Lecture 2: Intro to Logistic Regression

We will analyze the association between passive smoking and cancer.

First, we create the data set.

```
smoke <- data.frame(y=c(281,228),
                    n=c(491,507),
                    passive=c(1,0))

smoke$n.minus.y <- smoke$n-smoke$y

# use reshape() to get the data in a form for lm()
smoke.long <- reshape(smoke,direction='long',
                      varying=c('y','n.minus.y'),v.names='count',
                      timevar='cancer',times=1:0,
                      drop='n')
```

Linear regression with binary outcome

As in the lecture notes, we can think about modeling the cancer outcome as linear even though it can only take two values, `0` and `1`.

```
smoke.linmod <- lm(cancer ~ passive, weights=count, data=smoke.long)
anova(smoke.linmod)

## Analysis of Variance Table
##
## Response: cancer
##           Df  Sum Sq Mean Sq F value Pr(>F)
## passive    1   3.749   3.749  0.0305 0.8774
## Residuals  2 245.651 122.825

summary(smoke.linmod)

##
## Call:
## lm(formula = cancer ~ passive, data = smoke.long, weights = count)
##
## Weighted Residuals:
##    1.1    2.1    1.0    2.0
##  7.170  8.309 -8.293 -7.512
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.4497     0.4922   0.914    0.457
## passive       0.1226     0.7017   0.175    0.877
##
```

```
## Residual standard error: 11.08 on 2 degrees of freedom
## Multiple R-squared:  0.01503,    Adjusted R-squared:  -0.4775 
## F-statistic: 0.03052 on 1 and 2 DF,  p-value: 0.8774
```

Why does this give us different values for significance tests than SAS? It's because R doesn't make adjustment for the degrees of freedom when you use the `weights=` statement in `lm()`. Notice that we do get the same estimates for the coefficient values and sums of squares though.

If we want to have correct standard errors and degrees of freedom, we need to manually adjust the data set.

```
# we replicate each row of the data set "count" number of times
smoke.longrep <- smoke.long[rep(1:4,smoke.long$count),c('passive','cancer')]
# drops the now meaningless count variable and id

smoke.linmodrep <- lm(cancer ~ passive, data=smoke.longrep)
anova(smoke.linmodrep)
```

```
## Analysis of Variance Table
## 
## Response: cancer
##            Df  Sum Sq Mean Sq F value    Pr(>F)    
## passive     1   3.749  3.7490  15.201 0.0001031 ***
## Residuals 996 245.651  0.2466                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(smoke.linmodrep)
```

```
## 
## Call:
## lm(formula = cancer ~ passive, data = smoke.longrep)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -0.5723 -0.4497  0.4277  0.4277  0.5503 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.44970    0.02206  20.389  < 2e-16 ***
## passive      0.12260    0.03144   3.899 0.000103 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4966 on 996 degrees of freedom
## Multiple R-squared:  0.01503,    Adjusted R-squared:  0.01404 
## F-statistic:  15.2 on 1 and 996 DF,  p-value: 0.0001031
```

These values match up with the values we get from SAS.

To look at the predicted values we use the `predict()` function.

```
predict(smoke.linmod, newdata=list(passive=0:1))

##         1         2
## 0.4497041 0.5723014
```

The first number is the predicted mean of the outcome variable cancer for subjects with `passive=0`, the second is the mean with `passive=1`. Since the coefficient estimates with R are the same as with SAS, these values are also the same.

Logistic regression

```
smoke.logitmod <- glm(cbind(y,n-y) ~ # format for outcome is two columns:
number of successes, number of failures
              passive, # the covariate
            data=smoke, # gives the data set we are using to define variables
in the model
            family=binomial) # tells R to use logistic regression for a
binary outcome
summary(smoke.logitmod)

##
## Call:
## glm(formula = cbind(y, n - y) ~ passive, family = binomial, data = smoke)
##
## Deviance Residuals:
## [1]  0  0
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.20187    0.08928  -2.261 0.023750 *
## passive      0.49311    0.12764   3.863 0.000112 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance:  1.5041e+01  on 1  degrees of freedom
## Residual deviance: -5.0626e-14  on 0  degrees of freedom
## AIC: 17.299
##
## Number of Fisher Scoring iterations: 2
```

So there is a significant positive association between passive smoking and lung cancer.

We can get closer to how SAS treats this data by using the replicated data set.

```
smoke.logitmodrep <- glm(cancer ~ # now we just name the column with the
binary outcome
              passive, # the covariate
            data=smoke.longrep, # gives the data set we are using to define
variables in the model
            family=binomial) # tells R to use logistic regression for a
binary outcome
summary(smoke.logitmodrep)

## 
## Call:
## glm(formula = cancer ~ passive, family = binomial, data = smoke.longrep)
## 
## Deviance Residuals: 
##    Min      1Q  Median      3Q     Max  
## -1.303  -1.093   1.056   1.056   1.264  
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -0.20187    0.08928  -2.261 0.023749 *  
## passive      0.49311    0.12763   3.864 0.000112 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 1383.1  on 997  degrees of freedom
## Residual deviance: 1368.1  on 996  degrees of freedom
## AIC: 1372.1
## 
## Number of Fisher Scoring iterations: 3
```

We can look at odds ratios from either of these model estimates: they are the same coefficient and standard error estimates, but they differ in their log-likelihood values.

```
# odds ratio
exp(coef(smoke.logitmod)[2])

##  passive 
## 1.637406

# and we can get a confidence interval as well using standard errors on the
log-odds scale
V.smoke <- vcov(smoke.logitmod)
V.smoke # covariance matrix of parameter estimates

##               (Intercept)      passive
## (Intercept)  0.007970194 -0.007970194
## passive     -0.007970194  0.016290818

se.passive <- sqrt(diag(V.smoke))[2] # standard error on log-odds scale

alpha <- .05 # significance level
# exponentiate the endpoints on log scale to get the confidence interval on
```

```
the odds ratio scale
exp(coef(smoke.logitmod)[2]+c(-1,1)*qnorm(1-alpha/2)*se.passive)

## [1] 1.275008 2.102809
```

There is a quicker way to get the confidence intervals, however. We can use the built-in `confint()` function, but we have to make sure to use the default version to get the Wald intervals.

```
ci.tab <- confint.default(smoke.logitmod)
ci.tab

##                  2.5 %      97.5 %
## (Intercept) -0.3768438 -0.02688852
## passive      0.2429523  0.74327425

exp(ci.tab[2,])

##    2.5 %   97.5 %
## 1.275008 2.102809
```

Logistic regression with a continuous predictor

We want to look at the effect of age on coronary heart disease.

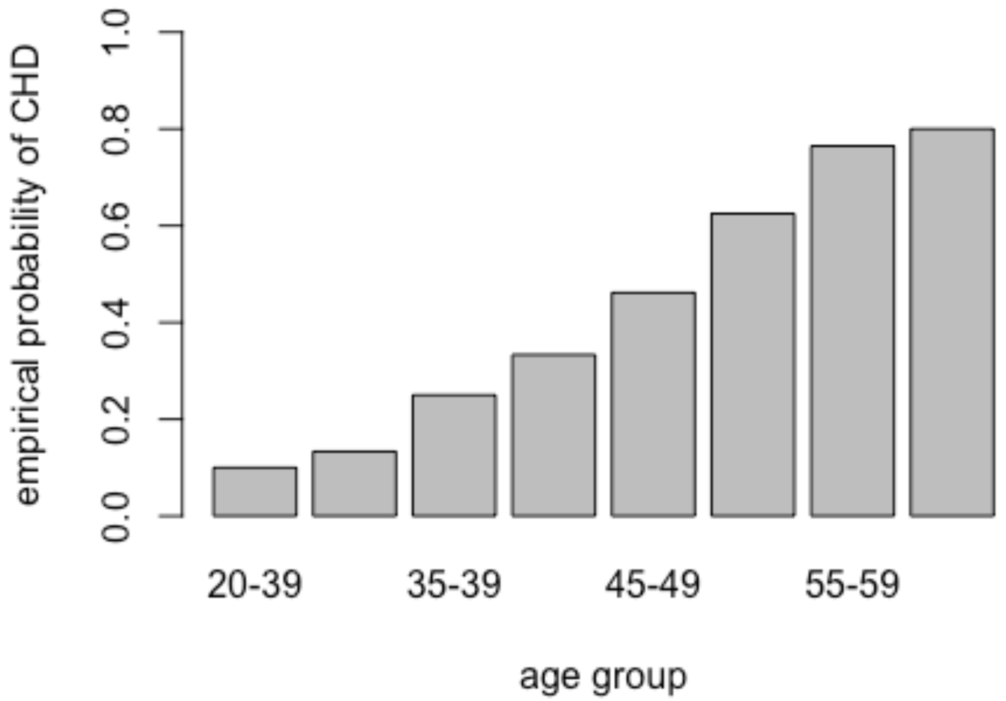
```
head(chdage)

##   id age agegrp chd agectr
## 1  1  20  20-39  No -24.38
## 2  2  23  20-39  No -21.38
## 3  3  24  20-39  No -20.38
## 4  4  25  20-39  No -19.38
## 5  5  25  20-39 Yes -19.38
## 6  6  26  20-39  No -18.38

# look at the association between age group and CHD risk
agechd.tab <- aggregate(I(chd=='Yes') ~ agegrp,data=chdage,FUN=mean)
agechd.tab

##   agegrp I(chd == "Yes")
## 1  20-39       0.1000000
## 2  30-34       0.1333333
## 3  35-39       0.2500000
## 4  40-44       0.3333333
## 5  45-49       0.4615385
## 6  50-54       0.6250000
## 7  55-59       0.7647059
## 8  60-69       0.8000000

barplot(agechd.tab[,2],names.arg=agechd.tab[,1],ylim=c(0,1),
        xlab='age group',ylab='empirical probability of CHD')
```



```
chdfit <- glm(I(chd=='Yes') ~ age, family=binomial,data=chdage)

summary(chdfit)

##
## Call:
## glm(formula = I(chd == "Yes") ~ age, family = binomial, data = chdage)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9718  -0.8456  -0.4576   0.8253   2.2859
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.30945    1.13365  -4.683 2.82e-06 ***
## age          0.11092    0.02406   4.610 4.02e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 136.66  on 99  degrees of freedom
## Residual deviance: 107.35  on 98  degrees of freedom
## AIC: 111.35
##
## Number of Fisher Scoring iterations: 4
```

It’s not really meaningful to interpret the intercept estimate here because it refers to someone with age 0, so we can center age at the mean to address this.

```
# mean age in this sample is
mean(chdage$age)

## [1] 44.38

# create the centered age variable
chdage$agectr <- chdage$age-mean(chdage$age)
chdfit.agectr <- glm(I(chd=='Yes') ~ agectr, family=binomial,data=chdage)
summary(chdfit.agectr)

##
## Call:
## glm(formula = I(chd == "Yes") ~ agectr, family = binomial, data = chdage)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9718  -0.8456  -0.4576   0.8253   2.2859
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.38677    0.23972  -1.613    0.107
## agectr       0.11092    0.02406   4.610 4.02e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 136.66  on 99  degrees of freedom
## Residual deviance: 107.35  on 98  degrees of freedom
## AIC: 111.35
##
## Number of Fisher Scoring iterations: 4
```

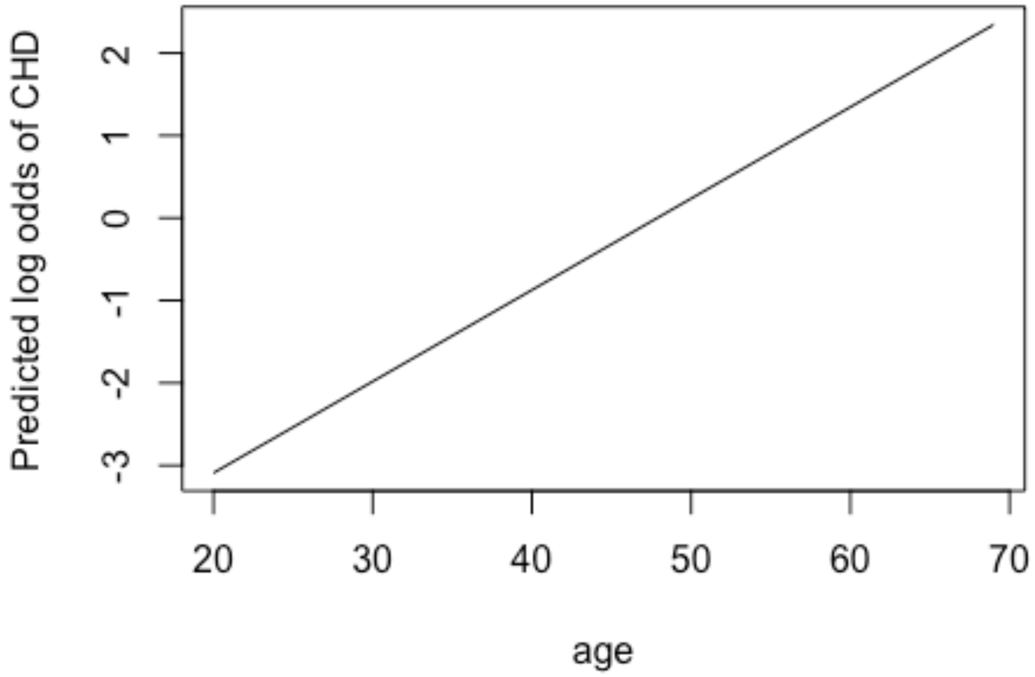
Note that only the intercept estimate changes with centering, not the slope estimate. Now we can interpret the intercept as the log odds of CHD for someone with average age.

Let’s think about using this model to predict someone’s probabilty of CHD. We can use either the centered-age model or the uncentered-age model.

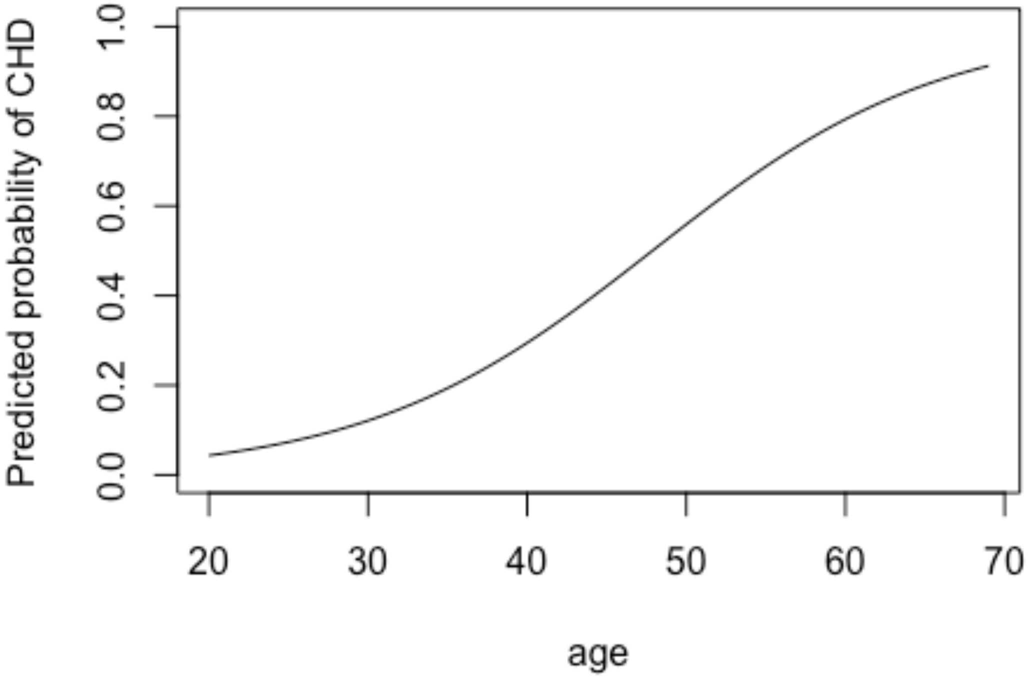
```
agevec <- seq(min(chdage$age),max(chdage$age),length.out=101)

pvec <- predict(chdfit,newdata=list(age=agevec),type='response')
# so we get predictions on the probability scale
# plot on the scale where covariate effect is linear
```

```r
plot(agevec,qlogis(pvec),type='l',xlab='age',
     ylab='Predicted log odds of CHD')
```



```r
# now on probability scale
plot(agevec,pvec,type='l', ylim=c(0,1),xlab='age',
     ylab='Predicted probability of CHD')
```


Predicted probability of CHD
0.0
0.2
0.4
0.6
0.8
1.0
20
30
40
50
60
70
age