Subject CS2

Corrections to 2022 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 2022 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.

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/**.)

This document was last updated on **1 September 2022**.

1 CMP Upgrade

Page 9 ***(added on 9 November 2021)***

A new section has been added at the bottom of this page for Section 3 of Chapter 15 and additional replacement pages have been included. Please see the latest CMP Upgrade for more details.

Page 15 ***(added on 9 November 2021)***

A new section has been added for Chapter 12. Please see the latest CMP Upgrade for more details.

2 Paper A Course Notes

Chapter 7

Page 5 ***(added on 9 November 2021)***

There is a mistake in the last line of the paragraph on this page. The paragraph should read:

Observing lives between (say) integer ages x and $x+1$, and limiting the period of investigation, are also forms of censoring. Censoring might still occur at unpredictable times – by lapsing a life policy, for example – but survivors will certainly be lost to observation at a known time, either on attaining age $x+1$ or when the investigation ends.

Chapter 12

Page 42 ***(added on 9 February 2022)***

The equation for the age-period-cohort version of the Lee-Carter model should be:

$$\ln m_{x,t} = a_x + b_x^1 k_t + b_x^2 h_{t-x} + \varepsilon_{x,t}$$

Page 54 ***(added on 1 September 2022)***

The given set of $B_j(x)$ functions do not form a basis for cubic splines. The solution should instead read:

The mortality projection model would now be:

$$\ln\left[E\left(D_{x,t}\right)\right] = \ln E_{x,t}^c + \sum_{j=1}^{J} \theta_j B_j(t)$$

where the $B_j(t)$ are a set of basis splines for cubic splines.

Page 54 ***(added on 9 November 2021)***

The penalised log-likelihood in the fourth bullet point should not include the factor of $\frac{1}{2}$ in the penalty term. It should be:

$$l_p(\theta) = l(\theta) - \lambda P(\theta)$$

Chapter 13

Page 20 ***(added on 9 November 2021)***

There is a mistake in the first paragraph of the moving average definition. The paragraph should read:

A *moving average process* of order q, denoted $MA(q)$, is a sequence $\{X_t\}$ defined by the rule:

Page 64 ***(added on 1 September 2022)***

The values given for the autocovariances in the solution to 13.5 should all be multiplied by σ^2. It should read:

The autocovariance function is:

$$\text{cov}(X_n, X_{n+k}) = \text{cov}(e_n - 5e_{n-1} + 6e_{n-2}, e_{n+k} - 5e_{n+k-1} + 6e_{n+k-2})$$

$$= \begin{cases} 62\sigma^2 & k = 0 \\ -35\sigma^2 & |k| = 1 \\ 6\sigma^2 & |k| = 2 \\ 0\sigma^2 & |k| > 2 \end{cases}$$

Page 65 ***(added on 1 September 2022)***

The example calculation at the top of the page should read:

For example:

$$\begin{aligned} \gamma_0 &= \text{cov}(X_n, X_n) \\ &= \text{cov}(e_n - 5e_{n-1} + 6e_{n-2}, e_n - 5e_{n-1} + 6e_{n-2}) \\ &= \text{cov}(e_n, e_n) + \text{cov}(-5e_{n-1}, -5e_{n-1}) + \text{cov}(6e_{n-2}, 6e_{n-2}) \\ &= \sigma^2 + (-5)^2\sigma^2 + 6^2\sigma^2 = 62\sigma^2 \end{aligned}$$

and similarly for the other values of k.

Chapter 14

Page 4 ***(added on 9 November 2021)***

The penultimate paragraph in the R box has a mistake in the name of the `ts.plot()` function. It should read:

As the `ts.plot()` function plots a line graph by default, the points can be added with the `points()` function:

Page 25 ***(added on 9 November 2021)***

There is a mistake in the last line of the second paragraph on this page. The paragraph should read:

The asymptotic variance of $\tilde{\phi}_k$ is $1/n$ for each $k > p$. Again a normal approximation can be used, so that values of the SPACF outside the range $\pm 2/\sqrt{n}$ may suggest that the $AR(p)$ model is inappropriate.

Chapter 15

Page 25 ***(added on 9 November 2021)***

The final equation on this page should be:

$$\left.\frac{d}{d\theta} l(\theta)\right|_{\theta=\hat{\theta}} = 0$$

Page 36 ***(added on 9 November 2021)***

The first paragraph on this page is Core Reading and should be bold. It should be:

The `fitdistr()` function uses a numerical algorithm for the Weibull distribution, which requires starting values. If no values are provided, then the function automatically calculates a starting point.

Chapter 17

Page 55 ***(added on 21 January 2022)***

The final expression in the solution to Question 17.5 has a mistake in the power. The power should be $2^{1/\alpha}-1$ instead of $2^{1/\alpha-1}$. The final limit should be:

$$\lim_{u\to 0^+} u^{2^{1/\alpha}-1}$$

Page 56 ***(added on 12 April 2022)***

The last paragraph of part (ii) of Solution 17.6 suggests using the co-monotonic copula to capture positive interdependence. The co-monotonic copula has perfect positive interdependence and may not be appropriate unless this appeared to be exhibited by the data. A copula such as the gaussian or Student's t with appropriate parameters could be used to capture a degree of positive interdependence throughout a joint distribution.

Chapter 19

Page 27 ***(added on 14 December 2021)***

There are some errors in the equations embedded in the text of the example in section 3.7. This example should read:

This model can be expanded to deal with expenses as the following example demonstrates.

Each year an insurance company issues a number of household contents insurance policies, for each of which the annual premium is £80. The aggregate annual claims from a single policy have a compound Poisson distribution; the Poisson parameter is 0.4 and individual claim amounts have a gamma distribution with parameters α and λ. The expense involved in settling a claim is a random variable uniformly distributed between £50 and £ b (> £50). The amount of the expense is independent of the amount of the associated claim. The random variable S represents the total aggregate claims and expenses in one year from this portfolio. It may be assumed that S has approximately a normal distribution.

(i) **Suppose that:**

$$\alpha = 1; \ \lambda = 0.01; \ b = 100$$

Show that the company must sell at least 884 policies in a year to be at least 99% sure that the premium income will exceed the claims and expenses outgo.

(ii) **Now suppose that the values of α, λ and b are not known with certainty but could be anywhere in the following ranges:**

$$0.95 \le \alpha \le 1.05; \ 0.009 \le \lambda \le 0.011; \ 90 \le b \le 110$$

By considering what, for the insurance company, would be the worst possible combination of values for α, λ and b, calculate the number of policies the company must sell to be at least 99% sure that the premium income will exceed the claims and expenses outgo.

Page 11 ***(added on 21 January 2022)***

The labelling of equations from Equation 19.4 onwards is incorrect. Equation 19.4 should be labelled 19.2 and so on.

Chapter 21

Page 32 ***(added on 28 February 2022)***

There is a typo in the solution to part (ii)(b). It should read:

Similarly, the estimated probability that a claim from Region 1 for a Large amount is fraudulent is $\frac{3}{3+176} = 0.0168$, *ie* 1.68%.

Page 35 ***(added on 9 November 2021)***

There is a mistake in the penultimate paragraph of this page. This paragraph is discussing the values of the quantity $\sum_{k=1}^{K} p_{jk}(1-p_{jk})$ and not the Gini index. The paragraph should read:

For a classification problem where the data points are divided into m distinct categories, this quantity must take a value between 0 and $1-\frac{1}{m}$. As $m \to \infty$, the upper limit of this quantity tends to 1.

Page 41 ***(added on 21 December 2021)***

Around halfway down the page, part (i)(a) should be part (i)(b).

Pages 45, 46, 47 ***(added on 21 December 2021)***

There is a typo in the titles of the graphs on these pages. The first line of the title should read:

Predicted vs. observed median house

Page 46 ***(added on 21 December 2021)***

There is an error in the section reference around halfway down the page. It should read:

When we introduced random forests in Section 3.3, we discussed considering subsets of the input variables at each split point.

Page 60 ***(added on 9 November 2021)***

The expression for the penalised log-likelihood in the Penalised generalised linear models section is incorrect. It should be:

Penalised generalised linear models

Penalised regression is an adaptation of the method of maximum likelihood where a penalty is applied to constrain the estimated values of the parameters to improve their reliability for making predictions. The method involves maximising the penalised likelihood:

$$l(\beta_0, \beta_1, \ldots, \beta_d \mid \mathbf{x}, \mathbf{y}) - \lambda g(\beta_0, \beta_1, \ldots, \beta_d)$$

3 Assignments

Assignment X1 Solutions

Question 3 ***(added on 9 November 2021)***

The solution for Chain 1 does not reflect the latest Core Reading on periodicity. The solution should read:

Chain 1 is not periodic or aperiodic. It is not possible to return to State 1 at all and State 2 is aperiodic.

Assignment X2 Questions

Question 5 ***(added on 9 November 2021)***

The part reference in part (ii) is incorrect. It should read:

Write down an integral expression for $p_{12}(x,x+t)$ in terms of transition rates and the probabilities in part (i).

Assignment X2 Solutions

Question 6 ***(added on 29 June 2022)***

There is a typo in the expression for the sum of a geometric series at the top of page 8. It should read:

$$a+ar+ar^2+\ldots+ar^{n-1}=\frac{a(1-r^n)}{1-r}$$

Assignment X4 Questions

Question 9, part (iv)(a) ***(added on 1 September 2022)***

There is a typo in the wording of the question. It should refer to the autocovariance function of X_t, not the autocorrelation function:

Show that the autocovariance function of Y_t, γ_k^Y, can be expressed in terms of the autocovariance function of X_t, γ_k, as follows:

Assignment X4 Solutions

Question 10, part (iii)(a) ***(added on 29 June 2022)***

There is a typo in the first line of the calculation at the top of page 19. It should be:

$$P(X_M \leq 495) = P\left(\frac{X_M - 500}{\frac{1}{50}500} \leq \frac{495-500}{\frac{1}{50}500}\right)$$

$$\approx \exp\left(\frac{495-500}{\frac{1}{50}500}\right)$$

$$= \exp(-0.5) = 0.60653$$

Question 10, part (iii)(c) ***(added on 29 June 2022)***

The comment on page 19 incorrectly refers to the distribution of the standardised sample mean instead of the standardised sample maximum. It should read:

The probabilities are similar, suggesting that the GEV distribution provides a reasonable approximation to the standardised sample maximum distribution for $n = 50$.

Question 10, part (iv)(c) ***(added on 29 June 2022)***

The first line of the solution on page 20 references the wrong question part. It should read:

Here we have that $W = X - 400 \mid X > 400$. Using the CDF from part (iv)(a), the required probability is:

Assignment X5 Solutions (clarification)

Question 10 ***(added on 15 March 2022)***

In part (ii), the naïve Bayes approach is being applied by considering random variables denoting the letter in each position of the message (ignoring spaces). Let these random variables be $X_p \;\; p \in \{1,2,...,21\}$. Let the set of values that these random variables can take be:

$$\{A, G, H, I, N, O, T, U, OTHER\}$$

For the message given in part (ii), for each of the 5 languages, we want to calculate:

$$P(L_j \mid X_1 = OTHER, X_2 = O, ..., X_{21} = T)$$

where $L_j \;\; j \in \{1,2,3,4,5\}$ represents the 5 languages, English, French, German, Spanish and Italian.

These probabilities can be written as:

$$P(L_j \mid X_1 = OTHER, X_2 = O, \ldots, X_{21} = T) \propto P(X_1 = OTHER, X_2 = O, \ldots, X_{21} = T \mid L_j)P(L_j)$$

Under the assumption of the naïve Bayes approach, we have:

$$P(X_1 = OTHER, X_2 = O, \ldots, X_{21} = T \mid L_j) = P(X_1 = OTHER \mid L_j)(X_2 = O \mid L_j)\ldots(X_{21} = T \mid L_j)$$

If we assume that the proportions in the table can be used for each of the RHS probabilities, we can work out this out for each language.

For example, for English:

$$P(X_1 = OTHER, X_2 = O, \ldots, X_{21} = T \mid English) = 0.51 \times 0.07 \times \ldots \times 0.09$$

$$= 0.07^1 \times 0.07^3 \times 0.07^4 \times 0.09^1 \times 0.03^3 \times 0.51^9$$

where the powers of the probabilities come from the counts of each letter in the message.

The counts are given in the table below:

Letter	A	G	H	I	N	O	T	U	Other
Count	$A_i = 0$	$G_i = 0$	$H_i = 0$	$I_i = 1$	$N_i = 3$	$O_i = 4$	$T_i = 1$	$U_i = 3$	$\Omega_i = 9$

This probability has the same underlying structure as that calculated in the approach used in the solutions, which treats the counts of each letter in the message as a sample from a multinomial distribution. The only difference is the multinomial coefficient, which is the same across languages.

Assignment Y1 Solutions

Question 3 ***(added on 21 January 2022)***

There is a typo in the second paragraph of part (v). It should read:

Specifically, as $e^{\hat{\beta}} = 0.3319$, then according to the model, the hazard for patients undergoing the new treatment is 66.81% lower than those that aren't.

Assignment Y2 Questions

Question 4 ***(added on 15 March 2022)***

Part (v)(a) should ask for a matrix with the same number of rows as the test data set, not the same number of rows as the entire `swiss` data set. It should read:

Repeat the steps in parts (iv)(a) and (iv)(b) to generate 1,000 decision trees on bootstrapped samples of the training data, calculating (and storing) the predicted value of `Fertility` for each province of the test data for each tree. You should store your results in a matrix called `preds` that has the same number of rows as the test data and 1,000 columns, one for each generated decision tree.

4 PBOR

Chapter 1 Poisson Processes – Solutions *(added on 21 January 2022)*

The solution for part (vi)(b) incorrectly calculates the probability that the shop collects more than £19,800 in any given week instead of at least £19,800. It should read:

We can use the `length()` function to find the proportion of entries in our `s` vector that are at least 19,800:

```
    length(s[s>=19800])/length(s)
[1] 0.805
```

Chapter 8 – Course Notes

Page 21 ***(added on 21 December 2021)***

There is an error in the section reference at the bottom of the quoted Core Reading. This has been corrected in the latest version of the document. In the old version, this should read:

The Breslow method is consistent with the theory presented in Section 4.2.

Chapter 9 – Course Notes

Page 13 ***(added on 15 March 2022)***

The exact exposed to risk quoted at the end of part (ix) is incorrect. It should read:

The census method assumes that the number of in-force policies varies linearly over each calendar year. Even though there were some quite big changes in these numbers from year to year in this example, the census estimate of the exposed to risk (70.5) was quite close to the exact value (70.39836). As a result, the two estimates of the force of mortality were quite similar.

Chapters 10 and 11 – Course Notes

Page 9 ***(added on 12 April 2022)***

The goodness-of-fit test in Section 2.3 doesn't consider the sizes of the expected values. Using the rule of thumb of ensuring that all expected values are larger than 5, one way of checking the expected values and combining the age groups is as follows:

```
    Grad$EXPECTED

 [1]   0.8041770   0.9131601   1.0315995   1.1677738   1.3120707
 [6]   1.4576501   1.6193731   1.8153408   2.0353466   2.2830085
[11]   2.5848017   2.9378952   3.3591593   3.8450230   4.3145740
[16]   4.7508712   5.2695751   5.9773790   6.8207455   7.7441477
…
[51] 227.0439163
```

Many of the early values are less than 5. Using `cumsum()` to check how many we need to combine for the first few ages:

```
cumsum(Grad$EXPECTED)
```

```
 [1]    0.804177    1.717337    2.748937    3.916710    5.228781
 [6]    6.686431    8.305804   10.121145   12.156492   14.439500
…
[51] 2299.975967
```

So, we need to combine the first 5 to get the expected value over 5. Checking the 6th age group onwards:

```
cumsum(Grad$EXPECTED[6:nrow(Grad)])
```

```
 [1]    1.457650    3.077023    4.892364    6.927711    9.210719
 …
[46] 2294.747186
```

So, we need to combine the 6th to 9th ages. Checking the 10th age onwards:

```
cumsum(Grad$EXPECTED[10:nrow(Grad)])
```

```
 [1]    2.283008    4.867810    7.805705   11.164865   15.009888
 …
[41] 2060.775559 2287.819475
```

We need to combine the 10th to 12th ages. Checking the 13th age onwards:

```
cumsum(Grad$EXPECTED[13:nrow(Grad)])
```

```
 [1]    3.359159    7.204182   11.518756   16.269627   21.539203
 …
[36] 1661.521327 1847.882379 2052.969854 2280.013770
```

We need to combine the 13th and 14th ages. Checking the 15th age onwards:

```
cumsum(Grad$EXPECTED[15:nrow(Grad)])
```

```
 [1]    4.314574    9.065445   14.335020   20.312399   27.133145
 …
[36] 2045.765671 2272.809588
```

We need to combine the 15th and 16th ages. The rest of the expected values are over 5. Combining the expected values:

```
E.comb = c(sum(Grad$EXPECTED[1:5]),
           sum(Grad$EXPECTED[6:10]),
           sum(Grad$EXPECTED[10:12]),
           sum(Grad$EXPECTED[13:14]),
           sum(Grad$EXPECTED[15:16]),
           Grad$EXPECTED[17:nrow(Grad)])
E.comb
```

```
 [1]   5.228781   9.210719   7.805705   7.204182   9.065445
 [6]   5.269575   5.977379   6.820746   7.744148   8.747494
…
[36] 149.847251 167.628481 186.361052 205.087475 227.043916
```

Combining the observed values:

```
O.comb = c(sum(Grad$DEATHS[1:5]),
           sum(Grad$DEATHS[6:10]),
           sum(Grad$DEATHS[10:12]),
           sum(Grad$DEATHS[13:14]),
           sum(Grad$DEATHS[15:16]),
           Grad$DEATHS[17:nrow(Grad)])
O.comb
```

```
 [1]   7  14   6   8  15   9   6   5   1   9  15  13   5   7  18
[16]  17  25  26  33  19  25  32  31  46  59  57  74  64  89  73
[31] 105 107 124 156 164 145 189 202 239 239
```

Calculating the observed value of the test statistic:

```
ZX.comb = (O.comb - E.comb) / sqrt(E.comb)

(obs.test.stat <- sum(ZX.comb^2))
```

```
[1] 73.33416
```

Calculating the critical value at the 5% level:

```
(m <- length(ZX.comb))
```

```
[1] 40
```

```
(dof <- m - 2)
```

```
[1] 38
```

```
qchisq(0.95, dof)
```

```
[1] 53.38354
```

So, as 73.33 > 53.38, there is sufficient evidence to reject the null hypothesis at the 5% significance level. Therefore, it is reasonable to conclude that the graduated rates are *not* the true underlying mortality rates.

Alternatively, we can check the *p*-value:

```
1 - pchisq(obs.test.stat, dof)
```

```
[1] 0.0005022726
```

As this is lower than 5%, we reach the same conclusion as above.

Chapters 10 and 11 – Solutions

Page 11 ***(added on 12 April 2022)***

The goodness-of-fit test in Question 10.2 doesn't consider the sizes of the expected values. Using the rule of thumb of ensuring that all expected values are larger than 5, one way of checking the expected values and combining the age groups is as follows:

```
(E = splines$ETR * splines$GRAD)
```

```
 [1]  6.465341  4.963548  3.687233  3.643057  2.117585  1.128956
 [7]  1.736819  1.302198  1.268237  1.428041  1.515226  2.273344
[13]  2.293974  3.354250  4.920564  5.313610  6.230630  7.381566
[19]  8.249834  8.891049  9.522978 10.042155  9.521691  7.506374
[25]  9.236048  8.901399  8.609211  7.354864  8.648749  9.079818
[31] 10.514835  9.910939 11.058651 11.071241  9.962045 12.413825
[37] 12.954818 14.630655 17.168363 17.723861
```

Many of the early values are less than 5, although the first values is over 5. Using `cumsum()` to check how many we need to combine for the first group of ages, ignoring the first age:

```
cumsum(E[-1])
```

```
 [1]   4.963548   8.650781  12.293838  14.411422  15.540379
 …
[36] 238.009362 252.640017 269.808380 287.532241
```

So, we need to combine the 2nd and 3rd ages (taking into account that we dropped the first age in the above) to get the expected value over 5. Checking the 4th age group onwards:

```
cumsum(E[4:nrow(splines)])
```

```
 [1]   3.643057   5.760642   6.889598   8.626417   9.928615
 …
[36] 261.157599 278.881460
```

So, we need to combine the 4th and 5th ages. Checking the 6th age onwards:

```
cumsum(E[6:nrow(splines)])
```

```
 [1]   1.128956   2.865775   4.167973   5.436210   6.864252
 …
[31] 210.643121 223.597939 238.228594 255.396958 273.120819
```

We need to combine the 6th to 9th ages. Checking the 10th age onwards:

```
cumsum(E[10:nrow(splines)])
```

```
 [1]   1.428041   2.943267   5.216611   7.510585  10.864835
 …
[31] 267.684608
```

We need to combine the 10th to 12th ages. Checking the 13th age onwards:

```
cumsum(E[13:nrow(splines)])
```

```
 [1]   2.293974   5.648224  10.568789  15.882398  22.113028
 …
[26] 227.575773 244.744136 262.467998
```


We need to combine the 13th and 14th ages. Checking the 15th age onwards:

```
cumsum(E[15:nrow(splines)])
```

```
 [1]   4.920564  10.234174  16.464804  23.846370  32.096204
 …
[26] 256.819773
```

This suggests combining the 15th and 16th ages. However, looking at the 16th age:

```
E[16]
```

```
[1] 5.31361
```

This is over 5, so instead we can put the 15th age into the previous group, containing the 13th and 14th ages. The rest of the expected values are over 5. Combining the expected values:

```
E.comb = c(sum(E[1]),
           sum(E[2:3]),
           sum(E[4:5]),
           sum(E[6:9]),
           sum(E[10:12]),
           sum(E[13:15]),
           E[16:nrow(splines)])

E.comb
```

```
 [1]  6.465341  8.650781  5.760642  5.436210  5.216611 10.568789
 [7]  5.313610  6.230630  7.381566  8.249834  8.891049  9.522978
[13] 10.042155  9.521691  7.506374  9.236048  8.901399  8.609211
[19]  7.354864  8.648749  9.079818 10.514835  9.910939 11.058651
[25] 11.071241  9.962045 12.413825 12.954818 14.630655 17.168363
[31] 17.723861
```

Combining the observed values:

```
O.comb = c(sum(O[1]),
           sum(O[2:3]),
           sum(O[4:5]),
           sum(O[6:9]),
           sum(O[10:12]),
           sum(O[13:15]),
           O[16:nrow(splines)])
O.comb
```

```
 [1]  8.00037  6.99989  5.00005  6.99996  6.00000  8.00018
 [7]  5.00016  6.99992  8.99980  8.99980  7.99976  8.99976
[13]  9.99999  9.99973  7.00040  8.99968  8.00037  8.99961
[19]  7.99968  9.00036  8.99968 10.99980  9.99992 10.00029
[25] 11.00028 10.00050 11.99992 13.00024 14.99983 18.00009
[31] 17.00028
```

Calculating the observed value of the test statistic:

```
ZX.comb = (O.comb - E.comb) / sqrt(E.comb)

(obs.test.stat <- sum(ZX.comb^2))
```

```
[1] 3.089105
```

Calculating the critical value at the 5% level:

```
(m <- length(ZX.comb))
```

```
[1] 31
```

```
(dof <- m - 6)
```

```
[1] 25
```

```
qchisq(0.95, dof)
```

```
[1] 37.65248
```

So, as 3.09 < 37.65, there is insufficient evidence to reject the null hypothesis at the 5% significance level. We conclude that it is reasonable to assume that the graduated rates reflect the true mortality rates according to this test.

Alternatively, we can check the *p*-value:

```
1 - pchisq(obs.test.stat, dof)
```

```
[1] 1
```

As this is higher than 5%, we reach the same conclusion as above.

Chapters 10 and 11 – Summary

Page 5 ***(added on 31 January 2022)***

There is an error in the code for counting the number of positive individual standardised deviations. The `abs()` function should not be used. The code should be:

```
(n1 = length(Grad$<ZX>[Grad$<ZX> > 0]))
```

Chapter 12 – Course Notes

Page 7 ***(added on 12 April 2022)***

There is a typo in the second line of Exercise 4. It should say:

For this Exercise you should use your estimates of b_x and k_t from Exercise 2 (ii)(c).

Page 22 ***(added on 12 April 2022)***

Throughout exercise 4, the object `kt.svd` should be used instead of `kt`. This affects the graphs slightly and the calculated fitted values and forecasts. The fitted values and forecasts should be:

```
(fitted.mort60 = exp(ax[1] + bx.svd[1] * kt.svd))
```

```
[1] 0.01300107 0.01268702 0.01253275 0.01242453 0.01236440
```

```
(mu = (kt.svd[5] - kt.svd[1]) / 4)
```

```
-0.04403012
```



```
(kt.forecasts = kt.svd[5] + mu * (2015:2030 - 2014))

 [1] -0.1102145 -0.1542446 -0.1982747 -0.2423048 -0.2863349 -0.3303650
 [7] -0.3743952 -0.4184253 -0.4624554 -0.5064855 -0.5505156 -0.5945457
[13] -0.6385758 -0.6826060 -0.7266361 -0.7706662

(proj.mort60 = exp(ax[1] + bx.svd[1] * kt.forecasts))

 [1] 0.01221017 0.01205786 0.01190745 0.01175891 0.01161223 0.01146738
 [7] 0.01132433 0.01118307 0.01104357 0.01090582 0.01076978 0.01063543
[13] 0.01050277 0.01037175 0.01024237 0.01011461
```

Chapters 13 and 14 – Introduction

Page 16 ***(added on 21 December 2021)***

There is an error in a section reference in the Core Reading for the method of seasonal means. This has been corrected in the latest version of the document. In the old version, this should read:

In R the function `decompose()` **can be used to obtain both the moving average and seasonal means described in Sections 1.5 and 1.6.**

Chapters 13 and 14 – Fitting a distribution – Solutions

Question 13-14.11 ***(added on 12 April 2022)***

This question fits an ARMA model to the series Yt.csv. However, looking at a graph of the series, it does not appear to be stationary. The series looks to have a linear trend, which should really be removed before modelling as an ARMA process.

Chapter 15 – Introduction

Page 16 ***(added on 21 December 2021)***

There is an error in the section reference near the bottom of the box. This has been corrected in the latest version of the document. In the old version, this should read

The method of percentiles is covered in Section 3.3.

Chapter 15 Fitting a distribution – Solutions

Page 8 ***(added on 21 December 2021)***

The labels for the axes in the Q-Q plot are the wrong way round. The correct Q-Q plot is created by:

```
qqplot(comparison.qs, x, xlab = "Quantiles of exp(0.003)",
ylab = "Sample quantiles", main = "Q-Q plot")
```

Chapters 19 and 20 – Summary

Page 8 ***(added on 9 February 2022)***

There is a typo in the R code near the bottom of the page. The for loop generating the sample aggregate claims for each policy should read:

```
for (j in 1:policies) {
S[j] <- sum(rXXX(N[j], parameters of claim distribution))
}
```


5 Revision Booklets

Revision Booklet 1

Page 25 ***(added on 29 June 2022)***

There is a typo in part (i). It should say:

(i) Explain whether Y_t is a Markov process. [2]

Revision Booklet 2

Page 80 ***(added on 9 February 2022)***

The solution for Question 4 part (iv) has an error in part of the workings. The second equation should be:

$$p_{L,L}(2) = P(X_{t+2} = L \mid X_t = L+) \times P(X_t = L+ \mid X_t = L+ \text{ or } X_t = L-)$$
$$+ P(X_{t+2} = L \mid X_t = L-) \times P(X_t = L- \mid X_t = L+ \text{ or } X_t = L-)$$

Revision Booklet 11

Page 86 ***(added on 29 June 2022)***

There is a typo in the calculation of σ^2 near the top of the page. It should say:

$$\sigma^2 = \ln\left(\frac{413{,}918.40}{2{,}213.06^2} + 1\right) = 0.0811318$$

6 Mocks

Mock 1 – Paper A – Solutions

Page 15 (Question 6) ***(added on 12 April 2022)***

There is a typo in the calculation of $F_B(40)$ around halfway down the page. The final number is correct, but the calculation should be:

$$F_B(40) = 1 - \left(\frac{300}{500}\right)^4 = 0.870400$$

Mock 1 – Paper B – Solutions

Page 22 (Question 3) ***(added on 12 April 2022)***

There is an error in the creation of `package.tree`. The `tree()` function should include AGE. It should be:

```
package.tree = tree(SALE ~ SEX + HIGH + MARRIED + CHILDREN + AGE,
data = happy_train)
```

The code in the R solutions file is correct and the rest of the solutions use the `package.tree` object created from this correct code.

7 ASET

September 2020 – Paper A solutions

Page 13 ***(added on 12 April 2022)***

There is a typo in the final line of the table at the top of the page. The last row should be:

j	t_j	n_j	d_j	$\hat{\lambda}_j = \frac{d_j}{n_j}$	$1-\hat{\lambda}_j$
…	…	…	…	…	…
6	11	1	1	1	0