# 11 Survival Analysis

By: Udit (based on ISLR)

## Setup

Using **Survival** library.
Using **BrainCancer** dataset.
Function **Surv()** for creating a survival object, and **survfit() (both fitting & predicting) and coxph()** for model fitting.

```
library(survival)
library(ISLR2)
names(BrainCancer)
```

```
## [1] "sex"       "diagnosis" "loc"       "ki"        "gtv"       "stereo"
## [7] "status"    "time"
```

```
dim(BrainCancer)
```

```
## [1] 88  8
```

```
attach(BrainCancer)
table(sex)
```

```
## sex
## Female   Male
##     45     43
```

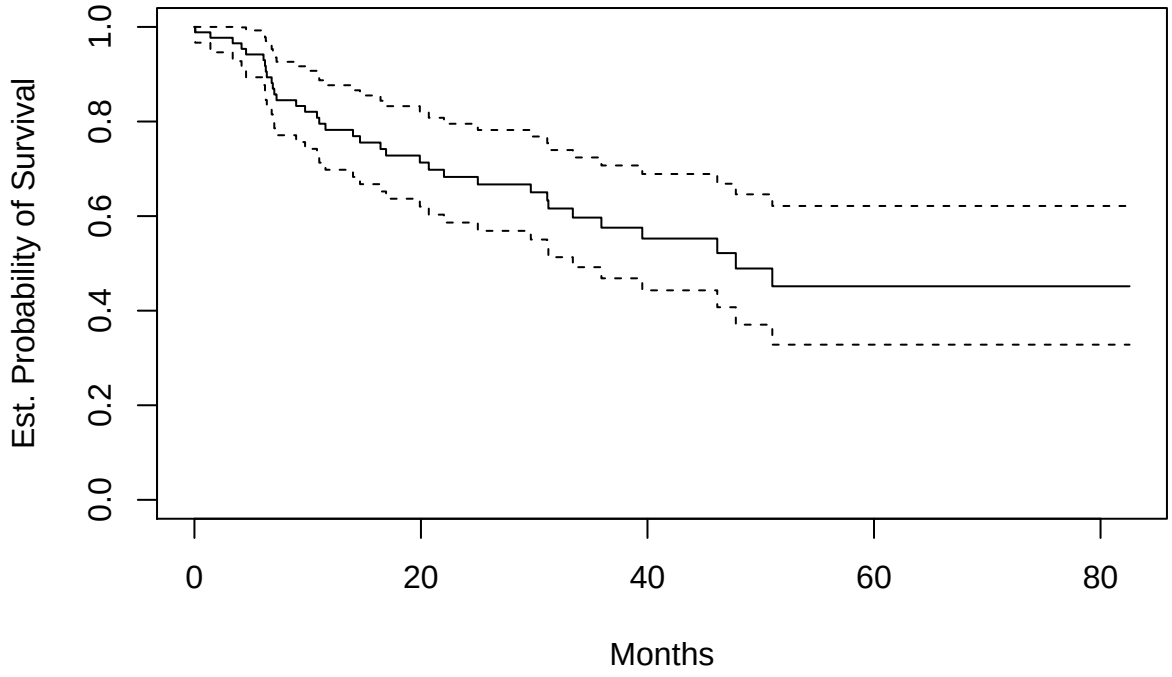
```
table(status)
```

```
## status
##  0  1
## 53 35
```

## Kaplan-Meier Survival Curves

Status = 1 indicates an uncensored observation, and status = 0 indicates a censored observation.

```
fit.surv = survfit(Surv(time, status)~1)
plot(fit.surv, xlab="Months", ylab="Est. Probability of Survival")
```

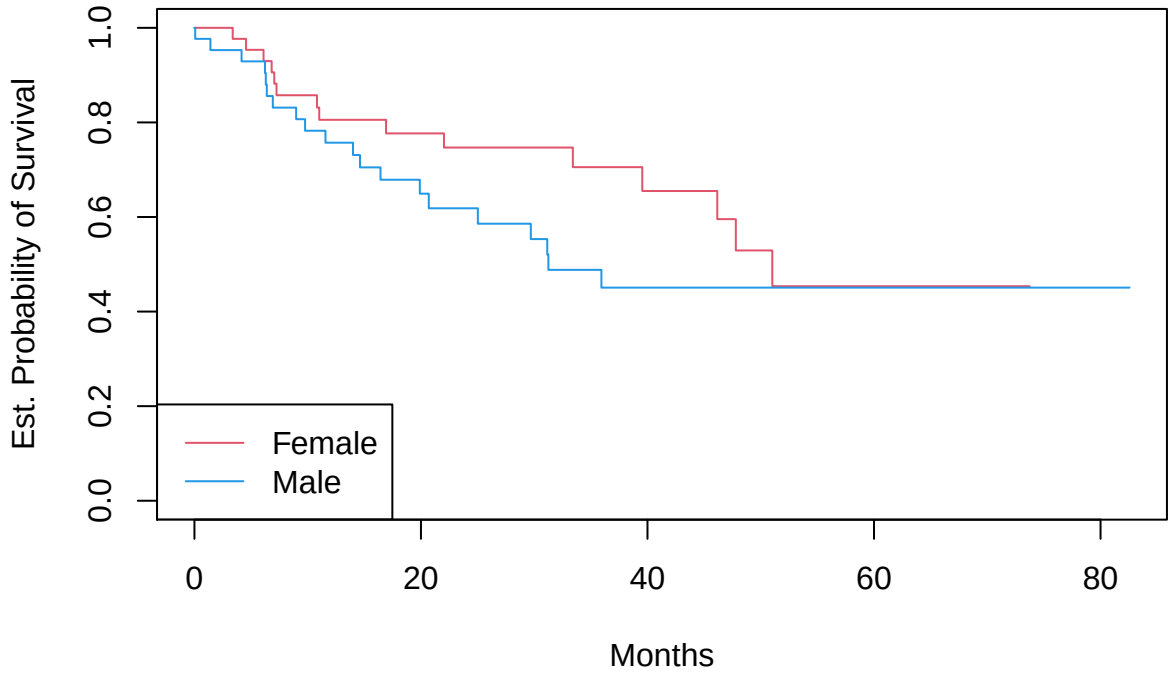



```
summary(fit.surv)
```

```
## Call: survfit(formula = Surv(time, status) ~ 1)
##
##   time n.risk n.event survival std.err lower 95% CI upper 95% CI
##   0.07     88       1    0.989  0.0113        0.967        1.000
##   1.41     86       1    0.977  0.0160        0.946        1.000
##   3.38     83       1    0.965  0.0197        0.928        1.000
##   4.16     82       1    0.954  0.0227        0.910        0.999
##   4.56     81       1    0.942  0.0253        0.894        0.993
##   6.10     78       1    0.930  0.0277        0.877        0.986
##   6.23     77       1    0.918  0.0298        0.861        0.978
##   6.30     76       1    0.906  0.0318        0.845        0.970
##   6.39     75       1    0.894  0.0336        0.830        0.962
##   6.82     74       1    0.881  0.0352        0.815        0.953
##   6.92     73       1    0.869  0.0368        0.800        0.944
##   7.05     72       1    0.857  0.0382        0.786        0.935
##   7.25     70       1    0.845  0.0395        0.771        0.926
##   8.98     69       1    0.833  0.0408        0.757        0.917
##   9.77     68       1    0.821  0.0420        0.742        0.907
##  10.82     65       1    0.808  0.0432        0.727        0.897
##  11.02     64       1    0.795  0.0444        0.713        0.887
##  11.57     61       1    0.782  0.0455        0.698        0.877
##  14.00     59       1    0.769  0.0466        0.683        0.866
##  14.62     57       1    0.756  0.0477        0.668        0.855
##  16.43     55       1    0.742  0.0488        0.652        0.844
##  16.92     54       1    0.728  0.0498        0.637        0.832
##  19.90     49       1    0.713  0.0509        0.620        0.820
##  20.69     47       1    0.698  0.0521        0.603        0.808
##  22.03     46       1    0.683  0.0531        0.586        0.795
```

```
##   25.02     43       1    0.667  0.0542        0.569        0.782
##   29.70     40       1    0.650  0.0553        0.550        0.768
##   31.15     38       1    0.633  0.0565        0.532        0.754
##   31.25     37       1    0.616  0.0575        0.513        0.740
##   33.41     32       1    0.597  0.0588        0.492        0.724
##   35.93     28       1    0.575  0.0605        0.468        0.707
##   39.54     25       1    0.552  0.0623        0.443        0.689
##   46.16     18       1    0.522  0.0659        0.407        0.668
##   47.80     16       1    0.489  0.0694        0.370        0.646
##   51.02     13       1    0.452  0.0736        0.328        0.621
```

```
fit.sex = survfit(Surv(time, status)~sex)
plot(fit.sex, xlab="Months", ylab="Est. Probability of Survival", col=c(2,4))
legend("bottomleft", levels(sex), col=c(2,4), lty=1)
```




```
# Log-Rank test
logrank.test = survdiff(Surv(time, status)~sex)
logrank.test  # p-value of 0.2 indicates null cannot be rejected
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ sex)
##
##              N Observed Expected (O-E)^2/E (O-E)^2/V
## sex=Female  45       15     18.5     0.676      1.44
## sex=Male    43       20     16.5     0.761      1.44
##
##  Chisq= 1.4  on 1 degrees of freedom, p= 0.2
```

## Cox Proportional Hazard model

The diagnosis variable has been coded so that the baseline corresponds to meningioma. The results indicate that the risk associated with HG glioma is more than eight times (i.e. $e^{2.15} = 8.62$) the risk associated with meningioma. In addition, larger values of the Karnofsky index, ki, are associated with lower risk, i.e. longer survival.

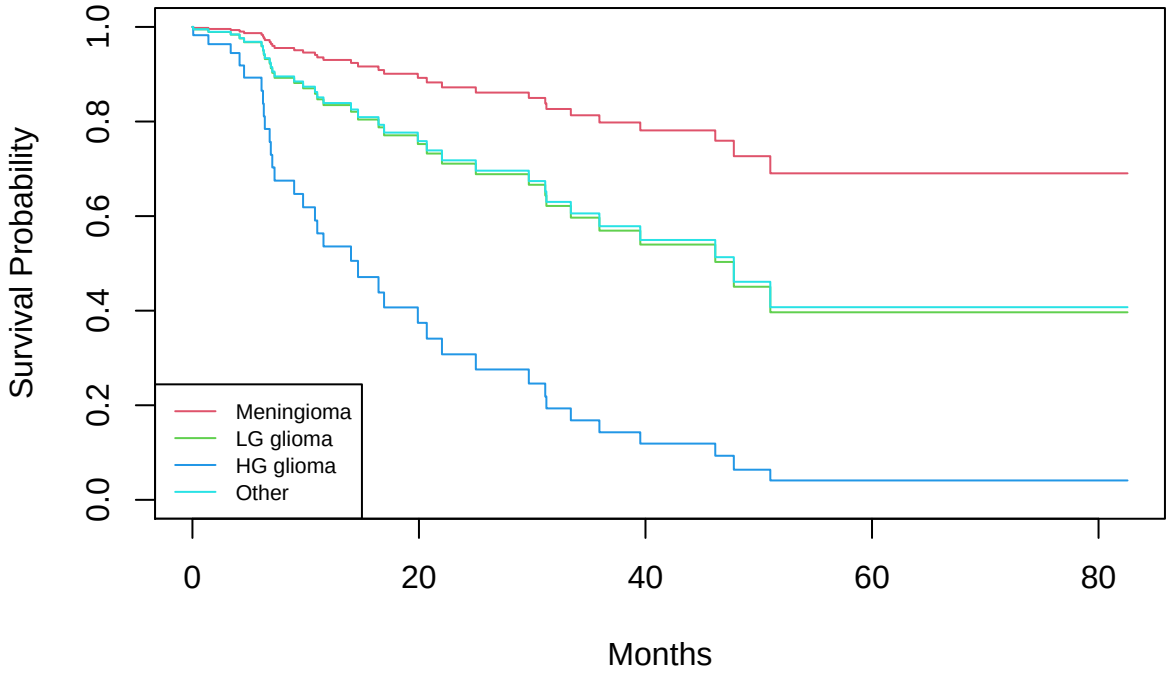
```
# Only uses "sex" as the predictor
fit.cox = coxph(Surv(time, status)~sex)
summary(fit.cox)  # p=0.2, no evidence for differenve in survival times by sex
```

```
## Call:
## coxph(formula = Surv(time, status) ~ sex)
##
##   n= 88, number of events= 35
##
##           coef exp(coef) se(coef)     z Pr(>|z|)
## sexMale 0.4077    1.5033   0.3420 1.192    0.233
##
##         exp(coef) exp(-coef) lower .95 upper .95
## sexMale     1.503     0.6652     0.769     2.939
##
## Concordance= 0.565  (se = 0.045 )
## Likelihood ratio test= 1.44  on 1 df,   p=0.2
## Wald test            = 1.42  on 1 df,   p=0.2
## Score (logrank) test = 1.44  on 1 df,   p=0.2
```

```
# Fitting model with more predictors
fit.all = coxph(Surv(time,status)~sex+diagnosis+loc+ki+gtv+stereo)
fit.all
```

```
## Call:
## coxph(formula = Surv(time, status) ~ sex + diagnosis + loc +
##     ki + gtv + stereo)
##
##                         coef exp(coef) se(coef)      z        p
## sexMale              0.18375   1.20171  0.36036  0.510  0.61012
## diagnosisLG glioma   0.91502   2.49683  0.63816  1.434  0.15161
## diagnosisHG glioma   2.15457   8.62414  0.45052  4.782 1.73e-06
## diagnosisOther       0.88570   2.42467  0.65787  1.346  0.17821
## locSupratentorial    0.44119   1.55456  0.70367  0.627  0.53066
## ki                  -0.05496   0.94653  0.01831 -3.001  0.00269
## gtv                  0.03429   1.03489  0.02233  1.536  0.12466
## stereoSRT            0.17778   1.19456  0.60158  0.296  0.76760
##
## Likelihood ratio test=41.37  on 8 df, p=1.776e-06
## n= 87, number of events= 35
##    (1 observation deleted due to missingness)
```

```
# Plotting survival curves for different diagnosis type
modal.data = data.frame(diagnosis = levels(diagnosis),
                        sex = rep("Female", 4),
                        loc = rep("Supratentorial", 4),
                        ki  = rep(mean(ki),4),
                        gtv = rep(mean(gtv),4),
                        stereo = rep("SRT", 4)
                        )
survplots = survfit(fit.all, newdata=modal.data)
plot(survplots, xlab="Months", ylab="Survival Probability", col=2:5)
legend("bottomleft", levels(diagnosis), col=2:5, lty=1, cex=0.7)
```

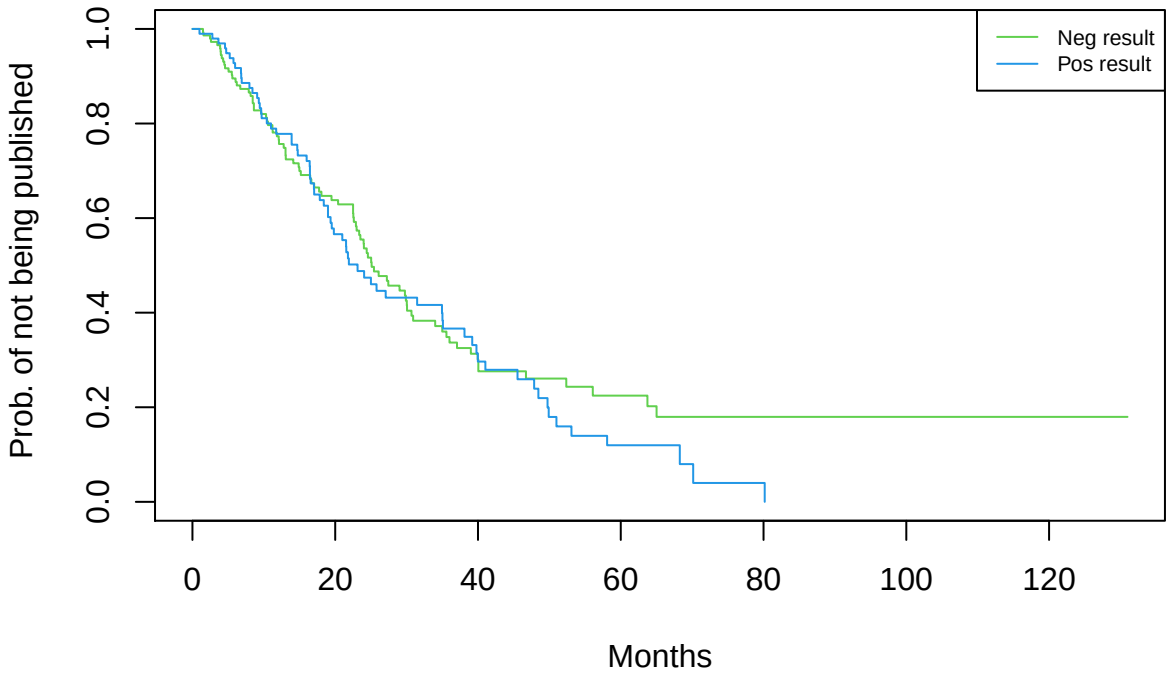



### Publication Data

```
# Kaplan-Meier curves
fit.posres = survfit(Surv(time, status)~posres, data=Publication)
fit.posres
```

```
## Call: survfit(formula = Surv(time, status) ~ posres, data = Publication)
##
##             n events median 0.95LCL 0.95UCL
## posres=0 146     87   25.1    23.0    30.7
## posres=1  98     69   23.1    19.4    35.1
```

```
plot(fit.posres, xlab="Months", ylab=" Prob. of not being published", col=3:4)
legend("topright", c("Neg result", "Pos result"), col=3:4, lty=1, cex=0.7)
```




```
# Cox's Proportional Hazard - only 1 predictor ('positive results')
fit.pub = coxph(Surv(time, status) ~ posres, data = Publication)
fit.pub
```

```
## Call:
## coxph(formula = Surv(time, status) ~ posres, data = Publication)
##
##          coef exp(coef) se(coef)     z    p
## posres 0.1481    1.1596   0.1616 0.916 0.36
##
## Likelihood ratio test=0.83  on 1 df, p=0.3611
## n= 244, number of events= 156
```

```
logrank.test = survdiff(Surv(time, status)~posres, data = Publication)
logrank.test
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ posres, data = Publication)
##
##            N Observed Expected (O-E)^2/E (O-E)^2/V
## posres=0 146       87     92.6     0.341     0.844
## posres=1  98       69     63.4     0.498     0.844
##
##  Chisq= 0.8  on 1 degrees of freedom, p= 0.4
```

```
# Cox's Proportional Hazard - all predictors
fit.pub2 = coxph(Surv(time, status) ~ . -mech, data = Publication)
fit.pub2
```

```
## Call:
## coxph(formula = Surv(time, status) ~ . - mech, data = Publication)
##
##                coef  exp(coef)   se(coef)      z       p
## posres    5.708e-01  1.770e+00  1.760e-01  3.244 0.00118
## multi    -4.086e-02  9.600e-01  2.512e-01 -0.163 0.87079
## clinend   5.462e-01  1.727e+00  2.620e-01  2.085 0.03710
## sampsize  4.678e-06  1.000e+00  1.472e-05  0.318 0.75070
## budget    4.385e-03  1.004e+00  2.465e-03  1.779 0.07518
## impact    5.832e-02  1.060e+00  6.676e-03  8.735 < 2e-16
##
## Likelihood ratio test=149.2  on 6 df, p=< 2.2e-16
## n= 244, number of events= 156
```

## Call Center Data - simulated

Simulated survival data using the sim.survdata() function, which is part of the **coxed** library. The simulated data will represent the observed wait times (in seconds) for 2,000 customers who have phoned a call center. In this context, censoring occurs if a customer hangs up before his or her call is answered.

The **sim.survdata()** function allows us to specify the maximum possible failure time, which in this case corresponds to the longest possible wait time for a customer - set at 1,000 seconds.

We find that differences between centers are highly significant, as are differences between times of day.

```
set.seed(4)
N = 2000
Operators = sample(5:15, N, replace=T)
Center = sample(c("A","B","C"), N, replace = T)
Time = sample(c("Morn.", "After.", "Even."), N, replace=T)
X = model.matrix(~Operators + Center + Time)
X = X[,-1]

true.beta = c(0.04, -0.3, 0, 0.2, -0.2)

# Baseline hazard function - with one argument, representing time
h0 = function(t) return(0.00001 * t)

library(coxed)
```

```
## Loading required package: rms

## Loading required package: Hmisc

## Loading required package: lattice

## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
## Loading required package: SparseM

##
## Attaching package: 'SparseM'

## The following object is masked from 'package:base':
##
##     backsolve

## Loading required package: mgcv

## Loading required package: nlme

## This is mgcv 1.8-34. For overview type 'help("mgcv-package")'.
```

```
queue = sim.survdata(N=N, T=1000, X=X, beta=true.beta, hazard.fun=h0)
```

```
## Warning in FUN(X[[i]], ...): 9 additional observations right-censored because the user-supplied hazard functi
##                                  is nonzero at the latest timepoint. To avoid these extra censored observati
```

```
names(queue)
```

```
## [1] "data"             "xdata"            "baseline"         "xb"
## [5] "exp.xb"           "betas"            "ind.survive"      "marg.effect"
## [9] "marg.effect.data"
```

```
head(queue$data)
```

```
##   Operators CenterB CenterC TimeEven. TimeMorn.   y failed
## 1        12       1       0         0         1 344   TRUE
## 2        15       0       0         0         0 241   TRUE
## 3         7       0       1         1         0 187   TRUE
## 4         7       0       0         0         0 279   TRUE
## 5        11       0       1         0         1 954   TRUE
## 6         7       1       0         0         1 455   TRUE
```

```
mean(queue$data$failed) # 90% of calls were answered
```

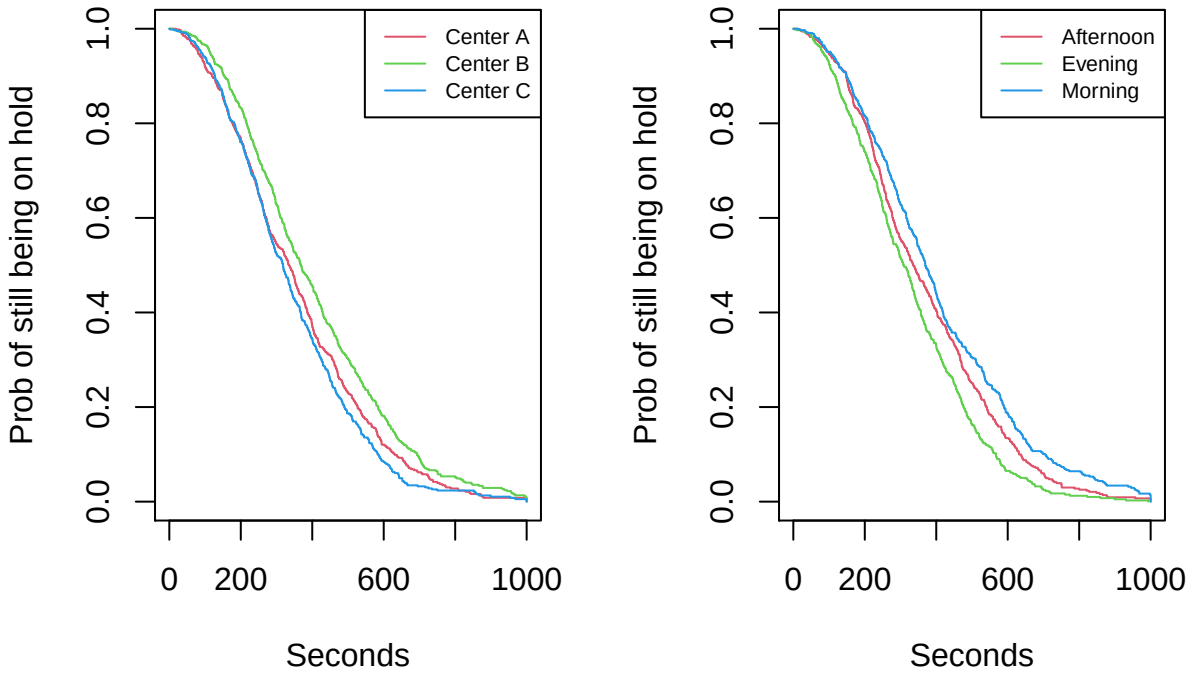
```
## [1] 0.89
```

```
# Kaplan-Meier survival curves
par(mfrow=c(1,2))

fit.Center = survfit(Surv(y, failed)~Center, data=queue$data)
plot(fit.Center, xlab="Seconds", ylab="Prob of still being on hold", col=2:4)
legend("topright", c("Center A", "Center B", "Center C"),col=2:4, lty=1, cex=0.7)
survdiff(Surv(y, failed)~Center, data=queue$data)
```

```
## Call:
## survdiff(formula = Surv(y, failed) ~ Center, data = queue$data)
##
##            N Observed Expected (O-E)^2/E (O-E)^2/V
## Center=A 683      603      579     0.971      1.45
## Center=B 667      600      701    14.641     24.64
## Center=C 650      577      499    12.062     17.05
##
##  Chisq= 28.3  on 2 degrees of freedom, p= 7e-07
```

```
fit.Time = survfit(Surv(y, failed)~Time, data=queue$data)
plot(fit.Time, xlab="Seconds", ylab="Prob of still being on hold", col=2:4)
legend("topright", c("Afternoon", "Evening", "Morning"),col=2:4, lty=1, cex=0.7)
```




```
survdiff(Surv(y, failed)~Time, data=queue$data)
```

```
## Call:
## survdiff(formula = Surv(y, failed) ~ Time, data = queue$data)
##
##               N Observed Expected (O-E)^2/E (O-E)^2/V
## Time=After. 688      616      619    0.0135     0.021
## Time=Even.  653      582      468   27.6353    38.353
## Time=Morn.  659      582      693   17.7381    29.893
##
##  Chisq= 46.8  on 2 degrees of freedom, p= 7e-11
```

```
# Cox's Proportional Hazard
fit.queue = coxph(Surv(y, failed)~., data = queue$data)
fit.queue
```

```
## Call:
## coxph(formula = Surv(y, failed) ~ ., data = queue$data)
##
##                coef exp(coef) se(coef)      z        p
## Operators   0.04174   1.04263  0.00759  5.500  3.8e-08
## CenterB    -0.21879   0.80349  0.05793 -3.777 0.000159
## CenterC     0.07930   1.08253  0.05850  1.356 0.175256
## TimeEven.   0.20904   1.23249  0.05820  3.592 0.000328
## TimeMorn. -0.17352   0.84070  0.05811 -2.986 0.002828
```

```
##
## Likelihood ratio test=102.8  on 5 df, p=< 2.2e-16
## n= 2000, number of events= 1780
```

The coefficient estimates resulting from the Cox model are fairly consistent with true estimates of 0.04, -0.3, 0, 0.2, -0.2.