Contents

The Concept of Causality

- **Causality**—A change in one variable will produce a change in (the distribution of) another variable.
- Three types of evidence to *infer* causality
  1. Concomitant variation/association between cause and effect: the extent to which the cause and effect occur together.
  2. Temporal precedence: cause *must* precede effect.
  3. Elimination of other possible causal factors, e.g., spurious association.
- *Pre-Experimental (observational) study*: subjects assign themselves to different groups (if any)—data scientist as little control over when and to whom treatment is given
- *True-experimental (*randomized controlled*) study*: data scientist assigns subjects to treatment and control groups at random
- *Quasi-experimental designs*: the data scientist is unable to achieve complete control over the scheduling of the treatments or cannot randomly assign respondents to experimental treatment conditions, but has stronger controls than with basic observational studies.

Nature of Experimentation

- **Experimentation**—The manipulation of one or more variables (cause) by the experimenter in such a way that its effect on one or more other variables can be measured
- A **treatment** is something that data scientists administer to experimental units.
- A **factor** is a controlled **independent variable**; a variable whose **levels** are set by the experimenter.
- **Dependent (or criterion) variables**—(Y's) Variables that will reflect the impact of the independent variable
- **Treatment group** (TG)—Portion of sample exposed to the treatment
- **Control group** (CG)—Portion of the sample not exposed to the treatment

Notation to Represent Experimental Designs

- X—exposure of an individual, group, or other entity to the experimental treatment
- O—observation or measurement of the test unit
- R—randomized assignment

Movement through time is represented by a horizontal arrangement of X's and O's from left to right. Simultaneous exposure or measurement by a vertical arrangement.

Your Turn: A new manufacturer of women's cosmetics was planning to retail the firm's products through mail order. The firm's management was considering the use of direct mail ads to stimulate sales of their products. Prior to committing themselves to advertising through direct mail, management conducted an experiment. A random sample of 1000 housewives was selected from Memphis, Tenn. The sample was divided into two groups with each prospect being assigned randomly to one of the two groups. Direct mail ads were sent twice over a period of one month to prospects of one of the groups. Two weeks later, both the groups were mailed the company's catalog of cosmetics. Sales to each group were monitored.

What makes a good test?

- Factors other than the experimental variable that affect the dependent variable are called *nuisance*, *confounding*, or *extraneous* factors/variables.
- When nuisance factors are not properly controlled, they are said to *confound* the effects of the experimental variable.
- **Internal validity** — The ability of the experiment to *unambiguously show a cause and effect relationship*, i.e., to what extent can we attribute the effect that was observed to the experimental variable and not other (confounding) factors? See THIS MESS.
- **External validity** — The extent to which the results of the experiment can be *generalized* to other people, settings (e.g., geography), and time (e.g., seasonality)

There is often a managerial trade-off between internal and external validity

Pre-Experimental Designs

- *After-only design (one-shot case study)*

$$X \qquad\qquad O$$

  - Provides no basis for comparing what happened in the presence of X with what happened when X was absent. Cannot infer causality.
  - Serious threats to IV: history, maturation, selection, mortality
  - Example (*Devliery system*). You are considering a new delivery system and wish to test whether delivery times are significantly different, on average, than your current system. It is well established that the mean delivery time of the current system is 2.38 days. A test of the new delivery system shows that, with 48 observations, the average delivery time is 1.91 days with a standard deviation of 0.43 days.

Pre-Experimental Designs

- *One-group Pretest-Posttest Design (Before-after)*

$$O_1 \qquad X \qquad O_2$$

  - The result of interest: $\hat{D} = O_2 - O_1$
  - Analysis: paired-sample t test
  - Threats to IV: history, maturation, pre-measurement, placebo effect.
  - Example: *"Heavy-up" advertising.* Sales of a product are monitored for one week (*pre-measure* or *historical control*). The advertising budget is doubled (*treatment*). Sales are monitored for one week after the advertising increase (*post-measure*).
  - Example: *SAT training* 20 high school juniors were recruited. Each took the SAT and then completed a 4-week SAT training course every Saturday. Each student then took the SAT again.
  - Example: *Speed reading.* Subjects read a text and took a comprehension test. Then they took a speed reading crouse. After the course they read another text and took comprehension test. Did speed-reading training decrease comprehension?

Threats to Internal Validity

- *Interaction (or interactive testing) effect* — When a pre-measure changes the respondent's sensitivity or responsiveness to the independent variable(s). This is only a threat to external validity.
  - In SAT training, perhaps people pay more attention to certain training modules because they were asked about it during the pre-measure.
- **Placebo effect** — respondents acts differently because they know that they are being exposed to the treatment
  - In the early 1960s a study was carried out to investigate the efficacy of a technique known as gastric freezing to treat ulcer patients. The treatment required patients to swallow a balloon, which was positioned in the stomach. A coolant was then passed through the balloon. It was reported that the patients given this treatment experience relief of the symptoms, and the treatment was recommended for ulcer patients. Later, a randomized controlled experiment was conducted, where the control group also swallowed a balloon without the coolant. There was no difference between the treatment and control groups.
  - If both the subject and the investigator are kept unaware of the treatment, the study is called a **double blind** study.

Pre-Experimental Designs

- **Static-Group Comparison**

TG:	X	O_1
CG:		O_2

  - The result of interest: $\hat{D} = O_1 - O_2$
  - Analysis: independent-sample t test or GLM/ANCOVA
  - Threats to IV: **selection**, maturation / mortality (if treatment unpleasant),
  - Example: *Department store patronage.* Two groups of respondents are recruited *on the basis of convenience*, e.g., one group in the morning and the other in the afternoon. One group is shown a TV commercial about a department store. Both groups are asked about their attitudes toward the store.
  - Example: *Magazine experiences.* Experiences such as “it’s my personal timeout” were measured for readers of 100 magazines. Respondents were shown an ad for bottled water (*treatment*) and asked standard measures of their attitude towards the ad (*post-measure*). After *controlling* for bottled water consumption and attitude towards ads in general (through a regression analysis), we found a highly significant positive relationship between “it’s my personal timeout” and the attitude towards the water ad across magazines.

Threats to Internal Validity

- **Experimental mortality** — Differential loss of respondents from different groups.
- **Selection bias** — When the groups formed for the purposes of the experiment are initially unequal with respect to the dependent variable or in the propensity to respond to the independent variable.

  Remedies:

  - **Randomization** — assign subjects to treatment and control group using a random procedure.
  - **Matching** — match treatment and control groups with respect to variables which you suspect influence response (used with small sample sizes, e.g., stores).
    - Form *blocks* of units that are similar on key nuisance factors
    - *Randomly* assign units within a block to treatment and control groups
    - Blocking is closely related to stratification. Blocking is used in experiments and stratification in surveys.
  - Control for other causal factors (forks) with regression.
  - *Propensity score models* for observational studies. Find matched "twin(s)" for each treated case that is as similar as possible prior to self-selection into treatment

Threats to Internal Validity

- **Statistical regression**—When individuals are assigned to groups because of their scores on some measurement, such as initial attitude towards a brand. (Also called the *regression effect*)
  - Every year, baseball's major leagues honor their outstanding first-year players with the title "Rookie of the Year." From 1949 to 1987, the overall batting average for the Rookies of the Year was 0.285, far above the major league average of 0.257. However, Rookies of the Year don't do so well in their second year: their overall second-season batting average was on 0.272. Baseball writers call this "sophomore slump," their explanation being that star players get distracted by outside activities like product endorsements and television appearances. Do you agree?

Collider bias

- Suppose there are two diseases that are (mostly) unrelated in the general population, e.g., bone disease and respiratory disease. If we do a study of people in the hospital, it shows a positive association. How can this be?

- Sackett data:

	General population			Hospitalized		
Respiratory disease	Bone disease			Bone disease		
	Yes	No	% Yes	Yes	No	% Yes
Yes	17	107	7.6	5	15	25.0
No	184	2,376	7.2	18	219	7.6

- See Berkson's paradox and Why are handsome men such jerks?

- The underlying cause is that you have selected your sample based on the outcome

Threats to External Validity

All the previous threats to internal validity are also threats to external validity. In addition, there are the following:

- **Surrogate situation** — Occurs when the environment, the population sampled, and/or the treatments are different from those that will be encountered in the actual situation, e.g., copy testing . . . forced attention.
- **Measurement timing** — When pre- or post-measurements are made at an inappropriate time to indicate the effect of the experimental treatment, e.g., effect of temporary price cut on forward buying.

True Experimental Designs

- *After-only with control group* sometimes called a **completely randomized design**

TG:	(R)	X	O_1
CG:	(R)		O_2

  - The result of interest: $\hat{D} = O_1 - O_2$
  - Analysis: independent-sample t test
  - Threats to IV: none
  - Large between-unit variation implies that larger samples will be required to detect differences compared with the before-after with control design

- **Before-after with control group**

TG:	(R)	O_1	X	O_2
CG:	(R)	O_3		O_4

  - The result of interest: $\hat{D} = (O_2 - O_1) - (O_4 - O_3)$
  - Analysis: Compute differences between post- and pre-measures and compare with independent sample t-test.
  - Threats to IV: Interactive testing effect

Single-Factor Experiments

- Now suppose that there is one **factor** variable with a **levels**
- The experimenter **randomly** assigns units to treatments levels and observes **response variable** y_{ij}, where $j \in \{1, \ldots, a\}$ is the level and $i = 1, \ldots n_j$ indexes units.
  - When the number of observations is the same for all treatments (i.e., $n_1 = \cdots = n_a$) the design is **balanced**,
  - Otherwise it is said to be **unbalanced**
- Without random assignment, beware of selection biases!
- Model
$$y_{ij} = \mu_j + \epsilon_{ij},$$
where μ_j is the mean response for level j and $\epsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$.
- Example: We must purchase a specific part and are considering three suppliers, 1=Amalgamated, 2=Bipolar and 3=Consolidated. We purchase a random sample of parts from each of the three suppliers ($n_1 = 18$, $n_2 = 21$ and $n_3 = 19$) and measure the quality of the parts. Identify all the terms and symbols discussed above.

Steps to analyze a single-factor CRD

1. Generate boxplots to compare the distributions — look for:

   - Equal variance across groups
   - Outliers, anomalies, skewed distributions

   For small samples it is important that the distributions for each group be normal

2. Fit ANOVA or (equivalently) linear model, check diagnostics

3. Perform overall "F" test

   $H_0 : \mu_1 = \mu_2 = \cdots = \mu_p$

   H_1 : At least one mean is different

   If you cannot reject H_0, stop; you are not allowed to do step 4.

4. If you reject H_0 in step 3, do **pairwise comparisons** of means

   $H_0 : \mu_1 = \mu_2$ against $H_1 : \mu_1 \neq \mu_2$

   $H_0 : \mu_1 = \mu_3$ against $H_1 : \mu_1 \neq \mu_3$

   $H_0 : \mu_2 = \mu_3$ against $H_1 : \mu_2 \neq \mu_3$

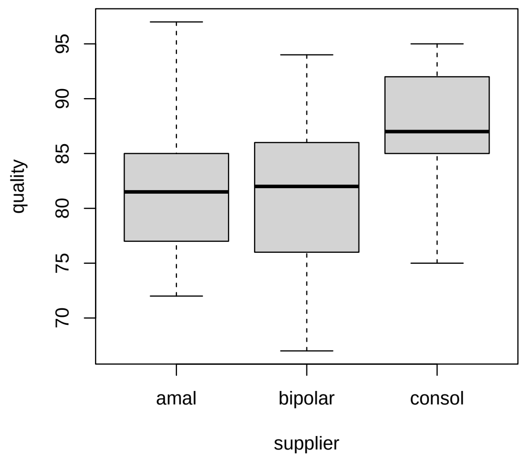
   $\vdots$

   There are $\binom{a}{2}$ comparisons and the F test protects against type-I errors. There are additional ways to adjust P-values for the number of tests performed (multiple comparison problem)

ABC Supplier Problem

```
abc = data.frame(
  supplier=factor(c(rep(1,18), rep(2,21), rep(3,19)),
    labels=c("amal", "bipolar", "consol")),
  quality=c(75,72,87,77,84,82,84,81,78,97,85,81,95,81,72,89,84,73,
  94,87,80,86,80,67,86,82,86,82,72,77,87,68,80,76,68,86,74,86,90,
  90,86,92,75,79,94,95,85,86,92,92,85,87,86,92,85,93,89,83)
)
```

```
> boxplot(quality ~ supplier, abc)
> # aov is a version of lm
> fit = aov(quality ~ supplier, abc)
> plot(fit)  # residuals, QQ, etc.

> library(dplyr) # descriptive stats
> abc %>%
+   group_by(supplier) %>%
+   summarise(n=n(),
  xbar=mean(quality), sd=sd(quality))
# A tibble: 3 x 4
  supplier     n  xbar    sd
  <fct>    <int> <dbl> <dbl>
1 amal        18  82.1  7.12
2 bipolar     21  80.7  7.60
3 consol      19  87.7  5.23
```



```
> summary(fit)
          Df  Sum Sq Mean Sq F value   Pr(>F)
supplier   2  538.16 269.081  5.8969 0.004783**
Residuals 55 2509.72  45.631
> TukeyHSD(fit)
 Tukey multiple comparisons of means
   95% family-wise confidence level

Fit: aov(quality~supplier, data=abc)
$supplier
                diff   lwr   upr  p adj
bipolar-amal   -1.39 -6.62  3.84 0.7987
consol-amal     5.63  0.28 10.98 0.0372
consol-bipolar  7.02  1.87 12.17 0.0050

> # lm equivalent, but TukeyHSD doesn’t work
> fit2 = lm(quality ~ supplier, abc)
> anova(fit2) # same as summary(fit)
> summary(fit2)

Coefficients:
                          Std.     t
               Estimate Error value  Pr(>|t|)
(Intercept)      82.056 1.592 51.536 <2e-16 ***
supplierbipolar  -1.389 2.170 -0.640 0.5248
supplierconsol    5.629 2.222  2.533 0.0142 *

Residual std error: 6.755 on 55 cv
Multiple R-squared:  0.1766
F-statistic: 5.897 on 2 and 55DF, P=0.004783
```

Randomized complete block design

- Sometimes you have measures of other characteristics of the experimental units that are expected to be related to the response. They are called **nuisance factors**.
- **Randomized complete block design** (RCBD):
  - Form **blocks** of units that are similar on nuisance factor
  - Randomly assign treatments within each block

  "Block what you can; randomize what you can't"
- When between-unit variation in response is large, blocks can reduce sample size requirements
- Let Y_{jk} be a measure of treatment level j from block k (assume one observation per level × block combination). Model:

  $$Y_{jk} = \mu + \tau_j + \beta_k + \epsilon_{jk},$$

  where $\epsilon_{jk} \sim \text{NID}(0, \sigma^2)$, τ_j is the effect of treatment j ($\sum_i \tau_i = 0$), and β_k is the effect of block k ($\sum_i \beta_i = 0$)
- Example (delivery system): block on package destination, then randomly assign to new or old system within blocks
- Example: What effect do different promotions have on the probability purchasing an airline ticket among members of frequent flier? What do you know about frequent fliers that should be blocked?

Restaurant example

Restaurant example: Six judges evaluate four restaurants. A rating scale from 0 (low) to 100 (high) is used.

```
> dat = expand.grid(judge=factor(1:6),
    rest=LETTERS[1:4])
> dat$rate = c(70,77,76,80,84,78,
             61,75,67,63,66,68,
             82,88,90,96,92,98,
             74,76,80,76,84,86)
> fit = aov(rate~rest+judge, dat)
> summary(fit)
                 Mean      F
      Df Sum Sq   Sq    value  Pr(>F)
rest   3 1787.5 595.8 39.758 2.2e-07***
judge  5  283.4  56.7  3.782  0.0205*
Resid 15  224.8  15.0

> fit2 = lm(rate~rest+judge, dat)
> anova(fit2) # same as summary(fit)
> summary(fit2)

Coefficients:
                  Std.     t
      Estimate  Error   value Pr(>|t|)
(Int)   70.625  2.371   29.79 9.21e-15***
restB  -10.833  2.235   -4.85 0.000213***
restC   13.500  2.235    6.04 2.26e-05***
restD    1.833  2.235    0.82 0.424905
judge2   7.250  2.737    2.65 0.018244*
judge3   6.500  2.737    2.38 0.031343*
judge4   7.000  2.737    2.56 0.021882*
judge5   9.750  2.737    3.56 0.002839**
judge6  10.750  2.737    3.93 0.001345**

Resid std error: 3.871 on 15 df
Multiple R-squared:  0.9021
F-statistic: 17.27 on 8 and 15 DF
p-value: 2.912e-06

> TukeyHSD(fit)
  Tukey multiple comparisons of means
    95% family-wise confidence level
```

```
Fit: aov(rate ~ rest + judge, data = dat)

$rest
       diff        lwr       upr     p adj
B-A -10.833 -17.275029 -4.391637 0.0010921
C-A  13.500   7.058304 19.941696 0.0001197
D-A   1.833  -4.608363  8.275029 0.8440099
C-B  24.333  17.891637 30.775029 0.0000001
D-B  12.667   6.224971 19.108363 0.0002346
D-C -11.667 -18.108363 -5.224971 0.0005379

$judge
     diff        lwr       upr     p adj
2-1  7.25 -1.6435455 16.143545 0.1449898
3-1  6.50 -2.3935455 15.393545 0.2256956
4-1  7.00 -1.8935455 15.893545 0.1686825
5-1  9.75  0.8564545 18.643545 0.0278360
6-1 10.75  1.8564545 19.643545 0.0138765
3-2 -0.75 -9.6435455  8.143545 0.9997425
4-2 -0.25 -9.1435455  8.643545 0.9999989
5-2  2.50 -6.3935455 11.393545 0.9370498
6-2  3.50 -5.3935455 12.393545 0.7917324
4-3  0.50 -8.3935455  9.393545 0.9999650
5-3  3.25 -5.6435455 12.143545 0.8361275
6-3  4.25 -4.6435455 13.143545 0.6386431
5-4  2.75 -6.1435455 11.643545 0.9093061
6-4  3.75 -5.1435455 12.643545 0.7432852
6-5  1.00 -7.8935455  9.893545 0.9989599

> # What happens if you don't include
> # the blocking variable?
> fit3 = aov(rate ~ judge, dat)
> summary(fit3)
            Df Sum Sq Mean Sq F value Pr(>F)
judge        5  283.4   56.67   0.507  0.767
Residuals   18 2012.2  111.79
```

Fabric example

An experiment was performed to assess the effect of four chemicals on the strength of fabric. Five fabric samples were selected and a RCBD was run by teaching each chemical once in random order on each fabric sample.

```
> dat = expand.grid(chem=LETTERS[1:4],
      fabric=letters[1:5])
> dat$strength = c(1.3, 2.2, 1.8, 3.9,
        1.6, 2.4, 1.7, 4.4,
        0.5, 0.4, 0.6, 2.0,
        1.2, 2.0, 1.5, 4.1,
        1.1, 1.8, 1.3, 3.4)
> fit = aov(strength ~ chem + fabric, dat)
> summary(fit)
       Df Sum Sq   Mean     F
       Df Sum Sq     Sq value   Pr(>F)
chem    3 18.044  6.015 75.89 4.52e-08***
fabric  4  6.693  1.673 21.11 2.32e-05***
Resid  12  0.951  0.079
> fit2 = lm(strength ~ chem + fabric, dat)
> summary(fit2)
Coefficients:
                    Std.      t
        Estimate   Error  value Pr(>|t|)
(Int) ‘   1.4800  0.1780  8.313 2.53e-06***
chemB     0.6200  0.1780  3.482  0.00453**
chemC     0.2400  0.1780  1.348  0.20256
chemD     2.4200  0.1780 13.592 1.19e-08***
fabricb   0.2250  0.1991  1.130  0.28043
fabricc  -1.4250  0.1991 -7.159 1.15e-05***
fabricd  -0.1000  0.1991 -0.502  0.62451
fabrice  -0.4000  0.1991 -2.009  0.06753.

Residual standard error: 0.2815 on 12 df
Multiple R-squared:  0.963
F-statistic: 44.59 on 7 and 12 DF
p-value: 1.184e-07

> TukeyHSD(fit)
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(strength ~ chem+fabric, data=dat)

$chem
     diff         lwr       upr     p adj
B-A  0.62  0.09140218 1.1485978 0.0204200
C-A  0.24 -0.28859782 0.7685978 0.5523215
D-A  2.42  1.89140218 2.9485978 0.0000001
C-B -0.38 -0.90859782 0.1485978 0.1973362
D-B  1.80  1.27140218 2.3285978 0.0000017
D-C  2.18  1.65140218 2.7085978 0.0000002

$fabric
      diff        lwr          upr     p adj
b-a  0.225 -0.4094912  0.859491196 0.7881457
c-a -1.425 -2.0594912 -0.790508804 0.0000922
d-a -0.100 -0.7344912  0.534491196 0.9855599
e-a -0.400 -1.0344912  0.234491196 0.3180795
c-b -1.650 -2.2844912 -1.015508804 0.0000212
d-b -0.325 -0.9594912  0.309491196 0.5059326
e-b -0.625 -1.2594912  0.009491196 0.0542123
d-c  1.325  0.6905088  1.959491196 0.0001857
e-c  1.025  0.3905088  1.659491196 0.0018293
e-d -0.300 -0.9344912  0.334491196 0.5771595

> fit3 = aov(strength ~ chem, dat)
> summary(fit3)
        Df Sum Sq Mean Sq F value   Pr(>F)
chem     3 18.044   6.015   12.59 0.000176 ***
Resid   16  7.644   0.478
```

Two-Factor Experiments

- Definition: an experimental design in which two variables are being manipulated.
- A company that prepares students for the ACT college entrance exam is testing new curriculum. They want to investigate 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, and then took the test.
  - A specific modality or course length is called a **factor level** and the combination of a modality and length is a **treatment**
  - **Factorial experiment**: all level combinations studied
  - When each combination has the same number of *replicates* the design is said to be *balanced* or *orthogonal*; otherwise the design is said to be *unbalanced.* Balanced designs are desirable because they avoid multicollinearity
  - Let y_{ijk} be the observed response to subject $i = 1, \ldots, n$ receiving modality j and length k.

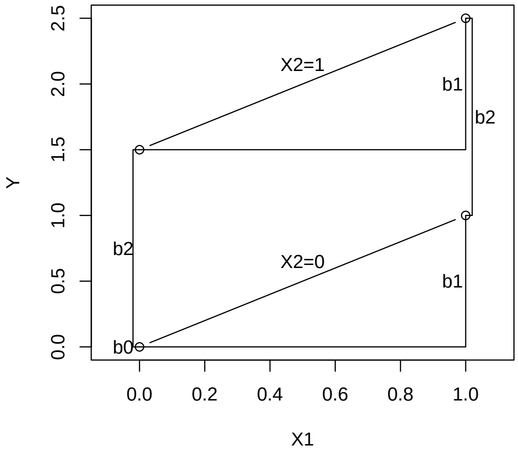
Course	Modality Levels	
length	in-person	distance
short	$y_{111}, \ldots, y_{11n}$	$y_{121}, \ldots, y_{12n}$
long	$y_{211}, \ldots, y_{21n}$	$y_{221}, \ldots, y_{22n}$

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

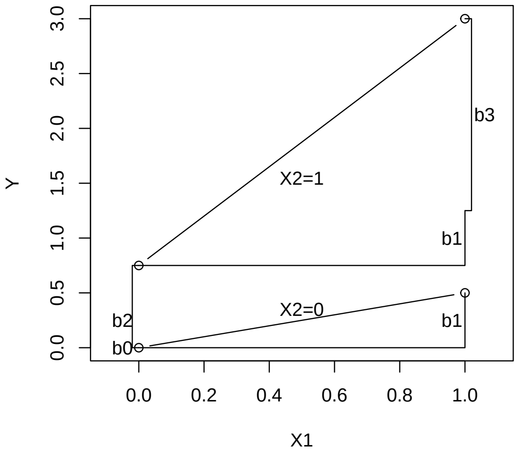
$$\hat{y} = b_0 + x_1 b_1 + x_2 b_2$$



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.

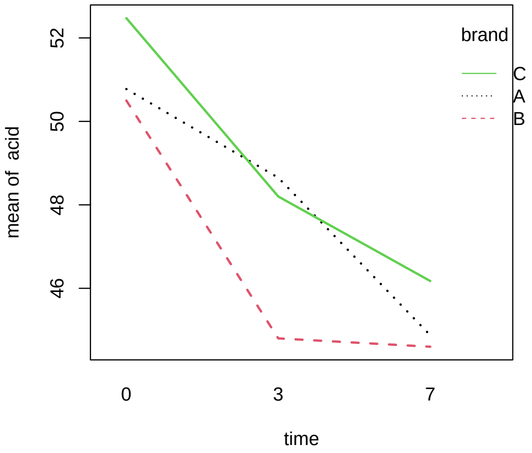
ACT Example

A company that prepares students for the ACT college entrance exam is testing new curriculum. They want to investigate 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)

> library(dplyr)
> course %>%
  group_by(type, length) %>%
  summarize(n=n(), mean=mean(act))

  type   length        n  mean
1 Trad   Condensed    10  21.9
2 Trad   Regular      10  27.9
3 Online Condensed    10  27
4 Online Regular      10  21.3

> fit$coef
             (Intercept)               typeOnline
                    21.9                      5.1
           lengthRegular typeOnline:lengthRegular
                     6.0                    -11.7
```

Orange Juice Example

To ascertain the stability of vitamin 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))
> with(oj, interaction.plot(time, brand, acid, col=1:3, lwd=2))
> 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>                  254.14 88.357
time:brand  4    17.301 271.44 82.728  0.4595 0.7647

> 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>              271.44  82.728
time    2   226.676 498.12 100.583 12.9438 8.191e-05 ***
brand   2    32.962 304.40  82.854  1.8822    0.1692
```

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)
> detach(capacitor)
```

Your Turn

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 whether 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))
   ```

Solutions

1. Paint problem

   ```
   fit = lm(y~type*time, paint)
   summary(fit)
   drop1(fit, test="F")
   summary(fit)
   with(paint, interaction.plot(type, time, y, col=1:3))
   with(paint, interaction.plot(time, type, y, col=1:2))
   ```

2. anode problem

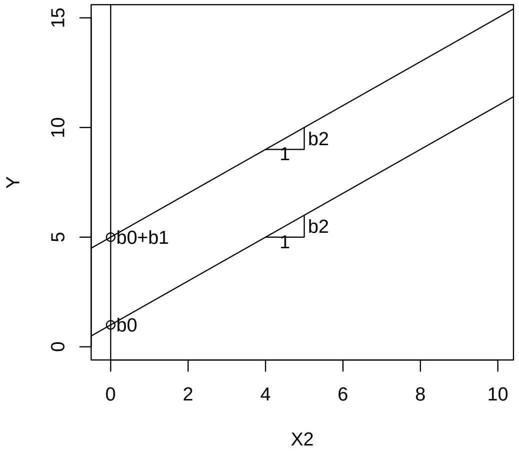
   ```
   fit = lm(density ~ pos*temp, anode)
   drop1(fit, test="F")
   summary(fit)
   with(anode, interaction.plot(pos, temp, density, col=1:3))
   with(anode, interaction.plot(temp, pos, density, col=1:2))
   fit = lm(density ~ pos+temp, anode)
   drop1(fit, test="F")
   summary(fit)
   ```

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 ($x_1 = 0$)

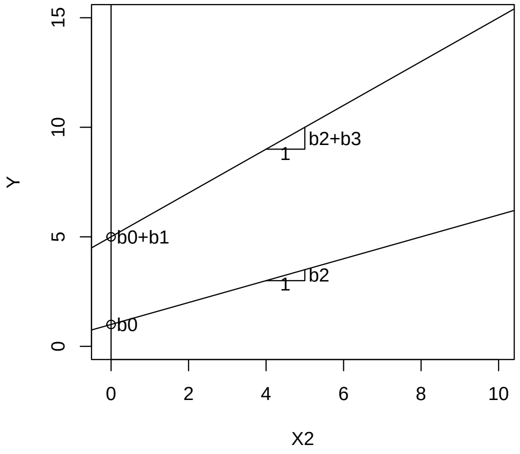
$$y = b_0 + b_2 x_2$$

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
```

Quasi-Experimental Designs

- **Quasi-Experimental Designs**: the data scientist is unable to achieve complete control over the scheduling of the treatments or cannot randomly assign respondents to experimental treatment conditions.
- **Before-After with Control Quasi Design**

TG:	O_1	X	O_2
CG:	O_3		O_4

  - The result of interest: $\hat{D} = (O_2 - O_1) - (O_4 - O_3)$
  - Analysis: Indepdendent-sample t test on differences
  - Somtimes, treatment and control *matched*: units assigned to treatment and control based on key factors.
  - Threats of validity: selection, interactive testing effects
  - See Differences in differences

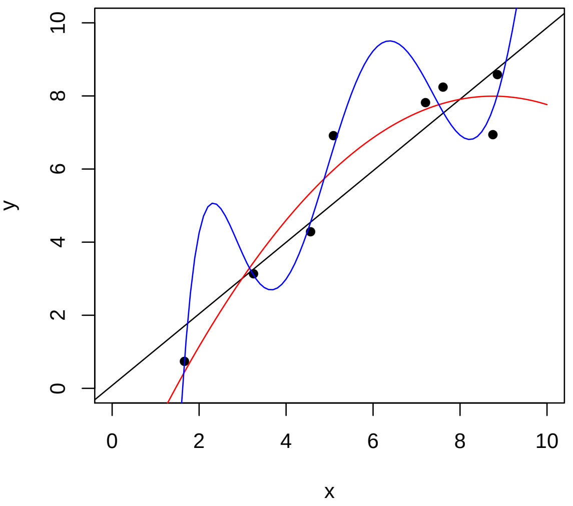
How to Measure Predictive Accuracy

- 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[1] with, e.g.,

$$\text{SSE} = \sum_{i=1}^{n} [y_i - f(\mathbf{x}_i)]^2, \quad \text{MSE} = \frac{\text{SSE}}{n}, \text{ or } \quad R^2 = 1 - \frac{\text{SSE}}{\text{SST}}$$

- 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
```



[1] Until now, MSE $=$ SSE$/(n-p-1)$, which is unbiased. The straight average, MSE $=$ SSE$/n$, is more commonly used when computing out-of-sample estimates.

Two Approaches for Honest Estimates of Prediction Error

- **Penalized estimates**, e.g.,

$$R_a^2 = 1 - \frac{\text{SSE}/(n-p-1)}{\text{SST}/(n-1)} \quad \text{or} \quad \text{AIC} = \text{deviance} + 2p$$

  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)

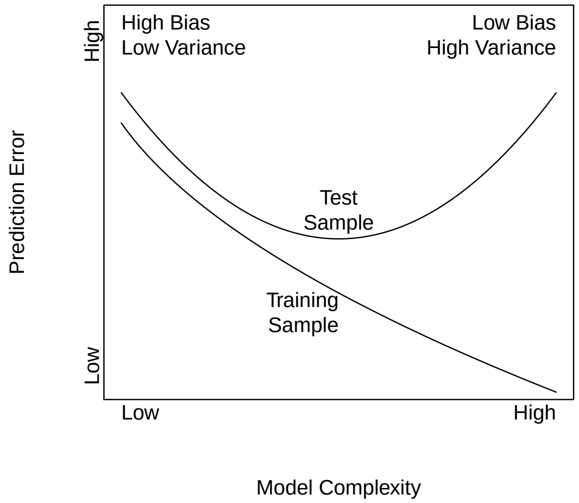
```
> train$x2 = runif(8)
> train$x3 = runif(8)
> test$x2 = runif(10000)
> test$x3 = runif(10000)

> fit4 = lm(y ~ x+x2+x3, train)
> summary(fit4)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2604     1.8061   0.144  0.89232
x             0.9881     0.1746   5.658  0.00481 **
x2            1.1695     1.8216   0.642  0.55580
x3           -2.8721     2.2520  -1.275  0.27123
> 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 (see Wikipedia for variations)
  - 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.)"

Good practice (at least when you have sufficient data) is have a three-way split

- **Training data**: used to estimate model parameters
- **Validation data**: used to select model hyper parameters (or use K-fold cross validation if you are data poor)
- **Test data**: used for final, inter-model comparisons

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 MSE from all data
[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 (SSE and deviance) — Adding predictors will . . .
  - always improve SSE on *estimation* sample
  - not necessarily improve SSE on *validation* data — SSE 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 SSE
  3. Repeat (2) until no significant improvement in SSE
- **Backward elimination**
  1. Begin with all candidate variables
  2. Drop variable that causes smallest non-significant increase in SSE
  3. Repeat (2) until dropping variable causes significant increase in SSE
- **Stepwise selection**
  1. (Usually) begin with no variables
  2. Drop variable that causes smallest *non-significant* increase in SSE
  3. Add variable that yields greatest significant improvement in SSE
  4. Repeat (2) and (3) until no improvement in SSE

Notes:

- These can be very slow compared with ridge/lasso
- Instead of significance tests, R uses AIC
- Treat these methods as exploratory

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
```

Shrinkage Estimation

- Consider estimate $\hat{\beta}$ of $\boldsymbol{\beta}$
- The **bias** of $\hat{\beta}$ is
$$\text{bias}(\hat{\beta}) = \mathbb{E}(\hat{\beta}) - \boldsymbol{\beta}$$
When $\mathbb{E}(\hat{\beta}) = \boldsymbol{\beta}$, we say that $\hat{\beta}$ is an **unbiased** estimate of $\boldsymbol{\beta}$.
- The **mean-squared error** of $\hat{\beta}$ is
$$\begin{aligned} \text{MSE}(\hat{\beta}) &= E[(\hat{\beta} - \boldsymbol{\beta})^{\mathsf{T}}(\hat{\beta} - \boldsymbol{\beta})] \\ &= \text{trace}[\mathbb{V}(\hat{\beta})] + \text{bias}(\hat{\beta})^{\mathsf{T}}\text{bias}(\hat{\beta}) \\ &= \text{variance} + \text{bias}^2 \end{aligned}$$
- The OLS estimate $\hat{\beta}$ is unbiased and therefore has
$$\text{MSE}(\hat{\beta}) = \sigma^2(\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}$$
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
- My lecture on bias and variance with math.
- My lecture on MSE with math

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

$$\hat{\beta}_\lambda = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left[\sum_{i=1}^{k} (y_i - \beta_0 - \mathbf{x}_i^\mathsf{T} \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right] \qquad (3.41)$$

$$\hat{\beta}_\lambda = (\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\mathsf{T} \mathbf{y} \qquad (3.44)$$

- $\lambda \geq 0$ 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 $\hat{\beta}_\lambda$ 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).

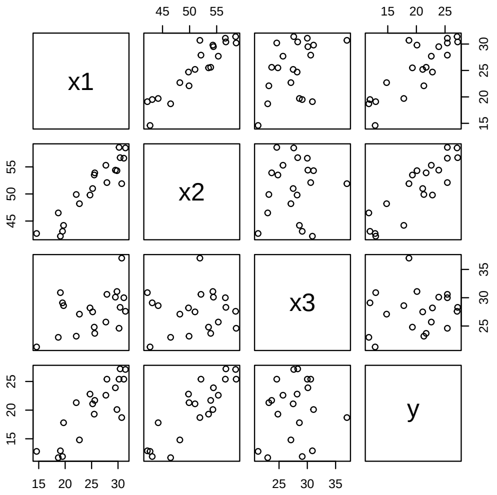
- Scaling of X variables matters—standardize when units are incommensurate (done by default by `glmnet`)

- Criticisms of ridge regression:
  - The optimal value of λ depends on β and σ^2 (error variance), which are being estimated by the regression
  - Lack of theoretical justification for particular penalty term—why unweighted sum of squares?

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.

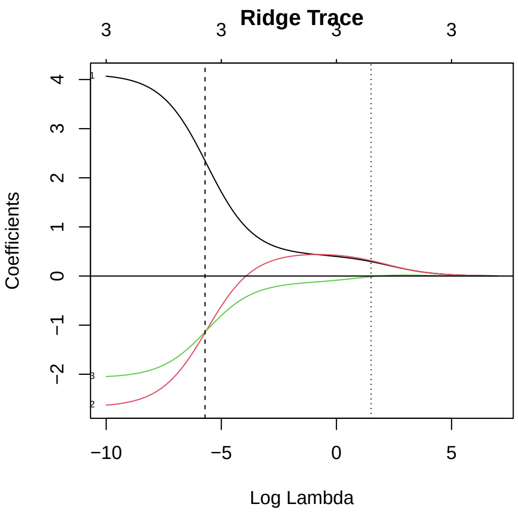
```
bodyfat = data.frame(
  x1=c(19.5,24.7,30.7,29.8,19.1,25.6,31.4,
    27.9,22.1,25.5,31.1,30.4,18.7,19.7,
    14.6,29.5,27.7,30.2,22.7,25.2),
  x2=c(43.1,49.8,51.9,54.3,42.2,53.9,58.5,
    52.1,49.9,53.5,56.6,56.7,46.5,44.2,
    42.7,54.4,55.3,58.6,48.2,51.0),
  x3=c(29.1,28.2,37.0,31.1,30.9,23.7,27.6,
    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 = lm(y~x1+x2+x3, bodyfat)
> coef(fit.lm)   # sign flips on x2, x3
> summary(fit.lm)
       Estimate Std Err t-value Pr(>|t|)
(Int)   117.085  99.782   1.173    0.258
x1        4.334   3.016   1.437    0.170
x2       -2.857   2.582  -1.106    0.285
x3       -2.186   1.595  -1.370    0.190
Multiple R-squared:  0.8014
F-stat: 21.52 on 3, 16 DF,  P=7.343e-06

> vif(fit.lm)
      x1       x2       x3
708.8429 564.3434 104.6060
```



- Corr$(y, x_j) > 0, \forall j$
- F is highly sig, but none of the t tests are sig. R^2 pretty big
- Sign flips for x_2 and x_3
- Corr$(x_j, x_{j'})$ modest (0.08) to large (0.92), but not very large. VIFs are giant!

Ridge Regression in glmnet

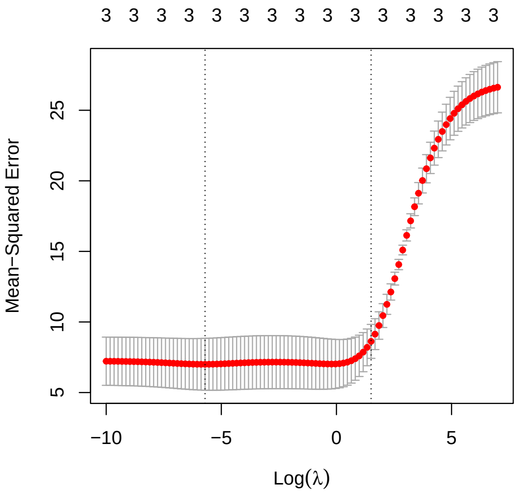
```
> # glmnet requires an X matrix:
> x = model.matrix(y ~ .-1, bodyfat)
> # -1 drops the intercept
> head(x)
    x1   x2   x3
1 19.5 43.1 29.1
2 24.7 49.8 28.2
3 30.7 51.9 37.0
4 29.8 54.3 31.1
5 19.1 42.2 30.9
6 25.6 53.9 23.7

> # we usually let glmnet pick lambda
> # I pick lambdas for class example
> lam = exp(seq(-10, 7, length=100))

> # alpha=0 specifies ridge
> # alpha=1 (default) lasso
> # we usually don't specify lambda=
> fit=glmnet(x, bodyfat$y, alpha=0,
    lambda=lam)

> # view ridge trace
> # label=T shows var nums on left
> plot(fit, xvar="lambda", label=T)
> abline(h=0) # add horizontal 0 line
> title("Ridge Trace")

> round(cbind(lambda=fit$lambda,
  loglambda=log(fit$lambda), t(fit$beta)),2)
100 x 5 sparse Matrix of class "dgCMatrix"
     lambda loglambda   x1    x2    x3
s0  1096.63      7.00 0.00  0.00  0.00
s1   923.60      6.83 0.00  0.00  0.00
s2   777.87      6.66 0.01  0.01  0.00
s3   655.14      6.48 0.01  0.01  0.00
...
```



- Each line shows the value of one slope as λ increases
- The lines "gently" approach 0 as $\lambda \to \infty$
- We observe sign flips, where a slope changes sign
- Which λ should we pick? Use cross validation (vertial lines from next page)

Cross Validation in glmnet

```
> set.seed(12345) # for replicability
> fit2 = cv.glmnet(x, bodyfat$y, alpha=0,
  lambda=lam, nfolds=5)
> # default is 10 folds, usually OK

> names(fit2)
> # glmnet.fit gives fit on entire data

> # lambda.min gives value to min MSE
> (l=fit2$lambda.min);log(l)
[1] 0.003322391
[1] -5.707071
> abline(v=log(l),lty=2)
> # add to plot previous page

> # more conservative possible lambda val
> (l2=fit2$lambda.1se);log(l2)
[1] 4.504381
[1] 1.505051
> abline(v=log(l2), lty=3) # add to plot

> plot(fit2) # create plot to right

> # How to find coefficients?
> predict(fit2$glmnet.fit, s=l, type="coef")
                   s1
(Intercept) 51.412645
x1           2.346499
x2          -1.155780
x3          -1.138833
> predict(fit2$glmnet.fit, s=l2, type="coef")
                     s1
(Intercept) -2.92369522
x1           0.29384750
x2           0.31326050
x3          -0.01255089
```



Vertial lines show lambda.min and lambda.1se values and match those added to previous plot with abline

The Lasso

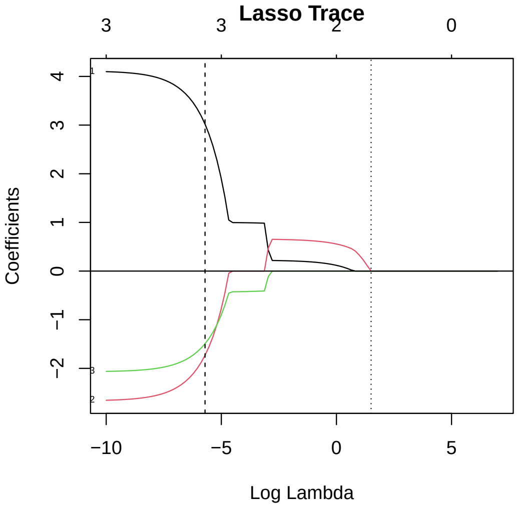
- Ridge penalizes $\sum_j \beta_j^2$, the **squared Euclidean length** (ℓ_2 norm) of the slope vector
- **Lasso** penalizes $\sum_j |\beta_j|$, the **taxi-cab length** (ℓ_1 norm) of the slope vector

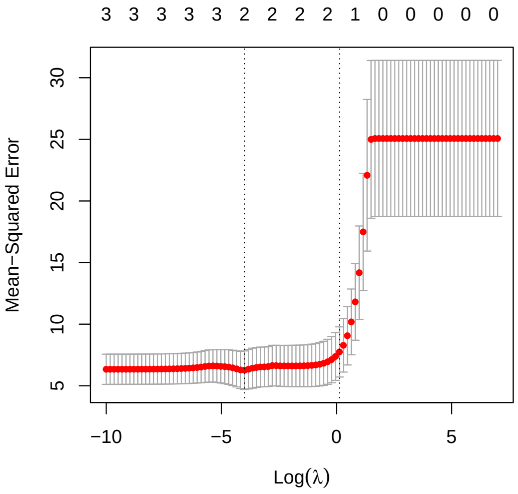
$$\hat{\boldsymbol{\beta}}_\lambda = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left[\sum_{i=1}^{k} (y_i - \beta_0 - \mathbf{x}_i^\mathsf{T} \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right]$$

- We can think of the methods as having different constraints:
  - Subset selection: $\min_{\boldsymbol{\beta}}$ SSE subject to $\sum_j \mathbb{1}(\beta_j \neq 0) \leq s$
  - Ridge: $\min_{\boldsymbol{\beta}}$ SSE subject to $\sum_j \beta_j^2 \leq s$
  - Lasso: $\min_{\boldsymbol{\beta}}$ SSE subject to $\sum_j |\beta_j| \leq s$
- The lasso and ridge regression shrink coefficients towards zero, but the lasso tends to force some coefficients to equal zero, similar to variable subset selection.
- Rule of thumb: use lasso if you think some variables should be dropped

Lasso Bodyfat in glmnet

```
> set.seed(12345)
> # I skip right to cv.glmnet
> # lasso default (alpha=0)
> fit.l1 = cv.glmnet(x, bodyfat$y,
  lambda=lam, nfolds=5)

> plot(fit.l1$glmnet.fit, xvar="lambda",
   label=T); abline(h=0)
> title("Lasso Trace")

> (l=fit2$lambda.min);log(l)
[1] 0.01805755
[1] -4.014191
> abline(v=log(l),lty=2)

> (l=fit2$lambda.1se);log(l)
[1] 1.43103
[1] 0.3583945
> abline(v=log(l), lty=3)

> round(cbind(
  lambda=fit.l1$glmnet.fit$lambda,
  loglambda=log(fit.l1$glmnet.fit$lambda),
    t(fit.l1$glmnet.fit$beta)), 2)

      lambda loglambda   x1    x2    x3
s0   1096.63      7.00 .     .     .
...
s32     4.50      1.51 .     .     .
s33     3.79      1.33 .     0.11  .
...
s37     1.91      0.65 0.02  0.46  .
...
s57     0.06     -2.79 0.22  0.65  .
s58     0.05     -2.96 0.43  0.47 -0.11
s59     0.04     -3.13 0.98  .    -0.41
...
s67     0.01     -4.51 1.00  .    -0.43
s68     0.01     -4.68 1.05 -0.05 -0.45
...
s99     0.00    -10.00 4.10 -2.66 -2.06
```





In lasso trace the lines "nosedive" into the 0 line (selection)

College Case JWHT Problem 6.9

This problem is a variation of 6.9 in JWHT on page 263. It uses the College data. The labels are on page 54.

1. Read the data into R, fix the row names, and create a test set.
2. How many cases are assigned to training? Test?
3. The dependent variable is `Apps`. Generate a histogram. Comment.
4. Regress `Apps` on all other variables using only the training data. Examine the residual plot and comment.
5. Replace `Apps` with its square root as a variance stabilizing transformation.
6. *Full model.* Do problem 6.9b. Note that you are using the square root of apps as the dependent variable and in your evaluation on the test set. For test set error, report the mean. Examine the residual plot and comment. What are the power predictors?
7. *Step model.* Apply the `step` function to the fitted "full" model from the previous part and report the test set error. What are the "power" predictors and signs?
8. *Ridge model.* Do problem 6.9c. Plot the ridge trace, and the fitted `cv.glmnet` object showing cross-validated MSE against λ. What is the optimal value of λ?
9. *Lasso model.* Do problem 6.9d. Report the same things as with ridge.
10. *Improved model.* 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.