Ebay Bid Time Analysis

DSTI S19 Cohort – Motoharu DEI

November 5th, 2019

Purpose

In this report I analyzed the time of trading of iPhones on an auction site "ebay", from the time when an iPhone is put on the bid until the time when that iPhone is sold.

Data Import and EDA

The data used in this report is the data about trading of iPhones prepared through 'ebay Finding API' (https://developer.ebay.com/devzone/finding/CallRef/index.html#CallIndex).

I used 'findCompletedItems' API call to get the data. Filtered data by following conditions:

- trades on ebay US
- in US dollar
- listed by fixed price (not an auction)

Extracted data has the following columns:

- condition: Information about the iPhone condition in either of Used, Manufacturer refurbished, Seller refurbished, For parts or not working, New, Open box.
- sold: 0/1 indicator if the iPhone was sold (=1) or not. Worked as censoring indicator in survival analysis.
- iphoneType: The model type of iPhone (e.g. 6S, XS, SE…)
- iphoneDiskSize: The disk size of iPhone on GB.
- unlocked: 0/1 indicator if the iPhone is locked (=0) or sim free (=1).
- dur: The number of days of each bids took until disappeared from ebay. Worked as survival time in the analysis.
- tot_price: Sales price of the iPhone including shipping cost.

I prepared two types of data sets, training data and testing data, based on the date and time of the end of the bid, meaning the iPhone was sold or the bid was cancelled.

- training data (n=1254): bid ended from 2019/10/24 5:53:57 to 2019/10/24 23:59:53
- testing data (n=427): bid ended from 2019/10/25 17:29:12 to 2019/10/25 23:58:49

Data extraction was done using Python commands because R API wrapper was old and inapplicable to current ebay API. The Jupyer Notebook files are avaiable here (https://github.com/daydreamersjp/DataScienceTechInstitute/tree/master/SurvivalAnalysis).

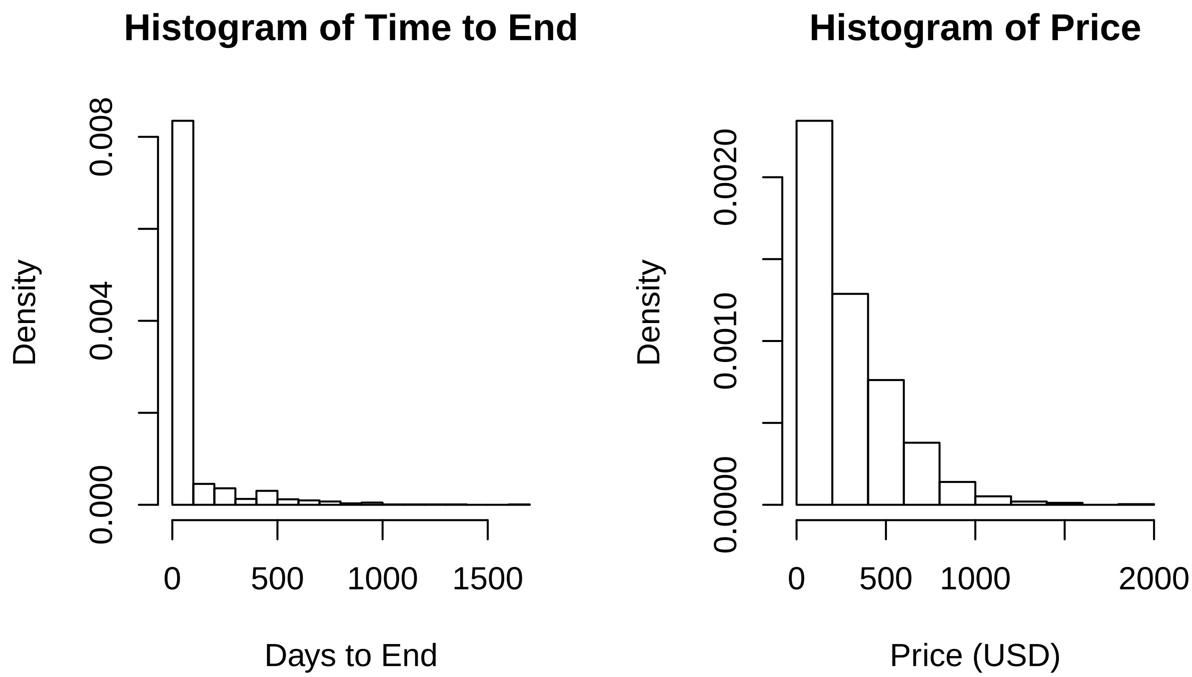
```r
library(tidyverse)
library(fastDummies)
library(survival)
library(glmnet)
library(pROC)
```

```r
dt <- read.csv('outputfile_df_final.csv')
dt_test <- read.csv('outputfile_df_final_test.csv')
```

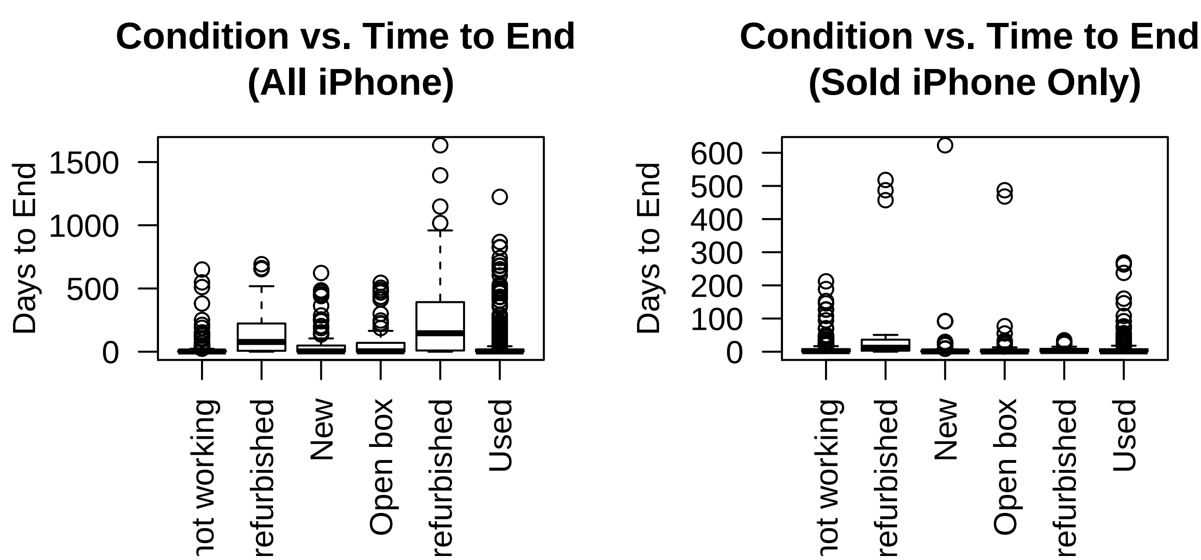
```
summary(dt,maxsum=30)
```

```
##                       condition        sold         iphoneType
##  For parts or not working:266   Min.   :0.0000   11     : 69
##  Manufacturer refurbished: 72   1st Qu.:0.0000   1st    :  3
##  New                     :132   Median :1.0000   3g     :  4
##  Open box                : 76   Mean   :0.7368   3gs    :  3
##  Seller refurbished      :139   3rd Qu.:1.0000   4      :  8
##  Used                    :569   Max.   :1.0000   4s     :  5
##                                                  5      : 30
##                                                  5c     : 18
##                                                  5s     : 38
##                                                  6      :101
##                                                  6plus  : 32
##                                                  6s     :106
##                                                  6splus: 49
##                                                  7      :143
##                                                  7plus  : 97
##                                                  8      : 76
##                                                  8plus  :111
##                                                  se     : 52
##                                                  x      :117
##                                                  xr     : 81
##                                                  xs     :111
##  iphoneDiskSize      unlocked            dur                tot_price
##  Min.   :  8.00   Min.   :0.0000   Min.   :   0.0003   Min.   :   6.0
##  1st Qu.: 32.00   1st Qu.:0.0000   1st Qu.:   0.2904   1st Qu.: 110.0
##  Median : 64.00   Median :1.0000   Median :   3.1816   Median : 219.0
##  Mean   : 86.28   Mean   :0.5128   Mean   :  73.4036   Mean   : 299.2
##  3rd Qu.: 64.00   3rd Qu.:1.0000   3rd Qu.:  31.6739   3rd Qu.: 430.0
##  Max.   :512.00   Max.   :1.0000   Max.   :1632.4643   Max.   :1805.0
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

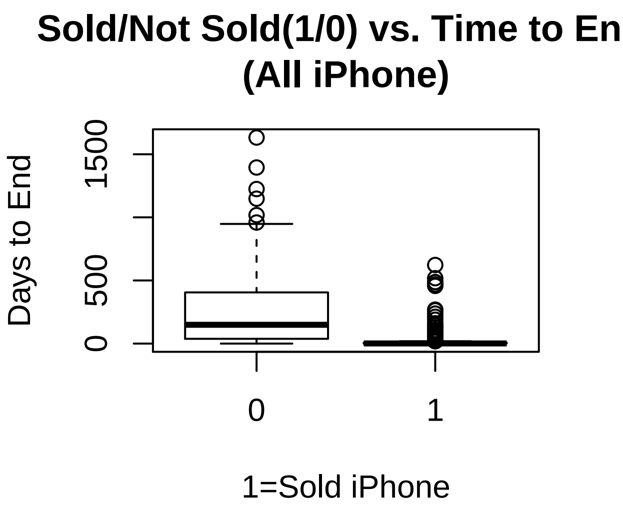
```
par(mfrow = c(1, 2))
hist(dt$dur,main="Histogram of Time to End",freq = F,xlab='Days to End')
hist(dt$tot_price, main="Histogram of Price",freq = F,xlab='Price (USD)')
```

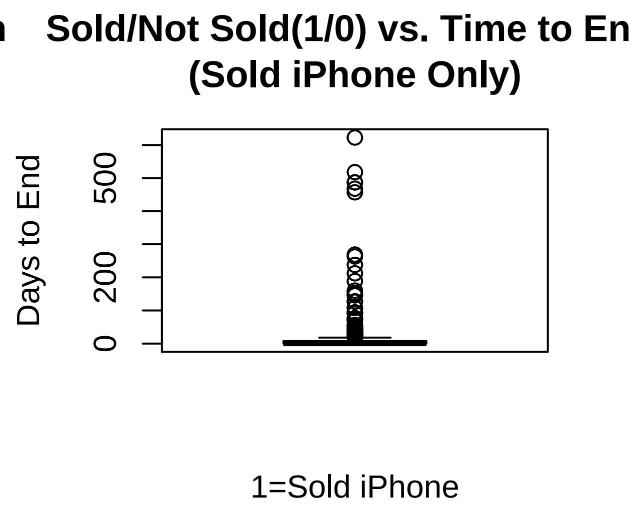



```r
par(mfrow = c(1, 2))
dts <- dt[dt$sold==1,]
plot(dur ~ condition, data=dt,las=2, main='Condition vs. Time to End \n(All iPhone)',xlab="",ylab="Days
plot(dur ~ condition, data=dts,las=2, main='Condition vs. Time to End \n(Sold iPhone Only)',xlab="",ylal
```

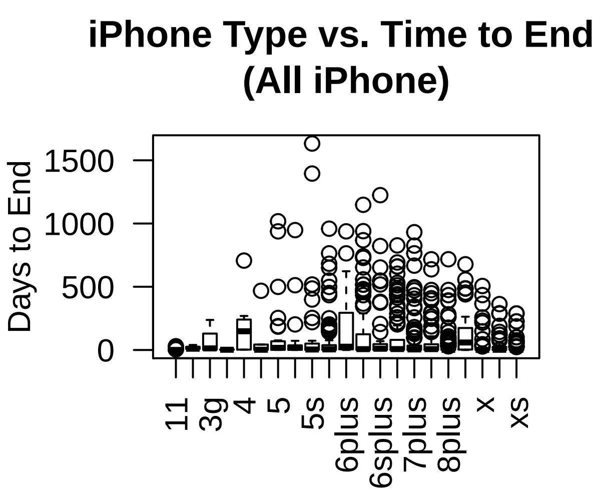


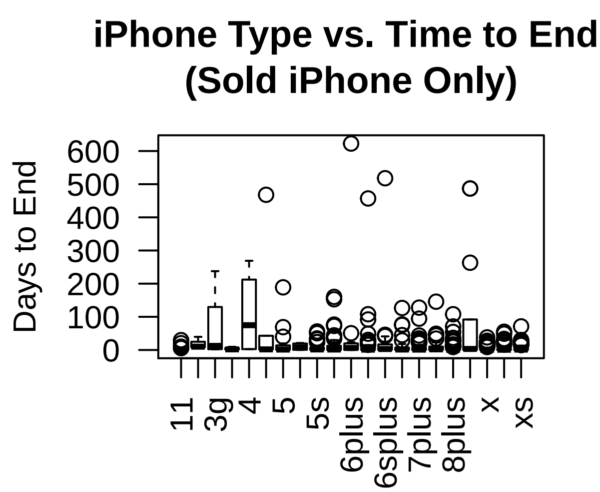

```r
plot(dur ~ factor(sold), data=dt, main='Sold/Not Sold(1/0) vs. Time to End \n(All iPhone)',xlab="1=Sold
plot(dur ~ factor(sold), data=dts, main='Sold/Not Sold(1/0) vs. Time to End \n(Sold iPhone Only)',xlab=
```



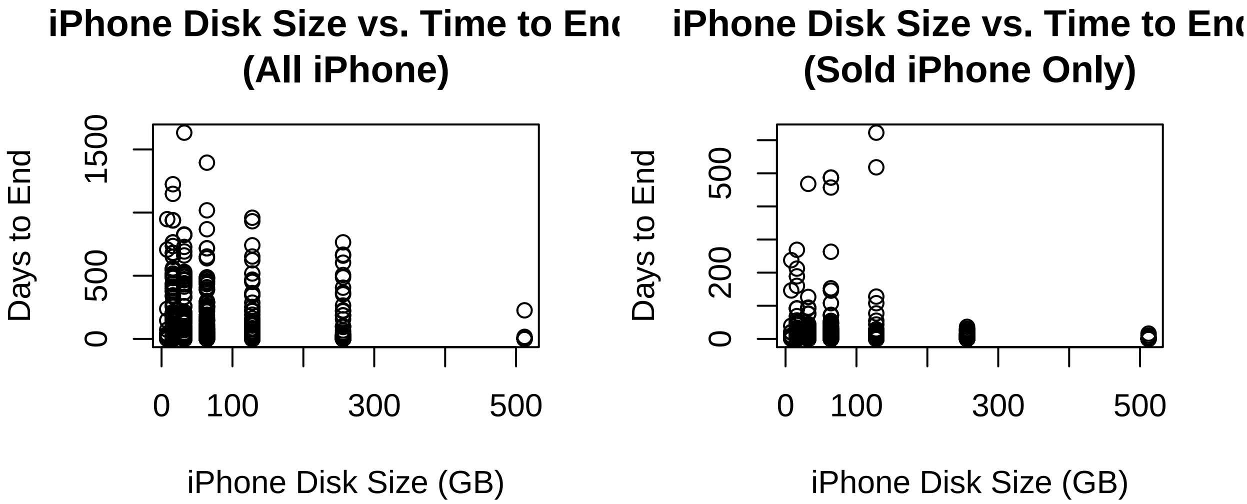



```
plot(dur ~ iphoneType, data=dt,las=2, main='iPhone Type vs. Time to End \n(All iPhone)',xlab="",ylab="Da
plot(dur ~ iphoneType, data=dts,las=2, main='iPhone Type vs. Time to End \n(Sold iPhone Only)',xlab="",y
```

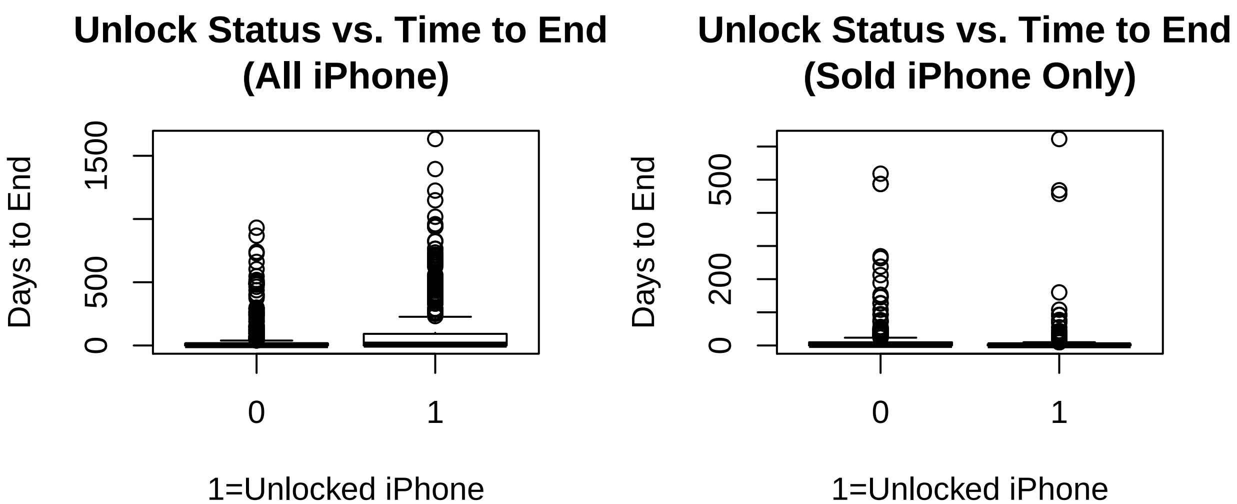




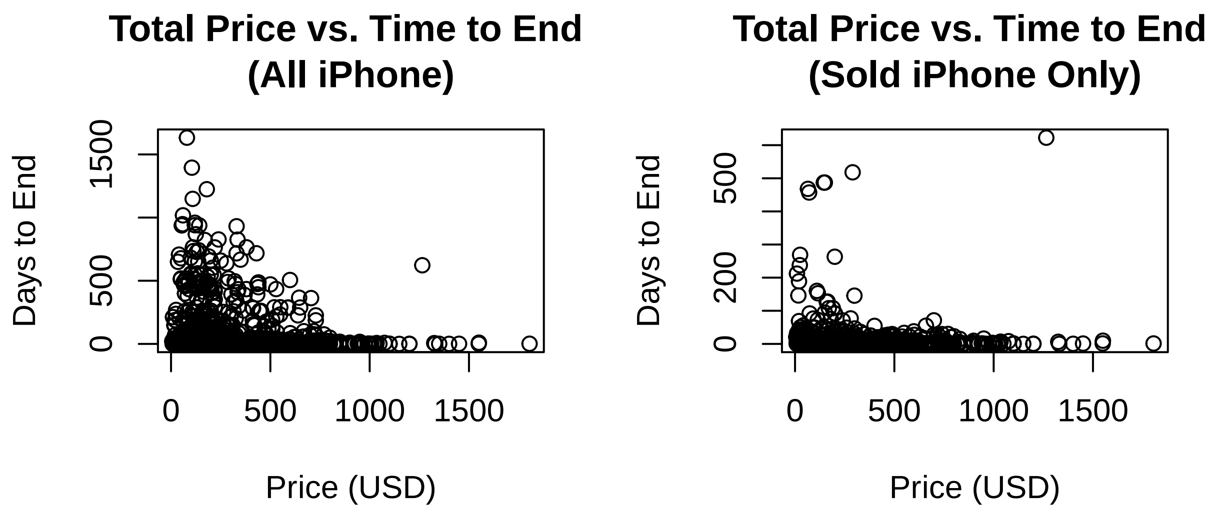

```
plot(dur ~ iphoneDiskSize, data=dt, main='iPhone Disk Size vs. Time to End \n(All iPhone)',xlab="iPhone
plot(dur ~ iphoneDiskSize, data=dts, main='iPhone Disk Size vs. Time to End \n(Sold iPhone Only)',xlab=
```




```
plot(dur ~ factor(unlocked), data=dt, main='Unlock Status vs. Time to End \n(All iPhone)',xlab="1=Unlocl
plot(dur ~ factor(unlocked), data=dts, main='Unlock Status vs. Time to End \n(Sold iPhone Only)',xlab="
```




```
plot(dur ~ tot_price, data=dt, main='Total Price vs. Time to End \n(All iPhone)',xlab="Price (USD)",ylal
plot(dur ~ tot_price, data=dts, main='Total Price vs. Time to End \n(Sold iPhone Only)',xlab="Price (USI
```

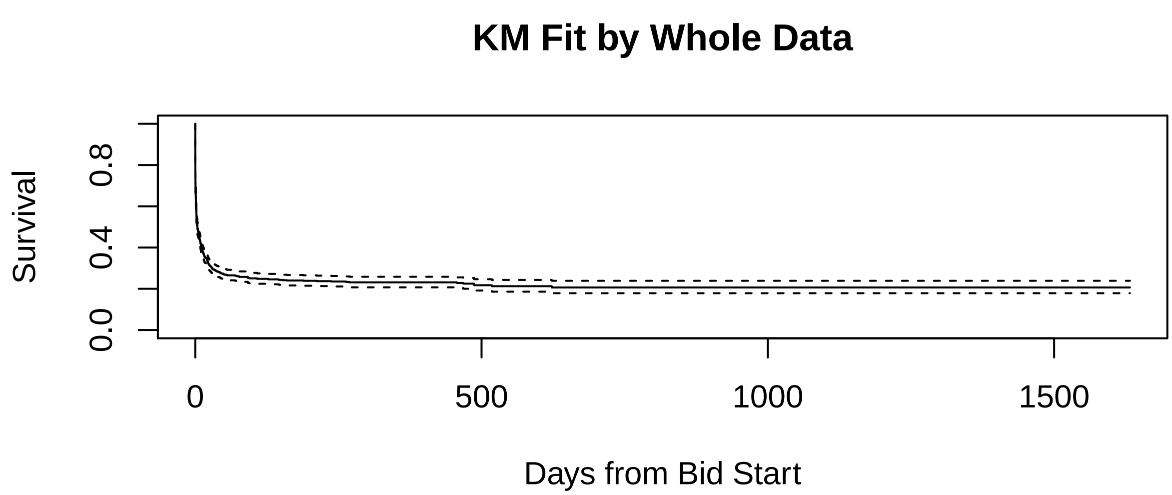


Kaplan-Meier Using Whole Data

```
kmfit_all <- survfit(Surv(dur,sold)~1, data=dt)
kmfit_all
```

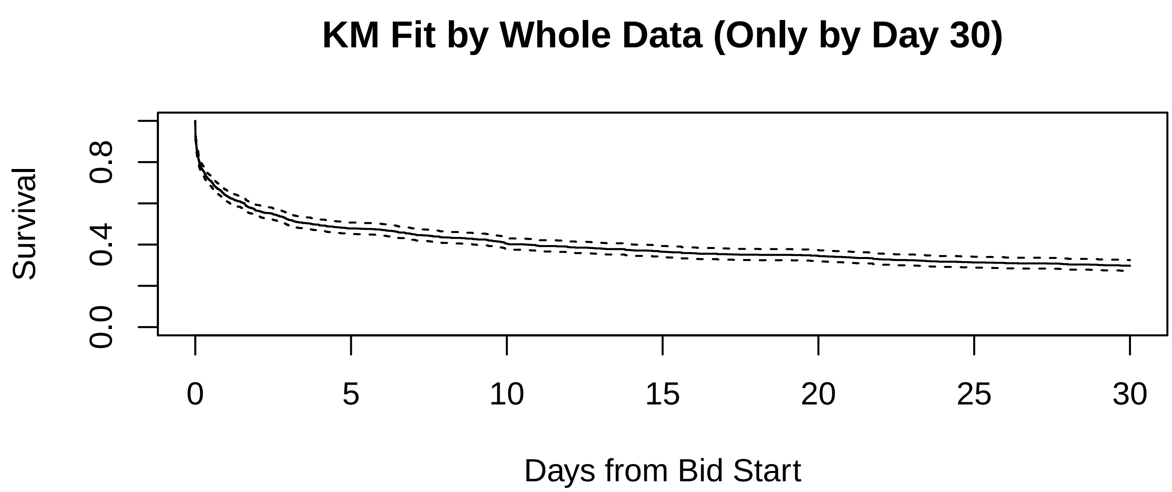
```
## Call: survfit(formula = Surv(dur, sold) ~ 1, data = dt)
##
##         n  events  median 0.95LCL 0.95UCL
## 1254.00  924.00    3.72    2.91    5.99
```

```
plot(kmfit_all,main="KM Fit by Whole Data",xlab='Days from Bid Start',ylab='Survival')
```



Kaplan-Meier Using Whole Data (only by day 30)

```
dt30 <- within(dt, {
  sold <- ifelse(dur > 30, 0, sold)
  dur <- ifelse(dur > 30, 30, dur)
})
dt30 <- mutate(dt30, condition = relevel(condition, ref = "Used"))
plot(survfit(Surv(dur,sold)~1, data=dt30),main="KM Fit by Whole Data (Only by Day 30)",xlab='Days from
```



50% of the iPhones on the bid were sold by 3.72 days, with fairly narrow confidence interval.

Cox-Proportional Hazard Model Using All Variables

```
fit.cph_all <- coxph(Surv(dur,sold)~condition+iphoneType+iphoneDiskSize+unlocked+tot_price,data=dt)
fit.cph_all
```

```
## Call:
## coxph(formula = Surv(dur, sold) ~ condition + iphoneType + iphoneDiskSize +
##     unlocked + tot_price, data = dt)
##
##                                        coef  exp(coef)   se(coef)       z
## conditionManufacturer refurbished -2.0763657  0.1253851  0.2742255  -7.572
## conditionNew                      -0.3492820  0.7051942  0.1585945  -2.202
## conditionOpen box                 -0.4436837  0.6416683  0.1646058  -2.695
## conditionSeller refurbished       -2.0361290  0.1305330  0.1991856 -10.222
## conditionUsed                     -0.1108951  0.8950327  0.0888588  -1.248
## iphoneType1st                     -1.8965336  0.1500880  0.6329416  -2.996
## iphoneType3g                      -1.9046205  0.1488791  0.5685654  -3.350
## iphoneType3gs                     -1.3945005  0.2479569  0.6445072  -2.164
## iphoneType4                       -2.4838703  0.0834197  0.4865371  -5.105
```

```
## iphoneType4s                     -1.7003319  0.1826229  0.5260555  -3.232
## iphoneType5                      -2.1137373  0.1207857  0.3515951  -6.012
## iphoneType5c                     -2.1651966  0.1147274  0.4261234  -5.081
## iphoneType5s                     -1.6341824  0.1951118  0.3169463  -5.156
## iphoneType6                      -1.7550673  0.1728956  0.2732847  -6.422
## iphoneType6plus                  -2.2436740  0.1060681  0.3382985  -6.632
## iphoneType6s                     -1.7223753  0.1786413  0.2767477  -6.224
## iphoneType6splus                 -1.9041404  0.1489506  0.2897694  -6.571
## iphoneType7                      -1.7063394  0.1815291  0.2515494  -6.783
## iphoneType7plus                  -1.4596727  0.2323123  0.2438341  -5.986
## iphoneType8                      -1.5246276  0.2177021  0.2466792  -6.181
## iphoneType8plus                  -1.1362348  0.3210255  0.2181616  -5.208
## iphoneTypese                     -2.4666895  0.0848653  0.3298924  -7.477
## iphoneTypex                      -0.9329487  0.3933920  0.1965820  -4.746
## iphoneTypexr                     -1.0726627  0.3420964  0.2092512  -5.126
## iphoneTypexs                     -0.7993065  0.4496407  0.1770030  -4.516
## iphoneDiskSize                    0.0010831  1.0010837  0.0004593   2.358
## unlocked                         -0.0109522  0.9891075  0.0769006  -0.142
## tot_price                        -0.0011046  0.9988961  0.0003307  -3.341
##                                         p
## conditionManufacturer refurbished 3.68e-14
## conditionNew                      0.027640
## conditionOpen box                 0.007030
## conditionSeller refurbished        < 2e-16
## conditionUsed                     0.212034
## iphoneType1st                     0.002732
## iphoneType3g                      0.000808
## iphoneType3gs                     0.030490
## iphoneType4                       3.30e-07
## iphoneType4s                      0.001228
## iphoneType5                       1.83e-09
## iphoneType5c                      3.75e-07
## iphoneType5s                      2.52e-07
## iphoneType6                       1.34e-10
## iphoneType6plus                   3.31e-11
## iphoneType6s                      4.86e-10
## iphoneType6splus                  4.99e-11
## iphoneType7                       1.17e-11
## iphoneType7plus                   2.15e-09
## iphoneType8                       6.39e-10
## iphoneType8plus                   1.91e-07
## iphoneTypese                      7.59e-14
## iphoneTypex                       2.08e-06
## iphoneTypexr                      2.96e-07
## iphoneTypexs                      6.31e-06
## iphoneDiskSize                    0.018367
## unlocked                          0.886748
## tot_price                         0.000836
##
## Likelihood ratio test=485.5  on 28 df, p=< 2.2e-16
## n= 1254, number of events= 924
```

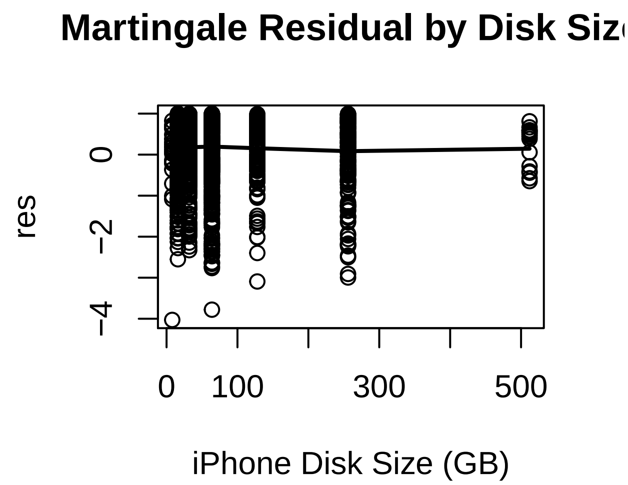
Schoenfeld Residuals Test

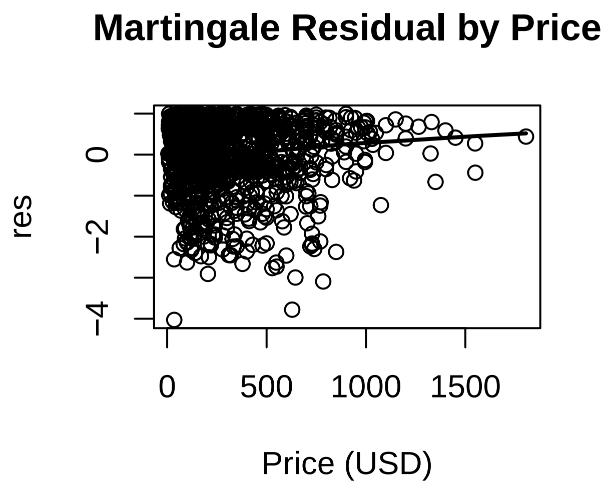
```
residual.sch <- cox.zph(fit.cph_all)
residual.sch
```

```
##                                      rho   chisq        p
## conditionManufacturer refurbished  0.0569   3.076 7.95e-02
## conditionNew                      -0.0540   3.017 8.24e-02
## conditionOpen box                 -0.1348  18.104 2.09e-05
## conditionSeller refurbished       -0.0612   3.759 5.25e-02
## conditionUsed                     -0.0945   8.768 3.07e-03
## iphoneType1st                      0.0883   7.086 7.77e-03
## iphoneType3g                       0.0944   8.060 4.53e-03
## iphoneType3gs                      0.1053  10.250 1.37e-03
## iphoneType4                        0.0924   7.668 5.62e-03
## iphoneType4s                       0.1030   9.826 1.72e-03
## iphoneType5                        0.1028  10.118 1.47e-03
## iphoneType5c                       0.0942   8.289 3.99e-03
## iphoneType5s                       0.0830   6.358 1.17e-02
## iphoneType6                        0.1006   9.156 2.48e-03
## iphoneType6plus                    0.0990   9.591 1.96e-03
## iphoneType6s                       0.0753   5.368 2.05e-02
## iphoneType6splus                   0.1106  11.067 8.79e-04
## iphoneType7                        0.0776   5.340 2.08e-02
## iphoneType7plus                    0.0676   3.934 4.73e-02
## iphoneType8                        0.0827   5.847 1.56e-02
## iphoneType8plus                    0.0493   2.078 1.49e-01
## iphoneTypese                       0.0939   8.484 3.58e-03
## iphoneTypex                        0.0747   4.664 3.08e-02
## iphoneTypexr                       0.0640   3.629 5.68e-02
## iphoneTypexs                       0.0929   7.742 5.40e-03
## iphoneDiskSize                     0.0283   0.723 3.95e-01
## unlocked                          -0.2186  52.553 4.19e-13
## tot_price                          0.0851   7.595 5.85e-03
## GLOBAL                                 NA 113.155 3.38e-12
```

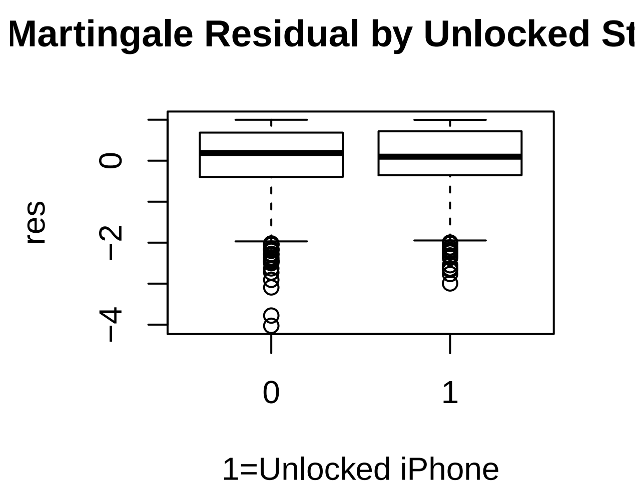
Martingale Residuals

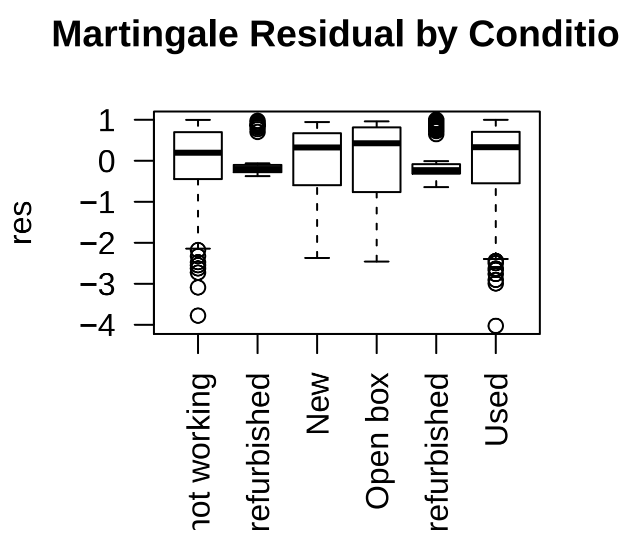
```
res <- residuals(fit.cph_all,type='martingale')

par(mfrow = c(1, 2))
plot(dt$iphoneDiskSize, res,main="Martingale Residual by Disk Size",xlab="iPhone Disk Size (GB)")
lines(lowess(dt$iphoneDiskSize,res),lwd=2)

plot(dt$tot_price, res,main="Martingale Residual by Price",xlab="Price (USD)")
lines(lowess(dt$tot_price,res),lwd=2)
```



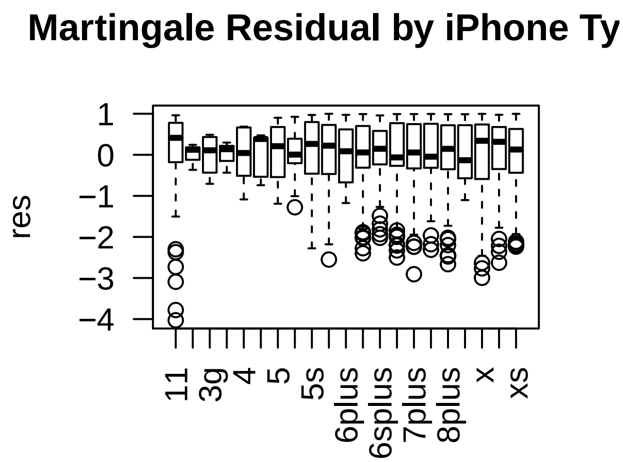



```
plot(res ~ factor(dt$unlocked),main="Martingale Residual by Unlocked Status",xlab="1=Unlocked iPhone")
plot(res ~ dt$condition,las=2,main="Martingale Residual by Condition",xlab="")
```






```
plot(res ~ dt$iphoneType,las=2,main="Martingale Residual by iPhone Type",xlab="")
```



AUCROC on Testing Data

```r
predproba <- exp(-predict(object=fit.cph_all, newdata=dt_test, type='expected'))
auc1 <- roc(dt_test$sold,predproba)$auc
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
auc1
```

```
## Area under the curve: 0.6105
```

The CPH coefficients were mostly significant but the p-values of Schoenfeld Residuals Test were small for most items, which meant there was a doubt in the proportionality assumption.

Looking at the Martingale Residual plots, condition and iphoneType, the variables with a lot of classes had larger up and down by class. We may be better off with selecting a part of their classes using variable selection algorithm such as LASSO as I did in the next session below.

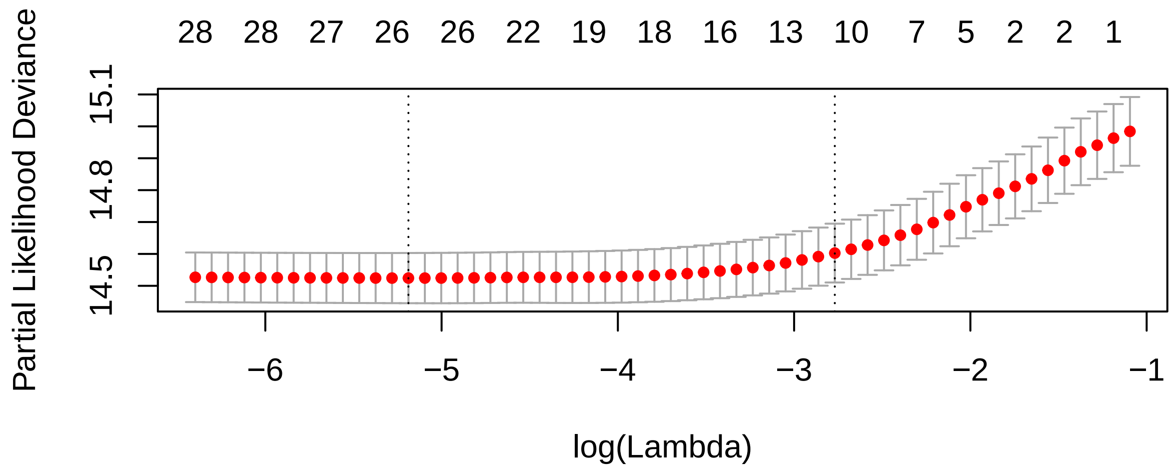
Another finding was that the coefficients of unlock, iphoneDiskSize, and tot_price were very small. It indicates that they have less influence on the survival than condition or iphoneType.

Cox-Proportional Hazard Model with LASSO

Variable Selection Using LASSO and 1 SE Rule

```r
dtdum <- dummy_cols(dt,select_columns = 'condition')
dtdum <- dummy_cols(dtdum,select_columns = 'iphoneType')
x <- dtdum %>% dplyr::select(-c('condition','iphoneType','sold','dur')) %>% as.matrix()
y <- Surv(dtdum$dur,dtdum$sold)
```

```r
set.seed(1234)
fit.cv10 <- cv.glmnet(x, y, family = "cox")
plot(fit.cv10)
```




```r
coef = coef(fit.cv10, s = "lambda.1se")
coef
```

```
## 30 x 1 sparse Matrix of class "dgCMatrix"
##                                               1
## iphoneDiskSize                     0.0006139844
## unlocked                          -0.0698934897
## tot_price                          0.0000152895
## condition_For parts or not working  0.2703676305
## condition_Manufacturer refurbished -1.0108761201
## condition_New                      .
## condition_Open box                 .
## condition_Seller refurbished      -1.1689409828
## condition_Used                     0.0747964588
## iphoneType_11                      0.4639158583
## iphoneType_1st                     .
## iphoneType_3g                      .
## iphoneType_3gs                     .
## iphoneType_4                       .
## iphoneType_4s                      .
## iphoneType_5                       .
## iphoneType_5c                      .
## iphoneType_5s                      .
## iphoneType_6                       .
## iphoneType_6plus                  -0.1366913791
## iphoneType_6s                      .
## iphoneType_6splus                  .
## iphoneType_7                       .
## iphoneType_7plus                   .
## iphoneType_8                       .
```

```
## iphoneType_8plus                     .
## iphoneType_se                       -0.3755731158
## iphoneType_x                         0.1206782630
## iphoneType_xr                        .
## iphoneType_xs                        .
```

Most of the iphoneType were eliminated in LASSO and some types were outstandingly high or low than others (e.g. iPhone 11 (the latest model) were sold much more quickly than other models).

Next, I will group the LASSO-eliminated classes and check the KM curve and proportionality by log-log plot.

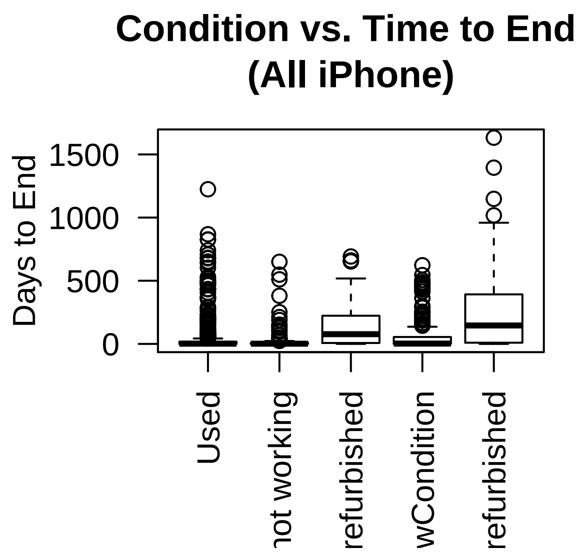
Deep dive to LASSO-selected Variables.

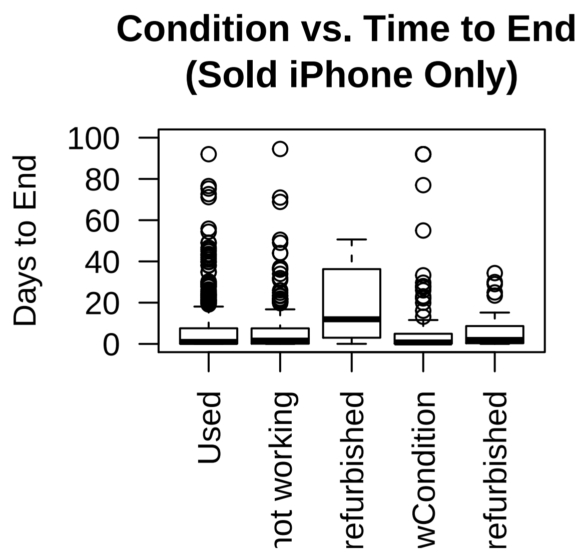
```r
# Relabeling the condition and iphoneType based on LASSO result.
condition_orig <- c('For parts or not working','Manufacturer refurbished','New','Open box','Seller refu

type_orig <- c('11','1st','3g','3gs','4','4s','5','5c','5s','6','6plus','6s','6splus','7','7plus','8','

condition_regrp <- c('For parts or not working','Manufacturer refurbished','NewCondition','NewCondition

type_regrp0 <- c('11','OthType','OthType','OthType','OthType','OthType','OthType','OthType','OthType','

look_c <- data.frame(condition_orig,condition_regrp)
look_t <- data.frame(type_orig,type_regrp0)

regroup0 <- function(dt){
  dt_regrp <- dt
  dt_regrp$condition_regrp <- sapply(dt_regrp$condition, function(x) look_c$condition_regrp[match(x,loo
  dt_regrp$iphoneType_regrp <- sapply(dt_regrp$iphoneType, function(x) look_t$type_regrp0[match(x,look_
  dt_regrp <- mutate(dt_regrp, condition_regrp = relevel(condition_regrp, ref = "Used"))
  dt_regrp <- mutate(dt_regrp, iphoneType_regrp = relevel(iphoneType_regrp, ref = "OthType"))
  regroup0 = dt_regrp
}
```

```r
dt_regrp <- regroup0(dt)
summary(dt_regrp[c('condition_regrp','iphoneType_regrp')],maxsum=30)
```

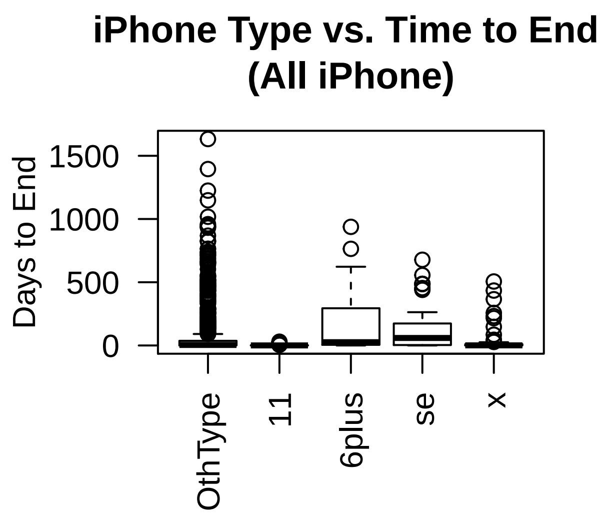
```
##                   condition_regrp iphoneType_regrp
##  Used                     :569    OthType:984
##  For parts or not working:266    11     : 69
##  Manufacturer refurbished: 72    6plus  : 32
##  NewCondition             :208    se     : 52
##  Seller refurbished       :139    x      :117
```

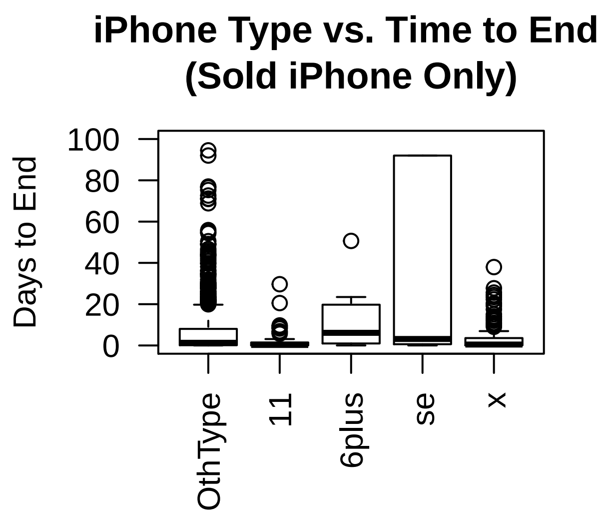
```r
par(mfrow = c(1, 2))
dt_regrps <- dt_regrp[dt_regrp$sold==1,]
plot(dur ~ condition_regrp, data=dt_regrp,las=2, main='Condition vs. Time to End \n(All iPhone)',xlab='
plot(dur ~ condition_regrp, data=dt_regrps,las=2, ylim=c(0,100), main='Condition vs. Time to End \n(Sol
```






```
plot(dur ~ iphoneType_regrp, data=dt_regrp,las=2, main='iPhone Type vs. Time to End \n(All iPhone)',xlal
plot(dur ~ iphoneType_regrp, data=dt_regrps,las=2, ylim=c(0,100), main='iPhone Type vs. Time to End \n(S
```





Survival Curve by Kaplan-Meier Based on Regrouped Variables (condition_regrp and iphoneType_regrp) Separately

```
dt30_regrp <- within(dt_regrp, {
  sold <- ifelse(dur > 30, 0, sold)
  dur <- ifelse(dur > 30, 30, dur)
})
```

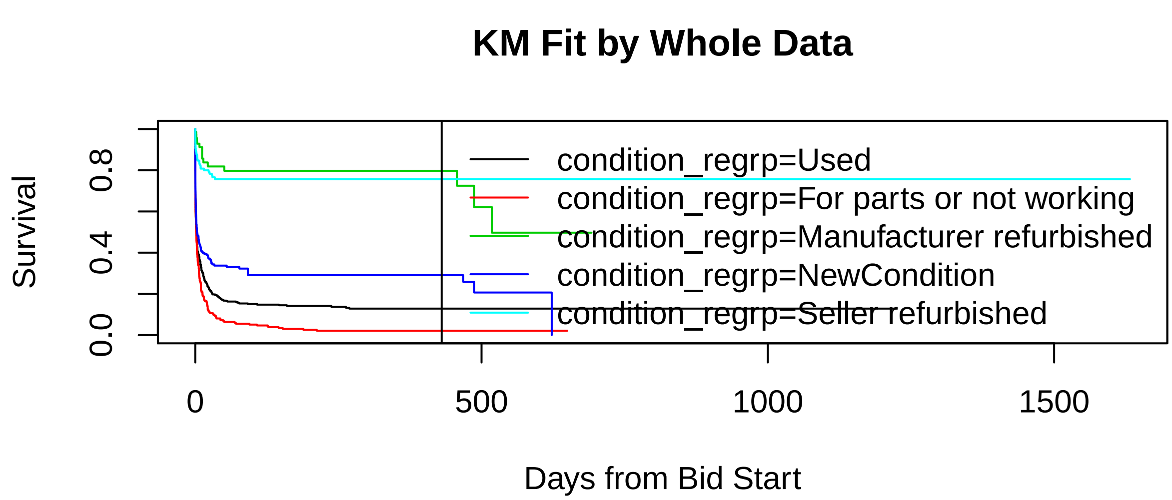
```r
fitkm <- survfit(Surv(dur,sold)~condition_regrp,data=dt_regrp)
fitkm
```

```
## Call: survfit(formula = Surv(dur, sold) ~ condition_regrp, data = dt_regrp)
##
##                                               n events median 0.95LCL 0.95UCL
## condition_regrp=Used                        569    479   1.93    1.33    2.71
## condition_regrp=For parts or not working 266    255   1.62    1.18    2.70
## condition_regrp=Manufacturer refurbished  72     15 518.00  487.00      NA
## condition_regrp=NewCondition              208    143   3.10    1.61    9.40
## condition_regrp=Seller refurbished        139     32     NA      NA      NA
```

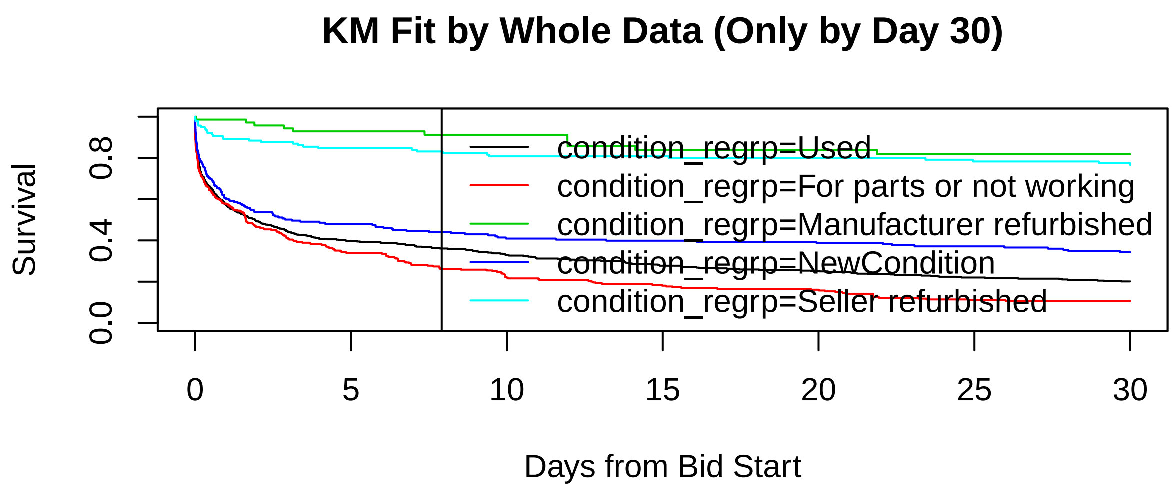
```r
survdiff(Surv(dur,sold)~condition_regrp,data=dt_regrp)
```

```
## Call:
## survdiff(formula = Surv(dur, sold) ~ condition_regrp, data = dt_regrp)
##
##                                               N Observed Expected (O-E)^2/E
## condition_regrp=Used                        569      479    370.2    32.005
## condition_regrp=For parts or not working 266      255    153.0    67.983
## condition_regrp=Manufacturer refurbished  72       15     83.2    55.876
## condition_regrp=NewCondition              208      143    153.9     0.775
## condition_regrp=Seller refurbished        139       32    163.7   105.993
##                                          (O-E)^2/V
## condition_regrp=Used                        53.829
## condition_regrp=For parts or not working    82.722
## condition_regrp=Manufacturer refurbished    61.800
## condition_regrp=NewCondition                 0.931
## condition_regrp=Seller refurbished         132.606
##
##  Chisq= 275  on 4 degrees of freedom, p= <2e-16
```

```r
plot(fitkm,col=1:5,main="KM Fit by Whole Data",xlab='Days from Bid Start',ylab='Survival')
legend("topright",lty=1,col=1:5,legend = names(fitkm$strata))
```



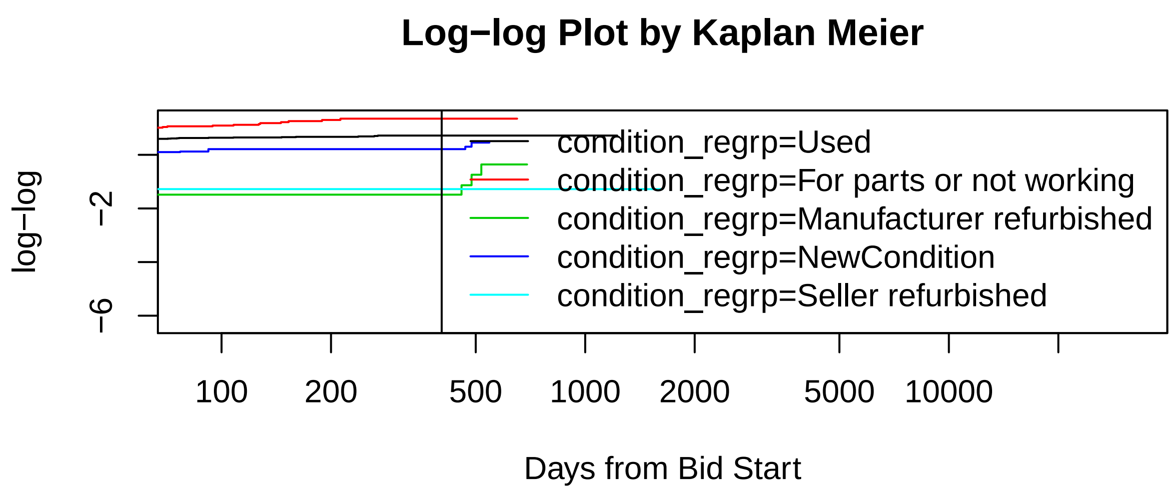
```r
plot(survfit(Surv(dur,sold)~condition_regrp,data=dt30_regrp),col=1:5,main="KM Fit by Whole Data (Only b
legend("topright",lty=1,col=1:5,legend = names(fitkm$strata))
```




```r
plot(fitkm, fun = "cloglog", col = 1:5,main="Log-log Plot by Kaplan Meier",xlab='Days from Bid Start',y
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 1 x value <= 0 omitted
## from logarithmic plot
```

```r
legend("bottomright",lty=1,col=1:5,legend = names(fitkm$strata))
```




```r
fitkm <- survfit(Surv(dur,sold)~iphoneType_regrp,data=dt_regrp)
fitkm
```

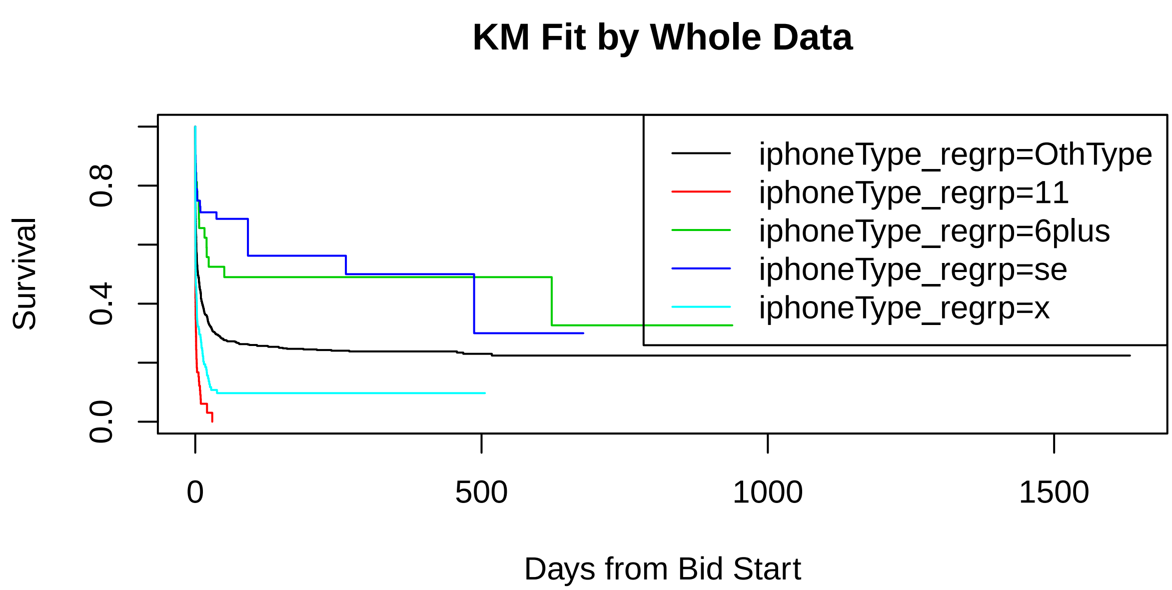
```
## Call: survfit(formula = Surv(dur, sold) ~ iphoneType_regrp, data = dt_regrp)
```

```
##
##                              n events  median 0.95LCL 0.95UCL
## iphoneType_regrp=OthType 984    715   4.694   3.255   7.054
## iphoneType_regrp=11       69     66   0.301   0.128   0.633
## iphoneType_regrp=6plus    32     17  50.644  16.169      NA
## iphoneType_regrp=se       52     23 263.103  92.000      NA
## iphoneType_regrp=x       117    103   0.798   0.420   2.673
```

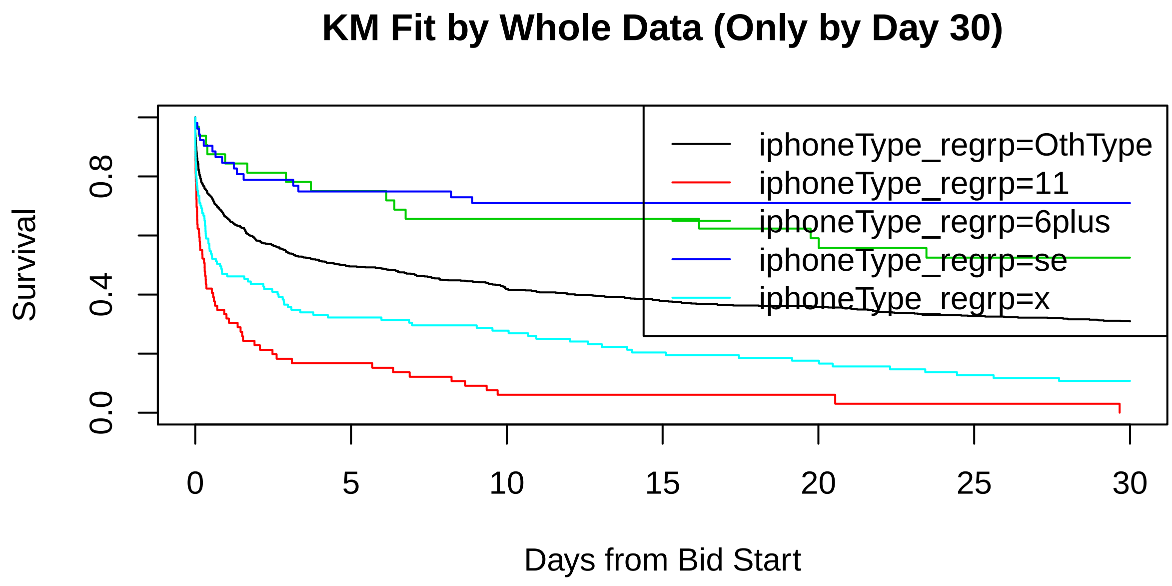
```
survdiff(Surv(dur,sold)~iphoneType_regrp,data=dt_regrp)
```

```
## Call:
## survdiff(formula = Surv(dur, sold) ~ iphoneType_regrp, data = dt_regrp)
##
##                              N Observed Expected (O-E)^2/E (O-E)^2/V
## iphoneType_regrp=OthType 984      715    744.8      1.19      6.13
## iphoneType_regrp=11       69       66     25.0     67.16     69.79
## iphoneType_regrp=6plus    32       17     33.9      8.41      8.76
## iphoneType_regrp=se       52       23     56.2     19.63     21.01
## iphoneType_regrp=x       117      103     64.1     23.57     25.44
##
##  Chisq= 122  on 4 degrees of freedom, p= <2e-16
```

```
plot(fitkm,col=1:5,main="KM Fit by Whole Data",xlab='Days from Bid Start',ylab='Survival')
legend("topright",lty=1,col=1:5,legend = names(fitkm$strata))
```



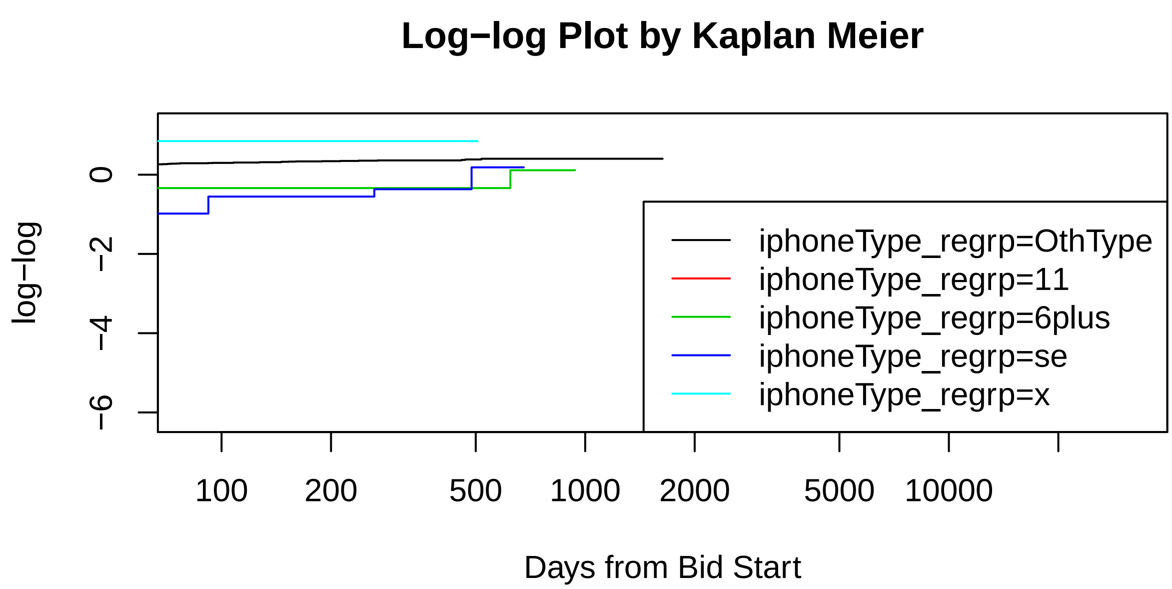

```
plot(survfit(Surv(dur,sold)~iphoneType_regrp,data=dt30_regrp),col=1:5,main="KM Fit by Whole Data (Only 
legend("topright",lty=1,col=1:5,legend = names(fitkm$strata))
```




```
plot(fitkm, fun = "cloglog", col = 1:5,main="Log-log Plot by Kaplan Meier",xlab='Days from Bid Start',yl
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 1 x value <= 0 omitted
## from logarithmic plot
```

```
legend("bottomright",lty=1,col=1:5,legend = names(fitkm$strata))
```



LogRank tests rejected the null hypothesis that the survival curves were all the same to every classes, which meant there was at least one class which was separate from others in condition and iphoneType.

Log-log plot showed the proportionality assumption was fair enough to both of condition and iphoneType with new groupings.

Among conditions, “for parts or not working”, “used”, and “new condition (regrouped from new and open box)” sold earlier in this order, and refurbished ones left unsold longer.

Among iPhone models, “11” and “x” sold earlier and “se” and “6plue” did solwer than other types.

Conclusion

The survival of bids of iPhone on ebay US (= how long it takes until the iPhone is sold on ebay) seems to highly depend on iPhone model types and condition, and less on price, sim free/unlocked, and iPhone disk size.

Refurbished iPhones from seller or manifactures are less popular than other conditions and takes longer to sell.

New iphones such as “11” or “X” tend to sell sooner, while some models such as “SE” and “6 plus” tend to do slower.