfect, but the magnitude of the pattern is reduced. (e) $P = 4.044 \times 10^{-5} < 5\%$, so we reject H_0 and conclude that there is lack of fit. It looks like MPG increases linearly until about 40MPH. The quadratic model provides a substantially better fit ($R^2 = .9188$) than the linear model ($R^2 = .02039$), but the the quadratic model could be improved further. (f) `predict(fit, data.frame(speed=62))` = 15.54618 (g) $-1.27/(2 \times -0.0145) = 43.8$ MPH (h)

```
> fit = lm(mpg~mph, speed)
> fit2 = lm(mpg~mph+I(mph^2), speed)
> fit3 = lm(mpg~as.factor(mph), speed)
> plot(speed)
> abline(fit)
> lines(10:75, predict(fit2, data.frame(mph=10:75)), col=2)
> lines(speed$mph[1:14], fit3$fit[1:14], type="b", col=4, pch=16)
```

Dummy Variables

- Question: How do we include nominal variables in a regression?
- Answer: Use dummy (also called indicator) variables
- Example: Quality Case. Let AM be a *dummy* variable that takes the value 1 if the observation comes from the morning shift and 0 otherwise (afternoon shift). As a first step we could regress defects on AM.

```
> fit = lm(defect~ am, quality)
> summary(fit)
...
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   16.920      4.308   3.927  0.00051 ***
am            20.440      6.093   3.355  0.00229 **
```

  Interpretation of 20.44: every unit increase in AM (i.e., going from PM to AM) is associated with a 20.44 change in defect rate. The AM shift has 20.44 more defects per thousand than the PM shift.

- This is equivalent to an independent-sample t-test with equal variances assumed:

```
> t.test(defect~am, quality, var.equal=T)
data:  defect by am
t = -3.3547, df = 28, p-value = 0.002295
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -32.920676  -7.959324
sample estimates:
mean in group 0 mean in group 1
          16.92           37.36
```

  The mean difference is $37.36 - 16.92 = 20.44$

- We reject $H_0 : \beta = 0$, or equivalently, $H_0 : \mu_{\text{AM}} = \mu_{\text{PM}}$ because $0.0023 < 0.05$.

Dummy Variables: Wholesaler Efficiency

If the Wholesaler is	Dummy Variable Coding Fair	Good	Outstanding
Fair	1	0	0
Good	0	1	0
Outstanding	0	0	1
Poor	?	?	?

```
> fit = lm(sales~ad+reps+as.factor(eff), data=click)
> drop1(fit, test="F")
               Df Sum of Sq    RSS    AIC F value    Pr(>F)
<none>                       71018 311.27
ad              1     41227 112245 327.58 19.7376 8.955e-05 ***
reps            1     54607 125625 332.09 26.1433 1.226e-05 ***
as.factor(eff)  3      4457  75475 307.71  0.7112     0.552

> summary(fit)
                Estimate Std. Error t value Pr(>|t|)
(Intercept)       45.051     36.631   1.230    0.227
ad                13.063      2.940   4.443 8.96e-05 ***
reps              40.948      8.009   5.113 1.23e-05 ***
as.factor(eff)2    9.239     27.916   0.331    0.743
as.factor(eff)3   20.283     29.344   0.691    0.494
as.factor(eff)4   33.260     28.440   1.169    0.250
```

$$\hat{y} = 45 + 13\texttt{ad} + 41\texttt{reps} + 9\texttt{fair} + 20\texttt{good} + 33\texttt{out}$$

- First test H_0 : all $\beta_j = 0$ using `drop1` in R.
- `as.factor(eff)` indicates that `eff` should be treated as categorical (i.e., create dummies).
- What null hypothesis does the P-value for OUT test ($P = .2503$)? What does this test mean in English?

SPSS Estimates

Note: In practice we should first look at the ANOVA table (next page).

Parameter Estimates

Parameter	B	Std Error	t	Sig
Intercept	78.311	26.993	2.901	0.006
AD	13.063	2.940	4.443	0.000
REPS	40.948	8.009	5.113	0.000
[EFF=1]	−33.260	28.440	−1.169	0.250
[EFF=2]	−24.020	19.339	−1.242	0.223
[EFF=3]	−12.976	18.643	−0.696	0.491
[EFF=4]	0^a	.	.	.

a. This parameter is set to zero because it is redundant.

- Estimated regression equation: $\hat{y} = 78.3 + 13$`ad` $+ 41$`reps` -33`Poor` -24`Fair` -13`Good` $+ 0$`Out`
- Why are estimates different than on page 100? Note that `ad` and `reps` are identical.
- What null hypothesis does the P-value for `Poor` test ($P = .250$)? What does this mean in English?
- Note that SPSS/Minitab/SAS do not give standardized regression coefficients for dummies. Why?

General Linear Test for Multiple Betas

```
> drop1(fit, test="F")
sales ~ ad + reps + as.factor(eff)
               Df Sum of Sq    RSS    AIC F value     Pr(F)
<none>                       71018    311
ad              1     41227 112245    328 19.7376 8.955e-05 ***
reps            1     54607 125625    332 26.1433 1.226e-05 ***
as.factor(eff)  3      4457  75475    308  0.7112     0.552

> deviance(fit)
[1] 71017.78

> anova(fit)
               Df Sum Sq Mean Sq  F value    Pr(>F)
ad              1 463451  463451 221.8787 < 2.2e-16 ***
reps            1  59327   59327  28.4032 6.414e-06 ***
as.factor(eff)  3   4457    1486   0.7112     0.552
Residuals      34  71018    2089
```

- The "ad" line tests $H_0 : \beta_1 = 0$ and is equivalent to the t test on the previous page. Likewise for the "reps" line.
- The "eff" line tests $H_0 : \beta_3 = \beta_4 = \beta_5 = 0$ (i.e., the three dummies for `eff` are all 0 meaning all levels of wholesaler efficiency are the same) versus H_1 : at least one of β_3, β_4, or β_5 is different from 0.
- The "< none >" line gives SSE

How to Handle Missing Values

```
agemiss = data.frame(
  age = c(NA,NA,35,NA,81,39,20,25,62,NA,45,57,36,39,NA,48,36,NA,NA,30,
     78,35,NA,20,26,28,44,30,31,32,72,33,33,NA,55,37,36,43,40,NA),
  y = c(2.9,2.8,8.4,2.8,4.5,8.3,9.4,9.1,5.6,2.9,7.4,6.3,7.7,8.1,3.2,6.5,
     7.9,3.0,3.0,9.0,5.1,8.8,3.4,9.5,8.9,8.3,7.4,8.3,8.6,8.7,5.3,8.3,
     7.8,3.2,6.6,8.4,8.6,7.8,7.6,3.7))
```

Solution: treat missing as a separate category and include a dummy:

1. Create dummy `xmiss` that equals 1 when `x` is missing and 0 otherwise.
2. When `x` is missing, set `x=0`
3. Regress `y` on both `x` and `xmiss`

```
> agemiss$xmiss=is.na(agemiss$age)     # 1. create dummy
> agemiss$age[is.na(agemiss$age)] = 0  # 2. set missings to 0
> plot(agemiss$age, agemiss$y)
> fit = lm(y ~ age + xmiss, agemiss)   # 3. regression
> summary(fit)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.040189   0.163265   67.62   <2e-16 ***
age         -0.080755   0.003737  -21.61   <2e-16 ***
xmissTRUE   -7.950189   0.191438  -41.53   <2e-16 ***
Residual standard error: 0.3161 on 37 degrees of freedom
```

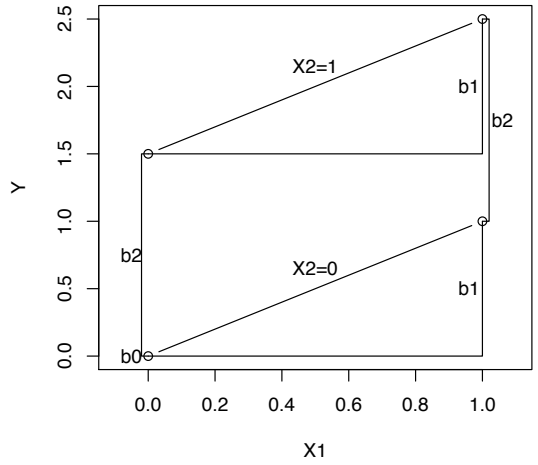
- When age is present, $y = 11.04 - 0.08\text{age} - 7.95(0)$.
- When age is missing, $y = 11.04 - 0.08(0) - 7.95(1)$

What is an Interaction?

Interaction terms are *nonlinear* combinations of *two or more* predictor variables. Suppose we have two **categorial** predictors (x_1 and x_2), which take only two value 0 and 1.

Linear (additive) model

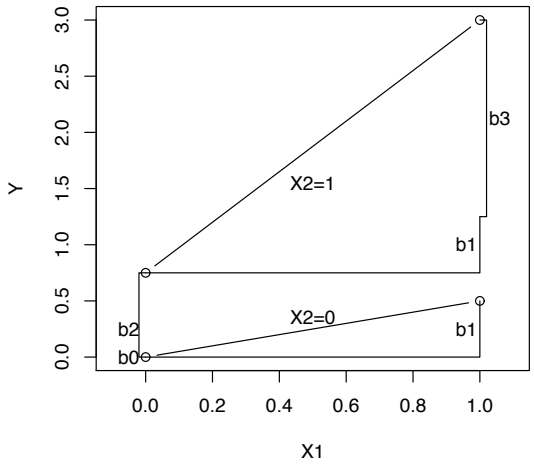
$$\hat{y} = b_0 + x_1 b_1 + x_2 b_2$$



x_1	x_2	$\hat{y}$
0	0	b_0
0	1	$b_0 + b_2$
1	0	$b_0 + b_1$
1	1	$b_0 + b_1 + b_2$

Linear model with product interaction term

$$\hat{y} = b_0 + x_1 b_1 + x_2 b_2 + x_1 x_2 b_3$$



x_1	x_2	$\hat{y}$
0	0	b_0
0	1	$b_0 + b_2$
1	0	$b_0 + b_1$
1	1	$b_0 + b_1 + b_2 + b_3$

Interactions in R

- `?formula` for help
- Additive effects: `x1 + x2`
- `x1:x2` = interaction between `x1` and `x2`, or use `*`

```
x1 + x2 + x1:x2 = x1*x2
```

- To specify main effects and all two-way interactions use

```
(a+b+c)^2 = a + b + c + a:b + a:c + b:c
```

- Use the minus sign to drop terms, e.g.,

```
(a+b+c)^2 - a:b = a + b + c + a:c + b:c
```

- To specify quadratic effects use `I( )`
- `as.factor(a)` casts `a` as a factor. To do this permanently you can type `dat$a = as.factor(dat$a)`
- Use `drop1` to determine if terms can be dropped.

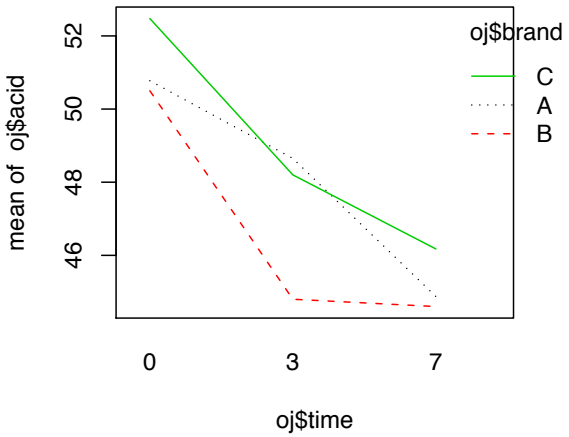
Example

A company that prepares students for the ACT college extrance exam is testing new curriculum. They want to investigage the length of the course (condensed 10-day course versus regular 30-day course) and the modality (traditional classroom versus online distance). Students were assigned at random to the four treatment combinations.

```
> course = data.frame(
  type=factor(c(rep(1,20), rep(2,20)), 1:2, c("Trad","Online")),
  length=factor(c(rep(1,10), rep(2,10), rep(1,10), rep(2,10)),
    1:2, c("Condensed","Regular")),
  act=c(26,27,25,21,21,18,24,19,20,18,  34,24,35,31,28,28,21,23,29,26,
        27,29,30,24,30,21,32,20,28,29,  24,16,22,20,23,21,19,19,24,25))

> fit = aov(act ~ type*length, course)
> summary(fit)
            Df Sum Sq Mean Sq F value   Pr(>F)
type         1    5.6     5.6   0.399    0.532
length       1    0.2     0.2   0.016    0.900
type:length  1  342.2   342.2  24.257 1.89e-05 ***
Residuals   36  507.9    14.1
> interaction.plot(course$length, course$type, course$act)
> tapply(course$act, course[,1:2], mean)
        length
type     Condensed Regular
  Trad        21.9    27.9
  Online      27.0    21.3
> library(sqldf)
> sqldf("select length, type, avg(act) from course group by length, type")
     length   type avg(act)
1 Condensed Online     27.0
2 Condensed   Trad     21.9
3   Regular Online     21.3
4   Regular   Trad     27.9
> fit$coef
             (Intercept)               typeOnline
                    21.9                      5.1
          lengthRegular typeOnline:lengthRegular
                     6.0                    -11.7
```

Orange Juice Example

To ascertain the stability of vitamic C in reconstituted frozen OJ concentrate stored in a refrigerator for a period of up to one week, a study was conducted on three brands of a three different times (days).




```
> oj = data.frame(brand=c(rep("A",12), rep("B",12), rep("C",12)),
  time = factor(rep(c(0,0,0,0,3,3,3,3,7,7,7,7), 3)),
  acid = c(52.6,54.2,49.8,46.5,49.4,49.2,42.8,53.2,42.7,48.8,40.4,47.6,56,48,
    49.6,48.4,48.8,44,44,42.4,49.2,44,42,43.2,52.5,52,51.8,53.6,48,47,48.2,
    49.6,48.5,43.4,45.2,47.6))
> interaction.plot(oj$time, oj$brand, oj$acid, col=1:3)
> fit = lm(acid ~ time*brand, oj)
> drop1(fit, test="F")
Single term deletions
Model:
acid ~ time * brand
           Df Sum of Sq    RSS    AIC F value Pr(F)
<none>                  287.78 86.833
time:brand  2   0.31095 288.09 82.871  0.0162 0.984

> fit = lm(acid ~ time+brand, oj)
> drop1(fit, test="F")
Single term deletions
Model:
acid ~ time + brand
       Df Sum of Sq    RSS     AIC F value     Pr(F)
<none>              288.09  82.871
time    1   210.024 498.12 100.583 23.3285 3.255e-05 ***
brand   2    32.962 321.05  82.771  1.8306    0.1767
```

Capacitor Example in R

```
> capacitor = data.frame(
  bondmat=factor(c(rep(1,12),rep(2,12),rep(3,12),rep(4,12))),
  substrate=rep(c(rep("A",4),rep("B",4),rep("C",4)), 4),
  y=c(1.51,1.96,1.83,1.98,1.63,1.92,1.8,1.71,3.04,3.16,3.09,3.5,2.62,2.82,
    2.69,2.93,3.12,2.94,3.23,2.99,1.91,2.11,1.78,2.25,2.96,2.82,3.11,3.11,
    2.91,2.93,3.01,2.93,3.04,2.91,2.48,2.83,3.67,3.4,3.25,2.9,3.48,3.51,
    3.24,3.45,3.47,3.42,3.31,3.76)
)
> attach(capacitor)
> table(substrate, bondmat)   # note orthogonal design
         bondmat
substrate 1 2 3 4
        A 4 4 4 4
        B 4 4 4 4
        C 4 4 4 4

> tapply(y, data.frame(substrate, bondmat), mean)
         bondmat
substrate      1      2     3     4
        A 1.8200 2.7650 3.000 3.305
        B 1.7650 3.0700 2.945 3.420
        C 3.1975 2.0125 2.815 3.490
> interaction.plot(substrate, bondmat, y, col=1:4)
> interaction.plot(bondmat, substrate, y, col=1:4)
> fit = aov(y~bondmat*substrate, capacitor)

> anova(fit)
Response: strength
                  Df Sum Sq Mean Sq F value    Pr(>F)
bondmat            3 8.4605 2.82017 80.7654 4.709e-16 ***
substrate          2 0.1953 0.09766  2.7968    0.0743 .
bondmat:substrate  6 7.5869 1.26449 36.2130 7.977e-14 ***
Residuals         36 1.2570 0.03492
> plot(fit)
```

Problems

1. *Montgomery 14.2.* An engineer suspects that the surface finish of metal parts is influenced by the type of paint used and the drying time. He selects three drying times and two types of paint. The data are as follows:

```
paint = data.frame(
  type=factor(c(rep(1,9), rep(2,9))),
  time=factor(rep(c(rep(20,3), rep(25,3), rep(30,3)), 2)),
  y=c(74,64,50,  73,61,44,  78,85,92,  92,86,68,  98,73,88,  66,45,85))
```

2. *Montgomery 14.4.* An experiment was conducted to determine wheter either firing temperature of furnace position affects the baked density of a carbon anode. Data are as follows:

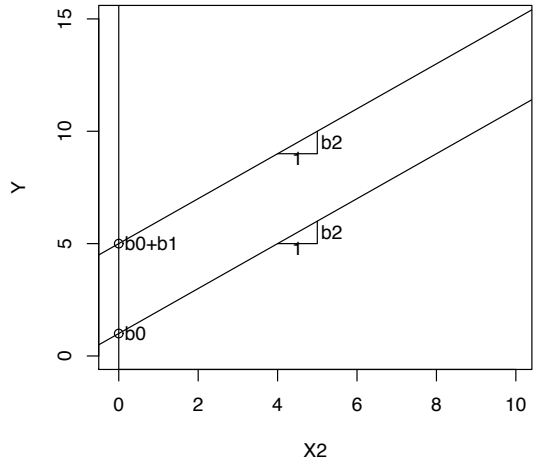
```
anode = data.frame(
  pos = factor(c(rep(1,9), rep(2,9))),
  temp = factor(rep(c(rep(800,3), rep(825,3), rep(850,3)), 2)),
  density = c( 570,565,583,  1063,1080,1043,  565,510,590,
               528,547,521,   988,1026,1004,  526,538,532))
```

What is an Interaction?

Now suppose that x_1 is **categorial** x_1 taking values 0 and 1, and x_2 is **numerical**.

Constant Slope Model

$$\hat{y} = b_0 + x_1 b_1 + x_2 b_2$$



Bottom Line ($x_1 = 0$)

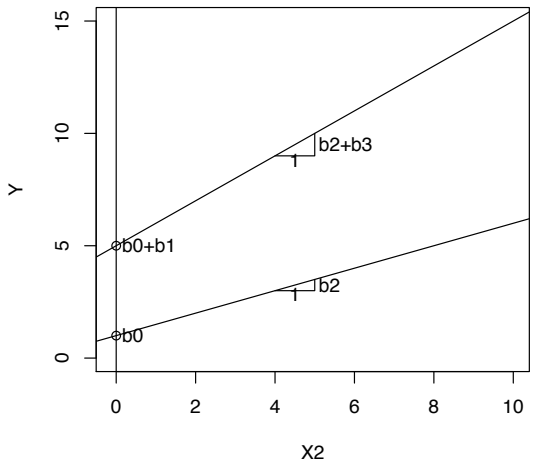
$$y = b_0 + b_2 x_2$$

Top Line ($x_1 = 1$)

$$y = (b_0 + b_1) + b_2 x_2$$

Different-Slope Model

$$\begin{aligned} \hat{y} &= b_0 + x_1 b_1 + x_2 b_2 + x_1 x_2 b_3 \\ &= (b_0 + x_1 b_1) + (b_2 + x_1 b_3) x_2 \end{aligned}$$



Bottom Line ($x_1 = 0$)

$$y = b_0 + b_2 x_2$$

Top Line ($x_1 = 1$)

$$y = (b_0 + b_1) + (b_2 + b_3) x_2$$

Newfood with a Numerical*Categorial Interaction

```
> fit = lm(sales ~ price*ad + volume, newfood)
> drop1(fit, test="F")
Single term deletions

Model:
sales ~ price * ad + volume
         Df Sum of Sq   RSS    AIC F value    Pr(>F)
<none>                26291 177.97
volume    1     51729 78019 202.08 37.3837 7.052e-06 ***
price:ad  1      8752 35042 182.87  6.3246   0.02107 *

> summary(fit)

Call: lm(formula = sales ~ price * ad + volume, data = newfood)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   12.391    111.196   0.111  0.91244
price         -7.259      2.657  -2.732  0.01324 *
ad           399.488    109.339   3.654  0.00169 **
volume        11.456      1.874   6.114 7.05e-06 ***
price:ad      -9.382      3.730  -2.515  0.02107 *

Residual standard error: 37.2 on 19 degrees of freedom
Multiple R-squared:  0.8572,Adjusted R-squared:  0.8271
F-statistic: 28.51 on 4 and 19 DF,  p-value: 8.55e-08
```

Quadratic Surfaces

Suppose we have two numerical predictors, x_1 and x_2

$$\begin{aligned}\hat{y} &= b_0 + b_1 x_1 + b_2 x_2 + b_{11} x_1^2 + b_{22} x_2^2 + b_{12} x_1 x_2 \\ &= b_0 + \mathbf{x}'\mathbf{b} + \mathbf{x}'\mathbf{B}\mathbf{x}\end{aligned}$$

where

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} b_{11} & b_{12}/2 \\ b_{12}/2 & b_{22} \end{pmatrix}.$$

We can find the optimum value as follows

$$\frac{\partial \hat{y}}{\partial \mathbf{x}} = \mathbf{b} + 2\mathbf{B}\mathbf{x} = 0$$

Solve to find the stationary point $\mathbf{x}_s$

$$\mathbf{x}_s = -\frac{1}{2}\mathbf{B}^{-1}\mathbf{b}.$$

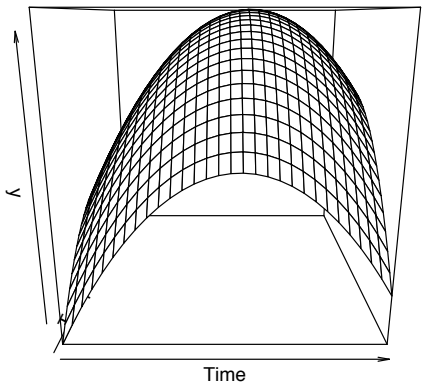
This result holds for k predictors as well.
Let λ_1 and λ_2 be the eigenvalues of $\mathbf{B}$.

- If $\lambda_1 > 0$ and $\lambda_2 > 0$, then $\mathbf{x}_s$ is a minimum
- If $\lambda_1 < 0$ and $\lambda_2 < 0$, then $\mathbf{x}_s$ is a maximum
- If $\lambda_1 \lambda_2 < 0$, then $\mathbf{x}_s$ is a saddle point

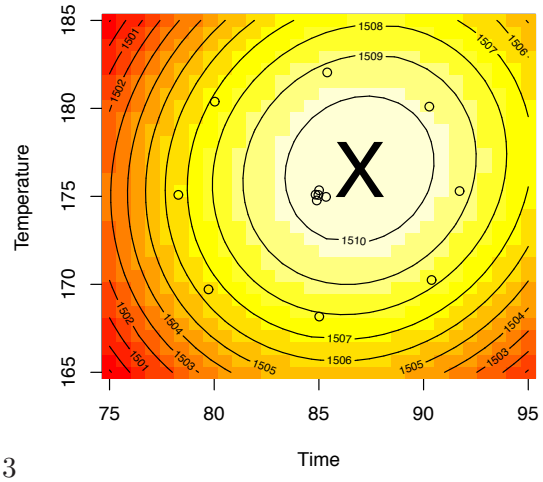
Chemical Engineering Example

```
> fit = lm(yield ~ time*temp+I(time^2)+I(temp^2), cheme)
> summary(fit)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.431e+03  1.529e+02  -9.360 3.30e-05 ***
time         7.809e+00  1.158e+00   6.744 0.000266 ***
temp         1.327e+01  1.485e+00   8.940 4.46e-05 ***
I(time^2)   -5.506e-02  4.039e-03 -13.630 2.69e-06 ***
I(temp^2)   -4.005e-02  4.039e-03  -9.916 2.26e-05 ***
time:temp    1.000e-02  5.326e-03   1.878 0.102519
```

```
> b = coef(fit)[2:3]
> B = matrix(
  c(coef(fit)[c(4,6,6,5)])*c(1,.5,.5,1),
  nrow=2)
> xs = -solve(B) %*% b/2
> xs
          [,1]
[1,]  86.94615
[2,] 176.52923
> eigen(B)$values
[1] -0.03853994 -0.05657147
```




```
> size = 30
> x1 = seq(75, 95, length=size)
> x2 = seq(165, 185, length=size)
> y = outer(x1, x2, function(x1 ,x2)
    -0.001431 + 7.809*x1
    + 13.27*x2 -0.05506*x1^2
    -0.04005*x2^2 + 0.01*x1*x2
  )
> persp(x1, x2, y, xlab="Time",
  ylab="Temperature")

> image(x1, x2, y, xlab="Time",
  ylab="Temperature")
> contour(x1, x2, y, add=T)
> points(jitter(cheme$time),
  jitter(cheme$temp))
> points(xs[1], xs[2], pch="X", cex=4)
```



Chemical Engineering Example

```
> fit = lm(yield ~ time*temp+I(time^2)+I(temp^2), cheme)
> summary(fit)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.431e+03  1.529e+02  -9.360 3.30e-05 ***
time         7.809e+00  1.158e+00   6.744 0.000266 ***
temp         1.327e+01  1.485e+00   8.940 4.46e-05 ***
I(time^2)   -5.506e-02  4.039e-03 -13.630 2.69e-06 ***
I(temp^2)   -4.005e-02  4.039e-03  -9.916 2.26e-05 ***
time:temp    1.000e-02  5.326e-03   1.878 0.102519

> b = coef(fit)[2:3]
> B = matrix(c(coef(fit)[c(4,6,6,5)])*c(1,.5,.5,1), nrow=2)
> xs = -solve(B) %*% b/2
> xs
          [,1]
[1,]  86.94615
[2,] 176.52923
> eigen(B)$values
[1] -0.03853994 -0.05657147

> size = 30
> x1 = seq(75, 95, length=size)
> x2 = seq(165, 185, length=size)
> y = outer(x1, x2, function(x1 ,x2)
    -0.001431 + 7.809*x1 + 13.27*x2 -0.05506*x1^2
    -0.04005*x2^2 + 0.01*x1*x2
  )
> persp(x1, x2, y, xlab="Time", ylab="Temperature")

> image(x1, x2, y, xlab="Time", ylab="Temperature")
> contour(x1, x2, y, add=T)
> points(jitter(cheme$time), jitter(cheme$temp))
> points(xs[1], xs[2], pch="X", cex=4)
```

IEMS 304: Homework 3
Professor Malthouse

1. Use the auto data set from JWHT problem 3.9 on page 136. The `origin` variable is categorical, where 1=US, 2=Europe and 3=Japan.

   (a) Regress `mpg` on `weight` and `year`. Examine the residual plot and comment.

   (b) Regress `mpg` on `weight`, `year`, and quadratic transformations of the two predictors. State the estimated regression equation.

   (c) Comment on the residual plot.

   (d) Is the quadratic effect for weight significant? State the null, and alternative hypothesis, and your conclusion, supported by numbers from your output.

   (e) Carefully describe the relationship between *each predictor* (i.e., weight and year) and `mpg` considering the linear and quadratic effect in combination (e.g., U or inverted-U shape, where is maximum?).

   (f) Create a plot showing how `mpg` is affected by `year` and its quadratic effect, holding the other variables constant.

2. Consider the following model:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + e_i, \quad i = 1, \ldots, n.$$

   This will be called the *uncentered* model. We discussed in class how, when $x \geq 0$, x and x^2 are often (highly) correlated. One way to reduce the correlation, yet have an equivalent model, is to mean center the x variables prior to estimation, i.e., let $\bar{x}$ be the mean of x. Let $\tilde{x}_i = x_i - \bar{x}$, then regress y on $\tilde{x}$ and $\tilde{x}^2$, i.e.,

$$y_i = \gamma_0 + \gamma_1 \tilde{x}_i + \gamma_2 \tilde{x}_i^2 + e_i = \gamma_0 + \gamma_1 (x_i - \bar{x}) + \gamma_2 (x_i - \bar{x})^2 + e_i,$$

   where γ_j are coefficients for the *centered* model. This problem will show you how the two models are equivalent to each other.

   (a) Write β_0, β_1 and β_2 as a functions of the γ_j's and $\bar{x}$. Hint: start with the second expression above, distribute γ_1 across $(x_i - \bar{x})$ and γ_2 across the expanded square. Collect the terms and set β_2 to the coefficient of x^2, β_1 equal to the coefficient of x, and β_0 equal to anything that does not include an x.

   (b) You will now check your work with the auto data set. Regress `mpg` on `weight` and `weight`^{2}. Note the coefficients.

   (c) What is the correlation between `weight` and `weight`^{2}?

   (d) What is the mean of weight?

   (e) What is the correlation between `weight` and `weight`^{2} after mean centering?

   (f) To understand why the correlation is reduced, plot weight squared against weight, and separately centered weight squared versus centered weight.

   (g) Regress `mpg` on centered weight and centered weight squared. Note the coefficients.

(h) Substitute your estimates from the previous part into the expressions you derived in part (a) and show that the equal the estimates from part (b).

3. Use the data set `part.csv`, available from canvas. This question investigates how participation in a social media contest about a brand affect future spending on the brand. A brand sponsored a social media contest. Customers in the company's database were invited to write about their relationship with the company on a social media forum. Those who participated by writing at least one word on the forum received a reward worth approximately $1, and the dummy variable `tx` indicates whether or not a customer participated. In total, 7089 customers participated, and there is a matched control group of 7089 consistent of customers who did not participate, but had similar purchase activities prior to the contest. The total sample size is thus $2 \times 7089 = 14,178$. The variable `y` records the amount spent by each customer in the week following the contest. The variable `x` gives the amount spent per week prior to the contest and will be used as a control variable to account for differences in customer loyalty. Finally, the `wc` variable gives the word count of the entries, where `wc` $= 0$ for all who did not participate. Word count measures *cognitive elaboration*. Note that `tx` $=$ (`wc` > 0).

   (a) **Model 1**: regress $\log(y+1)$ on $\log(x+1)$ and tx. Give the output.

   (b) **Model 2**: regress $\log(y+1)$ on $\log(x+1)$, tx and $\log(\texttt{wc}+1)$. Give the output.

   (c) Use the following notation in answering the questions:

$$\log(y+1) = \beta_0 + \beta_1 \log(x+1) + \beta_2 \texttt{tx} + \beta_3 \log(\texttt{wc}+1) + e,$$

   where β_3 is constrained to be 0 in Model 1 and log is the natural log. Based on Model 1, does participation have a significant effect on future spending? Explain. Note: to receive full credit you should state null and alternative hypotheses and do something to determine whether H_0 can be rejected at the 5% level.

   (d) Using Model 1, post-period *spending* is how many *times* greater for those who participate than for those who do not? Note that this question asks about *spending* and not log spending. Another way to ask this question is, suppose there are two people with identical pre-period spending, but one participates and the other does not. If y_1 is the post-contest spending of a participant, and y_0 is the post-contest spending of a non-participant, how many times greater is $(y_1 + 1)$ than $(y_0 + 1)$?

   (e) Is (y+1) proportional to (x+1), i.e., is spending in the week after the contest proportional to pre-contest spending? How do you know? Note: to receive full credit, state a null and alternative hypothesis and do something to determine whether or not H_0 can be rejected.

   (f) Why is the magnitude of the `tx` variable so different in Model 2 (0.050) than in Model 1 (0.244)?

   (g) Now consider Model 2. How do the results from Model 2 change your conclusions about how participation affects future spending. I am looking for you to summarize the key learnings from Model 2 succinctly.

   (h) Generate the normal probability plot for Model 2. What specifically does the plot tell you, and how does it (i.e., what the plot tells you) affect the conclusions you have drawn from the previous parts.

(i) What do the results of this analysis suggest the company should do in the future when designing social media contests?

Logistic Regression

Suppose I want to predict a probability as a function of some independent variables, e.g., a yes-no response variable.

Linear regression problematic:

1. Probabilities are between 0 and 1; $\alpha + \beta x$ is unbounded
2. Residuals can take only two values — certainly not normally distributed
3. Variance of response, $\pi(1 - \pi)$, depends on the mean and is therefore heteroscedastic

Possible solutions:

1. Logistic regression
2. Discriminant analysis and naive/idiot Bayes classifiers

Outline of lecture:

1. The logistic regression model
2. Interpreting the results
3. Scoring other data sets

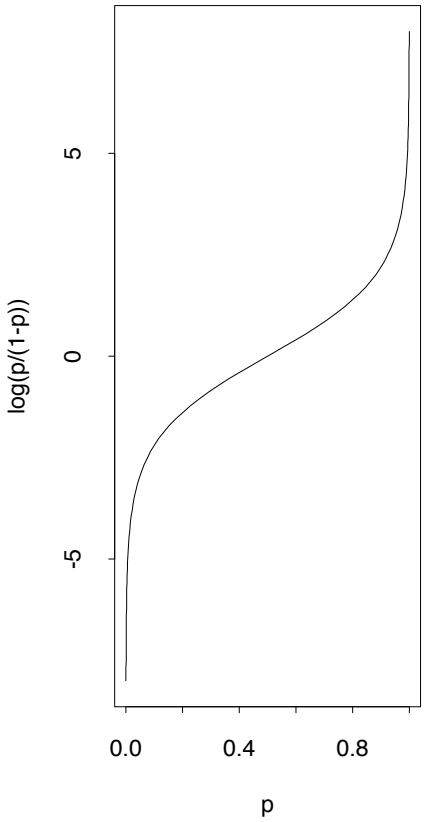
Logistic Regression Model

- Let $(x_1, y_1), \ldots, (x_n, y_n)$ be a random sample from some population where $y_i \in \{0, 1\}$
- Let $\pi_i = E(Y_i)$, i.e., probability person i responds "yes"

- We want to model π_i, but can't
- Instead, model *log-odds ratio* or *logit* of π_i
- Odds $= \pi/(1-\pi)$
- Logistic regression:

$$\begin{aligned} \text{logit}(\pi_i) &= \log\left(\frac{\pi_i}{1-\pi_i}\right) \\ &= \alpha + \beta x_i + e_i \end{aligned}$$

- Estimate α and β with maximum likelihood



In general we have p predictors with

$$\text{logit}(\pi) = \alpha + \beta_1 x_1 + \cdots \beta_p x_p$$

Estimating Probabilities

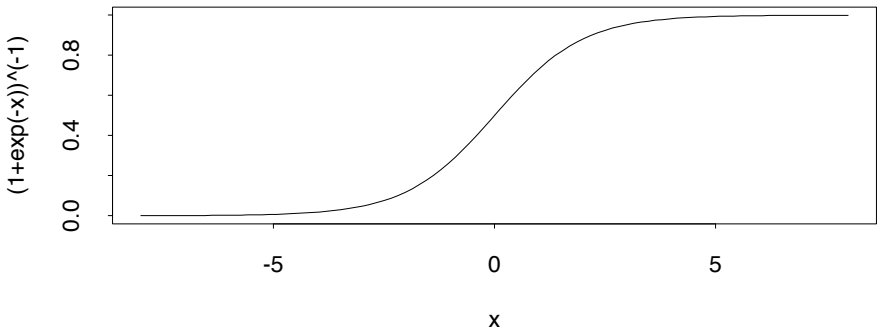
The logistic regression model is

$$\log\left(\frac{\hat{\pi}_i}{1-\hat{\pi}_i}\right) = \alpha + \beta x_i$$

The log-odds ratio is interesting, but we often need probabilities at the end of the analysis, i.e., what is the probability of response? Answer: solve the above equation for π.
Let $\eta_i = \alpha + \beta x_i$

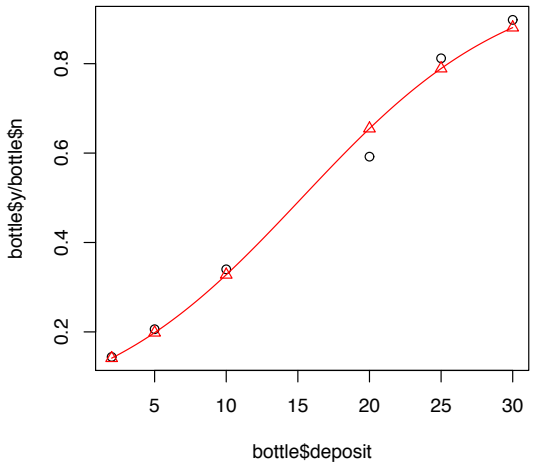
$$\frac{\hat{\pi}_i}{1-\hat{\pi}_i} = \exp(\eta_i)$$
$$\hat{\pi}_i = \exp(\eta_i) - \hat{\pi}_i \exp(\eta_i)$$
$$\hat{\pi}_i(1+\exp(\eta_i)) = \exp(\eta_i)$$
$$\hat{\pi}_i = \frac{\exp(\eta_i)}{(1+\exp(\eta_i))} = (1+\exp(-\eta_i))^{-1}$$

This is the *logistic function*. It's the most commonly used *squashing* or *sigmoidal* function used by neural network practitioners.



Bottle return problem

A carefully controlled experiment was conducted to study the effect of the size of the deposit on the likelihood that a returnable one-liter soft-drink bottle will be returned. A bottle return was coded 1 and no return was coded 0. The data show the number of bottles that were returned out of 500 sold at each of the six deposit levels. Plot estimated proportions against X. Estimate a logistic regression model and superimpose the fitted values. Interpret the parameter estimates.




```
bottle = data.frame(n=rep(500,6), deposit=c(2,5,10,20,25,30),y=c(72,103,170,296,406,449))
plot(bottle$deposit, bottle$y/bottle$n)
bot2 = data.frame(x=rep(bottle$deposit,2), y=c(rep(0,6), rep(1,6)),
  count=c(500-bottle$y, bottle$y))
fit = glm(y~x, bot2, family=binomial, weight=count)
points(bottle$deposit, fit$fitted.values[1:6], pch=2, col=2)
x = seq(2,30,length=100)
lines(x, predict(fit, data.frame(x=x), type="response"), col=2)
summary(fit)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.076565   0.084839  -24.48   <2e-16 ***
x            0.135851   0.004772   28.47   <2e-16 ***
```

$$\log\left(\frac{\pi}{1-\pi}\right) = -2.08 + 0.136x$$

More bottle return problem

- Find a 95% confidence interval for β and test whether it is different from 0.

```
> confint(fit)
                 2.5 %     97.5 %
(Intercept) -2.2449682 -1.9123046
x            0.1266071  0.1453175
```

- What is the probability that a bottle will be returned when the deposit is 15 cents?

```
> eta = fit$coef[1] + fit$coef[2]*15  # linear predictor
> eta
-0.03880299
> predict(fit, data.frame(x=15))
-0.03880299

> 1/(1+exp(-(fit$coef[1] + fit$coef[2]*15))) # unlogit it
  0.4903005
> predict(fit, data.frame(x=15), type="response")
0.4903005
```

- For which deposit amount do you expect 75% of the bottles to be returned?

$$\log\left(\frac{.75}{1-.75}\right) = -2.08+0.136x \quad \Rightarrow \quad x = \frac{\log 3 + 2.08}{0.136} = 23.37$$

```
(log(3)-fit$coef[1])/fit$coef[2]
  23.37253
```

Logistic Regression Model

```
> fit = glm(2-q11 ~ food+atmosph+service, binomial, pizzarest)
> summary(fit)

Call:
glm(formula = 2 - q11 ~ food + atmosph + service, family = binomial,
    data = pizzarest)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.40245    0.40647  -5.911 3.41e-09 ***
food         1.38776    0.09934  13.970  < 2e-16 ***
atmosph     -0.13490    0.10159  -1.328    0.184
service      0.39515    0.09903   3.990 6.60e-05 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2345.0  on 14732  degrees of freedom
Residual deviance: 2026.2  on 14729  degrees of freedom
  (1653 observations deleted due to missingness)
AIC: 2034.2
```

Odds Ratio: e^{β}

```
> fit$coef
(Intercept)        food     atmosph     service
 -2.4024497   1.3877596  -0.1349032   0.3951547

> exp(fit$coef)
(Intercept)        food     atmosph     service
  0.0904960   4.0058652   0.8738005   1.4846138
```

What does the odds ratio mean? Consider two people:

1. `food` $= 4$, `atmosph` $= 4$, `service` $= 4$
2. `food` $= 5$, `atmosph` $= 4$, `service` $= 4$

The odds for person 1 are ($\pi_1 =$ prob person 1 says "yes")

$$\frac{\pi_1}{1-\pi_1} = \exp(\alpha + 4\beta_1 + 4\beta_2 + 4\beta_3)$$

The odds for person 2 are ($\pi_2 =$ prob person 2 says "yes")

$$\begin{aligned}
\frac{\pi_2}{1-\pi_2} &= \exp(\alpha + 5\beta_1 + 4\beta_2 + 4\beta_3) \\
&= \exp(\alpha + 4\beta_1 + \beta_1 + 4\beta_2 + 4\beta_3) \\
&= \exp(\alpha + 4\beta_1 + 4\beta_2 + 4\beta_3)e^{\beta_1} \\
&= \frac{\pi_1}{1-\pi_1}e^{\beta_1}
\end{aligned}$$

Thus, by increasing `food` by 1, the odds of saying "yes" are multiplied by e^{β_1}

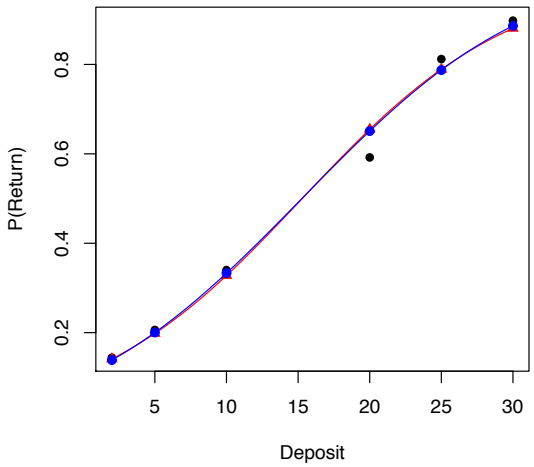
Generalized Linear Models

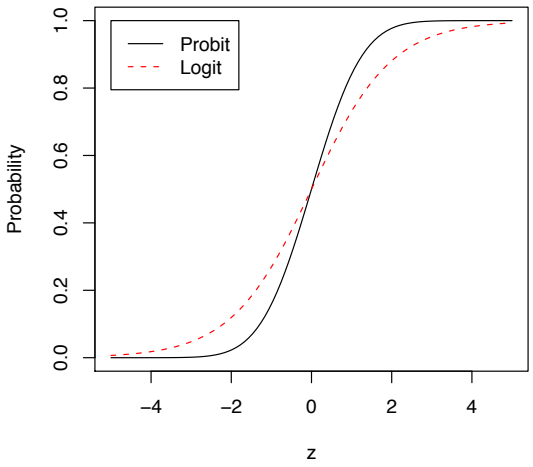
- We observe $(\mathbf{x}_i, y_i)$, $i = 1, \ldots n$ and $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^T$
- Classical linear model:
$$y_i = \boldsymbol{\beta}^T \mathbf{x}_i + \epsilon_i,$$
where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$
- Note that $E(y_i) = \mu_i = \boldsymbol{\beta}^T \mathbf{x}_i$
- The generalized linear model (GLM) involve two additional functions
$$g(\mu_i) = \eta_i = \boldsymbol{\beta}^T \mathbf{x}_i$$
  - *Linear component*: $\eta_i = \boldsymbol{\beta}^T \mathbf{x}_i$
  - *Link function*: let univariate function g be monotonic and differentiable
  - *Random component*: y_i are independent and from an exponential family, which implies the variance of y_i depends on μ_i through a variance function $\text{var}(y_i) = \phi V(\mu_i)$ where ϕ is called the *dispersion parameter*.
- The classical linear model assumes $g(\mu) = \mu$, the identity function, and y_i has a normal distribution
- Logistic regression assumes $g(\mu) = \log[\mu/(1-\mu)]$ and y_i is a Bernoulli trial $(V(y) = \mu(1-\mu))$. Other possible links for binary responses include
  - Probit $g(\mu) = \Phi^{-1}(\mu)$, where Φ is the cumulative standard normal distribution
  - Complementary log-log: $g(\mu) = \log(-\log(1-\mu))$
- Other frequently-used distributions for y include Poisson and Gamma

Bottle return problem: probit model

Add `(link="probit")` to change to probit link function.

```
> plot(bottle$deposit, bottle$y/bottle$n, xlab="Deposit", ylab="P(Return)", pch=16)
> fit = glm(y~x, bot2, family=binomial, weight=count)
> points(bottle$deposit, fit$fitted.values[1:6], pch=17, col=2)
> x = seq(2,30,length=100)
> lines(x, predict(fit, data.frame(x=x), type="response"), col=2)

# now fit and plot probit model
> fit2 = glm(y~x, bot2, family=binomial(link="probit"), weight=count)
> summary(fit2)
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.250031   0.047962  -26.06   <2e-16 ***
x            0.081897   0.002667   30.71   <2e-16 ***

> points(bottle$deposit, fit2$fitted.values[1:6], pch=19, col=4)
> lines(x, predict(fit2, data.frame(x=x), type="response"), col=4)

# show logit versus probit
> z = seq(-5, 5, length=100) # +/-5 std devs
> plot(z, pnorm(z), type="l", ylab="Probability")
> lines(z, 1/(1+exp(-z)), col=2, lty=2)
> legend(-5, 1, c("Probit", "Logit"), lty=1:2, col=1:2)
```





Pizza Hut Data

A random sample of $n = 220$ consumers were surveyed to evaluate the effect of price on the purchase of a pizza from Pizza Hut. Subjects were asked to suppose that they were going to have a large 2-topping pizza delivered to their residence. They were asked to select from either Pizza Hut or another pizzeria of their choice. The price they would have to pay to get a Pizza Hut pizza differed from survey to survey. The dependent variable is whether or not a student selected Pizza Hut and independent variables are price and sex.

1. Fit a logistic regression model.
2. Test whether the overall model is significant.
3. State the estimated regression equation.
4. Interpret the meaning of the coefficients and odds ratios.
5. Test whether each variable is significant.
6. Predict the probability that a female student will select Pizza Hut if the price is \$8.99. Repeat this for prices of \$11.49 and \$13.99.
7. Regress purchase on price for males only. Note the regression equation.
8. Regress purchase on price for females only. Note the regression equation.
9. Fit a logistic regression model with different slopes for males and females. Test whether the slopes are equal.

Maximum Likelihood Estimation

- Until now we've been using least squares to estimate parameters, e.g., regression

$$\min_{\alpha,\beta} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \min_{\alpha,\beta} \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2$$

- We've used RSS as the objective and to evaluate fit, e.g.,

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- Least squares won't work with other methods such as logistic regression or latent class analysis
- Alternative approach:
  - Maximum likelihood estimation (MLE)
  - (-2 log likelihood) is generalization of RSS
- Goals for today:
  - What is estimation by maximum likelihood?
  - What is -2 log likelihood?

MLEs of Proportions

- Suppose we draw a random sample of size $n = 5$ from some population, send them an offer, and $x = 2$ respond. What is our best guess of the response probability?
- Basic stats answer: $p = 2/5 = .4$. Rationale: common sense
- Maximum likelihood answer: pick π so that the probability of observing 2 responses in 5 tries given π is maximized
  - Let $L(\pi)$ be the probability (likelihood) of observing the data if the probability of response is π

$$L(\pi) = \binom{5}{2}\pi^2(1-\pi)^3$$

  - Let $l(\pi) = \log L(\pi)$, called the *log-likelihood*

$$l(\pi) = \log(10) + 2\log(\pi) + 3\log(1-\pi)$$

$$\frac{dl(\pi)}{d\pi} = \frac{2}{\pi} - \frac{3}{1-\pi} = 0 \Longrightarrow \hat{\pi} = \frac{2}{5}$$

Guess (π)	$L(\pi)$	$l(\pi)$	$-2l(\pi)$
.3	.3087	−1.1754	2.3508
.35	.3364	−1.0894	2.1788
.4	.3456	−1.0625	2.1249
.45	.3369	−1.0879	2.1759
.5	.3125	−1.1632	2.3263

MLEs of Means

- Suppose we draw a random sample of size $n = 3$ from a normal population with unknown mean μ and known variance $\sigma^2 = 5$. The observed values are $x_1 = 4$, $x_2 = 5$, and $x_3 = 6$. What is the best guess of μ?
- Basic stats answer: $\bar{x} = (4 + 5 + 6)/3 = 5$. Rationale: common sense
- Maximum likelihood answer: pick μ so that the probability of observing the three values given μ is maximized

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(x-\mu)^2\right]$$

$$L(\mu) = \prod_{i=1}^{3} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(x_i-\mu)^2\right]$$

$$\begin{aligned} l(\mu) &= \sum_{i=1}^{3}\left[-\frac{1}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}(x_i-\mu)^2\right] \\ &= -k_1 - k_2\sum_{i=1}^{3}(x_i-\mu)^2 \end{aligned}$$

$$\frac{dl(\mu)}{d\mu} = 2k_2\sum_{i=1}^{3}(x_i-\mu) = 0$$

$$\Longrightarrow \mu = \frac{1}{3}\sum_{i=1}^{3} x_i$$

$$\text{Note } \frac{d^2 l(\mu)}{d\mu^2} = -3 < 0$$

Likelihood Function for Logistic Regression

- Assume we have a random sample (observations independent, each have same probability of selection) and that we make two measurements on each observation (x_i, y_i), where y_i is a 0-1 variable and $i = 1, \ldots, n$
- Let $\pi_i = P(y_i = 1)$, i.e., prob(person i says yes), and

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \alpha + \beta x_i$$

- Note that the probability distribution for person i is

$$f_i(y_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i}$$

- Since observations are independent, the probability distribution (likelihood) and log-likelihood for our sample is

$$f(y_1, \ldots, y_n) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i}$$

$$\begin{aligned}
\log(f) &= \sum_{i=1}^{n} [y_i \log(\pi_i) + (1-y_i)\log(1-\pi_i)] \\
&= \sum_{i=1}^{n} \left[y_i \log(\frac{\pi_i}{1-\pi_i}) + \log(1-\pi_i) \right] \\
&= \sum_{i=1}^{n} y_i(\alpha + \beta x_i) - \sum_{i=1}^{n} \log[1 + \exp(\alpha + \beta x_i)]
\end{aligned}$$

We maximize this with respect to α and β

Log-likelihood, deviance, AIC

- Log-likelihood:

$$l = \log(L) = \sum_{i=1}^{n} y_i(\alpha + \beta x_i) - \sum_{i=1}^{n} \log[1 + \exp(\alpha + \beta x_i)]$$

- For bottle return problem:

```
> eta = fit$coef[1]+fit$coef[2]*bot2$x
> round(eta,2)
 [1] -1.80 -1.40 -0.72  0.64  1.32  2.00 -1.80 -1.40 -0.72  0.64  1.32  2.00
> fit$y
 1  2  3  4  5  6  7  8  9 10 11 12
 0  0  0  0  0  0  1  1  1  1  1  1
> sum(bot2$count*fit$y*eta) - sum(bot2$count*log(1+exp(eta)))
[1] -1531.436
> logLik(fit)
'log Lik.' -1531.436 (df=2)
```

- We will usually use the *deviance*: $-2l$. Think of deviance as RSS, measuring how much is unexplained by the model

```
> -2*logLik(fit)
[1] 3062.872
> deviance(fit)
[1] 3062.872
```

- Or we will use AIC = deviance + 2(number parameters), which penalizes the fit measure by $2p$

```
> AIC(fit)
[1] 3066.872
```

Residual versus null deviance

```
> summary(fit)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.076565   0.084839  -24.48   <2e-16 ***
x            0.135851   0.004772   28.47   <2e-16 ***

    Null deviance: 4158.9  on 11  degrees of freedom
Residual deviance: 3062.9  on 10  degrees of freedom
AIC: 3066.9
```

- R reports AIC and the *residual deviance* for the *full model*, i.e., the one with all predictors in the model.
- It also reports the *null deviance*, which is the deviance of the intercept-only model:

  ```
  > fit.null = glm(y~1, bot2, family=binomial, weight=count)
  > deviance(fit.null)
  [1] 4158.862
  > fit$null.deviance
  [1] 4158.862
  ```

  Think of the null deviance as SST, measuring how much variation in Y is unexplained by the intercept model.
- The *difference* between them measures how much variation is explained by the model and plays the role of extra sums of squares.

  ```
  > fit$null.deviance - fit$deviance
  [1] 1095.99
  ```

- The *difference* has a chi-squared distribution and can test overall signifiance, H_0 : all $\beta_j = 0$.

  ```
  > 1-pchisq(fit$null.deviance - fit$deviance, 1)
  [1] 0
  ```

Likelihood-Ratio Test

- Consider the *full model* with $p + q$ predictors

$$\log\left(\frac{\pi}{1-\pi}\right) = \alpha+\beta_1 x_1+\cdots+\beta_p x_p+\beta_{p+1}x_{p+1}+\cdots+\beta_{p+q}x_{p+q}$$

- The *reduced model* has only p predictors, i.e., the last q predictors have been dropped.

$$\log\left(\frac{\pi}{1-\pi}\right) = \alpha + \beta_1 x_1 + \cdots + \beta_p x_p$$

- Let D_1 be the deviance of the full model, and D_2 be the deviance of the reduced model.
- We can test $H_0 : \beta_{p+1} = \cdots = \beta_{p+q} = 0$ with the test statistic $D_2 - D_1$, which has a chi-square distributions with q degrees of freedom

Likelihood-Ratio Test For Single Parameters

How can we use the likelihood-ratio test to compare the fits of the two models:

$$\log\left(\frac{\pi}{1-\pi}\right) = \alpha + \beta_1\texttt{food} + \beta_2\texttt{atmosph} + \beta_3\texttt{service}$$

$$\log\left(\frac{\pi}{1-\pi}\right) = \alpha + \beta_1\texttt{food} + \beta_3\texttt{service}$$

i.e., does `atmosph` affect the response?

Answer:

- Let $-2l_3$ be the maximized log-likelihood for the three-predictor model (2026.183 from slide 123)
- Let $-2l_2$ be the maximized log-likelihood for the two-predictor model (2027.956)

Then $(-2l_2) - (-2l_3))$ has a chi-squared distribution with 1 degree of freedom.

```
> drop1(fit, test="Chisq")
Model:
2 - q11 ~ food + atmosph + service
        Df Deviance    AIC     LRT   Pr(Chi)
<none>       2026.2 2034.2
food     1   2207.7 2213.7 181.555 < 2.2e-16 ***
atmosph  1   2028.0 2034.0   1.773    0.1830
service  1   2041.6 2047.6  15.398  8.71e-05 ***

> d2=deviance(glm(2-q11~food+service,binomial,pizzarest,subset=(!is.na(atmosph))))
> d2
[1] 2027.956
> 1-pchisq(d2-deviance(fit),1)
[1] 0.1829724
```

LRT for Fitness Club Data

- Test whether payment type is signifiance, while controlling for log downpayment and log use.

```
> fit = glm(default~log(downpmt+1)+log(use+1)+pmttype, binomial, default)
> summary(fit)
Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        2.58488    0.08669   29.82   <2e-16 ***
log(downpmt + 1)  -0.69243    0.01822  -38.00   <2e-16 ***
log(use + 1)      -1.52831    0.04855  -31.48   <2e-16 ***
pmttypeStatement  -0.72340    0.05361  -13.49   <2e-16 ***
pmttypeCheck EFT  -3.99025    0.14181  -28.14   <2e-16 ***
pmttypeCredit EFT -2.94096    0.10685  -27.52   <2e-16 ***

    Null deviance: 17734  on 24842  degrees of freedom
Residual deviance: 11477  on 24837  degrees of freedom

> drop1(fit, test="Chisq")
Model:
default ~ log(downpmt + 1) + log(use + 1) + pmttype
                 Df Deviance   AIC    LRT   Pr(Chi)
<none>                11477 11489
log(downpmt + 1)  1    13236 13246 1758.9 < 2.2e-16 ***
log(use + 1)      1    12933 12943 1455.9 < 2.2e-16 ***
pmttype           3    14324 14330 2846.6 < 2.2e-16 ***
```

- Where does 2846.6 come from?

```
> fit2 = glm(default~log(downpmt+1)+log(use+1), binomial, default)
> deviance(fit2)-deviance(fit)
[1] 2846.583
> 1-pchisq(deviance(fit2)-deviance(fit), 3)
[1] 0
```

Two-Step Models

- Problem: build a regression model to predict a dependent variable that is a dollar amount
- Problem: sometimes models that predict response don't predict profitability and visa versa
- Problems with linear regression:
  1. A very high percentage of the y values are 0, e.g., response rates of 2% are not uncommon $\Longrightarrow$ 98% zeros.
  2. A person cannot spend negative dollars, but $\hat{y}$ can assume negative values (similar problem to logistic regression — $\beta_0 + \beta_1 x$ is unbounded)
- Possible improvement: two-step models
  1. *Response Model.* Predict response with logistic regression
  2. *Conditional Demand Model.* Predict dollar amount with a linear regression model for *only those who responded* (use select cases where response=yes)
  3. Compute expected dollar amount as follows:

$$E(Y_i) = 0P(\text{No}) + E(Y|\text{Yes})P(\text{Yes})$$

- See article by Elkan on Blackboard on "Heckman correction:" include $\hat{p}$ values from response model as predictor in conditional demand model

Predictive Accuracy of Classifiers

Suppose we are using a GLM (e.g., logit or probit) with p parameters to estimate a binary response

- GLMs minimize deviance, which has the same problems in detecting overfitting as RSS
- AIC = deviance + $2p$ is a commonly-used penalized measure
- Alternative measures include the *classification rate*, which is the percentage of correctly classified cases, and the *misclassification rate*, which is 1− classification rate. These should be computed on non-training data.
- Additional complication: how do we classify observations? Let c be some cutoff and p_i be the predicted probably for observation i. Classify i as a "yes" when $p_i > c$ and "no" otherwise.
- The choice of c depends on misclassification costs. Consider:
  - *Sensitivity*—true positive rate
  - *Specificity*—true negative rate

  Depending on the situation, the "cost" of a false negative may be much greater than the cost of a false positive, e.g., airport security screening or disease detection versus spam filters.
- *Receiver operating characteristic* (ROC) curves plot the true positive rate (specificity) against false positives (1− sensitivity) for different values of c.

Defaulting Customer Example

- Note that if we classify all observations as "no" then the classification rate is 88.3%. This is a stupid classifier, but we must beat it!

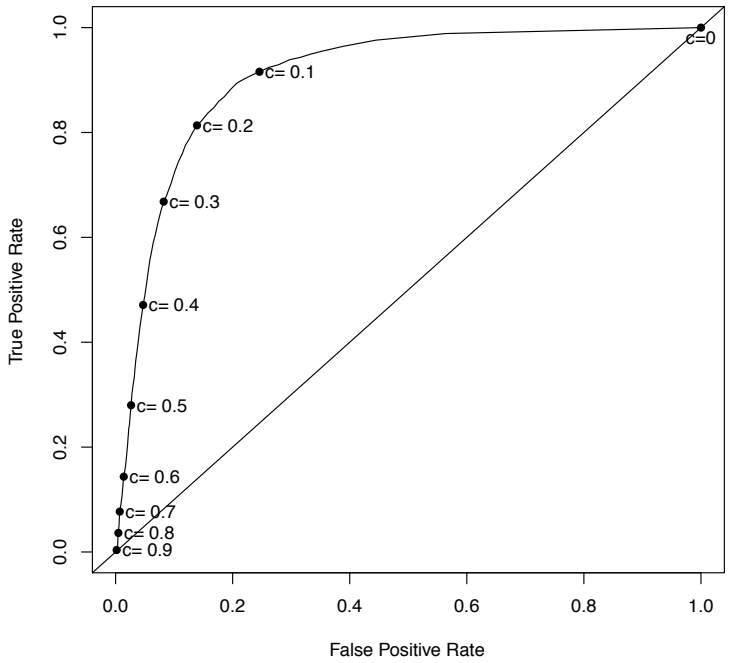
```
> table(default$default)/length(default$default)
       0        1
0.882963 0.117037
```

- Consider the following improved model

```
> fit = glm(default ~ log(downpmt+1)+pmttype+use+age+gender, binomial, default)
> tab=table(default$default, fit$fitted.values>.5)   # c=.5
> tab

    FALSE  TRUE
  0 21468   584
  1  2105   818
>
> sum(diag(tab))/sum(tab) # classification rate, (21468+818)/24975
[1] 0.8923323

> prop.table(tab,1)  # condition on observed values for FPR, TPR, etc.

         FALSE       TRUE
  0 0.97351714 0.02648286
  1 0.72015053 0.27984947

> tab=table(default$default, fit$fitted.values>.3) # c=.3
> sum(diag(tab))/sum(tab)
[1] 0.8886086
> prop.table(tab,1)  # TPR increases, but so does FPR

         FALSE       TRUE
  0 0.91783058 0.08216942
  1 0.33185084 0.66814916
```

ROC Curves

```
a = (0:100)/100 # different cut points, don't call it c!
tpr = rep(NA, 101)  # true positive rate
fpr = rep(NA, 101)  # false positive rate
denom=table(default$default)
for(i in 1:101){
  num=table(default$default[fit$fitted.values>=a[i]])
  fpr[i] = num[1]/denom[1]
  tpr[i] = num[2]/denom[2]
}
plot(fpr, tpr, type="l", xlab="False Positive Rate", ylab="True Positive Rate")
abline(0,1)
b = (0:10)*10+1
points(fpr[b], tpr[b], pch=16)
text(fpr[b[-1]]+.01, tpr[b[-1]], paste("c=",as.character(a[b[-1]])), adj=0)
text(1,.98, "c=0")

library(pROC)
plot.roc(default$default, fit$fitted.values)
```



Area Under the ROC Curve (AUC) is used evaulate models

IEMS 304: Homework 4
Professor Malthouse

1. In an experiment testing the effect of a toxic substance, 1,500 experimental insects were divided at random into six groups of 250 each. The insects in each group were exposed to a fixed dose of the toxic substance. A day later, each insect was observed. Death from exposure was scored 1 and survival was scored 0. The results are in the data frame below, where x_j is the dose level (on a logarithmic scale), administered to the insects in group j and y_j denotes the number of insects that dies out of the $n_j = 250$ in the group. As a hint, study the bottle return problem we worked in class.

```
toxicity = data.frame(x=1:6, n=rep(250,6), y=c(28,53,93,126,172,197))
```

   (a) Plot the estimated proportions $p_j = y_j/n_j$ against x_j. Does the plot support the analyst's belief that the logistic response function is appropriate?

   (b) Find the MLEs of the slope and intercept, e.g., using `glm` in R. State the fitted response function and superimpose it on the scatterplot from part (a).

   (c) Obtain $\exp(b_1)$ and interpret this number.

   (d) What is the estimated probability that an insect dies when the dose level is $x = 3.5$?

   (e) What is the estimated median lethal dose—that is, the dose for which 50% of the experimental insects are expected to die?

   (f) Find a 99% confidence interval for β_1. Convert it into ones for the odds ratio.

   (g) Generate an ROC curve and find the area under the curve.

2. The marketing manager for a large nationally franchised lawn service company would like to study the characteristics that differentiate home owners who do and do not have a lawn service. A random sample of 30 home owners located in a suburban area near a large city was selected. Predictor variables include household income ($K), lawn size (square feet K), attitude toward outdoor recreational activities (1=positive, 0=negative), number of teenagers in the household, and age of the head of household.

   (a) Generate a scatterplot matrix of the six variables. Comment on anything unusual or problematic.

   (b) Generate a correlation matrix of the six variables. Comment on substantial correlations.

   (c) Fit a logistic regression of whether a household has a lawn service (`lawnserv`) on the other five variables and test whether the overall regression model is significant.

   (d) State the estimated regression equation

   (e) Estimate the probability of purchasing a lawn service for a 45-year-old home owner with a family income of $70K, a lawn size of 3,000 square feet, a negative attitude towards outdoor recreation, and one teenager in the household.

   (f) Which predictor variables are significantly different from zero (use alpha=.05)?

   (g) Estimate a logistic regression model with income, attitude, and teenage as predictors. State the estimated regression equation and indicate which variables are significant.

(h) Estimate a logistic regression model with lawnsize, attitude, and teenage as predictors. State the estimated regression equation and indicate which variables are significant.

(i) Estimate a logistic regression model with HOH age, attitude, and teenage as predictors. State the estimated regression equation and indicate which variables are significant.

(j) Write a short paragraph with your final conclusions.

3. Suppose we draw a random sample of size n from a distribution with pdf $f(t) = \lambda e^{-\lambda t}$, where $t \geq 0$ and $\lambda > 0$ is unknown. Let $t_1, t_2, \ldots, t_n$ denote the observations from this pdf. State the likelihood function, $L(\lambda)$. Take logs of both sides to find the log-likelihood function, $l(\lambda)$. Maximize $l(\lambda)$, giving the MLE of λ.

4. Consider a subscription-based service such as Netflix. Assume that customers join at some point and pay some monthly fee until they decide to cancel the service. Assume that (1) after canceling they never return; (2) the retention rate r (i.e., the probability that a customer is retained in any period) is constant over time and that all customers have the same retention rate; (3) the event that a customer cancels in one period is independent of the event that the customer cancels in any other period. Let T be the time of cancelation. Then the probability mass funciton is

$$P(T = t) = r^{t-1}(1 - r).$$

Suppose you have a sample of n customers and that customer $i = 1, \ldots, n$ canceled at time t_i. Estimate the retention rate using maximum likelihood. Hint: $n\bar{t} = \sum_{i=1}^{n} t_i$, where $\bar{t}$ is the sample mean of the observed times.

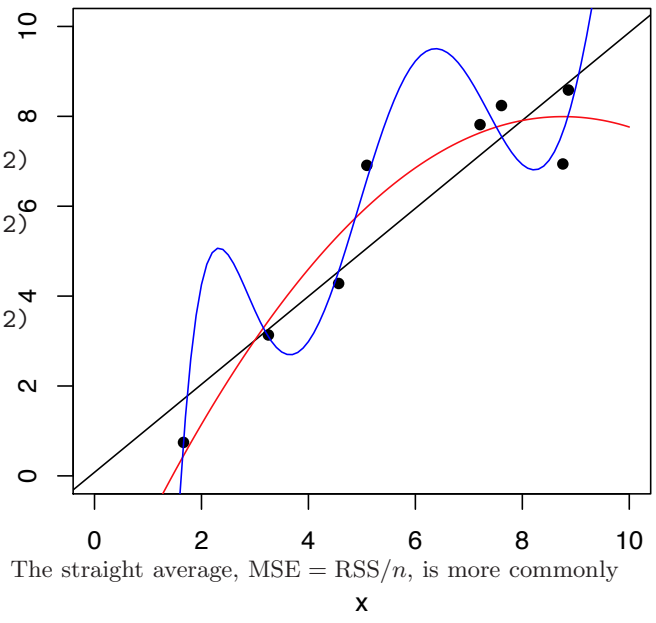
How to Measure Predictive Accuracy

- Assume *training data set* $(\mathbf{x}_i, y_i)$, where $i = 1, \ldots, n$ and y_i is numerical.
- Estimate model f with p parameters and summarize residuals[2] with, e.g.,

$$\text{RSS} = \sum_{i=1}^{n} [y_i - f(\mathbf{x}_i)]^2, \quad \text{MSE} = \frac{\text{RSS}}{n}, \text{or} \quad R^2 = 1 - \frac{\text{RSS}}{\text{TSS}}$$

- Problem: if measure computed on the data that was used to estimate the model, it will be *optimistic*. Such a rate is called the apparent rate. The model is "fine-tuned" to do well on this data set and may "capitalize on chance."
- Example: $y = x + e$ where $n = 8$ and $e \sim \mathcal{N}(0, 1)$

```
> set.seed(12345)
> train = data.frame(x = runif(8)*10)
> train$y = train$x + rnorm(8)

> fit1 = lm(y~x, train) #black
> fit2 = lm(y~x + I(x^2), train) #red
> fit3 = lm(y~poly(x, 6), train) #blue
> mean((train$y - predict(fit1, train))^2)
[1] 1.043647
> mean((train$y - predict(fit2, train))^2)
[1] 0.4878706
> # degree 6 polynomial gives best fit
> mean((train$y - predict(fit3, train))^2)
[1] 0.2301981
```



[2] Until now, MSE = RSS$(n - p - 1)$, which is unbiased. The straight average, MSE = RSS$/n$, is more commonly used when computing out-of-sample estimates.

Two Approaches for Honest Estimates of Prediction Error

- *Penalized estimates*, e.g.,

$$R_a^2 = 1 - \frac{\text{RSS}/(n-p-1)}{\text{TSS}/(n-1)} \quad \text{or} \quad \text{AIC} = \text{deviance} + 2p$$

  but for many models p is not known (e.g., neural networks) and there are different penalties

```
> summary(fit1)$adj.r.squared
[1] 0.8237407
> # adjusted R^2 gets it wrong!
> summary(fit2)$adj.r.squared
[1] 0.9011255
> summary(fit3)$adj.r.squared
[1] 0.7667342
```

```
> AIC(fit1)
[1] 29.04479
> AIC(fit2) # AIC gets it wrong!
[1] 24.96138
> AIC(fit3)
[1] 26.95249
```

- *Out-of-sample estimates*, e.g., "conjure up" a *test set* of 10,000 cases and use them to evaluate fit, but not to estimate models

```
> test = data.frame(x = runif(10000)*10)
> test$y = test$x + rnorm(10000)
> mean((test$y-predict(fit1, test))^2)  # model 1, close to true value
[1] 1.006563
> mean((test$y-predict(fit2, test))^2)  # model 2 overfits
[1] 2.450303
> mean((test$y-predict(fit3, test))^2)  # model 3 really overfits
[1] 636.7939
```

Variable Selection Problems

- Now add two "decoy" predictors (that are just noise)

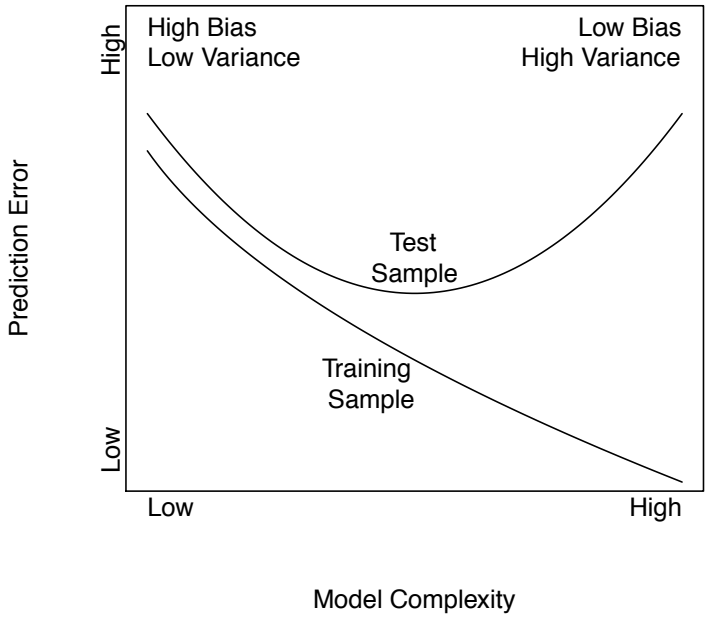
```
> train$x2 = runif(8)
> train$x3 = runif(8)
> test$x2 = runif(10000)
> test$x3 = runif(10000)

> fit4 = lm(y ~ x+x2+x3, train)
> mean((train$y - predict(fit4, train))^2)
[1] 0.7142952
> mean((test$y - predict(fit4, test))^2)
[1] 2.234198
> AIC(fit4)
[1] 30.01134
> summary(fit4)$adj.r.squared
[1] 0.8190464
```

  In this case R_a^2, AIC and the test set get it right.

- Key points
  - When building a model, the modeler must decide on (1) how flexible model should be (e.g., degree of polynomial) and (2) which variables should be included.
  - We should not use RSS, MSS, R^2, etc. because they are optimistic.
  - These decisions can be made with penalized or out-of-sample measures.

Model Complexity

- Models can be made more or less "complex," e.g.,
  - Number of variables in model (e.g., stepwise selection)
  - Flexibility, e.g., degree of polynomial terms in linear regression, number of hidden nodes in a neural network, number of leaves on a tree, number of bins in bin smoother
  - Use penalized least squares, e.g., ridge regression and lasso, smoothing splines, weight decay for neural networks
- The modeler must select the appropriate complexity so that the model does not capture idiosyncrasies of the particular data set (called *overfitting* the data)



Out-of-Sample Estimates of Prediction Error

- *Test sets*
  - Take the available data, draw a sample of observations, and set them in a "safe" while you build the model (called the *holdout* or *test* sample).
  - Use the remaining data, called the *estimation* or *training* sample, to build your model. The training set should be large enough to make reliable estimates, but overly large, which would unnecessarily waste computing resources.
  - Apply estimated model to the test sample and evaluate accuracy.
- K-Fold Cross validation
  - Step 1: Split available data into K roughly equal-sized parts
  - Step 2: For part k, fit the model on the other $K-1$ parts, apply the estimated model to part k, and evaluate fit
  - Step 3: Repeat step 2 for $k = 1, \ldots, K$ and combine the K evaluations of fit
  - Most data mining books suggest $K = 5$ or 10
  - This is computationally more expensive than training/test splits, but makes more efficient use of the data — **use when available sample size is small**

Out-of-Sample Estimates of Prediction Error: Fresh Data

"The second level of cross-validation, which, by analogy with the physician's "double-blind" study, we have called "double cross-validation", is to be had only by going to fresh data. These fresh data are best gathered after choosing form and coefficients. When fresh gathering is not feasible, good results can come from going to a body of data that has been kept in a locked safe where it has rested untouched and unscanned during all the choices and optimizations . For the full validating effect, the data placed in the safe must differ from those used to choose the procedure in ways that adequately represent the sources of variation anticipated in practice. For example, they may need to involve distinct school systems, distinct investigators, or distinct years of observations. (from Mosteller and Tukey (1977), *Data Analysis and Regression*, p. 38.)"

Test Sets in R

```
> set.seed(12345)
> employee$train = runif(nrow(employee))>.5  # assign to test/train set
> employee$salaryK = employee$salary/1000    # salary in $K

> dim(employee)  # note: 71 rows
[1] 71  8
> table(employee$train)
FALSE  TRUE
   36    35

> fit = lm(salaryK ~ ageyrs + expyrs, employee, subset=train)
> anova(fit)      # note 35 rows used to fit model
Analysis of Variance Table

Response: salaryK
          Df  Sum Sq Mean Sq F value  Pr(>F)
ageyrs     1  473.05  473.05  5.8292 0.02166 *
expyrs     1  518.18  518.18  6.3855 0.01665 *
Residuals 32 2596.82   81.15

> sum(fit$residuals^2)  # compute training RSS from fitted object
[1] 2596.816
> deviance(fit)         # Or just use deviance function
[1] 2596.816
> mean(fit$residuals^2)    # training MSE with n in denominator
[1] 74.19475

> yhat = predict(fit, employee[!employee$train,])  # apply model to test set
> length(yhat)   # note 36 predictions
[1] 36
> mean((employee$salaryK[!employee$train] - yhat)^2)    # test MSE
[1] 80.52033
```

K-Fold Cross-Validation in R

```
> set.seed(12345)
> employee$cv = as.integer(runif(nrow(employee))*5)
> table(employee$cv)
 0  1  2  3  4
13 15 10 19 14

> yhat = rep(NA, nrow(employee))   # set up vector for held-out predictions
> for(i in 0:4){
+ fit = lm(salaryK~ageyrs+expyrs, employee, subset=(cv!=i))
+ yhat[employee$cv==i] = predict(fit, employee[employee$cv==i,])
+ }
> mean((employee$salaryK-yhat)^2)  # test MSE
[1] 83.52445

> fit = lm(salaryK~ageyrs+expyrs, employee)
> mean(fit$residuals^2)  # compare with training MSE
[1] 76.01949
```

Choosing a Set of Good Predictors

- Ideally you will have domain knowledge (e.g., theory) to tell you what predictors to use.
- Which predictors should we use if we don't have a strong theory? (This is often the case with models where prediction is primary objective.)
- Model fit (RSS and deviance) — Adding predictors will ...
  - always improve RSS on *estimation* sample
  - not necessarily improve RSS on *validation* data — RSS will increase if we model idiosyncrasies of estimation sample
- Reasonable solutions:
  - **Selection**: Iterative model selection procedures, e.g., forward, backward, stepwise, lasso
  - **Shrinkage/Regularization**: Shrinkage estimation, e.g., ridge, lasso or dimensionality reduction models (e.g., PCR, PLS, EFA, CFA)

Iterative Model Selection

Suppose we have a large number of predictor variables and we want to build a parsimonious model *with good predictive accuracy*. Using these methods is questionable when interpretation is the goal. Three commonly used approaches are:

- *Forward selection*
  1. Begin with no variables
  2. Add variable that yields greatest significant improvement in RSS
  3. Repeat (2) until no significant improvement in RSS
- *Backward elimination*
  1. Begin with all candidate variables
  2. Drop variable variable that causes smallest non-significant increase in RSS
  3. Repeat (2) until dropping variable causes significant increase in RSS
- *Stepwise selection*
  1. (Usually) begin with no variables
  2. Drop variables that causes smallest *non-significant* increase in SSE
  3. Add variable that yields greatest significant improvement in RSS
  4. Repeat (2) and (3) until no improvement in RSS

Stepwise Regression: Click Ball Point Pens

```
> click$fair = as.numeric(click$eff==2)
> click$good = as.numeric(click$eff==3)
> click$outstand = as.numeric(click$eff==4)
> fit = lm(sales~1, click)
> fit2 = step(fit, scope=~ad+reps+fair+good+outstand, test="F")
Start:  AIC=386.52
sales ~ 1

           Df Sum of Sq    RSS    AIC  F value     Pr(F)
+ reps      1    465161 133092 328.40 132.8114 5.739e-14 ***
+ ad        1    463451 134802 328.91 130.6445 7.327e-14 ***
<none>                  598253 386.52
+ good      1     24847 573406 386.82   1.6466    0.2072
+ fair      1      9076 589177 387.90   0.5854    0.4489
+ outstand  1      6008 592245 388.11   0.3855    0.5384

Step:  AIC=328.4
sales ~ reps

           Df Sum of Sq    RSS    AIC  F value     Pr(F)
+ ad        1     57617  75475 307.71  28.2458 5.317e-06 ***
<none>                  133092 328.40
+ outstand  1      6008 127084 328.55   1.7492    0.1941
+ fair      1      2160 130932 329.74   0.6104    0.4396
+ good      1      2086 131006 329.76   0.5891    0.4476
- reps      1    465161 598253 386.52 132.8114 5.739e-14 ***

Step:  AIC=307.71
sales ~ reps + ad

           Df Sum of Sq    RSS    AIC F value     Pr(F)
<none>                  75475 307.71
+ outstand  1      3273  72202 307.93  1.6317    0.2096
+ fair      1      1289  74185 309.02  0.6257    0.4341
+ good      1         5  75470 309.70  0.0022    0.9626
- ad        1     57617 133092 328.40 28.2458 5.317e-06 ***
- reps      1     59327 134802 328.91 29.0842 4.167e-06 ***
```

Backward Selection: Click Ball Point Pens

```
> fit = lm(sales ~ ad+reps+fair+good+outstand, click)
> step(fit)
Start:  AIC=311.27
sales ~ ad + reps + fair + good + outstand

            Df Sum of Sq    RSS    AIC
- fair       1       229  71247 309.40
- good       1       998  72016 309.83
- outstand   1      2857  73875 310.85
<none>                    71018 311.27
- ad         1     41227 112245 327.58
- reps       1     54607 125625 332.09

Step:  AIC=309.4
sales ~ ad + reps + good + outstand

            Df Sum of Sq    RSS    AIC
- good       1       955  72202 307.93
<none>                    71247 309.40
- outstand   1      4223  75470 309.70
- ad         1     47039 118285 327.68
- reps       1     56521 127768 330.76

Step:  AIC=307.93
sales ~ ad + reps + outstand

            Df Sum of Sq    RSS    AIC
- outstand   1      3273  75475 307.71
<none>                    72202 307.93
- ad         1     54882 127084 328.55
- reps       1     60928 133129 330.41

Step:  AIC=307.71
sales ~ ad + reps

       Df Sum of Sq    RSS    AIC
<none>               75475 307.71
- ad    1     57617 133092 328.40
- reps  1     59327 134802 328.91

Call:
lm(formula = sales ~ ad + reps, data = click)

Coefficients:
(Intercept)           ad         reps
      69.33        14.16        37.53
```

Stepwise: quality control

```
> fit = lm(defect~., quality)
> step(fit)
Start:  AIC=123.01
defect ~ temp + density + rate + am

          Df Sum of Sq    RSS    AIC
- am       1    14.808 1312.1 121.35
- rate     1    19.847 1317.2 121.46
<none>                 1297.3 123.00
- density  1    90.862 1388.2 123.04
- temp     1   117.868 1415.2 123.61

Step:  AIC=121.35
defect ~ temp + density + rate

          Df Sum of Sq    RSS    AIC
- rate     1    40.591 1352.7 120.26
- density  1    76.366 1388.5 121.04
<none>                 1312.1 121.35
- temp     1   188.919 1501.0 123.38

Step:  AIC=120.26
defect ~ temp + density

          Df Sum of Sq    RSS    AIC
<none>                 1352.7 120.26
- density  1    142.69 1495.4 121.27
- temp     1    258.24 1611.0 123.50

Call:
lm(formula = defect ~ temp + density, data = quality)

Coefficients:
(Intercept)         temp      density
     46.256       18.049       -2.329
```

Logistic Backward selection

```
> set.seed(12345)
> defaultsmall = default[sample(1:nrow(default), 500),]  # sample 500 cases
> fit = glm(default~log(downpmt+1)+pmttype+age+gender, binomial, defaultsmall)
> step(fit)
Start:  AIC=277.52
default ~ log(downpmt + 1) + pmttype + age + gender

                   Df Deviance    AIC
- gender            1   263.57 275.58
- age               1   264.75 276.75
<none>                 263.52 277.52
- log(downpmt + 1)  1   291.07 303.07
- pmttype           3   311.35 319.35

Step:  AIC=275.58
default ~ log(downpmt + 1) + pmttype + age

                   Df Deviance    AIC
- age               1   264.85 274.85
<none>                 263.57 275.58
- log(downpmt + 1)  1   291.14 301.14
- pmttype           3   311.42 317.42

Step:  AIC=274.85
default ~ log(downpmt + 1) + pmttype

                   Df Deviance    AIC
<none>                 264.85 274.85
- log(downpmt + 1)  1   292.28 300.28
- pmttype           3   313.22 317.22

Coefficients:
      (Intercept)   log(downpmt + 1)   pmttypeStatement   pmttypeCheck EFT
           0.9093            -0.4900            -1.1228            -3.0655
pmttypeCredit EFT
          -2.7766

Degrees of Freedom: 499 Total (i.e. Null);  495 Residual
Null Deviance:     338.1
Residual Deviance: 264.9  AIC: 274.9
```

Stepwise selection

```
> fit = glm(default~1, binomial, defaultsmall)
> step(fit, scope=~log(downpmt+1)+pmttype+age+gender)
Start:  AIC=340.07
default ~ 1

                   Df Deviance    AIC
+ pmttype           3   292.28 300.28
+ log(downpmt + 1)  1   313.22 317.22
<none>                  338.07 340.07
+ age               1   336.33 340.33
+ gender            1   338.02 342.02

Step:  AIC=300.28
default ~ pmttype

                   Df Deviance    AIC
+ log(downpmt + 1)  1   264.85 274.85
<none>                  292.28 300.28
+ age               1   291.14 301.14
+ gender            1   292.13 302.13
- pmttype           3   338.07 340.07

Step:  AIC=274.85
default ~ pmttype + log(downpmt + 1)

                   Df Deviance    AIC
<none>                  264.85 274.85
+ age               1   263.57 275.58
+ gender            1   264.75 276.75
- log(downpmt + 1)  1   292.28 300.28
- pmttype           3   313.22 317.22

Coefficients:
      (Intercept)   pmttypeStatement   pmttypeCheck EFT  pmttypeCredit EFT
           0.9093            -1.1228            -3.0655            -2.7766
 log(downpmt + 1)
          -0.4900

Degrees of Freedom: 499 Total (i.e. Null);  495 Residual
Null Deviance:     338.1
Residual Deviance: 264.9  AIC: 274.9
```

Shrinkage Estimation

- Consider estimate $\mathbf{b}$ of $\boldsymbol{\beta}$
- The *bias* of $\mathbf{b}$ is

$$\text{bias}(\mathbf{b}) = E(\mathbf{b}) - \boldsymbol{\beta}$$

When $E(\mathbf{b}) = \boldsymbol{\beta}$, we say that $\mathbf{b}$ is an *unbiased* estimate of $\boldsymbol{\beta}$.

- The *mean-squared error* of $\mathbf{b}$ is

$$\begin{aligned} \text{MSE}(\mathbf{b}) &= E[(\mathbf{b} - \boldsymbol{\beta})^T(\mathbf{b} - \boldsymbol{\beta})] \\ &= \text{trace}(V(\mathbf{b})) + \text{bias}(\mathbf{b})^T\text{bias}(\mathbf{b}) \\ &= \text{variance} + \text{bias}^2 \end{aligned}$$

- The OLS estimate $\hat{\boldsymbol{\beta}}$ is unbiased and therefore has

$$\text{MSE}(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$$

The Gauss-Markov Theorem tells us that the OLS estimates are BLUE, and thus have the smallest MSE among unbiased estimates.

- Strategy of shrinkage estimation: introduce a bias that reduces the variance to give an estimate with lower overall mean squared error

Ridge Regression

- An alternative way to reduce the impact of any single predictor variable is to include a penalty term in the least-squares objective function

$$\hat{\boldsymbol{\beta}}_\lambda = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left[\sum_{i=1}^{k} (y_i - \beta_0 - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right] \tag{3.41}$$

$$\hat{\boldsymbol{\beta}}_\lambda = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \tag{3.44}$$

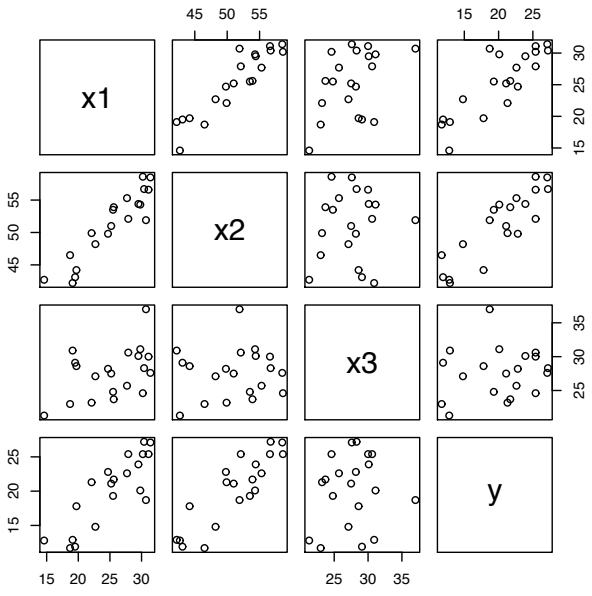
- $\lambda \geq 0$ is a constant, which determines how much to penalize large regression coefficients — when $\lambda = 0$ we get OLS and when λ is big the penalty is great and the coefficients will be close to 0
- Ridge existence theorem: there exist values of λ so that $\hat{\boldsymbol{\beta}}_\lambda$ has smaller mean squared error than OLS estimates of $\boldsymbol{\beta}$
- Simulations have shown that ridge regression produces $\hat{y}$ values that are closer to the true values than PCR and stepwise regression (See Frank and Friedman, 1993). Likewise for direct marketing gains charts (Malthouse, 1999)
- Scaling of X variables matters—standardize when units are incommensurate
- Criticisms of ridge regression:
  - The optimal value of λ depends on β and σ^2 (error variance), which are being estimated by the regression
  - No inference available
  - Lack of theoretical justification for particular penalty term—why unweighted sum of squares? (Ridge regression has Bayesian interpretation.)

Body Fat Example: Ridge Regression in R

Consider Body Fat example from Kutner (Table 7.1), `y` = body fat, `x1` = triceps skinfold thickness, `x2` = thigh circumference, and `x3` = midarm circumference.

```
bodyfat = data.frame(
  x1=c(19.5,24.7,30.7,29.8,19.1,25.6,31.4,27.9,22.1,25.5,31.1,30.4,
    18.7,19.7,14.6,29.5,27.7,30.2,22.7,25.2),
  x2=c(43.1,49.8,51.9,54.3,42.2,53.9,58.5,52.1,49.9,53.5,56.6,56.7,
    46.5,44.2,42.7,54.4,55.3,58.6,48.2,51.0),
  x3=c(29.1,28.2,37.0,31.1,30.9,23.7,27.6,30.6,23.2,24.8,30.0,28.3,
    23.0,28.6,21.3,30.1,25.7,24.6,27.1,27.5),
  y=c(11.9,22.8,18.7,20.1,12.9,21.7,27.1,25.4,21.3,19.3,25.4,27.2,
    11.7,17.8,12.8,23.9,22.6,25.4,14.8,21.1)
)
```

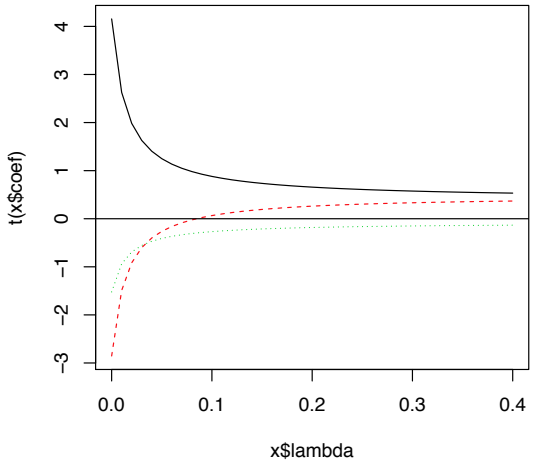
```
> plot(bodyfat)
> round(cor(bodyfat), 2)
     x1   x2   x3    y
x1 1.00 0.92 0.46 0.84
x2 0.92 1.00 0.08 0.88
x3 0.46 0.08 1.00 0.14
y  0.84 0.88 0.14 1.00

> fit = lm(y~x1+x2+x3, bodyfat)
> coef(fit)   # sign flips on x2, x3
> round(coef(fit),2)
(Intercept)      x1      x2      x3
     117.08    4.33   -2.86   -2.19

> vif(fit)  # matches table 11.3
      x1       x2       x3
708.8429 564.3434 104.6060
```



Body Fat Example

```
> install.packages("MASS")  # do this only once
> library(MASS)  # do this at start of session
> Zbodyfat = data.frame(scale(bodyfat))   # standardize data
> cor(Zbodyfat)     # matches correlation matrix

> fit2 = lm.ridge(y~x1+x2+x3-1, Zbodyfat, lambda=seq(0,0.4,length=41))
```

```
> # make ridge trace, Kutner Fig 11.3
> plot(fit2); abline(h=0)

> # matches Kutner Tab 11.2 with lambda=20c
> round(coef(fit2), 4)
         x1      x2      x3
0.00 4.2637 -2.9287 -1.5614
0.01 2.6950 -1.5295 -0.9613
0.02 2.0348 -0.9408 -0.7087
0.03 1.6710 -0.6166 -0.5695
0.04 1.4407 -0.4113 -0.4813
...
```

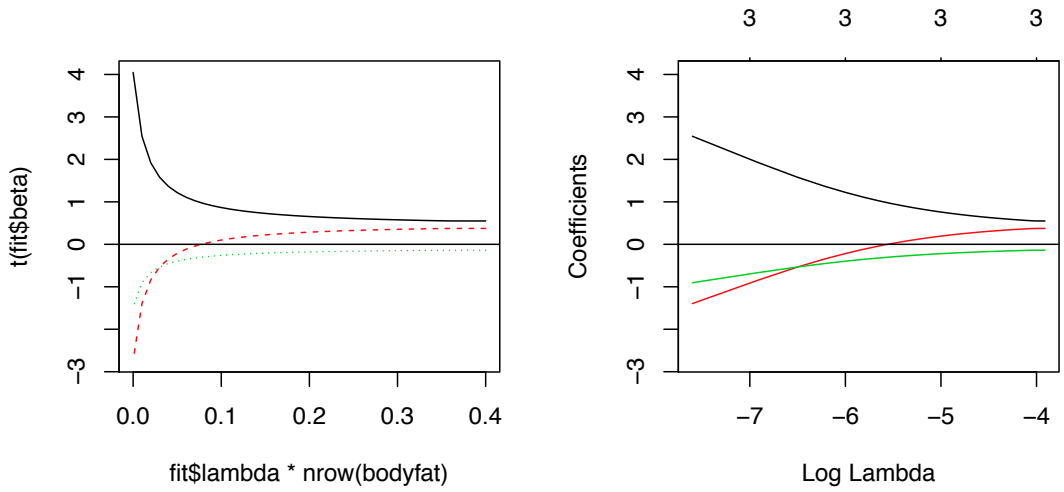


Old approach: pick λ when estimates "stabilize," e.g., $\lambda = .1$ or .2.

Body Fat Ridge Example in `glmnet`

- The `glmnet` package scales λ differently—divide by n to match `lm.ridge`.
- Set parameter `alpha=0` for ridge regression

```
lam = seq(0,.4,length=41)/nrow(Zbodyfat)
fit = glmnet(as.matrix(Zbodyfat[,1:3]), Zbodyfat$y, alpha=0, lambda=lam)
# This matches the plot from lm.ridge
matplot(fit$lambda*nrow(bodyfat), t(fit$beta), type="l"); abline(h=0)
plot(fit, xvar="lambda")   # plots log(lambda) instead
abline(h=0)
```

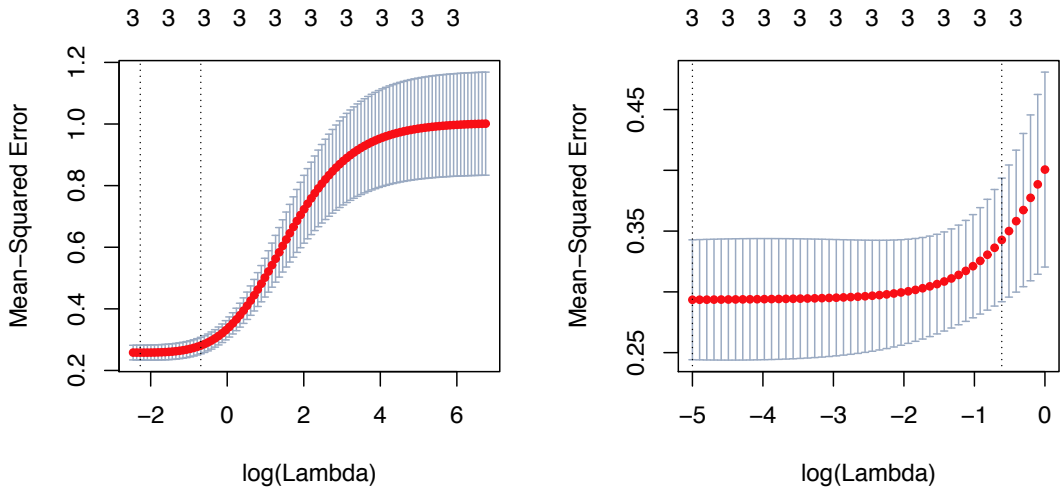


In practice the difference in λ scaling is not important as long as you are only using one package.

Body Fat Example With Cross Validation

- Picking λ when it "stabilizes" is subjective. Another approach is to use K-fold cross validation, which is implemented in R.
- Specifying λ values is optional.
- `nfolds=10` is the default for K

```
> fit = cv.glmnet(as.matrix(Zbodyfat[,1:3]), Zbodyfat$y, alpha=0, nfolds=5)
> fit$lambda.min
[1] 0.08558559
> plot(fit)
> fit2 = cv.glmnet(as.matrix(Zbodyfat[,1:3]), Zbodyfat$y, alpha=0, nfolds=5,
    lambda=exp(seq(-5,-0,length=50)))
> plot(fit2)
> fit2$lambda.min
[1] 0.006737947
```



Very little shrinkage is necessary. Having determined the appropriate level for λ, refit using all the data.

```
dat = read.csv("teach/304/dmef3/train.csv")
dat$custno = NULL
for(i in c(10:16,22)) dat[[i]][dat[[i]]<0] = 0   # set neg to 0
for(i in c(2:16, 21,22)) dat[[i]] = log(dat[[i]]+1)
dat$buy = as.numeric(dat$targamnt>0)
dat$targamnt[dat$targamnt<0] = 0
dat$targamnt = log(dat$targamnt + 1)

set.seed(12345)
train = runif(nrow(dat))<.2    # pick train/test split

# full model, targamnt
fit = lm(targamnt ~ ., dat[,-24], subset=train)
yhat = predict(fit, dat[!train,])
mean((dat$targamnt[!train] - yhat)^2)

fit2 = step(fit)
yhat = predict(fit2, dat[!train,])
mean((dat$targamnt[!train] - yhat)^2)

x = model.matrix(targamnt ~ ., dat[,-24])  # -24 drops buy variable
fit.ridge = glmnet(x[train,], dat$targamnt[train], alpha=0)
fit.cv = cv.glmnet(x[train,], dat$targamnt[train], alpha=0)
yhat = predict(fit.ridge, s=fit.cv$lambda.min, newx=x[!train,])
mean((dat$targamnt[!train] - yhat)^2)

fit.lasso = glmnet(x[train,], dat$targamnt[train], alpha=1)
fit.cv = cv.glmnet(x[train,], dat$targamnt[train], alpha=1)
yhat = predict(fit.lasso, s=fit.cv$lambda.min, newx=x[!train,])
mean((dat$targamnt[!train] - yhat)^2)

# full model for buy
fit = glm(buy ~ ., binomial, dat[,-23], subset=train)
phat = predict(fit, dat[!train,-23], type="resp")
roc(dat$buy[!train], phat)

# stepwise for buy
fit2 = step(fit)
phat = predict(fit2, dat[!train,-23], type="resp")
roc(dat$buy[!train], phat)

x = model.matrix(buy ~ ., dat[,-23])   # -23 drops targamnt
fit.ridge = glmnet(x[train,], dat$buy[train], family="binomial", alpha=0)
fit.cv = cv.glmnet(x[train,], dat$buy[train], family="binomial", alpha=0)
phat = predict(fit.ridge, s=fit.cv$lambda.min, newx=x[!train,])
roc(dat$buy[!train], phat)

fit.lasso = glmnet(x[train,], dat$buy[train], family="binomial", alpha=1)
fit.cv = cv.glmnet(x[train,], dat$buy[train], family="binomial", alpha=1)
phat = predict(fit.lasso, s=fit.cv$lambda.min, newx=x[!train,])
roc(dat$buy[!train], phat)
```

The Lasso

- Ridge regression penalizes $\sum_j \beta_j^2$, the squared Euclidean length of the slope vector (ℓ_2 norm)
- The *lasso* penalizes $\sum_j |\beta_j|$, the taxi-cab length of the slope vector (ℓ_1 norm)

$$\hat{\boldsymbol{\beta}}_\lambda = \underset{\boldsymbol{\beta}}{\text{argmin}} \left[\sum_{i=1}^{k} (y_i - \beta_0 - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right]$$

- The lasso and ridge regression shrink coefficients towards zero, but the lasso tends to force some coefficients to equal zero, similar to variable subect selection.
- We can think of the methods as having different constraints:
  - Subset selection: $\min_{\boldsymbol{\beta}} \text{RSS}$ subject to $\sum_j I(\beta_j \neq 0) \leq s$
  - Ridge: $\min_{\boldsymbol{\beta}} \text{RSS}$ subject to $\sum_j \beta_j^2 \leq s$
  - Lasso: $\min_{\boldsymbol{\beta}} \text{RSS}$ subject to $\sum_j |\beta_j| \leq s$
- Use glmnet library in R

IEMS 304: Homework 5
Professor Malthouse

1. Let $X_1, \ldots, X_n$ be independent random variables from a normal distribution with mean μ and variance σ^2. In class we discussed two estimators of σ^2:

$$S^2 = \frac{1}{n-1} \underbrace{\sum_{i=1}^{n} (x_i - \bar{x})^2}_{\text{TSS}} = \frac{\text{TSS}}{n-1}. \quad \text{and} \quad \hat{\sigma}^2 = \frac{\text{TSS}}{n}.$$

   We showed that $E(\text{TSS}) = (n-1)\sigma^2$ and $V(\text{TSS}) = 2(n-1)\sigma^4$. Now consider a third estimate, $\tilde{\sigma}^2 = \text{TSS}/(n+1)$.

   (a) Find the expected value and bias of $\tilde{\sigma}^2$.
   (b) Find the variance of $\tilde{\sigma}^2$.
   (c) Find the MSE of $\tilde{\sigma}^2$.
   (d) Show that $\text{MSE}(\tilde{\sigma}^2) < \text{MSE}(\hat{\sigma}^2)$.

2. This problem is a variation of 6.9 in JWHT on page 263. It uses the College data. If you need help see the lab exercises in section 6.6.

   (a) Read the data into R, fix the row names, and create a test set as follows:

```
college = read.csv("teach/304/data/College.csv")
row.names(college) = college$X
college$X=NULL
set.seed(12345)
train = runif(nrow(college))<.5    # pick train/test split
```

   How many cases are assigned to training? Test?

   (b) The dependent variable is `Apps`. Generate a histrogram. Comment.

   (c) Regress `Apps` on all other variables using only the training data. Examine the residual plot and comment.

   (d) Replace `Apps` with its square root, i.e.,

```
college$Apps = sqrt(college$Apps)
```

   (e) *Full model.* Do problem 6.9b. Note that you are using the square root of apps as the dependent variable and in your evalution on the test set. For test set error, report the mean, i.e.,

```
fit = lm(Apps ~ ., college, subset=train)
yhat = predict(fit, college[!train,])
mean((college$Apps[!train] - yhat)^2)
```

   (f) *Step model.* Apply the `step` function to the fitted "full" model from the previous part and report the test set error.

   (g) Do problem 6.9c. Plot the ridge trade, i.e., `plot(fit, xvar="lambda")`, and the fitted `cv.glmnet` object showing cross-validated MSE against λ. What is the optimal value of λ?

   (h) Do problem 6.9d. Report the same things as with ridge.

(i) Examine a scatterplot matrix for the training data. Suggest suitable transformations for predictor variables as necessary (continue to use the square root of apps as the dependent variable). Add these transformations to your model and see if the test-set MSE improves. Report which transformations end up improving your model, your preferred method of estimation (e.g., OLS, step, ridge, lasso), and the final test set error.

Introduction to Smoothing

- Problem: summarize the dependence of one numerical variable (Y) on $p = 1$ predictor variable (x)
- Possible solutions:
  - Scatterplot: great if you have at most hundreds of observations, but not so good if you have big data sets
  - Regress one variable on the other (assumes linear relationship)
- Smoothers summarize nonlinear relationships
  - If data set is small, superimpose the smoother on the scatterplot
  - If data set is big, plot smoother alone
- When do I use smoothers?
  - EDA: helps you identify nonlinear relationships (so that you can add appropriate transformations)
  - Presentations
  - Building block for additive models and other methods

Summary of Smoothing

- Smoothers model the relationship between x and y (Average$(Y|x)$) without assuming a rigid functional form. Recall that linear regression assumes

$$\text{Average}(Y|x) = \alpha + x\beta$$

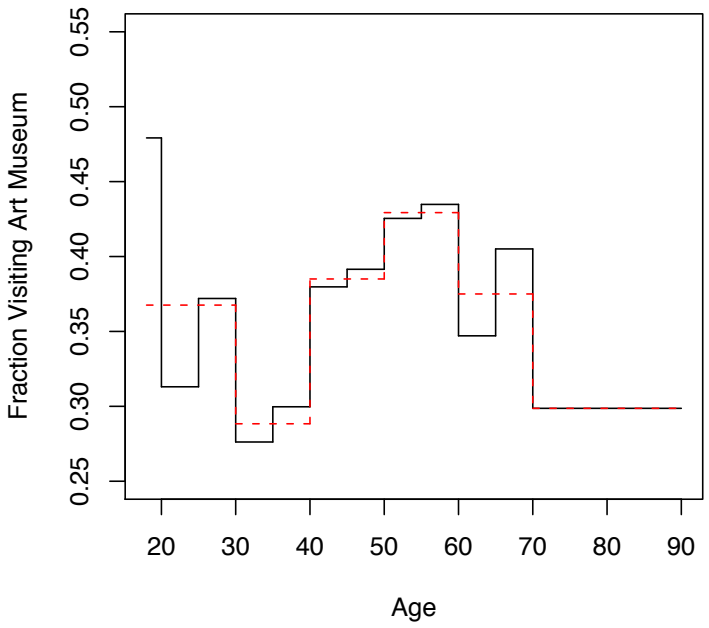
- The smoothness of the fit is moderated by a *smoothing parameter*
- At one extreme (very smooth) we have a flat line (the intercept model using 1 degree of freedom) — this is usually too smooth because it does not model any relationship between x and y
- The linear model (2 degrees of freedom) is more flexible than the intercept model, but often we need a more flexible model
- At the other extreme (very flexible), the fit goes through every point (assuming unique x's) — this is usually too flexible because it *overfits* the data by following every wiggle
- The trick is to use judgment to pick a value for the smoothing parameter that captures the relationship between x and y without overfitting

Bins, Running Means, and Kernels

- We observe $(x_1, y_1), \ldots, (x_n, y_n)$ $(a = x_1 < \ldots < x_n = b)$ where
$$y_i = f(x_i) + e_i,$$
where f is continuous, $E(e_i) = 0$, and $V(e_i) = \sigma^2$
- *Bin smoothers* (Regressograms)
  1. Partition the interval $[a, b]$ into k bins
  2. The bin smoother estimate of y_i in bin j is the average of all the y values with corresponding x value in bin j
- *Running means*
  1. Let $N_k(x_i)$ be a neighborhood around x_i, e.g., on each side of x_i, the k closest points ("nearest neighbors")
  2. The running mean estimate of y_i is the average of all y values with corresponding x values in $N_k(x_i)$
- *Kernel smoothers* — the Kernel smoother estimate of y_i is a weighted average of the y values, where points closer to x_i receive more weight than those further away from x_i. The *kernel function* specified how the weights are assigned, e.g., normal curve (Gaussian) or "Epanechnikov" parabola.

Art Museum Problem

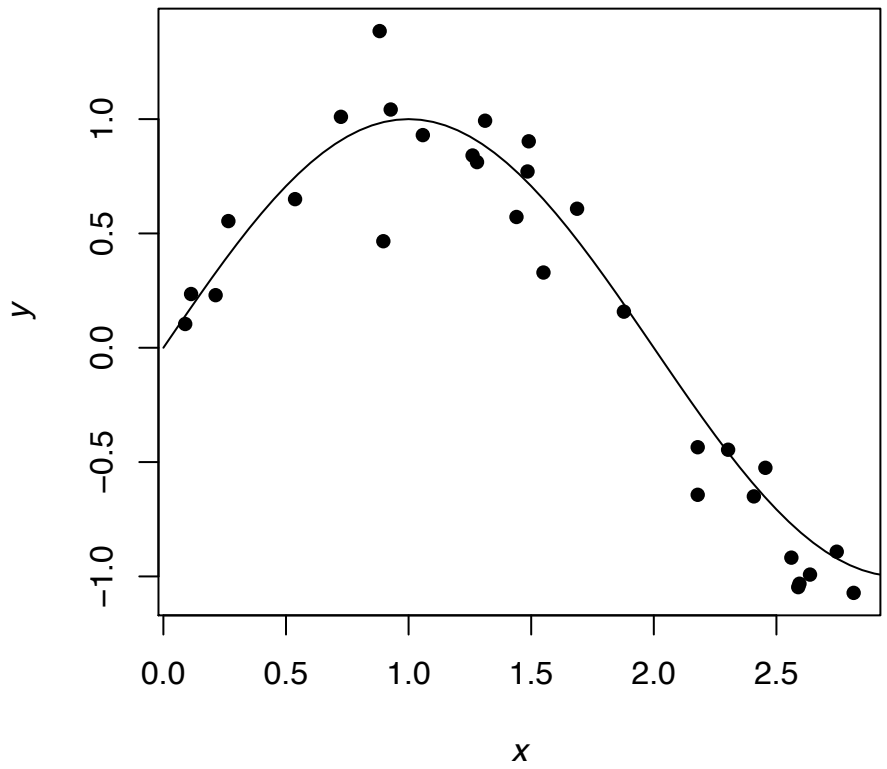
	$K = 12$ Bins (black line)					$K = 6$ Bins (red line)			
Response	Age Range	n	Number Visitors	$\overline{y}$	se($\overline{y}$)	Age Range	n_k	$\overline{y}$	se($\overline{y}$)
1	18–19	48	23	0.479	0.0721	18–29	370	0.368	0.0251
2	20–24	115	36	0.313	0.0432				
3	25–29	207	77	0.372	0.0336				
4	30–34	315	87	0.276	0.0252	30–39	652	0.288	0.0177
5	35–39	337	101	0.300	0.0250				
6	40–44	316	120	0.380	0.0273	40–49	574	0.385	0.0203
7	45–49	258	101	0.391	0.0304				
8	50–54	228	97	0.425	0.0327	50–59	389	0.429	0.0251
9	55–59	161	70	0.435	0.0391				
10	60–64	170	59	0.347	0.0365	60–69	328	0.375	0.0267
11	65–69	158	64	0.405	0.0391				
12	70+	298	89	0.299	0.0265	70+	298	0.299	0.0265



Wave Example

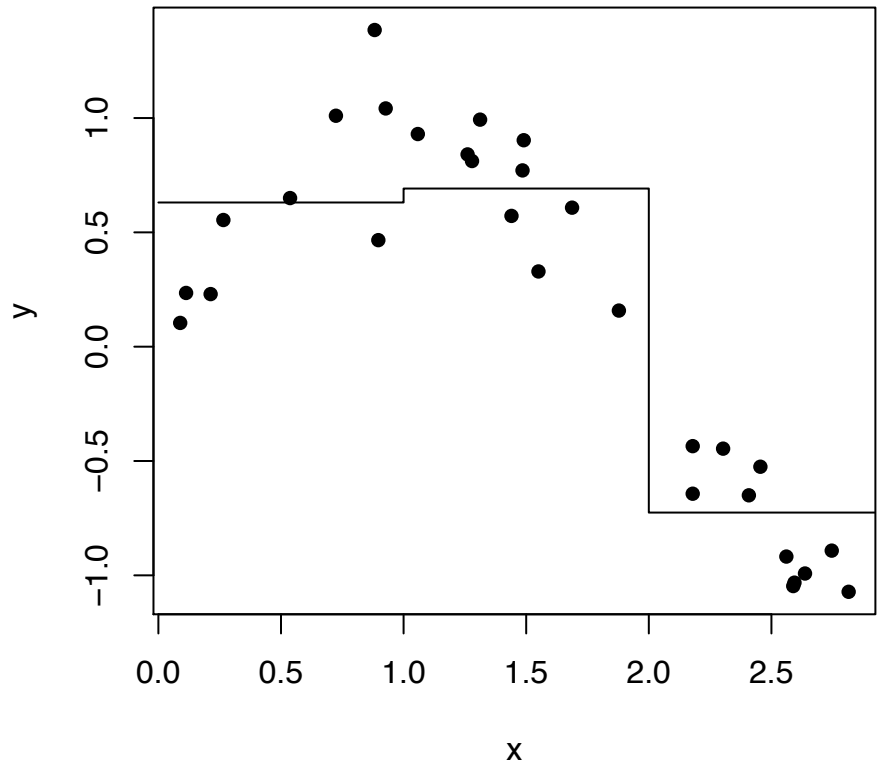
$$y = \sin(x\pi/2) + e, \quad e \sim \mathcal{N}(0, 1/16 = 0.0625)$$

```
set.seed(1234567)
n=30
x = runif(n)*3
y = sin(x*pi/2) + rnorm(n)/4
dat = data.frame(x=round(x,3), y=round(y,3))

xx = seq(0,3,.05)
plot(dat$x, dat$y, pch=16, xlab="x", ylab="y")
lines(xx, sin(xx*pi/2))
```



Bin Smoother with Three Bins

```
dat$bin = cut(dat$x, 0:3)
fit = lm(y~bin, dat)
summary(fit)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.63067    0.10754   5.864 3.03e-06 ***
bin(1,2]     0.06103    0.14824   0.412    0.684
bin(2,3]    -1.41721    0.14501  -9.773 2.32e-10 ***
```



How many bins?

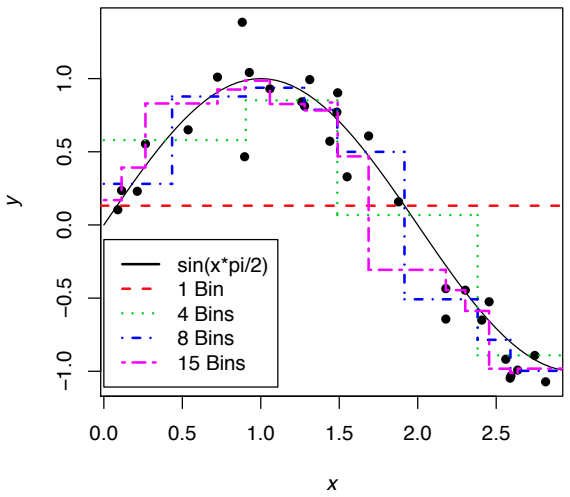
- Create a test set of 10,000

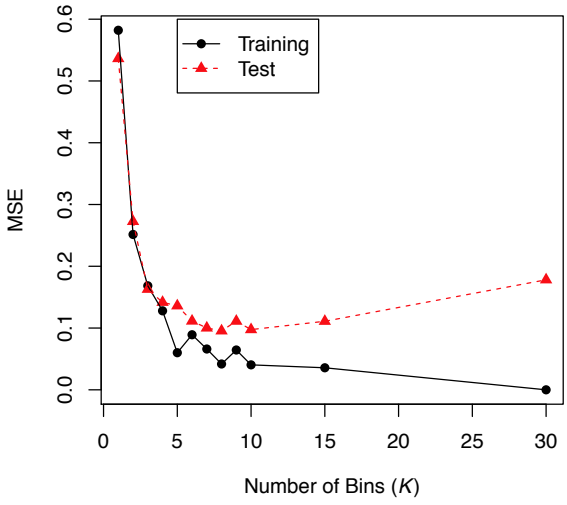
```
test = data.frame(x=runif(10000)*3)
test$y = sin(test$x*pi/2) + rnorm(10000)/4
```

- Fit models with different numbers of bins, e.g.,

```
dobin = function(dat, k, b){
  dat$bin = cut(dat$x, b);
  test$bin = cut(test$x, b);
  fit = lm(y ~ bin, dat);
  c(k, mean((fit$residuals)^2),
    mean((test$y-predict(fit, test))^2));
}

dobin(wave, 5, seq(0,3,length=5+1))
```

Number of Bins	Training MSE	Test MSE
1	0.5339	0.5163
2	0.2995	0.3159
3	0.1328	0.1780
4	0.1471	0.1238
5	0.0765	0.1155
6	0.0730	0.1205
7	0.0846	0.1024
8	0.0961	0.1067
9	0.0712	0.1031
10	0.0608	0.0902
15	0.0390	0.1196
30	0.0000	0.1781





Piecewise-Linear Functions

- Also called *hinge*, *hockey-stick*, or *elbow* functions
- We want to approximately continuous function f over $[a, b]$ with a line in each bin such that the lines match up at the knots.
- Let $c_1 = 0$ be the only knot. Consider the following function:

$$\hat{f}(x) = \beta_0 + \beta_1 x + \beta_2 (x)_+,$$

  where $(x)_+ = x$ for $x > 0$ and 0 otherwise.

  – For $x \leq 0$ the term $(x)_+ = 0$ and the estimated function is

$$\hat{f}_1(x) = \beta_0 + \beta_1 x$$

  – For $x \geq 0$, $(x)_+ = x$ and the estimated function is

$$\hat{f}_2(x) = \beta_0 + \beta_1 x + \beta_2 x = \beta_0 + (\beta_1 + \beta_2) x.$$

  Note that (1) $\hat{f}_1(0) = \hat{f}_2(0) = \beta_0$, so it's continuous, and (2) β_2 measures the change in slope at the knot ($\beta_2 = 0$ implies no elbow).

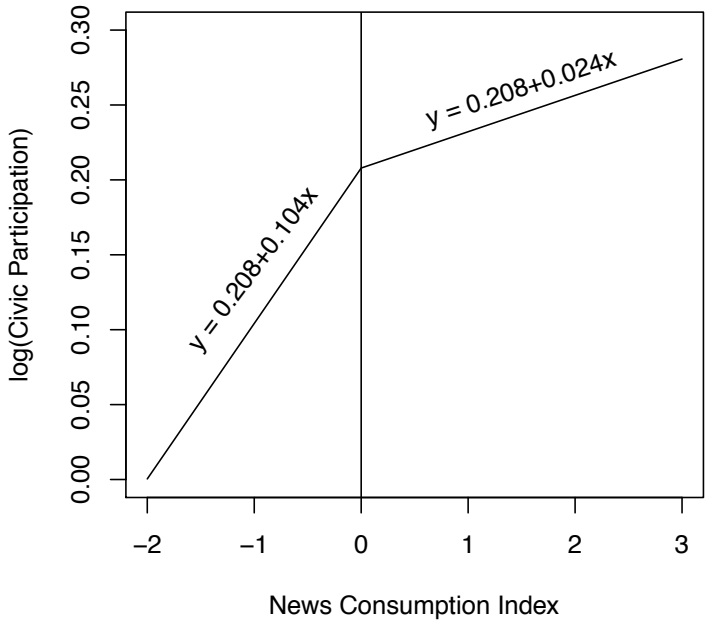
- The general **piecewise-linear function** with $K - 1$ predefined knots $c_1 < \cdots < c_{K-1}$ is:

$$\hat{f}(x) = \beta_0 + \beta_1 x + \sum_{k=1}^{K-1} \beta_{k+1} (x - c_k)_+.$$

Civic participation and news consumption

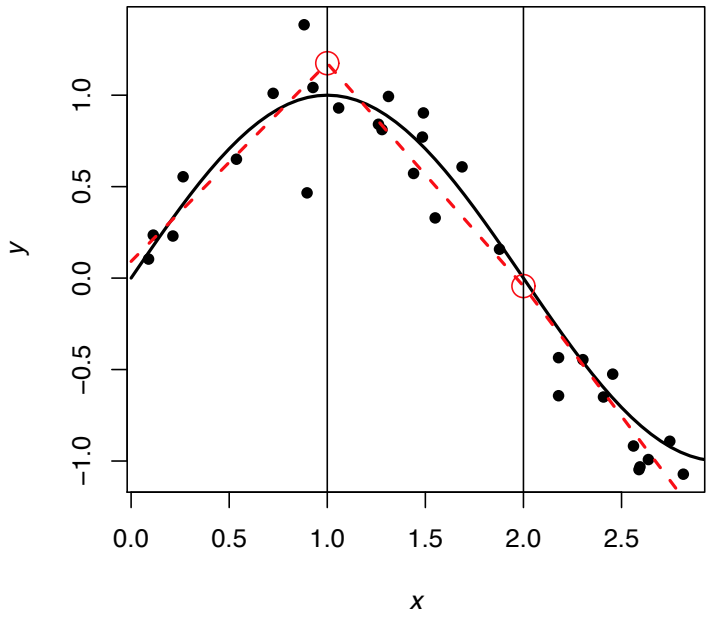
What is the relationship between civic participation and Total News Consumption and does the relationship differ for those who seek out or avoid the news? A random sample adults was surveyed about their civic participation (`CP`), news consumption across various media. Based on the responses to the news consumption, a numerical index of consumption was formed (`index`), where small values indicated low aggregate consumption and large values indicate high consumption of news. This index is standardized so that its mean is 0 and variance is 1. The research question asks whether the effect of the index on CP is different for those who seek out the news than those who avoid it. People with negative values of `index` (i.e., below average) were classified as news *avoiders* and those above average as news *seekers*.

Term	Slope	Std Err	t Value
Intercept	0.208	0.0029	71.34
Index	0.104	0.0042	24.54
$(\text{Index})_+$	-0.079	0.00062	-12.91



Piecewise-Linear Fit to Wave

```
plot(dat$x, dat$y, pch=16, xlab="x", ylab="y")
lines(xx, sin(xx*pi/2), lwd=2)
fit = lm(y ~ x + I((x-1)*(x>1)) + I((x-2)*(x>2)), dat)
lines(xx, predict(fit, data.frame(x=xx)), lty=2, col=2, lwd=2)
points(1:2, predict(fit, data.frame(x=1:2)), col=2, cex=2)
abline(v=c(1,2))

summary(fit)
Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           0.09125    0.12472   0.732    0.471
x                     1.08363    0.17348   6.246 1.31e-06 ***
I((x - 1) * (x > 1)) -2.30213    0.30408  -7.571 4.90e-08 ***
I((x - 2) * (x > 2)) -0.20386    0.37243  -0.547    0.589
```



Piecewise Cubic Model

- We observe pairs $(x_1, y_1), \ldots, (x_n, y_n)$ from the model

$$y_i = f(x_i) + \epsilon_i,$$

  where f is continuous, $E(\epsilon_i) = 0$, and $V(\epsilon_i) = \sigma^2$

- Let $c_1 < \cdots < c_K$ be *interior knots*. Let $c_0 < c_1$ and $c_{K+1} > c_K$ be *boundary knots*.

- Let f_k be a function defined in $[c_{k-1}, c_k]$. Then a piecewise model has the form

$$f(x|x \in [c_{k-1}, c_k)) = f_k(x)$$

- It is desirable to have f continuous, i.e.,

$$f_k(c_k) = f_{k+1}(c_k), \quad k = 1, 2, \ldots, K$$

- We may also want the derivatives of f to be continuous, e.g.,

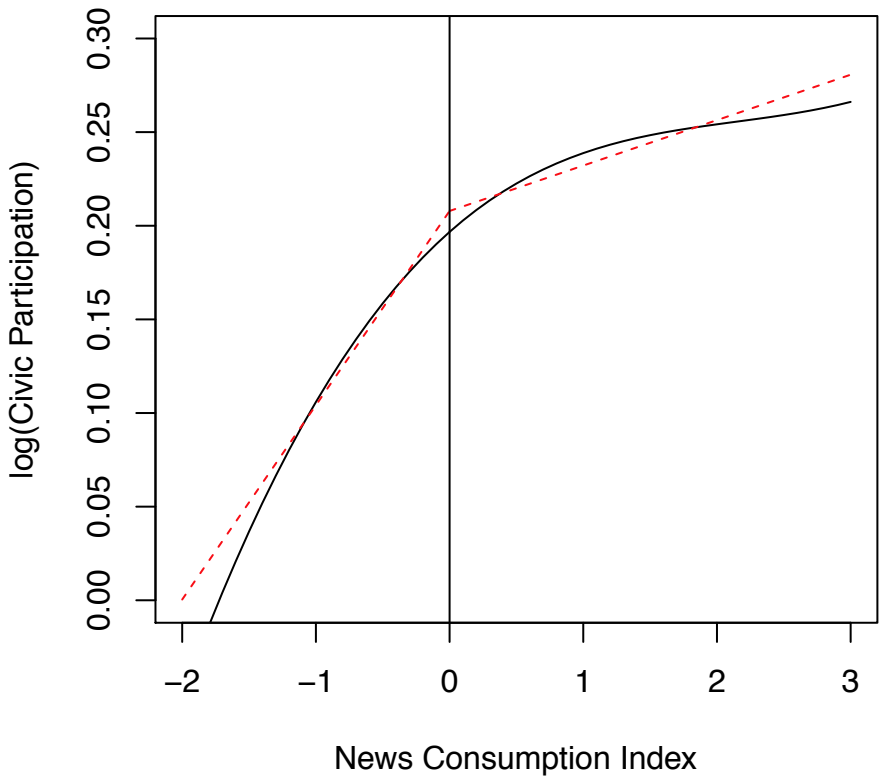
$$f'_k(c_k) = f'_{k+1}(c_k), \quad k = 1, 2, \ldots, K$$

- HTF note that the human eye can detect changes in slope (first derivative) and curvature (second derivative), but it cannot detect changes in higher derivatives. It is common to constrain f, f', and f'' to be continuous.

Cubic Spline for Civic Participation

```
fit = lm(log(cp) ~ index + I(index^2) + I(index^3) + I((index>0)*index^3)
```

Term	Slope	Std Err	t Value
Intercept	0.1967	0.0024	83.18
Index	0.0632	0.0045	14.11
Index^2	−0.0250	0.0048	−5.24
Index^3	0.0027	0.0071	0.38
$(\text{Index})_+^3$	0.0012	0.0079	0.15



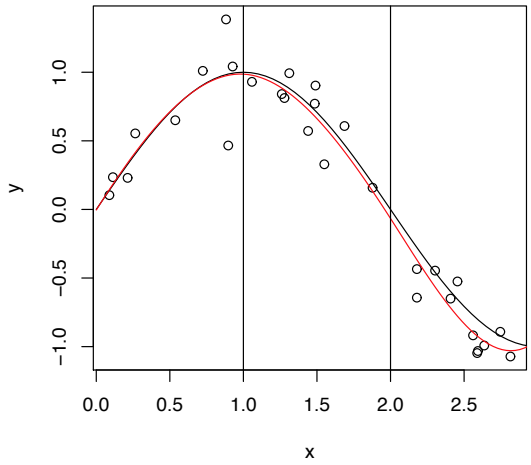
Your turn: write out equations left and right of knot, show function and first two derivatives are continuous

Piecewise-Cubic Polynomials for Wave

```
> fit = lm(y ~ x+I(x^2)+I(x^3)+I((x>1)*(x-1)^3)+I((x>2)*(x-2)^3), dat)
> summary(fit)
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)            -0.004156   0.235995  -0.018    0.986
x                       1.707623   1.549251   1.102    0.281
I(x^2)                 -0.387857   2.346949  -0.165    0.870
I(x^3)                 -0.329302   0.996933  -0.330    0.744
I((x > 1) * (x - 1)^3)  0.708704   1.405600   0.504    0.619
I((x > 2) * (x - 2)^3)  0.644970   1.512991   0.426    0.674
```

- $x \in [0,1] : \hat{y} = -0.004 + 1.71x - 0.39x^2 + 0.33^3$
- $x \in [1,2] : \hat{y} = -0.004 + 1.71x - 0.39x^2 + 0.33^3 + 0.71(x-1)^3$
- $x \in [2,3] : \hat{y} = -0.004 + 1.71x - 0.39x^2 + 0.33^3 + 0.71(x-1)^3 - 0.64(x-2)^3$

```
plot(dat[,1:2])
x = seq(0, 3, length=100)
lines(x, sin(x*pi/2))
lines(x, predict(fit, data.frame(x=x)), col=2)
abline(v=c(1,2))
```



Splines

- If f_k is a cubic polynomial such that f, f', and f'' are continuous, then f is called a *cubic spline* (also *regression splines*).
  - One basis is
$$s(x) = \beta_0 + x\beta_1 + x^2\beta_2 + x^3\beta_3 + \sum_{k=1}^{K} \theta_k (x - c_k)_+^3$$
  - This basis is not recommended for computational reasons. See HTF, Chapter 5 for discussion of B-splines for a better basis.
  - Smoothness is controlled by the number of knots K
- A *natural cubic spline* adds constraints that f be linear for $x < c_1$ and $x > c_K$
- Instead of controlling the smoothness of f with the number of knots, we could allow many knots and estimate with penalized least squares
- *Smoothing splines* — let function $\hat{f}$ be be a spline with many knots. We estimate $\hat{f}$ by minimizing the penalized least-squares objective function
$$\sum_{i=1}^{n} [y_i - \hat{f}(x_i)]^2 + \lambda \int_a^b [\hat{f}''(t)]^2 dt$$
  - λ is a smoothing parameter
  - $\int_a^b [\hat{f}''(t)]^2 dt$ measures the smoothness of $\hat{f}$ and is often called the *penalty term* or *roughness penalty*

Some properties

- The smoothness of each method is controlled by a *smoothing parameter*, which is selected either with graphical inspection or cross validation
  - Bin smoothers: number/width of bins
  - Running means and kernels: "radius" of neighborhoods
  - Loess: number of nearest neighbors
  - Smoothing splines: λ, the roughness penalty
- Definition a *linear smoother* computes the estimates of y_0 as a linear function of $\mathbf{y}$, e.g.,

$$\hat{\mathbf{y}} = \mathbf{S}\mathbf{y},$$

  where $\mathbf{S}$ is an $n \times n$ matrix (also called the *equivalent kernel* of a smoother or smoother matrix)
- The *equivalent number of parameters* or *degrees of freedom* of a linear smoother are given by $\text{tr}(\mathbf{S})$, which allows us to compare the amount of smoothing across smoothers
- Bins, kernels, loess, and smoothing splines are all linear smoothers — other familiar examples include
  - Linear regression ("hat" matrix) $\mathbf{S} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$
  - Ridge regression $\mathbf{S} = \mathbf{X}(\mathbf{X}'\mathbf{X} + k\mathbf{I})^{-1}\mathbf{X}'$

Which smoother should I use?

- The smoothers most commonly offered by commercial software packages include kernels, splines, and loess. In my experience, all three work well—use whatever is offered by your software.
- On picking the smoothing parameter: "Although cross-validation and other automatic methods for selecting a smoothing parameter seem well founded, their performance in practice is sometimes questionable. [they cite examples] . . . We tend to rely more on graphical methods for selecting the smoothing parameter, and use the degrees of freedom measures described in the next section to guide us in selecting reasonable values. Furthermore, we will see that automatic methods are less reliable and far more expensive to implement for additive models, where we need to select several smoothing parameters simultaneously."[3]

[3]Hastie and Tibshirani (1990), page 52.

Splines in Software

- S and R
  - `smooth.spline`
  - `s` within the `gam` function: smoothing spline
  - `bs` within `gam`, `glm`, `lm`: regression spline

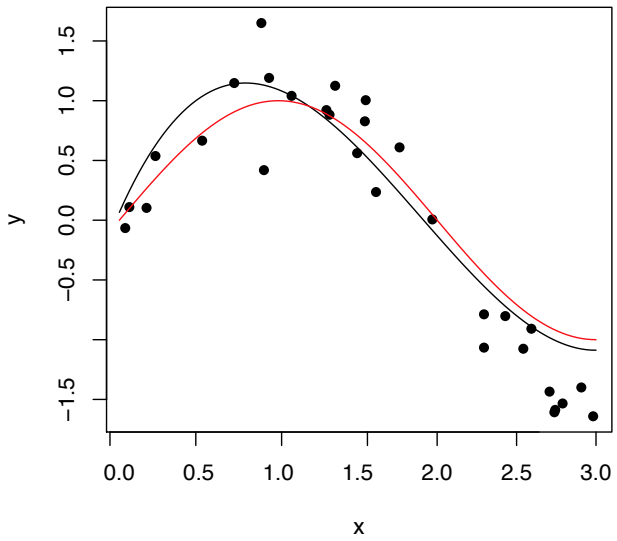
  R seems to have bugs. If `gam` does not work, try `bs` with `lm/glm` instead!
- SAS
  - `SGPLOT` with `PBSLINE`
  - `GAM`: smoothing splines
- Minitab: Graph / Scatterplot / Data View / Smoother
- SPSS: No longer available

Splines in R

```
> library(splines)
> fit = lm(y ~ bs(x), wave)    #  simplest way
> drop1(fit, test="F")
Single term deletions

Model:
y ~ bs(x)
       Df Sum of Sq     RSS     AIC F value    Pr(>F)
<none>               1.0016 -93.987
bs(x)   3     16.46 17.4619 -14.235  142.42 2.963e-16 ***
```

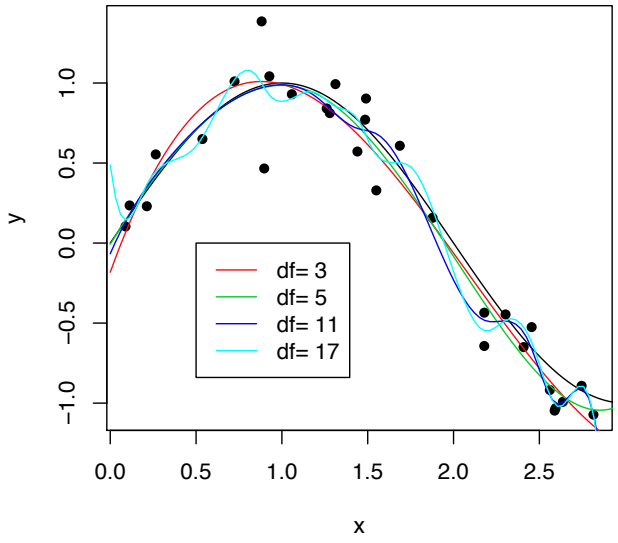
Now fix the end knots at 0 and 3 so that we can produce predicted values over the entire range:

```
fit = lm(y ~ bs(x, Boundary.knots=c(0,3)), dat)
plot(dat[,1:2], pch=16)
lines(x, predict(fit, data.frame(x=x)))
lines(x, sin(x*pi/2), col=2)    # true function
```



Controlling complexity with df=

```
plot(dat[,1:2], pch=16)
lines(x, sin(x*pi/2), col=1)    # true function

fit = lm(y ~ bs(x, df=3, Boundary.knots=c(0,3)), dat)
lines(x, predict(fit, data.frame(x=x)), col=2)
c(mean(fit$residuals^2), mean((test$y-predict(fit, test))^2))

fit = lm(y ~ bs(x, df=5, Boundary.knots=c(0,3)), dat)
lines(x, predict(fit, data.frame(x=x)), col=3)

fit = lm(y ~ bs(x, df=11, Boundary.knots=c(0,3)), dat)
lines(x, predict(fit, data.frame(x=x)), col=4)

fit = lm(y ~ bs(x, df=17, Boundary.knots=c(0,3)), dat)
lines(x, predict(fit, data.frame(x=x)), col=5)
legend(.5,0, paste("df=",c(3,5,11,17)),col=2:5,lty=1)
```



df	Training MSE	Test MSE
3	0.0334	0.0704
5	0.0308	0.0656
7	0.0307	0.0657
9	0.0303	0.0823
11	0.0267	0.1775
13	0.0258	0.4641
15	0.0246	0.3064
17	0.0233	0.2613
19	0.023	0.1027
21	0.0215	0.6122
23	0.0203	0.5368

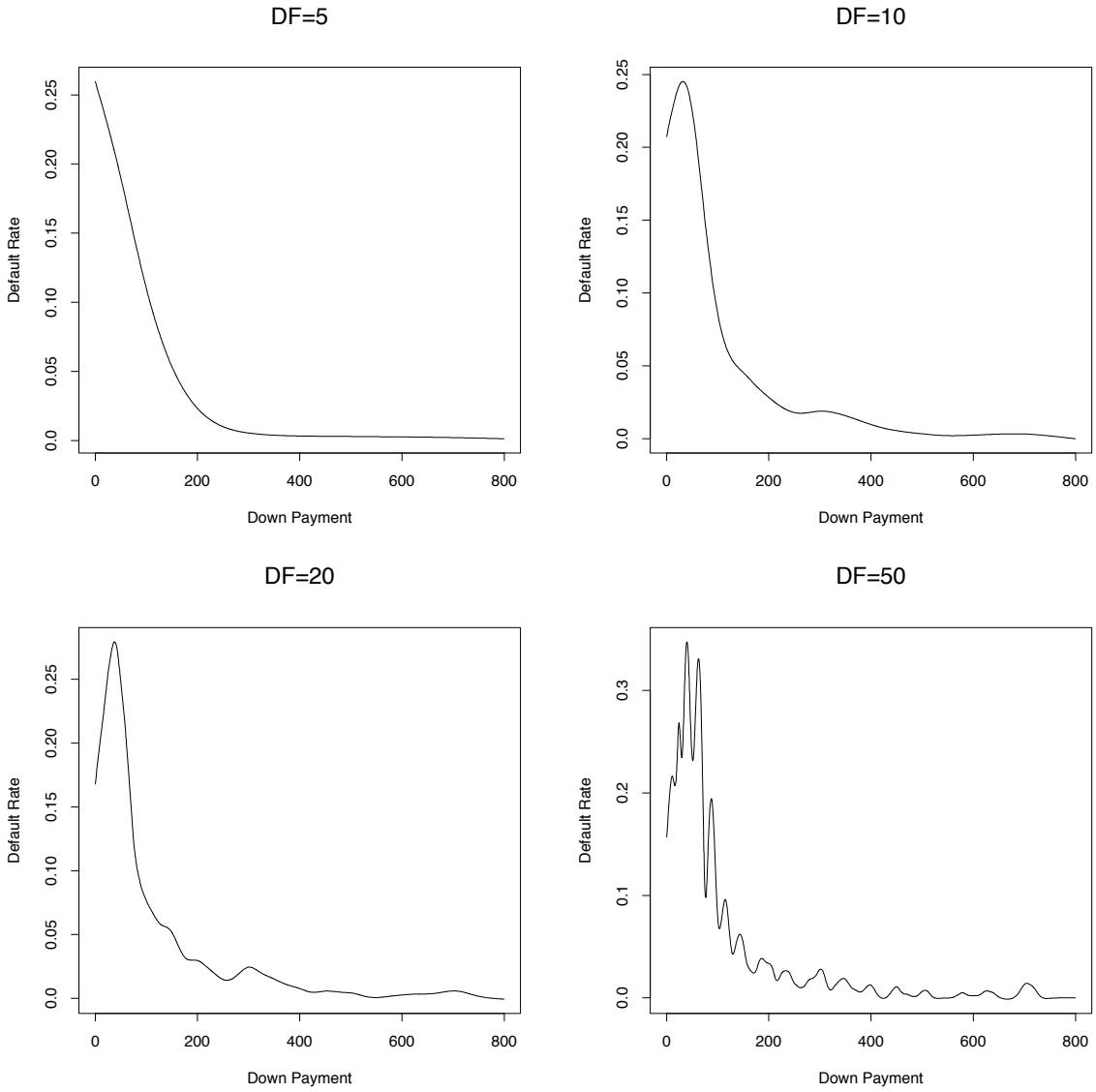
Recall $V(e) = 1/16 = .0625$

Linear Example

```
> library(gam)
> set.seed(12345)
> lin = data.frame(x=runif(100)*3)
> lin$y = 2*lin$x + rnorm(100)/3
> plot(lin)
> fit = lm(y ~ x + bs(x), lin)
> anova(fit)
Analysis of Variance Table

Response: y
          Df  Sum Sq Mean Sq   F value Pr(>F)
x          1 312.094 312.094 2191.9439 <2e-16 ***
bs(x)      2   0.506   0.253    1.7768 0.1747
Residuals 96  13.669   0.142

> fit = gam(y~s(x), data=lin)
> summary(fit)

    Null Deviance: 326.2683 on 99 degrees of freedom
Residual Deviance: 13.5424 on 95 degrees of freedom
AIC: 95.8534

Anova for Parametric Effects
          Df  Sum Sq Mean Sq F value    Pr(>F)
s(x)       1 312.094 312.094  2189.3 < 2.2e-16 ***
Residuals 95  13.542   0.143

Anova for Nonparametric Effects
            Npar Df Npar F  Pr(F)
(Intercept)
s(x)              3 1.4783 0.2254
```

Thus the F test tells us that the spline is unnecessary

Defaulting Customer Problem

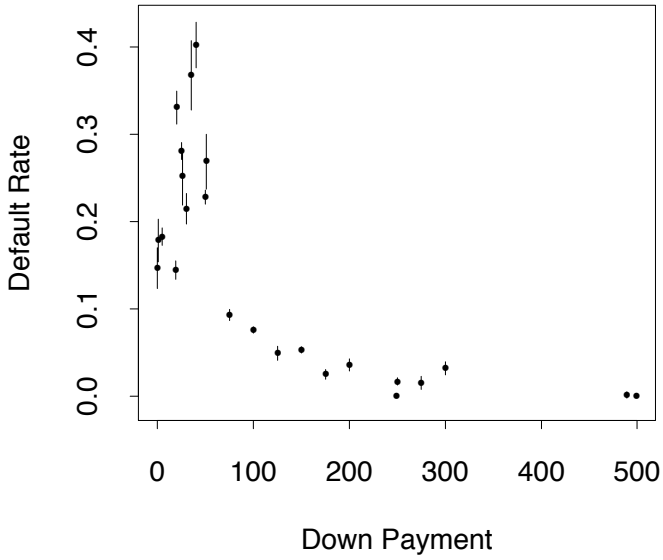


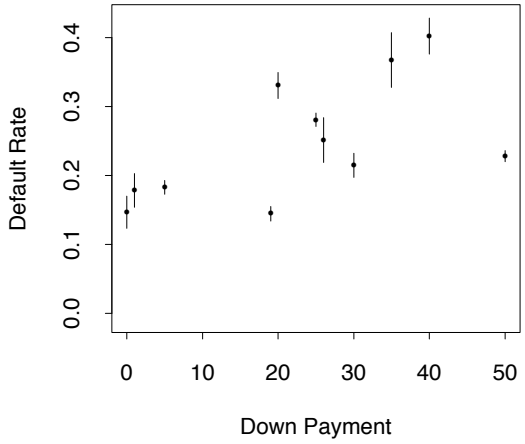
Bin Smoother

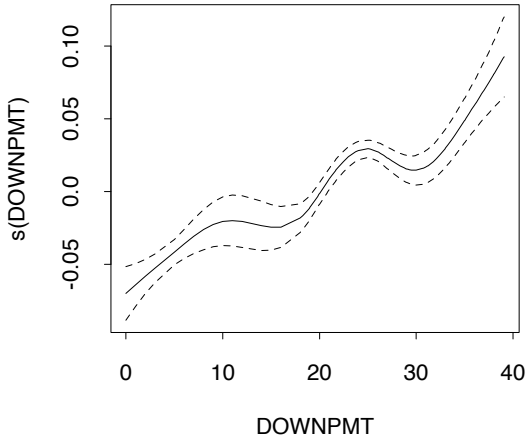
First select cases where **downpmt** is any of the following: 100, 150, 50, 75, 25, 5, 19, 250, 175, 125, 200, 20, 300, 30, 40, 275, 249, 1, 0, 51, 26, 499, 35, 489

Downpmt	FREQ	Default	Lower	Upper
0	892	0.1469	0.1236	0.1701
1	948	0.1783	0.1539	0.2027
5	5610	0.1829	0.1728	0.1930
19	4262	0.1445	0.1340	0.1551
20	2433	0.3305	0.3118	0.3492
25	7904	0.2807	0.2708	0.2907
26	684	0.2515	0.2189	0.2841
30	2138	0.2147	0.1973	0.2321
35	569	0.3673	0.3276	0.4070
40	1361	0.4019	0.3758	0.4280
50	10475	0.2282	0.2201	0.2362
51	774	0.2687	0.2374	0.3000
75	8005	0.0929	0.0866	0.0993
100	18562	0.0760	0.0721	0.0798
125	2780	0.0493	0.0412	0.0573
150	14595	0.0534	0.0497	0.0570
175	2864	0.0251	0.0194	0.0309
200	2665	0.0360	0.0289	0.0431
249	999	0.0000	.	.
250	3647	0.0170	0.0128	0.0212
275	1033	0.0155	0.0079	0.0230
300	2239	0.0322	0.0248	0.0395
489	544	0.0018	−.0018	0.0054
499	630	0.0000	.	.
Total	96613			

Graphs With 95% Confidence Intervals







Additive Models

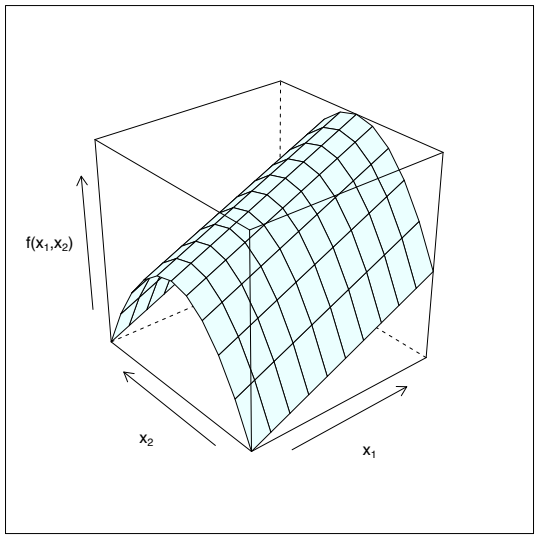
- Now suppose that we have p predictors (rather than 1)
- An *additive model* has the following form:

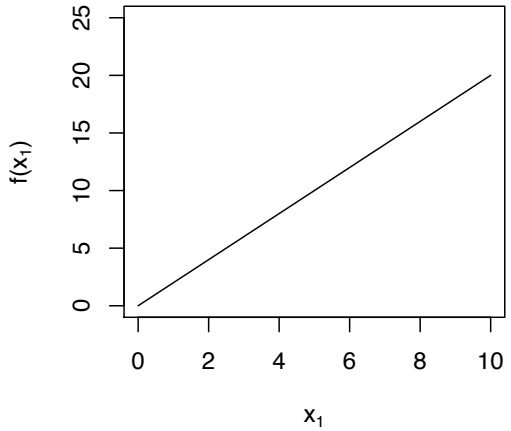
$$y = \beta_0 + f_1(x_1) + \cdots + f_p(x_p) + e$$

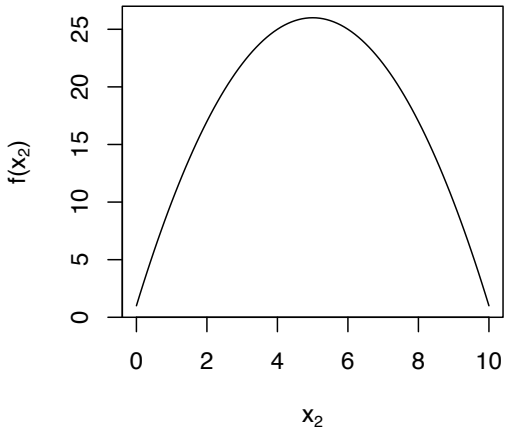
  where $f_j(\cdot)$ is a smooth, univariate function and e is a normal error with mean 0 and variance σ_e^2.
- We can control for other factors as we did with regression.
- When $f_j(x_j) = \beta_j x_j$ this reduces to linear regression.
- *Generalized additive models* (GAMs) allow for a link function and non-normal errors.
- GAMs are easy to interpret by plotting $f_j(x_j)$ against x_j
- GAMs are computationally easy to estimate
- **Additive models do not capture interactions** (but it's not difficult to introduce a product of a dummy with $f_j(x_j)$)
- A good modeling strategy is to consider a hierarchy of models for a particular x: line (df=1 slope), df=3, df=5, ... Choose the most parsimonious model that sufficiently captures nonlinearities.

Parabolic Sheet Example

$$\hat{y} = 1 + \underbrace{2x_1}_{f_1(x_1)} + \underbrace{10x_2 - x_2^2}_{f_2(x_2)} = f_1(x_1) + f_2(x_2), \quad x_1 \geq 0, x_2 \geq 0.$$



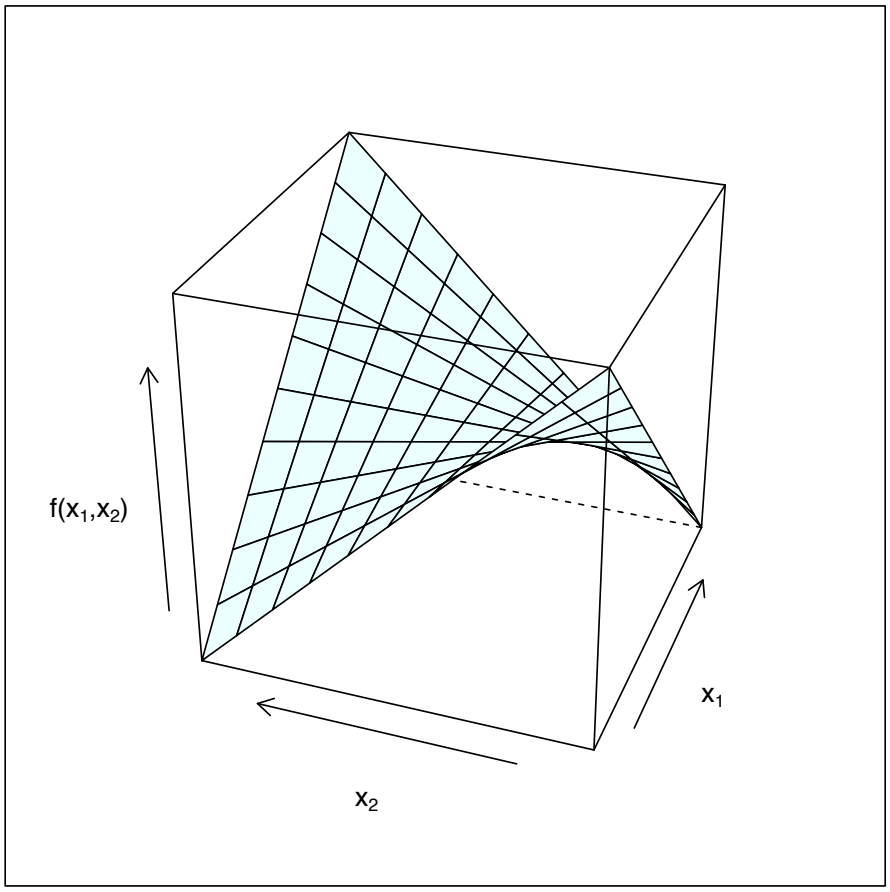




Saddle Example

$$y = f(x_1, x_2) = x_1 x_2, \quad x_j \in [-1, 1]$$

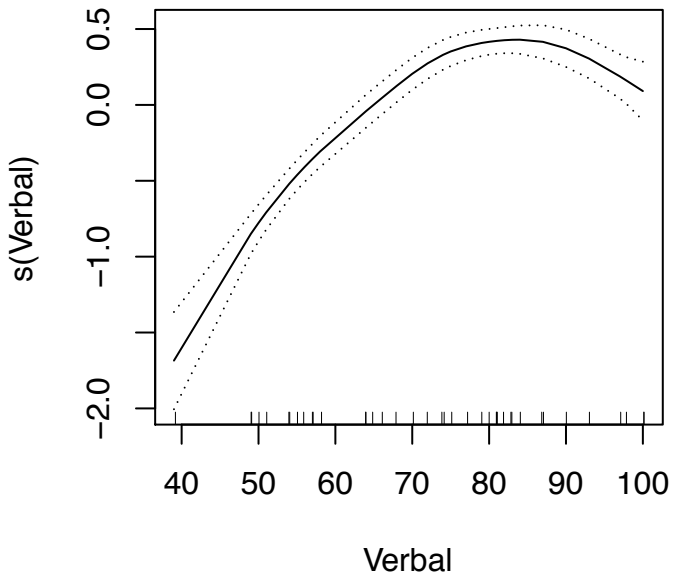
This function is not additive!

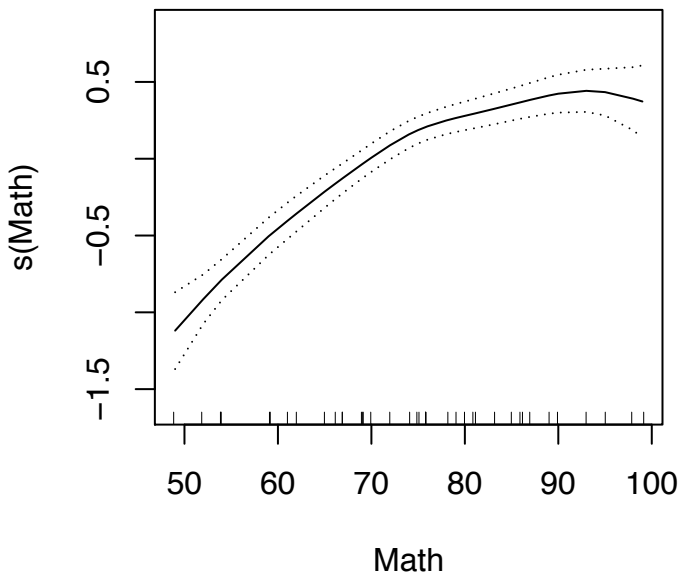


Test Scores Predicting GPA

```
library(gam)
> gpa = read.csv("GPA.csv")
> fit=gam(GPA~s(Verbal)+s(Math),data=gpa)
> summary(fit)
Anova for Parametric Effects
          Df Sum Sq Mean Sq F value Pr(>F)
s(Verbal)  1 5.1266  5.1266  138.16 <.0001
s(Math)    1 6.1549  6.1549  165.87 <.0001
Residuals 31 1.1503  0.0371

Anova for Nonparametric Effects
            Npar Df  Npar F     Pr(F)
(Intercept)
s(Verbal)         3 30.7613 2.025e-09 ***
s(Math)           3  9.7164 0.0001127 ***
```

```
> par(mfrow=c(1,2))
> plot.gam(fit, se=T, scale=2.5)

> # mean squared residuals
> mean(fit$residuals^2)
[1] 0.02875723

> # R-squared
> 1-fit$deviance/fit$null.deviance
[1] 0.938728
```





Civic Participation Example

- Suppose we control for the effects of age, education and income. Note below that news consumption is correlated with the controls.

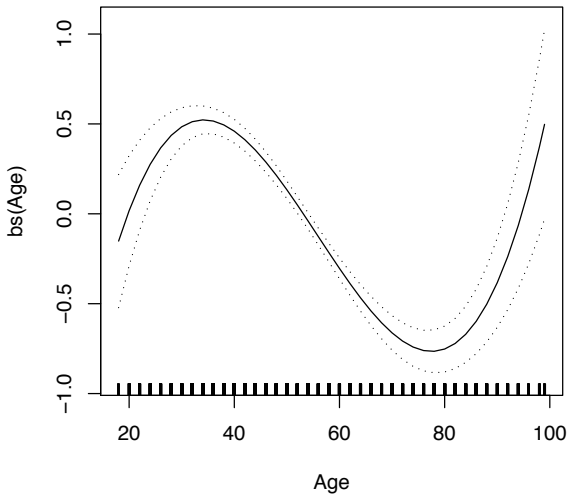
```
> library(splines)
> round(cor(civic),4)
            y   news    age income   educ
y      1.0000 0.2512 0.2053 0.1968 0.2853
news   0.2512 1.0000 0.2838 0.2649 0.3471
age    0.2053 0.2838 1.0000 0.2154 0.1001
income 0.1968 0.2649 0.2154 1.0000 0.4324
educ   0.2853 0.3471 0.1001 0.4324 1.0000
> fit = lm(y ~ news+I(news*(news>0))+age+income+educ, civic)
> summary(fit)

                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       0.0335534  0.0061686   5.439 5.43e-08 ***
news              0.0596484  0.0043220  13.801  < 2e-16 ***
I(news*(news>0)) -0.0496213  0.0060396  -8.216 2.27e-16 ***
age               0.0105728  0.0006225  16.985  < 2e-16 ***
income            0.0016610  0.0003817   4.351 1.36e-05 ***
educ              0.0144997  0.0006276  23.105  < 2e-16 ***
```

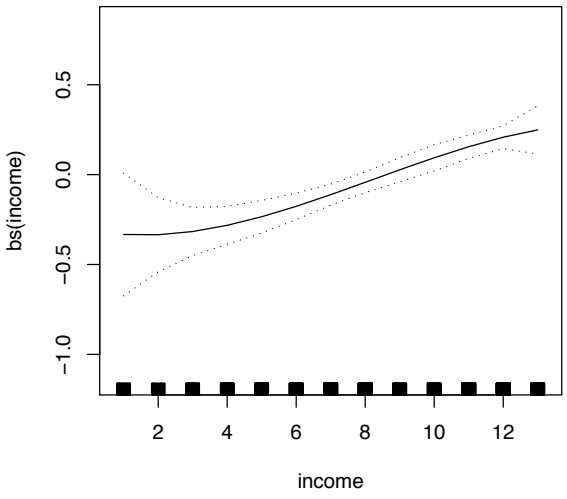
- Without the three controls:
  - For **news** ≤ 0: $y = 0.208 + 0.104$**news**
  - For **news** ≥ 0: $y = 0.208 + 0.024$**news**
- Thus, the effect of news consumption is overstated (biased upward) when we don't control for demographics.

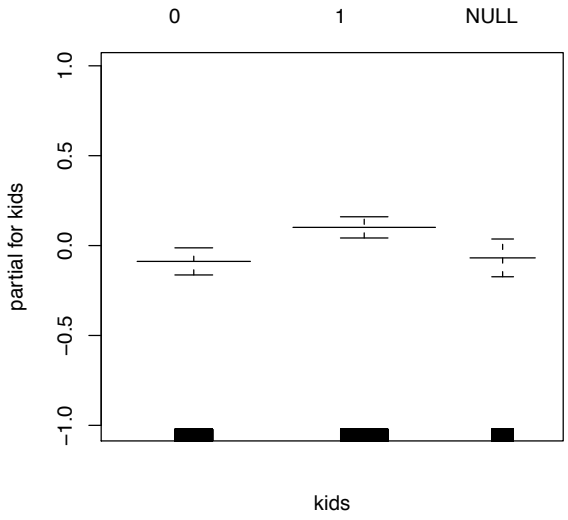
Additive model: Who downloads an app?

```
> library(splines)
> fit = glm(IsMobile ~ bs(Age) + bs(income) + kids, binomial, mobile)
```

```
> library(gam)
> plot.gam(fit, se=T, ask=T, scale=2)
> drop1(fit, test="Chisq")
Single term deletions

Model:
IsMobile ~ bs(Age) + bs(income) + kids
           Df Deviance     LRT   Pr(Chi)
<none>          9607.8
bs(Age)     3   9838.2 230.334 < 2.2e-16 ***
bs(income)  3   9655.9  48.088 2.040e-10 ***
kids        2   9619.6  11.781  0.002765 **
```







Stepwise additive model with interactions

```
> fit = glm(IsMobile ~ 1, binomial, mobile)
> fit2 = step(fit, scope=~bs(Age)*kids + bs(income)*kids + income*kids + Age*kids)


Start:  AIC=10008.83
IsMobile ~ 1

             Df Deviance     AIC
+ bs(Age)     3   9673.5  9681.5
+ Age         1   9738.3  9742.3
+ income      1   9907.8  9911.8
+ bs(income)  3   9906.0  9914.0
+ kids        2   9917.2  9923.2
<none>           10006.8 10008.8

Step:  AIC=9681.52
IsMobile ~ bs(Age)

             Df Deviance     AIC
+ income      1   9620.2  9630.2
+ bs(income)  3   9619.6  9633.6
+ kids        2   9655.9  9667.9
<none>            9673.5  9681.5
- bs(Age)     3  10006.8 10008.8

Step:  AIC=9630.25
IsMobile ~ bs(Age) + income

             Df Deviance    AIC
+ kids        2   9608.4 9622.4
<none>            9620.2 9630.2
+ bs(income)  2   9619.6 9633.6
- income      1   9673.5 9681.5
- bs(Age)     3   9907.8 9911.8

Step:  AIC=9622.43
IsMobile ~ bs(Age) + income + kids

               Df Deviance    AIC
+ kids:income   2   9601.8 9619.8
<none>              9608.4 9622.4
+ bs(income)    2   9607.8 9625.8
- kids          2   9620.2 9630.2
+ bs(Age):kids  6   9604.4 9630.4
- income        1   9655.9 9667.9
- bs(Age)       3   9839.2 9847.2

Step:  AIC=9619.77
IsMobile ~ bs(Age) + income + kids
  + income:kids

               Df Deviance    AIC
<none>              9601.8 9619.8
- income:kids   2   9608.4 9622.4
+ bs(income)    2   9601.2 9623.2
+ bs(Age):kids  6   9598.0 9628.0
- bs(Age)       3   9830.1 9842.1


> summary(fit2)
                Estimate Std. Error z value Pr(>|z|)
(Intercept)     -2.325513   0.235133  -9.890  < 2e-16 ***
bs(Age)1         2.478497   0.485903   5.101 3.38e-07 ***
bs(Age)2        -2.926973   0.280449 -10.437  < 2e-16 ***
bs(Age)3         0.675340   0.377958   1.787   0.0740 .
income           0.074135   0.015097   4.910 9.09e-07 ***
kids1            0.241915   0.193297   1.252   0.2107
kidsNULL         0.509951   0.219615   2.322   0.0202 *
income:kids1    -0.006241   0.019678  -0.317   0.7511
income:kidsNULL -0.059054   0.024424  -2.418   0.0156 *
```

Curse of Dimensionality

- Smoothing works very well when there is only one predictor, but we will encounter problems when there are more predictors
- One predictor: consider the domain [0, 1] and suppose we have 10 equal-spaced points covering it
- Two predictors: to achieve the same coverage of the unit square we would need $10^2 = 100$ points
- p predictors: to achieve the same coverage we would need 10^p points, which grows quickly with p.
- Fact: high-dimensional spaces will be sparse (even in the world of big data!)
- What to do? Two approaches are
  - Recursive partitioning of the domain
  - Projection onto lower-dimensional subspaces and then "smooth"

Regression Trees

- Let f be continuous and $\mathbf{x}_i$ $(p \times 1)$. Assume

$$y_i = f(\mathbf{x}_i) + e_i \quad (i = 1, \ldots, n)$$

- New notation:
  - $\mathcal{D} = \{1, \ldots, n\}$ contains the indicies of a "data set"
  - Let $\mathcal{D}_k \subseteq \mathcal{D}$ contain a subset of $n_k = |\mathcal{D}_k|$ cases. Note: $\mathcal{D}_k$ will correspond to "node k."
  - Let the mean and sum of squares for $\mathcal{D}_k$ be

$$\bar{y}_k = \frac{1}{n_k} \sum_{i \in \mathcal{D}_k} y_i \quad \text{and} \quad \text{RSS}_k = \sum_{i \in \mathcal{D}_k} (y_i - \bar{y}_k)^2$$

- Suppose we *split* $\mathcal{D}_k = \mathcal{D}_l \cup \mathcal{D}_r$ into nonempty, disjoint subsets (called the *left child* and *right child*). The *improvement* is

$$\Delta\text{RSS} = \text{RSS}_k - (\text{RSS}_l + \text{RSS}_r).$$

- The *tree algorithm* is as follows

  Function BuildTree(data set $\mathcal{D}$)

  Find best split $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_r$ to maximize ΔRSS

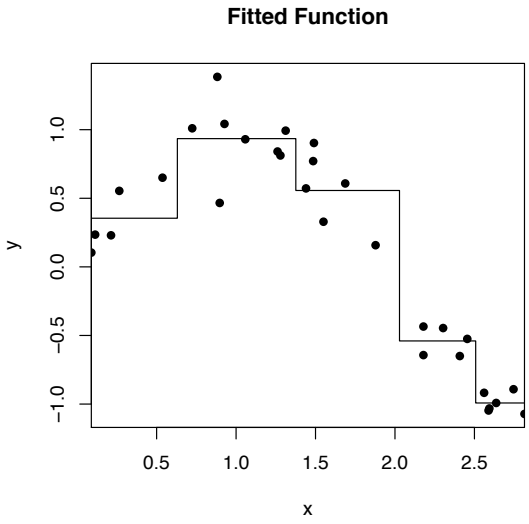
  If *stopping criteria* met, exit function

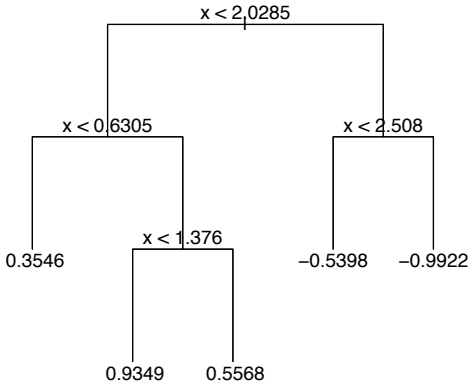
  BuildTree(D_l)

  BuildTree(D_r)

Wave Example Revisted

```
> library(tree)
> fit = tree(y ~ x, wave)
> partition.tree(fit, main="Fitted Function")
> points(wave$x, wave$y, pch=16)

> plot(fit) # Dendrogram
> text(fit, cex=.8)
> deviance(fit) # gives RSS
[1] 1.135359

> # RSS of NULL model
> sum((wave$y - mean(wave$y))^2)
[1] 17.46195

> fit
node), split, n, deviance, yval
      * denotes terminal node

 1) root 30 17.46000  0.1314
   2) x < 2.0285 19  2.20100  0.6628
     4) x < 0.6305 5  0.21970  0.3546 *
     5) x > 0.6305 14  1.33600  0.7729
      10) x < 1.376 8  0.46690  0.9349 *
      11) x > 1.376 6  0.37950  0.5568 *
   3) x > 2.0285 11  0.62740 -0.7865
     6) x < 2.508 5  0.04279 -0.5398 *
     7) x > 2.508 6  0.02650 -0.9922 *
```





Some terms

- A CART model divides the predictor ($\mathbf{x}$) space by successively splitting into rectangular regions and models the response (Y) as constant over each region
- This can be schematically represented as a "tree:"
  - each interior node of the tree indicates on which predictor variable you split and where you split
  - each terminal node (aka leaf) represents one region and indicates the value of the predicted response in that region
- *Edge*: connection between two *nodes*
- *Tree*: a collection of one or more *nodes* and *edges* that satisfy certain requirements[4]
- *Root node*: top node of a tree
- Each node (except the root) has exactly one node "above" it, called the *parent*
- The nodes "below" (directly connected by an edge) a node are called *children* of the parent node
- Terms such as *grandparent*, *grandchild*, and *sibling* are also used
- *Leaf node* or *terminal node*: a node with no children
- *Nonterminal node*: a node with at least one child
- *Binary tree*: a tree with two types of nodes
  1. terminal nodes
  2. nonterminal nodes with exactly two children, labeled the *left child* and *right child*
- *Split*: a partition of any single predictor variable
- *Depth*: maximum number of nodes in a path between the root and a terminal node

[4]See any textbook on Algorithms or Data Structures, e.g., Sedgewick (1992), *Algorithms in C++*, Addison-Wesley.

Fitting a Regression Tree

- A CART model is fit using a data frame structured just like in regression (one response column and many predictor columns)
- Fitting the model entails growing the tree one node at a time
  - At each step, the single best next split (which predictor and where to split) is the one that gives the biggest reduction in RSS
  - The fitted or predicted response over any region is simply the average response over that region. The errors used to calculate the RSS are the response values minus the fitted values.
  - Stop splitting when reduction in RSS with the next split is below a specified threshold, all node sizes are below a threshold, etc.
  - Most algorithms overfit then **prune** back branches
- After fitting a CART model, software returns the final fitted tree, which can be used for prediction/interpretation

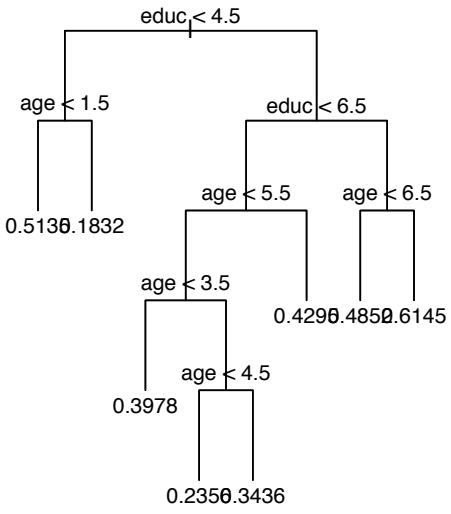
Art Museum Example (Regression Tree)

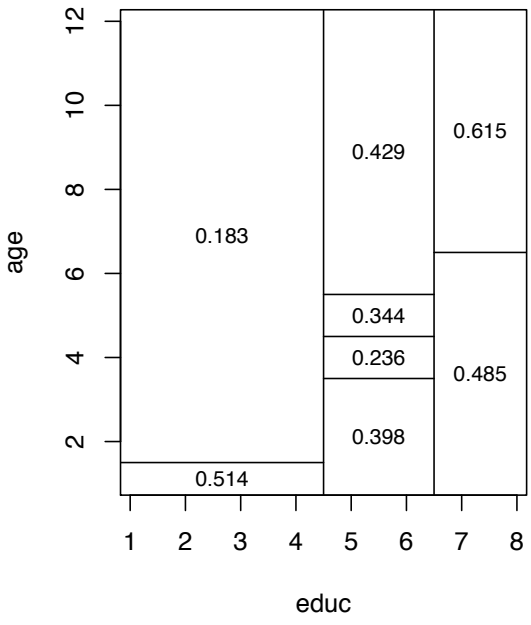
How do age and education affect whether or not a person goes to art museums?

```
library(tree)
amus = read.csv("amus.csv")
fit = tree(amus~age+educ, amus, mindev=1e-6)
fit2 = prune.tree(fit, best=8)
par(mfrow=c(1,2))
plot(fit2, type="uniform")
text(fit2, cex=.8)
partition.tree(fit2, cex=.8)
par(mfrow=c(1,1))
fit2
```

```
node), split, n, deviance, yval
      * denotes terminal node

 1) root 2595 593.500 0.3541
```

```
2) educ < 4.5 894 141.400 0.1969
  4) age < 1.5 37   9.243 0.5135 *
  5) age > 1.5 857 128.200 0.1832 *
3) educ > 4.5 1701 418.500 0.4368
  6) educ < 6.5 1156 273.200 0.3832
   12) age < 5.5 518 113.900 0.3263
     24) age < 3.5 181  43.360 0.3978 *
     25) age > 3.5 337  69.080 0.2878
       50) age < 4.5 174  31.340 0.2356 *
       51) age > 4.5 163  36.760 0.3436 *
   13) age > 5.5 638 156.300 0.4295 *
  7) educ > 6.5 545 134.900 0.5505
   14) age < 6.5 270  67.440 0.4852 *
   15) age > 6.5 275  65.140 0.6145 *
```

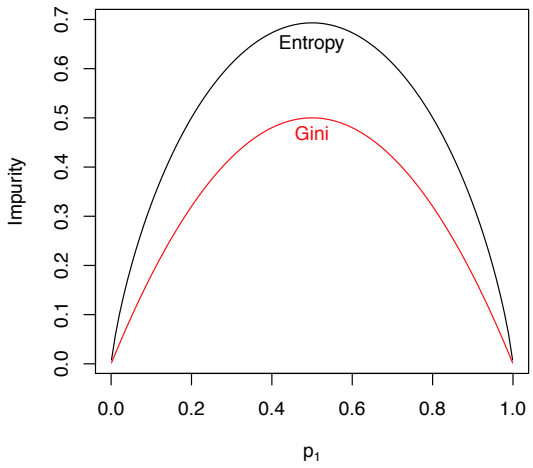




Impurity Measures for Classification Trees

- **Impurity measures**. Let t be a node and $i(t)$ be the impurity of t:
  - *Gini index*: $i(t) = 1 - \sum_j \hat{p}_j^2$
  - *Entropy*: $i(t) = -\sum_j \hat{p}_j \log \hat{p}_j$

  where $\hat{p}_j$ is the estimated probability of being in class j at node t. Smaller values indicate nodes with greater purity.



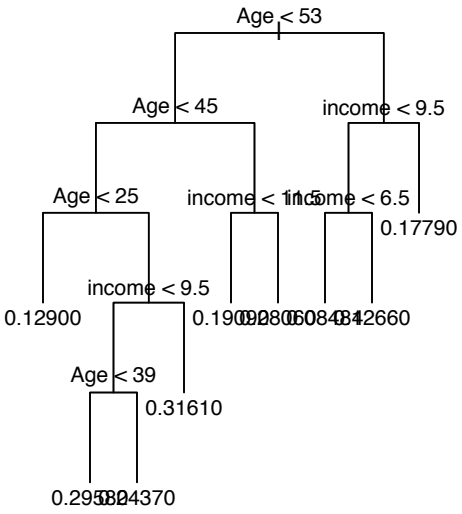
- **Change in impurity (improvement)**. Let t be a parent node with children t_L and t_R. Let p_L be the fraction of observations in the left child and p_R the fraction in the right ($p_L + p_R = 1$). The change in impurity (improvement) due to this split is

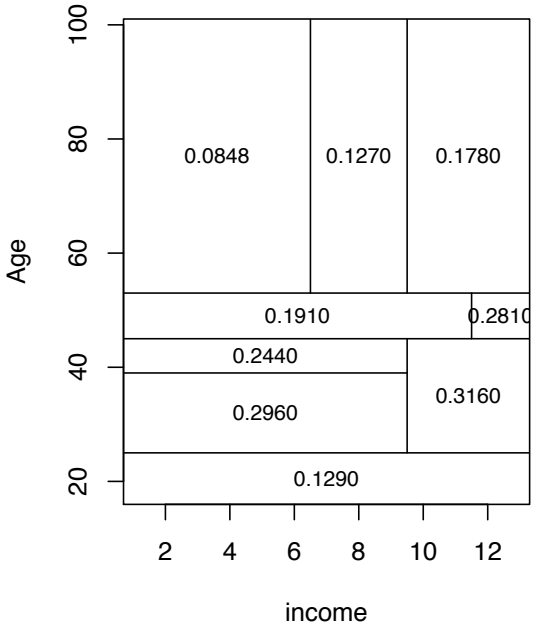
$$i(t) - p_L i(t_L) - p_R i(t_R)$$

Mobile app example

```
library(tree)
fit = tree(IsMobile~Age+income, mobile,
  mindev=.0001)
fit2 = prune.tree(fit, best=9)
plot(fit2, type="uniform")
text(fit2, cex=.8)
partition.tree(fit2, cex=.8,
  ordvars=c("income", "Age"))
fit2
```

```
 1) root 9864 1608.00 0.20500
   2) Age < 53 5444 1060.00 0.26470
     4) Age < 45 3518  720.50 0.28740
       8) Age < 25 155   17.42 0.12900 *
       9) Age > 25 3363  699.00 0.29470
        18) income < 9.5 1813  362.60 0.27630
          36) Age < 39 1136  236.60 0.29580 *
          37) Age > 39 677  124.80 0.24370 *
        19) income > 9.5 1550  335.10 0.31610 *
     5) Age > 45 1926  334.00 0.22330
      10) income < 11.5 1231  190.10 0.19090 *
      11) income > 11.5 695  140.30 0.28060 *
   3) Age > 53 4420  504.60 0.13140
     6) income < 9.5 2745  253.80 0.10310
      12) income < 6.5 1544  119.90 0.08484 *
      13) income > 6.5 1201  132.80 0.12660 *
     7) income > 9.5 1675  245.00 0.17790 *
```





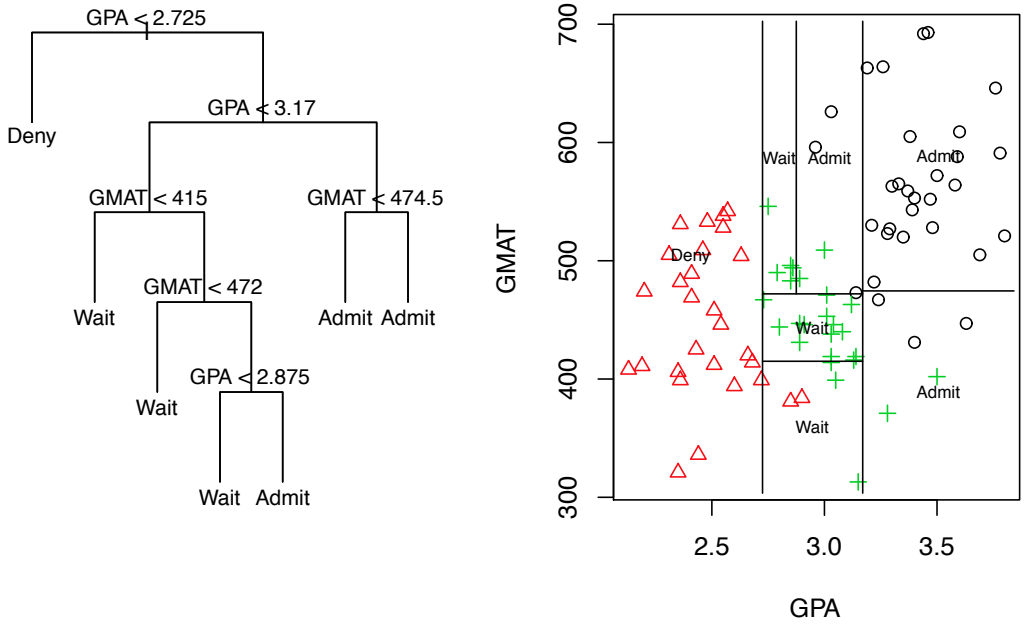
Explanation of Terms

- Two types of trees
  - Regression trees: numerical dependent variable
  - Classification trees: categorical dependent variable (in R, make the dependent variable a `factor`)
- There are many tree models, e.g.,
  - CART = Classification And Regression Trees
  - CHAID = CHi-Squared Automatic Interaction Detector
  - C4.5 = like CART
- Some common *stopping Rules*
  - All observations in the same class
  - Minimum number of observations in a node (`minsize`, default=10)
  - Changes in impurity (`mindev`). Note CHAID stops when no more significant splits
  - Tree depth (not used in R)

Admissions Example

When the decision boundaries are not aligned with the coordinate axes tree models have problems.

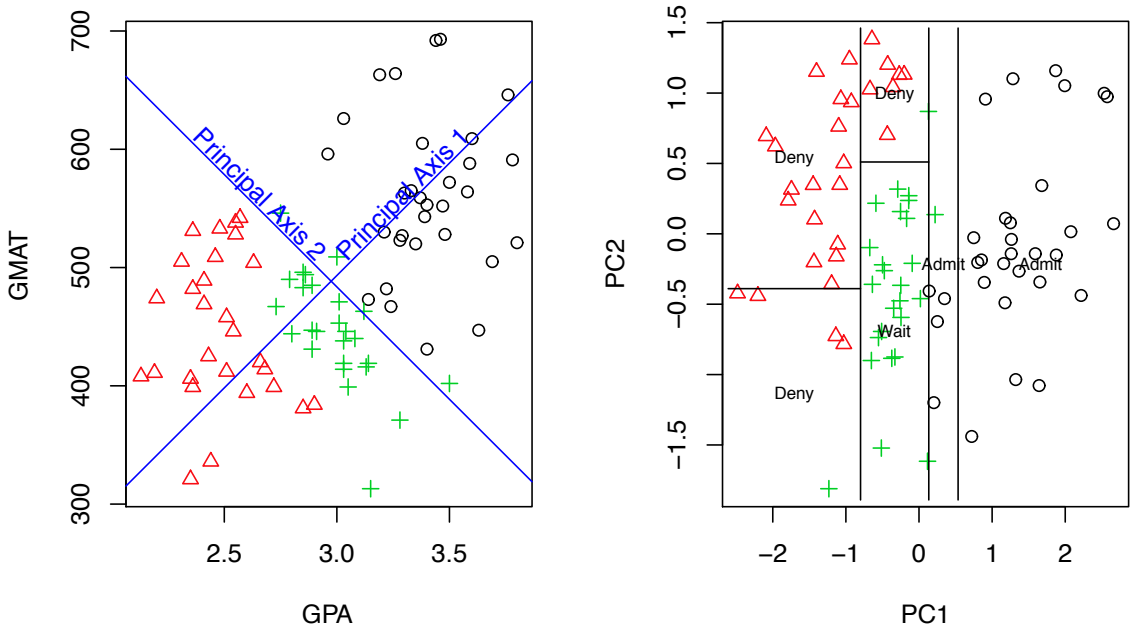
```
> library(tree)
> gmatgpa = read.table("teach2/304/gmatgpa.dat", header=T)
> gmatgpa$status = factor(gmatgpa$admit, 1:3, c("Admit","Deny","Wait"))
> fit = tree(status ~ GMAT+GPA, gmatgpa)
> par(mfrow=c(1,2))
> plot(fit, type="uniform")
> text(fit, cex=.8)
> with(gmatgpa, plot(GPA, GMAT, col=admit, pch=admit))
> partition.tree(fit, add=T, cex=.7)
```



Admissions Example with PCA

We can rotate the axes with **principal component analysis** (PCA). Note: PC1 is proportional to the average/sum of the standardized variables, while PC1 is proportional to the difference between them.

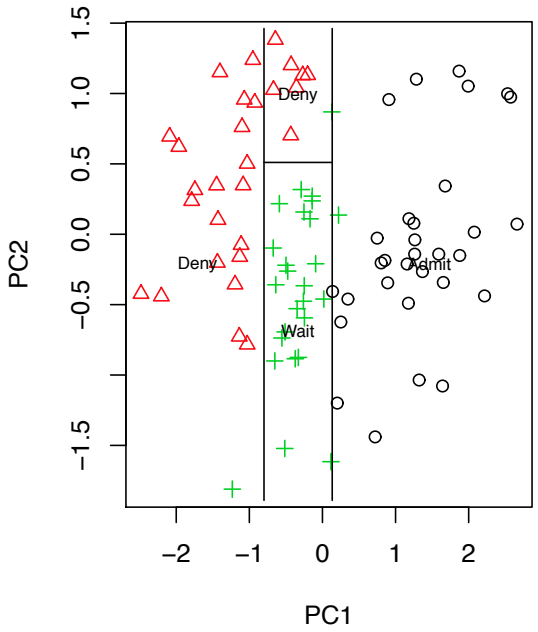
```
fit2 = prcomp(gmatgpa[,1:2], scale=T)
gmatgpa$PC1 = fit2$x[,1]
gmatgpa$PC2 = fit2$x[,2]
fit3 = tree(status ~ PC1+PC2, gmatgpa)

myline = function(x0, y0, m, ...) abline(y0-m*x0, m, ...)
par(mfrow=c(1,2))
with(gmatgpa, plot(GPA, GMAT, col=admit, pch=admit))
myline(fit2$center[1], fit2$center[2], fit2$scale[2]/fit2$scale[1], col=4)
myline(fit2$center[1], fit2$center[2], -fit2$scale[2]/fit2$scale[1], col=4)
text(fit2$center[1]+.05, fit2$center[2]+20, "Principal Axis 1", srt=45, adj=0, col=4)
text(fit2$center[1]-.05, fit2$center[2]+20, "Principal Axis 2", srt=-45, adj=1, col=4)
with(gmatgpa, plot(PC1, PC2, col=admit, pch=admit))
partition.tree(fit3, add=T, cex=.7)
```

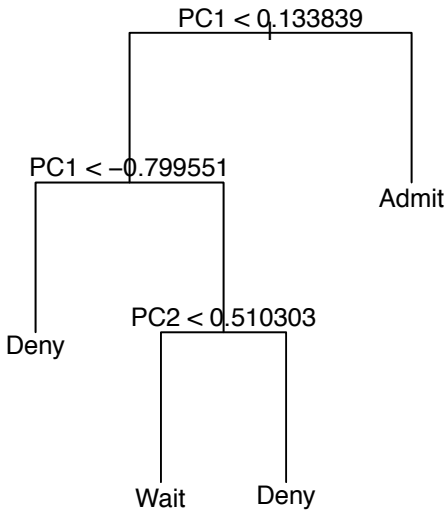


Admissions Example with PCA and Pruning

We can simplify the model by pruning

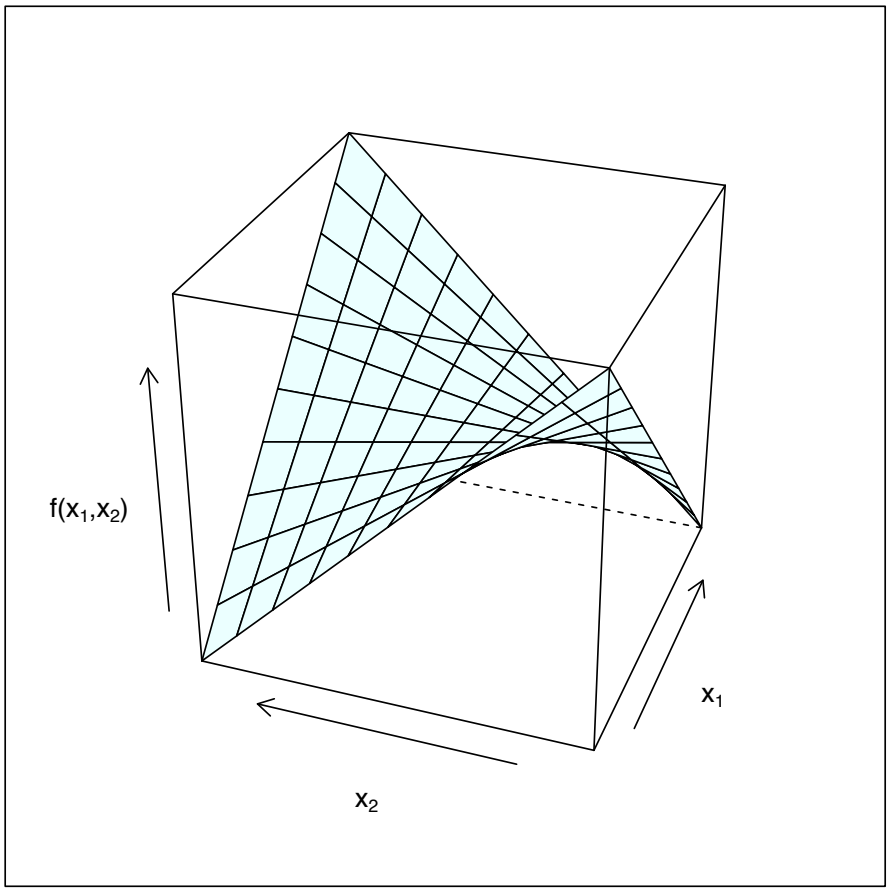
```
> fit4 = prune.tree(fit3, best=4)
> par(mfrow=c(1,2))
> with(gmatgpa, plot(PC1, PC2, col=admit, pch=admit))
> partition.tree(fit4, add=T, cex=.7)
> plot(fit4, type="uniform")
> text(fit4)
```





Saddle Function (xor)

$$y = f(x_1, x_2) = x_1 x_2, \quad x_j \in [-1, 1]$$



- No single split offers any improvement and with noise added to y, any single split offers only small improvements
- Two consequitive splits on different x's offers large improvements
- **Implication: overgrow trees to detect such interactions, then prune unnecessary splits**

Your Turn

Consider building a classification tree predicting dichotomous response `y` (taking values 0 and 1) from three dichotomous predictors `x1`, `x2`, and `x3`. You have a training set of $n = 800$ observations, summarized the the table below:

x1	x2	x3	n	$\sum y$
0	0	0	100	92
0	0	1	100	90
0	1	0	100	8
0	1	1	100	12
1	0	0	100	15
1	0	1	100	12
1	1	0	100	89
1	1	1	100	92

Notice that the design is balanced, with 100 observations for each of the 2^3 combinations of `x` variables. The last column gives the sum of the `y` values in the group. For example, the first row shows that, among the 100 cases with `x1=x2=x3=0`, there are 92 where `y`=1, and $100 - 92 = 8$ where `y` $= 0$.

1. Using the Gini index find the impurity of the root node, which has all 800 cases. Hint: $i(t) = 1 - \sum_j \hat{p}_j^2$ for node t, where $\hat{p}_j$ is the estimated probability of being in class j.

2. Consider splitting on `x1`, i.e., the left child has all 400 observations where `x1`=0 and the right child has 400 observations where `x1`=1. Find the impurity of the left and right children and compute the improvement. Hint: $i(t) - p_L i(t_L) - p_R i(t_R)$ is the improvement, where p_L is the fraction of cases in the left child t_L and p_R is the fraction of cases in right child t_R.

3. We will now see if it would be better to split on one of the other two variables. Compute the improvement for splitting on `x2` and also for splitting on `x3`. Show work on back or attach scratch paper if necessary.

4. Which of the three splits produces the greatest improvement? Hint: it should be `x1` or `x2`. Comment briefly on whether this split improved the predictions of `y` by much. Label your split on the tree next to the data table.

5. Now find the best split for the left child. Indicate the variable you split on, the impurity of the children and the improvement. Show work on back. Label the split on the tree on page 1.

6. Did this split improve the fit in comparison with the split of the root node?

7. Now find the best split for the right child of the root node. Indicate the variable you split on, the impurity of the children and the improvement. Label the split on the tree on page 1.

8. Consider splitting the left child of the left child of the root node (the left-most grandchild of the root) using the remaining variable. Does the split improve the fit much, or should this split be pruned?

Your Turn Solution

```
> dat = expand.grid(x3=0:1, x2=0:1, x1=0:1)
> dat2 = rbind(dat, dat)
> dat$y = c(rep(0,8), rep(1,8))
> dat2$n = c(8,10,92,88,85,88,11,8, 92,90,8,12,15,12,89,92)
> fit = tree(y ~ x1+x2+x3, dat2, weight=n, minsize=5, mindev=.0001)
> fit
```

Solution:

1. $i(t) = 1 - (410/800)^2 - (390/800)^2 = 319,800/640,000 \approx 0.4997$.
2. $i(t_L) = 1 - (202/400)^2 - (198/400)^2 \approx 0.5000$;
   $i(t_R) = 1 - (208/400)^2 - (192/400)^2 \approx 0.4992$; Improvement(`x1`) $=$
   $0.4997 - 0.5000/2 - 0.4992/2 \approx 0.000112$.
3. Splitting on `x2`: $i(t_L) = 1 - (209/400)^2 - (191/400)^2 \approx 0.4990$;
   $i(t_R) = 1 - (201/400)^2 - (199/400)^2 \approx 0.5000$; improvement(`x2`) $=$
   $0.4997 - 0.4990/2 - 0.5000/2 \approx 0.000200$

   Splitting on `x3`: $i(t_L) = 1 - (204/400)^2 - (196/400)^2 \approx 0.4998$;
   $i(t_R) = 1 - (206/400)^2 - (194/400)^2 \approx 0.4996$; improvement(`x3`) $=$
   $0.4997 - 0.4996/2 - 0.4992/2 \approx 0.000012$
4. Split on `x2`, because $0.000200 > 0.000112$ and $0.000200 > 0.000012$. The improvement is very small.
5. The best split is on `x1`: $i(t_L) = 1 - (182/200)^2 - (18/200)^2 \approx 0.1638$;
   $i(t_R) = 1 - (27/200)^2 - (173/200)^2 \approx 0.2336$; improvement(`x1`) $=$
   $0.4990 - 0.1638/2 - 0.2336/2 \approx 0.3003$
6. The improvement is large compared with that of the split of the root node.
7. The best split is on `x1`: $i(t_L) = 1 - (20/200)^2 - (180/200)^2 \approx 0.1800$;
   $i(t_R) = 1 - (181/200)^2 - (19/200)^2 \approx 0.1720$; improvement(`x1`) $=$
   $0.5000 - 0.1800/2 - 0.1720/2 \approx 0.2380$
8. Splitting on `x3` doesn't improve much. Prune it.

Strengths & Weaknesses of Trees

+ Computationally fast

+ Very interpretable (provided tree is fairly small)

+ Easy to explain to normal people

+ Easy to handle missing value and multi-level categorical predictors

+ Invariant to monotonic transformations of the predictor variables (no need to find transformations of x's)

+ Unaffected by outlying x values

+ Well-suited for modeling nonlinearities and interactions

+ Easy to score—output is if-then logic

− Not well-suited for modeling linear main effects, but it's easy to combine with smooth models: fit linear/GAM model, use residuals as dependent variable in tree

− Prediction surface not "smooth"

− High variance (unstable) estimate, just like stepwise regression (lends itself to bagging)

Introduction to Bagging and Random Forests

Trees are high-variance estimates in that the sequence of splits (and thus the resulting tree) are highly sensitive to sampling variation—if you fit a tree on a different sample from the same population, the tree will likely be very different. Stepwise regression has similar issues. Suppose we draw two samples of size $n = 100$ and fit a tree:

```
> bank = read.csv("bank.csv")
> set.seed(54321)
> tree(tips ~ ., dat[sample(1:477, 100),])
node), split, n, deviance, yval
      * denotes terminal node

  1) root 100 162.600 4.299
    2) auto < 4.5 81 112.400 4.046
      4) debit < 3.5 61  79.010 3.887
        8) income < 1.5 8  11.260 3.216 *
        9) income > 1.5 53  63.610 3.988
         18) medical < 1.5 15   7.683 4.464 *
         19) medical > 1.5 38  51.200 3.800
           38) gifts < 2.5 22  24.400 3.428
             76) travel < 2.5 16  13.530 3.256
              152) clothes < 2.5 8   2.772 3.710 *
              153) clothes > 2.5 8   7.450 2.801 *
             77) travel > 2.5 6   9.131 3.886 *
           39) gifts > 2.5 16  19.550 4.313 *
      5) debit > 3.5 20  27.170 4.532
       10) debit < 4.5 12   6.562 4.955 *
       11) debit > 4.5 8  15.250 3.898 *
    3) auto > 4.5 19  22.960 5.376
      6) debit < 2.5 7   6.469 6.084 *
      7) debit > 2.5 12  10.920 4.962 *
```

```
> tree(tips ~ ., dat[sample(1:477, 100),])
node), split, n, deviance, yval
      * denotes terminal node

  1) root 100 197.1000 4.207
    2) auto < 2.5 49  77.5700 3.700
      4) debit < 3.5 39  53.4500 3.389
        8) saving < 3.5 30  44.4400 3.211
         16) travel < 2.5 23  31.2900 3.010
           32) credit < 2.5 17  23.7900 3.259
             64) auto < 1.5 7   4.0040 2.682 *
             65) auto > 1.5 10  15.8200 3.664
              130) income < 4.5 5   5.7930 3.136 *
              131) income > 4.5 5   7.2450 4.191 *
           33) credit > 2.5 6   3.4450 2.303 *
         17) travel > 2.5 7   9.1790 3.870 *
        9) saving > 3.5 9   4.8510 3.985 *
      5) debit > 3.5 10   5.6340 4.914
       10) clothes < 2.5 5   0.5017 5.373 *
       11) clothes > 2.5 5   3.0250 4.455 *
    3) auto > 2.5 51  94.8500 4.694
      6) credit < 2.5 32  51.6300 4.288
       12) goout < 2.5 9  11.5100 3.384 *
       13) goout > 2.5 23  29.8700 4.642
         26) income < 3.5 8  12.6900 5.381 *
         27) income > 3.5 15  10.5000 4.248
           54) saving < 2.5 6   2.5360 4.742 *
           55) saving > 2.5 9   5.5200 3.919 *
      7) credit > 2.5 19  29.0700 5.378
       14) income < 4.5 8   3.2330 6.085 *
       15) income > 4.5 11  18.9200 4.864
         30) income < 5.5 5   5.0110 3.845 *
         31) income > 5.5 6   4.4080 5.712 *
```

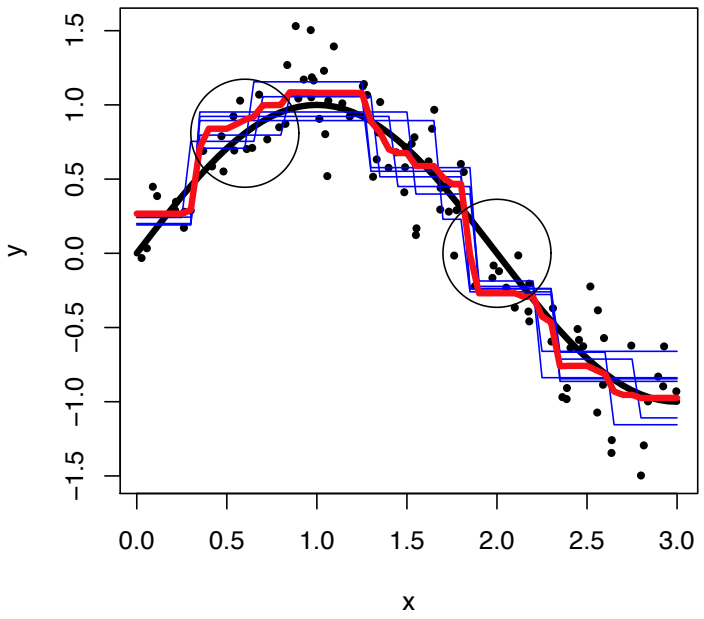
Introduction to Bagging

- Bootstrap aggregation (**bagging**) exploits this variation to produce a better estimate:
  1. Repeat the following "many" times (e.g., $B = 25$ or so):
     (a) Draw a sample of size n with replacement from the available data.
     (b) Estimate a tree
  2. Average the estimates from the B trees for the final estimates.
- "Tests on real and simulated data sets using CART and subset selection in linear regression show that bagging can give substantial gains in accuracy. The vital element is the instability of the prediction method. If perturbing the learning set can cause significant changes in the predictor constructed, then bagging can improve accuracy (Breiman, 1994, p. 1)."
- "Bagging goes a long ways towards making a silk purse out of a sow's ear, especially if the sow's ear is twitchy. It is a relatively easy way to improve an existing method, since all that needs adding is a loop in front that selects the bootstrap sample and sends it to the procedure and back back end that does the aggregation. What one loses, with the trees, is a simple and interpretable structure. What one gains is increased accuracy. (p. 17)"
- Bagging (and random forests) ...
  - Unlikely to overfit data (even when individual tree overfits)!
  - Lend themselves to parallel computation
  - Enable out-of-bag estimates of error

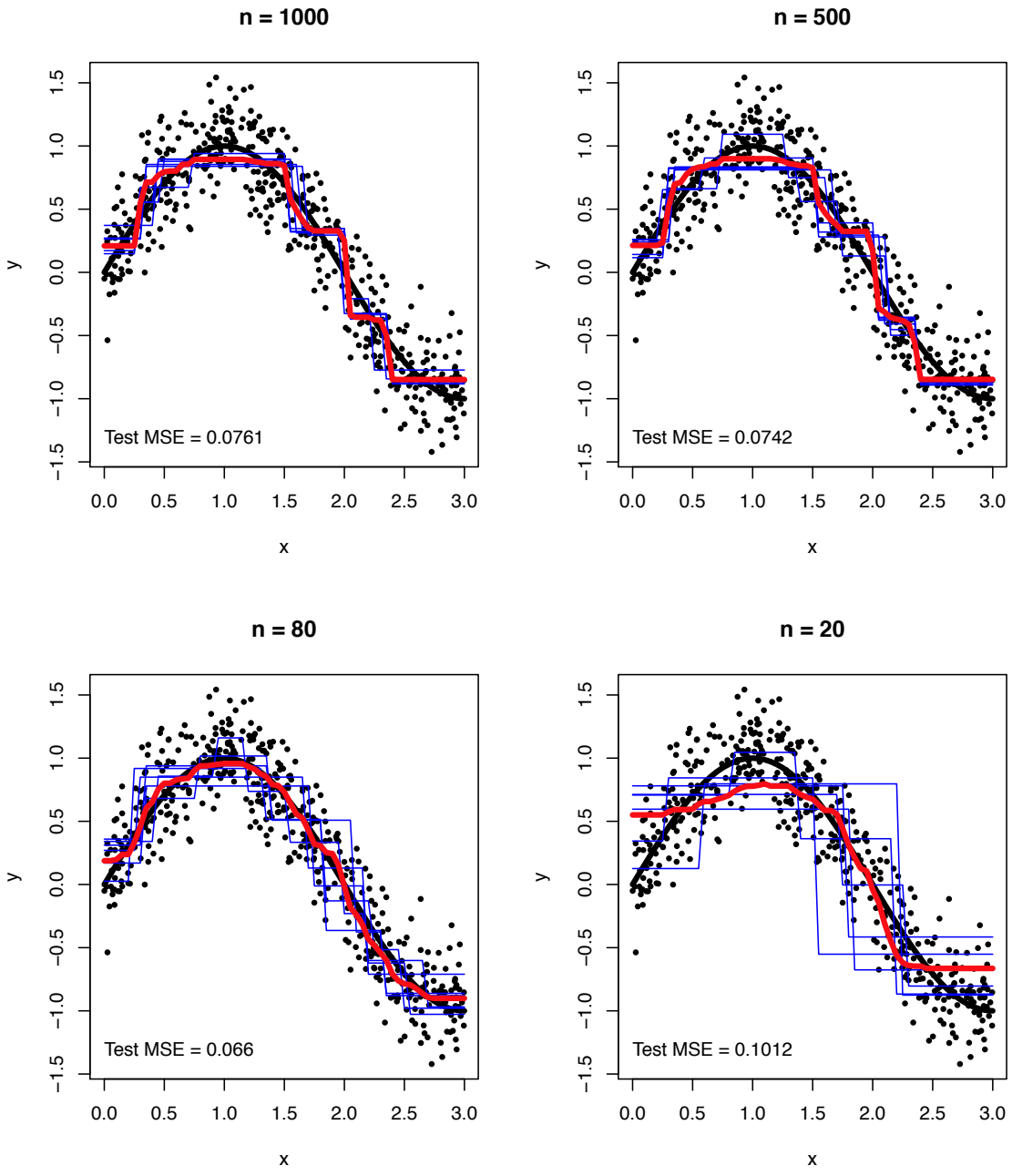
Bagged Tree Estimate of Wave Data

Estimate tree for $B = 20$ samples from wave data ($n = 100$), then evaluate function on an evenly-spaced grid of points.

```
n=100; nbag = 20
set.seed(1234567)
wave = data.frame(x = runif(n)*3)
wave$y = sin(wave$x*pi/2) + rnorm(n)/4
grid = seq(0,3,.05)
ans = double(length(grid))

plot(wave$x, wave$y, pch=16, xlab="x", ylab="y", cex=.7)
lines(grid, sin(grid*pi/2), lwd=4)
for(i in 1:nbag){
  train = sample(1:n, n, T)
  fit = tree(y ~ x, wave[train,])
  if(i<=6) lines(grid, predict(fit, data.frame(x=grid)), col=4)
  ans = ans + predict(fit, data.frame(x=grid))
}
ans = ans/nbag
lines(grid, ans, lwd=4, col=2)
```



Now Try Different Bootstrap Sample Sizes



Introduction to Random Forests

- Problem with simple bagging: the trees are sometimes too similar, i.e., the sow's ear is not sufficiently twitchy.
- Random forests extend this idea by introducing randomness in the process of building a tree. Only a random set of predictors are considered at any given split (reducing computational effort). If p is the number of predictors, then when considering each split use a random sample of predictors of size
  - $\sqrt{p}$ of them for classification trees
  - $p/3$ for regression trees
- The number of predictors considered at each split is a parameter, `mtry` in R. Setting `mtry` $= p$ gives bagging.

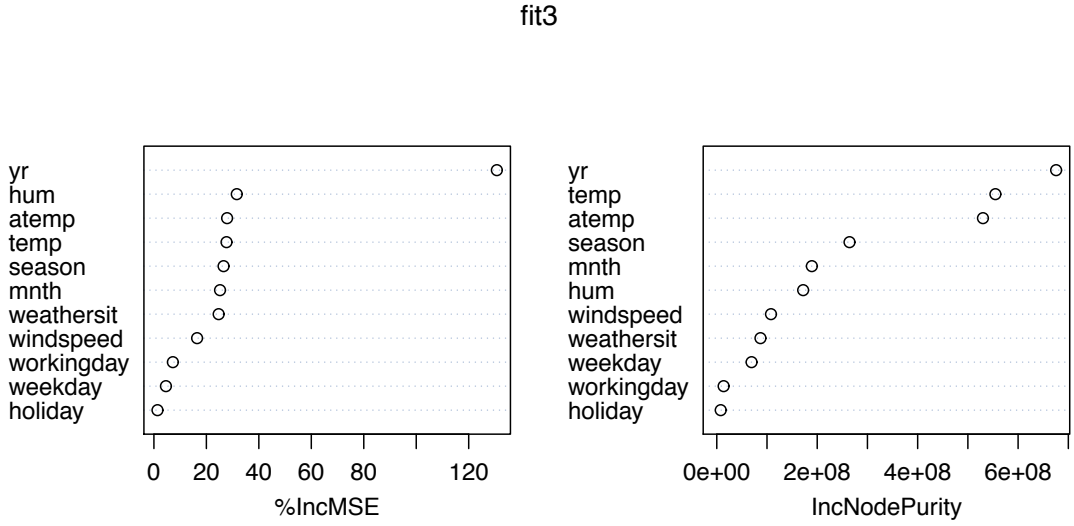
```
> fit = lm(count~season+yr+mnth+holiday+weekday+workingday+weathersit+temp+atemp+
hum+windspeed, bike)
> mean(fit$residuals^2)
[1] 645530.8
> summary(fit)$r.squared
[1] 0.8277507

> randomForest(count~season+yr+mnth+holiday+weekday+workingday+weathersit+
temp+atemp+hum+windspeed, bike, mtry=11) # bagging

                    Number of trees: 500
No. of variables tried at each split: 11

          Mean of squared residuals: 451414.4
                    % Var explained: 87.95

> randomForest(count~season+yr+mnth+holiday+weekday+workingday+weathersit+
temp+atemp+hum+windspeed, bike) # random forest
                    Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 448509.4
                    % Var explained: 88.03
```

Cross Validation and Variable Importance

- *Out-of-bag* (OOB) estimates of prediction error
  - Each tree in the forest is estimated from a random sample (of size n) drawn with replacement from the available data
  - Some observations will be omitted from the training sample
  - Use the omitted points as a test set

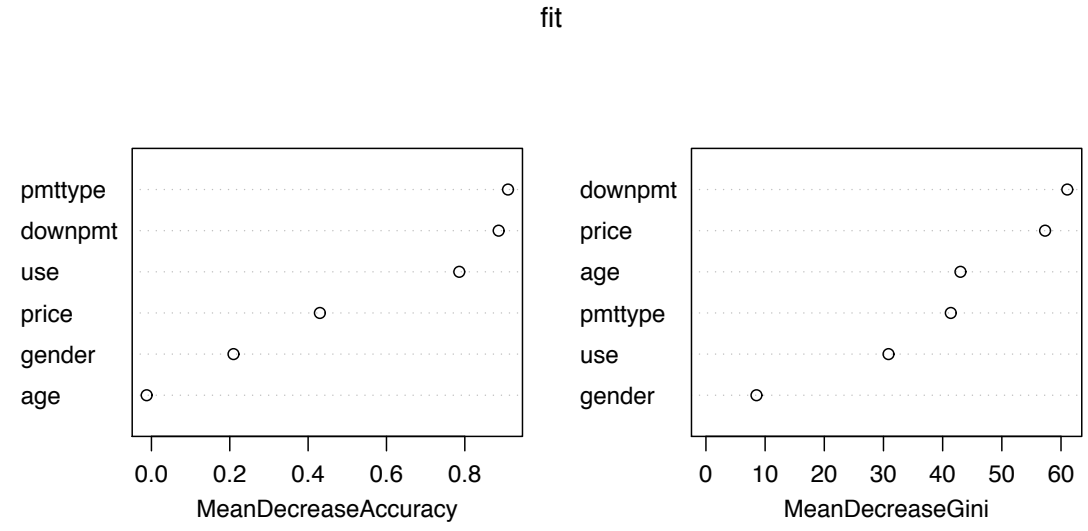
```
> set.seed(12345)
> sample(1:10, 10, replace=T)       # obs 1, 3 and 7 omitted
 [1]  8  9  8  9  5  2  4  6  8 10
> sample(1:10, 10, replace=T)       # obs 3, 6, 7 and 9 omitted
 [1]  1  2  8  1  4  5  4  5  2 10
> sample(1:10, 10, replace=T)       # obs 1, 2 and 9 omitted
 [1]  5  4 10  8  7  4  7  6  3  5
```

- Variable importance can be assessed by summing the change in impurity whenever a variable enters a split, akin to extra sum of squares. Use `importance=T` argument in R.
- The `ntree` argument sets the number of trees (default 500)



OOB and Variable Importance Example

```
> library(randomForest)
> set.seed(12345)
> ok = !is.na(default$use)&(runif(nrow(default))<.05)
> tiny = default[ok,]
> tiny$default = as.factor(tiny$default)
> randomForest(default~price+downpmt+pmttype+use+age+gender, tiny, importance=T)
Call:
 randomForest(formula = default ~ price + downpmt + pmttype +
use + age + gender, data = tiny, importance = T)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 10.02%
Confusion matrix:
     0  1 class.error
0 1074 34  0.03068592
1   93 66  0.58490566

> varImpPlot(fit)
```



Variable Importance Measure Caution

Variable importance measures are subject to the same issues with multicollinearity and the omitted variable bias as linear models. Interpret them with healthy skepticism!
Example, $n = 1000$, $x_1, x_3 \sim$ uniform$[0, 1]$, $x_2 \sim$ uniform$[-1, 1]$, $\text{cor}(x_1, x_2) = \text{cor}(x_1, x_3) = 0$, and $\text{cor}(x_2, x_3) = .95$

$$y = x_1^2 + 2x_2 + e,$$

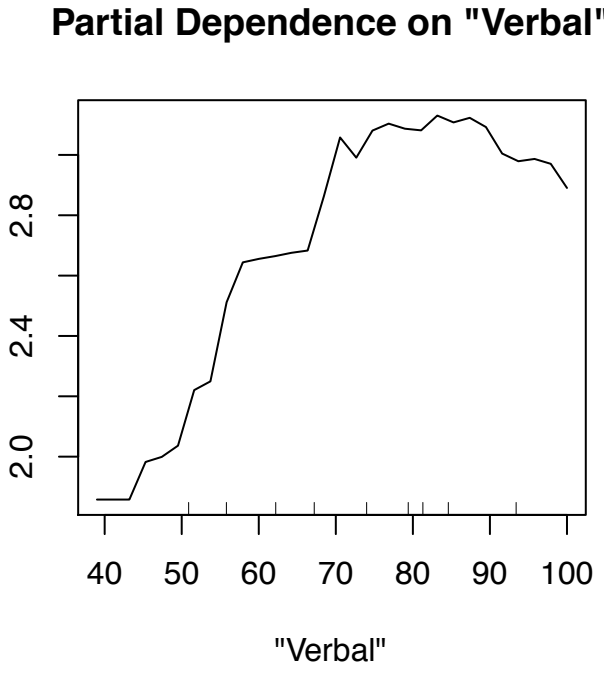
```
> library(randomForest)
> set.seed(12345)
> sigma = matrix(c(1,.95,.95,1), nrow=2)
> A = chol(sigma)
> Z = matrix(runif(2000), nrow=1000)
> dat = data.frame(Z %*% A)
> dat$X3 = dat$X1
> dat$X1 = 2*runif(1000)-1 # uncorrelated with X1, X2
> dat$y = dat$X1^2 + 2*dat$X2 + rnorm(1000)
> cor(dat)   # cor(X2, X3) = .95
           X1         X2         X3         y
X1 1.00000000 0.03825502 0.02974591 0.0425781
X2 0.03825502 1.00000000 0.94920846 0.4798642
X3 0.02974591 0.94920846 1.00000000 0.4481761
y  0.04257810 0.47986425 0.44817611 1.0000000

> importance(randomForest(y~X1+X2, dat))  #NOTE: X1 is more important without X2 in the model
   IncNodePurity
X1      502.1397
X2      719.3077

> importance(randomForest(y~., dat))     # with X3
   IncNodePurity
X1      357.0399
X2      449.2484
X3      417.7206
```

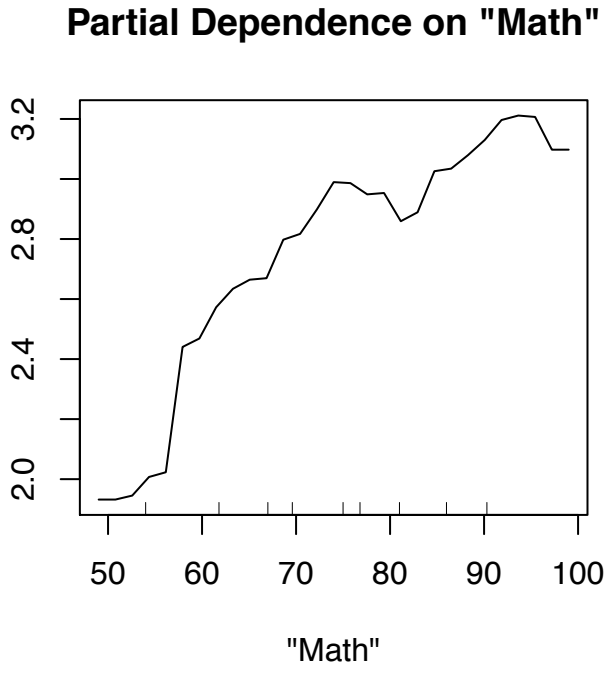
Test Scores Predicting GPA

```
library(gam)
> gpa = read.csv("prose/ajitbook/data/GPA.csv")
> fit = randomForest(GPA ~ Verbal+Math, data=gpa, importance=T)
> fit
Call:
 randomForest(formula = GPA ~ Verbal + Math, data = gpa, importance = T)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 1

          Mean of squared residuals: 0.1167213
                    % Var explained: 75.13
```

```
> par(mfrow=c(1,2))
> partialPlot(fit, gpa, "Verbal")
> partialPlot(fit, gpa, "Math")
```

```
> importance(fit)
        %IncMSE IncNodePurity
Verbal 29.33870      9.344022
Math   26.25216      7.813665
```

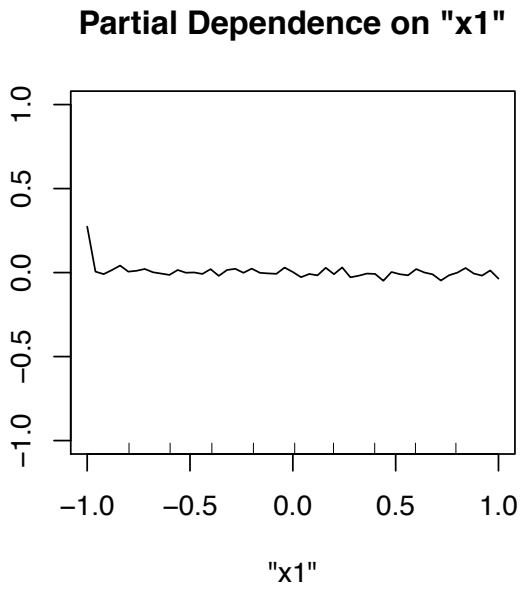


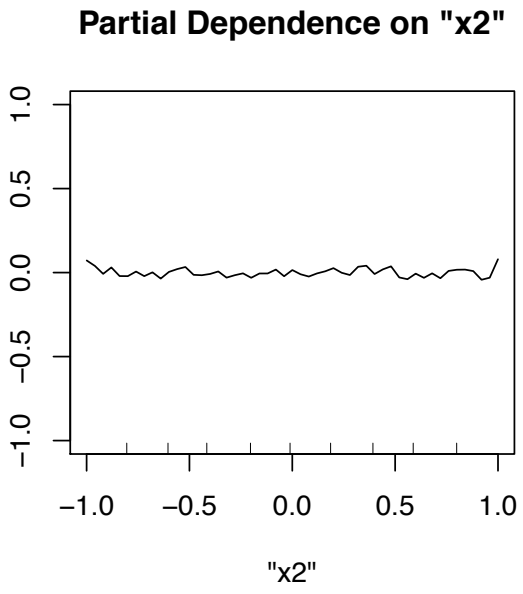


Saddle Function with Noise

Caution: partial dependence plots misleading when interactions are present!

```
> set.seed(12345)
> n=1000
> dat = data.frame(x1=runif(n)*2-1, x2=runif(n)*2-1)
> dat$y = dat$x1*dat$x2 + rnorm(n)/4
> fit = randomForest(y ~ ., dat, importance=T)
> fit       # note that MSE close to true value, 1/4^2 = .0625
 randomForest(formula = y ~ ., data = dat)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 1

          Mean of squared residuals: 0.06908491
                    % Var explained: 59.39

> importance(fit) # both variables equally important
   IncNodePurity
x1      719.7096
x2      893.2008

> par(mfrow=c(1,2))
> partialPlot(fit, dat, "x1", ylim=c(-1,1))
> partialPlot(fit, dat, "x2", ylim=c(-1,1))
```





Introduction to Boosted Trees

- Ideas:
  - Fit a series of simple trees. A tree with only one split is called a *stump* and one with a small number of splits is called a *shrub*
  - Each tree predicts residuals (what is left unexplained) thus far (so effort is spent where it is needed)
  - "Learn" slowly: use only a "sliver" of each stump/shrub
- Boosted Regression Tree Algorithm (see `gmb` library)
  1. Let $\hat{f}(x) = 0$ and $\hat{e}_i = y_i$ for all i in training set
  2. For $b = 1, \ldots, B$
     (a) Fit tree $\hat{f}^b$ with d splits predicting $\hat{e}$
     (b) Update estimate, where $0 < \lambda \leq 1$ is the *learning rate*:
     $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x)$$
     (c) Update residuals: $\hat{e}_i \leftarrow \hat{e}_i - \lambda \hat{f}^b(x)$
  3. The final model is
     $$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$$
- Usually pick $\lambda \in [.001, .01]$ (in R, `shrinkage`)
- $d = 1$ (stumps) implies an additive model (no interactions)

Tree Discusion

- Basic trees are
  - Fast, and easy to explain, interpret and implement
  - OK, but not great, predictors, and produce step functions (and thus not smooth)
- **All trees** are (1) robust to outlying x values, (2) invariant to monotonic transformations of x's, (3) good with missing values
- Random forests/bagged trees and boosted trees are (1) usually among the best predictors (especially for classification problems), (2) require little user intervention, and (3) fairly robust to overfitting
- **RF/bagged** trees lend themselves to parallel computation and RF are faster than bagging or boosting because fewer variables are searched at each split
- **Variable importance** measures do not account for multicollinearity. **Partial dependence plots** are misleading when there are interestions. Regard these measures as **exploratory** and compare with different appraoches, e.g., linear/logistic regression, stepwise, lasso, GAMs, etc.

Catalog Example

```
> set.seed(12345)
> train = runif(nrow(ascdata))<.2
> table(train)
train
FALSE  TRUE
45760 11449
>
> # Fit linear regression model
> fit = lm(logtarg ~ totdols+totfreq+recency+Ntotdols, ascdata[train,])
> yhat = predict(fit, newdata=ascdata[!train,])
> mean((yhat - ascdata$logtarg[!train])^2)
[1] 0.3465679
>
> # default tree model
> fit = tree(logtarg ~ totdols+totfreq+recency+Ntotdols, ascdata[train,])
> yhat = predict(fit, newdata=ascdata[!train,])
> mean((yhat - ascdata$logtarg[!train])^2)
[1] 0.3508628
>
> # default boosted tree
> fit = gbm(logtarg ~ totdols+totfreq+recency+Ntotdols, data=ascdata[train,])
Distribution not specified, assuming gaussian ...
> yhat = predict(fit, newdata=ascdata[!train,], n.trees=100)
> mean((yhat - ascdata$logtarg[!train])^2)
[1] 0.3634075
>
> # boosted tree increasing the number of trees
> fit = gbm(logtarg ~ totdols+totfreq+recency+Ntotdols, data=ascdata[train,], n.trees=1000)
Distribution not specified, assuming gaussian ...
> yhat = predict(fit, newdata=ascdata[!train,], n.trees=1000)
> mean((yhat - ascdata$logtarg[!train])^2)
[1] 0.3523262
>
> # boosted tree changing depth and learning rate
> fit = gbm(logtarg ~ totdols+totfreq+recency+Ntotdols, data=ascdata[train,], n.trees=5000,
interaction.depth=2, shrinkage=.01)
Distribution not specified, assuming gaussian ...
> yhat = predict(fit, newdata=ascdata[!train,], n.trees=5000)
> mean((yhat - ascdata$logtarg[!train])^2)
[1] 0.3489534
>
> # boosted tree changing depth only
> fit = gbm(logtarg ~ totdols+totfreq+recency+Ntotdols, data=ascdata[train,], n.trees=5000,
interaction.depth=1)
Distribution not specified, assuming gaussian ...
> yhat = predict(fit, newdata=ascdata[!train,], n.trees=5000)
```

```
> mean((yhat - ascdata$logtarg[!train])^2)
[1] 0.3440425
>
> # boosted tree default depth and learning rate
> fit = gbm(logtarg ~ totdols+totfreq+recency+Ntotdols, data=ascdata[train,], n.trees=5000,
interaction.depth=3)
Distribution not specified, assuming gaussian ...
> yhat = predict(fit, newdata=ascdata[!train,], n.trees=5000)
> mean((yhat - ascdata$logtarg[!train])^2)
[1] 0.343244
>
> # random forest, default options
> set.seed(12345)
> fit = randomForest(logtarg ~ totdols+totfreq+recency+Ntotdols, ascdata[train,], ntree=50)
> yhat = predict(fit, newdata=ascdata[!train,])
> mean((yhat - ascdata$logtarg[!train])^2)
[1] 0.3479717
>
> # random forest, mtry=2
> set.seed(12345)
> fit = randomForest(logtarg~totdols+totfreq+recency+Ntotdols, ascdata[train,], ntree=50, mtry=2)
> yhat = predict(fit, newdata=ascdata[!train,])
> mean((yhat - ascdata$logtarg[!train])^2)
[1] 0.3724772
>
>
> # random forest, mtry=4 (bagging)
> set.seed(12345)
> fit = randomForest(logtarg~totdols+totfreq+recency+Ntotdols, ascdata[train,], ntree=50, mtry=4)
> yhat = predict(fit, newdata=ascdata[!train,])
> mean((yhat - ascdata$logtarg[!train])^2)
[1] 0.3858764
```