sleep

Chris Parrish

January 23, 2016

sleep

reference:
- Cannon, et al., Stat2, chapter 09, example 9.1-9.3
- Cannon, et al., Student R Manual, chapter 9 - logit regression

Import the data.

```
data <- matrix(c(12, 34, 35, 79, 37, 77, 39, 65, 27, 41),
              nrow=2,
              dimnames=list(c("Less than 7 hours", "7 hours or more"),
                            c("14","15","16","17","18")))
data
```

```
##                   14 15 16 17 18
## Less than 7 hours 12 35 37 39 27
## 7 hours or more   34 79 77 65 41
```

`addmargins`

```
addmargins(data, 1)
```

```
##                   14  15  16  17 18
## Less than 7 hours 12  35  37  39 27
## 7 hours or more   34  79  77  65 41
## Sum               46 114 114 104 68
```

`prop.table`

conditional probabilities: $P(Y \mid Age = age)$

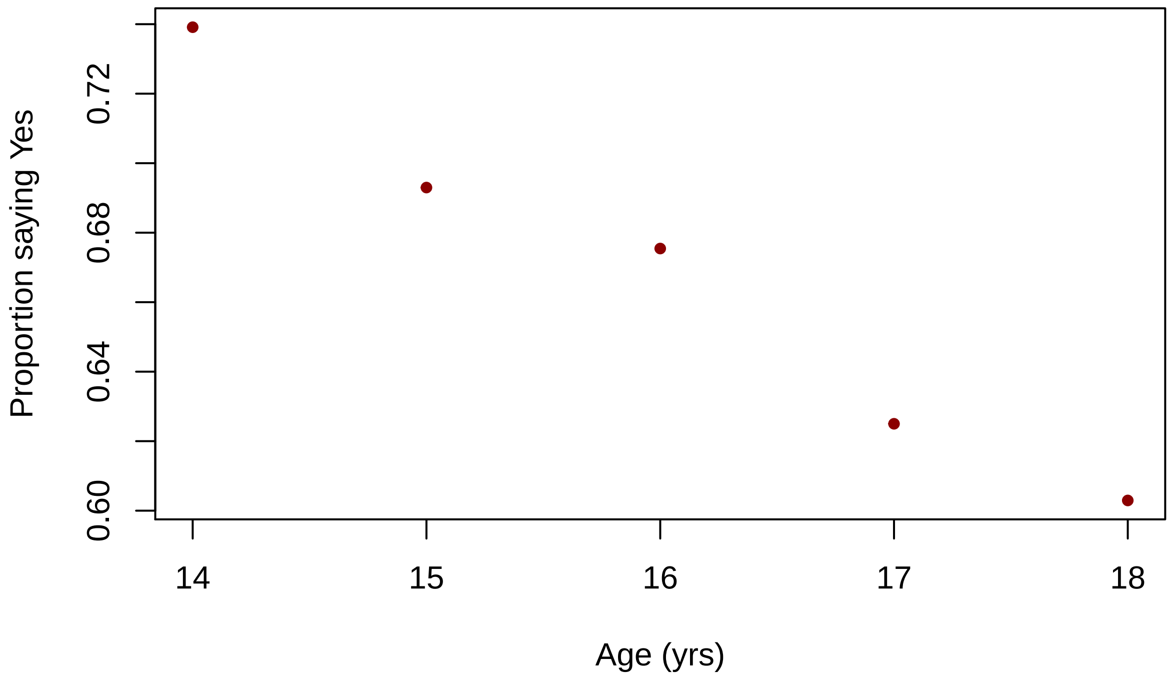
```
sleep.table <- prop.table(data, 2)
sleep.table
```

```
##                          14        15        16    17        18
## Less than 7 hours 0.2608696 0.3070175 0.3245614 0.375 0.3970588
## 7 hours or more   0.7391304 0.6929825 0.6754386 0.625 0.6029412
```

Illustration.

conditional probabilities: $P(Y = yes \mid Age = age)$

```
age <- 14:18
plot(age, sleep.table[2 , ],
     pch=20, col="darkred",
     xlab="Age (yrs)", ylab="Proportion saying Yes")
```



Logistic regression.

$\pi = P(Y = 1)$

$\log(odds) = \log(\frac{\pi}{1-\pi}) = \beta_0 + \beta_1 \cdot X_1$

```
sleep.data <- cbind(data[2,], data[1,])   # yes and no data as columns 1 and 2
sleep.data
```

```
##    [,1] [,2]
## 14   34   12
## 15   79   35
## 16   77   37
## 17   65   39
## 18   41   27
```

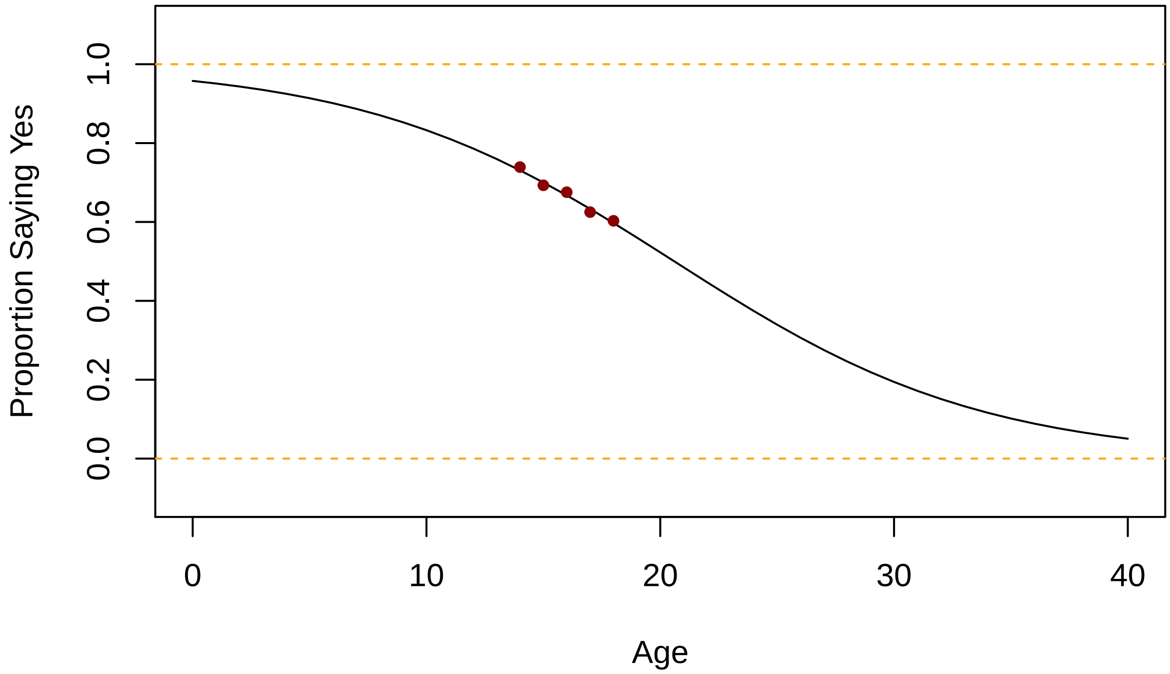
```
sleep.glm <- glm(sleep.data ~ age, family=binomial)
options(show.signif.stars=FALSE)
summary(sleep.glm)
```

```
##
## Call:
## glm(formula = sleep.data ~ age, family = binomial)
##
## Deviance Residuals:
##        14         15         16         17         18
##   0.12505   -0.16779    0.18035   -0.17107    0.09517
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.11864    1.33375   2.338   0.0194
## age          -0.15136    0.08235  -1.838   0.0661
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3.52088  on 4  degrees of freedom
## Residual deviance: 0.11464  on 3  degrees of freedom
## AIC: 27.916
##
## Number of Fisher Scoring iterations: 3
```

Prediction.

```
new.data <- data.frame(age=0:40)
yhat <- predict(sleep.glm, new.data, type="response")
plot(0:40, yhat,
     pch='', type="l", ylim=c(-0.1, 1.1), xlim=c(0,40),
     xlab="Age", ylab="Proportion Saying Yes")
points(age, sleep.table[2, ], pch=20, col="darkred")
abline(h=1, lty=2, col="orange")
abline(h=0, lty=2, col="orange")
```

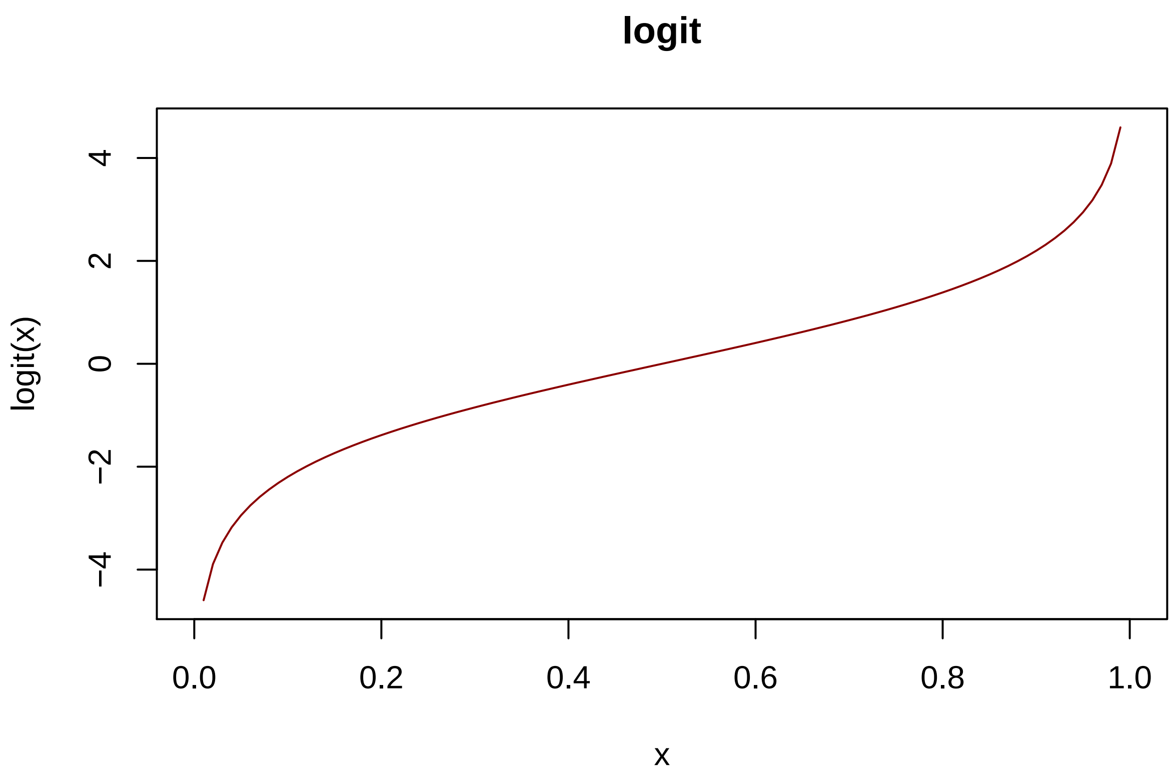


`logit` and `inv.logit` are in the `boot` package.

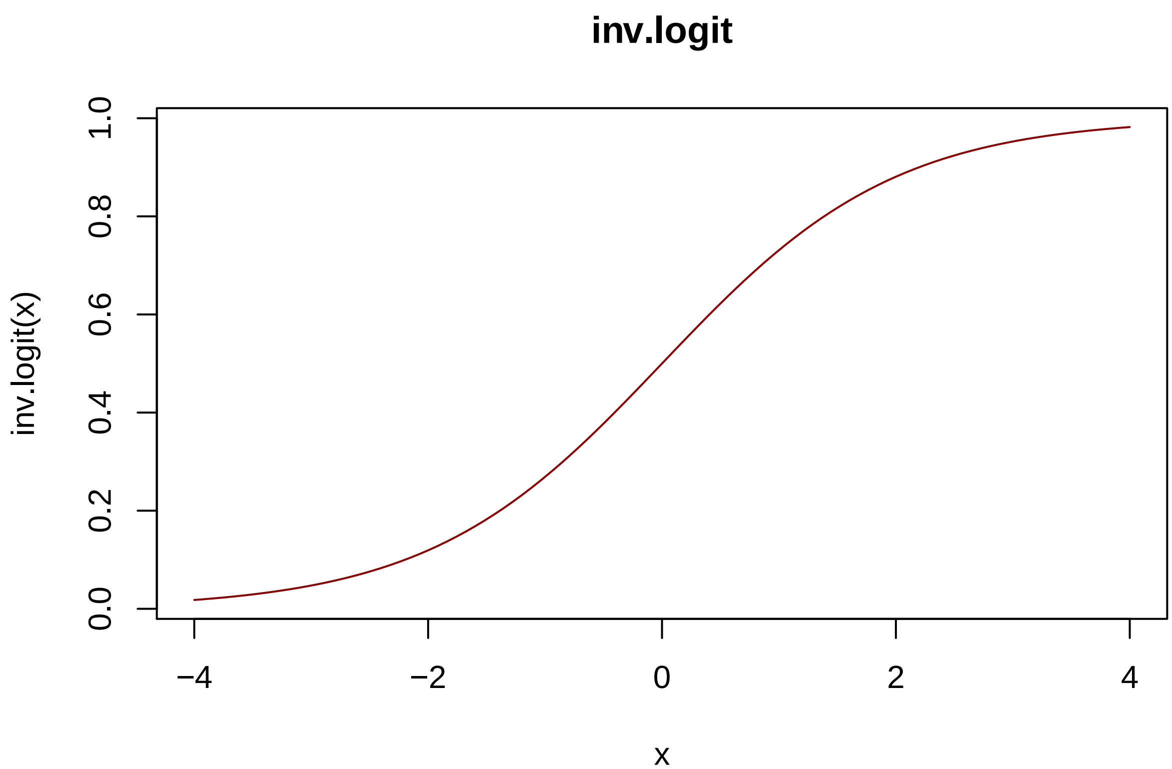
$logit : \pi \rightarrow log.odds \qquad logit(\pi) = \log(\frac{\pi}{1-\pi})$

$inv.logit : log.odds \rightarrow \pi \qquad inv.logit(lo) = \frac{\exp(lo)}{1+\exp(lo)}$

```
library(boot)
curve(logit(x), from=0, to=1,
      col="darkred", main="logit")
```




```r
curve(inv.logit(x), from=-4, to=4,
      col="darkred", main="inv.logit")
```

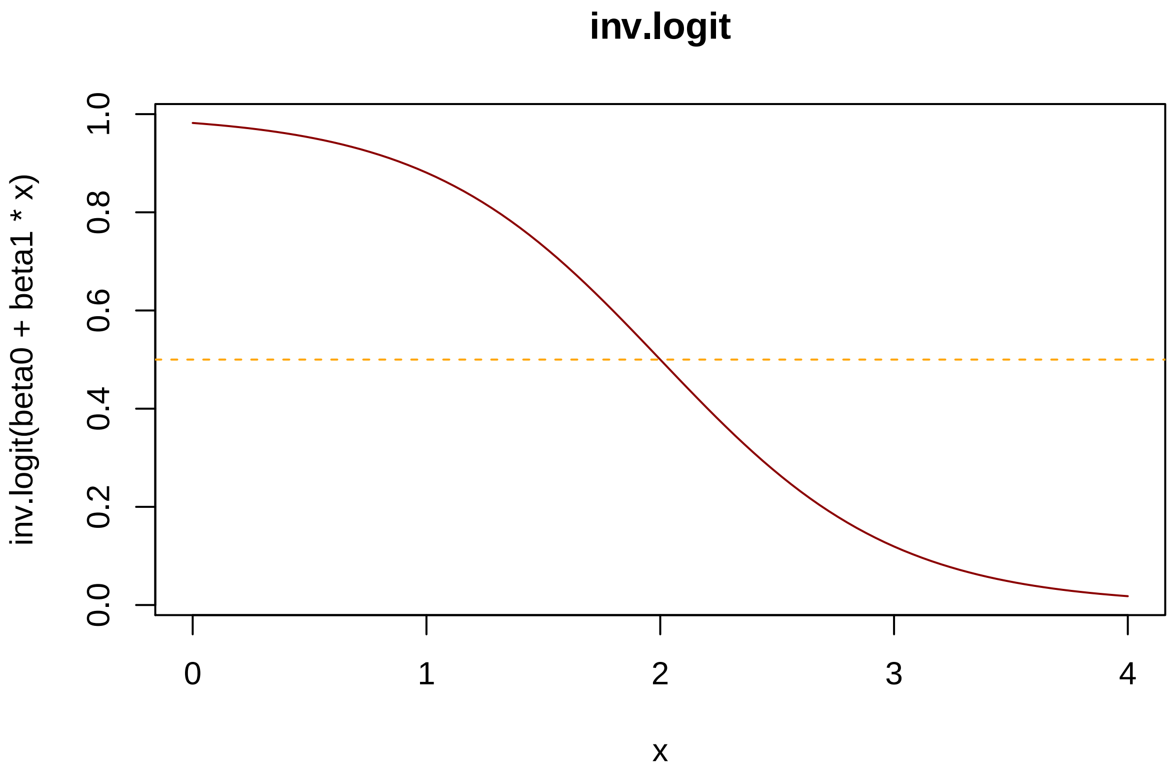


Parameter values determine the shape of the logistic curve.

$midpoint_{\pi=0.5}$ occurs at $x = -\beta_0/\beta_1$

$slope_{\pi=0.5}$ is $\beta_1/4$

```
beta0 <- 4
beta1 <- -2
curve(inv.logit(beta0 + beta1 * x), from=0, to=4,
      col="darkred", main="inv.logit")
abline(h=0.5, col="orange", lty=2)
```



Check the value of $inv.logit(\beta_0 + \beta_1 * x)$ and its derivative at $x = 2$

```
val.at.x.equal.2 <- inv.logit(beta0 + beta1 * 2.00)
val.at.x.equal.2
```

```
## [1] 0.5
```

```
approx.derivative.at.x.equal.2 <- (inv.logit(beta0 + beta1 * 2.01) -
                                     inv.logit(beta0 + beta1 * 2.00)) / 0.01
approx.derivative.at.x.equal.2
```

```
## [1] -0.4999833
```