Agenda

1. Logistic Regression
2. Assessing Fit in Logistic Regression

Interpretation of coefficients in logistic model

- β_0: Shifts the curve side-to-side, β_1: changes the shape
- Play with `http://rstudio.smith.edu:3838/log_app/` to see
- If π is a probability, then $\frac{\pi}{1-\pi}$ is the corresponding odds
- The log of the odds is *linear*
  - $\hat{\beta}_1$ is the typical change in $\log(odds)$ for each one unit increase
  - The odds of success are multiplied by $e^{\hat{\beta}_1}$ for each one unit increase
  - These changes are constant

$$odds_X = \frac{\hat{\pi}_X}{1-\hat{\pi}_X} = e^{\hat{\beta}_0+\hat{\beta}_1 X}$$
$$odds_{X+1} = \frac{\hat{\pi}_{X+1}}{1-\hat{\pi}_{X+1}} = e^{\hat{\beta}_0+\hat{\beta}_1 (X+1)}$$
$$\frac{odds_{X+1}}{odds_X} = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 (X+1)}}{e^{\hat{\beta}_0+\hat{\beta}_1 X}} = e^{\hat{\beta}_1}$$

Checking conditions

- Conditions:
  - Linearity of the logit (or $\log(odds)$)
  - Independence
  - Random
- Constant Variance and Normality are no longer applicable

Assessing fit

- Since we don't have sum of squares, we can't use R^2, ANOVA, or F-tests
- Instead, since we fit the model using MLE, we compute the likelihood:

$$L(success) = \hat{\pi}, \qquad L(failure) = 1-\hat{\pi}, \qquad L(model) = \prod_{i=1}^{n} L(y_i)$$

- Because these numbers are usually very small, it is more convenient to speak of the log-likelihood $\log(L)$, which are always negative
- A larger $\log(L)$ is closer to zero and therefore a better fit
- Likelihood Ratio Test (LRT) for simple logistic regression
- $H_0 : \beta_1 = \beta_2 = \beta_3 \cdots \beta_k = 0$, vs. $H_A : \exists \beta_i \neq 0$
- Test statistic $= G = -2\log(L_0) - (-2\log(L))$
- G follows a χ^2 distribution with k d.f.
- 2×2 tables are basically equivalent to logistic regression with binary response and a single binary explanatory variable

Lab Code

```
require(mosaic)
require(mosaicData)
require(lmtest)
cols =  trellis.par.get()$superpose.symbol$col

data(Whickham)
Whickham = Whickham %>%
  mutate(isAlive = 2 - as.numeric(outcome))

logm = glm(isAlive ~ age + smoker, data=Whickham, family=binomial)
summary(logm)

myplot = xyplot(jitter(isAlive) ~ age, groups=smoker, data=Whickham, alpha=0.5, pch=19, cex=2, ylab="isAlive")
fit.outcome = makeFun(logm)
plotFun(fit.outcome(age=x, smoker="Yes") ~ x, lwd=3, plot=myplot, add=TRUE)
plotFun(fit.outcome(age=x, smoker="No") ~ x, col=cols[2], lwd=3, add=TRUE)

Whickham = Whickham %>%
  mutate(ageGroup = cut(age, breaks=10))

favstats(~isAlive | ageGroup, data=Whickham)
# print(myplot)
binned.y = mean(~isAlive | ageGroup, data=Whickham)
binned.x = mean(~age | ageGroup, data=Whickham)
binplot = xyplot(binned.y ~ binned.x, cex=2, pch=19, col="orange", lwd=3)
plotFun(fit.outcome(age=x, smoker="Yes") ~ x, lwd=3, add=TRUE, plot=binplot)
plotFun(fit.outcome(age=x, smoker="No") ~ x, col=cols[2], lwd=3, add=TRUE)
xyplot(logit(binned.y) ~ binned.x, pch=19, cex=2, col="orange")

Whickham = Whickham %>%
  mutate(logm.link = predict(logm, type="link"))

ladd(with(subset(Whickham, smoker=="Yes"), panel.xyplot(age, logm.link, col=cols[1], type="l")))
ladd(with(subset(Whickham, smoker=="No"), panel.xyplot(age, logm.link, col=cols[2], type="l")))

exp(confint(logm))
logLik(logm)
pi = fitted.values(logm)
likelihood = ifelse(Whickham$isAlive == 1, pi, 1 - pi)
log(prod(likelihood))
lrtest(logm)

linteract = glm(isAlive ~ age + smoker + age*smoker, data=Whickham, family=binomial)
summary(linteract)
lquad = glm(isAlive ~ age + smoker + age*smoker + I(age^2) + I(age^2):smoker, data=Whickham, family=binomial)
summary(lquad)

lrtest(logm, linteract, lquad)

print(myplot)
fit.qalive = makeFun(lquad)
plotFun(fit.outcome(age=x, smoker="Yes") ~ x, add=TRUE, lty=2)
plotFun(fit.outcome(age=x, smoker="No") ~ x, col=cols[2], add=TRUE, lty=2)
plotFun(fit.qalive(age=x, smoker="Yes") ~ x, add=TRUE)
plotFun(fit.qalive(age=x, smoker="No") ~ x, col=cols[2], add=TRUE)

MedGPA = read.csv("http://www.math.smith.edu/~bbaumer/mth247/MedGPA.csv")
logm = glm(Acceptance ~ Sex, data=MedGPA, family=binomial)
summary(logm)

two.way = tally(~Acceptance | Sex, data=MedGPA, format="count")
two.way

fit.accept = makeFun(logm)
fit.accept(Sex="M")
fit.accept(Sex="F")
```

```
oddsRatio(two.way)
# Since the coefficients is negative, we add a negative here to match the 2-way table
exp(-coef(logm))
chisq.test(two.way[1:2,1:2], correct=FALSE)

Whickham = Whickham %>%
  mutate(isAlive = 2 - as.numeric(outcome))
logm = glm(isAlive ~ age + smoker, data=Whickham, family=binomial)
summary(logm)

plotPoints(jitter(isAlive) ~ age, groups=smoker, data=Whickham, alpha=0.3, pch=19, cex=2,
           ylab="Probability of Being Alive (units)",
           xlab="Age (years)", main="Whickham Study Outcomes",
           sub=paste("Number of Cases = ", nrow(Whickham)),
           auto.key=TRUE)
fit.outcome = makeFun(logm)
plotFun(fit.outcome(age=x, smoker="Yes") ~ x, add=TRUE)
plotFun(fit.outcome(age=x, smoker="No") ~ x, col=cols[2], add=TRUE)

Whickham = Whickham %>%
  mutate(fitted = fitted.values(logm)) %>%
  mutate(fit.alive = ifelse(fitted >= 0.5, 1, 0))

tally(~isAlive | fit.alive, data=Whickham)
tbl = tally(~isAlive | fit.alive, data=Whickham, format="count")

sum(diag(tbl)) / nrow(Whickham)
mean(~isAlive, data=Whickham)

Whickham = Whickham %>%
   mutate(fit.alive = sample(c(0,1), size=1314, replace=TRUE))
tally(~isAlive | fit.alive, data=Whickham)

X = data.frame(a = runif(10000), b = runif(10000))
require(Hmisc)
rcorrcens(a~b, data=X)
rcorrcens(isAlive ~ fitted.values(logm), data=Whickham)
```