R for Health Data Science

Week 08: Biostats II

Sam Stewart

2021-04-09

```r
library(Hmisc)
library(kableExtra)
library(tidyverse)
library(ggplot2)
library(boomer)
library(survival)
library(survminer)
library(haven)
library(lme4)
library(mlmRev)

dat = read.csv("data/framinghamFirst.csv",header=TRUE,
               na.strings=".",stringsAsFactors=FALSE)
dat$BMIGroups = cut(dat$BMI,breaks=c(0,18.5,25,30,Inf),
                    labels=c("Underweight","Normal","Overweight","Obese"))
dat$SEX = factor(dat$SEX,levels=1:2,labels=c("Male","Female"))
dat$DIABETES = factor(dat$DIABETES,levels=0:1,labels=c("No Diabetes","Diabetes"))
dat$HYPERTEN = factor(dat$HYPERTEN,levels=0:1,labels=c("Normotensive","Hypertensive"))

#for the survival analysis
dat.lung = lung %>% tibble %>%
  mutate(
    sex = factor(sex,levels=1:2,labels=c("M","F")),
    ageGroup = cut(age,breaks=c(0,50,60,70,100)),
    ph.ecog = factor(ph.ecog)
  )

#for the multilevel models
imm = read_dta("https://stats.idre.ucla.edu/stat/examples/imm/imm10.dta")

dat.imm = imm %>%
  mutate(homework.fact = factor(homework))
```

This week we're going to get back to the Biostats/Epi world and cover some more complex modeling techniques, focusing on

1. Survival Analysis
2. Multilevel Models
3. ~~Generalized Estimating Equations~~

1 Survival Analysis

Survival analysis is the analysis of time to event data

- Time from surgery to infection
- Time from breast cancer diagnosis to death
- Time from entry into addiction treatment to relapse

We need three time points to perform a survival analysis

- Starting time (often 0)
- Stopping time
- Censoring - an indicator if the event occurred

We'll start with Kaplan Meier curves, then move onto Cox Proportional Hazards Regression. We'll use a built-in R dataset here called `lung` on the survival rate of patients with advanced lung cancer. See `?lung` for more details.

1.1 Kaplan-Meier

The command to fit a Kaplan-Meier curve is `survfit()`, and it uses the formula approach as with `lm()` and `glm()`. The difference here is that our outcome is a combination of either 2 or 3 variables - in this case there is a `time` variable that records the amount of time followed, and a `status` variable that records whether whether the subject died or not. We'll pass those to a `Surv()` object on the left side of the equation, and then the stratifying variables are on the right.

```
km.lung00 = survfit(Surv(time,status)~1,data=dat.lung)
km.lung00
```

```
## Call: survfit(formula = Surv(time, status) ~ 1, data = dat.lung)
##
##       n  events  median 0.95LCL 0.95UCL
##     228     165     310     285     363
```

```
summary(km.lung00)
```

```
## Call: survfit(formula = Surv(time, status) ~ 1, data = dat.lung)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     5    228       1   0.9956 0.00438       0.9871        1.000
##    11    227       3   0.9825 0.00869       0.9656        1.000
##    12    224       1   0.9781 0.00970       0.9592        0.997
##    13    223       2   0.9693 0.01142       0.9472        0.992
##    15    221       1   0.9649 0.01219       0.9413        0.989
##    26    220       1   0.9605 0.01290       0.9356        0.986
##    30    219       1   0.9561 0.01356       0.9299        0.983
##    31    218       1   0.9518 0.01419       0.9243        0.980
##    53    217       2   0.9430 0.01536       0.9134        0.974
##    54    215       1   0.9386 0.01590       0.9079        0.970
##    59    214       1   0.9342 0.01642       0.9026        0.967
##    60    213       2   0.9254 0.01740       0.8920        0.960
##    61    211       1   0.9211 0.01786       0.8867        0.957
##    62    210       1   0.9167 0.01830       0.8815        0.953
##    65    209       2   0.9079 0.01915       0.8711        0.946
##    71    207       1   0.9035 0.01955       0.8660        0.943
##    79    206       1   0.8991 0.01995       0.8609        0.939
##    81    205       2   0.8904 0.02069       0.8507        0.932
##    88    203       2   0.8816 0.02140       0.8406        0.925
```

```
##    92    201       1   0.8772 0.02174       0.8356        0.921
##    93    199       1   0.8728 0.02207       0.8306        0.917
##    95    198       2   0.8640 0.02271       0.8206        0.910
##   105    196       1   0.8596 0.02302       0.8156        0.906
##   107    194       2   0.8507 0.02362       0.8056        0.898
##   110    192       1   0.8463 0.02391       0.8007        0.894
##   116    191       1   0.8418 0.02419       0.7957        0.891
##   118    190       1   0.8374 0.02446       0.7908        0.887
##   122    189       1   0.8330 0.02473       0.7859        0.883
##   131    188       1   0.8285 0.02500       0.7810        0.879
##   132    187       2   0.8197 0.02550       0.7712        0.871
##   135    185       1   0.8153 0.02575       0.7663        0.867
##   142    184       1   0.8108 0.02598       0.7615        0.863
##   144    183       1   0.8064 0.02622       0.7566        0.859
##   145    182       2   0.7975 0.02667       0.7469        0.852
##   147    180       1   0.7931 0.02688       0.7421        0.848
##   153    179       1   0.7887 0.02710       0.7373        0.844
##   156    178       2   0.7798 0.02751       0.7277        0.836
##   163    176       3   0.7665 0.02809       0.7134        0.824
##   166    173       2   0.7577 0.02845       0.7039        0.816
##   167    171       1   0.7532 0.02863       0.6991        0.811
##   170    170       1   0.7488 0.02880       0.6944        0.807
##   175    167       1   0.7443 0.02898       0.6896        0.803
##   176    165       1   0.7398 0.02915       0.6848        0.799
##   177    164       1   0.7353 0.02932       0.6800        0.795
##   179    162       2   0.7262 0.02965       0.6704        0.787
##   180    160       1   0.7217 0.02981       0.6655        0.783
##   181    159       2   0.7126 0.03012       0.6559        0.774
##   182    157       1   0.7081 0.03027       0.6511        0.770
##   183    156       1   0.7035 0.03041       0.6464        0.766
##   186    154       1   0.6989 0.03056       0.6416        0.761
##   189    152       1   0.6943 0.03070       0.6367        0.757
##   194    149       1   0.6897 0.03085       0.6318        0.753
##   197    147       1   0.6850 0.03099       0.6269        0.749
##   199    145       1   0.6803 0.03113       0.6219        0.744
##   201    144       2   0.6708 0.03141       0.6120        0.735
##   202    142       1   0.6661 0.03154       0.6071        0.731
##   207    139       1   0.6613 0.03168       0.6020        0.726
##   208    138       1   0.6565 0.03181       0.5970        0.722
##   210    137       1   0.6517 0.03194       0.5920        0.717
##   212    135       1   0.6469 0.03206       0.5870        0.713
##   218    134       1   0.6421 0.03218       0.5820        0.708
##   222    132       1   0.6372 0.03231       0.5769        0.704
##   223    130       1   0.6323 0.03243       0.5718        0.699
##   226    126       1   0.6273 0.03256       0.5666        0.694
##   229    125       1   0.6223 0.03268       0.5614        0.690
##   230    124       1   0.6172 0.03280       0.5562        0.685
##   239    121       2   0.6070 0.03304       0.5456        0.675
##   245    117       1   0.6019 0.03316       0.5402        0.670
##   246    116       1   0.5967 0.03328       0.5349        0.666
##   267    112       1   0.5913 0.03341       0.5294        0.661
##   268    111       1   0.5860 0.03353       0.5239        0.656
##   269    110       1   0.5807 0.03364       0.5184        0.651
##   270    108       1   0.5753 0.03376       0.5128        0.645
```

```
##    283    104          1   0.5698 0.03388        0.5071         0.640
##    284    103          1   0.5642 0.03400        0.5014         0.635
##    285    101          2   0.5531 0.03424        0.4899         0.624
##    286     99          1   0.5475 0.03434        0.4841         0.619
##    288     98          1   0.5419 0.03444        0.4784         0.614
##    291     97          1   0.5363 0.03454        0.4727         0.608
##    293     94          1   0.5306 0.03464        0.4669         0.603
##    301     91          1   0.5248 0.03475        0.4609         0.597
##    303     89          1   0.5189 0.03485        0.4549         0.592
##    305     87          1   0.5129 0.03496        0.4488         0.586
##    306     86          1   0.5070 0.03506        0.4427         0.581
##    310     85          2   0.4950 0.03523        0.4306         0.569
##    320     82          1   0.4890 0.03532        0.4244         0.563
##    329     81          1   0.4830 0.03539        0.4183         0.558
##    337     79          1   0.4768 0.03547        0.4121         0.552
##    340     78          1   0.4707 0.03554        0.4060         0.546
##    345     77          1   0.4646 0.03560        0.3998         0.540
##    348     76          1   0.4585 0.03565        0.3937         0.534
##    350     75          1   0.4524 0.03569        0.3876         0.528
##    351     74          1   0.4463 0.03573        0.3815         0.522
##    353     73          2   0.4340 0.03578        0.3693         0.510
##    361     70          1   0.4278 0.03581        0.3631         0.504
##    363     69          2   0.4154 0.03583        0.3508         0.492
##    364     67          1   0.4092 0.03582        0.3447         0.486
##    371     65          2   0.3966 0.03581        0.3323         0.473
##    387     60          1   0.3900 0.03582        0.3258         0.467
##    390     59          1   0.3834 0.03582        0.3193         0.460
##    394     58          1   0.3768 0.03580        0.3128         0.454
##    426     55          1   0.3700 0.03580        0.3060         0.447
##    428     54          1   0.3631 0.03579        0.2993         0.440
##    429     53          1   0.3563 0.03576        0.2926         0.434
##    433     52          1   0.3494 0.03573        0.2860         0.427
##    442     51          1   0.3426 0.03568        0.2793         0.420
##    444     50          1   0.3357 0.03561        0.2727         0.413
##    450     48          1   0.3287 0.03555        0.2659         0.406
##    455     47          1   0.3217 0.03548        0.2592         0.399
##    457     46          1   0.3147 0.03539        0.2525         0.392
##    460     44          1   0.3076 0.03530        0.2456         0.385
##    473     43          1   0.3004 0.03520        0.2388         0.378
##    477     42          1   0.2933 0.03508        0.2320         0.371
##    519     39          1   0.2857 0.03498        0.2248         0.363
##    520     38          1   0.2782 0.03485        0.2177         0.356
##    524     37          2   0.2632 0.03455        0.2035         0.340
##    533     34          1   0.2554 0.03439        0.1962         0.333
##    550     32          1   0.2475 0.03423        0.1887         0.325
##    558     30          1   0.2392 0.03407        0.1810         0.316
##    567     28          1   0.2307 0.03391        0.1729         0.308
##    574     27          1   0.2221 0.03371        0.1650         0.299
##    583     26          1   0.2136 0.03348        0.1571         0.290
##    613     24          1   0.2047 0.03325        0.1489         0.281
##    624     23          1   0.1958 0.03297        0.1407         0.272
##    641     22          1   0.1869 0.03265        0.1327         0.263
##    643     21          1   0.1780 0.03229        0.1247         0.254
##    654     20          1   0.1691 0.03188        0.1169         0.245
```

```
##    655     19       1   0.1602 0.03142       0.1091        0.235
##    687     18       1   0.1513 0.03090       0.1014        0.226
##    689     17       1   0.1424 0.03034       0.0938        0.216
##    705     16       1   0.1335 0.02972       0.0863        0.207
##    707     15       1   0.1246 0.02904       0.0789        0.197
##    728     14       1   0.1157 0.02830       0.0716        0.187
##    731     13       1   0.1068 0.02749       0.0645        0.177
##    735     12       1   0.0979 0.02660       0.0575        0.167
##    765     10       1   0.0881 0.02568       0.0498        0.156
##    791      9       1   0.0783 0.02462       0.0423        0.145
##    814      7       1   0.0671 0.02351       0.0338        0.133
##    883      4       1   0.0503 0.02285       0.0207        0.123
```

```
km.lung01 = survfit(Surv(time,status)~sex,data=dat.lung)
km.lung01
```

```
## Call: survfit(formula = Surv(time, status) ~ sex, data = dat.lung)
##
##           n events median 0.95LCL 0.95UCL
## sex=M 138    112    270     212     310
## sex=F  90     53    426     348     550
```

```
summary(km.lung01)
```

```
## Call: survfit(formula = Surv(time, status) ~ sex, data = dat.lung)
##
##                 sex=M
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    11    138       3   0.9783  0.0124       0.9542        1.000
##    12    135       1   0.9710  0.0143       0.9434        0.999
##    13    134       2   0.9565  0.0174       0.9231        0.991
##    15    132       1   0.9493  0.0187       0.9134        0.987
##    26    131       1   0.9420  0.0199       0.9038        0.982
##    30    130       1   0.9348  0.0210       0.8945        0.977
##    31    129       1   0.9275  0.0221       0.8853        0.972
##    53    128       2   0.9130  0.0240       0.8672        0.961
##    54    126       1   0.9058  0.0249       0.8583        0.956
##    59    125       1   0.8986  0.0257       0.8496        0.950
##    60    124       1   0.8913  0.0265       0.8409        0.945
##    65    123       2   0.8768  0.0280       0.8237        0.933
##    71    121       1   0.8696  0.0287       0.8152        0.928
##    81    120       1   0.8623  0.0293       0.8067        0.922
##    88    119       2   0.8478  0.0306       0.7900        0.910
##    92    117       1   0.8406  0.0312       0.7817        0.904
##    93    116       1   0.8333  0.0317       0.7734        0.898
##    95    115       1   0.8261  0.0323       0.7652        0.892
##   105    114       1   0.8188  0.0328       0.7570        0.886
##   107    113       1   0.8116  0.0333       0.7489        0.880
##   110    112       1   0.8043  0.0338       0.7408        0.873
##   116    111       1   0.7971  0.0342       0.7328        0.867
##   118    110       1   0.7899  0.0347       0.7247        0.861
##   131    109       1   0.7826  0.0351       0.7167        0.855
##   132    108       2   0.7681  0.0359       0.7008        0.842
##   135    106       1   0.7609  0.0363       0.6929        0.835
##   142    105       1   0.7536  0.0367       0.6851        0.829
```

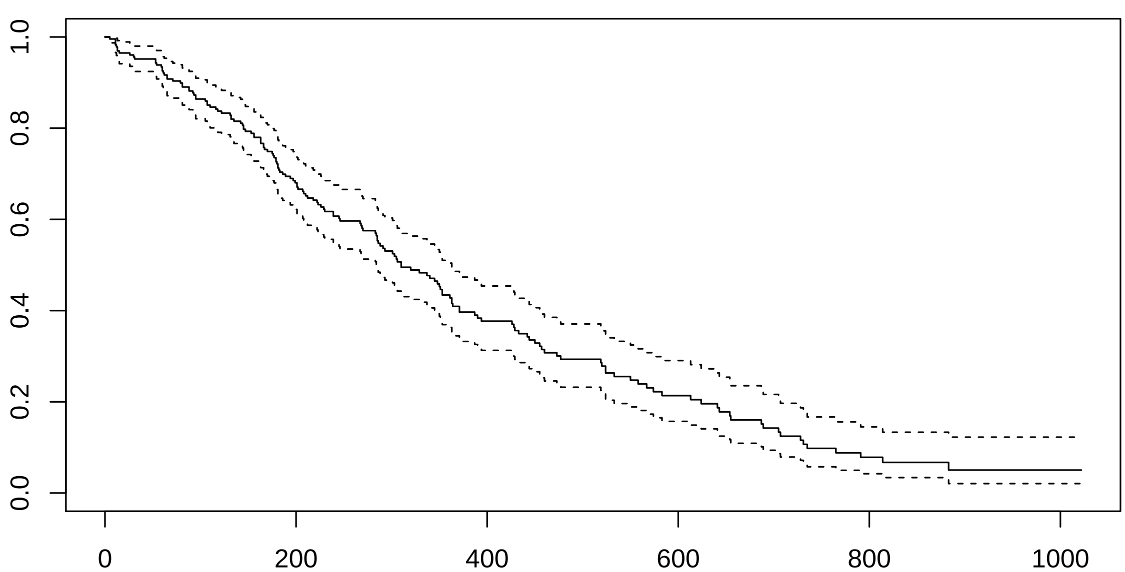
```
##   144    104        1   0.7464  0.0370       0.6772        0.823
##   147    103        1   0.7391  0.0374       0.6694        0.816
##   156    102        2   0.7246  0.0380       0.6538        0.803
##   163    100        3   0.7029  0.0389       0.6306        0.783
##   166     97        1   0.6957  0.0392       0.6230        0.777
##   170     96        1   0.6884  0.0394       0.6153        0.770
##   175     94        1   0.6811  0.0397       0.6076        0.763
##   176     93        1   0.6738  0.0399       0.5999        0.757
##   177     92        1   0.6664  0.0402       0.5922        0.750
##   179     91        2   0.6518  0.0406       0.5769        0.736
##   180     89        1   0.6445  0.0408       0.5693        0.730
##   181     88        2   0.6298  0.0412       0.5541        0.716
##   183     86        1   0.6225  0.0413       0.5466        0.709
##   189     83        1   0.6150  0.0415       0.5388        0.702
##   197     80        1   0.6073  0.0417       0.5309        0.695
##   202     78        1   0.5995  0.0419       0.5228        0.687
##   207     77        1   0.5917  0.0420       0.5148        0.680
##   210     76        1   0.5839  0.0422       0.5068        0.673
##   212     75        1   0.5762  0.0424       0.4988        0.665
##   218     74        1   0.5684  0.0425       0.4909        0.658
##   222     72        1   0.5605  0.0426       0.4829        0.651
##   223     70        1   0.5525  0.0428       0.4747        0.643
##   229     67        1   0.5442  0.0429       0.4663        0.635
##   230     66        1   0.5360  0.0431       0.4579        0.627
##   239     64        1   0.5276  0.0432       0.4494        0.619
##   246     63        1   0.5192  0.0433       0.4409        0.611
##   267     61        1   0.5107  0.0434       0.4323        0.603
##   269     60        1   0.5022  0.0435       0.4238        0.595
##   270     59        1   0.4937  0.0436       0.4152        0.587
##   283     57        1   0.4850  0.0437       0.4065        0.579
##   284     56        1   0.4764  0.0438       0.3979        0.570
##   285     54        1   0.4676  0.0438       0.3891        0.562
##   286     53        1   0.4587  0.0439       0.3803        0.553
##   288     52        1   0.4499  0.0439       0.3716        0.545
##   291     51        1   0.4411  0.0439       0.3629        0.536
##   301     48        1   0.4319  0.0440       0.3538        0.527
##   303     46        1   0.4225  0.0440       0.3445        0.518
##   306     44        1   0.4129  0.0440       0.3350        0.509
##   310     43        1   0.4033  0.0441       0.3256        0.500
##   320     42        1   0.3937  0.0440       0.3162        0.490
##   329     41        1   0.3841  0.0440       0.3069        0.481
##   337     40        1   0.3745  0.0439       0.2976        0.471
##   353     39        2   0.3553  0.0437       0.2791        0.452
##   363     37        1   0.3457  0.0436       0.2700        0.443
##   364     36        1   0.3361  0.0434       0.2609        0.433
##   371     35        1   0.3265  0.0432       0.2519        0.423
##   387     34        1   0.3169  0.0430       0.2429        0.413
##   390     33        1   0.3073  0.0428       0.2339        0.404
##   394     32        1   0.2977  0.0425       0.2250        0.394
##   428     29        1   0.2874  0.0423       0.2155        0.383
##   429     28        1   0.2771  0.0420       0.2060        0.373
##   442     27        1   0.2669  0.0417       0.1965        0.362
##   455     25        1   0.2562  0.0413       0.1868        0.351
##   457     24        1   0.2455  0.0410       0.1770        0.341
```

```
##   460     22       1   0.2344  0.0406       0.1669        0.329
##   477     21       1   0.2232  0.0402       0.1569        0.318
##   519     20       1   0.2121  0.0397       0.1469        0.306
##   524     19       1   0.2009  0.0391       0.1371        0.294
##   533     18       1   0.1897  0.0385       0.1275        0.282
##   558     17       1   0.1786  0.0378       0.1179        0.270
##   567     16       1   0.1674  0.0371       0.1085        0.258
##   574     15       1   0.1562  0.0362       0.0992        0.246
##   583     14       1   0.1451  0.0353       0.0900        0.234
##   613     13       1   0.1339  0.0343       0.0810        0.221
##   624     12       1   0.1228  0.0332       0.0722        0.209
##   643     11       1   0.1116  0.0320       0.0636        0.196
##   655     10       1   0.1004  0.0307       0.0552        0.183
##   689      9       1   0.0893  0.0293       0.0470        0.170
##   707      8       1   0.0781  0.0276       0.0390        0.156
##   791      7       1   0.0670  0.0259       0.0314        0.143
##   814      5       1   0.0536  0.0239       0.0223        0.128
##   883      3       1   0.0357  0.0216       0.0109        0.117
## 
##                 sex=F 
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     5     90       1   0.9889  0.0110       0.9675        1.000
##    60     89       1   0.9778  0.0155       0.9478        1.000
##    61     88       1   0.9667  0.0189       0.9303        1.000
##    62     87       1   0.9556  0.0217       0.9139        0.999
##    79     86       1   0.9444  0.0241       0.8983        0.993
##    81     85       1   0.9333  0.0263       0.8832        0.986
##    95     83       1   0.9221  0.0283       0.8683        0.979
##   107     81       1   0.9107  0.0301       0.8535        0.972
##   122     80       1   0.8993  0.0318       0.8390        0.964
##   145     79       2   0.8766  0.0349       0.8108        0.948
##   153     77       1   0.8652  0.0362       0.7970        0.939
##   166     76       1   0.8538  0.0375       0.7834        0.931
##   167     75       1   0.8424  0.0387       0.7699        0.922
##   182     71       1   0.8305  0.0399       0.7559        0.913
##   186     70       1   0.8187  0.0411       0.7420        0.903
##   194     68       1   0.8066  0.0422       0.7280        0.894
##   199     67       1   0.7946  0.0432       0.7142        0.884
##   201     66       2   0.7705  0.0452       0.6869        0.864
##   208     62       1   0.7581  0.0461       0.6729        0.854
##   226     59       1   0.7452  0.0471       0.6584        0.843
##   239     57       1   0.7322  0.0480       0.6438        0.833
##   245     54       1   0.7186  0.0490       0.6287        0.821
##   268     51       1   0.7045  0.0501       0.6129        0.810
##   285     47       1   0.6895  0.0512       0.5962        0.798
##   293     45       1   0.6742  0.0523       0.5791        0.785
##   305     43       1   0.6585  0.0534       0.5618        0.772
##   310     42       1   0.6428  0.0544       0.5447        0.759
##   340     39       1   0.6264  0.0554       0.5267        0.745
##   345     38       1   0.6099  0.0563       0.5089        0.731
##   348     37       1   0.5934  0.0572       0.4913        0.717
##   350     36       1   0.5769  0.0579       0.4739        0.702
##   351     35       1   0.5604  0.0586       0.4566        0.688
##   361     33       1   0.5434  0.0592       0.4390        0.673
```

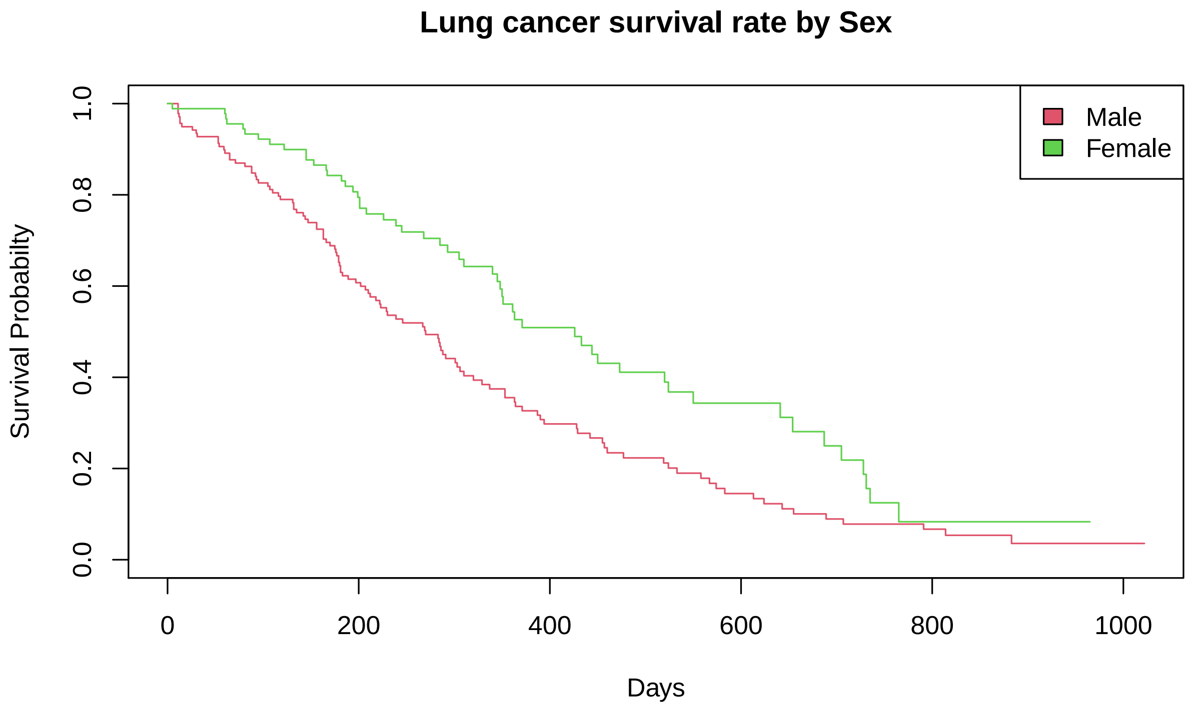
```
##    363      32        1    0.5265  0.0597        0.4215         0.658
##    371      30        1    0.5089  0.0603        0.4035         0.642
##    426      26        1    0.4893  0.0610        0.3832         0.625
##    433      25        1    0.4698  0.0617        0.3632         0.608
##    444      24        1    0.4502  0.0621        0.3435         0.590
##    450      23        1    0.4306  0.0624        0.3241         0.572
##    473      22        1    0.4110  0.0626        0.3050         0.554
##    520      19        1    0.3894  0.0629        0.2837         0.534
##    524      18        1    0.3678  0.0630        0.2628         0.515
##    550      15        1    0.3433  0.0634        0.2390         0.493
##    641      11        1    0.3121  0.0649        0.2076         0.469
##    654      10        1    0.2808  0.0655        0.1778         0.443
##    687       9        1    0.2496  0.0652        0.1496         0.417
##    705       8        1    0.2184  0.0641        0.1229         0.388
##    728       7        1    0.1872  0.0621        0.0978         0.359
##    731       6        1    0.1560  0.0590        0.0743         0.328
##    735       5        1    0.1248  0.0549        0.0527         0.295
##    765       3        1    0.0832  0.0499        0.0257         0.270
```

The printout of the model reports the basic summary stats, and the summary reports the lifetable. We can also plot the data, either with the basic `plot()` function, or with the ggplot-friendly `survminer::ggsurvplot()`. `ggsurvplot()` doesn't follow the `tidyverse` principles perfectly, but it's still a marked improvement from the base library, see this blog for a good exploration of the options.

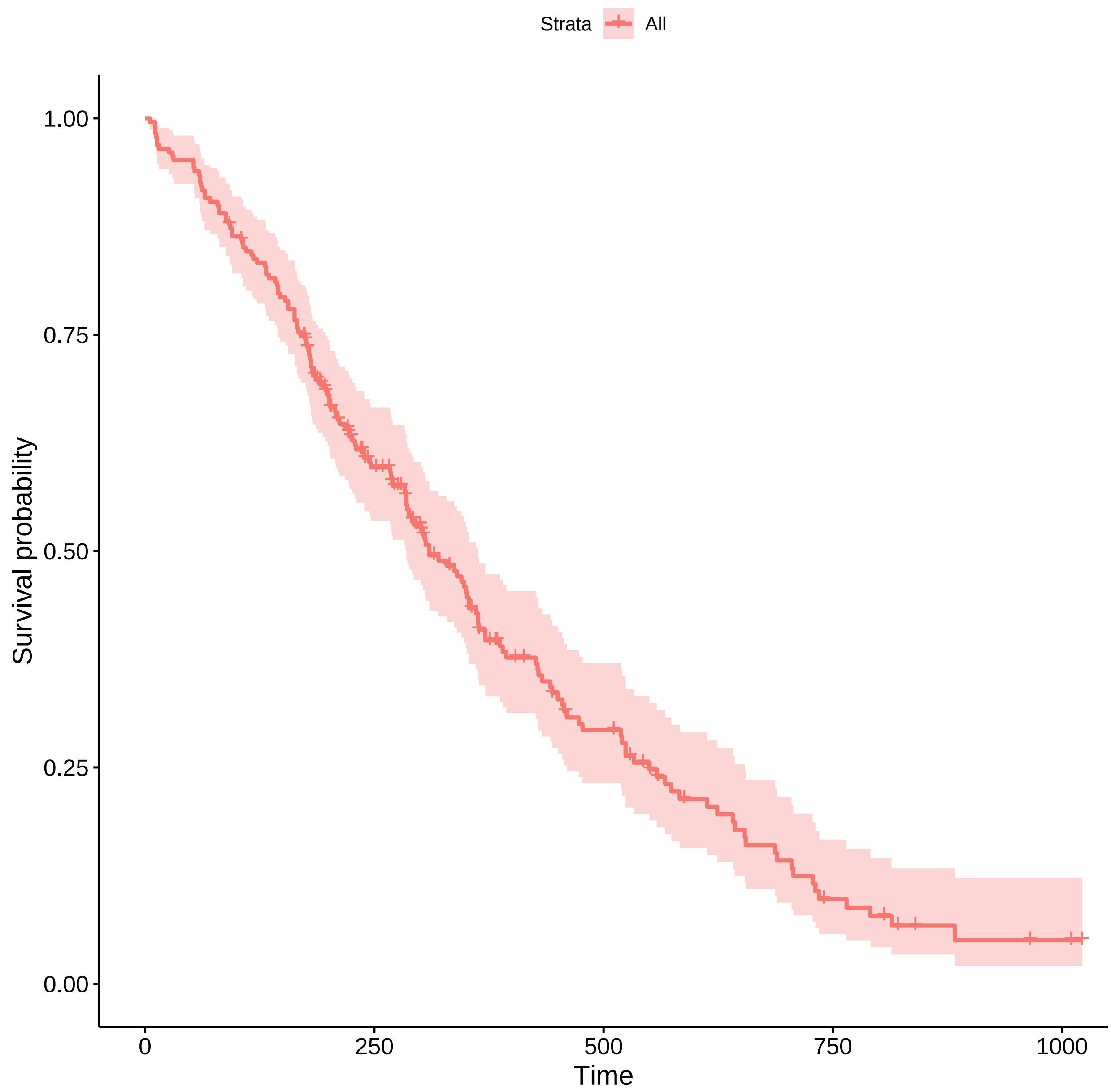
```
plot(km.lung00)
```



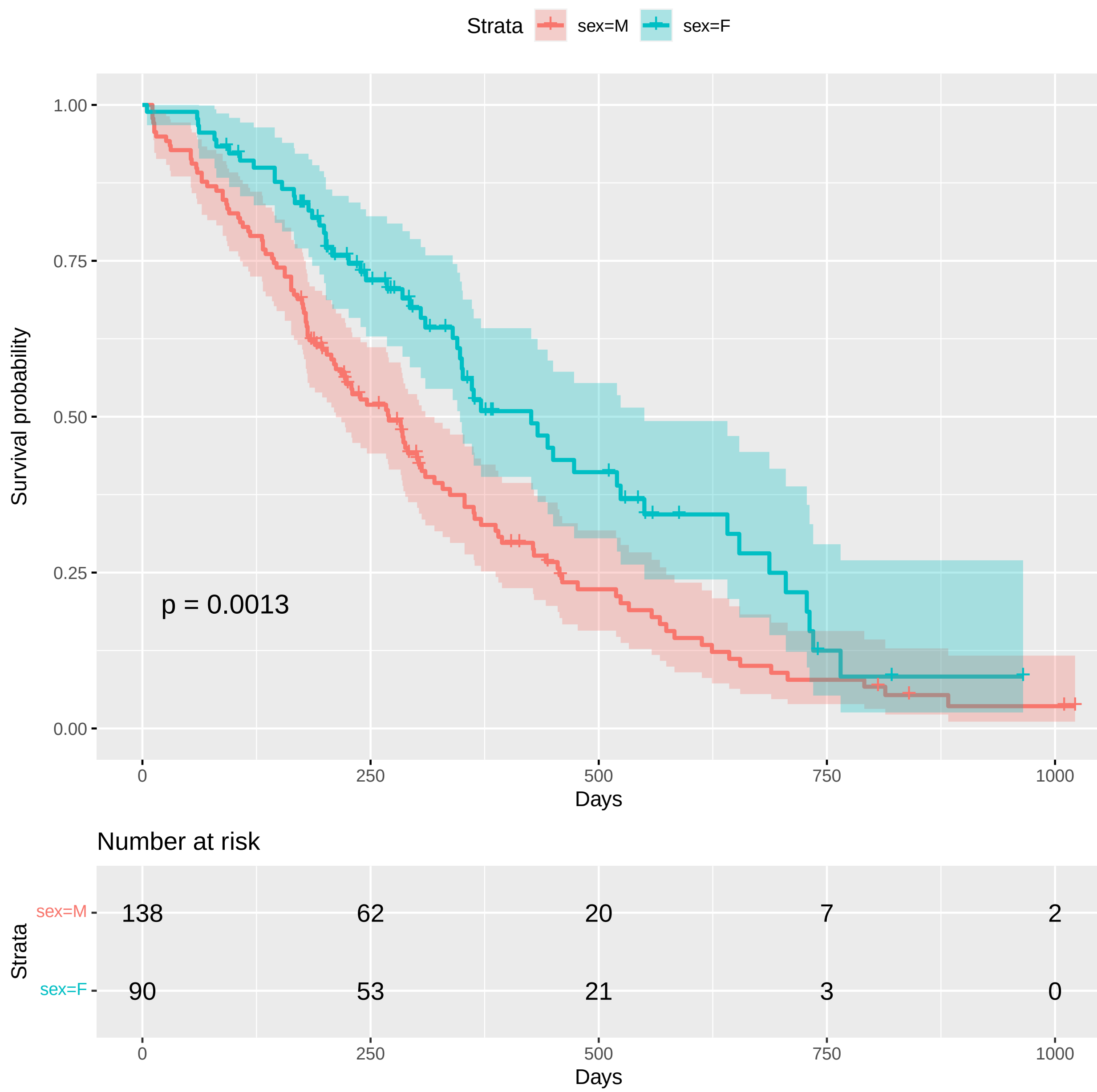

```
plot(km.lung01,col=2:3,xlab='Days',ylab='Survival Probabilty',main='Lung cancer survival rate by Sex')
legend("topright",fill=2:3,legend=c("Male","Female"))
```




```
ggsurvplot(km.lung00)
```




```
ggsurvplot(km.lung01,
           ggtheme=theme_grey(),
           conf.int=TRUE,
           pval=TRUE,
           risk.table=TRUE)+
  labs(x="Days",Strata="")
```



1.2 Cox PH Regression

We'll use the `coxph()` command to fit Cox PH regression models - they follow the same general format as `survfit()`, but produce a regression model. We'll evaluate them using the `car::Anova()` command, which will evaluate the varialbes using log-likelihood tests.

```
#simple model predicting survival with sex
mod.lung01 = coxph(Surv(time,status)~sex,data=dat.lung)
#Anova evaluates each variables
car::Anova(mod.lung01)
```

```
## Analysis of Deviance Table
##  Cox model: response is Surv(time, status)
## Terms added sequentially (first to last)
##
```

```
##       loglik  Chisq Df Pr(>|Chi|)
## NULL -749.91
## sex  -744.59 10.634  1   0.001111 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#getting the coefficient table
summary(mod.lung01)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ sex, data = dat.lung)
##
##   n= 228, number of events= 165
##
##         coef exp(coef) se(coef)      z Pr(>|z|)
## sexF -0.5310    0.5880   0.1672 -3.176  0.00149 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      exp(coef) exp(-coef) lower .95 upper .95
## sexF     0.588      1.701    0.4237     0.816
##
## Concordance= 0.579  (se = 0.021 )
## Likelihood ratio test= 10.63  on 1 df,   p=0.001
## Wald test            = 10.09  on 1 df,   p=0.001
## Score (logrank) test = 10.33  on 1 df,   p=0.001
```

```
#full model predicting survival with sex, ageGroup and ECOG score
mod.lung02 = coxph(Surv(time,status)~sex+ageGroup+ph.ecog,data=dat.lung)
car::Anova(mod.lung02)
```

```
## Analysis of Deviance Table (Type II tests)
##          LR Chisq Df Pr(>Chisq)
## sex       11.6240  1  0.0006511 ***
## ageGroup   3.4347  3  0.3293293
## ph.ecog   18.0770  3  0.0004240 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod.lung02)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ sex + ageGroup + ph.ecog,
##     data = dat.lung)
##
##   n= 227, number of events= 164
##    (1 observation deleted due to missingness)
##
##                     coef exp(coef) se(coef)      z Pr(>|z|)
## sexF             -0.5606    0.5709   0.1688 -3.321 0.000897 ***
## ageGroup(50,60]   0.2478    1.2812   0.2929  0.846 0.397446
## ageGroup(60,70]   0.1207    1.1283   0.2821  0.428 0.668697
## ageGroup(70,100]  0.4590    1.5825   0.3009  1.525 0.127136
## ph.ecog1          0.4309    1.5386   0.2015  2.138 0.032507 *
## ph.ecog2          0.9485    2.5819   0.2341  4.053 5.07e-05 ***
## ph.ecog3          2.1541    8.6205   1.0343  2.083 0.037284 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                   exp(coef) exp(-coef) lower .95 upper .95
## sexF                 0.5709     1.7517    0.4101    0.7947
## ageGroup(50,60]      1.2812     0.7805    0.7217    2.2747
## ageGroup(60,70]      1.1283     0.8863    0.6491    1.9614
## ageGroup(70,100]     1.5825     0.6319    0.8775    2.8541
## ph.ecog1             1.5386     0.6499    1.0366    2.2838
## ph.ecog2             2.5819     0.3873    1.6320    4.0847
## ph.ecog3             8.6205     0.1160    1.1353   65.4566
##
## Concordance= 0.642  (se = 0.025 )
## Likelihood ratio test= 32.94  on 7 df,   p=3e-05
## Wald test            = 33.07  on 7 df,   p=3e-05
## Score (logrank) test = 35.36  on 7 df,   p=1e-05
```

That's good, but we need to get this in production-quality, so we'll format using our `tidyverse` skills.

```
#in multiple steps like a sane person
#a) extract the coefficients
coef = summary(mod.lung02)$coefficients %>% as.data.frame() %>%
  rownames_to_column()
#b) extract the confidence intervals
ci = summary(mod.lung02)$conf.int %>% as.data.frame %>%
              rownames_to_column()
#c) join the two, and reduce to useful columns
out01 = left_join(coef,ci) %>%
  select(rowname,'exp(coef)','lower .95','upper .95','Pr(>|z|)') %>%
  rename(RR=`exp(coef)`,
         variable = rowname)
#d) print the columns
out01 %>% kable(digits=c(0,3,2,2,4)) %>% kable_styling()
```

variable	RR	lower .95	upper .95	Pr(>|z|)
sexF	0.571	0.41	0.79	0.0009
ageGroup(50,60]	1.281	0.72	2.27	0.3974
ageGroup(60,70]	1.128	0.65	1.96	0.6687
ageGroup(70,100]	1.583	0.88	2.85	0.1271
ph.ecog1	1.539	1.04	2.28	0.0325
ph.ecog2	2.582	1.63	4.08	0.0001
ph.ecog3	8.620	1.14	65.46	0.0373

```
#OR
#do it all at once
summary(mod.lung02)$coefficients %>% as.data.frame %>%
  rownames_to_column() %>%
  left_join(summary(mod.lung02)$conf.int %>% as.data.frame %>%
              rownames_to_column()
              ) %>%
  select(rowname,'exp(coef)','lower .95','upper .95','Pr(>|z|)') %>%
  rename(RR=`exp(coef)`,
         variable = rowname) %>%
  kable(digits=c(0,3,2,2,4)) %>% kable_styling() %>%
```

```
pack_rows("Sex",1,1)%>%
pack_rows("Age Group",2,4)%>%
pack_rows("ECOG Score",5,7)
```

variable	RR	lower .95	upper .95	Pr(>|z|)
Sex				
sexF	0.571	0.41	0.79	0.0009
Age Group				
ageGroup(50,60]	1.281	0.72	2.27	0.3974
ageGroup(60,70]	1.128	0.65	1.96	0.6687
ageGroup(70,100]	1.583	0.88	2.85	0.1271
ECOG Score				
ph.ecog1	1.539	1.04	2.28	0.0325
ph.ecog2	2.582	1.63	4.08	0.0001
ph.ecog3	8.620	1.14	65.46	0.0373

2 Multilevel Models

There are several ways to fit multilevel models in R, but the most common are with the `lmer()` and `glmer()` functions from the `lme4` library. I'll focus on their use for multilevel models with random intercepts and/or random slopes - for longitudinal data I prefer `gee()`, but if you like mixed-effects for longitudinal data you can use `lmer()`.

For this section I'm going to follow along with the UCLA Stats page on multilevel modeling, available here.

3 Syntax

The traditional syntax is `y~x` for a simple model, or `y~x1+x2+...+xk` for a multiple model. With multilevel modeling we want to be able to adjust for random slopes and/or random intercepts, which we control with additional random components in brackets `()` at the end of the equation. For a **random intercept** the format will be

`y ~ x1+x2+x3+(1|group)`

The `1` in the random component refers to the intercept, in a similar fashion to a regression model `y~1` that fits y predicted with only an intercept. This fits a model that predicts `y` with `x1, x2, x3`, but adds a random intercept for each level of `group`.

If we also want random slopes the format would be

`y ~ x1+x2+x3+(x2|group)`

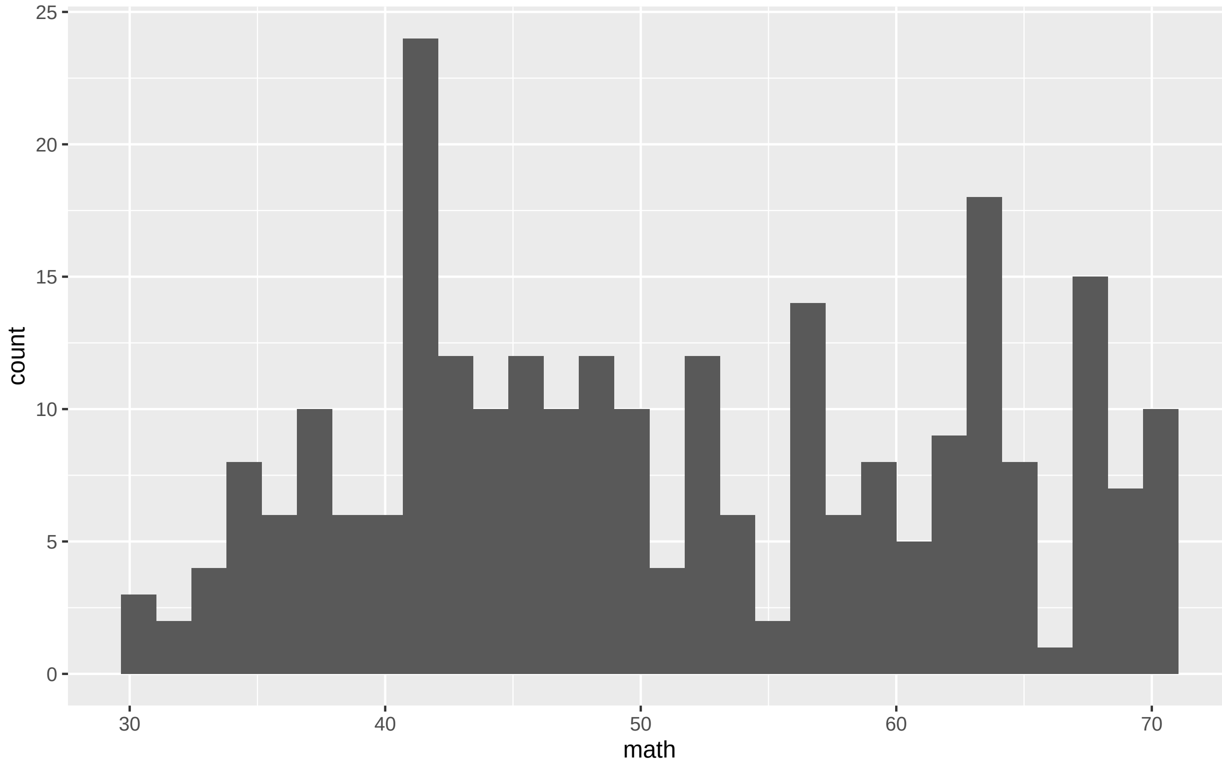
In this equation we're adding a random slope for the variable `x2`, while also including a random intercept by default. Note that `x2` doesn't need to be in both variable lists, that was our choice for this model.

3.1 `lmer` example from UCLA stats

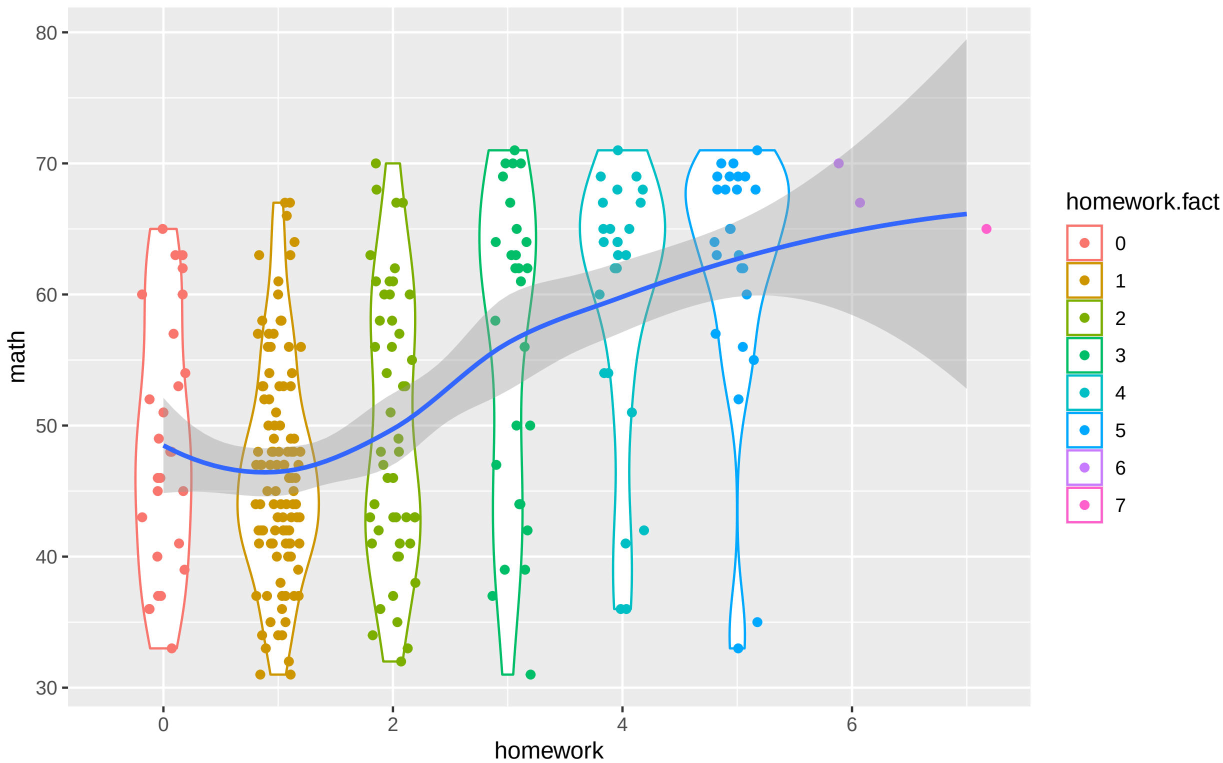
I'm having a hard time finding the provenance of the `imm10` dataset - it's from the UCLA Stats textbook *Introduction to Multilevel Modeling*, but I can't find a help file or description of it, so I'm going to infer most of it.

It has 260 subjects from 10 different schools - we're going to focus on their math scores as a function of their dedication to homework.

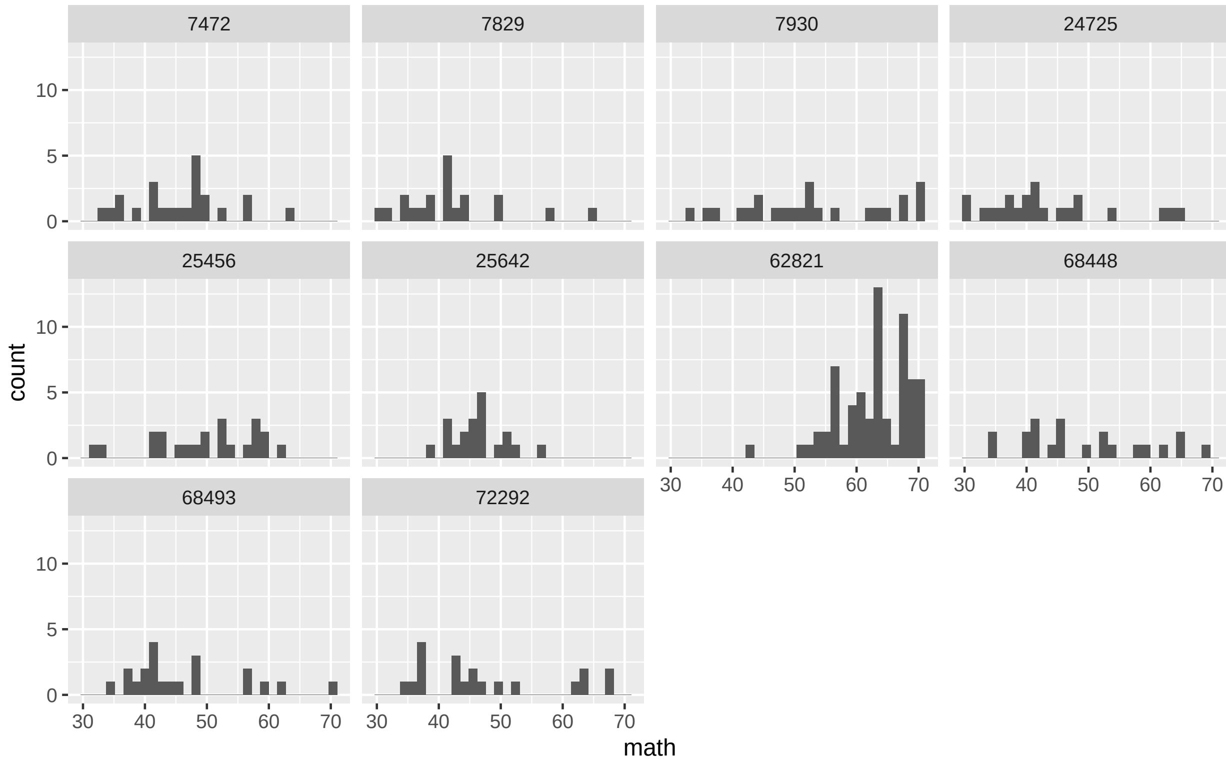
```
dat.imm %>%
  ggplot(aes(x=math))+
  geom_histogram()
```



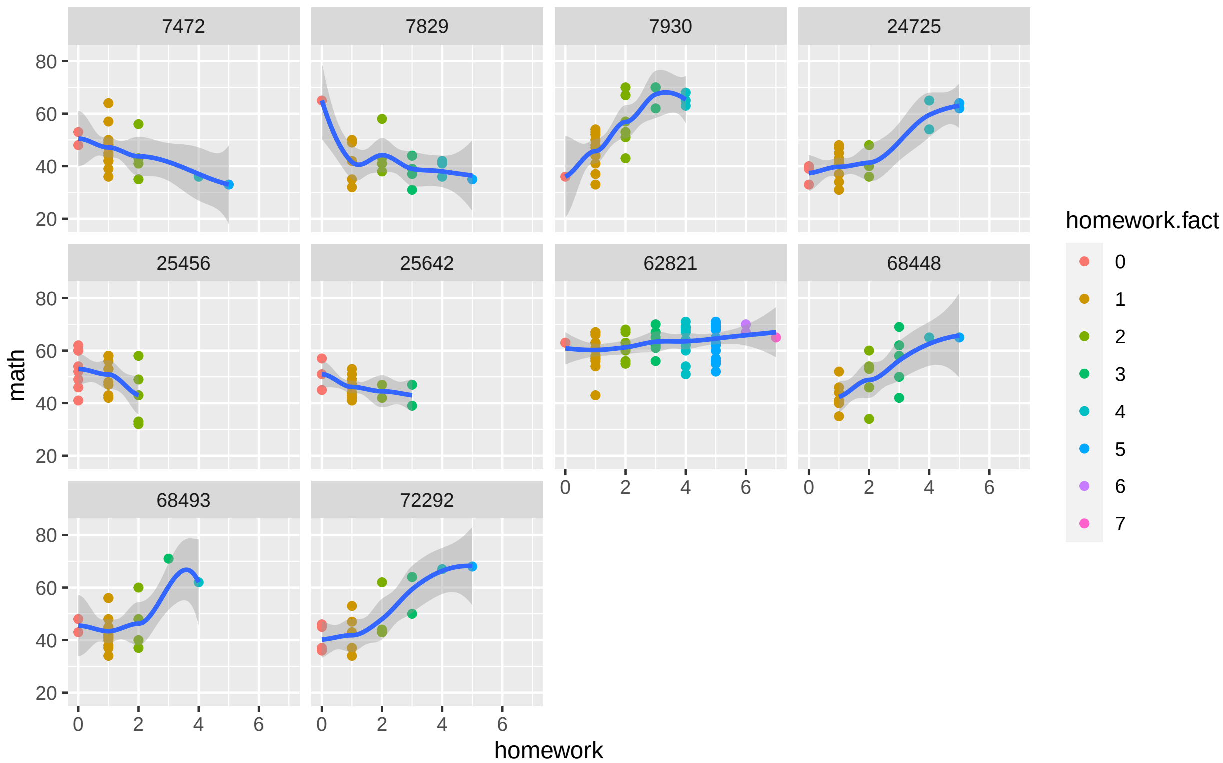

```
#geom_jitter is like geom_point, but can scatter things a bit randomly, useful for overlapping datapoin
dat.imm %>%
  ggplot(aes(x=homework,y=math))+
  geom_violin(aes(group=homework,colour=homework.fact))+
#  geom_point(aes(colour=homework.fact))+
  geom_jitter(aes(colour=homework.fact),width=0.2,height=0)+
  geom_smooth()
```




```
dat.imm %>%
  ggplot(aes(x=math))+
  geom_histogram()+
  facet_wrap(~schid)
```




```
dat.imm %>%
  ggplot(aes(x=homework,y=math))+
  geom_point(aes(colour=homework.fact))+
  geom_smooth()+
  facet_wrap(~schid)
```



The graphics demonstrate a relationship between math score and homework, but probably significant variation between schools. We'll start with individual linear regression models.

```
lin.mods = by(dat.imm,dat.imm$schid,function(x){lm(math~homework,data=x)})
```

The `by` function stratifies a dataset by a variable and performs a function on each stratified dataset - in this case I fit a simple linear regression predict math score with homework for each school individually. I want to see how the models differ.

```
t(sapply(lin.mods,coef))
```

```
##        (Intercept)  homework
## 7472      50.68354 -3.553797
## 7829      49.01229 -2.920123
## 7930      38.75000  7.909091
## 24725     34.39382  5.592664
## 25456     53.93863 -4.718412
## 25642     49.25896 -2.486056
## 62821     59.21022  1.094640
## 68448     36.05535  6.496310
## 68493     38.52000  5.860000
## 72292     37.71392  6.335052
```

These numbers confirm what our last plot showed - that the relationship between math and homework does not seem to be consistent across schools.

Based on the differences in both intercept and slope values, we'll fit a mixed effects model with a random slope for homework per school (and a random intercept per school).

```
mod.lme01 = lmer(math~homework+(homework|schid),data=dat.imm)
mod.lme01
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: math ~ homework + (homework | schid)
##    Data: dat.imm
## REML criterion at convergence: 1763.954
## Random effects:
##  Groups   Name        Std.Dev. Corr
##  schid    (Intercept) 8.325
##           homework    4.738    -0.81
##  Residual             6.563
## Number of obs: 260, groups:  schid, 10
## Fixed Effects:
## (Intercept)     homework
##       44.77         2.04
```

```
car::Anova(mod.lme01)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: math
##           Chisq Df Pr(>Chisq)
## homework 1.7236  1     0.1892
```

```
summary(mod.lme01)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: math ~ homework + (homework | schid)
##    Data: dat.imm
##
## REML criterion at convergence: 1764
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5110 -0.5357  0.0175  0.6121  2.5708
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  schid    (Intercept) 69.31    8.325
##           homework    22.45    4.738    -0.81
##  Residual             43.07    6.563
## Number of obs: 260, groups:  schid, 10
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   44.771      2.744  16.318
## homework       2.040      1.554   1.313
##
## Correlation of Fixed Effects:
##          (Intr)
## homework -0.804
```

And we can expand the model to other fixed effects as well (including school-level effects)

```
mod.lme02 = lmer(math~homework+public+(homework|schid),data=dat.imm)
mod.lme02
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: math ~ homework + public + (homework | schid)
```

```
##    Data: dat.imm
## REML criterion at convergence: 1743.966
## Random effects:
##  Groups   Name        Std.Dev. Corr
##  schid    (Intercept) 6.771
##           homework    4.895    -0.97
##  Residual             6.554
## Number of obs: 260, groups:  schid, 10
## Fixed Effects:
## (Intercept)     homework       public
##      58.041        1.952      -14.661
```

```
car::Anova(mod.lme02)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: math
##           Chisq Df Pr(>Chisq)
## homework  1.491  1     0.2221
## public   48.403  1   3.47e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod.lme02)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: math ~ homework + public + (homework | schid)
##    Data: dat.imm
##
## REML criterion at convergence: 1744
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.63743 -0.56210 -0.05468  0.64216  2.58576
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  schid    (Intercept) 45.85    6.771
##           homework    23.96    4.895    -0.97
##  Residual             42.96    6.554
## Number of obs: 260, groups:  schid, 10
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   58.041      2.943  19.721
## homework       1.952      1.599   1.221
## public       -14.661      2.107  -6.957
##
## Correlation of Fixed Effects:
##          (Intr) homwrk
## homework -0.741
## public   -0.636  0.008
```

```
mod.lme03 = lmer(math~homework*public+(homework|schid),data=dat.imm)
mod.lme03
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: math ~ homework * public + (homework | schid)
##    Data: dat.imm
## REML criterion at convergence: 1738.737
## Random effects:
##  Groups   Name        Std.Dev. Corr
##  schid    (Intercept) 7.200
##           homework    5.222    -0.98
##  Residual             6.554
## Number of obs: 260, groups:  schid, 10
## Fixed Effects:
##    (Intercept)         homework           public  homework:public
##        59.2102           1.0946         -15.9657           0.9512
```

```
car::Anova(mod.lme03)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: math
##                   Chisq Df Pr(>Chisq)
## homework         1.3112  1     0.2522
## public          48.8206  1  2.805e-12 ***
## homework:public  0.0295  1     0.8637
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod.lme03)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: math ~ homework * public + (homework | schid)
##    Data: dat.imm
##
## REML criterion at convergence: 1738.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.64023 -0.56436 -0.05373  0.65209  2.58438
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  schid    (Intercept) 51.84    7.200
##           homework    27.27    5.222    -0.98
##  Residual             42.96    6.554
## Number of obs: 260, groups:  schid, 10
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)      59.2102     7.4074   7.993
## homework          1.0946     5.2432   0.209
## public          -15.9657     7.8292  -2.039
## homework:public   0.9512     5.5424   0.172
##
## Correlation of Fixed Effects:
##             (Intr) homwrk public
## homework    -0.964
```

```
## public      -0.946  0.912
## homwrk:pblc  0.912 -0.946 -0.963
```

4 Breakout Activity

1. There's another built-in dataset called `veteran` that reports the results of a trial comparing two treatment regimens for people with lung cancer. Use the dataset to
   - Draw a Kaplan-Meier curve survival with treatment
   - Fit a Cox PH model predicting survival with treatment, cancer type and age
2. For mixed effects model there's a library called `mlmRev` that has a dataset called `Exam` - it records the exam scores for 4059 students from 65 schools in Inner London. I want you to build a model to predict their normalized exam score (`normexam`) based on the available data, see `?Exam` after you install and load the library.