CS624 Final

Duc Le

12/17/2020

Problem 1

```
pension = read.csv("pension.csv")
pension = pension %>% select(-c(X, id)) %>%
  na.omit(pension)

base.model1 = lm(wealth89~., data = pension)
final1 = stepAIC(base.model1, trace = 0)
summary(final1)
```

```
##
## Call:
## lm(formula = wealth89 ~ age + finc50 + finc75 + finc100 + finc101 +
##     stckin89 + irain89, data = pension)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -413.65 -113.98  -46.41   69.79 1147.64
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -593.247    227.590  -2.607 0.009897 **
## age           10.677      3.736   2.858 0.004758 **
## finc50        58.452     39.542   1.478 0.141065
## finc75       168.494     48.878   3.447 0.000703 ***
## finc100      151.098     47.842   3.158 0.001857 **
## finc101      350.426     70.951   4.939 1.76e-06 ***
## stckin89     109.376     34.821   3.141 0.001963 **
## irain89       90.154     33.367   2.702 0.007542 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 211.1 on 183 degrees of freedom
## Multiple R-squared:  0.2938, Adjusted R-squared:  0.2668
## F-statistic: 10.88 on 7 and 183 DF,  p-value: 1.888e-11
```

Looking at the summary of our model, a person is estimated to have negative wealth (-$593), given no effects from all the other predictors. There seems to be a huge emphasis on how retirement contribution and investing habits (stocks, IRA) affect one's overall wealth. The majority of variables related to financial contribution holds significance for the linear regression model. The best/simplest model from stepAIC holds an R-squared of 0.2938, indicating a poor fit. This could simply be maybe these predictors just aren't ideal to be used for predicting wealth.

Problem 2

```r
travel = read.table("Travel.txt", header = T)
travel = na.omit(travel)

travel$orig_destination_distance = as.numeric(travel$orig_destination_distance)
travel = na.omit(travel)
for (i in 2:length(travel)){
  travel[,i] = as.factor(travel[,i])
}
travel$srch_adults_cnt = as.numeric(travel$srch_adults_cnt)
travel$srch_rm_cnt = as.numeric(travel$srch_rm_cnt)
travel$orig_destination_distance = as.numeric(travel$orig_destination_distance)

base.model2 = glm(is_booking~orig_destination_distance+
                    is_mobile+is_package+channel+
                    prop_is_branded+srch_adults_cnt + srch_rm_cnt+
                    prop_starrating+distance_band+
                    hist_price_band+popularity_band, data = travel, family = "binomial")
final2 = stepAIC(base.model2, trace = 0)
summary(final2)
```

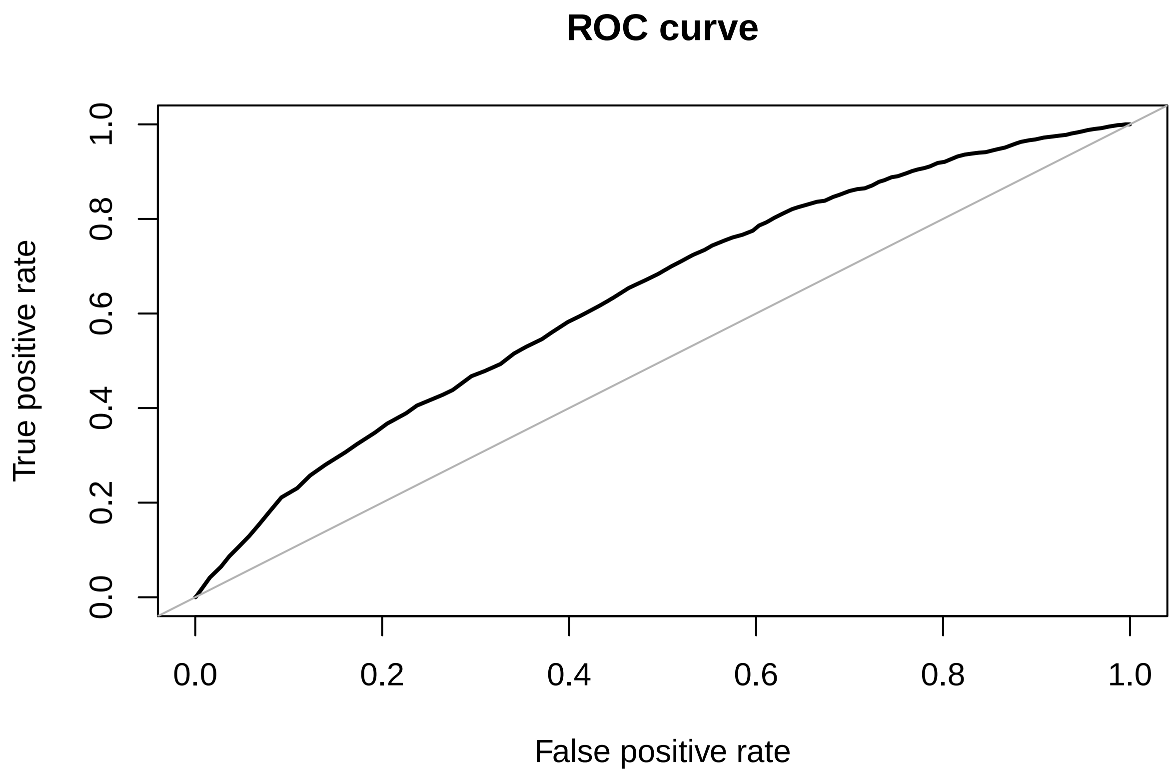
```
##
## Call:
## glm(formula = is_booking ~ is_mobile + is_package + channel +
##     prop_is_branded + srch_adults_cnt + srch_rm_cnt + prop_starrating +
##     hist_price_band + popularity_band, family = "binomial", data = travel)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8811  -0.4812  -0.4083  -0.3072   2.8218
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -2.51300    0.32548  -7.721 1.16e-14 ***
## is_mobile1           -0.12329    0.06263  -1.968  0.04903 *
## is_package1          -0.91826    0.08843 -10.385  < 2e-16 ***
## channel262           -0.10475    0.10937  -0.958  0.33821
## channel293           -0.37351    0.10425  -3.583  0.00034 ***
## channel324            0.10830    0.11490   0.943  0.34593
## channel355            0.45414    0.19706   2.305  0.02119 *
## channel386            0.20782    0.17921   1.160  0.24619
## channel417          -10.52529  162.32793  -0.065  0.94830
## channel448           -0.89349    0.36925  -2.420  0.01553 *
## channel479            0.53786    0.49341   1.090  0.27568
## channel510           -0.03039    0.09400  -0.323  0.74645
## channel541           -0.02693    0.07832  -0.344  0.73099
## prop_is_branded1      0.24636    0.05511   4.470 7.81e-06 ***
## srch_adults_cnt      -0.14559    0.03470  -4.196 2.72e-05 ***
## srch_rm_cnt           0.12764    0.06875   1.857  0.06337 .
## prop_starrating1      0.60909    0.66568   0.915  0.36020
## prop_starrating2      0.71843    0.28758   2.498  0.01248 *
## prop_starrating3      0.60413    0.27973   2.160  0.03080 *
## prop_starrating4      0.14103    0.28196   0.500  0.61697
```

```
## prop_starrating5   -0.09939    0.29364  -0.338  0.73501
## hist_price_bandL   -0.07327    0.08706  -0.842  0.39999
## hist_price_bandM   -0.04597    0.07239  -0.635  0.52539
## hist_price_bandVH   0.15868    0.10035   1.581  0.11381
## hist_price_bandVL  -0.26306    0.11514  -2.285  0.02233 *
## popularity_bandL   -0.71005    0.18130  -3.916 8.99e-05 ***
## popularity_bandM   -0.07574    0.06986  -1.084  0.27825
## popularity_bandVH   0.30430    0.06226   4.887 1.02e-06 ***
## popularity_bandVL  -0.48615    0.42381  -1.147  0.25135
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 11877  on 19992  degrees of freedom
## Residual deviance: 11495  on 19964  degrees of freedom
## AIC: 11553
##
## Number of Fisher Scoring iterations: 11
```

```r
exp(final2$coefficients)-1
```

```
##       (Intercept)        is_mobile1       is_package1        channel262
##       -0.91897515       -0.11599085       -0.60078786       -0.09944750
##        channel293        channel324        channel355        channel386
##       -0.31168622        0.11438152        0.57481544        0.23099075
##        channel417        channel448        channel479        channel510
##       -0.99997315       -0.59077563        0.71233228       -0.02993483
##        channel541  prop_is_branded1   srch_adults_cnt       srch_rm_cnt
##       -0.02656750        0.27936099       -0.13549204        0.13614718
##  prop_starrating1  prop_starrating2  prop_starrating3  prop_starrating4
##        0.83875512        1.05120573        0.82965291        0.15145346
##  prop_starrating5  hist_price_bandL  hist_price_bandM hist_price_bandVH
##       -0.09460851       -0.07065004       -0.04493251        0.17196060
## hist_price_bandVL  popularity_bandL  popularity_bandM popularity_bandVH
##       -0.23130149       -0.50838219       -0.07294563        0.35567416
## popularity_bandVL
##       -0.38500779
```

```r
p2 = predict(final2, type = "response")
roc.curve(travel$is_booking, p2)
```




```
## Area under the curve (AUC): 0.636
```

Within this abundant list of variables, it seems like only a subset of them has significance when it comes to predicting whether a customer books a hotel room. For ex: whether a customer is using a mobile app (is_mobile) has statistical significance. Whether or not the room comes in a package (is_package) is also an important predictor. Another important predictor worth noting is the popularity band of hotels relative to each other in the same destination.

Looking at the coefficients, we can exponentiate them to make it easier to interpret the odds of booking a hotel room. For ex: if the user is using a mobile app, then the odds of booking will decrease by ~0.115. If the booking is included in a package, the odds of book decrease by ~0.6.

Problem 3

```
fitglm = read.delim("FITglm2.txt", sep="\t")

fitglm2 = fitglm %>%
  select(-c(alloc, nosp)) %>% na.omit(fitglm)

risk.levels = levels(fitglm2$riskcat4)
fitglm2$riskcat4 = mapvalues(factor(fitglm2$riskcat4), from = risk.levels, to = seq(length(risk.levels)
```

```
## The following `from` values were not present in `x`:
```

```
rt.levels = levels(fitglm2$rtgroup)
fitglm2$rtgroup = mapvalues(factor(fitglm2$rtgroup), from = rt.levels, to = seq(length(rt.levels)))
```

```
## The following `from` values were not present in `x`:
```

```r
base.model3 = glm(numnosp~., data = fitglm2, family = "poisson")
final3 = stepAIC(base.model3, trace = 0)
summary(final3)
```

```
##
## Call:
## glm(formula = numnosp ~ frx + trialyrs, family = "poisson", data = fitglm2)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.53378  -0.00001  -0.00001  -0.00001   2.06252
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -23.64817  941.33476  -0.025   0.9800
## frx          23.38587  941.33474   0.025   0.9802
## trialyrs      0.09065    0.04537   1.998   0.0457 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 4115.61  on 6365  degrees of freedom
## Residual deviance:  295.88  on 6363  degrees of freedom
## AIC: 2023.9
##
## Number of Fisher Scoring iterations: 21
```

```r
1-pchisq(final3$deviance, final3$df.residual)
```

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

```r
(exp(final3$coefficients) -1)*100
```

```
##   (Intercept)           frx      trialyrs
## -1.000000e+02  1.433351e+12  9.488201e+00
```

The resulted poisson regression model from stepAIC only uses 2 covariates to predict the # of non-spinal bone fractures in women with low bone densities. "Trialyrs", the duration of follow-up is the only variable that holds statistical significance. Analyzing the effect sizes, for every unit increase in "trialyrs" or duration of follow-up, the # of expected fractures increases by 9.48%. Similarly, if the patient has spinal fractures, (frx = 1), the rate of fractures increases by 1.43%. The resulted chi-square test using the model's deviance & residual is 1, indicating the model was a good fit.

Problem 4

```r
data(wine)
base.model4 = vglm(Type~., multinomial(refLevel = 1), data = wine)
(exp(base.model4@coefficients)-1)*100
```

```
##     (Intercept):1     (Intercept):2         Alcohol:1         Alcohol:2
##      4.974049e+42      7.395403e+16     -9.922253e+01     -5.442847e+01
##           Malic:1           Malic:2             Ash:1             Ash:2
##     -8.507296e+01     -4.807708e+01     -9.999978e+01      4.223493e+02
##      Alcalinity:1      Alcalinity:2       Magnesium:1       Magnesium:2
##      1.774465e+02      6.311822e+01     -1.090468e+00     -1.200543e+01
```

```
##         Phenols:1         Phenols:2      Flavanoids:1      Flavanoids:2 
##      1.248020e+03      1.347025e+04     -6.980202e+01     -9.989260e+01 
##   Nonflavanoids:1   Nonflavanoids:2 Proanthocyanins:1 Proanthocyanins:2 
##      7.133476e+05     -9.999999e+01      1.294184e+02     -2.402657e+01 
##           Color:1           Color:2             Hue:1             Hue:2 
##     -7.864126e+01      5.409236e+02      3.564642e+06     -9.339913e+01 
##        Dilution:1        Dilution:2         Proline:1         Proline:2 
##     -9.397942e+01     -9.964835e+01     -2.390362e+00     -1.024325e+00
```

Since the reference level chosen is 1, this means the 2 lines of coefficients produced will represent the log odds of type 2 and 3 in respect of type 1. Looking at Alcohol1, every unit increase will result in a -99% for the odds of one choosing type2 wine. Every increase of Alcohol2 will result in a -54% for the odds of one picking type3 wine. There are contrasting coefficients for both odds equations, such as every unit increase in Hue, will lead to a huge bump for the odds of Type2 but a decrease in the odds for Type3.

Problem 5

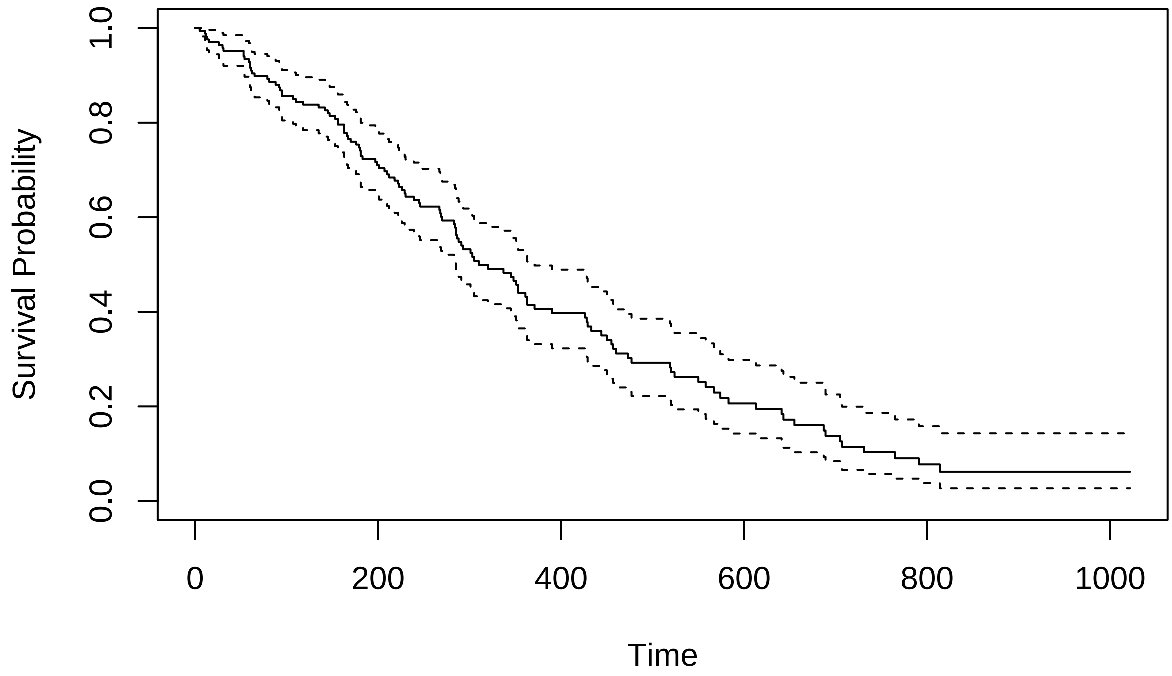
```r
data(lung)

df3 = na.omit(lung)
attach(df3)

surv.obj = Surv(time = time, event = status)

kmsurvival = survfit(surv.obj~1)
summary(kmsurvival)
```

```
## Call: survfit(formula = surv.obj ~ 1)
## 
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     5    167       1   0.9940 0.00597       0.9824        1.000
##    11    166       1   0.9880 0.00842       0.9717        1.000
##    12    165       1   0.9820 0.01028       0.9621        1.000
##    13    164       1   0.9760 0.01183       0.9531        1.000
##    15    163       1   0.9701 0.01319       0.9446        0.996
##    26    162       1   0.9641 0.01440       0.9363        0.993
##    30    161       1   0.9581 0.01551       0.9282        0.989
##    31    160       1   0.9521 0.01653       0.9203        0.985
##    53    159       2   0.9401 0.01836       0.9048        0.977
##    54    157       1   0.9341 0.01919       0.8973        0.973
##    59    156       1   0.9281 0.01998       0.8898        0.968
##    60    155       2   0.9162 0.02145       0.8751        0.959
##    61    153       1   0.9102 0.02213       0.8678        0.955
##    62    152       1   0.9042 0.02278       0.8606        0.950
##    65    151       1   0.8982 0.02340       0.8535        0.945
##    79    150       1   0.8922 0.02400       0.8464        0.941
##    81    149       1   0.8862 0.02457       0.8394        0.936
##    88    148       1   0.8802 0.02512       0.8323        0.931
##    92    147       1   0.8743 0.02566       0.8254        0.926
##    93    146       1   0.8683 0.02617       0.8185        0.921
##    95    145       2   0.8563 0.02715       0.8047        0.911
##   107    142       1   0.8503 0.02762       0.7978        0.906
##   110    141       1   0.8442 0.02807       0.7910        0.901
##   118    140       1   0.8382 0.02851       0.7841        0.896
```

```
##    135    139       1   0.8322 0.02894       0.7773        0.891
##    142    138       1   0.8261 0.02935       0.7706        0.886
##    145    137       1   0.8201 0.02975       0.7638        0.881
##    147    136       1   0.8141 0.03013       0.7571        0.875
##    153    135       1   0.8080 0.03051       0.7504        0.870
##    156    134       2   0.7960 0.03122       0.7371        0.860
##    163    132       3   0.7779 0.03221       0.7173        0.844
##    166    129       1   0.7719 0.03252       0.7107        0.838
##    167    128       1   0.7658 0.03282       0.7041        0.833
##    170    127       1   0.7598 0.03311       0.6976        0.828
##    176    124       1   0.7537 0.03341       0.6910        0.822
##    179    122       1   0.7475 0.03370       0.6843        0.817
##    180    121       1   0.7413 0.03398       0.6776        0.811
##    181    120       2   0.7290 0.03452       0.6644        0.800
##    183    118       1   0.7228 0.03478       0.6577        0.794
##    197    114       1   0.7164 0.03505       0.6510        0.789
##    199    112       1   0.7101 0.03531       0.6441        0.783
##    201    111       1   0.7037 0.03557       0.6373        0.777
##    207    108       1   0.6971 0.03583       0.6303        0.771
##    210    107       1   0.6906 0.03608       0.6234        0.765
##    212    105       1   0.6840 0.03633       0.6164        0.759
##    218    104       1   0.6775 0.03658       0.6094        0.753
##    222    102       1   0.6708 0.03681       0.6024        0.747
##    223    100       1   0.6641 0.03705       0.5953        0.741
##    226     97       1   0.6573 0.03730       0.5881        0.735
##    229     96       1   0.6504 0.03753       0.5809        0.728
##    230     95       1   0.6436 0.03776       0.5737        0.722
##    239     93       1   0.6367 0.03798       0.5664        0.716
##    245     90       1   0.6296 0.03821       0.5590        0.709
##    246     89       1   0.6225 0.03843       0.5516        0.703
##    267     85       1   0.6152 0.03867       0.5439        0.696
##    268     84       1   0.6079 0.03890       0.5362        0.689
##    269     83       1   0.6005 0.03911       0.5286        0.682
##    270     81       1   0.5931 0.03933       0.5208        0.675
##    283     79       1   0.5856 0.03954       0.5130        0.668
##    284     78       1   0.5781 0.03974       0.5052        0.661
##    285     76       2   0.5629 0.04012       0.4895        0.647
##    286     74       1   0.5553 0.04029       0.4817        0.640
##    288     73       1   0.5477 0.04045       0.4739        0.633
##    291     72       1   0.5401 0.04060       0.4661        0.626
##    293     69       1   0.5322 0.04076       0.4581        0.618
##    301     66       1   0.5242 0.04093       0.4498        0.611
##    303     64       1   0.5160 0.04110       0.4414        0.603
##    305     62       1   0.5077 0.04127       0.4329        0.595
##    310     61       1   0.4993 0.04143       0.4244        0.588
##    320     60       1   0.4910 0.04157       0.4160        0.580
##    337     58       1   0.4826 0.04170       0.4074        0.572
##    345     57       1   0.4741 0.04182       0.3988        0.564
##    348     56       1   0.4656 0.04192       0.3903        0.555
##    351     55       1   0.4572 0.04201       0.3818        0.547
##    353     54       2   0.4402 0.04212       0.3650        0.531
##    361     52       1   0.4318 0.04215       0.3566        0.523
##    363     51       2   0.4148 0.04217       0.3399        0.506
##    371     49       1   0.4064 0.04215       0.3316        0.498
```

```
## 390 45 1 0.3973 0.04217 0.3227 0.489
## 426 42 1 0.3879 0.04221 0.3134 0.480
## 428 41 1 0.3784 0.04223 0.3041 0.471
## 429 40 1 0.3690 0.04222 0.2948 0.462
## 433 39 1 0.3595 0.04218 0.2856 0.452
## 444 38 1 0.3500 0.04212 0.2765 0.443
## 450 37 1 0.3406 0.04203 0.2674 0.434
## 455 36 1 0.3311 0.04192 0.2584 0.424
## 457 35 1 0.3217 0.04177 0.2494 0.415
## 460 33 1 0.3119 0.04163 0.2401 0.405
## 473 32 1 0.3022 0.04145 0.2309 0.395
## 477 31 1 0.2924 0.04124 0.2218 0.386
## 519 29 1 0.2823 0.04104 0.2123 0.375
## 520 28 1 0.2722 0.04079 0.2030 0.365
## 524 27 1 0.2622 0.04051 0.1937 0.355
## 550 25 1 0.2517 0.04022 0.1840 0.344
## 558 23 1 0.2407 0.03993 0.1739 0.333
## 567 21 1 0.2293 0.03964 0.1634 0.322
## 574 20 1 0.2178 0.03928 0.1529 0.310
## 583 19 1 0.2063 0.03885 0.1427 0.298
## 613 18 1 0.1949 0.03835 0.1325 0.287
## 641 17 1 0.1834 0.03777 0.1225 0.275
## 643 16 1 0.1720 0.03711 0.1126 0.262
## 655 15 1 0.1605 0.03636 0.1029 0.250
## 687 14 1 0.1490 0.03552 0.0934 0.238
## 689 13 1 0.1376 0.03459 0.0840 0.225
## 705 12 1 0.1261 0.03355 0.0749 0.212
## 707 11 1 0.1146 0.03240 0.0659 0.199
## 731 10 1 0.1032 0.03112 0.0571 0.186
## 765  8 1 0.0903 0.02979 0.0473 0.172
## 791  7 1 0.0774 0.02818 0.0379 0.158
## 814  5 1 0.0619 0.02646 0.0268 0.143
```

```
plot(kmsurvival, xlab = "Time", ylab = "Survival Probability")
```



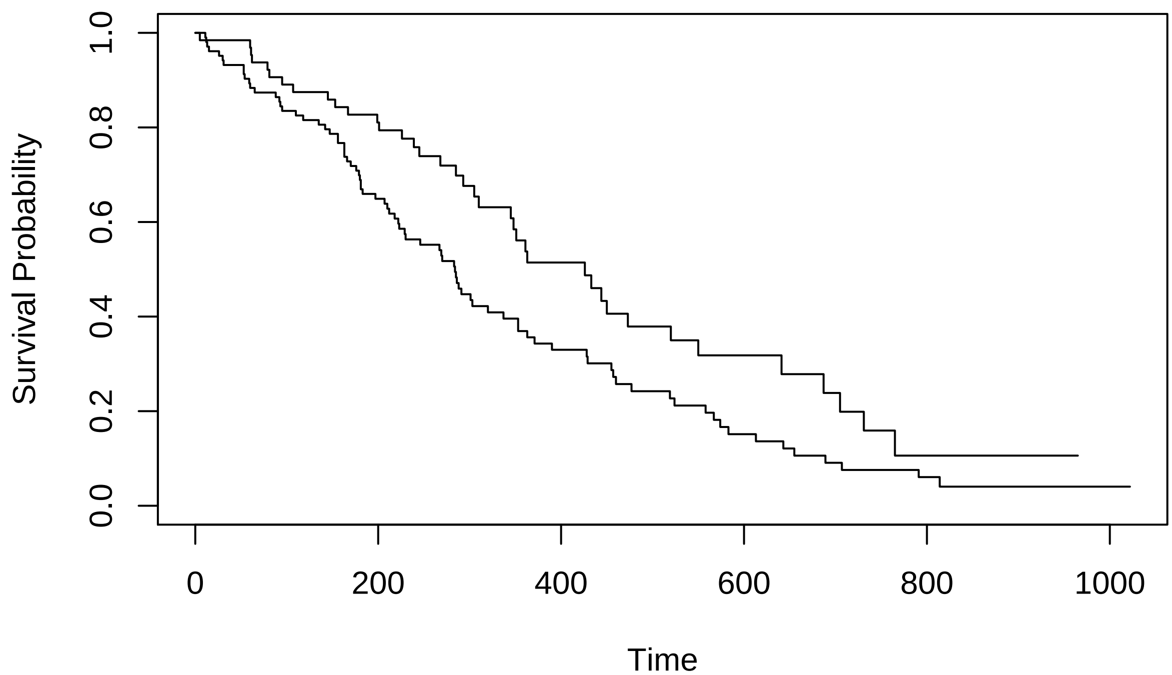

```r
kmsurvival.sex = survfit(surv.obj~ sex)
summary(kmsurvival.sex)
```

```
## Call: survfit(formula = surv.obj ~ sex)
##
##                 sex=1
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    11    103       1   0.9903 0.00966       0.9715        1.000
##    12    102       1   0.9806 0.01360       0.9543        1.000
##    13    101       1   0.9709 0.01657       0.9389        1.000
##    15    100       1   0.9612 0.01904       0.9246        0.999
##    26     99       1   0.9515 0.02118       0.9108        0.994
##    30     98       1   0.9417 0.02308       0.8976        0.988
##    31     97       1   0.9320 0.02480       0.8847        0.982
##    53     96       2   0.9126 0.02782       0.8597        0.969
##    54     94       1   0.9029 0.02917       0.8475        0.962
##    59     93       1   0.8932 0.03043       0.8355        0.955
##    60     92       1   0.8835 0.03161       0.8237        0.948
##    65     91       1   0.8738 0.03272       0.8119        0.940
##    88     90       1   0.8641 0.03377       0.8004        0.933
##    92     89       1   0.8544 0.03476       0.7889        0.925
##    93     88       1   0.8447 0.03569       0.7775        0.918
##    95     87       1   0.8350 0.03658       0.7663        0.910
##   110     86       1   0.8252 0.03742       0.7551        0.902
##   118     85       1   0.8155 0.03822       0.7440        0.894
##   135     84       1   0.8058 0.03898       0.7329        0.886
```

```
##    142      83        1   0.7961 0.03970       0.7220        0.878
##    147      82        1   0.7864 0.04038       0.7111        0.870
##    156      81        2   0.7670 0.04165       0.6895        0.853
##    163      79        3   0.7379 0.04333       0.6576        0.828
##    166      76        1   0.7282 0.04384       0.6471        0.819
##    170      75        1   0.7184 0.04432       0.6366        0.811
##    176      73        1   0.7086 0.04479       0.6260        0.802
##    179      72        1   0.6988 0.04523       0.6155        0.793
##    180      71        1   0.6889 0.04566       0.6050        0.784
##    181      70        2   0.6692 0.04642       0.5842        0.767
##    183      68        1   0.6594 0.04677       0.5738        0.758
##    197      64        1   0.6491 0.04716       0.5629        0.748
##    207      62        1   0.6386 0.04755       0.5519        0.739
##    210      61        1   0.6282 0.04791       0.5409        0.729
##    212      60        1   0.6177 0.04824       0.5300        0.720
##    218      59        1   0.6072 0.04855       0.5191        0.710
##    222      57        1   0.5966 0.04885       0.5081        0.700
##    223      55        1   0.5857 0.04915       0.4969        0.690
##    229      52        1   0.5745 0.04948       0.4852        0.680
##    230      51        1   0.5632 0.04977       0.4736        0.670
##    246      50        1   0.5519 0.05004       0.4621        0.659
##    267      48        1   0.5404 0.05030       0.4503        0.649
##    269      47        1   0.5289 0.05053       0.4386        0.638
##    270      46        1   0.5174 0.05072       0.4270        0.627
##    283      45        1   0.5059 0.05088       0.4154        0.616
##    284      44        1   0.4944 0.05101       0.4039        0.605
##    285      42        1   0.4827 0.05113       0.3922        0.594
##    286      41        1   0.4709 0.05122       0.3805        0.583
##    288      40        1   0.4591 0.05128       0.3689        0.571
##    291      39        1   0.4473 0.05129       0.3573        0.560
##    301      36        1   0.4349 0.05135       0.3451        0.548
##    303      34        1   0.4221 0.05141       0.3325        0.536
##    320      32        1   0.4089 0.05147       0.3195        0.523
##    337      31        1   0.3957 0.05147       0.3067        0.511
##    353      30        2   0.3694 0.05131       0.2813        0.485
##    363      28        1   0.3562 0.05114       0.2688        0.472
##    371      27        1   0.3430 0.05092       0.2564        0.459
##    390      26        1   0.3298 0.05064       0.2441        0.446
##    428      23        1   0.3154 0.05043       0.2306        0.432
##    429      22        1   0.3011 0.05014       0.2173        0.417
##    455      21        1   0.2868 0.04976       0.2041        0.403
##    457      20        1   0.2724 0.04929       0.1911        0.388
##    460      18        1   0.2573 0.04882       0.1774        0.373
##    477      17        1   0.2422 0.04824       0.1639        0.358
##    519      16        1   0.2270 0.04754       0.1506        0.342
##    524      15        1   0.2119 0.04672       0.1375        0.326
##    558      14        1   0.1968 0.04577       0.1247        0.310
##    567      13        1   0.1816 0.04468       0.1121        0.294
##    574      12        1   0.1665 0.04344       0.0998        0.278
##    583      11        1   0.1514 0.04205       0.0878        0.261
##    613      10        1   0.1362 0.04048       0.0761        0.244
##    643       9        1   0.1211 0.03870       0.0647        0.227
##    655       8        1   0.1059 0.03671       0.0537        0.209
##    689       7        1   0.0908 0.03444       0.0432        0.191
```

```
##   707      6        1   0.0757 0.03185       0.0332        0.173
##   791      5        1   0.0605 0.02886       0.0238        0.154
##   814      3        1   0.0404 0.02533       0.0118        0.138
##
##                 sex=2
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     5     64       1    0.984  0.0155       0.9545        1.000
##    60     63       1    0.969  0.0217       0.9270        1.000
##    61     62       1    0.953  0.0264       0.9027        1.000
##    62     61       1    0.938  0.0303       0.8800        0.999
##    79     60       1    0.922  0.0335       0.8584        0.990
##    81     59       1    0.906  0.0364       0.8376        0.981
##    95     58       1    0.891  0.0390       0.8174        0.970
##   107     56       1    0.875  0.0414       0.7972        0.960
##   145     55       1    0.859  0.0436       0.7774        0.949
##   153     54       1    0.843  0.0456       0.7581        0.937
##   167     53       1    0.827  0.0475       0.7390        0.925
##   199     50       1    0.810  0.0493       0.7194        0.913
##   201     49       1    0.794  0.0510       0.7000        0.900
##   226     45       1    0.776  0.0528       0.6794        0.887
##   239     43       1    0.758  0.0546       0.6584        0.873
##   245     40       1    0.739  0.0564       0.6366        0.859
##   268     37       1    0.719  0.0583       0.6136        0.843
##   285     34       1    0.698  0.0603       0.5894        0.827
##   293     32       1    0.676  0.0623       0.5647        0.810
##   305     30       1    0.654  0.0641       0.5394        0.792
##   310     29       1    0.631  0.0658       0.5146        0.774
##   345     27       1    0.608  0.0674       0.4892        0.755
##   348     26       1    0.584  0.0687       0.4642        0.736
##   351     25       1    0.561  0.0698       0.4397        0.716
##   361     24       1    0.538  0.0707       0.4155        0.696
##   363     23       1    0.514  0.0714       0.3918        0.675
##   426     19       1    0.487  0.0726       0.3639        0.653
##   433     18       1    0.460  0.0734       0.3366        0.629
##   444     17       1    0.433  0.0739       0.3100        0.605
##   450     16       1    0.406  0.0741       0.2839        0.581
##   473     15       1    0.379  0.0739       0.2585        0.556
##   520     13       1    0.350  0.0738       0.2314        0.529
##   550     11       1    0.318  0.0736       0.2020        0.501
##   641      8       1    0.278  0.0744       0.1648        0.470
##   687      7       1    0.239  0.0736       0.1303        0.437
##   705      6       1    0.199  0.0713       0.0984        0.401
##   731      5       1    0.159  0.0672       0.0695        0.364
##   765      3       1    0.106  0.0623       0.0335        0.335
```

```r
plot(kmsurvival.sex, xlab = "Time", ylab = "Survival Probability")
```



From the survival model plot in relation to Sex, we see that over time, males actually have a lower survival rate than women despite starting off with a higher survival probablity.

```
coxph.model = coxph(surv.obj~age+sex+ph.ecog+pat.karno+meal.cal+wt.loss, data = df3)
summary(coxph.model)
```

```
## Call:
## coxph(formula = surv.obj ~ age + sex + ph.ecog + pat.karno +
##     meal.cal + wt.loss, data = df3)
##
##   n= 167, number of events= 120
##
##                 coef  exp(coef)   se(coef)      z Pr(>|z|)
## age        5.610e-03  1.006e+00  1.127e-02  0.498  0.61878
## sex       -5.362e-01  5.850e-01  2.017e-01 -2.658  0.00785 **
## ph.ecog    4.424e-01  1.557e+00  1.713e-01  2.583  0.00979 **
## pat.karno -9.933e-03  9.901e-01  8.099e-03 -1.226  0.22005
## meal.cal   1.503e-05  1.000e+00  2.529e-04  0.059  0.95261
## wt.loss   -1.340e-02  9.867e-01  7.711e-03 -1.738  0.08228 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           exp(coef) exp(-coef) lower .95 upper .95
## age          1.0056     0.9944    0.9836    1.0281
## sex          0.5850     1.7094    0.3940    0.8686
## ph.ecog      1.5565     0.6425    1.1126    2.1775
## pat.karno    0.9901     1.0100    0.9745    1.0060
```

```
## meal.cal     1.0000     1.0000    0.9995    1.0005
## wt.loss      0.9867     1.0135    0.9719    1.0017
##
## Concordance= 0.654  (se = 0.03 )
## Likelihood ratio test= 23.95  on 6 df,   p=5e-04
## Wald test            = 23.1  on 6 df,   p=8e-04
## Score (logrank) test = 23.89  on 6 df,   p=5e-04
```

```r
percentages = round(((exp(coxph.model$coefficients)-1)*100),3)
percentages
```

```
##       age       sex   ph.ecog pat.karno  meal.cal   wt.loss
##     0.563   -41.500    55.652    -0.988     0.002    -1.331
```

Analyzing the hazard rates for each variable, we see that for every unit increase in age the subject is 0.56% more likely to die from lung cancer. Looking at ph.ecog, the more "bedbound" the subject is, the more likely he/she is to die from lung cancer. Their probability of dying increases by 55% for this predictor.