

-Solutions Manual-

Actuarial
Statistics
with
R

Theory and Case Studies

Guojun Gan, PhD, FSA Emiliano A. Valdez, PhD, FSA

Copyright © 2018 by ACTEX Learning, a division of SRBooks Inc.

All rights reserved. No portion of this book may be reproduced in any form or by any means without the prior written permission of the copyright owner.

Requests for permission should be addressed to
ACTEX Learning
4 Bridge Street
New Hartford CT 06057

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Cover design by Jeff Melaragno

ISBN: 978-1-63588-552-1

Contents

1 Simple Linear Regression	1
2 Case Study: Implementing the CAPM	9
3 Multiple Linear Models	19
4 Case Study: Predicting Intraday Movements	27
5 Case Study: Estimating Fair Market Values	37
6 Generalized Linear Models	41
7 Case Study: Predicting Demand	49
8 Case Study: Modeling the Number of Auto Claims	55
9 Case Study: Modeling the Loss Severity	59
10 Decision Trees	65
11 Case Study: Decision Trees	69
12 Data Clustering	71
13 Case Study: Clustering Variable Annuity Policies	77
14 Principal Component Analysis	81
15 Case Study: PCA on Interest Rate Swaps	85
16 Time Series Models	89
17 Case Study: Forecasting Exchange Rates	95

18 Case Study: Profitability Analysis	105
19 Case Study: Simulating the Future Lifetime	113
A Introduction to R	121

Chapter 2

Case Study: Implementing the Capital Asset Pricing Model

Answer 2.1 If we do not specify `stringsAsFactors=FALSE` in the call of `read.csv`, the dates are automatically converted to factors. As a result, the mode of the `Date` variable is numeric.

Answer 2.2 We can convert the vector of strings to a vector of dates as follows

```
1 as.Date(mfc$Date, '%Y-%m-%d')
```

Answer 2.3

(a) We can write the function `calculateStd(x)` as follows:

```
1 calculateStd <- function(x) {  
2   n <- length(x)  
3   barx <- sum(x) / n  
4   varx <- sum( (x-barx)^2 ) / (n-1)  
5   return(sqrt(varx))  
6 }
```

Calling `calculateStd(dat10y$mfc)` returns 0.1145846.

(b) We can write the function `calculateCorr(x,y)` as follows:

```
1 calculateCorr <- function(x, y) {  
2   n <- length(x)  
3   barx <- sum(x) / n  
4   bary <- sum(y) / n
```

```

5  sx <- calculateStd(x)
6  sy <- calculateStd(y)
7  covxy <- sum( (x - barx) * (y - bary)) / (n - 1)
8  return( covxy / (sx * sy) )
9  }

```

Calling `calculateCorr(dat10y$mfc, dat10y$sp500)` returns 0.725671.

Answer 2.4 We can calculate the regression coefficients using the formulas as follows:

```

1  r <- with(dat10y, cor(mfc, sp500))
2  sx <- with(dat10y, sd(sp500))
3  sy <- with(dat10y, sd(mfc))
4  barx <- with(dat10y, mean(sp500))
5  bary <- with(dat10y, mean(mfc))
6  beta <- r * sy / sx
7  alpha <- bary - beta * barx

```

Answer 2.5

(a) We can calculate the sum as follows:

```

1  barx <- mean(x)
2  sx <- sd(x)
3  n <- length(x)
4  w <- (x - barx) / (sx^2 * (n-1))
5  sum(w * y)

```

(b) Yes, the value of the sum is equal to the slope estimate $\hat{\beta}$.

Answer 2.6

(a) We can calculate these quantities as follows:

```

1  haty <- fit$fitted.values
2  bary <- mean(dat10y$mfc)
3  SST <- sum( (dat10y$mfc - bary)^2 )
4  SSE <- sum( (dat10y$mfc - haty)^2 )
5  SSR <- sum( (haty - bary)^2 )
6  SSE + SSR - SST

```