Statistical Analysis

Therri Usher

Tuesday, March 24, 2015

Review: Data Processing

An easier way to create new variables, other than using if statements and for loops, is as follows:

1. Create a new variable in matrix form.
2. Use row indexing to define the new variable.
3. Format the variable as a factor and assign levels if needed.

Review: Data Processing

Say that I want to define a variable for non-Hispanic black and non-Hispanic white in NHATS.

```
library(foreign)

dat <- read.dta("C:/Users/owner/Dropbox/Ph.D/Health Dispari

n.obs <- dim(dat)[1]

attach(dat)
```

Review: Data Processing

```
race <- matrix(nrow=n.obs)
race[rl1yourrace1==1] <- 0
race[rl1yourrace2==1] <- 1
race[rl1yourrace3==1] <- 2
race[rl1yourrace4==1] <- 3
race[rl1yourrace5==1] <- 4
race[rl1yourrace6==1] <- 5
race[rl1yourrace7==1] <- 6
race[rl1yourrace8==1] <- 7
race <- as.factor(race)
levels(race) <- c("White", "AA", "NativeAm", "Asian",
                  "NativeHawaii", "PacIsl", "Other")
```

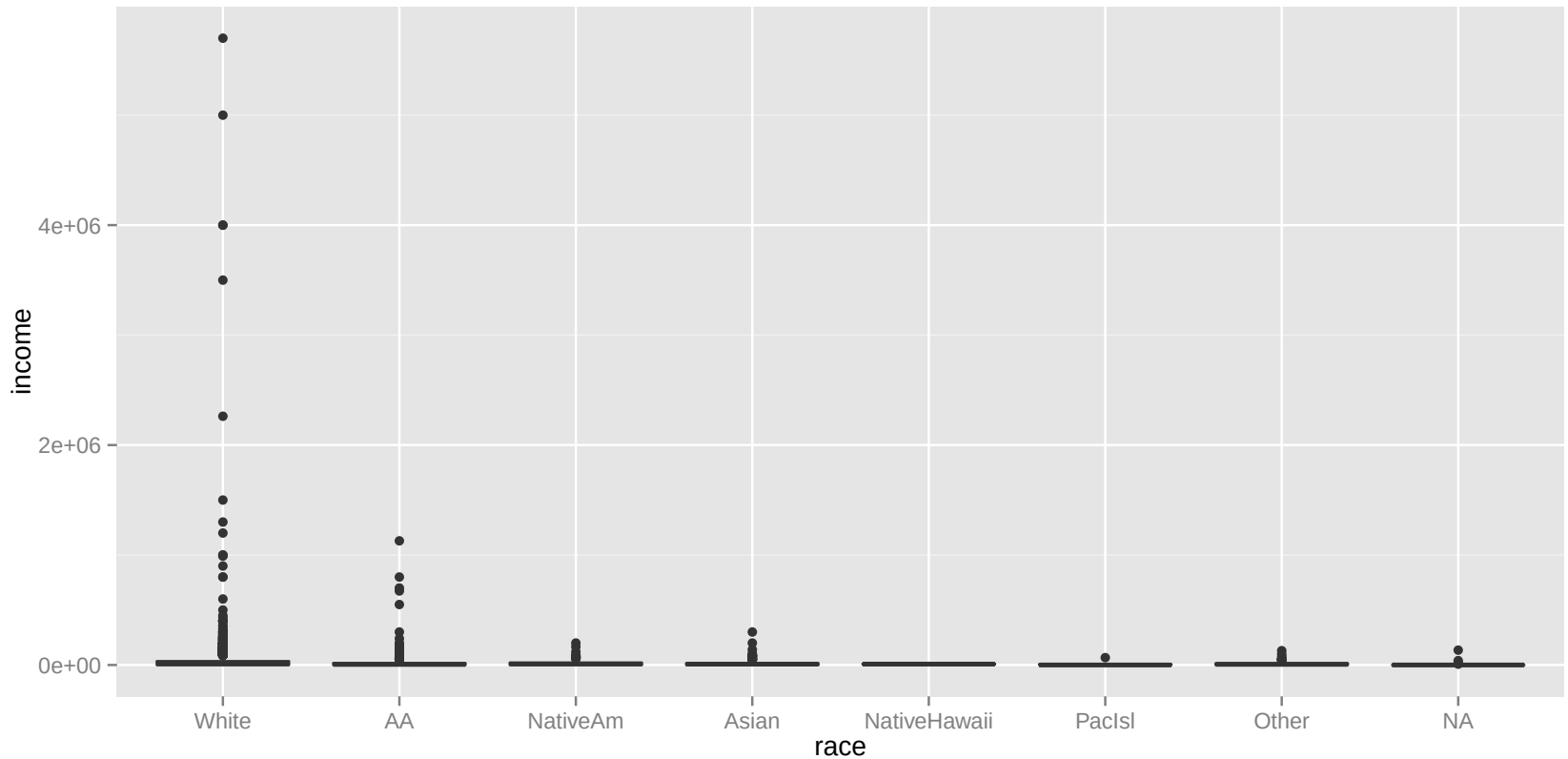
Review: Data Processing

```
hispanic <- matrix(nrow=n.obs)
hispanic[rl1hisplatno==2] <- 0
hispanic[rl1hisplatno==1] <- 1
hispanic <- as.factor(hispanic)
levels(hispanic) <- c("Not Hispanic", "Hispanic")

race.eth <- matrix(nrow=n.obs)
race.eth[race=="White" & hispanic=="Not Hispanic"] <- 0
race.eth[race=="AA" & hispanic=="Not Hispanic"] <- 1

table(race.eth)
```

```
## race.eth
##    0    1
## 5078 1611
```

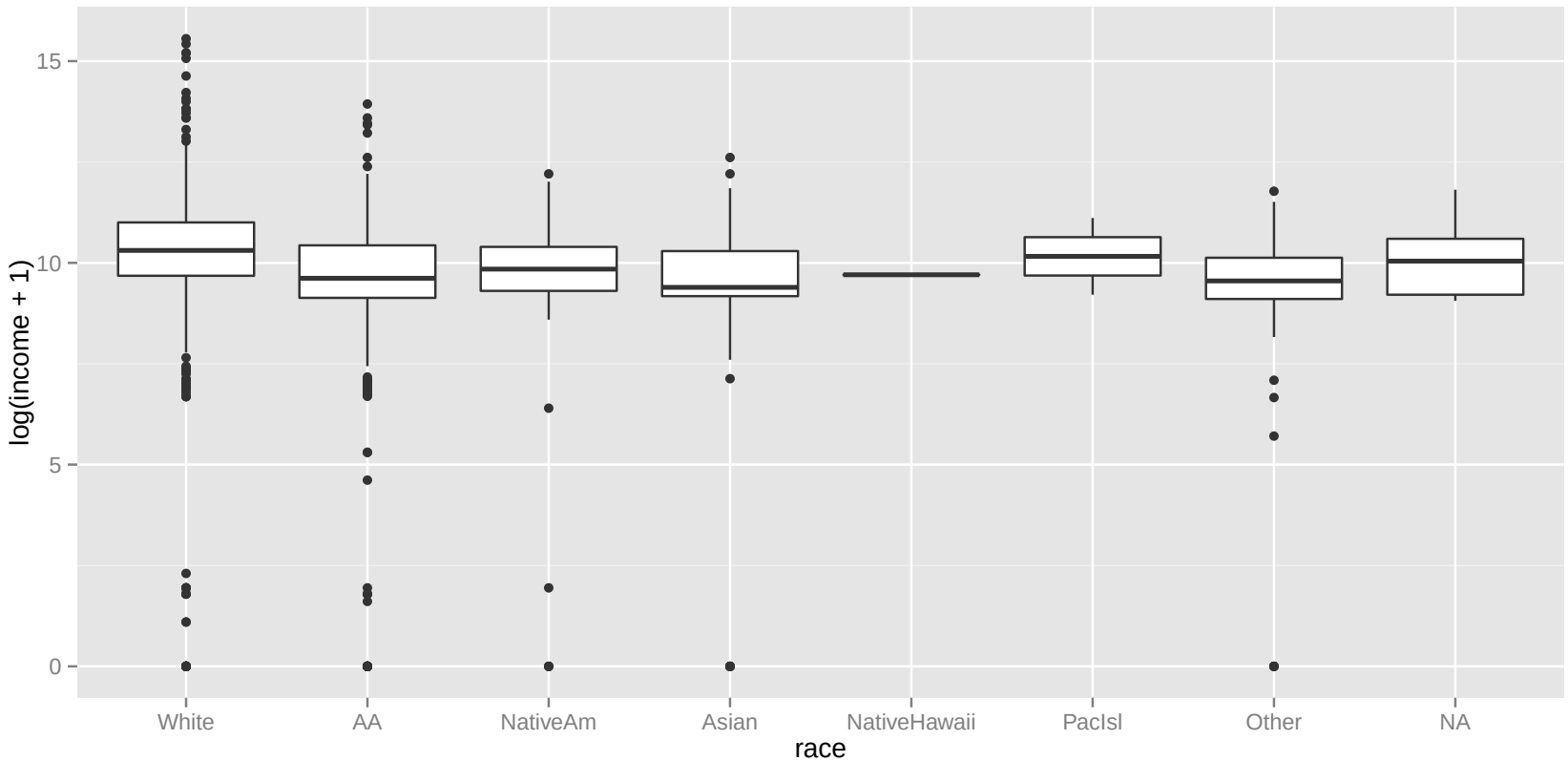
Review: Exploratory Data Analysis

```
library(ggplot2)
income <- ia1totinc

qplot(race, income, geom="boxplot")
```



Review: Exploratory Data Analysis

```
qplot(race, log(income+1), geom="boxplot")
```



Review: Exploratory Data Analysis

```
t.test(income ~ race.eth)
```

```
##
##  Welch Two Sample t-test
##
## data:  income by race.eth
## t = 6.935, df = 6562, p-value = 4.439e-12
## alternative hypothesis: true difference in means is not
## 95 percent confidence interval:
##  11313 20229
## sample estimates:
## mean in group 0 mean in group 1
##           32189           16418
```

Review: Exploratory Data Analysis

```
table(phys.act)
```

```
## phys.act
##    0    1
## 2621 4981
```

```
table(phys.act, race.eth)
```

```
##          race.eth
## phys.act    0    1
##        0 1650  655
##        1 3423  956
```

Review: Exploratory Data Analysis

```
chisq.test(phys.act, race.eth)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity corre
##
## data:  phys.act and race.eth
## X-squared = 35.44, df = 1, p-value = 2.634e-09
```

What Is A Statistical Model?

"A statistical model embodies a set of assumptions concerning the generation of the observed data, and similar data from a larger population. A model represents, often in considerably idealized form, the data-generating process." - Wikipedia

What Is A Statistical Model?

> *All models are wrong, but some are useful. - George E. P. Box*

We use models that we know are wrong all the time. Even if they are not wrong, we do not always assume the model represents how the data was generated. Mostly, we are interested in models that explain the variation in the data.

Models To Be Discussed

We will (hopefully) discuss the following models in the next 2 weeks:

- Linear regression
- Logistic regression
- Generalized linear regression (GLM)
- Generalized estimating equations (GEE)
- Prediction (black-box) models

For each model, we will discuss the following points:

1. Application
2. Assumptions
3. Model Structure
4. Interpretation
5. Fitting in R

Models To Be Discussed

We will also discuss the following aspects of statistical modeling:

- Producing standard errors
- Tests of significance
- Power analysis

Discussion Time

What do you know about linear and logistic regression? Think about them in terms of their applications, assumptions, interpretations, and fitting in R.

Discuss with your group then we will discuss as a class.

Linear Regression

Linear Regression

Application

- Perhaps the most common model in the history of statistics
- Used to estimate associations between covariates and outcomes independent of other covariates in the model
- Can also be used to predict the values of the outcome for each subject as well as estimate the variance of Y

Linear Regression

Assumptions

L: The outcome is a **linear** combination of the covariates.

I: Observations are **independent** (uncorrelated) of each other.

N: The residuals of the outcome has a **normal** distribution

E: The outcome has **equal** residual variance regardless of the values of the covariates.

Not often mentioned: The independent variables are fixed and measured without error.

Linear Regression

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$$
$$\epsilon \sim N(0, \sigma^2)$$

β_0 is the predicted value of Y when $X_1 = 0$ and $X_2 = 0$

β_1 is the expected change in Y when X_1 changes by one unit and X_2 remains fixed

β_2 is the expected change in Y when X_2 changes by one unit and X_1 remains fixed

σ^2 is the variance of the residuals and therefore the variance of Y

Linear Regression

Use the lm command in R to fit a linear regression:

```
data(cars)

linear.fit <- lm(speed ~ dist, data=cars)
linear.fit
```

```
##
## Call:
## lm(formula = speed ~ dist, data = cars)
##
## Coefficients:
## (Intercept)         dist
##       8.284        0.166
```

Linear Regression

```
summary(linear.fit)
```

```
##
## Call:
## lm(formula = speed ~ dist, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.529 -2.155  0.362  2.438  6.418
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.2839     0.8744    9.47  1.4e-12 ***
## dist          0.1656     0.0175    9.46  1.5e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
##
## Residual standard error: 3.16 on 48 degrees of freedom
```

Linear Regression

```
confint(linear.fit)
```

```
##                 2.5 %  97.5 %
## (Intercept) 6.5258 10.0420
## dist        0.1304  0.2007
```

Linear Regression

```
names(linear.fit)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "ra
##  [5] "fitted.values" "assign"        "qr"            "di
##  [9] "xlevels"       "call"          "terms"         "mo
```

Logistic Regression

Logistic Regression

Application

- Perhaps the second-most common model in the history of statistics
- Used to estimate associations between covariates and outcomes in the form of odds ratios, independent of other covariates in the model
- Can also be used to predict the probabilities of experiencing the outcome

Logistic Regression

Assumptions

- The probabilities are a logistic function of the independent variables.
- Observations are independent of each other.
- The outcome is dichotomous, or Bernoulli-distributed.
- Independent variables are fixed and measured without error.

NOTE: Logistic regression requires larger sample sizes than linear regression. Logistic uses maximum likelihood and ML estimates are large sample estimates.

Logistic Regression

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

β_0 is the estimated log odds of the outcome when $X_1 = 0$ and $X_2 = 0$

β_1 is the change in log odds of having the outcome when X_1 changes by one unit

β_2 is the change in log odds of having the outcome when X_2 changes by one unit

Logistic Regression

e^{β_1} is the odds of having the outcome when $X_1 = k + 1$ compared to when $X_1 = k$ and X_2 remains fixed

e^{β_2} is the odds of having the outcome when $X_2 = k + 1$ compared to when $X_2 = k$ and X_1 remains fixed

Ex: If Y is disease status, X_1 represents age, and $e^{\beta_1} = 1.5$, then the odds of having the disease are 1.5 times larger with every 1-year increase in age.

Logistic Regression

```
data(diamonds)
n.obs <- dim(diamonds)[1]
big <- matrix(nrow=n.obs)

big[diamonds$carat < 1] <- 0
big[diamonds$carat >= 1] <- 1
```

Logistic Regression

```
logit.fit <- glm(big ~ cut, data=diamonds,
                 family="binomial"(link="logit"))
```

Logistic Regression

```
summary(logit.fit)
```

```
##
## Call:
## glm(formula = big ~ cut, family = binomial(link = "logit
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.190  -0.934  -0.797   1.266   1.613
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.4421     0.0130  -33.98   <2e-16 ***
## cut.L        -0.5633     0.0347  -16.24   <2e-16 ***
## cut.Q        -0.0117     0.0310   -0.38      0.7
## cut.C        -0.4741     0.0271  -17.52   <2e-16 ***
## cut^4        -0.2346     0.0221  -10.62   <2e-16 ***
## ---
```

Logistic Regression

```
exp(logit.fit$coef)
```

```
## (Intercept)         cut.L         cut.Q         cut.C          cu
##      0.6427        0.5693        0.9883        0.6224         0.7
```

Question

Have you learned anything new about linear or logistic regression?

Review

You have already been exposed to linear and logistic regression, so this should have been a review. Even though you might not have known, you have been exposed to generalized linear models too...

Generalized Linear Regression

Generalized Linear Regression

It's pretty much what the name says. It's a generalizable collection of regression models that allow you to model outcomes of various distributions.

It also allows the linear model to be related to the expected value of the outcome through a link function. Lastly, it allows for the modeling of the variance as a function of the expected value.

Finally, it estimates associations between covariates and the outcome as well as predicted expected values.

Generalized Linear Regression

To use generalized linear regression, you need a probability distribution from the exponential family and three functions:

1. Linear predictor: $\eta_i = \beta_0 + \beta_1 X_{1i} + \ldots + \beta_p X_{pi}$
2. Link function: $g(\mu_i) = \eta_i$ where $\mu_i = E[Y_i]$
   - Each distribution has a "canonical link", a link function that has special properties for that distribution. For binomial and Bernoulli, it's the logit.
3. Variance function: $Var(Y_i) = \phi V(\mu_i)$, where ϕ is a constant called the dispersion parameter

Generalized Linear Regression

Assumptions

- The linear predictor, link and variance functions are correctly specified.
- There is no overdispersion.
- The observations are independent from each other.
- Independent variables are fixed and measured without error.

Generalized Linear Regression

$$g(\mu_i) = \beta_0 + \beta_1 X_{1i} + \ldots + \beta_p X_{pi}$$
$$Var(Y_i) = \phi V(\mu_i)$$

From Generalized Linear Regression to Logistic Regression

Let $Y_i \sim Bernoulli(p_i)$. Then $E[Y_i] = \mu_i = p_i$ and $Var(Y_i) = p_i(1 - p_i) = \mu_i(1 - \mu_i)$.

The link function must map $\mu_i = p_i$ to $(-\infty, \infty)$. Therefore, we can use $g(\mu_i) = logit(\mu_i) = \log \frac{\mu_i}{1-\mu_i}$. Notice that $\phi = 1$ in this case.

Therefore, we get

$$\log \frac{p_i}{1 - p_i} = \beta_0 + \beta_1 X_{1i} + \ldots + \beta_p X_{pi}$$

Generalized Linear Regression

To fit generalized linear models in R, use the glm command: glm(formula, family, data)

```
logit.fit <- glm(big ~ cut, data=diamonds,
                 family="binomial"(link="logit"))
same.logit.fit <- glm(big ~ cut, data=diamonds,
                      family="binomial")

linear.fit <- glm(speed ~ dist, data=cars)
same.linear.fit <- glm(speed ~ dist, data=cars,
                       family="gaussian"(link="identity"))
```

Poisson Regression

Let $Y \sim Poisson(\lambda_i)$. Then, $E[Y_i] = Var(Y_i) = \lambda_i$. Therefore, $V(\mu_i) = \mu_i$ and $\phi = 1$.

The link function must map to $(-\infty, \infty)$. The canonical link for Poisson is log.

$$\log(\lambda_i) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i}$$

Poisson is commonly used to model count data. Therefore, it can have the following interpretations:

β_0 is the log of the expected count when $X_{1i} = 0$ and $X_{2i} = 0$

β_1 is the change in the log of the expected count when there is a 1-unit change in X_{1i} and X_{2i} is fixed

β_2 is the change in the log of the expected count when there is a 1-unit change in X_{2i} and X_{1i} is fixed

CAUTION: Survey Weights

Statistics as you know it is built on the assumption that all observations have an equal probability of being selected. In some studies, such as NHANES, they sample non-randomly and/or assign a higher probability of selection to certain subgroups in order to sample more of them. This is known as oversampling.

When we perform analyses using such data, we must incorporate the survey weights into the analysis. Otherwise, the estimates are biased, incorrect inferences can be made, and the model does not represent the population of interest.

See the “Survey Weighting" R script for the code to define NHANES sample weights and examples of how to use the survey weighting in GLM models.

Active Learning Exercise

Use the mpg data set in the ggplot2 package. Model the city miles per gallon and determine if the transmission is associated with it.

- Decide the proper GLM
- Describe why it is appropriate, addressing the assumptions
- Create the linear predictor, including potential confounders
- Fit the model in R
- Interpret the results

Generalized Estimating Equation

Discussion Time

Suppose we want to analyze data that violates the independence assumption, i.e. subjects are genetically related or are measured across time. How would you deal with the data so that you could fit a model to it?

Discuss in your groups before we discuss as a class.

Generalized Estimating Equation

Generalized estimating equation (GEE) is used to estimate GLM parameters but incorporates correlation structures to model dependence between observations. It also allows for modeling overdispersion.

It estimates the parameters of a mean model, such as associations between covariates and the outcome and predicted expected values, as well as the parameters of a correlation structure, which allows for estimation of the correlation between observations in a cluster.

Generalized Estimating Equation

Similar to GLM, you need the probability distribution of the outcome, as well as the following models:

- Mean model: Consists of the linear predictor and the link function – $g(\mu_i) = \beta_0 + \beta_1 X_{1i} + \ldots + \beta_p X_{pi}$
- Covariance structure: A specified matrix that defines the correlation between observations in a cluster

NOTE: Observations in different clusters are assumed independent!

Parameters estimates converge to their true values even if the covariance structure is not entirely correct. However, the mean model must be correctly specified.

Generalized Estimating Equation

Assumptions

- The mean model must be correctly specified.
- Independent variables are considered fixed and measured without error.

Interpretation: The same as GLM

NOTE: GEE does not use a full likelihood. It uses a quasi-likelihood estimation, which only places assumptions on the mean and variance.

Generalized Estimating Equation

Independence

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Exchangeable Correlation

$$\begin{bmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho \\ \rho & \rho & 1 \end{bmatrix}$$

Autoregressive Correlation

$$\begin{bmatrix} 1 & \rho & \rho^2 \\ \rho & 1 & \rho \\ \rho^2 & \rho & 1 \end{bmatrix}$$

Generalized Estimating Equation

Toeplitz Correlation

$$\begin{bmatrix} 1 & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_1 \\ \rho_2 & \rho_1 & 1 \end{bmatrix}$$

Unstructured Correlation

$$\begin{bmatrix} 1 & \rho_{12} & \rho_{13} \\ \rho_{21} & 1 & \rho_{23} \\ \rho_{31} & \rho_{32} & 1 \end{bmatrix}$$

WARNING: Be careful with unstructured correlation matrix! Model may not converge if using unstructured.

Generalized Estimating Equation

To fit GEE in R, install the gee or geepack package.

```
require(geepack)
```

```
## Loading required package: geepack
```

```
wages <- read.table("http://www.ats.ucla.edu/stat/r/example
```

Generalized Estimating Equation

```
gee.ind <- geeglm(lnw ~ exper, id=id, data=wages,
                  family="gaussian", corstr="independence")
summary(gee.ind)
```

```
##
## Call:
## geeglm(formula = lnw ~ exper, family = "gaussian", data
##     id = id, corstr = "independence")
##
##  Coefficients:
##             Estimate Std.err  Wald Pr(>|W|)
## (Intercept)  1.69054 0.01145 21798   <2e-16 ***
## exper        0.05209 0.00269   375   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
##
## Estimated Scale Parameters:
##             Estimate Std.err
```

Generalizing Estimating Equation

```
gee.exch <- geeglm(lnw ~ exper, id=id, data=wages,
                   family="gaussian",
                   corstr="exchangeable")
summary(gee.exch)
```

```
##
## Call:
## geeglm(formula = lnw ~ exper, family = "gaussian", data
##     id = id, corstr = "exchangeable")
##
##  Coefficients:
##             Estimate Std.err  Wald Pr(>|W|)
## (Intercept)  1.71572 0.01091 24722   <2e-16 ***
## exper        0.04569 0.00233   384   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
##
## Estimated Scale Parameters:
```

Generalizing Estimating Equation

If you are using longitudinal data or data that might violate the independence assumption in your final project, *come discuss it with me*.

Active Learning Exercise

Use the respiratory data set in the geepack package. Create a regression model with the variable outcome as the outcome. Assess if treatment type is associated with the outcome. Experiment with different correlation structures and see how the estimates change.

- Decide the proper mean model
- Describe why it is appropriate, addressing the assumptions
- Create the linear predictor, including potential confounders
- Fit the model in R, experimenting with different correlation structures
- Interpret the results

Standard Errors and Tests of Regression Coefficients

Estimating Standard Errors

For linear regression, the standard errors of the regression coefficients are well-known:

$$Var(\hat{\beta}) = \sigma^2(X^T X)^{-1}$$

Estimating Standard Errors

The standard errors generated from linear regression, logistic regression, GLM, and GEE all can be estimated using Huber sandwich estimator, or robust sandwich estimator.

Without going into the mathematical details, the sandwich estimator can estimate the variance of the coefficients even when the underlying model is incorrect, hence "robust".

However, if the model is near correct, its standard errors might be a bit larger than needed. If the model is entirely incorrect, the standard errors are somewhat correct but the estimates are still biased.

Estimating Standard Errors

The sandwich package in R allows you to calculate variances and covariances of the regression coefficients.

```
library(sandwich)

linear.fit <- lm(speed ~ dist, data=cars)
sandwich(linear.fit)
```

```
##             (Intercept)      dist
## (Intercept)      0.7636 -0.014425
## dist            -0.0144  0.000361
```

Estimating Standard Errors

For maximum likelihood estimates (ex: all estimates generated using GLM), the sandwich estimator is equivalent to

$$\frac{1}{I(\hat{\beta})} = \frac{1}{-E\left[\frac{\partial^2}{\partial^2 \beta} \log f(X;\beta)|\beta = \hat{\beta}\right]}$$

Estimating Standard Errors

Another common method is bootstrapping. In bootstrapping, we treat the sample of observations as if it was the population. We then take a sample (with replacement) from our new "population" and fit the regression model. We repeat this process multiple times and get multiple estimates of the regression coefficients, which serves as a sampling distribution. The standard deviation of the sampling distribution is a reasonable estimate of the standard error of the estimate of the regression coefficients.

How Does R Estimate Standard Errors?

That's a good question... still working on finding that out.

Testing Regression Coefficients

There are various ways to test regression coefficients.

Linear model

- t-test

Non-linear model

- F test
- Wald test
- Likelihood ratio test
- Lagrange multiplier test
- Score test

Student's t test

You might find this familiar...

Ho: $\mu = \mu_0$

Ha: $\mu \neq \mu_0$

$$\frac{\bar{X} - \mu_0}{\frac{s}{\sqrt{n}}} \sim t_{1-\frac{\alpha}{2}, n-1}$$

Student's t test

Recall that under linear regression, coefficient estimates are normally distributed. Therefore, the t-test can also be used for testing one parameter at a time in linear regression models:

Ho: $\beta_1 = c$

Ha: $\beta_1 \neq c$

$$\frac{\beta_1 - c}{\sqrt{k^T \sigma^2 (X^T X)^{-1} k}} \sim t_{1-\frac{\alpha}{2}, n-p}$$

F test

The F test can be used for testing more than one regression coefficient at a time in regression models.

The F test p-value given by R at the end of linear and other regression summary output tests whether all coefficients except the intercept are 0, with the alternative being that at least one coefficient differs significantly from 0.

F test

```
summary(linear.fit)$fstatistic
```

```
## value numdf dendf
##  89.6   1.0  48.0
```

Wald Test

This is the one of the most common ways of testing regression coefficients for various kinds of models. It can also test functions of regression coefficients, such as the difference of regression coefficients, and more than one hypothesis at a time.

Wald Test

Test statistic:

$$\frac{(\hat{\beta} - \beta_0)^2}{Var(\hat{\beta})} \xrightarrow{D} \chi^2_Q$$

where Q is the number of hypotheses being tested.

Wald Test

To perform Wald tests in R, use the wald.test command in the aod package.

```
library(aod)
wald.test(b = coef(logit.fit), Sigma = vcov(logit.fit),
          Terms=2:5)
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 1342.4, df = 4, P(> X2) = 0.0
```

Wald Test

```
l <- cbind(0, 1, 0, 0, -1)
wald.test(b = coef(logit.fit), Sigma = vcov(logit.fit),
          L = l)
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 58.1, df = 1, P(> X2) = 2.5e-14
```

Likelihood Ratio Test

We will discuss this under “Model Checking and Evaluation” in a couple of weeks.

Statistical Power

Discussion Time

Define type I error, type II error, and power with regards to hypothesis testing. How are they related? Provide a real-life example for the definitions.

Discuss in your groups. Then, we will discuss as a class.

Power and Sample Size Analysis

If you ever want to win a grant to do any kind of scientific research whatsoever, you must be able to show that your sample size can generate enough power to detect difference, or effect size, of a certain amount.

Power

Power is the ability to detect significant differences. It is defined as P(reject Ho | Ha is true) or

Ho: $\beta = \beta_0$

Ha: $\beta \neq \beta_0$

Test Statistic: $x \xrightarrow{D} X$

$$\text{Power} = P(|x| > X|\beta \neq \beta_0)$$

Power

Statistical power increases when:

1. Standard error of the estimate of interest decreases
2. Significance level increases
3. Effect size increases
4. Sample size increases
    - In many cases, sample size and power can be defined as functions of each other.

Power Formula

There are many, *many*, *MANY* equations for calculating power and sample size. The following is an "all-purpose" power formula:

$$Z_{\beta} + Z_{1-\frac{\alpha}{2}} = \frac{\Delta}{SE(\Delta)}$$

where β is the power percentage (typically 80%), α is the significance level, and Δ is the effect size defined as the difference in parameters. Keep in mind that standard errors are typically a function of sample sizes.

Power Formula

Example: Difference of Means

$\Delta = \bar{X}_1 - \bar{X}_2$

$\sigma_1^2 = \sigma_2^2, n_1 = n_2$

$SE(\Delta) = \sqrt{\frac{\sigma^2}{n} + \frac{\sigma^2}{n}}$

$$Z_\beta + Z_{1-\frac{\alpha}{2}} = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{\sigma^2}{n} + \frac{\sigma^2}{n}}}$$

$$n = \frac{2\sigma^2(Z_\beta + Z_{1-\frac{\alpha}{2}})^2}{\Delta^2}$$

Power Calculations

The pwr package in r has a power command for each kind of test, such as t-tests (pwr.t.test) and one-sample proportion tests (pwr.p.test).

Each function requires

- Sample size (or power if you want to estimate sample size)
- Effect size
- Significance level

Power Calculations

```
library(pwr)
pwr.anova.test(k=3, f=0.1, power=0.8, sig.level=0.05)
```

```
##
##      Balanced one-way analysis of variance power calcula
##
##               k = 3
##               n = 322
##               f = 0.1
##       sig.level = 0.05
##           power = 0.8
##
## NOTE: n is number in each group
```

Power Calculations

```
pwr.anova.test(k=3, f=0.1, power=0.5, sig.level=0.05)
```

```
##
##      Balanced one-way analysis of variance power calcula
##
##               k = 3
##               n = 166
##               f = 0.1
##       sig.level = 0.05
##           power = 0.5
##
## NOTE: n is number in each group
```

Power and Sample Size Analysis

This is by no means meant to be an exhaustive education on power and sample size. It is not necessary for your final project.

My suggestion, for those of you working with researchers, is to ask them how they perform their power analyses and learn from what they do. If you want to go into medical or public health research, this is worth learning!

Active Learning Exercise

1. Fit a linear regression of speed on distance using the cars data set in R. Perform a Wald test to determine if the coefficient for distance significantly differs from 0. Does the finding agree with the F statistic generated by the linear model fit? With the p-value of distance generated by the linear model?
2. Calculate the sample size necessary for a two-sample t-test for significance level of 0.05, power of 0.8, and effect size of 0.5. Change the power and effect size and observe how the sample size changes. Is it in the direction you expect?

Prediction

What's So Great About Prediction?

If your only purpose is to prediction and not to draw inference, the possibilities for modeling become even more open and interesting.

Steps For Building A Prediction Model

1. Find the right data
2. Define the error rate
3. Split data into training, validation, and testing sets
4. Build a prediction model on the training set
5. Fine tune it using the validation set
6. Apply **once** to the testing set to obtain the error rate for the model

Common Error Rates

- Mean Squared Error: $E[(\hat{\theta} - \theta)^2] = \text{Bias}(\hat{\theta}, \theta)^2 + \text{Var}(\hat{\theta}) = E[\hat{\theta} - \theta] + \text{Var}(\hat{\theta})$
- Sensitivity (True Positive): P(Test is $+$ | Person is truly $+$)
- Specificity (True Negative): P(Test is - | Person is truly -)
- Accuracy: $\frac{\text{True Positives} + \text{True Negatives}}{\text{Sample Size}}$
- Positive Predictive Value: P(Person is truly $+$ | Test is $+$)
- Negative Predictive Value: P(Person is truly - | Test is -)
- Kappa (Concordance)

Creating Training, Validation, and Testing Sets

There are various ways you can split the data set:

- Random sub-sampling: Randomly assign each observation to training, validation, or test set
- K-fold: Use the first third of the data as training, second third as validation, and final third as test set
- Leave one out: Use only one observation as the test set

With all these options, a good idea is to re-assign the data to the sets, repeat the model building, produce error rates, then average over prediction models and the error rates.

Aspects of Prediction Models

Prediction models tend to have the following four properties:

1. Accuracy
2. Overfitting
3. Interpretability
4. Computational Speed

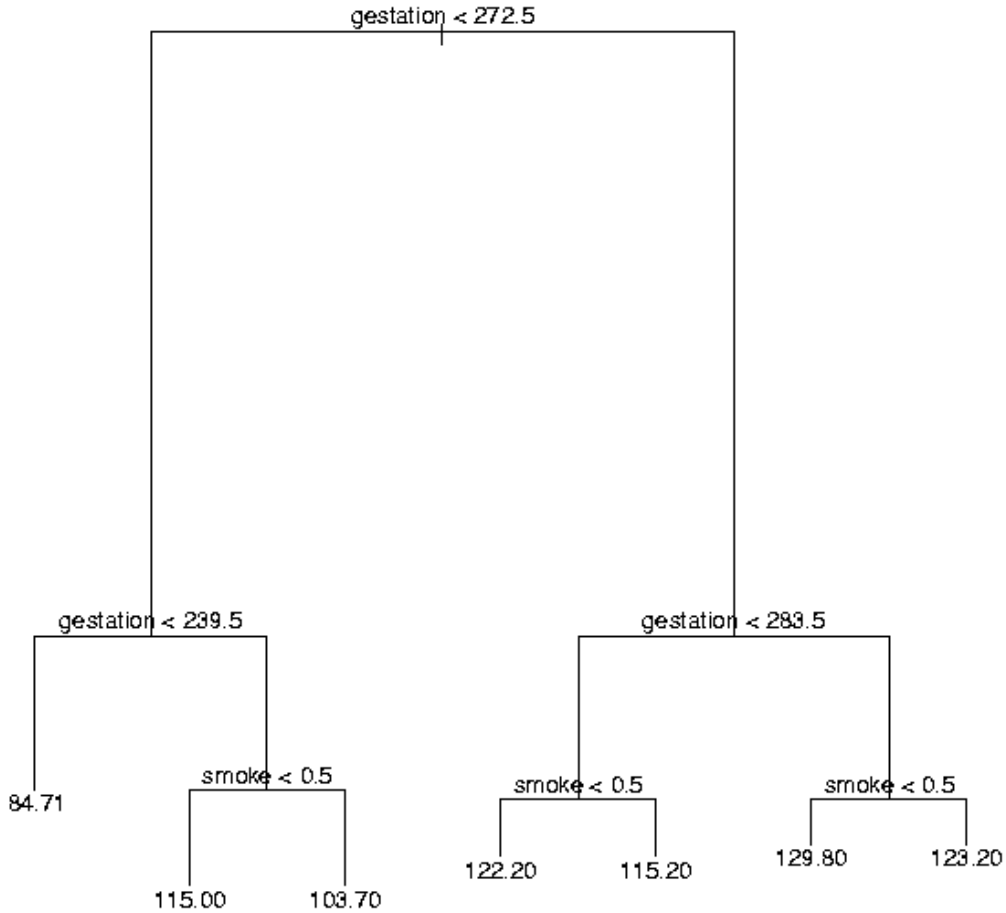
NOTE: GLMs are considered prediction models.

Decision Trees

Decision trees are prediction models that use a set of binary rules to calculate a target value, either a classification (categorical variables) or a regression value (continuous variables).

The binary rules try to find the "best" split at a node, or branch. Many prediction models can be considered decision trees. They each try to find the "best" split, as well as stopping points, in their own way.

Decision Trees



Classification and Regression Trees

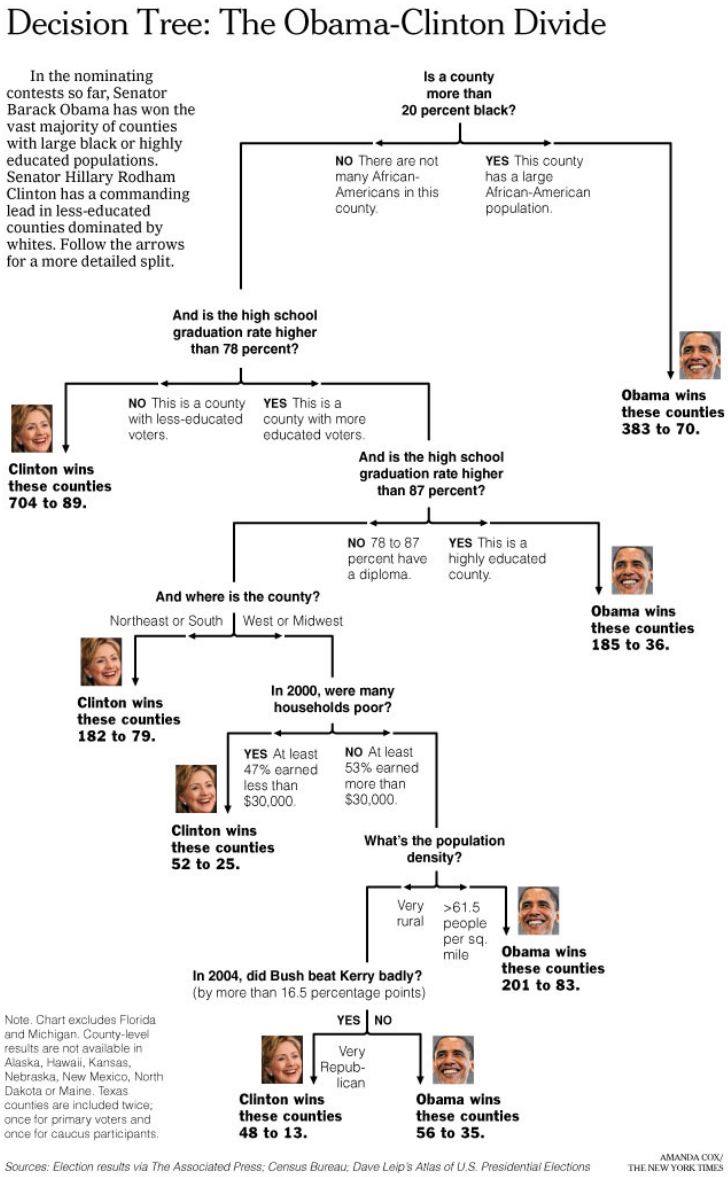
Classification and Regression Trees (CART) are decision trees and operate in the same fashion.

Basic Idea: At each node, create a binary rule that creates a split using the best predictor. The best predictor is determined by some splitting criteria. The process continues within each group separately until reaching a stopping point, which at that point, a classification or regression value is assigned.

Pros: Easy interpretation, tends to be accurate Cons: Susceptible to overfitting, can be computationally complex, only performs binary splits

Possible Fixes: Pruning the decision tree

Classification and Regression Trees



Classification and Regression Trees

To fit CART trees, you can use either the tree package or the more recent rpart package in R.

```
library(rpart)
data(car.test.frame)
tree.fit <- rpart(Reliability~Country+Mileage+Type,
                  data=car.test.frame)
```

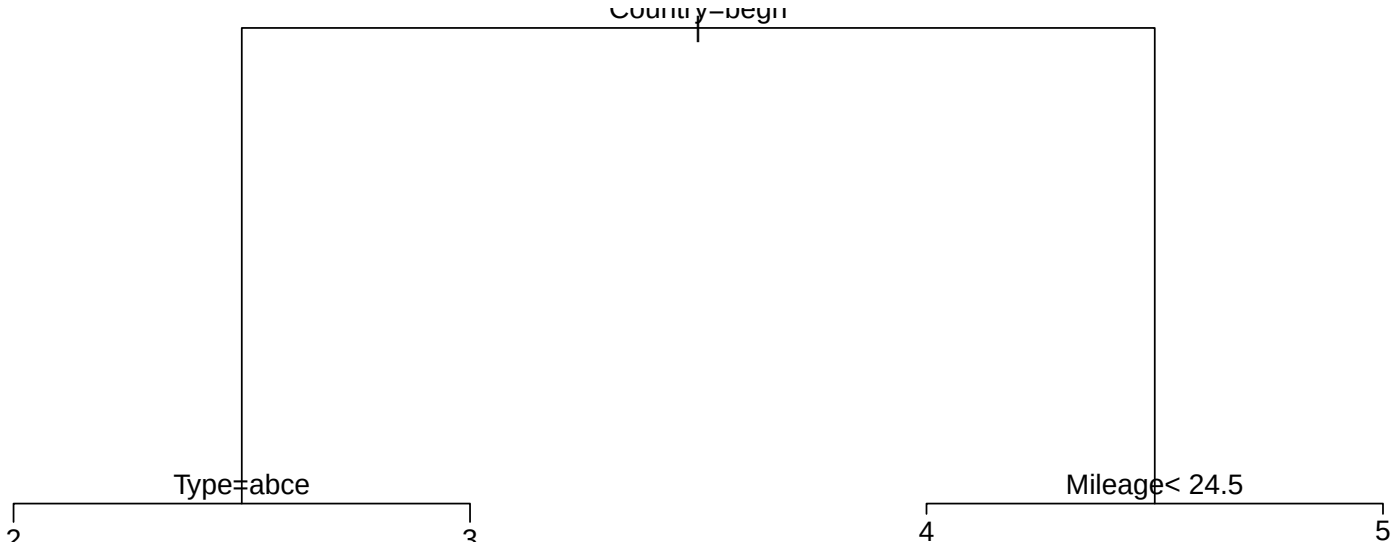
Classification and Regression Trees

```
tree.fit
```

```
## n=49 (11 observations deleted due to missingness)
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
## 1) root 49 102.000 3.39
##   2) Country=Germany,Korea,Sweden,USA 27  26.000 2.33
##     4) Type=Compact,Large,Medium,Sporty 20  16.600 2.15
##     5) Type=Small,Van 7   6.860 2.86 *
##   3) Country=Japan,Japan/USA,Mexico 22   8.770 4.68
##     6) Mileage< 24.5 10   6.400 4.40 *
##     7) Mileage>=24.5 12   0.917 4.92 *
```

Classification and Regression Trees

```
plot(tree.fit)
text(tree.fit)
```



Logic Regression

Logic regression is structured like a generalized linear model but the covariates are actually logical statements.

$$g(E[Y]|L) = \beta_0 + \beta_1 L_1 + \beta_2 L_2 + \dots$$

where, as an example,

$$L_1 = (X_1 \text{ and } X_2), L_2 = (X_1 \text{ or not } X_2)$$

Logic regression produces what the creators call a logic tree, which looks similar to a decision tree.

Pros: Similar to CART, good for logical rules

Cons: Computationally intensive, only allows dichotomous predictors, binary splits

Logic Regression

Use the LogicReg package in R to fit logic regression.

```
require(LogicReg)
```

```
## Loading required package: LogicReg
## Loading required package: survival
## Loading required package: splines
##
## Attaching package: 'survival'
##
## The following object is masked from 'package:aod':
##
##     rats
```

```
data(logreg.testdat)
```

Logic Regression

```
logic.fit <- logreg(resp=logreg.testdat[,1],
                    bin=logreg.testdat[,2:21], type=2,
                    select=1, ntrees=5, nleaves=10)
logic.fit
```

```
## score 0.963
## 1 +2.12 * ((not X3) and (not X4)) +1.21 * (X1 or (X2 or
```

Random Forest

Random Forest is an "ensemble classifier", meaning that it combines the results from multiple decision trees. It can be used for classification or regression.

1. Take bootstrap samples (sample with replacement) from the training set.
2. Create a Random Forest prediction model for the sample.
3. Use the remaining data in the training set to produce errors and measures of variable importance.
4. Repeat the process many times.
5. Average over the models to obtain classification or regression.
   - For classification, assign the class that received the most votes from the models.
   - For regression, assign the average value of the predictions from all the models.

Pros: Accuracy, interpretability

Cons: Overfitting, computational speed

Random Forest

Use the randomForest package in R.

```
require(randomForest)
```

```
## Loading required package: randomForest
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
data(imports85)

rf.fit <- randomForest(cityMpg ~ horsepower
                       +numOfCylinders+fuelType
                       +bodyStyle, data=imports85,
                       mtry=4, importance=TRUE,
                       na.action=na.omit)
```

Random Forest

```
rf.fit
```

```
##
## Call:
##  randomForest(formula = cityMpg ~ horsepower + numOfCyli
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##           Mean of squared residuals: 6.73
##                     % Var explained: 84.3
```

Discussion Time

Would you feel comfortable using a black-box model for a prediction project in which you are working with other medical or public health professionals? Think about this question in terms of model fitting, interpretation, etc.

Discuss within groups first before we discuss it as a class.

Prediction Models

There are many other prediction models that can be used. This is only a very short introduction to prediction modeling!

Active Learning Exercise

Fit a CART model using the car.test.frame data set in the rpart package in order to predict the price of the car models. Use variables in the data set to predict the price. What variables do you find influential in predicting price? Print out the plot with labels to get an idea of the decision tree. Feel free to play around with options in the command.

Final Words

You have seen a ton of models. These models can all be studied in a year-long class so 2 weeks is certainly not enough time to fully understand them. I encourage you all to develop your knowledge of statistical modeling beyond this course. Please feel free to come to me and ask questions about models of interest, particularly if you have questions about model fitting for your final project.

References

1. Logistic Regression
2. UCLA Logistic Regression Example
3. Generalized Linear Model
4. Wald Test
5. Power Analysis
6. Cross-Validation
7. Cohen's kappa
8. Tree-Based Models
9. Introduction to decision trees and random forests