trolley

Chris Parrish

June 30, 2016

Contents

trolley 1
data . . . 1
illustration . . . 2
log cumulative odds . . . 3
illustration . . . 4
model . . . 5
map . . . 5
stan . . . 6
compare . . . 7

trolley

reference: McElreath, Statistical Rethinking, chap 11, p.333

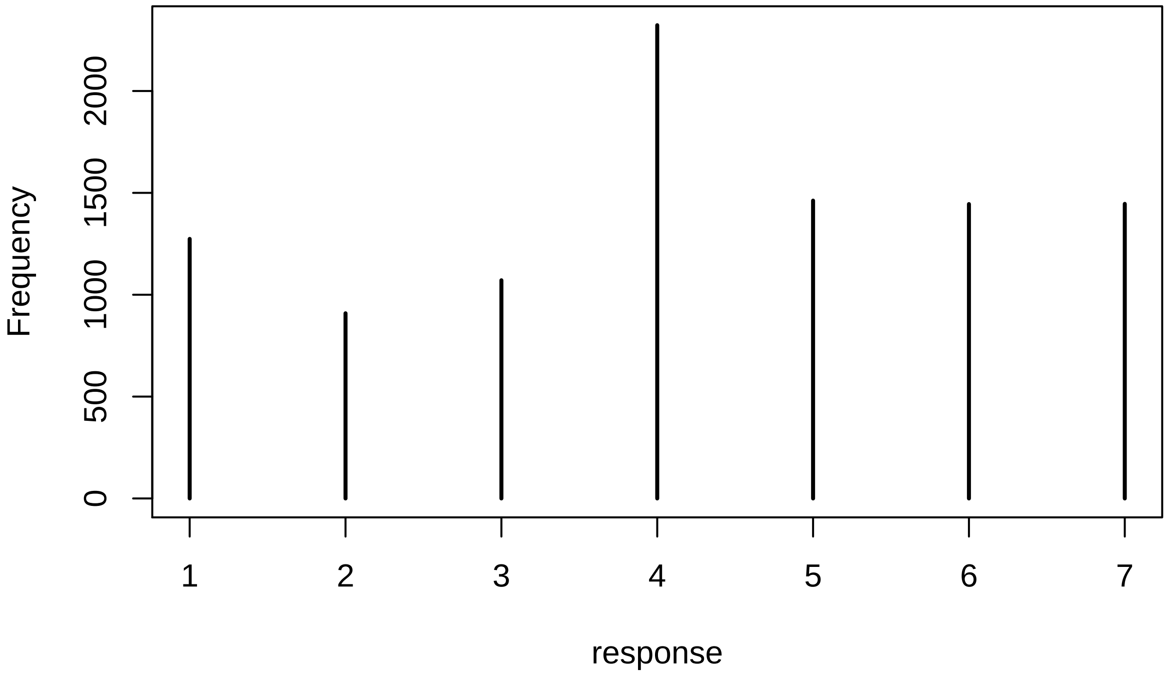
```
library(rethinking)
library(ggplot2)
```

trolley

data

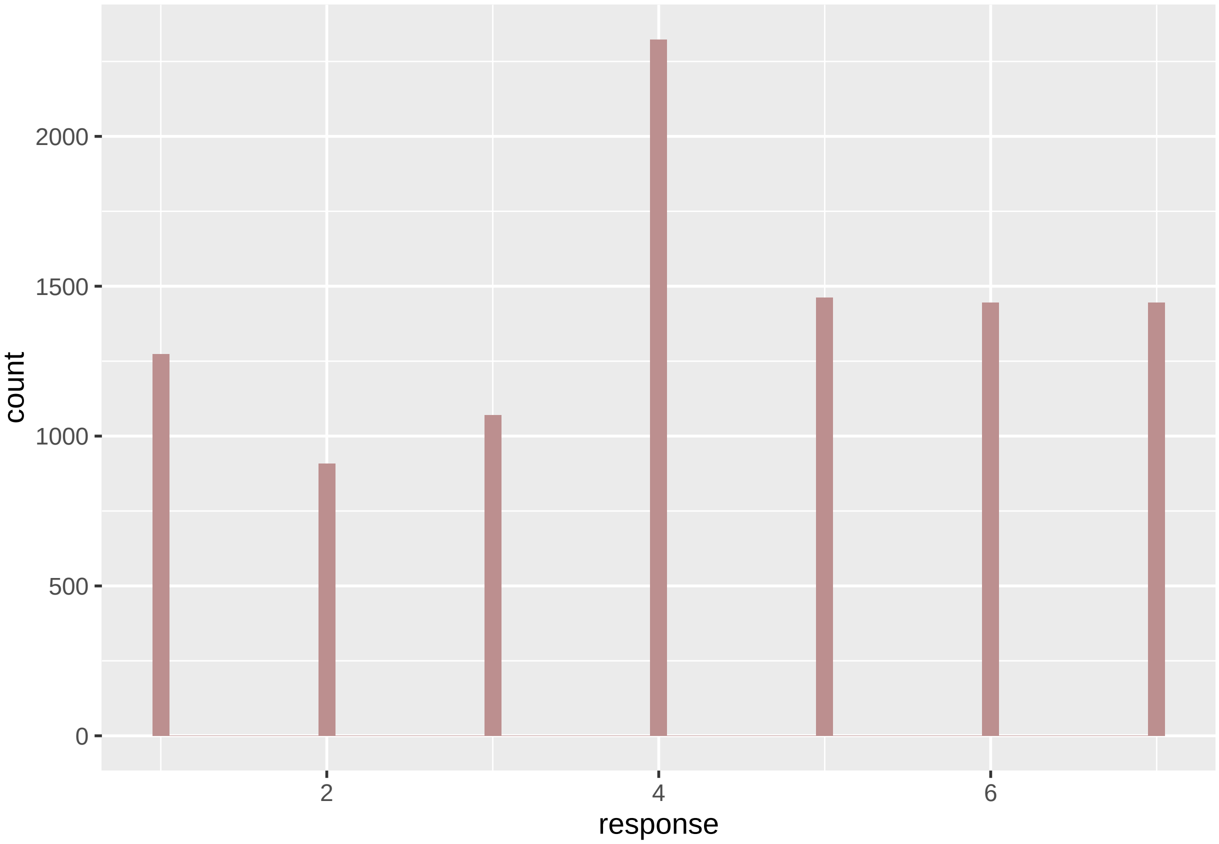
```
## R code 11.1
data(Trolley)
d <- Trolley

## R code 11.2
simplehist( d$response , xlim=c(1,7) , xlab="response" )
```



illustration

```
data <- data.frame(response = d$response)
ggplot(data, aes(response)) +
  geom_histogram(boundary = -0.05, binwidth = 0.1, fill = "rosybrown")
```

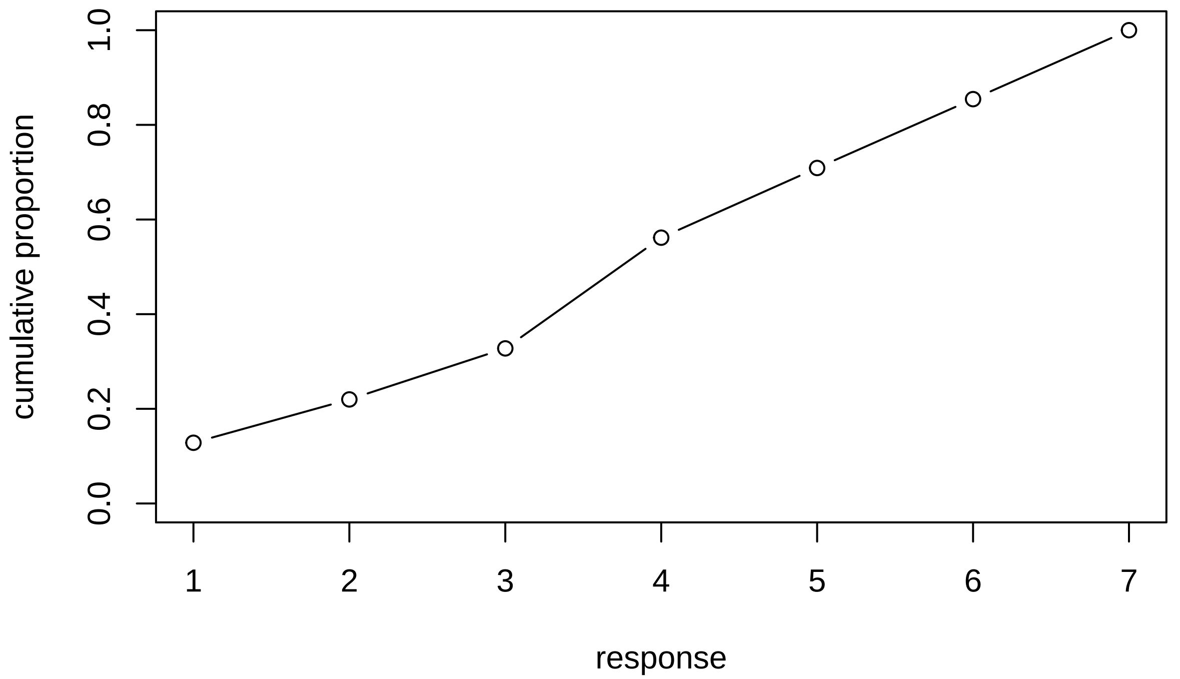


log cumulative odds

```
## R code 11.3
# discrete proportion of each response value
pr_k <- table( d$response ) / nrow(d)

# cumsum converts to cumulative proportions
cum_pr_k <- cumsum( pr_k )

# plot
plot( 1:7 , cum_pr_k , type="b" , xlab="response" ,
ylab="cumulative proportion" , ylim=c(0,1) )
```

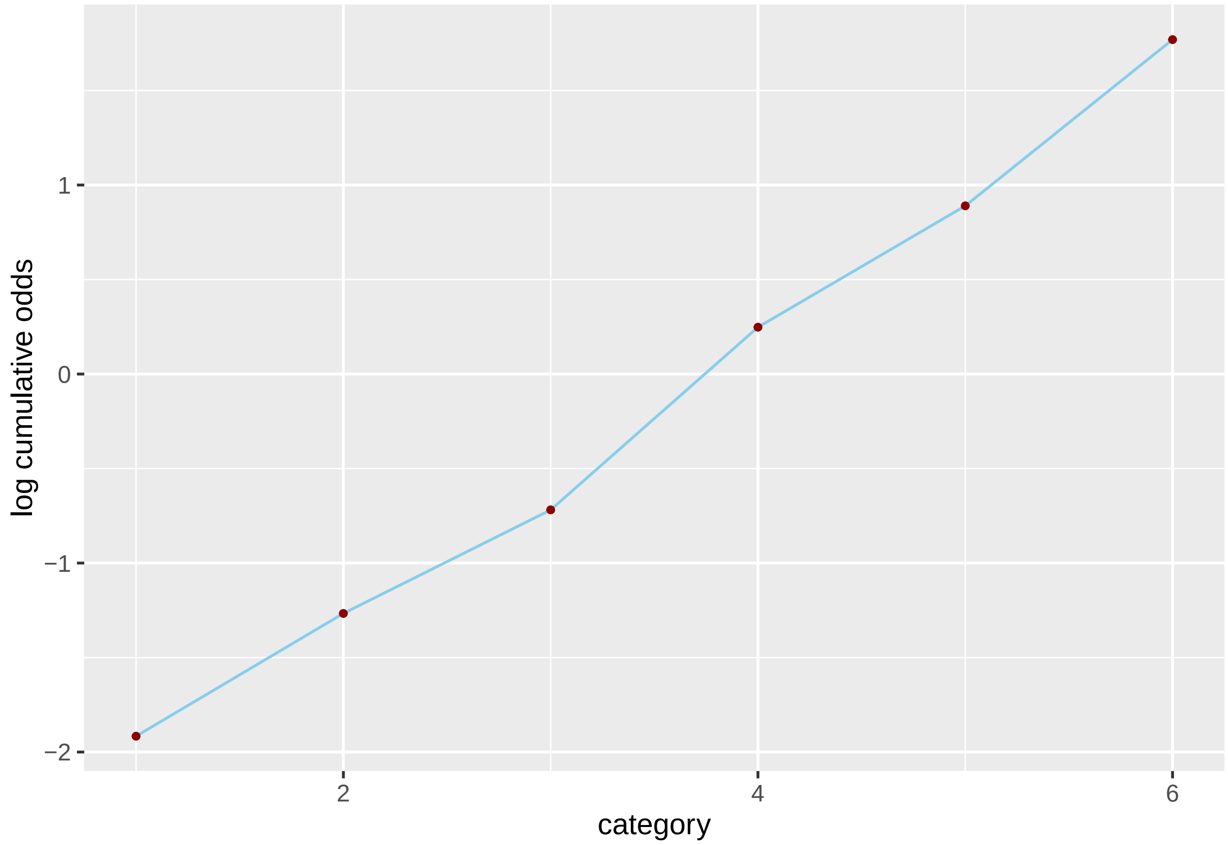


```
## R code 11.4
logit <- function(x) log(x/(1-x)) # convenience function
( lco <- logit( cum_pr_k ) )
```

```
##          1          2          3          4          5          6 
## -1.9160912 -1.2666056 -0.7186340  0.2477857  0.8898637  1.7693809 
##          7 
##        Inf
```

illustration

```
data <- data.frame(x = 1:7, lco = lco)
data <- data[-7, ]              # last logit value is infinity
ggplot(data, aes(x, lco)) +
  geom_line(color = "skyblue") +
  geom_point(shape = 20, color = "darkred") +
  labs(x = "category", y = "log cumulative odds")
```



model

$$
\begin{aligned}
R_i &\sim Ordered(p) \\
logit(p_k) &= \alpha_k \\
\alpha_k &\sim Normal(0, 10)
\end{aligned}
$$

map

```
## R code 11.5
m11.1 <- map(
    alist(
        response ~ dordlogit( phi , c(a1,a2,a3,a4,a5,a6) ),
        phi <- 0,
        c(a1,a2,a3,a4,a5,a6) ~ dnorm(0,10)
    ) ,
    data=d ,
    start=list(a1=-2,a2=-1,a3=0,a4=1,a5=2,a6=2.5) )

## R code 11.6
precis(m11.1)
```

```
##     Mean StdDev  5.5% 94.5%
## a1 -1.92   0.03 -1.96 -1.87
## a2 -1.27   0.02 -1.31 -1.23
## a3 -0.72   0.02 -0.75 -0.68
## a4  0.25   0.02  0.22  0.28
## a5  0.89   0.02  0.85  0.93
## a6  1.77   0.03  1.72  1.81
```

```
## R code 11.7
logistic(coef(m11.1))
```

```
##        a1        a2        a3        a4        a5        a6
## 0.1283005 0.2198398 0.3276948 0.5616311 0.7088609 0.8543786
```

stan

```
## R code 11.8
# note that data with name 'case' not allowed in Stan
# so will pass pruned data list
m11.1stan <- map2stan(
    alist(
        response ~ dordlogit( phi , cutpoints ),
        phi <- 0,
        cutpoints ~ dnorm(0,10)
    ) ,
    data=list(response=d$response),
    start=list(cutpoints=c(-2,-1,0,1,2,2.5)) ,
    chains=2 , cores=2 )
```

```
##
## SAMPLING FOR MODEL 'response ~ dordlogit(phi, cutpoints)' NOW (CHAIN 1).
## WARNING: No variance estimation is
##          performed for num_warmup < 20
##
##
## Chain 1, Iteration: 1 / 1 [100%]  (Sampling)
##  Elapsed Time: 5e-06 seconds (Warm-up)
##                0.00555 seconds (Sampling)
##                0.005555 seconds (Total)
```

```
## Computing WAIC
```

```
## Constructing posterior predictions
```

```
## [ 200 / 2000 ]
[ 400 / 2000 ]
[ 600 / 2000 ]
[ 800 / 2000 ]
[ 1000 / 2000 ]
[ 1200 / 2000 ]
[ 1400 / 2000 ]
[ 1600 / 2000 ]
[ 1800 / 2000 ]
[ 2000 / 2000 ]
```

```
# need depth=2 to show vector of parameters
precis(m11.1stan,depth=2)
```

```
##                Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
## cutpoints[1] -1.92   0.03      -1.96      -1.87  1728    1
## cutpoints[2] -1.27   0.02      -1.31      -1.23  2000    1
## cutpoints[3] -0.72   0.02      -0.75      -0.68  2000    1
## cutpoints[4]  0.25   0.02       0.21       0.28  2000    1
## cutpoints[5]  0.89   0.02       0.85       0.92  2000    1
## cutpoints[6]  1.77   0.03       1.73       1.82  2000    1
```

```
## R code 11.9
( pk <- dordlogit( 1:7 , 0 , coef(m11.1) ) )
```

```
## [1] 0.12830051 0.09153931 0.10785502 0.23393627 0.14722982 0.14551766
## [7] 0.14562142
```

```
## R code 11.10
sum( pk*(1:7) )
```

```
## [1] 4.199294
```

```
## R code 11.11
( pk <- dordlogit( 1:7 , 0 , coef(m11.1)-0.5 ) )
```

```
## [1] 0.08195550 0.06401015 0.08221206 0.20910054 0.15897033 0.18438530
## [7] 0.21936612
```

```
## R code 11.12
sum( pk*(1:7) )
```

```
## [1] 4.72974
```

```
## R code 11.13
m11.2 <- map(
    alist(
        response ~ dordlogit( phi , c(a1,a2,a3,a4,a5,a6) ) ,
        phi <- bA*action + bI*intention + bC*contact,
        c(bA,bI,bC) ~ dnorm(0,10),
        c(a1,a2,a3,a4,a5,a6) ~ dnorm(0,10)
    ) ,
    data=d ,
    start=list(a1=-1.9,a2=-1.2,a3=-0.7,a4=0.2,a5=0.9,a6=1.8) )

## R code 11.14
m11.3 <- map(
    alist(
        response ~ dordlogit( phi , c(a1,a2,a3,a4,a5,a6) ) ,
        phi <- bA*action + bI*intention + bC*contact +
            bAI*action*intention + bCI*contact*intention ,
        c(bA,bI,bC,bAI,bCI) ~ dnorm(0,10),
        c(a1,a2,a3,a4,a5,a6) ~ dnorm(0,10)
    ) ,
    data=d ,
    start=list(a1=-1.9,a2=-1.2,a3=-0.7,a4=0.2,a5=0.9,a6=1.8) )
```

compare

```
## R code 11.15
coeftab(m11.1,m11.2,m11.3)
```

```
##      m11.1   m11.2   m11.3
## a1     -1.92   -2.84   -2.63
## a2     -1.27   -2.15   -1.94
## a3     -0.72   -1.57   -1.34
## a4      0.25   -0.55   -0.31
## a5      0.89    0.12    0.36
## a6      1.77    1.02    1.27
## bA        NA   -0.71   -0.47
## bI        NA   -0.72   -0.28
## bC        NA   -0.96   -0.33
## bAI       NA      NA   -0.45
## bCI       NA      NA   -1.27
## nobs    9930    9930    9930
```

```
## R code 11.16
compare( m11.1 , m11.2 , m11.3 , refresh=0.1 )
```

```
## Constructing posterior predictions
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
## Constructing posterior predictions
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
## Constructing posterior predictions
```

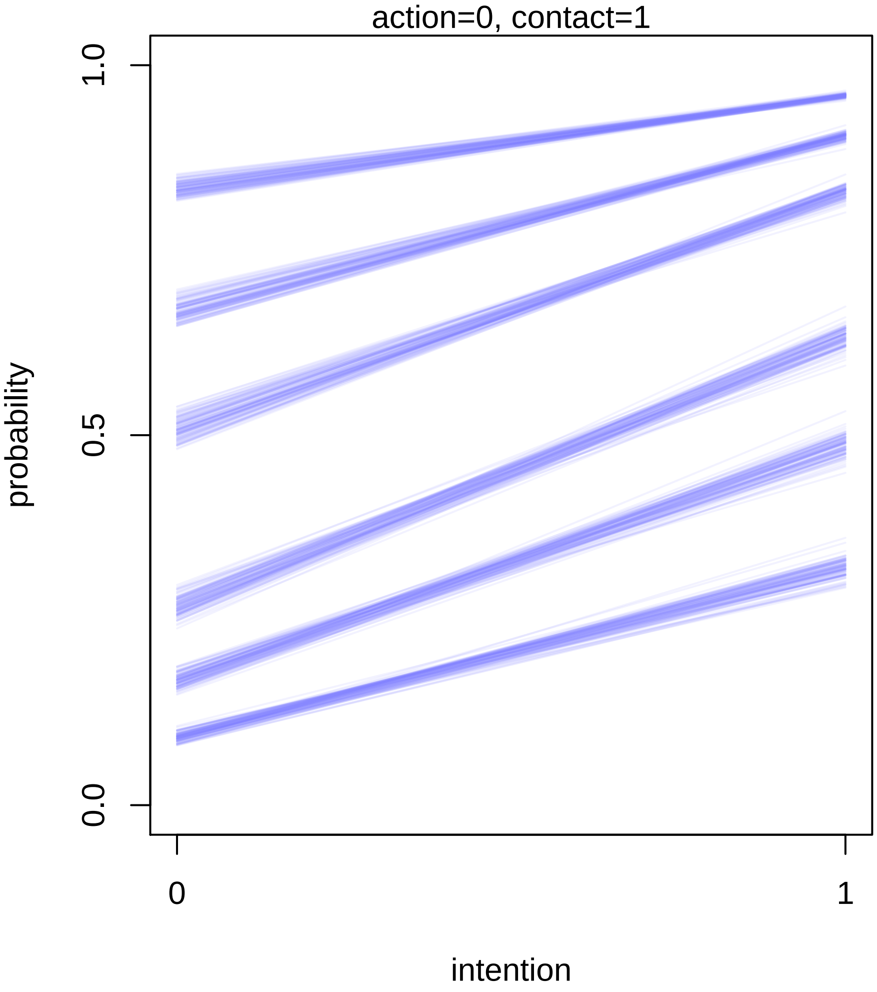
```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
##           WAIC pWAIC dWAIC weight    SE   dSE
## m11.3 36928.9  10.9   0.0      1 81.20    NA
## m11.2 37090.0   9.1 161.1      0 76.24 25.76
## m11.1 37854.7   6.1 925.8      0 57.63 62.71
```

```
## R code 11.17
post <- extract.samples( m11.3 )

## R code 11.18
plot( 1 , 1 , type="n" , xlab="intention" , ylab="probability" ,
    xlim=c(0,1) , ylim=c(0,1) , xaxp=c(0,1,1) , yaxp=c(0,1,2) )

## R code 11.19
kA <- 0     # value for action
kC <- 1     # value for contact
kI <- 0:1   # values of intention to calculate over
for ( s in 1:100 ) {
    p <- post[s,]
    ak <- as.numeric(p[1:6])
    phi <- p$bA*kA + p$bI*kI + p$bC*kC +
        p$bAI*kA*kI + p$bCI*kC*kI
    pk <- pordlogit( 1:6 , a=ak , phi=phi )
    for ( i in 1:6 )
        lines( kI , pk[,i] , col=col.alpha(rangi2,0.1) )
}
mtext( concat( "action=",kA,", contact=",kC ) )
```


action=0, contact=1
probability
1.0
0.5
0.0
0
1
intention