

# Subject CS2

## Corrections to 2021 study material

### 0 Introduction

This document contains details of any errors and ambiguities that have been brought to our attention in the Subject CS2 study materials for the 2021 exams. We will incorporate these changes into the study material each year. We are always happy to receive feedback from students, particularly details concerning any errors, contradictions or unclear statements in the courses. If you have any such comments on this course please email them to [CS2@bpp.com](mailto:CS2@bpp.com).

You may also find it useful to refer to the Subject CS2 threads on the ActEd Discussion Forum. (You can reach the Forums by clicking on the 'Discussion Forums' button at the top of the ActEd homepage, or by going to [www.acted.co.uk/forums/](http://www.acted.co.uk/forums/).)

This document was last updated on **14 April 2021**.

# 1 Paper A Course Notes

## Chapter 15

*Page 14*

*(added on 1 March 2021)*

The R code to create the `qpareto()` function uses the incorrect character for a minus sign in two places (meaning it won't run when copied into R). It should be:

```
qpareto <- function(p,a,lambda){
  lambda * ((1-p)^(-1/a)-1)
}
```

## Chapter 19

*Page 15*

*(added on 1 April 2021)*

The equation near the bottom of the page showing that  $\left. \frac{d^3}{dt^3} \log M_S(t) \right|_{t=0}$  is equal to  $\lambda m_3$  is

incorrect. It should be:

$$\left. \frac{d^3}{dt^3} \log M_S(t) \right|_{t=0} = \lambda \left[ \left. \frac{d^3}{dt^3} (M_X(t) - 1) \right]_{t=0} = \lambda m_3$$

## 2 Assignments

### Assignment Y1 Solutions

#### Question 1

*(added on 15 December 2020)*

The solutions to part (iii) are incorrect.

#### Part (iii)(a)

For part (iii)(a), the solutions incorrectly give the three-step transition probability from level 3+ to level 1 instead of the three-step transition probability from level 4+ to level 1.

After calculating the three-step transition probability matrix, the solution should read:

So the probability is 0.0478. [1]

*We can also extract the probability using indexing:*

```
P3[5,1]
```

```
[1] 0.047785
```

```
P3["level 4+", "level 1"]
```

```
[1] 0.047785
```

At the end of the solution, the alternative using the `markovchain` object should read:

```
(mc^3)@transitionMatrix[5,1]
```

```
[1] 0.047785
```

#### Part (iii)(b)

For part (iii)(b), the solutions sum probabilities from the wrong row of the five-step transition probability matrix. After calculating the five-step transition probability matrix, the solution should read:

Having a discount of 15% or less means being on level 1, 2, 3+ or 3-. For a driver currently on level 5, the probability of this being the case in 5 years' time is the sum of the first four numbers on the last row of  $P_5$ .

Calculating this in R:

```
sum(P5[7, 1:4]) # [1]
```

```
[1] 0.1489257
```

So the probability is 14.89%. [1]

[Total 5]

*Alternatively, using the matrix power function:*

```
sum(Pn(P, 5)[7, 1:4])
```

```
[1] 0.1489257
```

As another alternative, using the markovchain object:

```
sum((mc^5)@transitionMatrix[7, 1:4])
[1] 0.1489257
```

### Question 3

(added on 14 April 2021)

#### Part (iv)

The cox model fitted in part (iv) does not use Breslow's approximation for ties. The solution should read:

```
cox.fit = coxph(surv.obj ~ surv.data$status, ties = "breslow")
summary(cox.fit)
```

Call:

```
coxph(formula = surv.obj ~ surv.data$status, ties = "breslow")
```

```
n= 200, number of events= 150

              coef exp(coef) se(coef)      z Pr(>|z|)
surv.data$statusT -1.1028    0.3319   0.1874 -5.886 3.96e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

              exp(coef) exp(-coef) lower .95 upper .95
surv.data$statusT    0.3319      3.013   0.2299   0.4792

Concordance= 0.633 (se = 0.02 )
Likelihood ratio test= 35.34 on 1 df,  p=3e-09
Wald test               = 34.64 on 1 df,  p=4e-09
Score (logrank) test = 37.69 on 1 df,  p=8e-10
```

So, we estimate  $\beta$  to be **-1.1028**.

#### Part (v)

The stated value of  $e^{\hat{\beta}}$  should instead be 0.3319. The stated  $p$ -value should instead be  $3.96 \times 10^{-9}$ . These are in line with the corrected output shown for part (iv).

#### Part (vii)

Using the updated value of  $\hat{\beta}$ , the probability for part (a) should be:

```
exp(-integrate(h1, 0, 40)$value)
[1] 0.950294
```

The probability for part (b) should be:

```
exp(-integrate(h0, 0, 40)$value)
[1] 0.857615
```

**Parts (vi) and (vii)**

These parts are not significantly affected by the updated value of  $\hat{\beta}$ . The graphical output is very similar and the same commentary still applies.

**Assignment Y2 Solutions****Question 1****(added on 1 April 2021)**

There is a mistake in the alternative R code provided at the end of the paragraph under the definition of the `fMLE` function. The minus sign should go outside the `log` function. It should read:

This function uses the `dweibull` command, which gives the probability density function of the Weibull distribution. This is easier than writing the log-likelihood function ourselves. We have taken logs by setting `log=TRUE`. (If we had used the default `log=FALSE`, we would have had to use `sum(-log(dweibull...))`.)

### 3 PBOR

## Chapter 7 Solutions

Pages 8 and 9

(added on 14 April 2021)

In Q7.2(iv)(c), the plot of the integrated hazard should start at 0 and go up to time 25, the time when the last life was censored. The final code after incorporating the adjusting the  $y$ -axis limits should be:

```
end.time = surv.data$Time[nrow(surv.data)]
end.i.haz = deaths$Lj[length(deaths$Lj)]

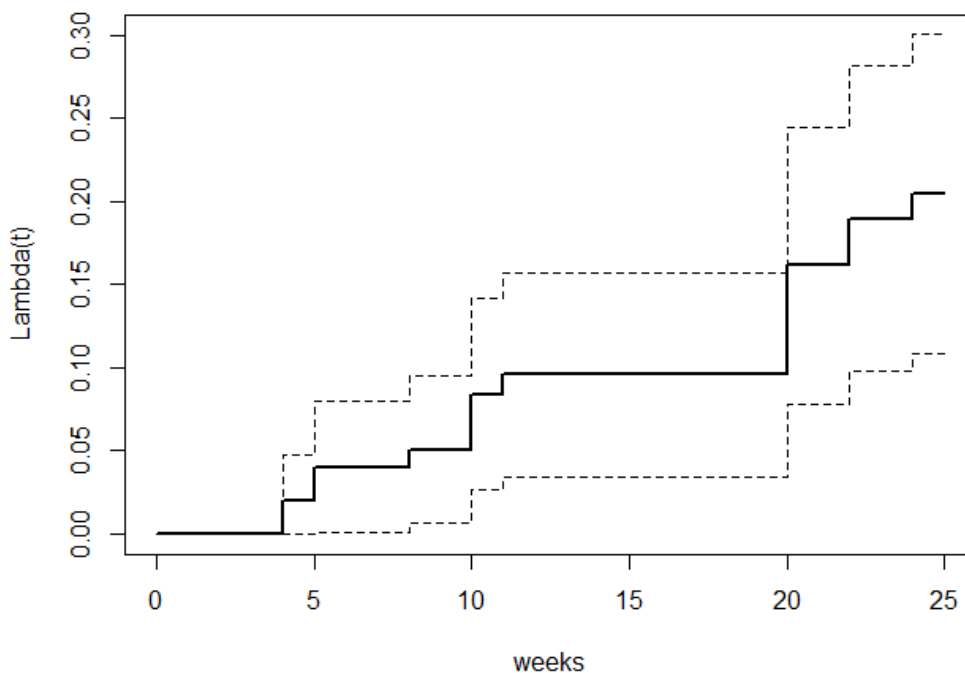
plot(c(0, deaths$tj, end.time),
     c(0, deaths$Lj, end.i.haz),
     type = "s", lwd = 2,
     main = "Nelson-Aalen estimate of the integrated hazard
function",
     xlab = "weeks",
     ylab = "Lambda(t)",
     ylim = c(0, 0.3))

end.lower = Lj.CI$lower[length(Lj.CI$lower)]
end.upper = Lj.CI$upper[length(Lj.CI$upper)]

lines(c(0, deaths$tj, end.time),
      c(0, Lj.CI$lower, end.lower), type = "s", lty = 2)

lines(c(0, deaths$tj, end.time),
      c(0, Lj.CI$upper, end.upper), type = "s", lty = 2)
```

**Nelson-Aalen estimate of the integrated hazard function**



**Page 11****(added on 14 April 2021)**

In Q7.2(v)(c) the  $y$ -axis label should be  $SNA(t)$  instead of  $SKM(t)$  as it is a plot of the Nelson-Aalen estimate. The code should be:

```
plot(fitna,
     main = "Nelson-Aalen estimate of survival function",
     xlab = "weeks",
     ylab = "SNA(t)", ylim = c(0.6, 1))
```

**Pages 12 and 13****(added on 14 April 2021)**

In Q7.2(vi)(b) the  $y$ -axis label should be  $S(t)$  instead of  $SKM(t)$  as it is a plot of both estimates (initially the Nelson-Aalen estimate is plotted and then the Kaplan-Meier estimate is subsequently added to the plot). The `plot()` function code should be:

```
plot(fitna,
     main = "Nelson-Aalen estimate of survival function",
     xlab = "weeks",
     ylab = "S(t)", ylim = c(0.7, 1))
```

**Chapter 11****Page 3****(added on 9 February 2021)**

The `for` loop in the code on page 3 has an error in the calculation of  $\Phi_{ixj}$ . The final term should be added instead of subtracted. In this particular case, this makes no difference to the final output.

The code should be:

```
for(x in xvalues){
  term = a0 + a1*x;
  for(j in 1:(n-2)){
    c1 = (xx[n] - xx[j])/(xx[n] - xx[n-1])
    c2 = (xx[n-1] - xx[j])/(xx[n] - xx[n-1])
    Phixj = phi(x,j) - c1*phi(x,n-1) + c2*phi(x,n)
    term=term+b[j]*Phixj}
  yvalues = c(yvalues,term)
}
```

The formula for  $\Phi_j(x)$  given near the bottom of page 3 also contains the same error. The formula should be:

$$\Phi_j(x) = \phi_j(x) - \left( \frac{x_n - x_j}{x_n - x_{n-1}} \right) \phi_{n-1}(x) + \left( \frac{x_{n-1} - x_j}{x_n - x_{n-1}} \right) \phi_n(x)$$

**Page 7****(added on 9 February 2021)**

The same error occurs when this code is repeated at the bottom of page 7. The preceding text and code should read:

The variable `Phixj` corresponds to  $\Phi_j(x)$ , which is calculated by subtracting the adjustment term

$c_1 \phi_{n-1}(x)$  and adding the adjustment term  $c_2 \phi_n(x)$ , where  $c_1 = \frac{x_n - x_j}{x_n - x_{n-1}}$  and  $c_2 = \frac{x_{n-1} - x_j}{x_n - x_{n-1}}$ ,

from the value of  $\phi_j(x)$ .

```
for(j in 1:(n-2)){
  c1 = (xx[n] - xx[j])/(xx[n] - xx[n-1])
  c2 = (xx[n-1] - xx[j])/(xx[n] - xx[n-1])
  Phixj = phi(x, j) - c1*phi(x, n-1) + c2*phi(x, n)
  term = term + b[j]*Phixj}
```