HPI Bayesian Course

Contents

Set options **1**

Terms and abbreviations **1**

Week 1 **1**

Discrete random variables . . . 1
Discrete random variables: the binomial . . . 2
Continuous random variables (PDFs) . . . 4
Continous RVs: the normal distribution . . . 4
Maximum Likelihood estimates . . . 5
Bivariate and multivariate distributions (Discrete case) . . . 6

Set options

```
options(scipen = 999)
```

Terms and abbreviations

- k: number of successes
- n: number of trials
- θ: the probability of success

Week 1

Discrete random variables

- random variables are functions that map one set to the set of real numbers
  - associates to each outcome to a particular real number

There's a set of events that *can* happen, like tossing a coin. A random variable maps each of these possible events to a real number. In tossing a coin until you get a heads, a random variable would be the number of tosses until you get a 'success' (heads). In principle it could be an infinite set, bcause you could in theory toss the coin an infinite number of times without ever getting a heads (though this is extremely improbable).

Alternatively, the random variable can be a finite set if we are interested in looking whether *one* toss results in a head or a tails.

PMF: probability mass function, PDF: probability density function

- PMF = for discrete distributions
- PDF: for continuous
- CDF: cumulative distribution function; gives a mapping from a particular numerical value to a probability; means the probability of getting that number or something less than it.

Simulate tossing a coin 10 times with probability of heads = 0.5, with a Bernoulli random variable.

```
# simulate tossing a coin 10 times
extraDistr::rbern(n = 10, prob = .5)
```

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

What's the probablity of getting a tails or a heads? The d-family of functions:

```
extraDistr::dbern(0,prob=.5)
```

```
## [1] 0.5
```

```
extraDistr::dbern(1,prob=.5)
```

```
## [1] 0.5
```

Cumulative probability distribution function: the p-family of functions

```
# probability distribution function

## for whichever case we coded as 0
extraDistr::pbern(0,.5)
```

```
## [1] 0.5
```

```
## for whichever case we coded as 1
extraDistr::pbern(1,.5)
```

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

Discrete random variables: the binomial

When we toss a coin only once, this is a Bernoulli random variable. If we toss the coin more than once, this is a *binomial*. Both Bernoulli and Binomial have a PMF associated with them.

```
# generate random binomial data
rbinom(n = 10, size = 1, prob = .5)
```

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

```
# compute probabilites of a particular outcome
probs <- round(dbinom(0:10, size = 10, prob = .5),3); probs
```

```
##  [1] 0.001 0.010 0.044 0.117 0.205 0.246 0.205 0.117 0.044 0.010 0.001
```

```
x <- 0:10
rbind(x,probs)
```

```
##        [,1] [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10]  [,11]
## x     0.000 1.00 2.000 3.000 4.000 5.000 6.000 7.000 8.000  9.00 10.000
## probs 0.001 0.01 0.044 0.117 0.205 0.246 0.205 0.117 0.044  0.01  0.001
```

```
# compute cumulative probabilities of all possible outcomes
pbinom(0:10, size=10, prob = .5)
```

```
##  [1] 0.0009765625 0.0107421875 0.0546875000 0.1718750000 0.3769531250
##  [6] 0.6230468750 0.8281250000 0.9453125000 0.9892578125 0.9990234375
## [11] 1.0000000000
```

Compute quantiles using the inverse of the CDF: what is the quantile q such that the probability of X is greater than x?

```r
# generate distribution (0:10 outcomes or less, for 10 repetitions, with probability .5)
probs <- pbinom(0:10,size = 10, prob = .5)


# compute the inverse CDF; qbinom takes as its input a probability of an outcome and outputs the invers
qbinom(probs, size = 10, prob=.5)
```

```
##  [1]  0  1  2  3  4  5  6  7  8  9 10
```

Binomial distribution quiz

Q1: Consider participating in a lottery ten times. Each time the probability of winning a prize is 0.10. What is the probability of winning exactly 5 times?

```r
# use dbinom to compute PDF
round(
  dbinom(5, # produce probs for 5 successes
       size = 10, # out of 10 tries/trials
       prob = .1 # with a prob of success on each trial = .1
       ),
3)
```

```
## [1] 0.001
```

Q2: Consider lending 10 books from a library. The probability of getting a damaged book is 0.15. Compute the cumulative probability of having 2 or fewer damaged books rounded to three digits.

```r
# use pbinom to compute CDF
round(
  pbinom(2, # successes
    size = 10, # out of 10
    prob = .15 # with prob of .15

  ),
3)
```

```
## [1] 0.82
```

My summary notes

d-p-q-r family of functions

RV	Function(arguments)	Outcome
	`rbern(n,prob)`	generate random data
Bernoulli	`dbern(x,prob)`	PDF: probability of outcome x
	`pbern(q,prob)`	CDF: cumulative PDF of <=x
	`rbinom(n, size, prob)`	generate random data with n = of successes
	`dbinom(q,size,prob)`	PMF: probability of outcome x
Binomial	`pbinom(q,size,prob)`	CMF: cumulative PDF of <=x
	`qbinom(prob,size)`	inverse CMF: cumulative PMF of >=x
	`rnorm(n,mean,sd)`	generate random data
	`dnorm(x,mean,sd)`	PDF: probability of outcome x
normal	`pnorm(q,mean,sd)`	CDF: cumulative PDF of <=x
	`pnorm(q,mean,sd, lower.tail=F)`	inverse CDF: cumulative PDF of >=x

RV	Function(arguments)	Outcome
	`qnorm(prob,size)`	Compute quantiles corresponding to probabilities

Continuous random variables (PDFs)

- in discrete RV cases, we compute the probability of a *particular* outcome
- in continous RV cases, probability is defined by the ***area under the curve*** (AUC)
- we use the cumulative distribution function associated with a normal distribution to compute this AUC; i.e., the probability of observing a value between x1 and x2

Continous RVs: the normal distribution

- the standard normal distribution:
  - Normal(mean = 0, sd = 1)
  - Prob(-1 < X < 1) = 68%
  - Prob(-1.96 < X < 1.96) = 95%
  - Prob(-3 < X <3) = 99.7%
- for any other normal distribution (with varying mean and sd):
  - Prob(-1*sd < X < 1*sd) = 68%
  - Prob(-1.96*sd < X < 1.96*sd) = 95%
  - Prob(-3*sd < X <3*sd) = 99.7%
- the continuous vales on the x-axis (here, |/- 1,2,3) are the **quantiles**

`rnorm(n,mean,sd)`: Generate random data usin

```
rnorm(n=5,
      mean = 0,
      sd = 1)
```

```
## [1] -0.02858303  0.06615036  0.33049926  2.21893255  0.44400354
```

`pnorm(q)`: Compute probabilities using CDF

```
pnorm(q = 2,
      mean = 0,
      sd = 1)
```

```
## [1] 0.9772499
```

`pnorm(q,lower.tail=F)`: Compute inverse probabilities using CDF

```
pnorm(q = 2,
      lower.tail=F,
      mean = 0,
      sd = 1) # don't look the left (equal to or less than q), but the right (equal or greater than q)
```

```
## [1] 0.02275013
```

`qnorm(p,mean,sd))`: Compute quantiles corresponding to probabilities using the inverse of the CDF

```
qnorm(p = 0.9772499,
      mean = 0,
      sd = 1)
```

```
## [1] 2.000001
```

`dnorm()`: Compute the probability ***density*** for a quantile value - not the probability of a particular outcome - rather the **density** of that particular value, i.e., the y-axis value

```r
dnorm(x = 2,
      mean = 0,
      sd = 1)
```

```
## [1] 0.05399097
```

Normal distribution quiz

Q1: Given a standard normal distribution, what is the probability of getting a value lower than -3?

```r
pnorm(-3)
```

```
## [1] 0.001349898
```

Q2: Given a standard normal distribution, what is the probability of getting a value higher than -3?

```r
pnorm(-3, lower.tail=F)
```

```
## [1] 0.9986501
```

Maximum Likelihood estimates

Spoiler: it's pretty much the mean/mode/median of normally distributed data (where the three be the same); gives us the most likely value that the parameter θ has *given the data*

- the 'expectation' (discrete case)
- if you were to do an experment with a larger and larger sample size, you'd get closer to the expected value (e.g., value of .5 successes in repeated coin tosses)
- we can estimate θ (θ-hat), but we'll never know the true value of θ

From my book notes from SMLP 2022:

> In real exerimental situations we never know the true value of θ (probability of an outcome), but it can be derived from the data: θ *hat* $= k/n$, where k = number of observed successess, n = number of trials, and θ *hat* = observed proportion of successes. θ *hat* = ***maximum likelihood estimate*** of the true but unknown parameter θ. Basically, the **mean** of the binomial distribution. The **variance** can also be estimated by computing *(n(θ))(1 - θ)*. These estimates can be be used for statistical inference.

From my class notes from SMLP 2022:

> - common misunderstanding of the **maximum likelihood estimate (MLE)**: it doens't represent the true value of θ, because it's the MLE (best guess) for the *data you have*
> - but the MLE will be closer to the true value of θ assample size increases

- e.g., the PMF (binomial) contains three terms:
  - k: number of successes
  - n: total number of trials
  - θ: probability of success (can have values between 0 and 1)
- the PMF is a function of these parameters, based on the data (given our data we know the values of k and n so they are fixed quantities and no longer random), so θ can be treated as a variable
- the **likelihood function** is the PMF (or PDF) as a function of the parameters, rather than a function of the data
- the MLE from a particular *sample* of data doesn't necessarily give us an accurate estimate of the true value of θ (because it's just a sample, of course)

- larger sample sizes get MLEs that more accurately represent the true value of θ (again, duh because of the point made above regarding the expectation)

Bivariate and multivariate distributions (Discrete case)