HW1

Matt Kaye

9/12/2018

2H1

```r
prior <- c(.5, .5)
lik <- c(.1, .2)

posterior <- lik*prior / sum(lik*prior)

prob <- sum(posterior * lik)
```

The probability of the next baby birth being twins is 0.167.

2H2

```r
temp <- posterior[1]
```

The probability of the panda that we have being from species A is $p(A|twins)$ which is equal to the posterior probability that we calculated earlier: 0.333.

2H3

We use Bayes rule here to determine the posterior probability that the panda is species A. To do this, we recondition our original result from 2H1, using the posterior from that problem as our prior and making the likelihood the probabilities of not having twins.

```r
newLik <- c(.9, .8)
p <- posterior*newLik / sum(posterior*newLik)
```

This gives a result of 0.36 as the probability that this panda is species A.

2H4

```r
a <- .8
b <- .65

vetLik <- c(a, 1-b)
vetPrior <- c(.5, .5)

vetPost <- vetPrior * vetLik / sum(vetPrior * vetLik)
```

To find the posterior probability that the panda is species A, we use Bayes' rule. This gives a posterior probability of 0.696. This answer makes sense because it not only accounts for a false positive result for species A, but also a false negative for B.

```
p <- (p * vetLik) / sum(p * vetLik)
```

When we factor in our prior information, our confidence in the test diminishes further. In this case, the new posterior probability of the panda being of species A is 0.5625

5

a)

```
mu <- c(20, 30, 40, 50, 60, 70)
prior <- c(.1, .15, .25, .25, .15, .1)
snowfall <- c(38.6, 42.4, 57.5, 40.5, 51.7, 67.1, 33.4, 60.9, 64.1, 40.1, 40.7, 6.4)

snow <- data_frame(mu, prior)
snow
```

```
## # A tibble: 6 x 2
##      mu prior
##   <dbl> <dbl>
## 1    20  0.1
## 2    30  0.15
## 3    40  0.25
## 4    50  0.25
## 5    60  0.15
## 6    70  0.1
```

b)

```
yBar <- mean(snowfall)
yBar
```

```
## [1] 45.28333
```

c) HANDWRITTEN

d)

```
snow <- snow %>%
  mutate(likelihood = dnorm(mu, yBar, 10))

snow
```

```
## # A tibble: 6 x 3
##      mu prior likelihood
##   <dbl> <dbl>      <dbl>
## 1    20  0.1     0.00163
## 2    30  0.15    0.0124
## 3    40  0.25    0.0347
## 4    50  0.25    0.0357
## 5    60  0.15    0.0135
## 6    70  0.1     0.00188
```

e)

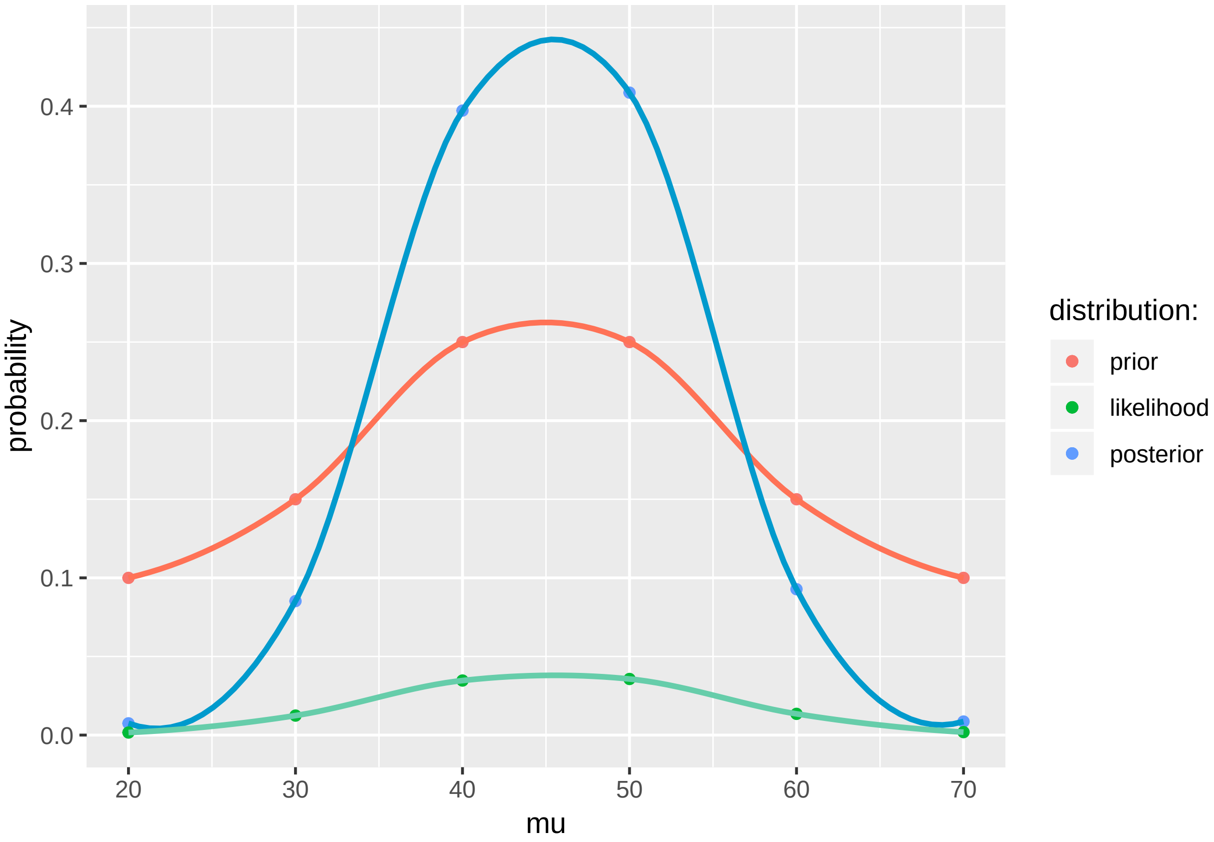
```
snow <- snow %>%
  mutate(posterior = prior * likelihood / sum(prior * likelihood))
```

```
snow
```

```
## # A tibble: 6 x 4
##      mu prior likelihood posterior
##   <dbl> <dbl>      <dbl>     <dbl>
## 1    20  0.1     0.00163   0.00748
## 2    30  0.15    0.0124    0.0852
## 3    40  0.25    0.0347    0.397
## 4    50  0.25    0.0357    0.409
## 5    60  0.15    0.0135    0.0928
## 6    70  0.1     0.00188   0.00861
```

f)

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 6

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 6

## Warning in sqrt(sum.squares/one.delta): NaNs produced

## Warning in stats::qt(level/2 + 0.5, pred$df): NaNs produced

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 6

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 6

## Warning in sqrt(sum.squares/one.delta): NaNs produced

## Warning in stats::qt(level/2 + 0.5, pred$df): NaNs produced

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 6

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 6

## Warning in sqrt(sum.squares/one.delta): NaNs produced

## Warning in stats::qt(level/2 + 0.5, pred$df): NaNs produced
```


0.4
0.3
0.2
0.1
0.0
probability
20
30
40
50
60
70
mu
distribution:
● prior
● likelihood
● posterior