putts

Chris Parrish

January 24, 2016

putts

reference:
- Cannon, et al., Stat2, chapter 09, example 9.10, 9.16
- Cannon, et al., Student R Manual, chapter 9

Import the data.

```
data <- read.csv("Putts1.csv", header=TRUE)
head(data)
```

```
##   Length Made
## 1      3    1
## 2      3    1
## 3      3    1
## 4      3    1
## 5      3    1
## 6      3    1
```

```
dim(data)
```

```
## [1] 587   2
```

```
data.table <- read.csv("Putts2.csv", header=TRUE)
data.table
```

```
##   Length Made Missed Trials
## 1      3   84     17    101
## 2      4   88     31    119
## 3      5   61     47    108
## 4      6   61     64    125
## 5      7   44     90    134
```

Putt statistics.

```
prop.successes <- with(data.table, Made / (Made + Missed))
odds.success <- prop.successes / (1 - prop.successes)
empirical.logit <- with(data.table, log(Made / Missed))
putting.prowess <- round(rbind(prop.successes, odds.success, empirical.logit), 3)
colnames(putting.prowess) <- 3:7
putting.prowess
```

```
##                     3     4     5      6      7
## prop.successes  0.832 0.739 0.565  0.488  0.328
## odds.success    4.941 2.839 1.298  0.953  0.489
## empirical.logit 1.598 1.043 0.261 -0.048 -0.716
```

Empirical odds ratios.

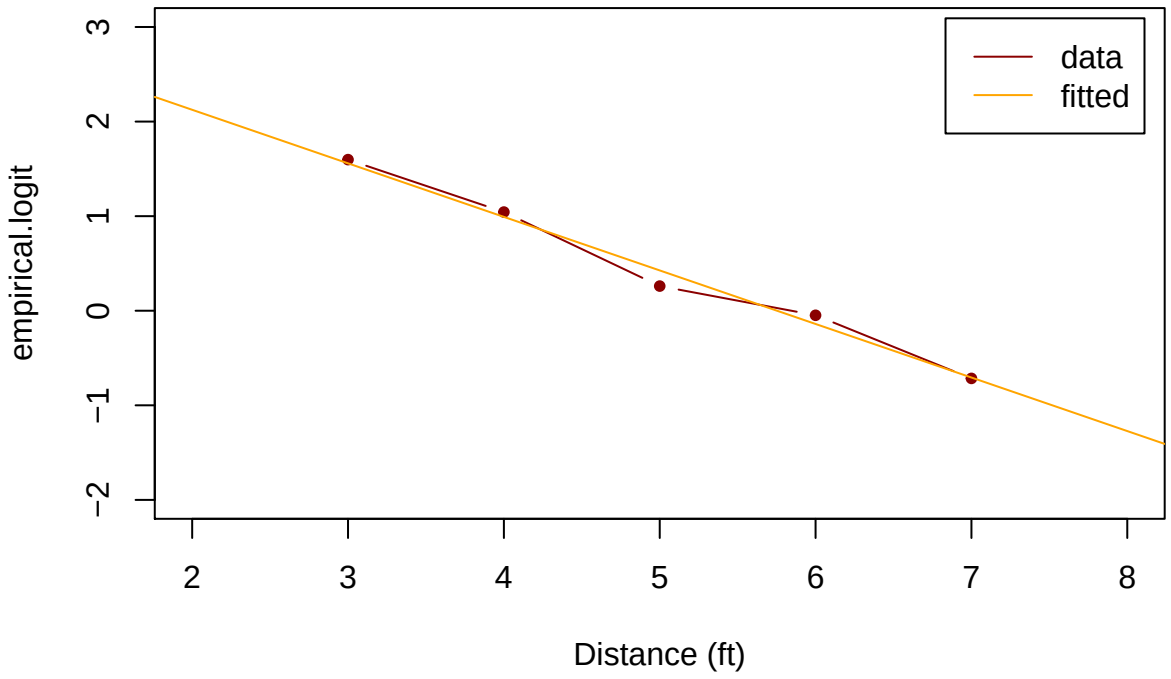
Compare with table on p.469.

```
slope <- empirical.logit[2:5] - empirical.logit[1:4]
empirical.odds.ratio <- exp(slope)
fitted.odds.ratio <- rep(0.5677116, 4)
putt.table <- round(rbind(slope, empirical.odds.ratio, fitted.odds.ratio), 3)
colnames(putt.table) <- c("4 to 3", "5 to 4", "6 to 5", "7 to 6")
putt.table
```

```
##                       4 to 3 5 to 4 6 to 5 7 to 6
## slope                 -0.554 -0.783 -0.309 -0.668
## empirical.odds.ratio   0.575  0.457  0.734  0.513
## fitted.odds.ratio      0.568  0.568  0.568  0.568
```

Illustration.

```
plot(3:7, empirical.logit,
     pch=20, col="darkred", type="b",
     xlim=c(2, 8), ylim=c(-2, 3), xlab="Distance (ft)")
putts.glm <- glm(Made ~ Length, data=data, family=binomial)
abline(putts.glm, col="orange")
legend("topright", legend=c("data", "fitted"),
       lty=1, col=c("darkred", "orange"), inset=0.02)
```



glm.

Compare with table on p.481.

```r
options(show.signif.stars=FALSE)
summary(putts.glm)
```

```
##
## Call:
## glm(formula = Made ~ Length, family = binomial, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8705  -1.1186   0.6181   1.0026   1.4882
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.25684    0.36893   8.828   <2e-16
## Length      -0.56614    0.06747  -8.391   <2e-16
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 800.21  on 586  degrees of freedom
## Residual deviance: 719.89  on 585  degrees of freedom
## AIC: 723.89
##
## Number of Fisher Scoring iterations: 4
```

Odds ratio.

$\beta_1 = log(OR)$

$exp(\beta_1) = OR$

```r
coef(putts.glm)
```

```
## (Intercept)      Length
##   3.2568384  -0.5661417
```

```r
beta1 <- coef(putts.glm)[2]
OR <- exp(beta1)
OR
```

```
##    Length
## 0.5677116
```

CI for odds ratio.

$\beta_1 = log(OR)$

$exp(\beta_1) = OR$

```r
exp(confint(putts.glm))
```

```
## Waiting for profiling to be done...
```

```
##                   2.5 %     97.5 %
## (Intercept) 12.7974573 54.4505172
## Length       0.4960611  0.6464444
```

Formal inference: tests and intervals.

```
summary(putts.glm)
```

```
##
## Call:
## glm(formula = Made ~ Length, family = binomial, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8705  -1.1186   0.6181   1.0026   1.4882
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.25684    0.36893   8.828   <2e-16
## Length      -0.56614    0.06747  -8.391   <2e-16
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 800.21  on 586  degrees of freedom
## Residual deviance: 719.89  on 585  degrees of freedom
## AIC: 723.89
##
## Number of Fisher Scoring iterations: 4
```

HT:

$H_0 : \beta_1 = 0$

$H_a : \beta_1 \neq 0$

```
null.deviance <-800.21
residual.deviance <- 719.89
G <- null.deviance - residual.deviance
df <- 586 - 585
p.value <- 1 - pchisq(G, df=df)
p.value
```

```
## [1] 0
```