Commuter Mode Data

```
> module(discreteChoice)

> mdc.commuter[1:8,]
  id alt choice     cost     time c.Carpool c.Bus c.Rail
1  1   1      1 1.507010 18.50320         0     0      0
2  1   2      1 2.335612 26.33823         1     0      0
3  1   3      1 1.800512 20.86779         0     1      0
4  1   4      1 2.358920 30.03347         0     0      1
5  2   1      4 6.056999 31.31111         0     0      0
6  2   2      4 2.896919 34.25696         1     0      0
7  2   3      4 2.237129 67.18189         0     1      0
8  2   4      4 1.855451 60.29313         0     0      1
```

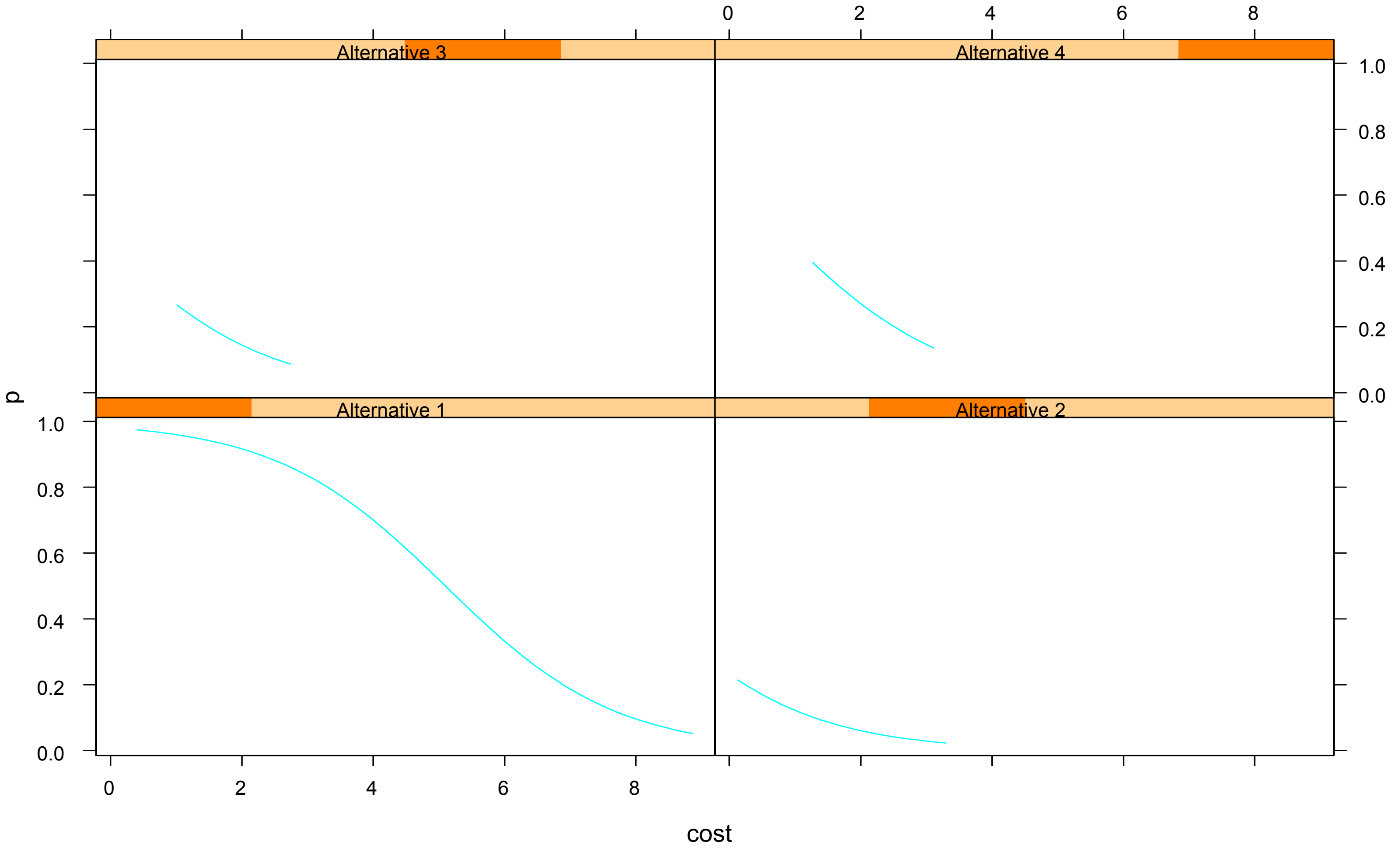
Logit Model Fit

```
> summary(fit.logit,cor=F)
Call:
 mdc.logit(formula = choice ~ cost + I(time/10) + c.Carpool
+ c.Bus + c.Rail, data = mdc.commuter)
Results:
              estimate std.error   z.stat p.value
        cost   -0.7723    0.0919  -8.4033       0
  I(time/10)   -0.8536    0.0775 -11.0154       0
   c.Carpool   -4.1976    0.3926 -10.6928       0
       c.Bus   -3.2925    0.3171 -10.3842       0
      c.Rail   -2.6647    0.2886  -9.2339       0

Actual counts:
   1  2  3   4
 218 32 81 122
```

Logit Model Fit

```
Shares:
           alt.1  alt.2  alt.3  alt.4
   actual 0.4812 0.0706 0.1788 0.2693
predicted 0.4812 0.0706 0.1788 0.2693

Percent correctly predicted: 68.4327

Likelihood ratio index: 0.4356
Number of observations: 453
Iterations: 24 f.evals: 25 g.evals: 25
Message: RELATIVE FUNCTION CONVERGENCE
Mean log-likelihood: -0.7825
```

Own Marginal Effects



Predictions

```
> predict.logit = predict(fit.logit,newdata=mdc.commuter)
> predict.logit[1,]
  id     alt.1       alt.2      alt.3      alt.4
1  1 0.9599263 0.003898084 0.02323984 0.01293574
# raise cost of car by 50% and forecast new probabilities
> mdc.commuter.new = mdc.commuter
> mdc.commuter.new[mdc.commuter.new$alt == 1,"cost"] =
+   mdc.commuter.new[mdc.commuter.new$alt == 1,"cost"]*1.5
> predict.logit.new =
predict(fit.logit,newdata=mdc.commuter.new)
> predict.logit.new[1,]
  id     alt.1       alt.2      alt.3      alt.4
1  1 0.9304854 0.006761891 0.04031347 0.02243926
> (predict.logit.new[1,] -
predict.logit[1,])/predict.logit[1,]
  id       alt.1     alt.2     alt.3     alt.4
1  0 -0.03067001 0.7346707 0.7346707 0.7346707
```

Probit Model Fit

```
> logit.coef = coef(fit.logit)
> start.coef = c(logit.coef,1,1,1,1,1)
> fit.probit = mdc.probit(choice ~ cost + I(time/10)
+              c.Carpool + c.Bus + c.Rail,
+              data = mdc.commuter,
+              init.par = start.coef,
+              control = mdc.probit.control(nrep=125,
+                        err.fn = "mdc.err.uniform"))
> dos.time(mdc.probit(…))
  6.081083 # minutes
```

Probit Model Fit

```
> summary(fit.probit,cor=F)
Results:
              estimate std.error  z.stat p.value
        cost   -0.2860    0.0647 -4.4171  0.0000
  I(time/10)   -0.3228    0.0928 -3.4770  0.0005
   c.Carpool   -2.0870    0.2474 -8.4365  0.0000
       c.Bus   -1.2571    0.3083 -4.0774  0.0000
      c.Rail   -1.0504    0.2555 -4.1111  0.0000
          k1    0.2907    0.2056  1.4140  0.1574
          k2    0.5926    0.2057  2.8810  0.0040
          k3   -0.2317    0.2724 -0.8507  0.3950
          k4    0.3272    0.1635  2.0011  0.0454
          k5    0.5178    0.2187  2.3673  0.0179

Actual counts:
   1  2  3   4
 218 32 81 122
```

Probit Model Fit

```
Estimated covariance matrix of errors:
  1       2      3       4
1 0  0.0000 0.0000  0.0000
2 0  1.0000 0.2907 -0.2317
3 0  0.2907 0.4356  0.1265
4 0 -0.2317 0.1265  0.4288

Shares:
           alt.1  alt.2  alt.3  alt.4
   actual 0.4812 0.0706 0.1788 0.2693
predicted 0.4832 0.0698 0.1760 0.2692

Percent correctly predicted: 67.1082

Likelihood ratio index: 0.517
Number of observations: 453
Iterations: 49 f.evals: 53 g.evals: 584
Message: RELATIVE FUNCTION CONVERGENCE
Mean log-likelihood: -0.7718
```

Predictions

```
> predict.probit =
predict(fit.probit,newdata=mdc.commuter)
> predict.probit[1,]
  id     alt.1       alt.2      alt.3       alt.4
1  1 0.9752639 0.004589405 0.01519994 0.005271149

# raise cost of car by 50% and forecast new shares
> predict.probit.new =
predict(fit.probit,newdata=mdc.commuter.new)
> predict.probit.new[1,]
  id     alt.1       alt.2      alt.3      alt.4
1  1 0.9425927 0.008004964 0.03239861 0.01230512

> (predict.probit.new[1,] -
predict.probit[1,])/predict.probit[1,]
  id       alt.1     alt.2    alt.3    alt.4
1  0 -0.03349988 0.7442267 1.131496 1.334429
```

Mixed Logit Model for Commuter Data: Alternative Specific Random Effects

```
> start.vals = c(-0.7723,-0.8536,
+ -4.1976,0.1,-3.2925,0.1,-2.6647,0.1)

> fit.mlogit = mdc.logit(choice ~ cost +
I(time/10) + mdc.normal(c.Carpool)+
+ mdc.normal(c.Bus) + mdc.normal(c.Rail),
+ data = mdc.commuter,
+ init.par = start.vals,
+ control=mdc.logit.control(nrep=1000,
+ err.fn="mdc.err.uniform"))
```

Mixed Logit Model Fit

```
Results:
                             estimate std.error  z.stat p.value
                       cost   -1.1715    0.3887 -3.0138  0.0026
                  I(time/10)  -1.3031    0.4849 -2.6874  0.0072
 mdc.normal(c.Carpool) mean   -8.3918    3.1429 -2.6701  0.0076
mdc.normal(c.Carpool) scale   -3.6655    1.8106 -2.0245  0.0429
     mdc.normal(c.Bus) mean   -5.1471    1.9629 -2.6222  0.0087
    mdc.normal(c.Bus) scale    1.7339    1.4081  1.2313  0.2182
    mdc.normal(c.Rail) mean   -4.1620    1.5363 -2.7091  0.0067
   mdc.normal(c.Rail) scale    1.8538    1.3206  1.4038  0.1604

Actual counts:
   1  2  3   4
 218 32 81 122
```

Note: negative
scale estimate
is possible

Mixed Logit Model Fit

```
Shares:
           alt.1  alt.2  alt.3  alt.4
   actual 0.4812 0.0706 0.1788 0.2693
predicted 0.4811 0.0705 0.1779 0.2704

Percent correctly predicted: 67.9912

Likelihood ratio index: 0.4424
Number of observations: 453
Iterations: 57 f.evals: 59 g.evals: 58
Message: RELATIVE FUNCTION CONVERGENCE
Mean log-likelihood: -0.773
```

Electricity Supplier Data

```
> mdc.electric[1:48,1:6]
   id time choice alt kWhprice length
 1  1    1      4   1        7      5
 2  1    1      4   2        9      1
 3  1    1      4   3        0      0
 4  1    1      4   4        0      5
 5  1    2      3   1        7      0
 6  1    2      3   2        9      5
 7  1    2      3   3        0      1
 8  1    2      3   4        0      5

…
45  1   12      4   1        0      1
46  1   12      4   2        7      5
47  1   12      4   3        9      5
48  1   12      4   4        0      0
```

Logit Model Fit

```
> fit.logit.panel = mdc.logit(choice ~ kWhprice +
length + local + wellknown + TOD + seasonal,
+ data = mdc.electric,
+ panel = T, time.column = "time")
> fit.logit.panel
Call:
 mdc.logit(formula = choice ~ kWhprice + length +
local + wellknown + TOD + seasonal,
data = mdc.electric, time.column = "time", panel = T)
Estimated values:
        var  type   tag   value
1  kWhprice fixed value -0.6252
2    length fixed value -0.1083
3     local fixed value  1.4423
4 wellknown fixed value  0.9955
5       TOD fixed value -5.4628
6  seasonal fixed value -5.8400
```

Mixed Logit Specification

```
> start.vals = c(-0.6252, -0.1083, 0.1,
+ 1.4423,0.1,0.9955, 0.1, -5.4628, 0.1,
+ -5.8400, 0.1)

> fit.mlogit.panel = mdc.logit(choice ~
+ kWhprice + mdc.normal(length) +
+ mdc.normal(local) +
+ mdc.normal(wellknown) + mdc.normal(TOD)
+ mdc.normal(seasonal),
+ data = mdc.electric,
+ init.par = start.vals,
+ panel = T, time.column = "time",
+ control = mdc.logit.control(nrep=1000,
+           err.fn="mdc.err.uniform",
+           seed=123))
```

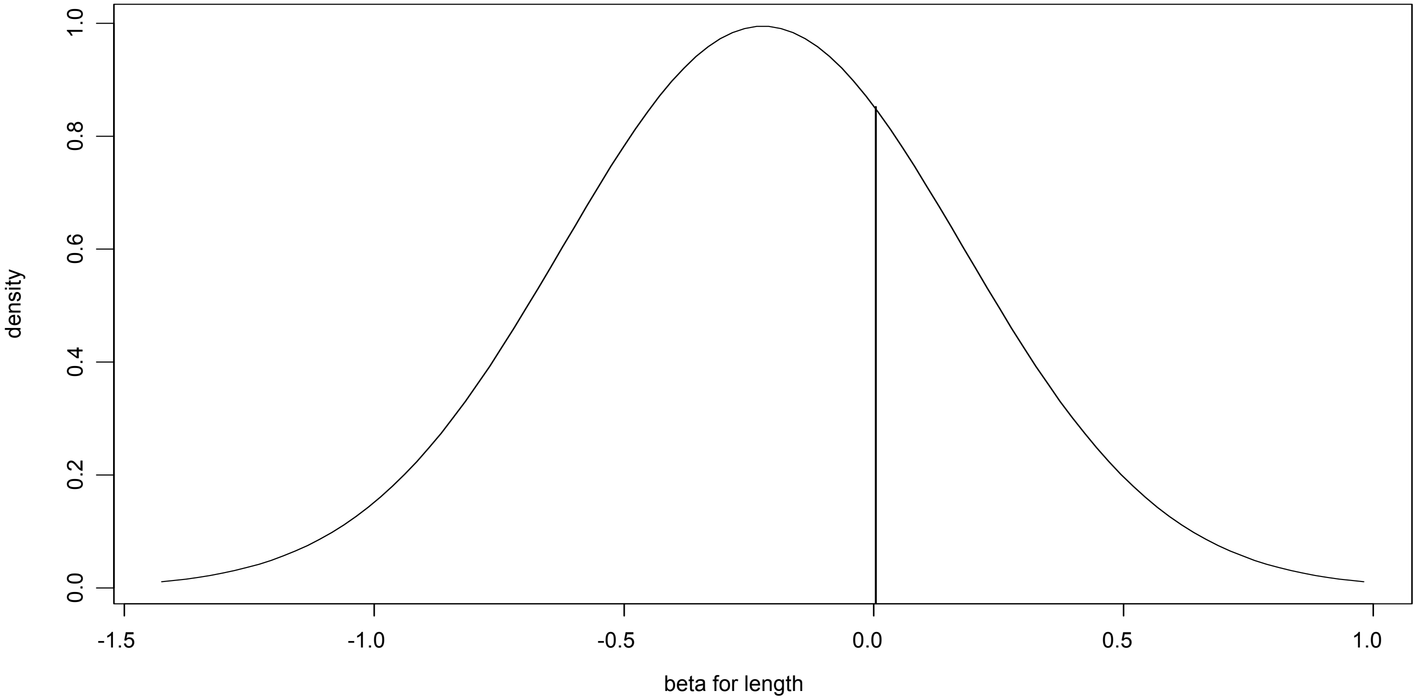
Mixed Logit Model Fit

```
Results:
                            estimate std.error   z.stat p.value
                  kWhprice   -0.9288    0.0346 -26.8445       0
    mdc.normal(length) mean  -0.2222    0.0249  -8.9227       0
   mdc.normal(length) scale   0.4010    0.0258  15.5400       0
     mdc.normal(local) mean   2.2186    0.1227  18.0859       0
    mdc.normal(local) scale   1.7445    0.1185  14.7264       0
 mdc.normal(wellknown) mean   1.6569    0.0929  17.8318       0
mdc.normal(wellknown) scale   1.1732    0.0938  12.5063       0
       mdc.normal(TOD) mean  -9.0251    0.3364 -26.8291       0
      mdc.normal(TOD) scale   2.9664    0.1899  15.6236       0
  mdc.normal(seasonal) mean  -9.3176    0.3296 -28.2661       0
 mdc.normal(seasonal) scale   2.1613    0.1510  14.3099       0

> dos.time(mdc.logit(…)) = 9.733  # minutes
```

Mixed Logit Model Fit

```
Actual counts:
   1    2    3    4
 978 1137 1026 1167
Shares:
           alt.1  alt.2  alt.3  alt.4
   actual 0.2270 0.2639 0.2382 0.2709
predicted 0.2332 0.2604 0.2331 0.2733
Percent correctly predicted: 47.3073

Likelihood ratio index: 0.3442
Number of observations: 4308
Iterations: 68 f.evals: 71 g.evals: 69
Message: RELATIVE FUNCTION CONVERGENCE
Mean log-likelihood: -0.9091
```

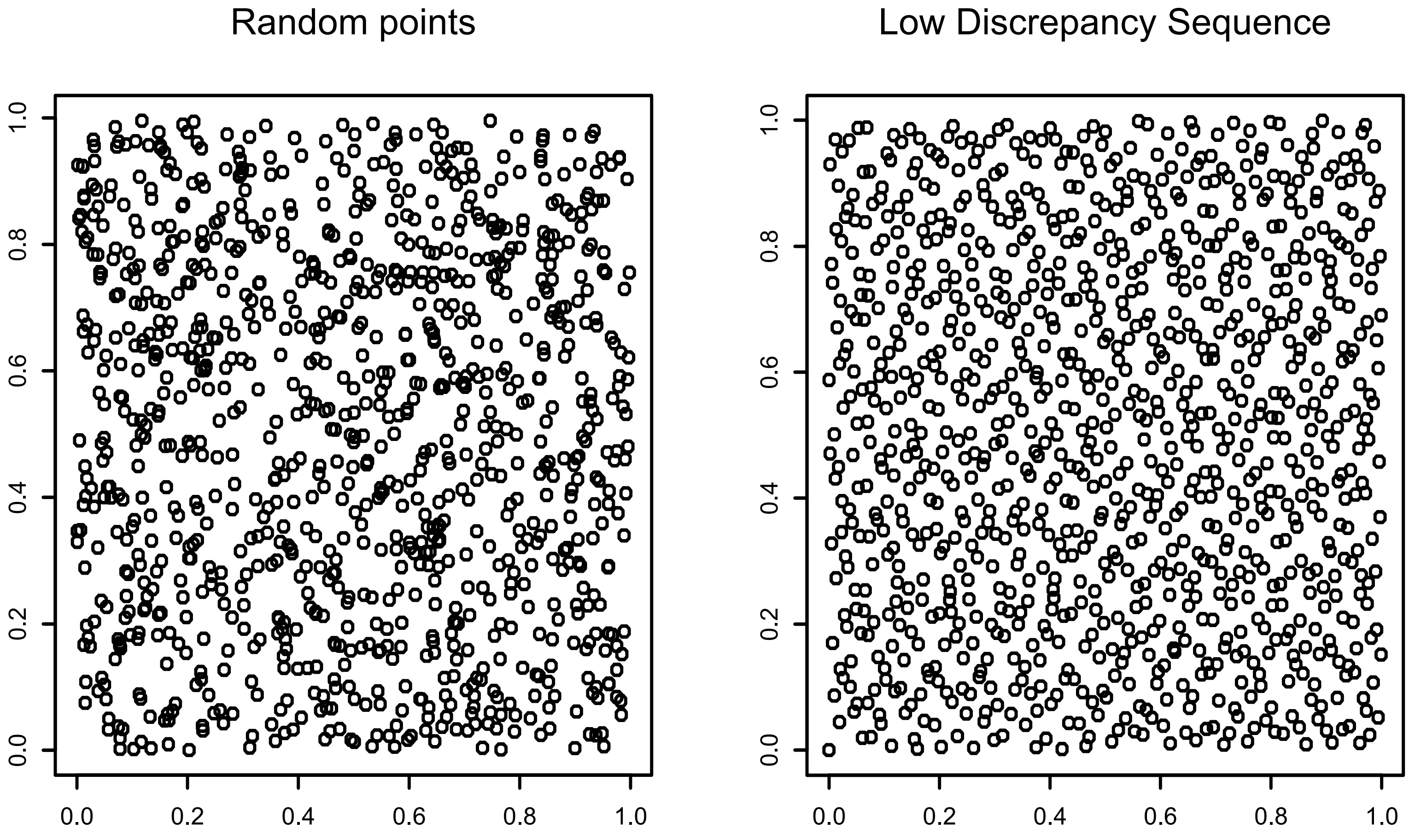
Density for Length of Contract



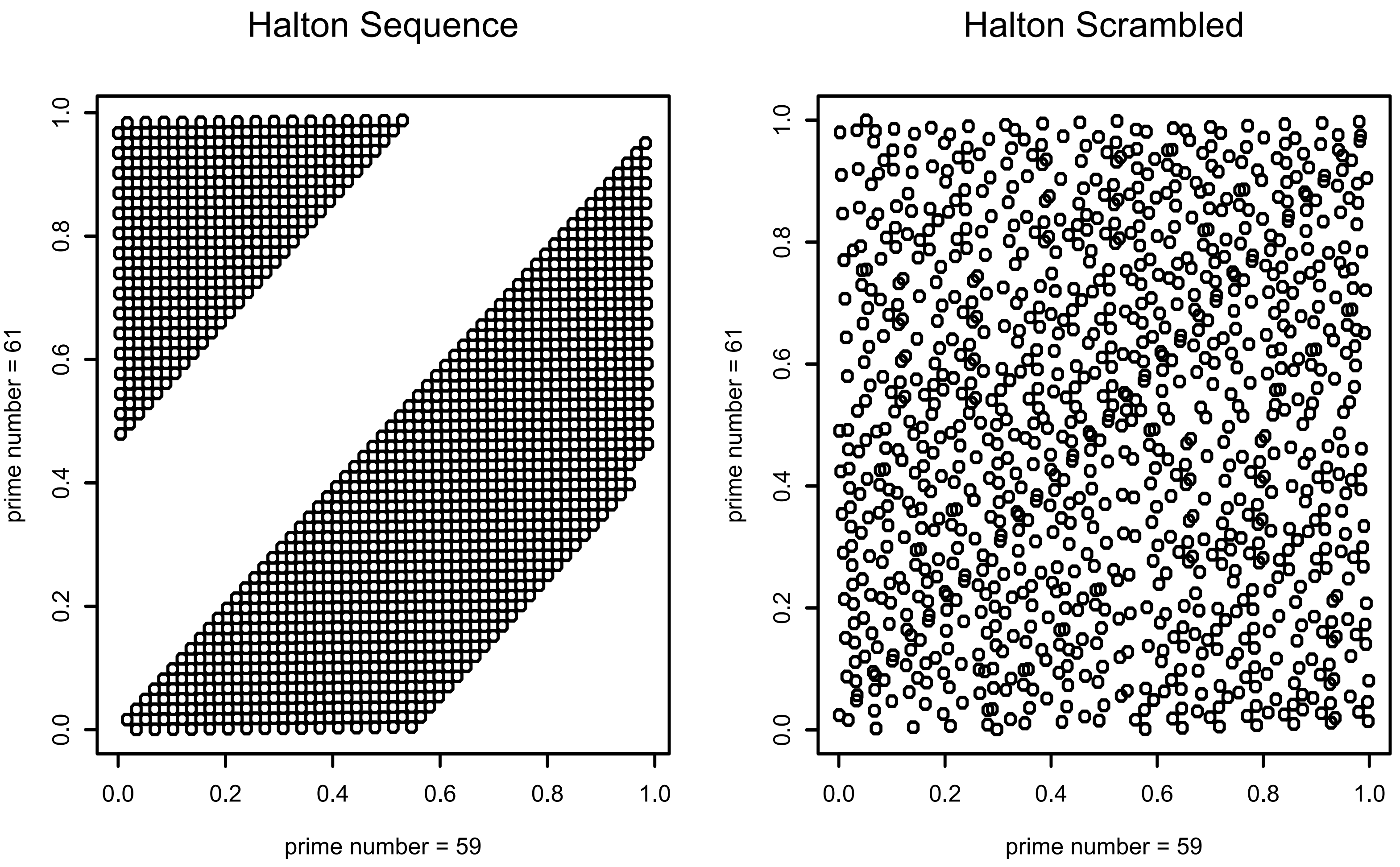
$$\beta_{length} \sim N(-0.22,(0.40)^2)$$

$$\Pr(\beta_{length} \leq 0) = 0.71$$

Random Points vs LDS



Halton vs. Scrambled Halton



Mixed Logit with LDS

```
> fit.mlogit.panel = mdc.logit(choice ~
+ kWhprice + mdc.normal(length) +
+ mdc.normal(local) +
+ mdc.normal(wellknown) + mdc.normal(TOD)
+ mdc.normal(seasonal),
+ data = mdc.electric,
+ init.par = start.vals,
+ panel = T, time.column = "time",
+ control = mdc.logit.control(nrep=1000,
+           err.fn="mdc.err.halton"))
```

Option to specify LDS for simulating choice probabilities

Mixed Logit Fit Using 125 Halton Draws

```
Results:
                            estimate std.error   z.stat p.value
                  kWhprice   -0.8825    0.0323 -27.3243       0
    mdc.normal(length) mean  -0.2080    0.0220  -9.4728       0
   mdc.normal(length) scale   0.3849    0.0229  16.8129       0
     mdc.normal(local) mean   2.0639    0.1050  19.6510       0
    mdc.normal(local) scale   1.5971    0.1048  15.2450       0
 mdc.normal(wellknown) mean   1.5178    0.0782  19.4216       0
mdc.normal(wellknown) scale   0.9209    0.0838  10.9927       0
       mdc.normal(TOD) mean  -8.3714    0.3033 -27.6020       0
      mdc.normal(TOD) scale   2.8220    0.1746  16.1600       0
  mdc.normal(seasonal) mean  -8.7452    0.2951 -29.6314       0
 mdc.normal(seasonal) scale   1.9830    0.1241  15.9792       0

> dos.time(mdc.logit(…)) = 1.184   # minutes

# compare to 9.733 minutes for 1000 random draws
```

Probit Fit with LDS

```
> fit.probit = mdc.probit(choice ~ cost +
+ I(time/10) + c.Carpool + c.Bus + c.Rail,
+ data = mdc.commuter,
+ init.par = start.coef,
+ control = mdc.probit.control(nrep=125,
+           err.fn = "mdc.err.niedx"))
```

Specify LDS for simulating choice probability