`stocins` Package

Nathan Esau

April 3, 2017

Contents

Chapter 1

Overview

This package can be used to analyze the risk of an insurance portfolio using stochastic interest rate models. The focus is on calculating the first three moments of the present value of benefit random variable for a portfolio of endowment or term insurance contracts. Several stochastic interest rate models are implemented including the Wiener process, the Ornstein-Uhlenbeck process and a Second Order Stochastic Differential equation for the force of interest. The references for this package are (Parker, 1992) and (Parker, 1997).

This package implements the following:

- Functions for an Wiener process, Ornstein-Uhlenbeck process and Second order stochastic differential equation as shown in Chapter 2 and 3 of (Parker, 1992).
- Moments and density function for the present value of a term and endowment insurance on a single life as shown in Chapter 4 of (Parker, 1992).
- The first three moments for a portfolio of term and endowment contracts as shown in Chapter 5 of (Parker, 1992).
- The first two moments for a group of term and endowment contracts as shown in (Parker, 1997).

Various classes implement this functionality:

- `insurance` classes
  - `isingle` classes, i.e. `termsingle` and `endowsingle` for a life insurance policy issued to a single life
  - `iport` classes, i.e. `termport` and `endowport` for a portfolio of identical insurance policies
  - `igroup` class for a group of portfolios of insurance policies
- `iratemodel` classes for the force of interest
  - `determ` for the deterministic interest rate

- `wiener` for the wiener process
- `ou` for the Ornstein-Uhlenbeck process
- `second` for the Second Order Stochastic Differential Equation
- `ar1` for the AR(1) process
- `arma` fopr the ARMA(2,1) process

Chapter 2

Reproducing Parker (1992) Results

Some results from (Parker, 1992) are reproduced here.

```
> library(stocins)
```

2.1 Interest Rate Models

First, the Wiener process.

```
> wienermodel = iratemodel(list(delta = 0.10, sigma = 0.01),
+                           "wiener")
> plot(function(t) delta.ev(t, wienermodel), 0, 50, col = 'black',
+      ylim = c(-0.05, 0.25), xlab = "time", ylab = "force of interest")
> plot(function(t) delta.ev(t, wienermodel) -
+        1.645 * sqrt(delta.var(t, wienermodel)), 0, 50,
+      add = TRUE, lty = 2)
> plot(function(t) delta.ev(t, wienermodel) +
+        1.645 * sqrt(delta.var(t, wienermodel)), 0, 50,
+      add = TRUE, lty = 2)
```

Next, the Ornstein-Uhlenbeck process.

```
> oumodel = iratemodel(list(delta0 = 0.1, delta = 0.06,
+     alpha = 0.1, sigma = 0.01), "ou")
> plot(function(t) delta.ev(t, oumodel), 0, 50, col = 'black',
+      ylim = c(0, 0.12), xlab = "time", ylab = "force of interest")
> plot(function(t) delta.ev(t, oumodel) -
+        1.645 * sqrt(delta.var(t, oumodel)), 0, 50,
+      add = TRUE, lty = 2)
> plot(function(t) delta.ev(t, oumodel) +
+        1.645 * sqrt(delta.var(t, oumodel)), 0, 50,
+      add = TRUE, lty = 2)
```

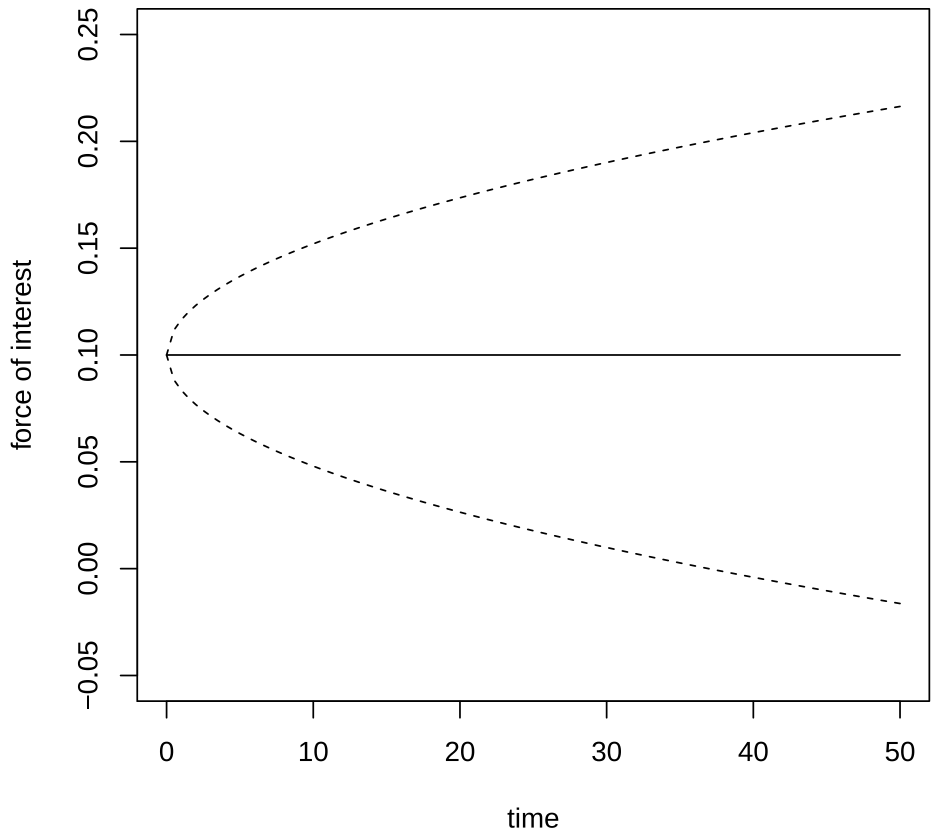

Figure 2.1: Figure 2.1 from Parker (1992)

Next, the second order stochastic differential equation.

```
> secondmodel = iratemodel(params = list(alpha1 = -0.50, alpha2 = -0.04,
+   delta0prime = 0.05, delta0 = 0.10, delta = 0.06, sigma = 0.01), "second")
> plot(function(t) delta.ev(t, secondmodel), 0, 50, col = 'black',
+      ylim = c(-0.05, 0.25), xlab = "time", ylab = "force of interest")
> plot(function(t) delta.ev(t, secondmodel) -
+        1.645 * sqrt(delta.var(t, secondmodel)), 0, 50,
+      add = TRUE, lty = 2)
> plot(function(t) delta.ev(t, secondmodel) +
+        1.645 * sqrt(delta.var(t, secondmodel)), 0, 50,
+      add = TRUE, lty = 2)
```

2.2 Present Value

First, the expected value plots with varying α.

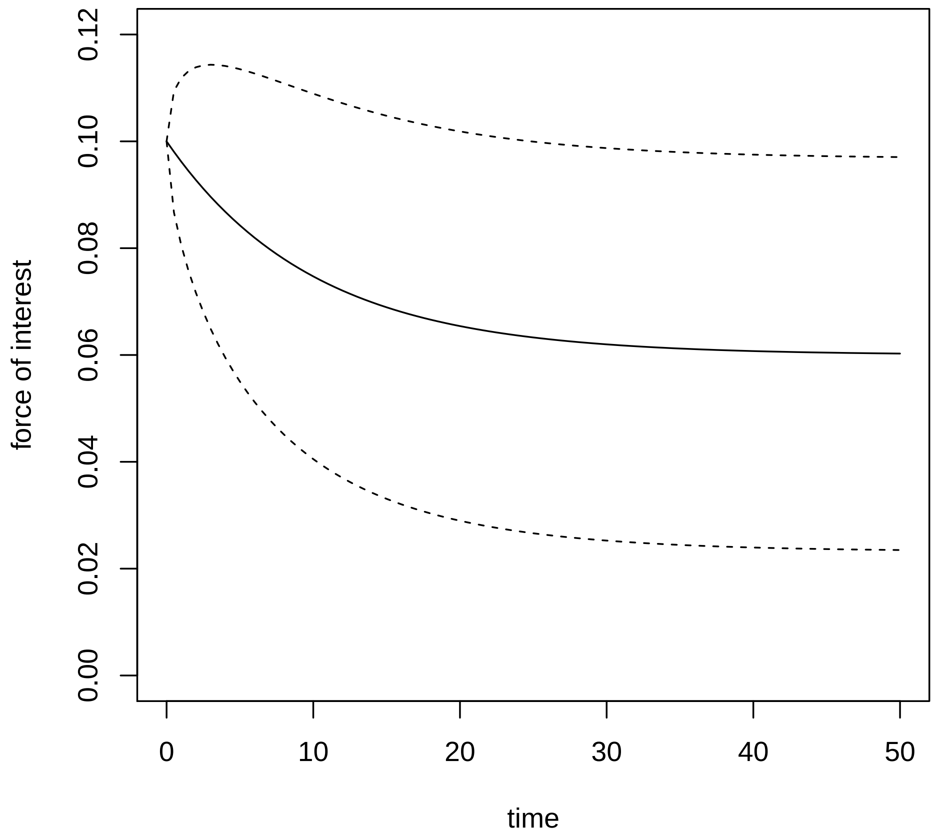

Figure 2.2: Figure 2.2 from Parker (1992)

```
> irm = iratemodel(list(delta0 = 0.1, delta = 0.06,
+     alpha = 0.1, sigma = 0.01), "ou")
> g <- expand.grid(x = seq(0, 100, 5), y = seq(0.05, 0.5, 0.05))
> for(i in 1:nrow(g))
+ {
+   irm = iratemodel(list(delta0 = 0.1, delta = 0.06,
+     alpha = g$y[i], sigma = 0.01), "ou")
+   g$z[i] = pv.moment(1, t = g$x[i], irm)
+ }
> lattice::wireframe(z ~ x * y, data = g, drape = TRUE, col = 'black',
+             col.regions = 'white', aspect = c(1, 0.65),
+             colorkey = FALSE, xlab = "t", ylab = expression(alpha),
+             zlab = "", screen = list(z = 340, x = -70, y = -20),
+             scales = list(arrows = FALSE, col = 'black', font = 10),
+             cex = 0.8, ylim = c(0.00, 0.50),
+             par.settings = list(regions=list(alpha = 0.3),
+             axis.line = list(col = "transparent")), zoom = 0.95)
```

Next, the expected value plot with varying σ.

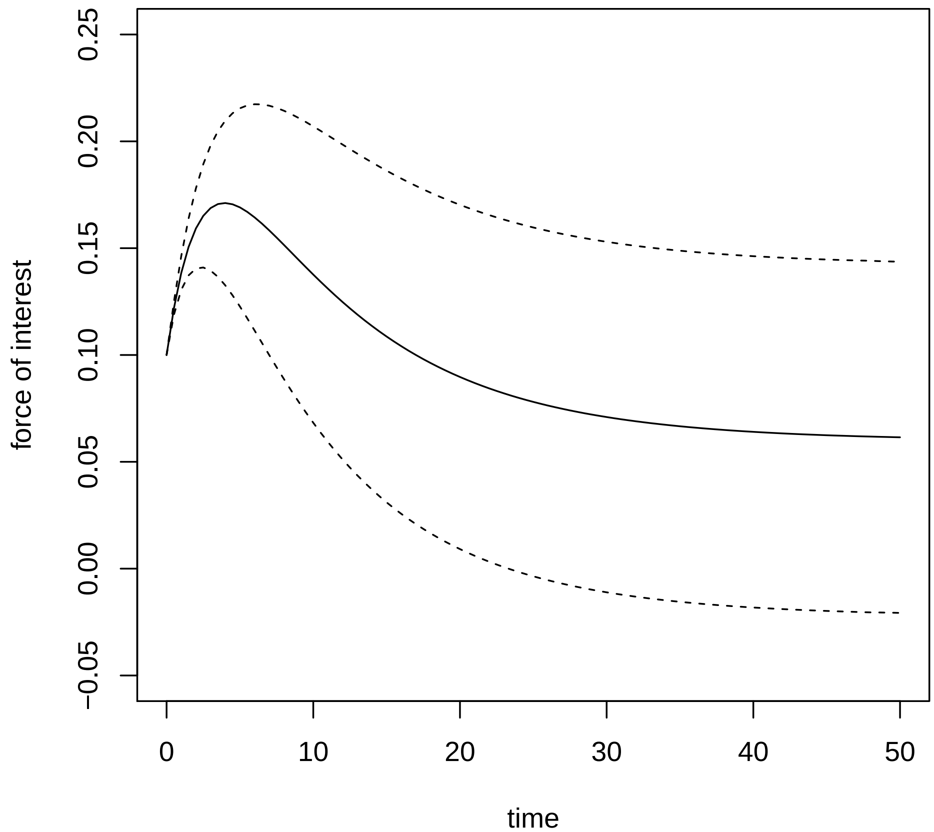

Figure 2.3: Figure 2.3 from Parker (1992)

```
> g <- expand.grid(x = seq(0, 100, 5), y = seq(0, 0.03, 0.005) + 0.001)
> for(i in 1:nrow(g))
+ {
+   irm = iratemodel(list(delta0 = 0.1, delta = 0.06,
+     alpha = 0.1, sigma = g$y[i]), "ou")
+   g$z[i] = pv.moment(1, t = g$x[i], irm)
+ }
> lattice::wireframe(z ~ x * y, data = g, drape = TRUE, col = 'black',
+             col.regions = 'white', aspect = c(1, 0.65),
+             colorkey = FALSE, xlab = "t", ylab = expression(sigma),
+             zlab = "", screen = list(z = 340, x = -70, y = -20),
+             scales = list(arrows = FALSE, col = 'black', font = 10),
+             cex = 0.8, ylim = c(0.00, 0.03) + 0.001,
+             par.settings = list(regions=list(alpha = 0.3),
+             axis.line = list(col = "transparent")), zoom = 0.95)
```

Next, the standard deviation plot with varying α.

```
> g <- expand.grid(x = seq(0, 100, 5), y = seq(0.05, 0.5, 0.05))
```

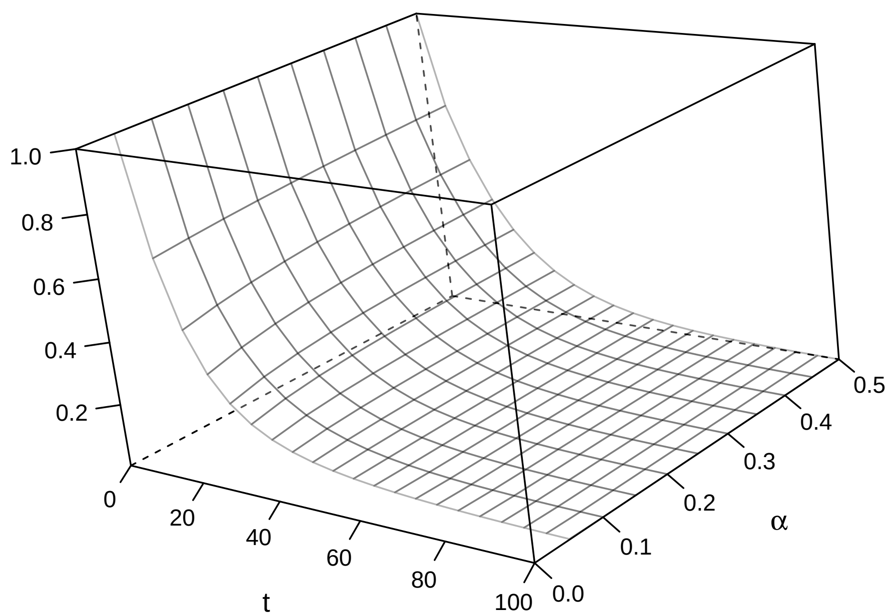

Figure 2.4: Figure 3.1(a)from Parker (1992)

```
> for(i in 1:nrow(g))
+ {
+   irm = iratemodel(list(delta0 = 0.1, delta = 0.06,
+     alpha = g$y[i], sigma = 0.01), "ou")
+   g$z[i] = sqrt(pv.var(t = g$x[i], irm))
+ }
> lattice::wireframe(z ~ x * y, data = g, drape = TRUE, col = 'black',
+             col.regions = 'white', aspect = c(1, 0.65),
+             colorkey = FALSE, xlab = "t", ylab = expression(alpha),
+             zlab = "", screen = list(z = 340, x = -70, y = -20),
+             scales = list(arrows = FALSE, col = 'black', font = 10),
+             cex = 0.8, ylim = c(0.00, 0.50),
+             par.settings = list(regions=list(alpha = 0.3),
+             axis.line = list(col = "transparent")), zoom = 0.95)
```

Next, the stnadard deviation plot with varying σ.

```
> g <- expand.grid(x = seq(0, 100, 5), y = seq(0, 0.025, 0.005) + 0.001)
> for(i in 1:nrow(g))
```

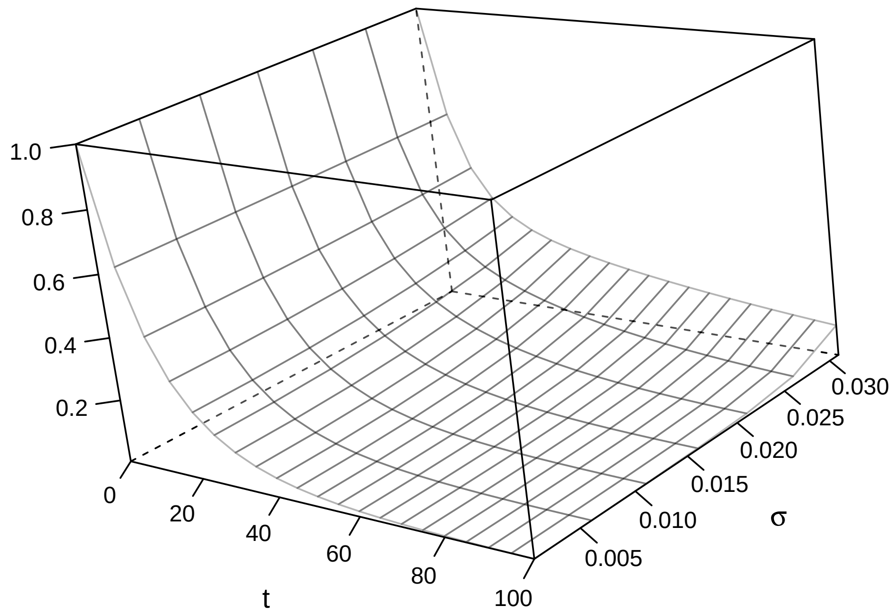

Figure 2.5: Figure 3.1(b) from Parker (1992)

```
+ {
+   irm = iratemodel(list(delta0 = 0.1, delta = 0.06,
+     alpha = 0.1, sigma = g$y[i]), "ou")
+   g$z[i] = sqrt(pv.var(t = g$x[i], irm))
+ }
> lattice::wireframe(z ~ x * y, data = g, drape = TRUE, col = 'black',
+            col.regions = 'white', aspect = c(1, 0.65),
+            colorkey = FALSE, xlab = "t", ylab = expression(sigma),
+            zlab = "", screen = list(z = 340, x = -70, y = -20),
+            scales = list(arrows = FALSE, col = 'black', font = 10),
+            cex = 0.8, ylim = c(0.00, 0.025) + 0.001,
+            par.settings = list(regions=list(alpha = 0.3),
+            axis.line = list(col = "transparent")), zoom = 0.95)
```

Next, the skewness plot with varying α.

```
> pv.sk <- function(t, irm)
+ {
+   u1 = pv.moment(t, 1, irm)
```

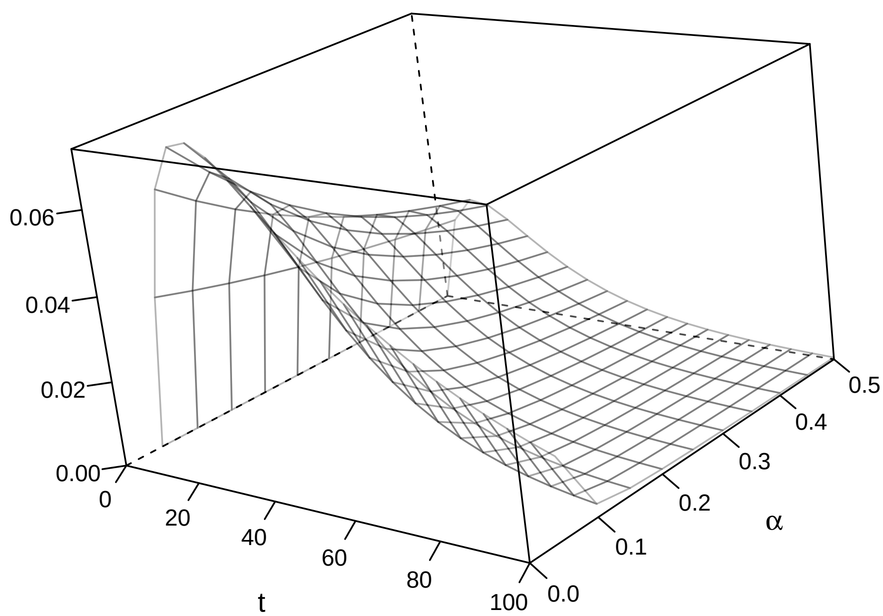

Figure 2.6: Figure 3.2(a) from Parker (1992)

```
+   u2 = pv.moment(t, 2, irm)
+   u3 = pv.moment(t, 3, irm)
+
+   num = u3 - 3*u2*u1 + 2*u1^3
+   den = pv.var(t, irm)^(3/2)
+
+   num / den
+ }
> g <- expand.grid(x = seq(0, 100, 5), y = seq(0.05, 0.5, 0.05))
> for(i in 1:nrow(g))
+ {
+   irm = iratemodel(list(delta0 = 0.1, delta = 0.06,
+     alpha = g$y[i], sigma = 0.01), "ou")
+   g$z[i] = pv.sk(t = g$x[i], irm)
+ }
> lattice::wireframe(z ~ x * y, data = g, drape = TRUE, col = 'black',
+             col.regions = 'white', aspect = c(1, 0.65),
+             colorkey = FALSE, xlab = "t", ylab = expression(alpha),
```

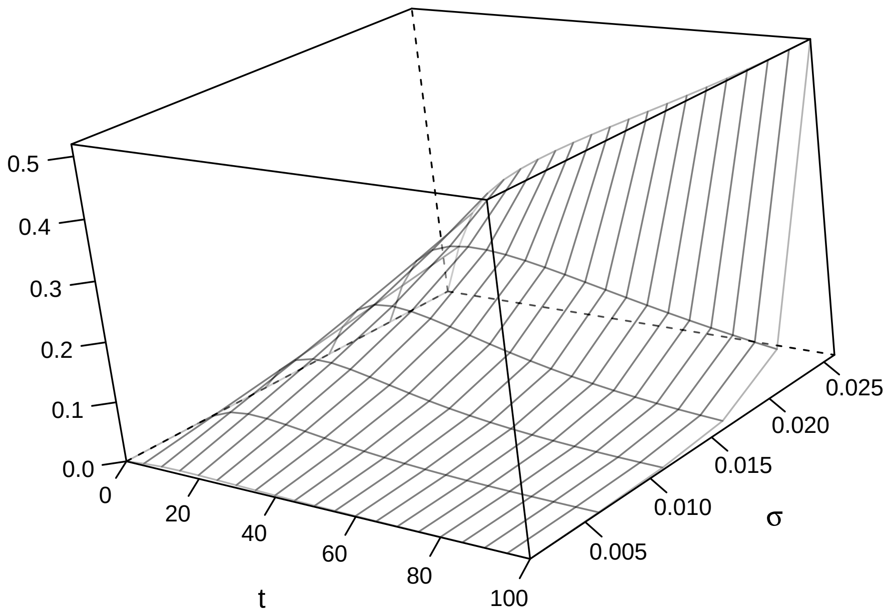

Figure 2.7: Figure 3.2(b) from Parker (1992)

```
+             zlab = "", screen = list(z = 340, x = -70, y = -20),
+             scales = list(arrows = FALSE, col = 'black', font = 10),
+             cex = 0.8, ylim = c(0.00, 0.50),
+             par.settings = list(regions=list(alpha = 0.3),
+             axis.line = list(col = "transparent")), zoom = 0.925)
```

Next, the skewness plot with varying σ.

```
> g <- expand.grid(x = seq(0, 100, 5), y = seq(0, 0.03, 0.005) + 0.001)
> for(i in 1:nrow(g))
+ {
+   irm = iratemodel(list(delta0 = 0.1, delta = 0.06,
+     alpha = 0.1, sigma = g$y[i]), "ou")
+   g$z[i] = pv.sk(t = g$x[i], irm)
+ }
> lattice::wireframe(z ~ x * y, data = g, drape = TRUE, col = 'black',
+             col.regions = 'white', aspect = c(1, 0.65),
+             colorkey = FALSE, xlab = "t", ylab = expression(sigma),
+             zlab = "", screen = list(z = 340, x = -70, y = -20),
```

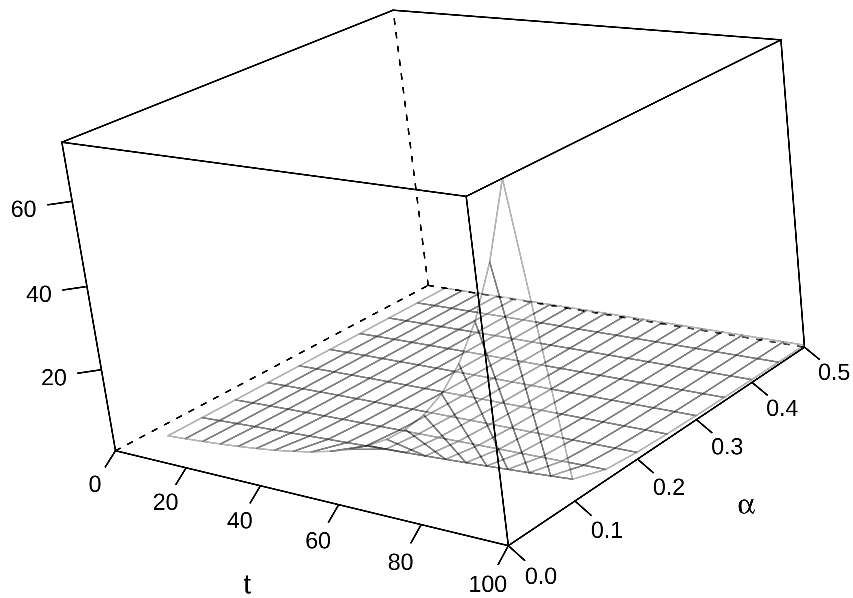

Figure 2.8: Figure 3.3(a) from Parker (1992)

```
+            scales = list(arrows = FALSE, col = 'black', font = 10),
+            cex = 0.8, ylim = c(0.00, 0.03) + 0.001,
+            par.settings = list(regions=list(alpha = 0.3),
+            axis.line = list(col = "transparent")), zoom = 0.925)
```

2.3 Single Life Insurance

First, the expected value plot.

```
> g = expand.grid(x = seq(20, 70, 10), y = seq(1, 75, 10))
> g$z = 0
> oumodel = iratemodel(list(delta0 = 0.1, delta = 0.06,
+                           alpha = 0.1, sigma = 0.01), "ou")
> for(i in 1:nrow(g)) {
+   term = insurance(list(n = g$y[i], d = 1), "isingle", "term")
+   mort = mortassumptions(list(x = g$x[i], table = "MaleMort82"))
+   g$z[i] = z.moment(1, term, mort, oumodel)
```

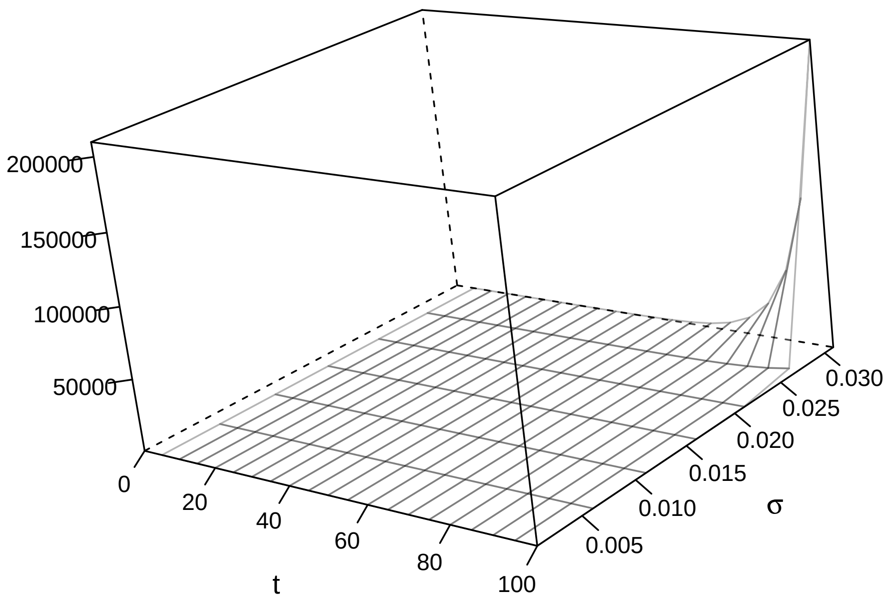

Figure 2.9: Figure 3.3(b) from Parker (1992)

```
+ }
> lattice::wireframe(z ~ x * y, data = g, drape = TRUE, col = 'black',
+         col.regions = 'white', aspect = c(1.0, 0.8), colorkey = FALSE,
+         xlab = "issue age", ylab = "n", zlab = "",
+         screen = list(z = 340, x = -70),
+     scales = list(arrows = FALSE, col="black", font = 10, cex= 1.0),
+         par.settings = list(regions=list(alpha = 0.3),
+                 axis.line = list(col = "transparent")),
+         zoom = 0.95, zlim = c(0,0.50))
```

Next, the standard deviation plot.

```
> g <- expand.grid(x = seq(1, 80, 5), y = seq(20, 70, 10))
> for(i in 1:nrow(g)) {
+   term = insurance(list(n = g$x[i], d = 1), "isingle", "term")
+   mort = mortassumptions(list(x = g$y[i], table = "MaleMort82"))
+   g$z[i] = z.sd(term, mort, oumodel)
+ }
> lattice::wireframe(z ~ x * y, data = g, drape = TRUE, col = 'black',
```

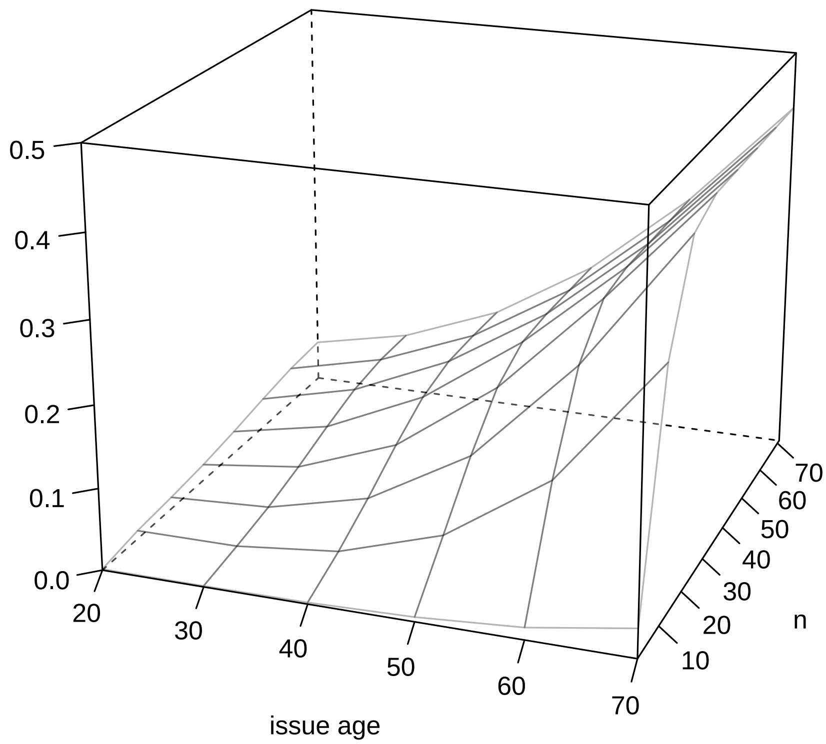

Figure 2.10: Figure 4.1 from Parker (1992)

```
+                        col.regions = 'white', aspect = c(1, 0.8),
+                        colorkey = FALSE, xlab = "n", ylab = "issue age",
+                        zlab = "", screen = list(z = 340, x = -70, y = -20),
+                        scales = list(arrows = FALSE, col = 'black', font = 10),
+                        cex = 0.8, ylim = c(20, 70),
+                        zlim = c(0, 0.40),
+                        par.settings = list(regions=list(alpha = 0.3),
+                        axis.line = list(col = "transparent")), zoom = 0.95)
```

Next, the skewness plot.

```
> g <- expand.grid(x = seq(1, 70, 5), y = seq(20, 70, 10))
> for(i in 1:nrow(g)) {
+   term = insurance(list(n = g$x[i], d = 1), "isingle", "term")
+   mort = mortassumptions(list(x = g$y[i], table = "MaleMort82"))
+   g$z[i] = z.sk(term, mort, oumodel)
+ }
> lattice::wireframe(z ~ y * x, data = g, drape = TRUE, col = 'black',
+                        col.regions = 'white', aspect = c(1, 0.8),
+                        colorkey = FALSE, xlab = "issue age", ylab = "n",
+                        zlab = "", screen = list(z = 340, x = -70, y = -20),
+                        scales = list(arrows = FALSE, col = 'black', font = 10),
```

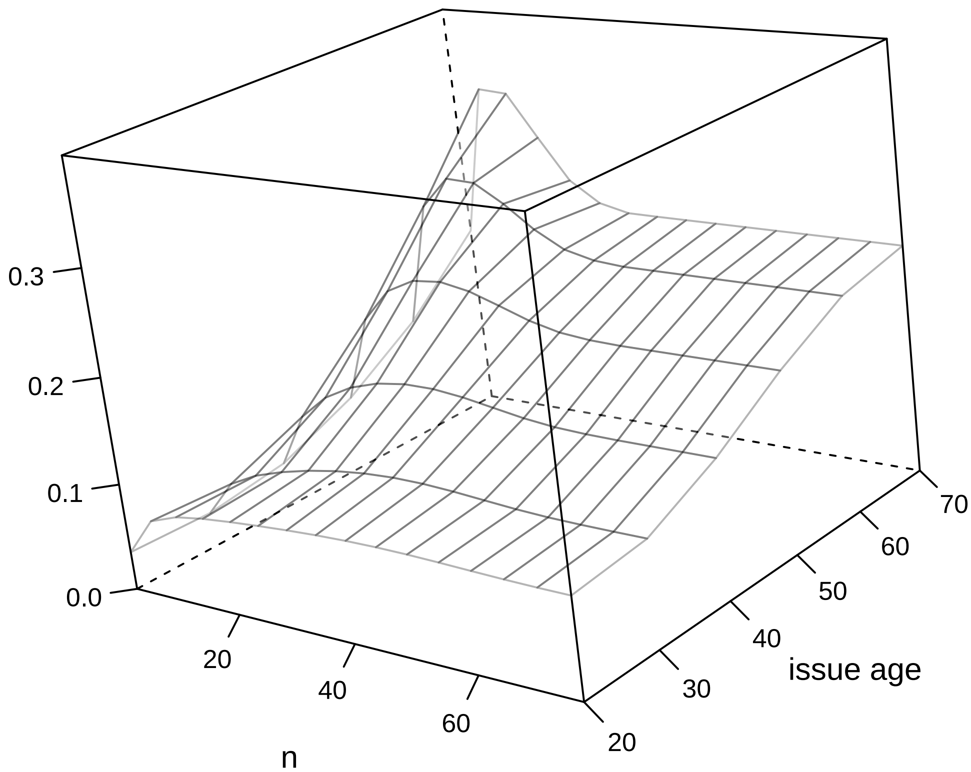

Figure 2.11: Figure 4.2 from Parker (1992)

```
+                     cex = 0.8, zlim = c(0, 30), ylim= c(0,70),
+                     par.settings = list(regions=list(alpha = 0.3),
+                     axis.line = list(col = "transparent")), zoom = 0.95)
```

Next, the pdf plot.

```
> oumodel = iratemodel(list(delta0 = 0.1, delta = 0.06,
+                           alpha = 0.1, sigma = 0.01), "ou")
> term5 = insurance(list(n = 5, d = 1), "isingle", "term")
> mort = mortassumptions(list(x = 30, table = "MaleMort82"))
> term25 = insurance(list(n = 25, d = 1), "isingle", "term")
> mort = mortassumptions(list(x = 30, table = "MaleMort82"))
> plot(function(z) z.pdf(z, term5, mort, oumodel), 0.01, 1.0,
+      ylim = c(0, 0.4), lty = 1, xlab = "z", ylab = "f(z)")
> plot(function(z) z.pdf(z, term25, mort, oumodel), 0.01, 1.0,
+      ylim = c(0, 0.4), add = TRUE, lty = 2)
> legend('topright', leg = c(paste0("term 25 [P(Z=0) = ",
```

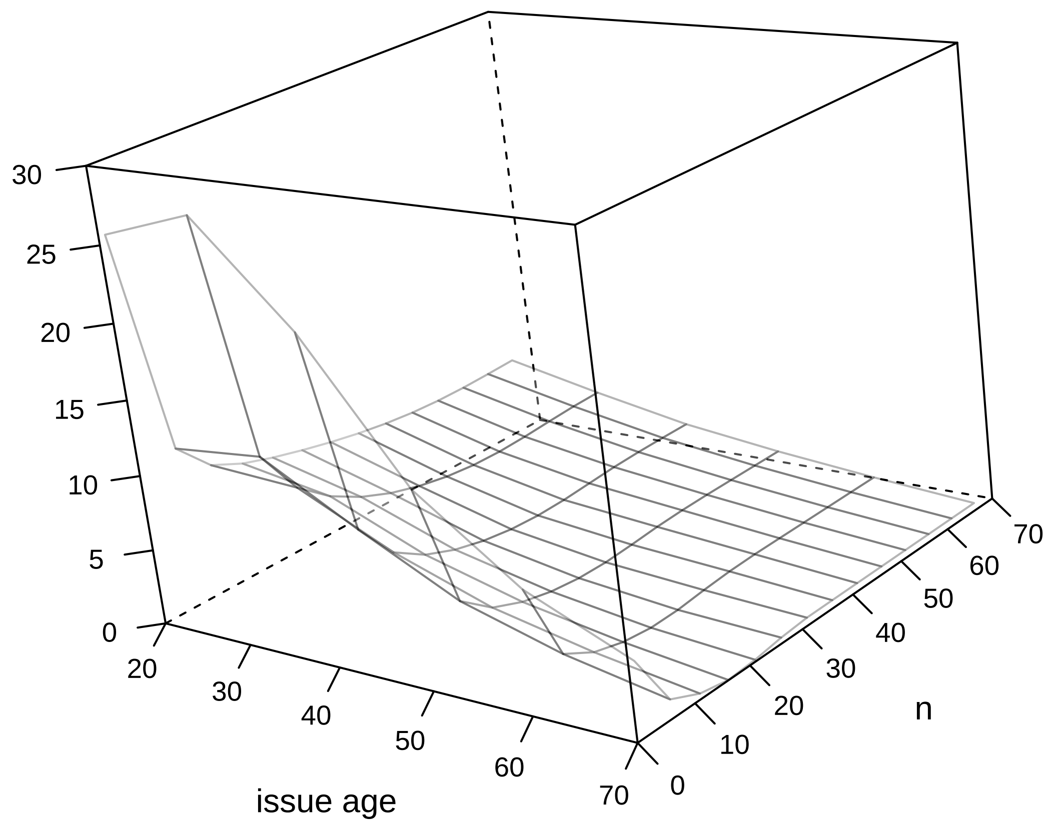

Figure 2.12: Figure 4.3 from Parker (1992)

```
+                                  round(kpx(25, mort),5), "]"),
+                            paste0("term 5   [P(Z=0) = ",
+                                  round(kpx(5, mort),5), "]")),
+        lty = c(2,1), cex = 0.8)
```

For an endowment insurance, the expected value plot.

```
> library(stocins)
> g = expand.grid(x = seq(20, 70, 10), y = seq(1, 75, 10))
> g$z = 0
> oumodel = iratemodel(list(delta0 = 0.1, delta = 0.06,
+                           alpha = 0.1, sigma = 0.01), "ou")
> for(i in 1:nrow(g)) {
+   endow = insurance(list(n = g$y[i], d = 1, e = 1), "isingle", "endow")
+   mort = mortassumptions(list(x = g$x[i], table = "MaleMort82"))
+   g$z[i] = z.moment(1, endow, mort, oumodel)
```

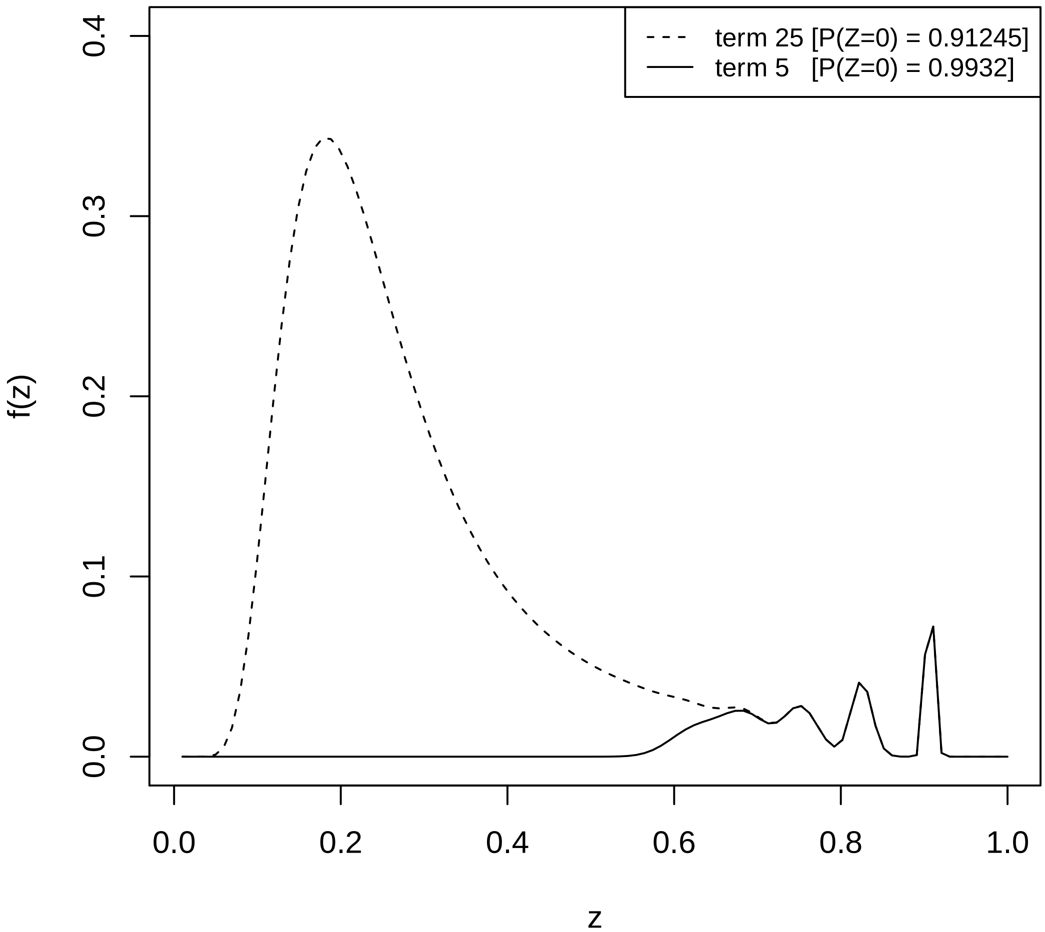

Figure 2.13: Figure 4.4 from Parker (1992)

```
+ }
> lattice::wireframe(z ~ x * y, data = g, drape = TRUE, col = 'black',
+         col.regions = 'white', aspect = c(1.0, 0.8), colorkey = FALSE,
+         xlab = "issue age", ylab = "n", zlab = "",
+         screen = list(z = 340, x = -70),
+     scales = list(arrows = FALSE, col="black", font = 10, cex= 1.0),
+         par.settings = list(regions=list(alpha = 0.3),
+                 axis.line = list(col = "transparent")),
+         zoom = 0.95, zlim = c(0,1.00))
```

Next, the standard deviation plot.

```
> g <- expand.grid(x = seq(1, 80, 5), y = seq(20, 70, 10))
> for(i in 1:nrow(g)) {
+   endow = insurance(list(n = g$x[i], d = 1, e = 1), "isingle", "endow")
+   mort = mortassumptions(list(x = g$y[i], table = "MaleMort82"))
+   g$z[i] = z.sd(endow, mort, oumodel)
```

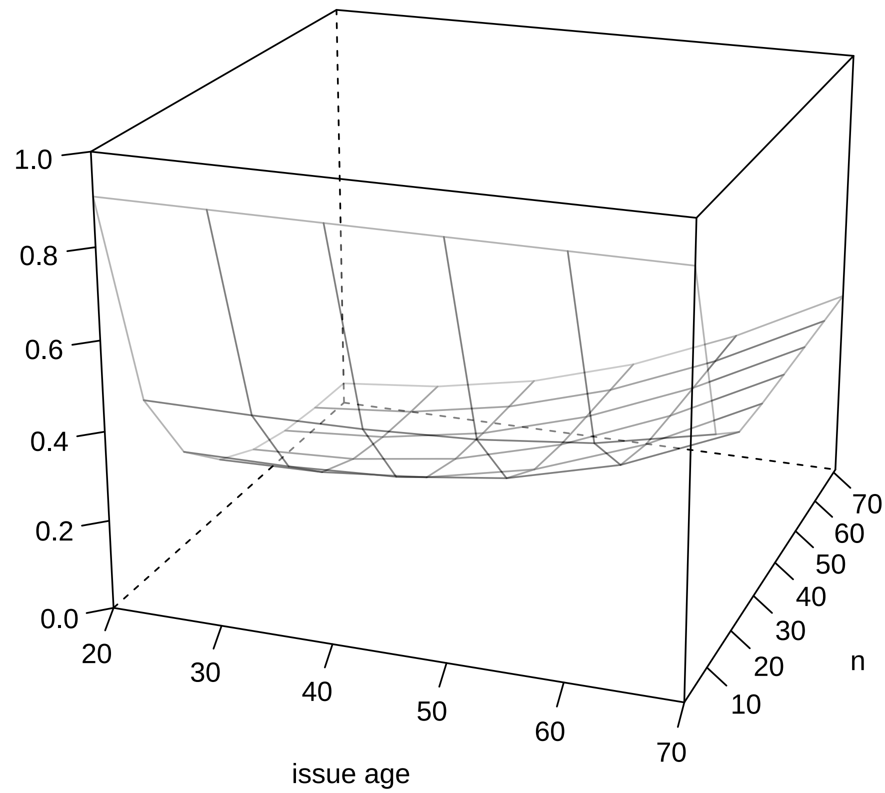

Figure 2.14: Figure 4.5 from Parker (1992)

```
+ }
> lattice::wireframe(z ~ x * y, data = g, drape = TRUE, col = 'black',
+                    col.regions = 'white', aspect = c(1, 0.8),
+                    colorkey = FALSE, xlab = "n", ylab = "issue age",
+                    zlab = "", screen = list(z = 340, x = -70, y = -20),
+                    scales = list(arrows = FALSE, col = 'black', font = 10),
+                    cex = 0.8, ylim = c(20, 70),
+                    zlim = c(0, 0.25),
+                    par.settings = list(regions=list(alpha = 0.3),
+                    axis.line = list(col = "transparent")), zoom = 0.95)
```

Next, the skewness plot.

```
> g <- expand.grid(x = seq(1, 70, 5), y = seq(20, 70, 10))
> for(i in 1:nrow(g)) {
+   endow = insurance(list(n = g$x[i], d = 1, e = 1), "isingle", "endow")
+   mort = mortassumptions(list(x = g$y[i], table = "MaleMort82"))
+   g$z[i] = z.sk(endow, mort, oumodel)
+ }
> lattice::wireframe(z ~ y * x, data = g, drape = TRUE, col = 'black',
+                    col.regions = 'white', aspect = c(1, 0.8),
```

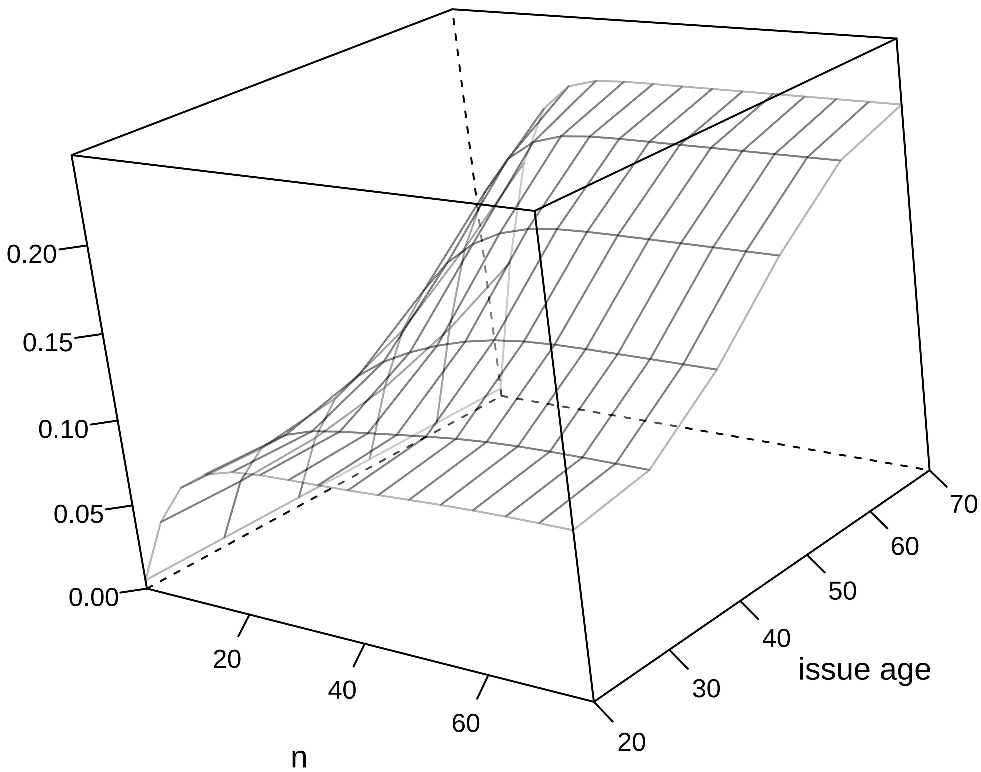

Figure 2.15: Figure 4.6 from Parker (1992)

```
+                          colorkey = FALSE, xlab = "issue age", ylab = "n",
+                          zlab = "", screen = list(z = 340, x = -70, y = -20),
+                          scales = list(arrows = FALSE, col = 'black', font = 10),
+                          cex = 0.8, zlim = c(0, 6), ylim= c(0,70),
+                          par.settings = list(regions=list(alpha = 0.3),
+                          axis.line = list(col = "transparent")), zoom = 0.95)
```

Next, the pdf plot.

```
> oumodel = iratemodel(list(delta0 = 0.1, delta = 0.06,
+                                 alpha = 0.1, sigma = 0.01), "ou")
> endow5 = insurance(list(n = 5, d = 1, e = 1), "isingle", "endow")
> mort = mortassumptions(list(x = 30, table = "MaleMort82"))
> endow25 = insurance(list(n = 25, d = 1, e = 1), "isingle", "endow")
> mort = mortassumptions(list(x = 30, table = "MaleMort82"))
> plot(function(z) z.pdf(z, endow5, mort, oumodel), 0.01, 1.0,
+      ylim = c(0, 12), lty = 1, xlab = "z", ylab = "f(z)")
```

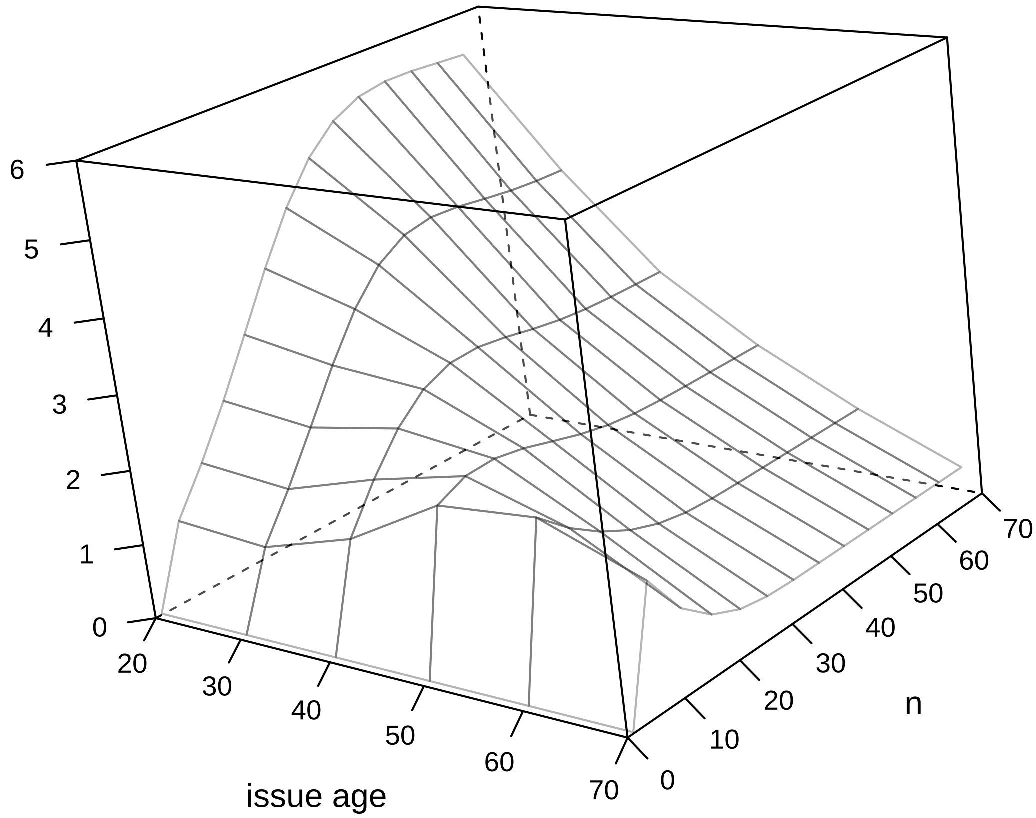

Figure 2.16: Figure 4.7 from Parker (1992)

```
> plot(function(z) z.pdf(z, endow25, mort, oumodel), 0.01, 1.0,
+      ylim = c(0, 12), add = TRUE, lty = 2)
> legend('topright', leg = c("endow 25", "endow 5"),
+        lty = c(2,1), cex = 0.8)
```

Some numerical results are reproduced below.

```
> library(stocins)
> library(xtable)
> oumodel = iratemodel(list(delta0 = 0.1, delta = 0.06,
+                           alpha = 0.1, sigma = 0.01), "ou")
> wholelife = insurance(list(n = 100, d = 1), "isingle", "term")
> x = c(20,30,40,50,60,70,80,90,100)
> first = numeric(length(x))
> sdev = numeric(length(x))
> skew = numeric(length(x))
```

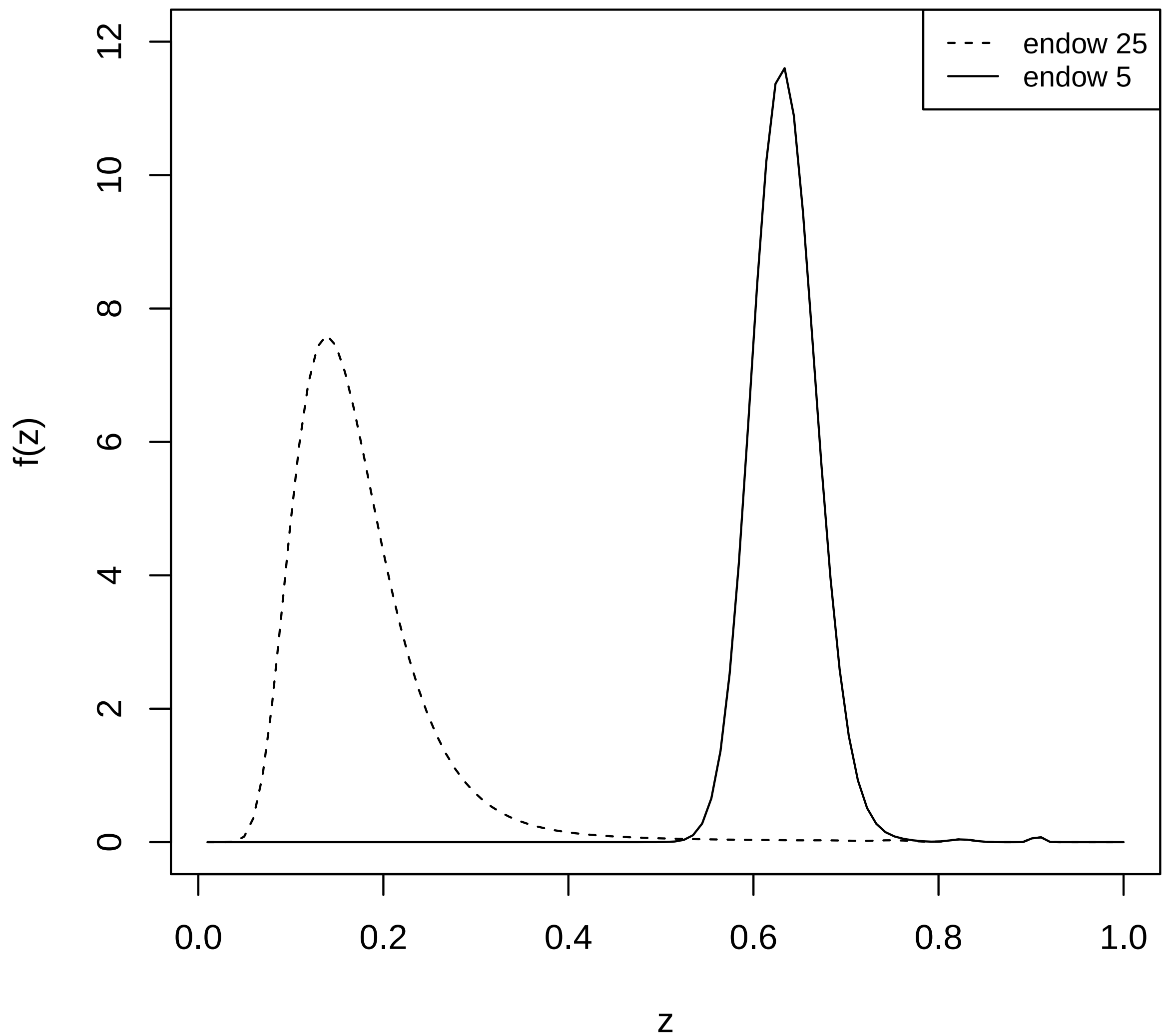

Figure 2.17: Figure 4.8 from Parker (1992)

```
> for(i in 1:length(x)) {
+   mort = mortassumptions(list(x = x[i], table = "MaleMort82"))
+   first[i] = z.moment(1, wholelife, mort, oumodel)
+   sdev[i] = z.sd(wholelife, mort, oumodel)
+   skew[i] = z.sk(wholelife, mort, oumodel)
+ }
> table41 = data.frame(Age = x, Mean = first,
+                      "Standard Deviation" = sdev,
+                      "Skewness" = skew,
+                      "Coefficient of Variation" = sdev/first)
> print(xtable(table41,
+       digits = c(0,0,6,6,6,6),
+       caption = "Table 4.1 from Parker (1992)"),
```

```
+       include.rownames = FALSE)
```

Age	Mean	Standard.Deviation	Skewness	Coefficient.of.Variation
20	0.051187	0.090805	5.411855	1.773981
30	0.076342	0.097460	3.915181	1.276631
40	0.123992	0.127706	2.632900	1.029956
50	0.199394	0.167886	1.783115	0.841979
60	0.303412	0.200298	1.100983	0.660153
70	0.432234	0.213380	0.523390	0.493667
80	0.573185	0.200033	-0.009555	0.348985
90	0.698856	0.161555	-0.388251	0.231171
100	0.883526	0.041424	-1.502273	0.046885

Table 2.1: Table 4.1 from Parker (1992)

The pdf for a whole life insurance is shown below.

```
> oumodel = iratemodel(list(delta0 = 0.1, delta = 0.06,
+                           alpha = 0.1, sigma = 0.01), "ou")
> whole = insurance(list(n = 100, d = 1, e = 1), "isingle", "term")
> mort = mortassumptions(list(x = 30, table = "MaleMort82"))
> plot(function(z) z.pdf(z, whole, mort, oumodel), 0.00001, 1.0,
+      ylim = c(0, 15), lty = 1, xlab = "z", ylab = "f(z)")
```

2.4 Portfolio of Policies

Some results for a portfolio of policies are reproduced below.

```
> oumodel = iratemodel(list(delta0 = 0.1, delta = 0.06,
+                           alpha = 0.1, sigma = 0.01), "ou")
> mort = mortassumptions(list(x = 30, table = "MaleMort82"))
> n = seq(1,70,1)
> sdev1 = numeric(length(n))
> sdev10 = numeric(length(n))
> sdev100 = numeric(length(n))
> sdev1000 = numeric(length(n))
> sdevInf = numeric(length(n))
> for(i in 1:length(n))
+ {
+   term = insurance(list(n=n[i], d=1), "isingle", "term")
+   port1 = insurance(list(single = term, c = 1), "iport", "term")
+   port10 = insurance(list(single = term, c = 10), "iport", "term")
+   port100 = insurance(list(single = term, c = 100), "iport", "term")
+   port1000 = insurance(list(single = term, c = 1000), "iport", "term")
+   portInf = insurance(list(single = term, c = 1e18), "iport", "term")
```

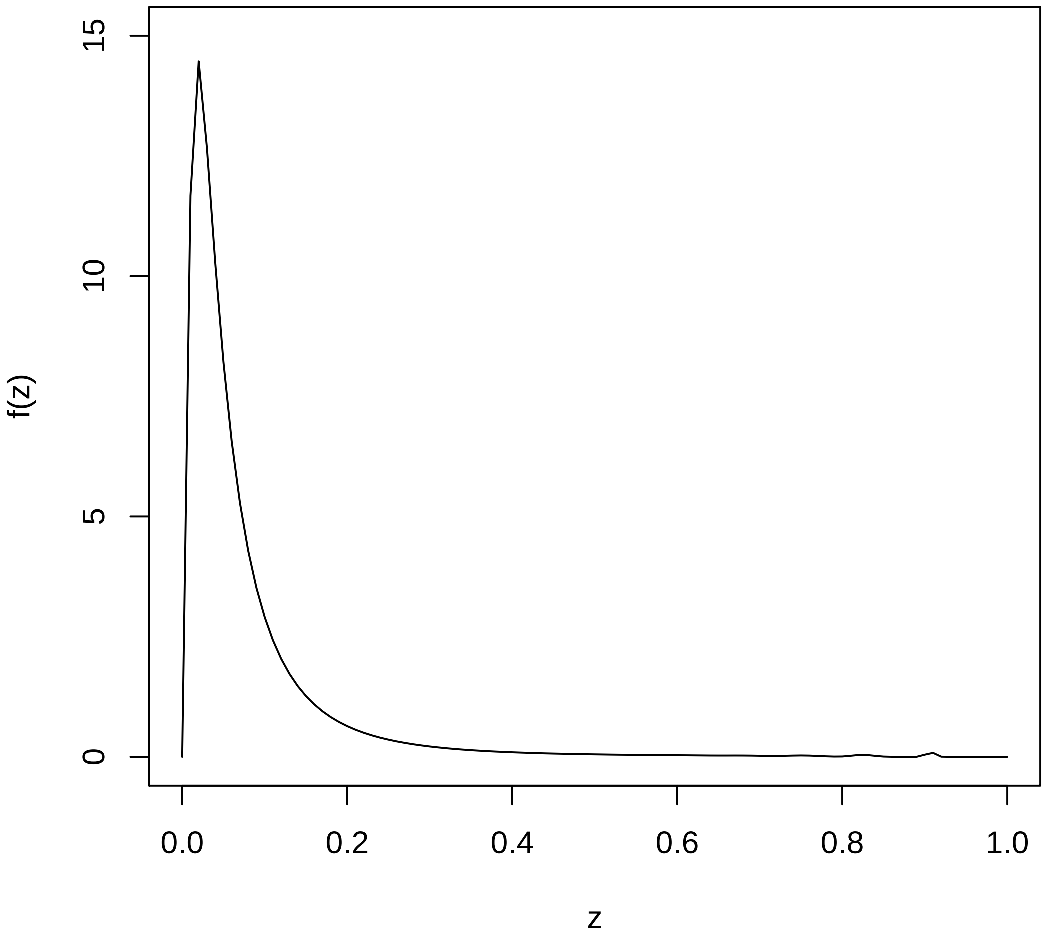

Figure 2.18: Figure 4.9 from Parker (1992)

```
+
+   sdev1[i] = z.sd(port1,mort,oumodel)
+   sdev10[i] = z.sd(port10,mort,oumodel)
+   sdev100[i] = z.sd(port100,mort,oumodel)
+   sdev1000[i] = z.sd(port1000,mort,oumodel)
+   sdevInf[i] = z.sd(portInf,mort,oumodel)
+ }
> plot(x = n, y = sdev1, type = 'l', ylab = "sd", xlab = "n",
+      ylim = c(0, 0.15))
> lines(x = n, y = sdev10/10, type = 'l', lty = 2)
> lines(x = n, y = sdev100/100, type = 'l', lty = 3)
> lines(x = n, y = sdev1000/1000, type = 'l', lty = 4)
> lines(x = n, y = sdevInf/1e18, type = 'l', lty = 5)
> legend('topright', leg = c("c=1","c=10","c=100","c=1000","c=Inf"),
+         lty = c(1,2,3,4,5), ncol = 5, cex = 0.9)
```

The standard deviation for an endowment policy is below.

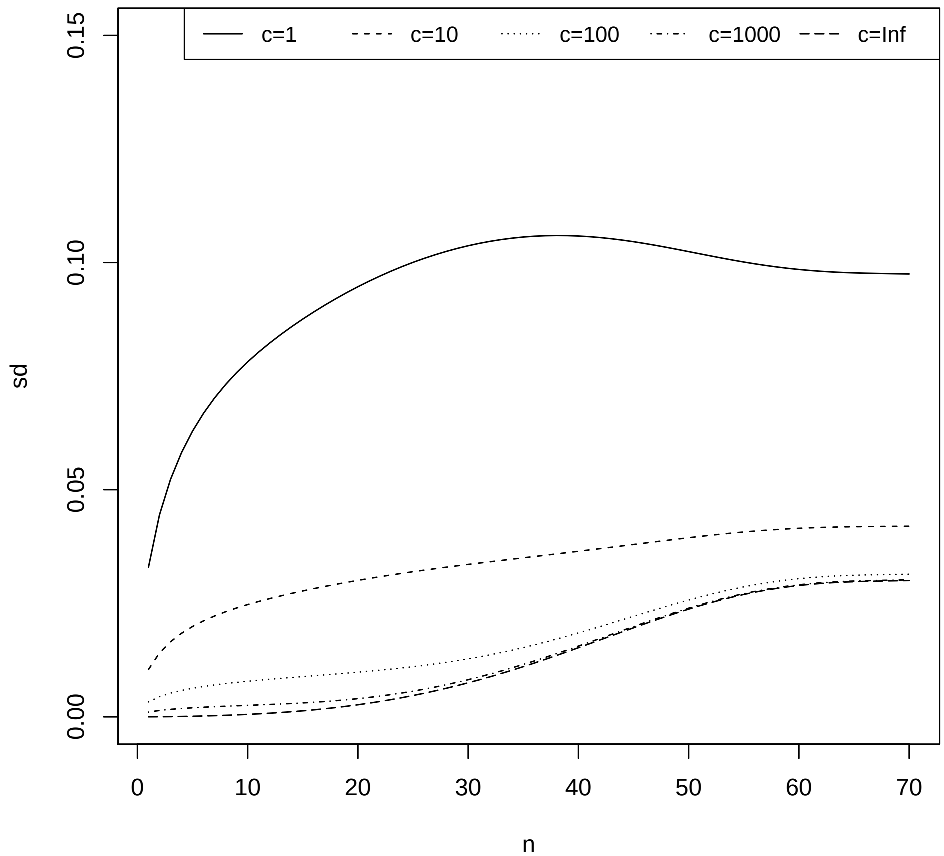

Figure 2.19: Figure 5.1 from Parker (1992)

```
> oumodel = iratemodel(list(delta0 = 0.1, delta = 0.06,
+                           alpha = 0.1, sigma = 0.01), "ou")
> mort = mortassumptions(list(x = 30, table = "MaleMort82"))
> n = seq(1,70,1)
> sdev1 = numeric(length(n))
> sdev10 = numeric(length(n))
> sdev100 = numeric(length(n))
> sdev1000 = numeric(length(n))
> sdevInf = numeric(length(n))
> for(i in 1:length(n))
+ {
+   endow = insurance(list(n=n[i], d=1, e=1), "isingle", "endow")
+   port1 = insurance(list(single = endow, c = 1), "iport", "endow")
+   port10 = insurance(list(single = endow, c = 10), "iport", "endow")
+   port100 = insurance(list(single = endow, c = 100), "iport", "endow")
+   port1000 = insurance(list(single = endow, c = 1000), "iport", "endow")
+   portInf = insurance(list(single = endow, c = 1e18), "iport", "endow")
+
```

```
+   sdev1[i] = z.sd(port1,mort,oumodel)
+   sdev10[i] = z.sd(port10,mort,oumodel)
+   sdev100[i] = z.sd(port100,mort,oumodel)
+   sdev1000[i] = z.sd(port1000,mort,oumodel)
+   sdevInf[i] = z.sd(portInf,mort,oumodel)
+ }
> plot(x = n, y = sdev1, type = 'l', ylab = "sd", xlab = "n",
+      ylim = c(0, 0.15))
> lines(x = n, y = sdev10/10, type = 'l', lty = 2)
> lines(x = n, y = sdev100/100, type = 'l', lty = 3)
> lines(x = n, y = sdev1000/1000, type = 'l', lty = 4)
> lines(x = n, y = sdevInf/1e18, type = 'l', lty = 5)
> legend('topright', leg = c("c=1","c=10","c=100","c=1000","c=Inf"),
+         lty = c(1,2,3,4,5), ncol = 5, cex = 0.9)
```

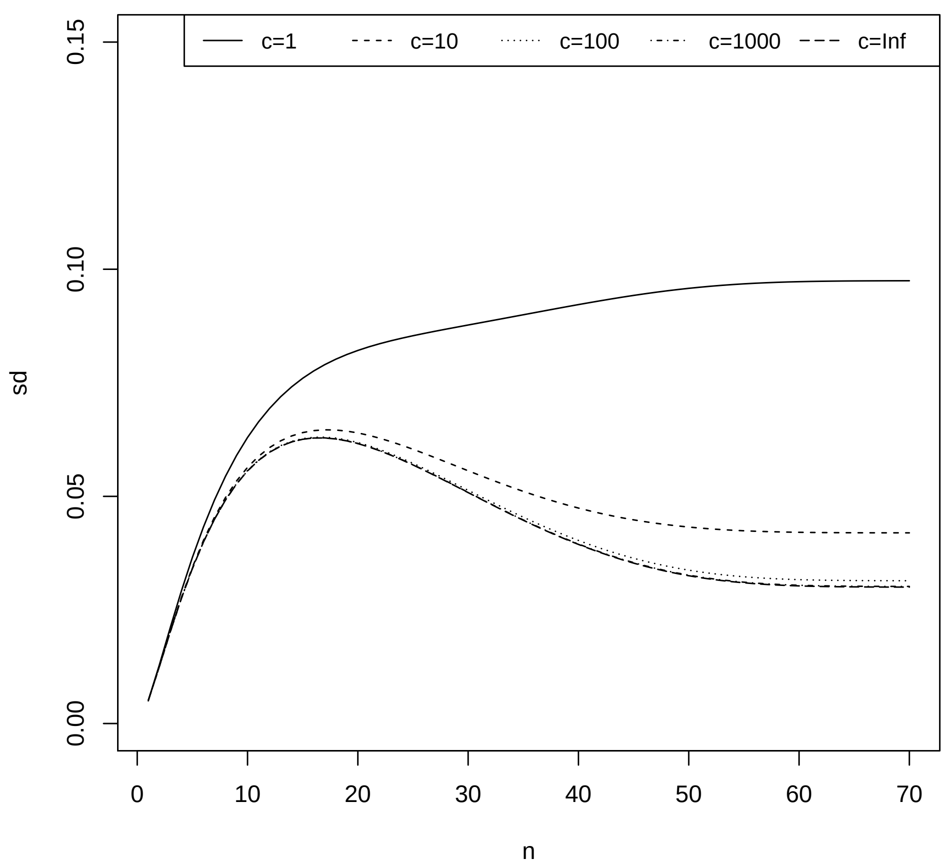

Figure 2.20: Figure 5.3 from Parker (1992)

Chapter 3

Reproducing Parker (1997) Results

Some results from Table 6 of (Parker, 1997) are reproduced here.

```
> oumodel = iratemodel(list(delta0 = 0.08, delta = 0.06,
+                          alpha = 0.1, sigma = 0.01), "ou")
```

3.1 Insurance and Investment Risk

```
> s1 = insurance(list(n=10, d=50,  e=50), "isingle", "endow")
> s2 = insurance(list(n=5,  d=100, e=50), "isingle", "endow")
> s3 = insurance(list(n=10, d=150, e=0),  "isingle", "endow")
> s4 = insurance(list(n=10, d=50,  e=0),  "isingle", "endow")
> s5 = insurance(list(n=10, d=100, e=100),"isingle", "endow")
> s6 = insurance(list(n=5,  d=75,  e=0),  "isingle", "endow")
> s7 = insurance(list(n=5,  d=25,  e=0),  "isingle", "endow")
> s8 = insurance(list(n=10, d=50,  e=50), "isingle", "endow")
> m1 = mortassumptions(list(x = 30, table = "MaleMort82"))
> m2 = mortassumptions(list(x = 35, table = "MaleMort82"))
> m3 = mortassumptions(list(x = 50, table = "MaleMort82"))
> m4 = mortassumptions(list(x = 30, table = "FemaleMort82"))
> m5 = mortassumptions(list(x = 40, table = "FemaleMort82"))
> m6 = mortassumptions(list(x = 40, table = "MaleMort82Reduced"))
> m7 = mortassumptions(list(x = 45, table = "FemaleMort82Reduced"))
> m8 = mortassumptions(list(x = 55, table = "FemaleMort82"))
> p1 = insurance(list(single = s1, c = 1000), "iport", "endow")
> p2 = insurance(list(single = s2, c = 2500), "iport", "endow")
> p3 = insurance(list(single = s3, c = 2000), "iport", "endow")
> p4 = insurance(list(single = s4, c = 1500), "iport", "endow")
> p5 = insurance(list(single = s5, c = 500), "iport", "endow")
> p6 = insurance(list(single = s6, c = 2500), "iport", "endow")
```

```
> p7 = insurance(list(single = s7, c = 3000), "iport", "endow")
> p8 = insurance(list(single = s8, c = 500), "iport", "endow")
> insgroup = insurance(list(p1, p2, p3, p4, p5, p6, p7, p8),
+                      "igroup")
> mortgroup = list(m1, m2, m3, m4, m5, m6, m7, m8)
> z.moment(1, insgroup, mortgroup, oumodel) / insgroup$c

[1] 12.64324

> z.moment(2, insgroup, mortgroup, oumodel) / insgroup$c^2

[1] 160.8296

> z.sd(insgroup, mortgroup, oumodel) / insgroup$c

[1] 0.9890085

> z.invrisk(insgroup, mortgroup, oumodel) / insgroup$c^2

[1] 0.9675637

> z.insrisk(insgroup, mortgroup, oumodel) / insgroup$c^2

[1] 0.01057409
```

Chapter 4

Other Demonstrations

Some other functionality of the `stocins` package is shown here.

4.1 Survival Models

```
> mort = mortassumptions(list(x = 40, table = "MaleMort82"))
> plot(x = seq(0, 70, 1), y = kpx(seq(0, 70, 1), mort), xlab = "k",
+      ylab = "kpx", type = 'l')
```

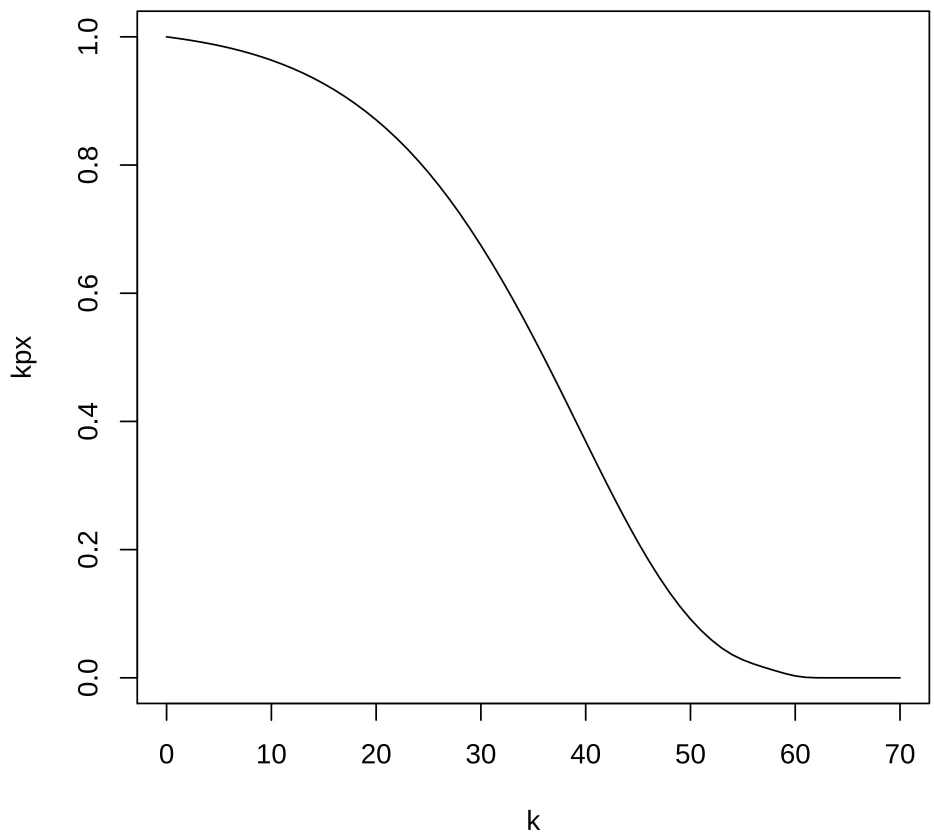

Figure 4.1: Example of a survival probability plot created using the `stocins` package

References

Parker, G. (1992). *An application of stochastic interest rate models in life assurance* (Unpublished doctoral dissertation). Heriot-Watt University.

Parker, G. (1997). Stochastic analysis of the interaction between investment and insurance risks. *North American actuarial journal*, *1*(2), 55–71.