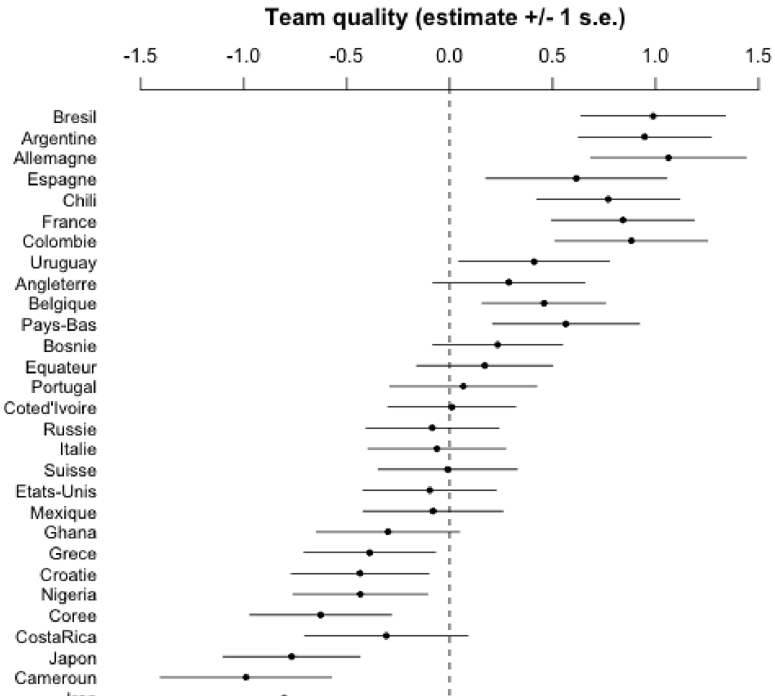
# Bayesian workflow

Andrew Gelman
Dept of Statistics and Dept of Political Science
Columbia University, New York

Department of Biostatistics
University of Washington, Seattle
26 Oct 2017



Team quality (estimate +/- 1 s.e.)
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
Bresil
Argentine
Allemagne
Espagne
Chili
France
Colombie
Uruguay
Angleterre
Belgique
Pays-Bas
Bosnie
Equateur
Portugal
Coted'Ivoire
Russie
Italie
Suisse
Etats-Unis
Mexique
Ghana
Grece
Croatie
Nigeria
Coree
CostaRica
Japon
Cameroun

worldcup2012.txt

```
Bresil 3 Croatie 1
Mexique 1 Cameroun 0
Bresil 0 Mexique 0
Cameroun 0 Croatie 4
Cameroun 1 Bresil 4
Croatie 1 Mexique 3
Espagne 1 Pays-Bas 5
Chili 3 Australie 1
Espagne 0 Chili 2
Australie 2 Pays-Bas 3
Australie 0 Espagne 3
Pays-Bas 2 Chili 0
Colombie 3 Grece 0
Coted'Ivoire 2 Japon 1
Colombie 2 Coted'Ivoire 1
Japon 0 Grece 0
Japon 1 Colombie 4
Grece 2 Coted'Ivoire 1
Uruguay 1 CostaRica 3
Angleterre 1 Italie 2
Uruguay 2 Angleterre 1
```

soccerpowerindex.txt

```
Bresil
Argentine
Allemagne
Espagne
Chili
France
Colombie
Uruguay
Angleterre
Belgique
Pays-Bas
Bosnie
Equateur
Portugal
Coted'Ivoire
Russie
Italie
Suisse
Etats-Unis
Mexique
```

# The (abridged) model in Stan

```
parameters {
  real b;
  real<lower=0> sigma_a;
  real<lower=0> sigma_y;
  vector[nteams] eta_a;
}
transformed parameters {
  vector[nteams] a;
  a = b*prior_score + sigma_a*eta_a;
}
model {
  eta_a ~ normal(0,1);
  sqrt_dif ~ student_t(df, a[team1] - a[team2], sigma_y);
}
```

# Load Stan and data into R

```
library("rstan")
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())

teams <- as.vector(unlist(read.table("soccerpowerindex.txt",
  header=FALSE)))
nteams <- length(teams)
prior_score <- rev(1:nteams)
prior_score <- (prior_score - mean(prior_score))/
  (2*sd(prior_score))
data2014 <- read.table("worldcup2014.txt", header=FALSE)
ngames <- nrow (data2014)

team1 <- match (as.vector(data2014[[1]]), teams)
score1 <- as.vector(data2014[[2]])
team2 <- match (as.vector(data2014[[3]]), teams)
score2 <- as.vector(data2014[[4]])

df <- 7
```

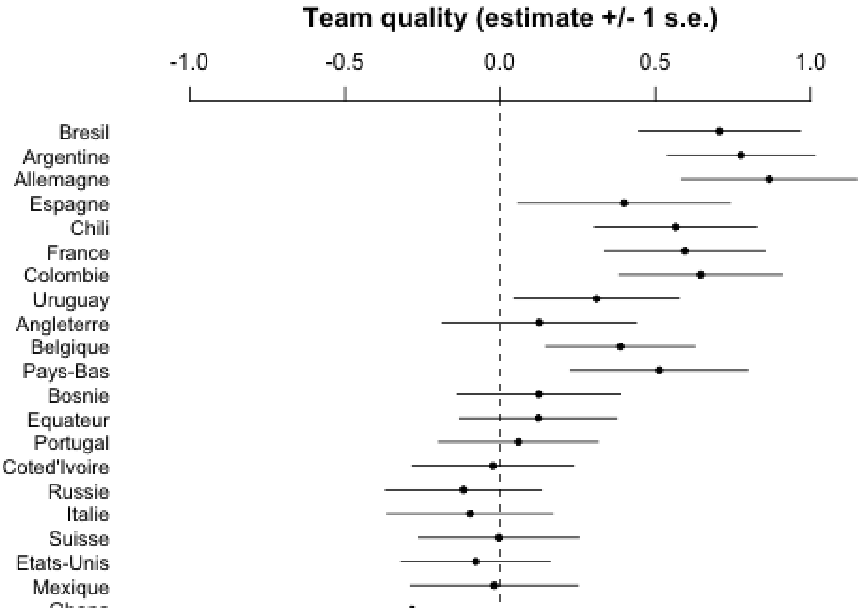
# Fit the model

```
fit <- stan("worldcup_first_try.stan")
print(fit)
```

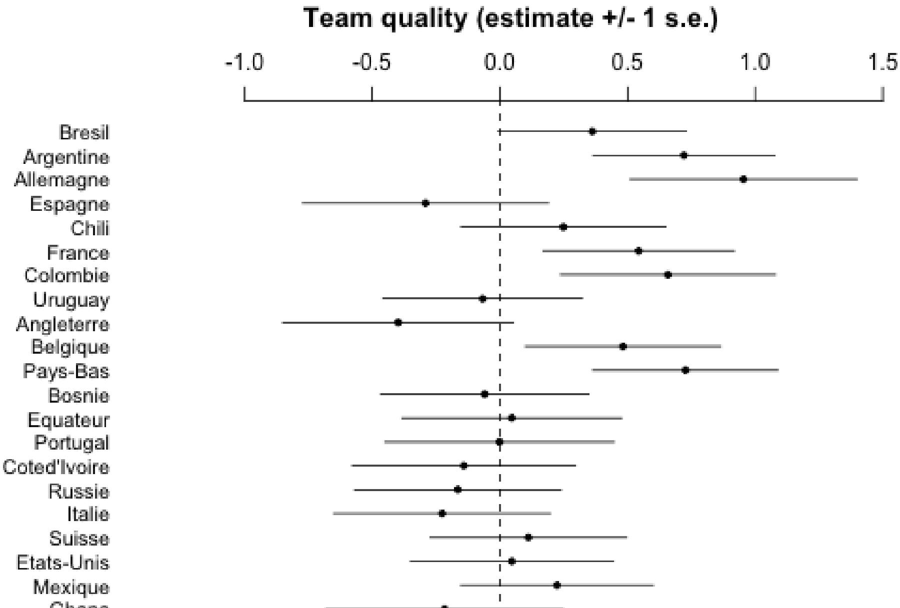
# Check convergence

```
Inference for Stan model: worldcup_first_try.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

           mean se_mean   sd   25%   50%   75% n_eff Rhat
b          0.46    0.00 0.09  0.40  0.46  0.52  1369    1
sigma_a    0.13    0.00 0.07  0.08  0.13  0.18   653    1
sigma_y    0.42    0.00 0.05  0.39  0.42  0.45  1560    1
eta_a[1]  -0.18    0.02 0.84 -0.74 -0.18  0.38  2506    1
eta_a[2]   0.18    0.01 0.82 -0.35  0.18  0.73  3219    1
eta_a[3]   0.58    0.02 0.91  0.00  0.60  1.20  1864    1
eta_a[4]  -0.59    0.02 1.00 -1.28 -0.62  0.10  2284    1
eta_a[5]   0.03    0.02 0.88 -0.54  0.00  0.58  3163    1
. . .
```

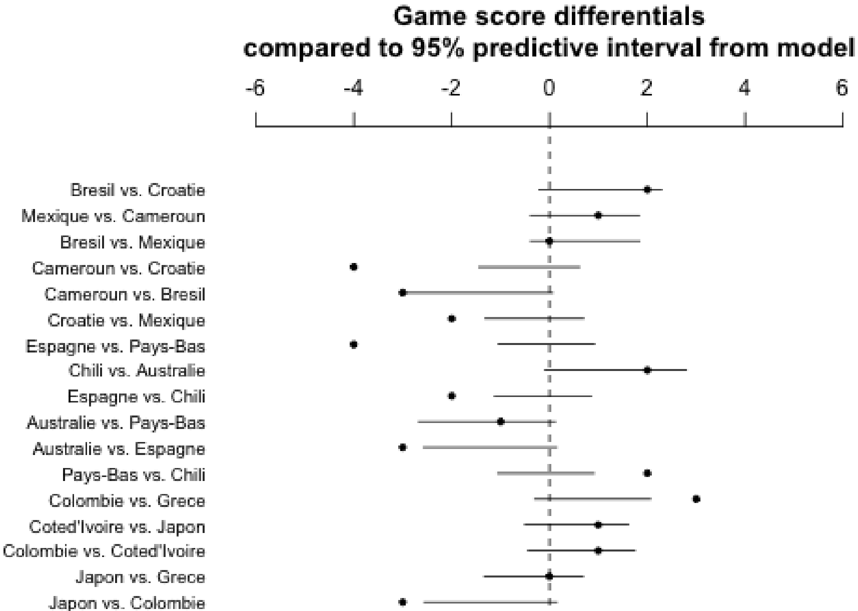
# Graph the estimates



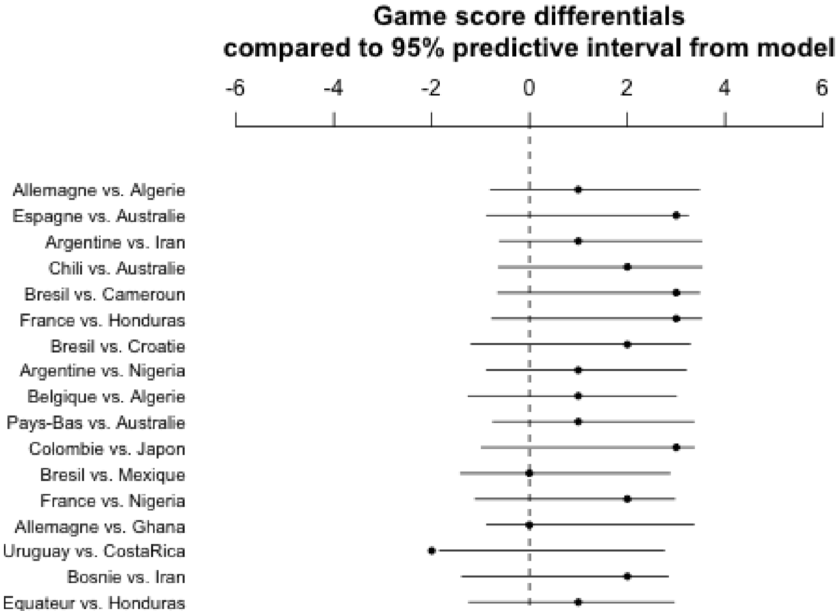
# Compare to model fit without prior rankings

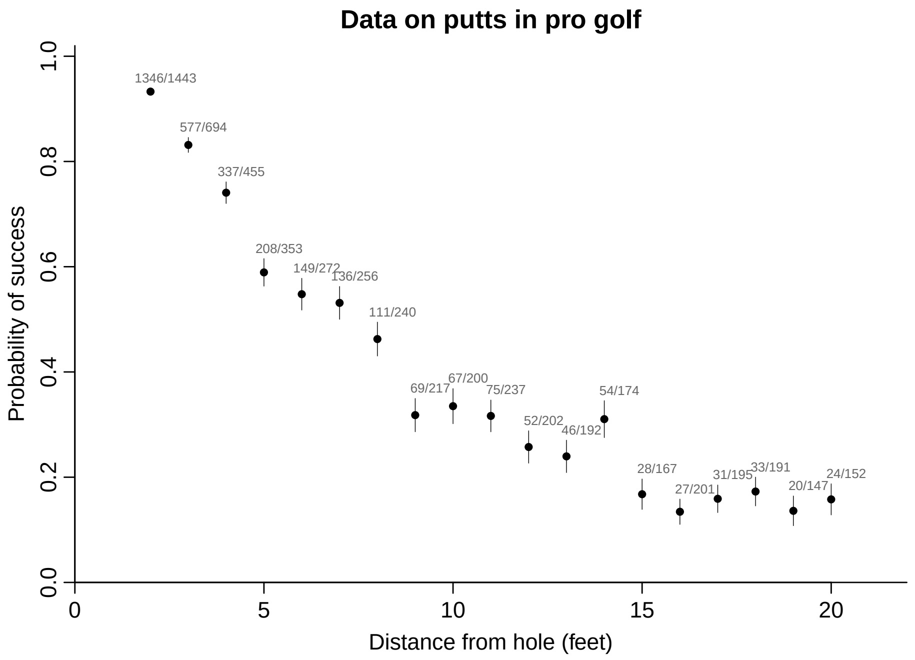


# Compare model to predictions

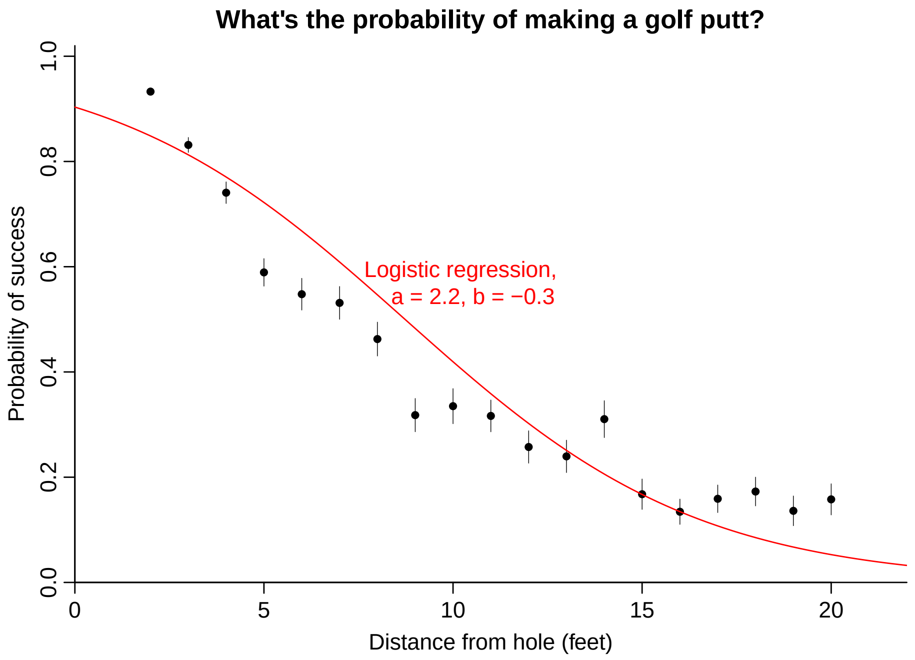


# After finding and fixing a bug

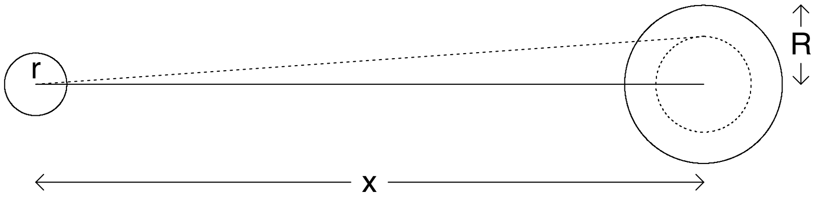


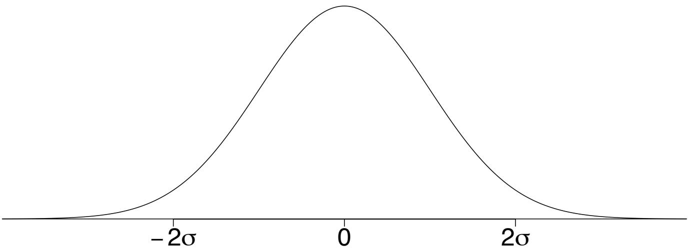


Data on putts in pro golf
Probability of success
Distance from hole (feet)
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
1346/1443
577/694
337/455
208/353
149/272
136/256
111/240
69/217
67/200
75/237
52/202
46/192
54/174
28/167
27/201
31/195
33/191
20/147
24/152


What's the probability of making a golf putt?
Probability of success
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
Distance from hole (feet)
Logistic regression,
a = 2.2, b = −0.3

# Geometry-based model



# Stan code

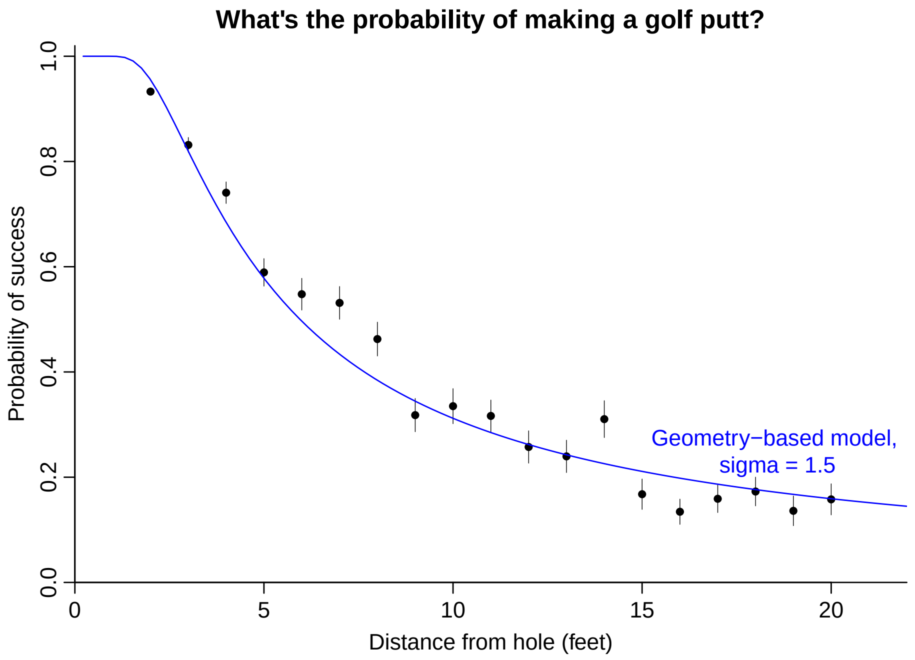
```
data {
  int J;
  int n[J];
  real x[J];
  int y[J];
  real r;
  real R;
}
parameters {
  real<lower=0> sigma;
}
model {
  real p[J];
  p = 2*Phi(asin((R-r)/x) / sigma) - 1;
  y ~ binomial(n, p);
}
```

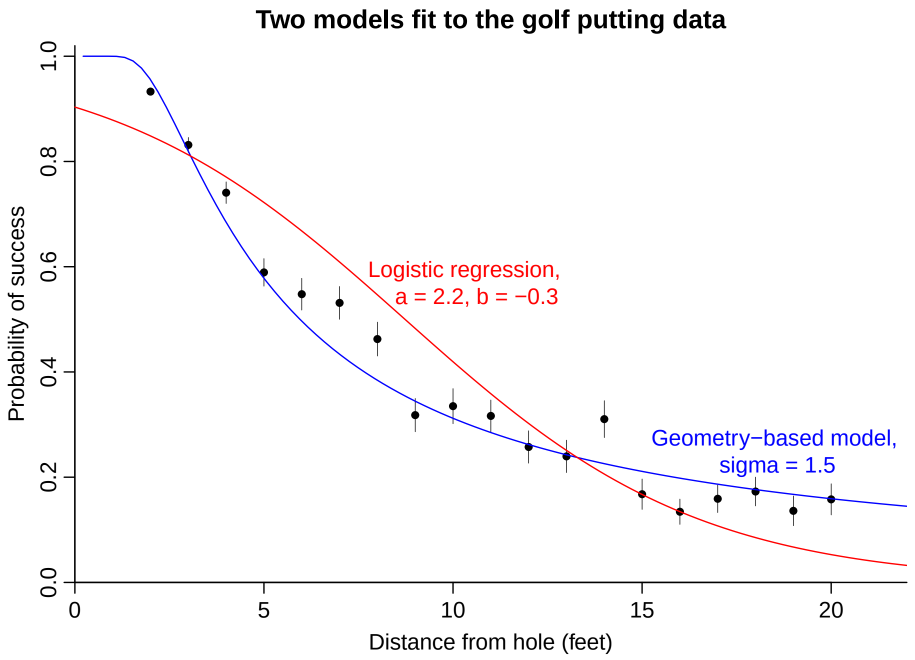
# Fit the model

```
golf <- read.table("golf.txt", header=TRUE, skip=2)
x <- golf$x
y <- golf$y
n <- golf$n
J <- length(y)
r <- (1.68/2)/12
R <- (4.25/2)/12

fit1 <- stan("golf1.stan")
```

# Check convergence

```
> print(fit1)
Inference for Stan model: golf1.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

                  mean se_mean   sd  25%  50%  75% n_eff Rhat
sigma             0.03    0.00 0.00 0.03 0.03 0.03  1692    1
sigma_degrees     1.53    0.00 0.02 1.51 1.53 1.54  1692    1
```


What's the probability of making a golf putt?
Probability of success
Distance from hole (feet)
Geometry−based model,
sigma = 1.5
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20


Two models fit to the golf putting data
Probability of success
0.0
0.2
0.4
0.6
0.8
1.0
Distance from hole (feet)
0
5
10
15
20
Logistic regression,
a = 2.2, b = −0.3
Geometry−based model,
sigma = 1.5

# Birthdays!

Social Science & Medicine 73 (2011) 1246–1248

Contents lists available at SciVerse ScienceDirect

Social Science & Medicine

journal homepage: www.elsevier.com/locate/socscimed

Short report

## Influence of Valentine's Day and Halloween on Birth Timing

Becca R. Levy*, Pil H. Chung, Martin D. Slade

*Yale University, School of Public Health, Division of Social & Behavioral Sciences, 60 College Street, New Haven, CT 06520-8034, United States*

A R T I C L E I N F O

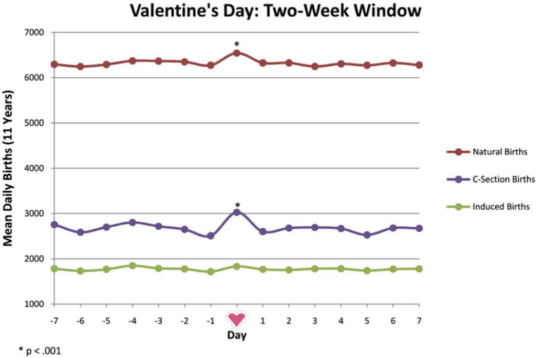
*Article history:*
Available online 28 July 2011

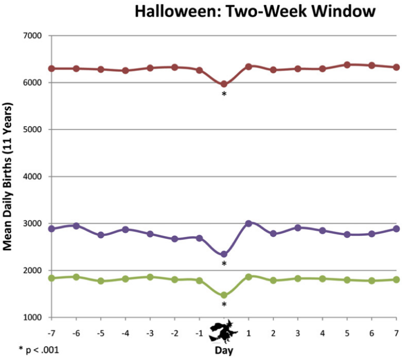
*Keywords:*
United States
Culture
Birth timing
Holidays
Pregnancy
Biocultural
Birth

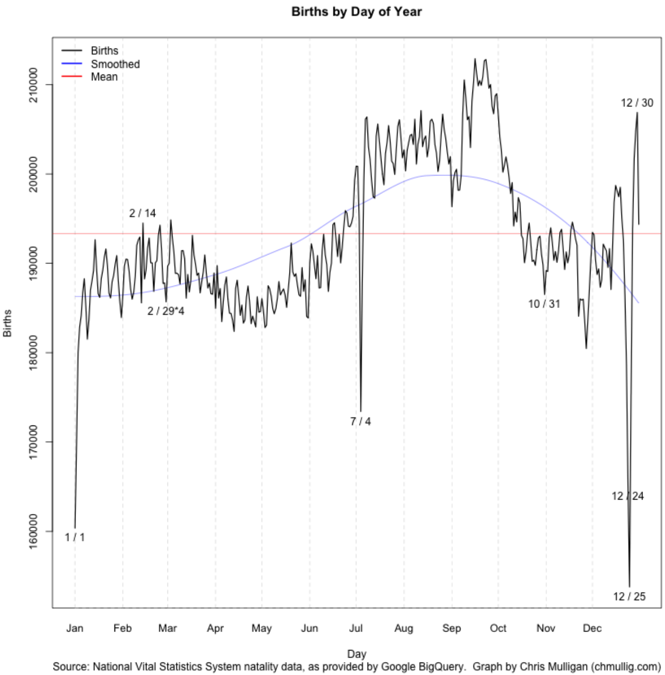
A B S T R A C T

It is known that cultural representations, in the form of stereotypes, can influence functional health. We predicted that the influence of cultural representations, in the form of salient holidays, would extend to birth timing. On Valentine's Day, which conveys positive symbolism, there was a 3.6% increase in spontaneous births and a 12.1% increase in cesarean births. Whereas, on Halloween, which conveys negative symbolism, there was a 5.3% decrease in spontaneous births and a 16.9% decrease in cesarean births. These effects reached significance at $p < .0001$, after adjusting for year and day of the week. The sample was based on birth-certificate information for all births in the United States within one week on either side of each holiday across 11 years. The Valentine's-Day window included 1,676,217 births and the Halloween window included 1,809,304 births. Our findings raise the possibility that pregnant women may be able to control the timing of spontaneous births, in contrast to the traditional assumption, and that scheduled births are also influenced by the cultural representations of the two holidays.

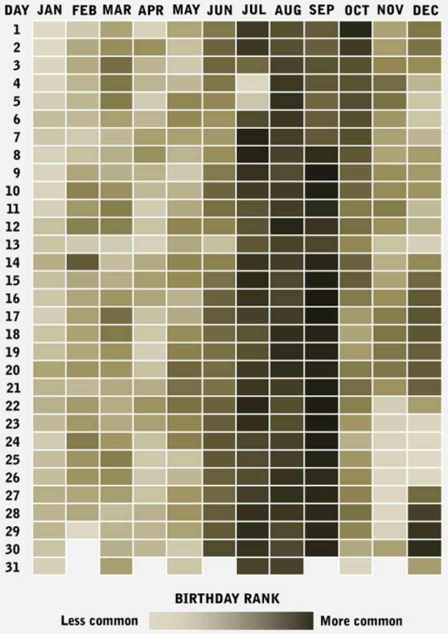
# The published graphs show data from 30 days in the year

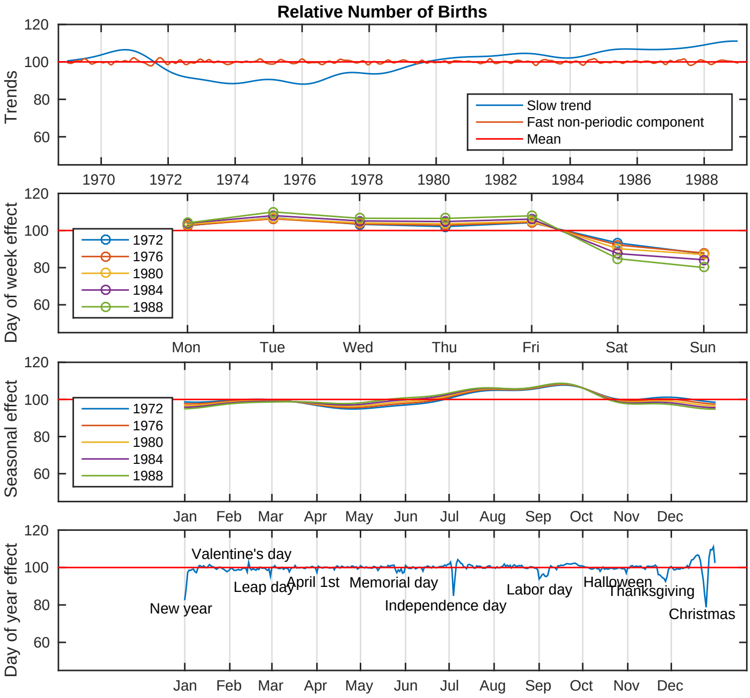


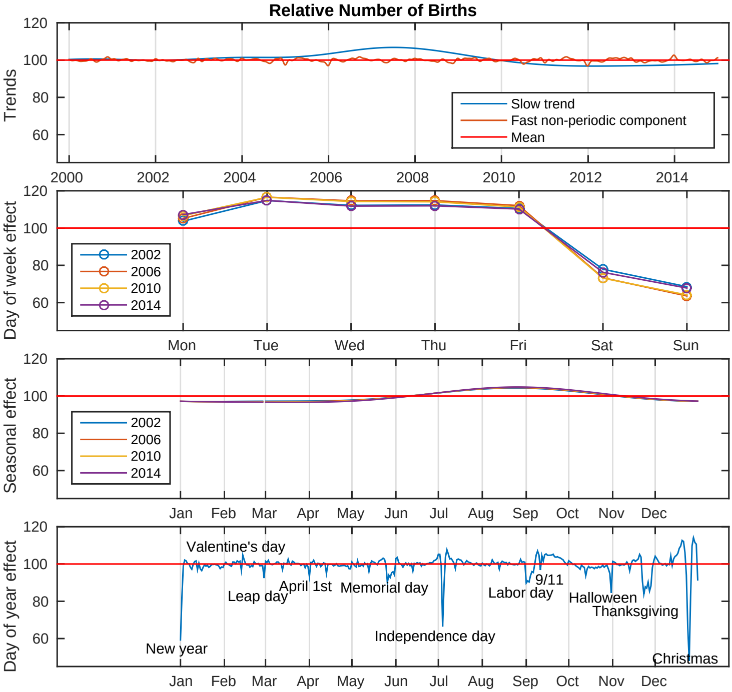

Births by Day of Year
Births
Smoothed
Mean
2 / 14
2 / 29*4
7 / 4
10 / 31
12 / 30
12 / 24
12 / 25
1 / 1
210000
200000
190000
180000
170000
160000
Births
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Day
Source: National Vital Statistics System natality data, as provided by Google BigQuery. Graph by Chris Mulligan (chmullig.com)

## Which Birth Dates Are Most Common?

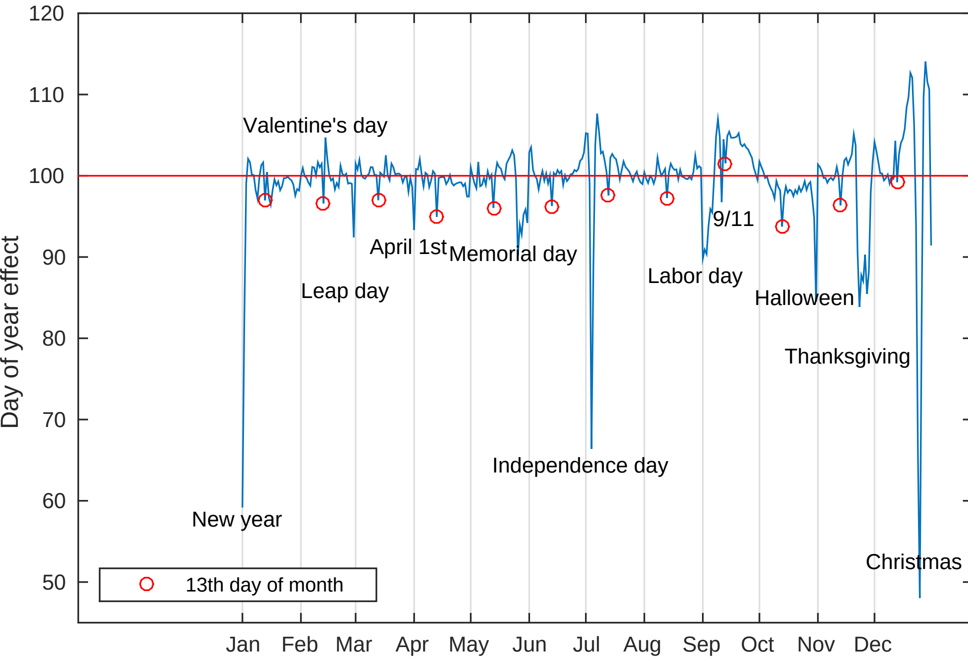




Relative Number of Births
Trends
Slow trend
Fast non-periodic component
Mean
1970
1972
1974
1976
1978
1980
1982
1984
1986
1988
Day of week effect
1972
1976
1980
1984
1988
Mon
Tue
Wed
Thu
Fri
Sat
Sun
Seasonal effect
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Day of year effect
Valentine's day
Leap day
April 1st
Memorial day
Independence day
Labor day
Halloween
Thanksgiving
Christmas
New year
120
100
80
60



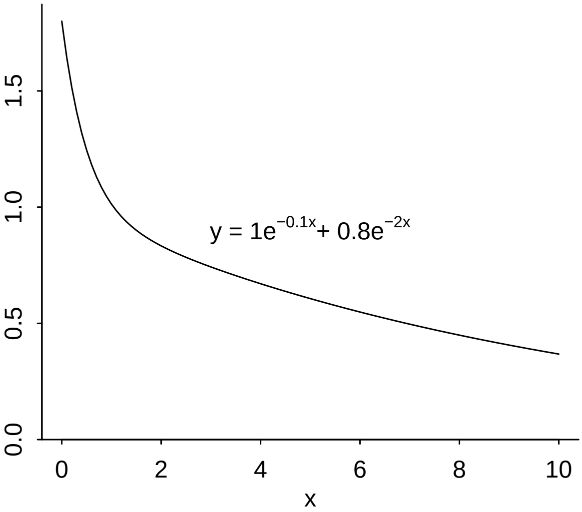
Relative Number of Births
Trends
Slow trend
Fast non-periodic component
Mean
2000
2002
2004
2006
2008
2010
2012
2014
Day of week effect
2002
2006
2010
2014
Mon
Tue
Wed
Thu
Fri
Sat
Sun
Seasonal effect
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Day of year effect
Valentine's day
Leap day
April 1st
Memorial day
9/11
Labor day
Halloween
Thanksgiving
New year
Independence day
Christmas



120
110
100
90
80
70
60
50
Day of year effect
Valentine's day
April 1st
Memorial day
Leap day
9/11
Labor day
Halloween
Thanksgiving
Independence day
New year
Christmas
13th day of month
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

# A surprisingly tricky model

- Sum of declining exponentials: $y = a_1 e^{-b_1 x} + a_2 e^{-b_2 x}$
- Statistical version: $y_i = \left(a_1 e^{-b_1 x_i} + a_2 e^{-b_2 x_i}\right) \cdot \epsilon_i$

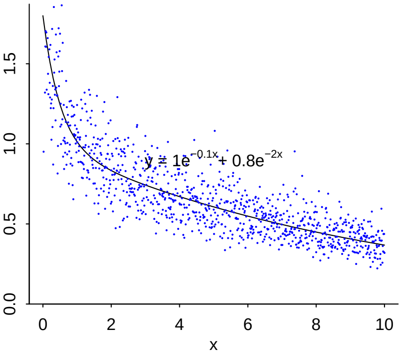


# Stan code

```
data {
  int N;
  vector[N] x;
  vector[N] y;}
parameters {
  vector[2] a;
  positive_ordered[2] b;
}
model {
  vector[N] ypred;
  ypred = a[1]*exp(-b[1]*x) + a[2]*exp(-b[2]*x);
  y ~ lognormal(log(ypred), sigma);
}
```

# Simulate fake data in R

```
a <- c(1, 0.8)
b <- c(0.1, 2)
sigma <- 0.2

x <- (1:1000)/100
N <- length(x)
ypred <- a[1]*exp(-b[1]*x) + a[2]*exp(-b[2]*x)
y <- ypred*exp(rnorm(N, 0, sigma))
```



# Fit the model in Stan

- Remember true values:

```
a <- c(1, 0.8)
b <- c(0.1, 2)
sigma <- .2
```

```
Inference for Stan model: exponentials.
4 chains, each with iter=1000; warmup=500; thin=1;
post-warmup draws per chain=500, total post-warmup draws=2000.

      mean se_mean   sd  25%  50%  75% n_eff Rhat
a[1]  1.00    0.00 0.03 0.99 1.00 1.02   494    1
a[2]  0.70    0.00 0.08 0.65 0.69 0.75   620    1
b[1]  0.10    0.00 0.00 0.10 0.10 0.10   532    1
b[2]  1.71    0.02 0.34 1.48 1.67 1.90   498    1
sigma 0.19    0.00 0.00 0.19 0.19 0.20   952    1
```

# Try again with new parameter values

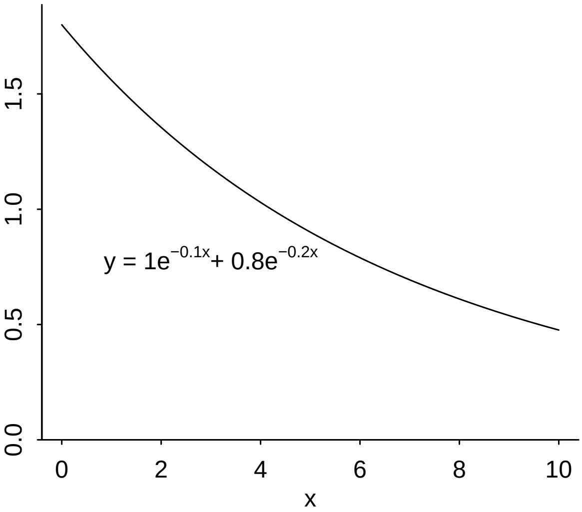
- Simulate new data using these new parameter values:

```
a <- c(1, 0.8)
b <- c(0.1, 0.2)
```

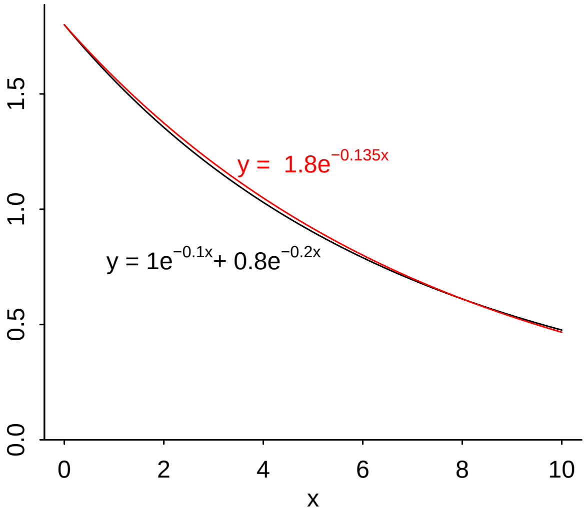
- Then fit the model:

|  | mean | se_mean | sd | 25% | 50% | 75% | n_eff | Rhat |
|---|---|---|---|---|---|---|---|---|
| a[1] | 1.33e+00 | 0.54 | 0.77 | 1.28 | 1.77e+00 | 1.79e+00 | 2 | 44.2 |
| a[2] | 2.46e+294 | Inf | Inf | 0.00 | 0.00e+00 | 1.77e+00 | 2000 | NaN |
| b[1] | 1.00e-01 | 0.04 | 0.06 | 0.10 | 1.30e-01 | 1.30e-01 | 2 | 33.6 |
| b[2] | 3.09e+305 | Inf | Inf | 0.50 | 1.15e+109 | 4.77e+212 | 2000 | NaN |
| sigma | 2.00e-01 | 0.00 | 0.00 | 0.19 | 2.00e-01 | 2.00e-01 | 65 | 1.0 |

# What went wrong?



# What went wrong?



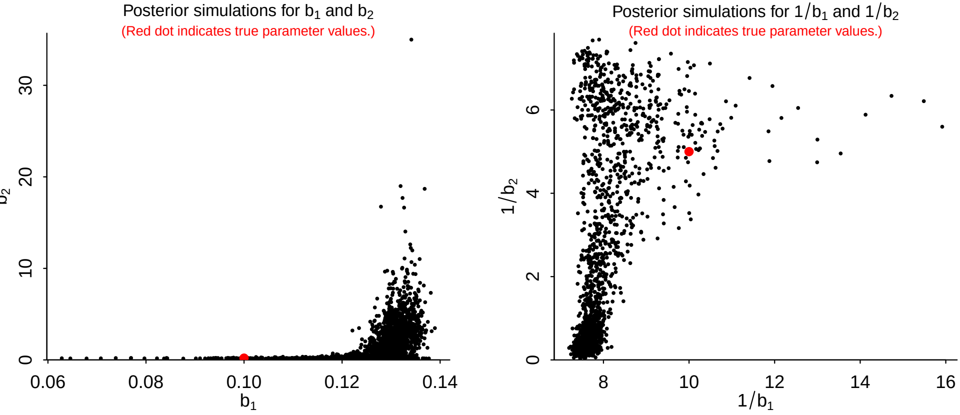
# Informative prior distribution

```
log_a ~ normal(0, 1);
log_b ~ normal(0, 1);
```

# Happy ending

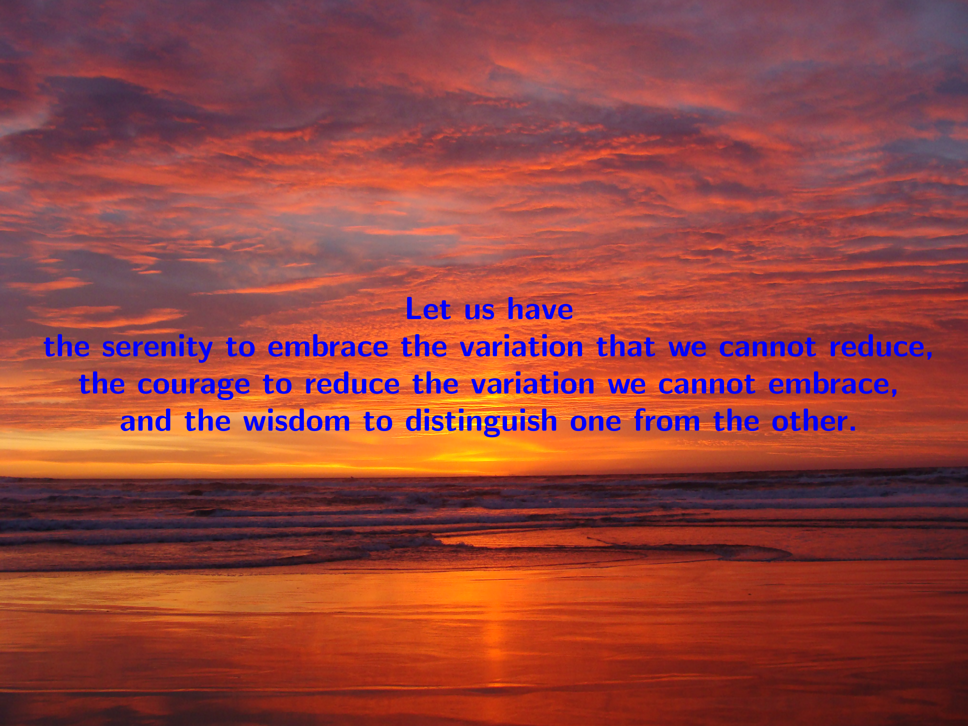
```
a <- c(1, 0.8)
b <- c(0.1, 0.2)
sigma <- 0.2

      mean se_mean   sd  25%  50%  75% n_eff Rhat
a[1]  1.56    0.09 0.32 1.52 1.72 1.75    13 1.25
a[2]  0.32    0.08 0.28 0.14 0.22 0.37    13 1.20
b[1]  0.13    0.00 0.01 0.12 0.13 0.13    22 1.14
b[2]  1.94    0.20 2.29 0.22 1.26 3.00   127 1.05
sigma 0.20    0.00 0.00 0.19 0.20 0.20   656 1.00
```

# Skewed posterior distribution



# Some ideas in Bayesian workflow

- Putting parameters on unit scale
- Weakly informative priors
- Predictive model checking
- Predictive model evaluation
- Predictive model averaging
- Fake-data checking
- The network of models



Let us have
the serenity to embrace the variation that we cannot reduce,
the courage to reduce the variation we cannot embrace,
and the wisdom to distinguish one from the other.