Hw4 - Beimnet Taye

2023-02-20

Worked with Joan Shim and Lucas Yoshida

P1

1

- They are identical, identically distributed, not independent.

2

- B: They are identically distributed and independent. C: They are identically distributed and independent.

3

- A and B

4

- all three have the same marginal distribution for X

P2

1

- Random Variable: Let C be a random variable that takes an individual and assess whether they are infected with covid or not.
- Domain: $f_C^{-1} = \{individual\}$
- Range: $F_C(individual) = \{I = infected, H = Healthy\}$
- Random Variable: Let t be a random variable that takes an individual and assess whether they have tested positive or negative.
- Domain: $f_t^{-1} = \{individual\}$
- Range: $F_t(individual) = \{+ = positive, - = negative\}$
- $P(t = +) = 0.02$, $P(C = I \cap t = +) = 0.01$
- $P(t = +)$ is a marginal probability and $P(C = I \cap t = +) = 0.01$ is a joint probability

```r
pp <- 0.02
ppi <- 0.01
```

2

- $P(C = I|t = +) = \frac{P(C=I,t=+)}{P(t=+)}$ (Ven diagram in Figures section at end of pdf)

```r
pigp <- ppi/pp
pigp
```

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

3

- $P(t = +|C = I) = 0.9$
- $P(C = I) = 0.009$

```r
ppgi <- 0.9
pi <- 0.009
```

4

$P(I|+) = \frac{P(I)*P(+|I)}{P(+)}$

```r
pigp <- (pi*ppgi)/pp
pigp
```

```
## [1] 0.405
```

P3

1

- They are identically distributed. Not independent.

2

- They are identical, identically distributed, and not independent.

P4

1

```r
X = function() {
  c(X = rnorm(1))
}
Y_given_X = function(x) {
  c(Y = rexp(1, rate=abs(x)))
}
Y_and_X = function() {
  x = X()
  y = Y_given_X(x)
  return(list(X=x, Y=y))
}
sample = function(dist, n, ...) {
  map_df(1:n, function(i) dist(...))
}

prob_x <- X %>%
  sample(10000) %>%
  count(X<0) %>%
  mutate(p = n/sum(n))

pull(prob_x[2,3])
```

```
## [1] 0.5019
```

```r
prob_ygx <- Y_given_X %>%
  sample(10000, x= 1) %>%
  count(Y>1) %>%
  mutate(p = n/sum(n))

pull(prob_ygx[2,3])
```

```
## [1] 0.3635
```

```r
prob_yandx <- Y_and_X %>%
  sample(10000) %>%
  count(X<0 & Y > 1) %>%
  mutate(p = n/sum(n))

pull(prob_yandx[2,3])
```

```
## [1] 0.2631
```

2

- It's not possible to compute with the given functions alone without manipulation since Y is continuous and we can't get it to exactly equal 1 when we sample from the joint and thus the probability using this sampling code is zero.

3

- Sample from the joint function and filter for when $Y > 1$ to get the probability when $X < 0$.

```
prob_xgy1 <- Y_and_X %>%
  sample(10000) %>%
  filter(Y > 1) %>%
  count(X<0) %>%
 mutate(p = n/sum(n))

pull(prob_xgy1[2,3])
```

```
## [1] 0.4840618
```

P5

1

```
f_X = function(x) {dnorm(x)
}

f_Y_given_X = function(y,x) {
  dexp(y, rate=abs(x))
}

f_Y_and_X <- function(y,x){
  X <-  f_X(x)
  Y <- f_Y_given_X(y,x)
  return(c(X,Y))

}
```

2

```
f_Y_given_X_is_0 <- function(y){
  dexp(y, rate=abs(0))
}
```

3

```
p1 <- integrate(f_X,-1,0)

p2 <- integrate(f_Y_given_X_is_0,-1,0)

p3 <- int2(f_Y_and_X, a = c(0,0), b = c(1,1))

p1[["value"]]
```

```
## [1] 0.3413447
```

```r
p2[["value"]]
```

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

```r
p3
```

```
## [1] 0.3413447 0.3413447
```

FIGURES

P2.2