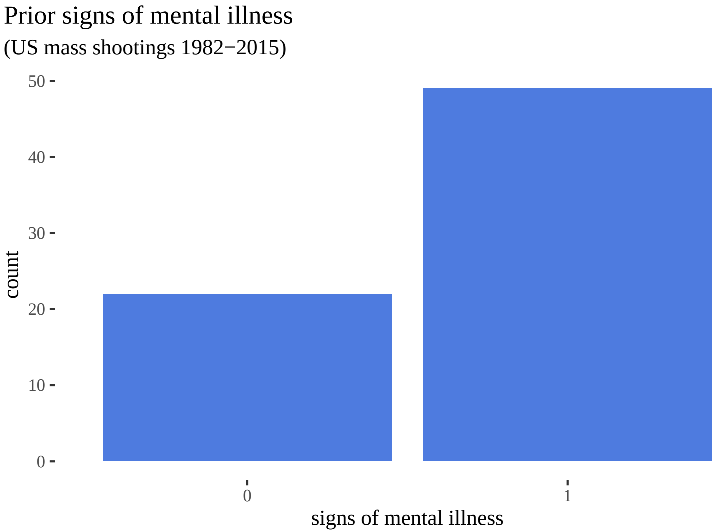
Sampling and Uncertainty

Nikodem Lewandowski

Intro

- We know how to build easy models:
  - use distributions to represent observed and unobserved variables
  - update priors with likelihood (taking into account all possible hypotheses)
  - obtain a posterior
- What we can do with a posterior?
- We can ask it questions by **sampling** (as our models are generative) and **evaluating samples**

Mass shootings



Proportion estimates and sampling

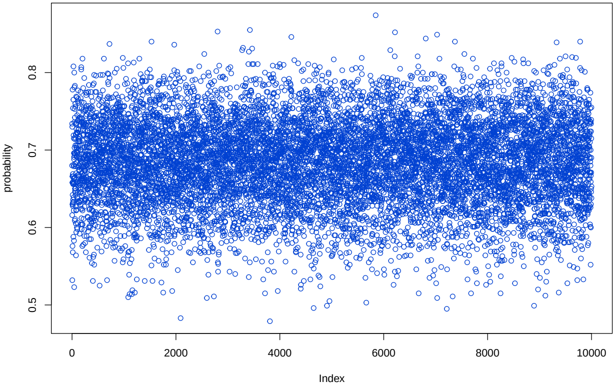
- grid approximated model of a parameter: being mentally ill

```
p_grid <- seq(0,1, length.out = 1001)
prior <- rep( 1, 1001)

likelihood <- dbinom( sum(sh$mental),   # 49 cases
                      size = nrow(sh),  # 71 total number
                      prob = p_grid)    # all possible hypotheses

posteriorUnnormalized <- prior * likelihood

posterior <- posteriorUnnormalized / sum(posteriorUnnormalized)
```

Proportion estimates and sampling

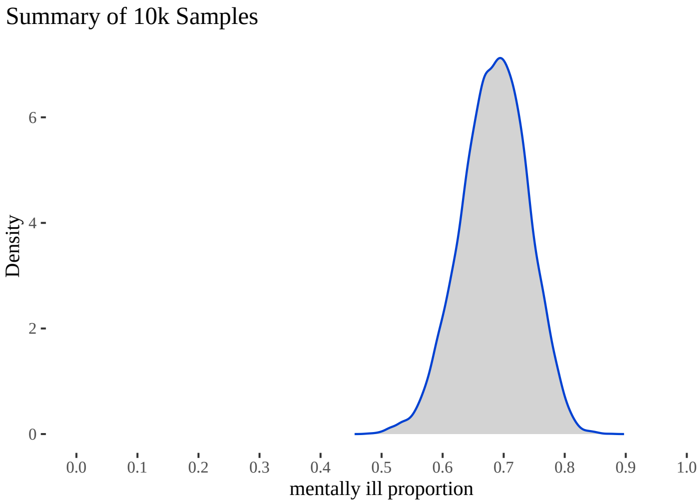
- Now we will make 10k samples from our posterior, so simulate 10k scenarios
  - randomly choose a value from the posterior distribution

```r
samples <- sample(p_grid, prob = posterior, size = 1e4,
                  replace = TRUE)
plot(samples, col= UGblue, ylab= 'probability')
```



Proportion estimates and sampling

```
dens(samples)
```



Evaluation

You evaluate sample with questions like:

- How much posterior probability lies below some parameter value?
- How much posterior probability lies between two parameter values?
- Which parameter value marks the lower 5% of the posterior probability?
- Which range of parameter values contains 90% of the posterior probability?
- Which parameter value has highest posterior probability?

Proportion estimates and sampling

```r
sum(posterior[p_grid > .6]) # probability that p is smaller than 0.6
```

```
[1] 0.9357886
```

```r
sum(samples > .6) / 1e4
```

```
[1] 0.9333
```

```r
sum(samples > .6 & samples < .7) / 1e4
```

```
[1] 0.5259
```

Proportion estimates and sampling

```
quantile(samples, c(.1,.9)) # 0.8 credible interval
```

```
  10%   90% 
0.614 0.754 
```

```
PI(samples, .8) # Percentile Interval
```

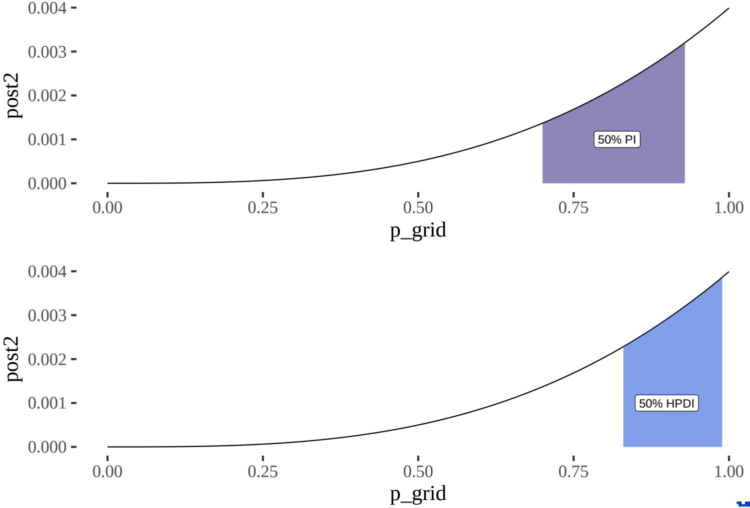
```
  10%   90% 
0.614 0.754 
```

```
HPDI(samples, .8) # Highest Posterior Density Interval
```

```
 |0.8  0.8| 
0.615 0.754 
```

University of Gdańsk

PI vs HPDI


0.004
0.003
0.002
0.001
0.000
post2
50% PI
0.00
0.25
0.50
0.75
1.00
p_grid
50% HPDI
p_grid

Now with model building

- Mass Shootings dataset again, this time weapons obtained legally variable

```
sh$WEAPONSOBTAINEDLEGALLY
```

```
 [1] "Yes" "Yes" "No"  ""    "Yes" "Yes" "Yes" "Yes" "Yes" "No"  "Yes" "Yes"
[13] "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "No"  "Yes" "Yes" "Yes"
[25] "Yes" "Yes" "No"  "No"  "Yes" "Yes" "No"  "Yes" "Yes" "Yes" "No"  "Yes"
[37] "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "No"  "No"  "No"  "Yes"
[49] "Yes" "No"  "Yes" "Yes" "Yes" ""    "Yes" "Yes" "No"  "Yes" "Yes" "Yes"
[61] "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "No"  "Yes"
```

$$\text{legal} \sim \text{Binomial}(N, \theta)$$
$$\theta \sim \text{Uniform}(0, 1)$$

Now with model building

```
legal <- sum(sh$WEAPONSOBTAINEDLEGALLY == "Yes")
illegal <- sum(sh$WEAPONSOBTAINEDLEGALLY == "No")
total <- legal + illegal

datweapons = list (legal = legal, illegal = illegal,
                   total = total)


weaponsModel <- ulam(
  alist(
    legal ~ dbinom( total , theta),
    theta ~ dunif(0,1)
  ) ,
  data= datweapons )
```

Model Summary and Sampling

```
precis(weaponsModel)
```

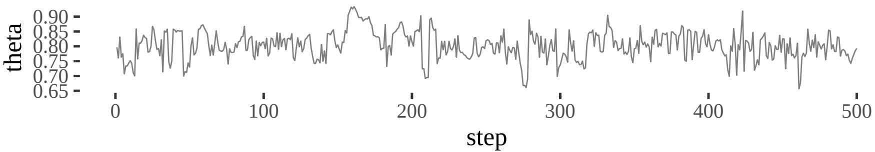
```
           mean          sd       5.5%      94.5%     rhat ess_bulk
theta 0.8028208 0.04678213 0.7264796 0.8698296 1.017784 75.64862
```

```
weaponsSamples <- as.data.frame(extract.samples(weaponsModel))

weaponsSamples$step <- 1:500

head(weaponsSamples)
```

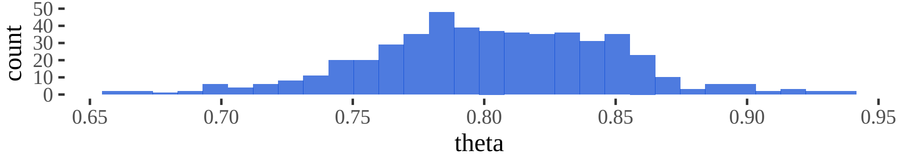
```
     theta step
1 0.796276    1
2 0.760616    2
3 0.831108    3
4 0.763048    4
5 0.775173    5
6 0.707148    6
```

Samples and Density

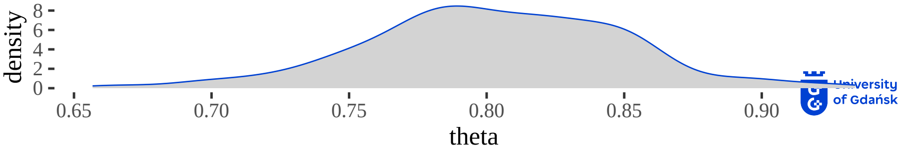
Parameter visits



Posterior counts



Posterior density



Beyond binomial: lots of small factors

```r
set.seed(212)
runif(1,-1,1)
```

```
[1] -0.1890287
```

```r
person1 <- runif(40,-1,1)
person1[1:15]
```

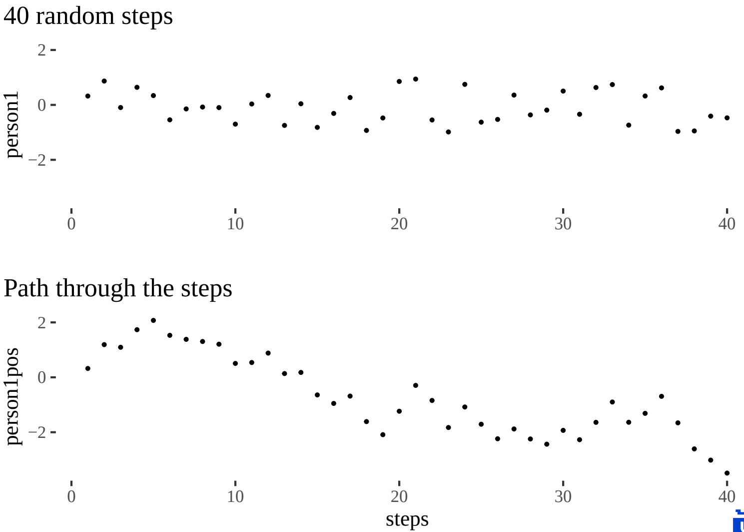
```
 [1]  0.68966791  0.50155319 -0.64601248 -0.98532639 -0.85444486  0.78535156
 [7]  0.84621936 -0.25596481 -0.15321414  0.12277733 -0.08131306  0.70020409
[13] -0.48404981 -0.56437817  0.56608362
```

```r
person1pos <- cumsum(person1)
person1pos
```

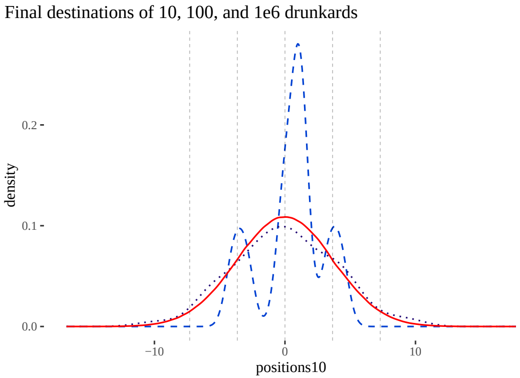
```
 [1]  0.68966791  1.19122110  0.54520862 -0.44011778 -1.29456263 -0.50921107
 [7]  0.33700829  0.08104348 -0.07217066  0.05060667 -0.03070639  0.66949770
[13]  0.18544790 -0.37893027  0.18715334  0.65272745 -0.06900282  0.16638617
[19]  0.99165463  1.11116518  1.26215532  1.65187720  0.65364415  0.09298024
[25]  0.80587173  1.74517056  2.55006080  3.11954291  2.30616824  1.32723905
[31]  1.94501019  1.11388091  0.48335103 -0.20944751 -0.26375439 -1.09592587
[37] -1.24518649 -1.27292195 -0.91852718 -0.50006697
```

University of Gdańsk

Beyond binomial: lots of small factors



Beyond binomial: lots of small factors




```r
sd(positions1e6)
```

```
[1] 3.651049
```

```r
# the proportion of values one sd from the mean
mean(abs(positions1e6) < abs(sd(positions1e6)))
```

```
[1] 0.681833
```

```r
mean(abs(positions1e6) < 2 * abs(sd(positions1e6)) ) # two sds from the mean
```

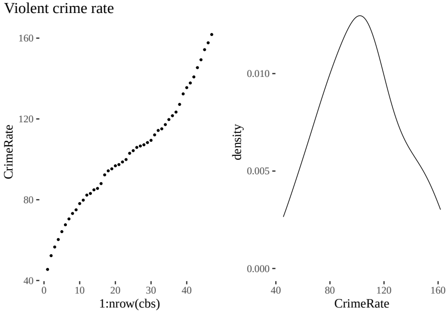
```
[1] 0.954926
```

Crime rates and normal distribution

```
cbs <-  read.csv(file = "../../datasets/CrimeByState.csv")
#these are registered violent incidents per 100k citizens
cbs$CrimeRate
```

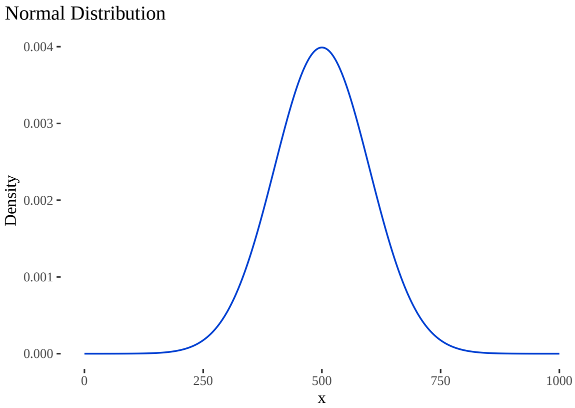
```
 [1]  45.5  52.3  56.6  60.3  64.2  67.6  70.5  73.2  75.0  78.1  79.8  82.3
[13]  83.1  84.9  85.6  88.0  92.3  94.3  95.3  96.8  97.4  98.7  99.9 103.0
[25] 104.3 105.9 106.6 107.2 108.3 109.4 112.1 114.3 115.1 117.2 119.7 121.6
[37] 123.4 127.2 132.4 135.5 137.8 140.8 145.4 149.3 154.3 157.7 161.8
```

```
cbsPlot <- grid.arrange(ggplot(cbs)+geom_point(aes(x=1:nrow(cbs),y = CrimeRate))+th+
            ggtitle("Violent crime rate"),
ggplot(cbs)+geom_density(aes(x=CrimeRate))+th, ncol=2)
```



Normal Distribution

```r
x <- seq(0, 1000, 1)
dnorm(x, mean = 500, sd = 100)
```



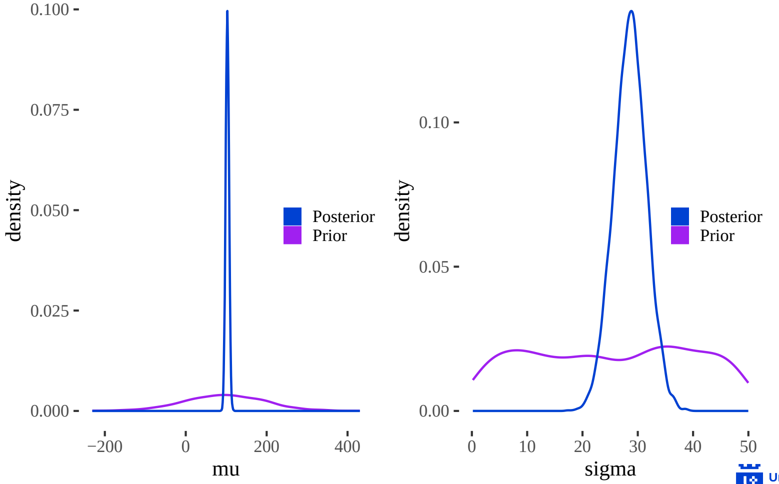
$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Crime rates and normal distribution

$$
\begin{aligned}
\text{rate} &\sim \text{Normal}(\mu, \sigma) \\
\mu &\sim \text{Normal}(100, 100) \\
\sigma &\sim \text{Uniform}(0, 50)
\end{aligned}
$$

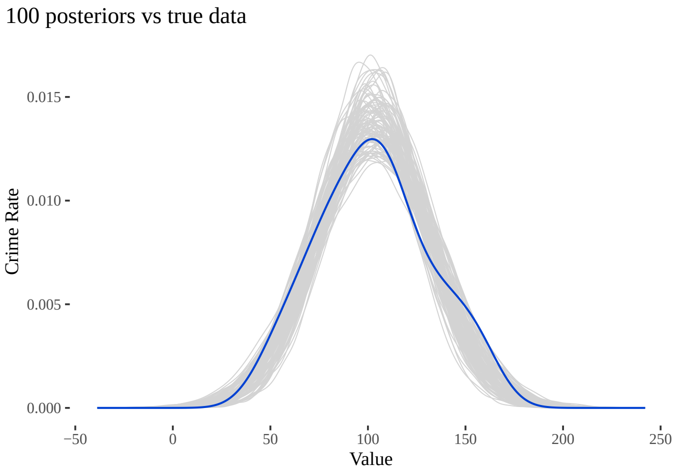
```r
dat <- list(rate = cbs$CrimeRate)

set.seed(123)
meanModel <- quap(
  alist(
  rate ~ dnorm( mu , sigma ) ,
  mu ~ dnorm( 100 , 100 ) ,
  sigma ~ dunif( 0 , 50 )
  ), data = dat
)
```

1k samples from the prior and the posterior



University of Gdańsk

Evaluating posteriors



- result of extracting 10k samples of mu and sigma, sub-setting them to 0.89 HPDI
- then sampling 100 values of mu and sigma from that subset
- creating their normal distributions (10k samples for each)
- plotted as gray lines contrasted with blue real distribution of crime rate

Simulating predictions

- The model is generative!

```
precis(meanModel)
```

```
          mean        sd     5.5%     94.5%
mu    102.79877 4.165765 96.14107 109.45646
sigma  28.58386 2.948141 23.87216  33.29556
```

```
pred <- sim(meanModel) # sampling 1k values for 47 states

str(pred)
```

```
 num [1:1000, 1:47] 122.7 121.3 122.4 112.1 89.2 ...
```

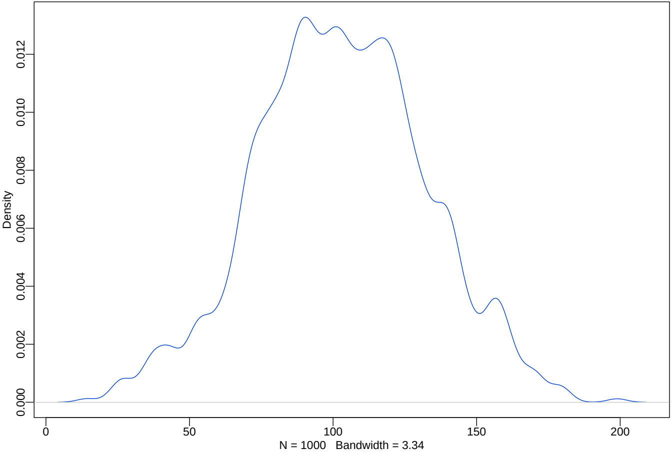
```
pred[1:5, 1:5] # 5 first predictions of 5 first states
```

```
          [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 122.67885  59.50806 126.56907 140.01684 114.28618
[2,] 121.34297  96.89885  96.61248  91.25465 104.74498
[3,] 122.36585  89.71487  83.79871  63.91233  77.50755
[4,] 112.13829 134.25018  70.44354 126.19677 115.70459
[5,]  89.24531  94.29293  59.70020  73.55355 155.21582
```

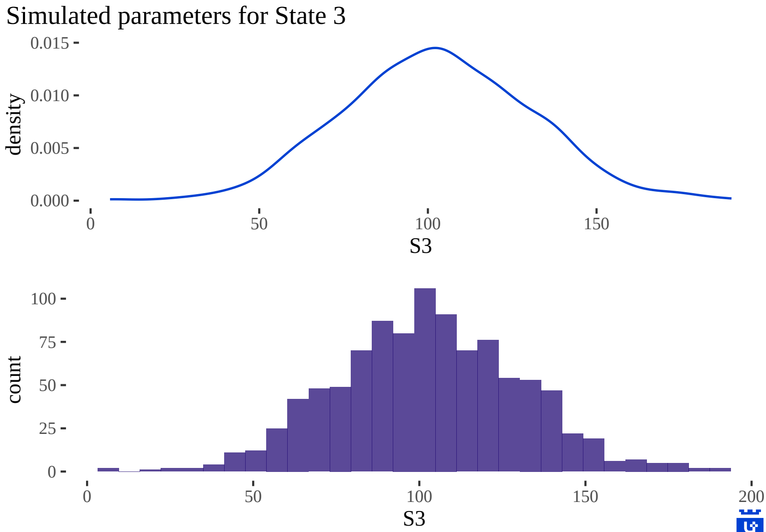
University of Gdańsk

Predictions for 1 chosen state

```
dens (pred[,1], col= UGblue)
```



Density and Counts



Summarizing our predictions

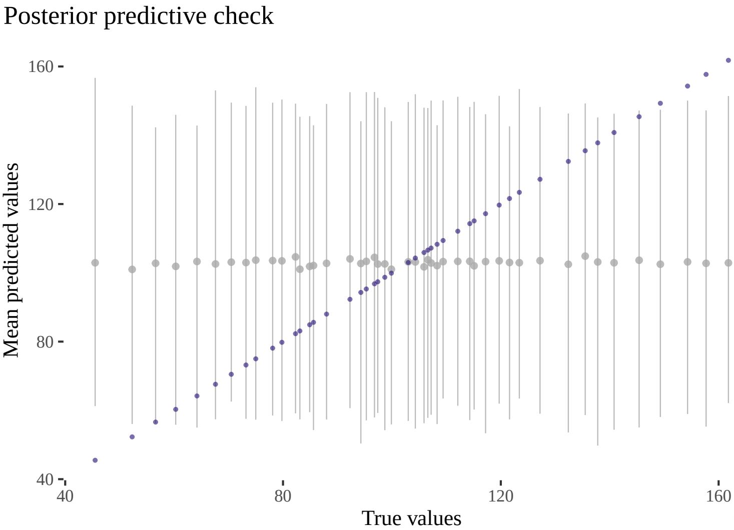
```r
# calculating the mean of predictions for all of the states
(meanpreds <- apply(pred, MARGIN = 2, FUN = mean))
```

```
 [1] 102.9239 100.9950 102.7737 101.8810 103.2923 102.5786 103.1034 102.9789
 [9] 103.6875 103.5586 103.4715 104.6334 101.0434 101.8701 102.1192 102.7659
[17] 104.0520 102.7163 103.3319 104.4945 102.5480 102.5901 101.0424 103.1863
[25] 103.1677 101.7346 103.7943 102.8545 102.0906 103.2580 103.3311 103.3371
[33] 102.0324 103.2610 103.4838 103.0043 102.9440 103.5319 102.4780 104.8545
[41] 103.1585 102.9238 103.6589 102.4998 103.1766 102.7622 102.8955
```

```r
# calculating 0.89 HPDI for all of the states predictions
hpdipreds <- as.data.frame(t(apply(pred, MARGIN = 2, FUN = HPDI)))
head(hpdipreds, n=10)
```

```
      |0.89    0.89|
1  61.23970 156.6802
2  56.04788 148.6105
3  56.25650 142.3157
4  55.83247 145.9335
5  54.99760 142.8291
6  57.41194 153.0368
7  62.57198 149.4996
8  57.55402 148.5420
9  57.31166 153.9692
10 58.52165 149.4642
```

Posterior evaluation with all the scenerios



Levels of uncertainty

```
rate ~ dnorm( mu , sigma ) ,
mu ~ dnorm( 100 , 100 ) ,
sigma ~ dunif( 0 , 50 )
```

```
           mean       sd     5.5%     94.5%
mu    102.79877 4.165765 96.14107 109.45646
sigma  28.58386 2.948141 23.87216  33.29556
```

Levels of uncertainty

```r
est <- extract.samples( meanModel )
pred <- sim( meanModel)

head(est)
```

```
         mu    sigma
1 107.63613 32.53144
2 108.84932 27.21955
3  99.94191 26.31424
4 100.35519 27.11859
5  99.98968 28.35499
6 101.65342 31.55342
```

```r
str(pred)
```

```
 num [1:1000, 1:47] 115 73.3 132.7 100.9 90.8 ...
```

Levels of uncertainty