S632 HW1

Erik Parker

January 18th, 2018

1. ALR 10.2: Use the data file *Highway*

```
rm(list = ls())

library(alr4)
library(leaps)

highway <- Highway
```

10.2.1: For the highway data, verify the forward selection and backward elimination subsets that are given in section 10.2.2

```
fs <- lm(log(rate) ~ log(len), data = highway)
bs <- lm(log(rate) ~ ., data = highway)

highway$sigs1 = with(highway, (sigs * len + 1)/len)
f = ~log(len) + shld + log(adt) + log(trks) + lane + slim + lwid + itg + log(sigs1) +
    acpt + htype

m.fwd = step(fs, scope = f, direction = "forward")
```

```
## Start:  AIC=-72.51
## log(rate) ~ log(len)
##
##              Df Sum of Sq    RSS     AIC
## + slim        1   2.54718 2.9366 -94.866
## + acpt        1   2.10148 3.3823 -89.355
## + shld        1   1.70693 3.7769 -85.052
## + log(sigs1)  1   0.96128 4.5225 -78.025
## + htype       3   1.33997 4.1438 -77.436
## + log(trks)   1   0.72812 4.7557 -76.065
## + log(adt)    1   0.42857 5.0552 -73.682
## <none>                    5.4838 -72.509
## + lane        1   0.26267 5.2211 -72.423
## + itg         1   0.21704 5.2667 -72.084
## + lwid        1   0.18502 5.2988 -71.847
##
## Step:  AIC=-94.87
## log(rate) ~ log(len) + slim
##
##              Df Sum of Sq    RSS     AIC
## + acpt        1   0.28844 2.6482 -96.898
## + log(trks)   1   0.26317 2.6734 -96.528
## <none>                    2.9366 -94.866
## + log(sigs1)  1   0.14671 2.7899 -94.865
## + htype       3   0.33646 2.6002 -93.612
```

```
## + shld         1   0.03265 2.9040 -93.302
## + log(adt)     1   0.02563 2.9110 -93.208
## + lwid         1   0.01664 2.9200 -93.088
## + lane         1   0.00343 2.9332 -92.912
## + itg          1   0.00265 2.9340 -92.901
##
## Step:  AIC=-96.9
## log(rate) ~ log(len) + slim + acpt
##
##               Df Sum of Sq    RSS     AIC
## + log(trks)    1  0.172940 2.4752 -97.532
## <none>                     2.6482 -96.898
## + log(sigs1)   1  0.120061 2.5281 -96.708
## + shld         1  0.034595 2.6136 -95.411
## + log(adt)     1  0.015190 2.6330 -95.122
## + lane         1  0.014872 2.6333 -95.118
## + itg          1  0.013501 2.6347 -95.097
## + lwid         1  0.012646 2.6355 -95.085
## + htype        3  0.217478 2.4307 -94.240
##
## Step:  AIC=-97.53
## log(rate) ~ log(len) + slim + acpt + log(trks)
##
##               Df Sum of Sq    RSS     AIC
## <none>                     2.4752 -97.532
## + shld         1  0.065299 2.4099 -96.575
## + log(sigs1)   1  0.050568 2.4247 -96.337
## + log(adt)     1  0.031220 2.4440 -96.027
## + htype        3  0.259505 2.2157 -95.851
## + lwid         1  0.019009 2.4562 -95.833
## + itg          1  0.010964 2.4643 -95.705
## + lane         1  0.003299 2.4719 -95.584
```

```r
m1 = update(fs, f)

m.bck = step(m1, scope = list(lower = ~log(len), upper = m1), direction = "backward")
```

```
## Start:  AIC=-94.2
## log(rate) ~ log(len) + shld + log(adt) + log(trks) + lane + slim +
##     lwid + itg + log(sigs1) + acpt + htype
##
##               Df Sum of Sq    RSS     AIC
## - shld         1   0.00052 1.6999 -96.188
## - itg          1   0.00147 1.7008 -96.166
## - lane         1   0.00259 1.7019 -96.140
## - lwid         1   0.00644 1.7058 -96.052
## - acpt         1   0.03790 1.7372 -95.339
## - log(trks)    1   0.04613 1.7455 -95.155
## <none>                     1.6993 -94.199
## - htype        3   0.30045 1.9998 -93.850
## - log(adt)     1   0.12981 1.8292 -93.329
## - slim         1   0.17897 1.8783 -92.294
## - log(sigs1)   1   0.44263 2.1420 -87.172
##
## Step:  AIC=-96.19
```

```
## log(rate) ~ log(len) + log(adt) + log(trks) + lane + slim + lwid +
##     itg + log(sigs1) + acpt + htype
##
##              Df Sum of Sq    RSS     AIC
## - itg         1   0.00133 1.7012 -98.157
## - lane        1   0.00274 1.7026 -98.125
## - lwid        1   0.00715 1.7070 -98.024
## - acpt        1   0.04678 1.7466 -97.129
## - log(trks)   1   0.05564 1.7555 -96.932
## <none>                    1.6999 -96.188
## - htype       3   0.32844 2.0283 -95.298
## - log(adt)    1   0.13653 1.8364 -95.175
## - slim        1   0.34049 2.0404 -91.067
## - log(sigs1)  1   0.48141 2.1813 -88.463
##
## Step:  AIC=-98.16
## log(rate) ~ log(len) + log(adt) + log(trks) + lane + slim + lwid +
##     log(sigs1) + acpt + htype
##
##              Df Sum of Sq    RSS      AIC
## - lane        1   0.00248 1.7037 -100.100
## - lwid        1   0.00670 1.7079 -100.004
## - acpt        1   0.04553 1.7467  -99.127
## - log(trks)   1   0.05678 1.7580  -98.877
## <none>                    1.7012  -98.157
## - log(adt)    1   0.15520 1.8564  -96.752
## - htype       3   0.55968 2.2609  -93.065
## - slim        1   0.37950 2.0807  -92.304
## - log(sigs1)  1   0.48116 2.1823  -90.443
##
## Step:  AIC=-100.1
## log(rate) ~ log(len) + log(adt) + log(trks) + slim + lwid + log(sigs1) +
##     acpt + htype
##
##              Df Sum of Sq    RSS      AIC
## - lwid        1   0.00790 1.7116 -101.920
## - acpt        1   0.04694 1.7506 -101.040
## - log(trks)   1   0.05483 1.7585 -100.865
## <none>                    1.7037 -100.100
## - log(adt)    1   0.18342 1.8871  -98.113
## - slim        1   0.38450 2.0882  -94.164
## - htype       3   0.61293 2.3166  -94.115
## - log(sigs1)  1   0.48936 2.1930  -92.253
##
## Step:  AIC=-101.92
## log(rate) ~ log(len) + log(adt) + log(trks) + slim + log(sigs1) +
##     acpt + htype
##
##              Df Sum of Sq    RSS      AIC
## - acpt        1   0.05018 1.7617 -102.793
## - log(trks)   1   0.06194 1.7735 -102.534
## <none>                    1.7116 -101.920
## - log(adt)    1   0.17584 1.8874 -100.106
## - slim        1   0.38280 2.0944  -96.048
```

```
## - htype       3   0.61149 2.3230  -96.007
## - log(sigs1)  1   0.49474 2.2063  -94.018
##
## Step:  AIC=-102.79
## log(rate) ~ log(len) + log(adt) + log(trks) + slim + log(sigs1) +
##     htype
##
##              Df Sum of Sq    RSS      AIC
## - log(trks)   1   0.06865 1.8304 -103.302
## <none>                    1.7617 -102.793
## - log(adt)    1   0.14925 1.9110 -101.622
## - htype       3   0.72689 2.4886  -95.321
## - log(sigs1)  1   0.55723 2.3190  -94.075
## - slim        1   0.57997 2.3417  -93.695
##
## Step:  AIC=-103.3
## log(rate) ~ log(len) + log(adt) + slim + log(sigs1) + htype
##
##              Df Sum of Sq    RSS      AIC
## <none>                    1.8304 -103.302
## - log(adt)    1   0.13847 1.9689 -102.458
## - htype       3   0.80988 2.6403  -95.015
## - slim        1   0.55707 2.3875  -94.940
## - log(sigs1)  1   0.75127 2.5817  -91.890
```

This procedure successfully verified the data presented in section 10.2.2 of ALR4.

10.2.2: Use as response *log(rate X len)* and treat *lwid* as the focal regressor. Use both forward selection and backward elimination to assess the importance of *lwid.*

```r
m.lwid <- lm(log(rate * len) ~ lwid, data = highway)

f2 = ~shld + log(adt) + log(trks) + lane + slim + itg + log(sigs1) + acpt +
    htype + lwid

m.upper <- update(m.lwid, f2)

m.sw.up = step(m.lwid, scope = f2)
```

```
## Start:  AIC=-54.06
## log(rate * len) ~ lwid
##
##              Df Sum of Sq    RSS     AIC
## + shld        1   2.36799 6.4335 -64.280
## + log(adt)    1   1.57348 7.2280 -59.738
## + htype       3   1.66818 7.1333 -56.253
## + lane        1   0.86973 7.9318 -56.115
## + slim        1   0.79623 8.0053 -55.755
## + itg         1   0.70020 8.1013 -55.290
## + acpt        1   0.45639 8.3451 -54.134
## <none>                    8.8015 -54.057
## + log(sigs1)  1   0.08316 8.7183 -52.427
## + log(trks)   1   0.02384 8.7776 -52.163
## - lwid        1   1.03754 9.8390 -51.711
##
```

```
## Step:  AIC=-64.28
## log(rate * len) ~ lwid + shld
##
##               Df Sum of Sq    RSS     AIC
## <none>                     6.4335 -64.280
## + log(adt)     1   0.25691 6.1766 -63.869
## + log(sigs1)   1   0.10586 6.3276 -62.927
## + itg          1   0.07272 6.3608 -62.723
## + slim         1   0.06354 6.3700 -62.667
## + lane         1   0.04292 6.3906 -62.541
## + log(trks)    1   0.02366 6.4098 -62.423
## + acpt         1   0.00038 6.4331 -62.282
## - lwid         1   1.17434 7.6078 -59.741
## + htype        3   0.22480 6.2087 -59.667
## - shld         1   2.36799 8.8015 -54.057
```

```r
m.sw.down = step(m.upper, scope = list(lower = ~lwid, upper = m.upper), direction = "both")
```

```
## Start:  AIC=-47.69
## log(rate * len) ~ shld + log(adt) + log(trks) + lane + slim +
##     itg + log(sigs1) + acpt + htype + lwid
##
##               Df Sum of Sq    RSS     AIC
## - htype        3   0.18628 6.0812 -52.476
## - log(trks)    1   0.00007 5.8950 -49.689
## - slim         1   0.00200 5.8969 -49.676
## - lane         1   0.00242 5.8973 -49.673
## - log(sigs1)   1   0.00521 5.9001 -49.655
## - itg          1   0.04647 5.9414 -49.383
## - acpt         1   0.04806 5.9430 -49.373
## - log(adt)     1   0.08268 5.9776 -49.146
## <none>                     5.8949 -47.689
## - shld         1   0.35127 6.2462 -47.432
##
## Step:  AIC=-52.48
## log(rate * len) ~ shld + log(adt) + log(trks) + lane + slim +
##     itg + log(sigs1) + acpt + lwid
##
##               Df Sum of Sq    RSS     AIC
## - log(sigs1)   1   0.00066 6.0818 -54.472
## - itg          1   0.00112 6.0823 -54.469
## - log(trks)    1   0.00377 6.0850 -54.452
## - slim         1   0.00860 6.0898 -54.421
## - lane         1   0.03960 6.1208 -54.223
## - acpt         1   0.05202 6.1332 -54.144
## - log(adt)     1   0.12314 6.2043 -53.694
## <none>                     6.0812 -52.476
## - shld         1   0.46029 6.5415 -51.631
## + htype        3   0.18628 5.8949 -47.689
##
## Step:  AIC=-54.47
## log(rate * len) ~ shld + log(adt) + log(trks) + lane + slim +
##     itg + acpt + lwid
##
##               Df Sum of Sq    RSS     AIC
```

```
## - itg         1   0.00278 6.0846 -56.454
## - log(trks)   1   0.00575 6.0876 -56.435
## - slim        1   0.01142 6.0933 -56.399
## - lane        1   0.03905 6.1209 -56.222
## - acpt        1   0.05149 6.1333 -56.143
## - log(adt)    1   0.18006 6.2619 -55.334
## <none>                    6.0818 -54.472
## - shld        1   0.45999 6.5418 -53.628
## + log(sigs1)  1   0.00066 6.0812 -52.476
## + htype       3   0.18173 5.9001 -49.655
##
## Step:  AIC=-56.45
## log(rate * len) ~ shld + log(adt) + log(trks) + lane + slim +
##     acpt + lwid
##
##              Df Sum of Sq    RSS     AIC
## - log(trks)   1   0.00622 6.0908 -58.414
## - slim        1   0.01385 6.0985 -58.365
## - lane        1   0.04755 6.1322 -58.151
## - acpt        1   0.05046 6.1351 -58.132
## - log(adt)    1   0.21226 6.2969 -57.117
## <none>                    6.0846 -56.454
## - shld        1   0.50076 6.5854 -55.370
## + itg         1   0.00278 6.0818 -54.472
## + log(sigs1)  1   0.00232 6.0823 -54.469
## + htype       3   0.13472 5.9499 -51.327
##
## Step:  AIC=-58.41
## log(rate * len) ~ shld + log(adt) + lane + slim + acpt + lwid
##
##              Df Sum of Sq    RSS     AIC
## - slim        1   0.02034 6.1112 -60.284
## - lane        1   0.04412 6.1350 -60.133
## - acpt        1   0.04472 6.1356 -60.129
## - log(adt)    1   0.21095 6.3018 -59.086
## <none>                    6.0908 -58.414
## - shld        1   0.55605 6.6469 -57.007
## + log(trks)   1   0.00622 6.0846 -56.454
## + log(sigs1)  1   0.00523 6.0856 -56.448
## + itg         1   0.00325 6.0876 -56.435
## + htype       3   0.14022 5.9506 -53.323
##
## Step:  AIC=-60.28
## log(rate * len) ~ shld + log(adt) + lane + acpt + lwid
##
##              Df Sum of Sq    RSS     AIC
## - acpt        1   0.02530 6.1365 -62.123
## - lane        1   0.05111 6.1623 -61.959
## - log(adt)    1   0.27897 6.3902 -60.543
## <none>                    6.1112 -60.284
## + slim        1   0.02034 6.0908 -58.414
## + log(sigs1)  1   0.01590 6.0953 -58.386
## + log(trks)   1   0.01272 6.0985 -58.365
## + itg         1   0.00693 6.1043 -58.328
```

```
## - shld        1   0.75088 6.8621 -57.765
## + htype       3   0.15554 5.9556 -55.290
##
## Step:  AIC=-62.12
## log(rate * len) ~ shld + log(adt) + lane + lwid
##
##              Df Sum of Sq    RSS     AIC
## - lane        1   0.04011 6.1766 -63.869
## - log(adt)    1   0.25409 6.3906 -62.541
## <none>                    6.1365 -62.123
## + acpt        1   0.02530 6.1112 -60.284
## + itg         1   0.00248 6.1340 -60.139
## + log(sigs1)  1   0.00223 6.1343 -60.137
## + log(trks)   1   0.00109 6.1354 -60.130
## + slim        1   0.00092 6.1356 -60.129
## - shld        1   1.08816 7.2246 -57.756
## + htype       3   0.09768 6.0388 -56.749
##
## Step:  AIC=-63.87
## log(rate * len) ~ shld + log(adt) + lwid
##
##              Df Sum of Sq    RSS     AIC
## - log(adt)    1   0.25691 6.4335 -64.280
## <none>                    6.1766 -63.869
## + lane        1   0.04011 6.1365 -62.123
## + acpt        1   0.01429 6.1623 -61.959
## + itg         1   0.01141 6.1652 -61.941
## + slim        1   0.00457 6.1720 -61.898
## + log(sigs1)  1   0.00333 6.1733 -61.890
## + log(trks)   1   0.00104 6.1756 -61.876
## - shld        1   1.05141 7.2280 -59.738
## + htype       3   0.13342 6.0432 -58.721
##
## Step:  AIC=-64.28
## log(rate * len) ~ shld + lwid
##
##              Df Sum of Sq    RSS     AIC
## <none>                    6.4335 -64.280
## + log(adt)    1   0.25691 6.1766 -63.869
## + log(sigs1)  1   0.10586 6.3276 -62.927
## + itg         1   0.07272 6.3608 -62.723
## + slim        1   0.06354 6.3700 -62.667
## + lane        1   0.04292 6.3906 -62.541
## + log(trks)   1   0.02366 6.4098 -62.423
## + acpt        1   0.00038 6.4331 -62.282
## + htype       3   0.22480 6.2087 -59.667
## - shld        1   2.36799 8.8015 -54.057
```

While treating *log(rateXlen)* as the response and *lwid* as the focal regressor, we see that both forward selection and backward elimination arrive at the same result: a final model comprising the regressors *lwid* and *shld*. From looking at the AIC values which result from the stepwise process, it seems like *lwid* is an important regressor as both the FS and BE methods agree that the only improvement to a model already containing *lwid* is an inclusion of *shld*. This seems to suggest that *lwid* by itself is quite important.

10.2.3: Repeat problem 10.2.2 but use *log(rate)* as the response and *-log(len)* as an offset. Is the analusis the same or different?

```
m.lwid <- lm(log(rate) ~ lwid, data = highway, offset = -log(len))

f2 = ~shld + log(adt) + log(trks) + lane + slim + itg + log(sigs1) + acpt +
    htype + lwid

m.upper <- update(m.lwid, f2)

m.sw.up = step(m.lwid, scope = f2)
```

```
## Start:  AIC=-54.06
## log(rate) ~ lwid
##
##              Df Sum of Sq    RSS     AIC
## + shld        1   2.36799 6.4335 -64.280
## + log(adt)    1   1.57348 7.2280 -59.738
## + htype       3   1.66818 7.1333 -56.253
## + lane        1   0.86973 7.9318 -56.115
## + slim        1   0.79623 8.0053 -55.755
## + itg         1   0.70020 8.1013 -55.290
## + acpt        1   0.45639 8.3451 -54.134
## <none>                    8.8015 -54.057
## + log(sigs1)  1   0.08316 8.7183 -52.427
## + log(trks)   1   0.02384 8.7776 -52.163
## - lwid        1   1.03754 9.8390 -51.711
##
## Step:  AIC=-64.28
## log(rate) ~ lwid + shld
##
##              Df Sum of Sq    RSS     AIC
## <none>                    6.4335 -64.280
## + log(adt)    1   0.25691 6.1766 -63.869
## + log(sigs1)  1   0.10586 6.3276 -62.927
## + itg         1   0.07272 6.3608 -62.723
## + slim        1   0.06354 6.3700 -62.667
## + lane        1   0.04292 6.3906 -62.541
## + log(trks)   1   0.02366 6.4098 -62.423
## + acpt        1   0.00038 6.4331 -62.282
## - lwid        1   1.17434 7.6078 -59.741
## + htype       3   0.22480 6.2087 -59.667
## - shld        1   2.36799 8.8015 -54.057
```

```
m.sw.down = step(m.upper, scope = list(lower = ~lwid, upper = m.upper), direction = "both")
```

```
## Start:  AIC=-47.69
## log(rate) ~ shld + log(adt) + log(trks) + lane + slim + itg +
##     log(sigs1) + acpt + htype + lwid
##
##              Df Sum of Sq    RSS     AIC
## - htype       3   0.18628 6.0812 -52.476
## - log(trks)   1   0.00007 5.8950 -49.689
## - slim        1   0.00200 5.8969 -49.676
## - lane        1   0.00242 5.8973 -49.673
```

```
## - log(sigs1)  1   0.00521 5.9001 -49.655
## - itg         1   0.04647 5.9414 -49.383
## - acpt        1   0.04806 5.9430 -49.373
## - log(adt)    1   0.08268 5.9776 -49.146
## <none>                    5.8949 -47.689
## - shld        1   0.35127 6.2462 -47.432
##
## Step:  AIC=-52.48
## log(rate) ~ shld + log(adt) + log(trks) + lane + slim + itg +
##     log(sigs1) + acpt + lwid
##
##              Df Sum of Sq    RSS     AIC
## - log(sigs1)  1   0.00066 6.0818 -54.472
## - itg         1   0.00112 6.0823 -54.469
## - log(trks)   1   0.00377 6.0850 -54.452
## - slim        1   0.00860 6.0898 -54.421
## - lane        1   0.03960 6.1208 -54.223
## - acpt        1   0.05202 6.1332 -54.144
## - log(adt)    1   0.12314 6.2043 -53.694
## <none>                    6.0812 -52.476
## - shld        1   0.46029 6.5415 -51.631
## + htype       3   0.18628 5.8949 -47.689
##
## Step:  AIC=-54.47
## log(rate) ~ shld + log(adt) + log(trks) + lane + slim + itg +
##     acpt + lwid
##
##              Df Sum of Sq    RSS     AIC
## - itg         1   0.00278 6.0846 -56.454
## - log(trks)   1   0.00575 6.0876 -56.435
## - slim        1   0.01142 6.0933 -56.399
## - lane        1   0.03905 6.1209 -56.222
## - acpt        1   0.05149 6.1333 -56.143
## - log(adt)    1   0.18006 6.2619 -55.334
## <none>                    6.0818 -54.472
## - shld        1   0.45999 6.5418 -53.628
## + log(sigs1)  1   0.00066 6.0812 -52.476
## + htype       3   0.18173 5.9001 -49.655
##
## Step:  AIC=-56.45
## log(rate) ~ shld + log(adt) + log(trks) + lane + slim + acpt +
##     lwid
##
##              Df Sum of Sq    RSS     AIC
## - log(trks)   1   0.00622 6.0908 -58.414
## - slim        1   0.01385 6.0985 -58.365
## - lane        1   0.04755 6.1322 -58.151
## - acpt        1   0.05046 6.1351 -58.132
## - log(adt)    1   0.21226 6.2969 -57.117
## <none>                    6.0846 -56.454
## - shld        1   0.50076 6.5854 -55.370
## + itg         1   0.00278 6.0818 -54.472
## + log(sigs1)  1   0.00232 6.0823 -54.469
## + htype       3   0.13472 5.9499 -51.327
```

```
##
## Step:  AIC=-58.41
## log(rate) ~ shld + log(adt) + lane + slim + acpt + lwid
##
##              Df Sum of Sq    RSS     AIC
## - slim        1   0.02034 6.1112 -60.284
## - lane        1   0.04412 6.1350 -60.133
## - acpt        1   0.04472 6.1356 -60.129
## - log(adt)    1   0.21095 6.3018 -59.086
## <none>                    6.0908 -58.414
## - shld        1   0.55605 6.6469 -57.007
## + log(trks)   1   0.00622 6.0846 -56.454
## + log(sigs1)  1   0.00523 6.0856 -56.448
## + itg         1   0.00325 6.0876 -56.435
## + htype       3   0.14022 5.9506 -53.323
##
## Step:  AIC=-60.28
## log(rate) ~ shld + log(adt) + lane + acpt + lwid
##
##              Df Sum of Sq    RSS     AIC
## - acpt        1   0.02530 6.1365 -62.123
## - lane        1   0.05111 6.1623 -61.959
## - log(adt)    1   0.27897 6.3902 -60.543
## <none>                    6.1112 -60.284
## + slim        1   0.02034 6.0908 -58.414
## + log(sigs1)  1   0.01590 6.0953 -58.386
## + log(trks)   1   0.01272 6.0985 -58.365
## + itg         1   0.00693 6.1043 -58.328
## - shld        1   0.75088 6.8621 -57.765
## + htype       3   0.15554 5.9556 -55.290
##
## Step:  AIC=-62.12
## log(rate) ~ shld + log(adt) + lane + lwid
##
##              Df Sum of Sq    RSS     AIC
## - lane        1   0.04011 6.1766 -63.869
## - log(adt)    1   0.25409 6.3906 -62.541
## <none>                    6.1365 -62.123
## + acpt        1   0.02530 6.1112 -60.284
## + itg         1   0.00248 6.1340 -60.139
## + log(sigs1)  1   0.00223 6.1343 -60.137
## + log(trks)   1   0.00109 6.1354 -60.130
## + slim        1   0.00092 6.1356 -60.129
## - shld        1   1.08816 7.2246 -57.756
## + htype       3   0.09768 6.0388 -56.749
##
## Step:  AIC=-63.87
## log(rate) ~ shld + log(adt) + lwid
##
##              Df Sum of Sq    RSS     AIC
## - log(adt)    1   0.25691 6.4335 -64.280
## <none>                    6.1766 -63.869
## + lane        1   0.04011 6.1365 -62.123
## + acpt        1   0.01429 6.1623 -61.959
```

```
## + itg          1   0.01141 6.1652 -61.941
## + slim         1   0.00457 6.1720 -61.898
## + log(sigs1)   1   0.00333 6.1733 -61.890
## + log(trks)    1   0.00104 6.1756 -61.876
## - shld         1   1.05141 7.2280 -59.738
## + htype        3   0.13342 6.0432 -58.721
##
## Step:  AIC=-64.28
## log(rate) ~ shld + lwid
##
##               Df Sum of Sq    RSS     AIC
## <none>                     6.4335 -64.280
## + log(adt)     1   0.25691 6.1766 -63.869
## + log(sigs1)   1   0.10586 6.3276 -62.927
## + itg          1   0.07272 6.3608 -62.723
## + slim         1   0.06354 6.3700 -62.667
## + lane         1   0.04292 6.3906 -62.541
## + log(trks)    1   0.02366 6.4098 -62.423
## + acpt         1   0.00038 6.4331 -62.282
## + htype        3   0.22480 6.2087 -59.667
## - shld         1   2.36799 8.8015 -54.057
```

When repeating problem 10.2.2 while using *-log(len)* as an offset, we see that the end result of both the FS and BE methods is the same: a final model with the regressors *lwid* and *shld*. This seems to be because AIC ignores constants which are the same for every candidate subset, and the offset sets a known coefficient equal to -1 for *log(len)*.

2. ALR 10.4: For the boys in the Berkeley Guidance Study find a model for *HT18* as a function of the other variables for ages 9 and earlier. Perform a complete analysis including selection of transformations and diagnostic analysis, and summarize your results.

```r
boys <- BGSboys

bc1 = powerTransform(cbind(WT2, HT2, WT9, HT9, LG9, ST9) ~ 1, boys)
summary(bc1)
```

```
## bcPower Transformations to Multinormality
##     Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## WT2   -1.2909           0      -2.8558       0.2740
## HT2   -1.9862           1      -6.3379       2.3654
## WT9   -1.3138          -1      -1.9487      -0.6789
## HT9   -1.1724           1      -5.5175       3.1727
## LG9   -2.1851          -1      -3.6035      -0.7667
## ST9    0.5611           1      -0.1592       1.2815
##
## Likelihood ratio test that transformation parameters are equal to 0
##  (all log transformations)
##                                       LRT df       pval
## LR test, lambda = (0 0 0 0 0 0) 23.54392  6 0.00063335
##
## Likelihood ratio test that no transformations are needed
##                                       LRT df       pval
## LR test, lambda = (1 1 1 1 1 1) 65.16733  6 3.9876e-12
```

```r
# So log transformation for WT2, and inverse for WT9 and LG9.

m0 <- lm(HT18 ~ 1, data = boys)

full <- ~log(WT2) + HT2 + I(1/WT9) + HT9 + I(1/LG9) + ST9

m.fwd <- step(m0, scope = full, direction = "forward")
```

```
## Start:  AIC=248.42
## HT18 ~ 1
##
##             Df Sum of Sq     RSS    AIC
## + HT9        1   2113.28  647.75 154.73
## + HT2        1    899.73 1861.30 224.40
## + I(1/WT9)   1    866.61 1894.42 225.56
## + log(WT2)   1    574.94 2186.09 235.01
## + ST9        1    510.51 2250.52 236.93
## + I(1/LG9)   1    437.56 2323.47 239.04
## <none>                   2761.03 248.42
##
## Step:  AIC=154.73
## HT18 ~ HT9
##
##             Df Sum of Sq    RSS    AIC
## + I(1/LG9)   1    55.425 592.33 150.83
## + I(1/WT9)   1    41.669 606.08 152.35
## <none>                   647.75 154.73
## + HT2        1    10.560 637.19 155.65
## + ST9        1     0.635 647.12 156.67
## + log(WT2)   1     0.032 647.72 156.73
##
## Step:  AIC=150.83
## HT18 ~ HT9 + I(1/LG9)
##
##             Df Sum of Sq    RSS    AIC
## <none>                   592.33 150.83
## + HT2        1    9.0408 583.29 151.82
## + log(WT2)   1    8.7655 583.56 151.85
## + ST9        1    7.3625 584.97 152.00
## + I(1/WT9)   1    0.3116 592.02 152.80
```

```r
# From FW, HT9 and inverse LG9 are most necessary, AIC of 150.83

m1 <- update(m0, full)

m.bck = step(m1, scope = list(lower = m0, upper = m1), direction = "backward")
```

```
## Start:  AIC=153.99
## HT18 ~ log(WT2) + HT2 + I(1/WT9) + HT9 + I(1/LG9) + ST9
##
##             Df Sum of Sq     RSS    AIC
## - I(1/WT9)   1      0.09  550.54 152.00
## - ST9        1     10.57  561.02 153.25
## <none>                    550.45 153.99
## - log(WT2)   1     21.00  571.45 154.46
```

```
## - I(1/LG9)  1     23.81  574.26 154.79
## - HT2       1     27.01  577.46 155.15
## - HT9       1    834.59 1385.04 212.89
##
## Step:  AIC=152
## HT18 ~ log(WT2) + HT2 + HT9 + I(1/LG9) + ST9
##
##             Df Sum of Sq     RSS    AIC
## - ST9        1     10.48  561.02 151.25
## <none>                    550.54 152.00
## - log(WT2)  1     21.70  572.23 152.55
## - HT2        1     27.07  577.61 153.17
## - I(1/LG9)  1     84.84  635.37 159.46
## - HT9        1   1183.91 1734.45 225.74
##
## Step:  AIC=151.25
## HT18 ~ log(WT2) + HT2 + HT9 + I(1/LG9)
##
##             Df Sum of Sq     RSS    AIC
## <none>                    561.02 151.25
## - log(WT2)  1     22.27  583.29 151.82
## - HT2        1     22.54  583.56 151.85
## - I(1/LG9)  1     74.58  635.59 157.48
## - HT9        1   1210.14 1771.15 225.12
```

```r
# This BE method also adds log of WT2 and HT2 for an AIC of 151.25.

bc2 = powerTransform(HT18 ~ log(WT2) + HT2 + HT9 + I(1/LG9), boys)
summary(bc2)
```

```
## bcPower Transformation to Normality
##    Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## Y1    0.6006           1       -2.792       3.9931
##
## Likelihood ratio test that transformation parameter is equal to 0
##  (log transformation)
##                              LRT df    pval
## LR test, lambda = (0) 0.1206478  1 0.72833
##
## Likelihood ratio test that no transformation is needed
##                               LRT df    pval
## LR test, lambda = (1) 0.05315145  1 0.81767
```

```r
# Don't need to transform HT18 with these regressors

m.boysfull <- lm(HT18 ~ log(WT2) * HT2 * HT9 * I(1/LG9), boys)

Anova(m.boysfull)
```

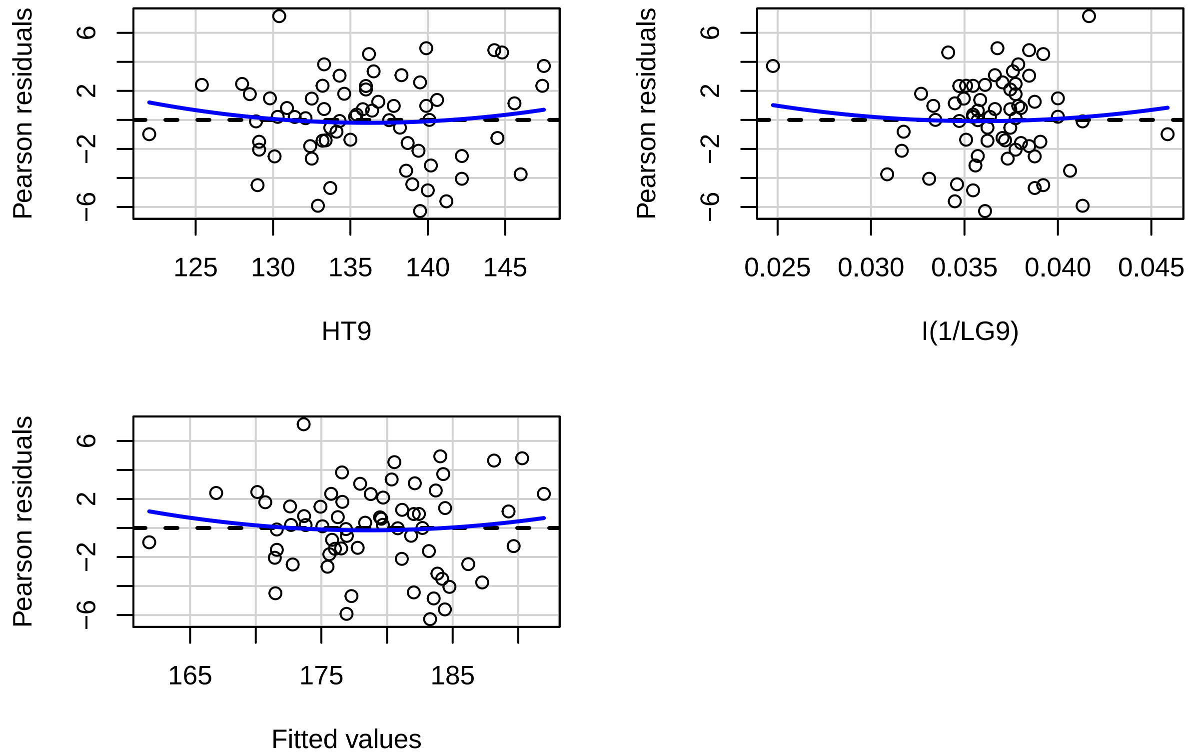
```
## Anova Table (Type II tests)
##
## Response: HT18
##                             Sum Sq Df F value    Pr(>F)
## log(WT2)                     51.17  4  1.4731  0.224400
## HT2                          12.67  2  0.7295  0.487197
```

```
## HT9                        1198.07  3 45.9827 2.075e-14 ***
## I(1/LG9)                    158.80  3  6.0947  0.001286 **
## log(WT2):HT2                  0.57  1  0.0652  0.799508
## log(WT2):HT9                  6.57  1  0.7566  0.388560
## HT2:HT9                       0.13  1  0.0149  0.903278
## log(WT2):I(1/LG9)             6.09  1  0.7013  0.406346
## HT2:I(1/LG9)                  1.14  1  0.1311  0.718813
## HT9:I(1/LG9)                 36.21  1  4.1692  0.046458 *
## log(WT2):HT2:HT9              6.91  1  0.7957  0.376651
## log(WT2):HT2:I(1/LG9)         6.62  1  0.7624  0.386746
## log(WT2):HT9:I(1/LG9)         5.96  1  0.6863  0.411358
## HT2:HT9:I(1/LG9)              3.80  1  0.4372  0.511499
## log(WT2):HT2:HT9:I(1/LG9)     4.18  1  0.4813  0.491038
## Residuals                   434.25 50
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# From this, seems like only HT9, inverse of LG9, and their interaction are
# needed.

m.boys <- lm(HT18 ~ HT9 * I(1/LG9), boys)
m.boys.main <- lm(HT18 ~ HT9 + I(1/LG9), boys)

anova(m.boys.main, m.boys)
```

```
## Analysis of Variance Table
##
## Model 1: HT18 ~ HT9 + I(1/LG9)
## Model 2: HT18 ~ HT9 * I(1/LG9)
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     63 592.33
## 2     62 549.00  1    43.326 4.8929 0.03066 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# So, interaction seems necessary. Can't reject the null that the
# interaction adds to our understanding of the response.

rp1 <- residualPlots(m.boys)
```




```
##             Test stat Pr(>|Test stat|)
## HT9            1.0630           0.2920
## I(1/LG9)       1.2142           0.2294
## Tukey test     0.7536           0.4511
```

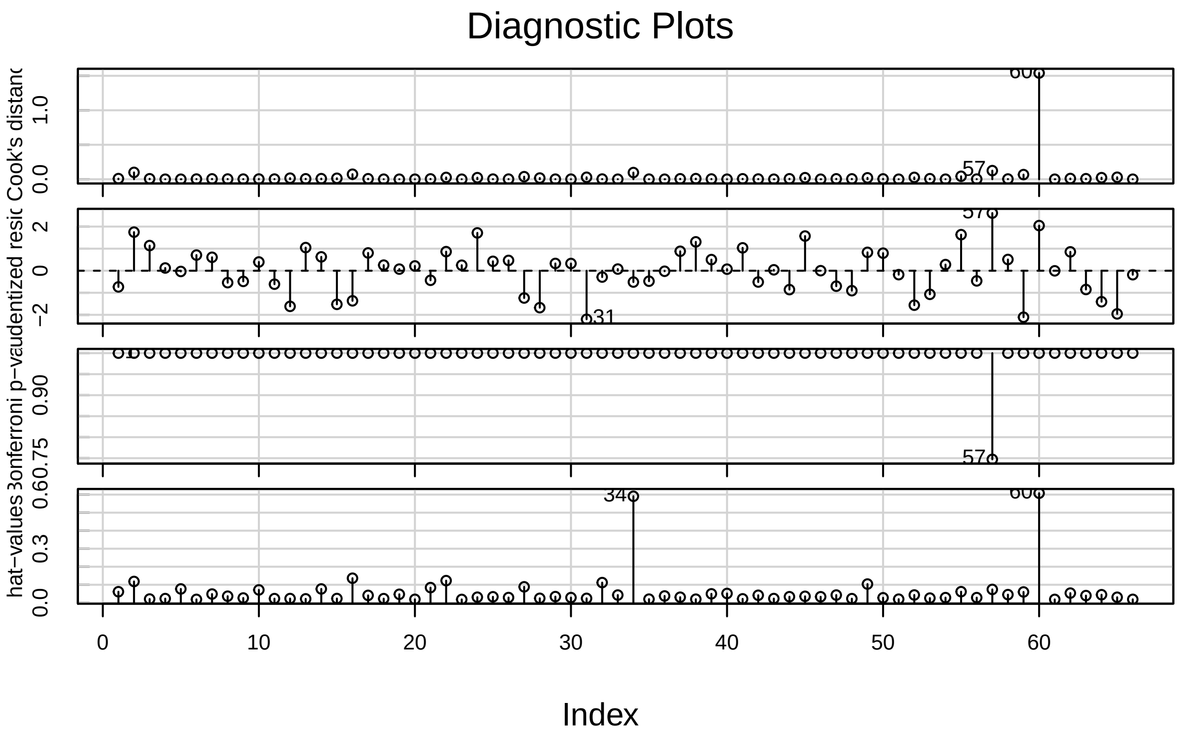
```r
rp1
```

```
##             Test stat Pr(>|Test stat|)
## HT9         1.0629861        0.2919784
## I(1/LG9)    1.2141998        0.2293513
## Tukey test  0.7535821        0.4511002
```

```r
ncvTest(m.boys)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 2.360638    Df = 1     p = 0.1244313
```

```r
# So, no polynomial term seems to be necessary, and can't reject the null
# hypothesis that the variance is constant.

influenceIndexPlot(m.boys, id.n = 3)
```




```
outlierTest(m.boys)
```

```
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##    rstudent unadjusted p-value Bonferonni p
## 57 2.611827           0.011322      0.74722
```

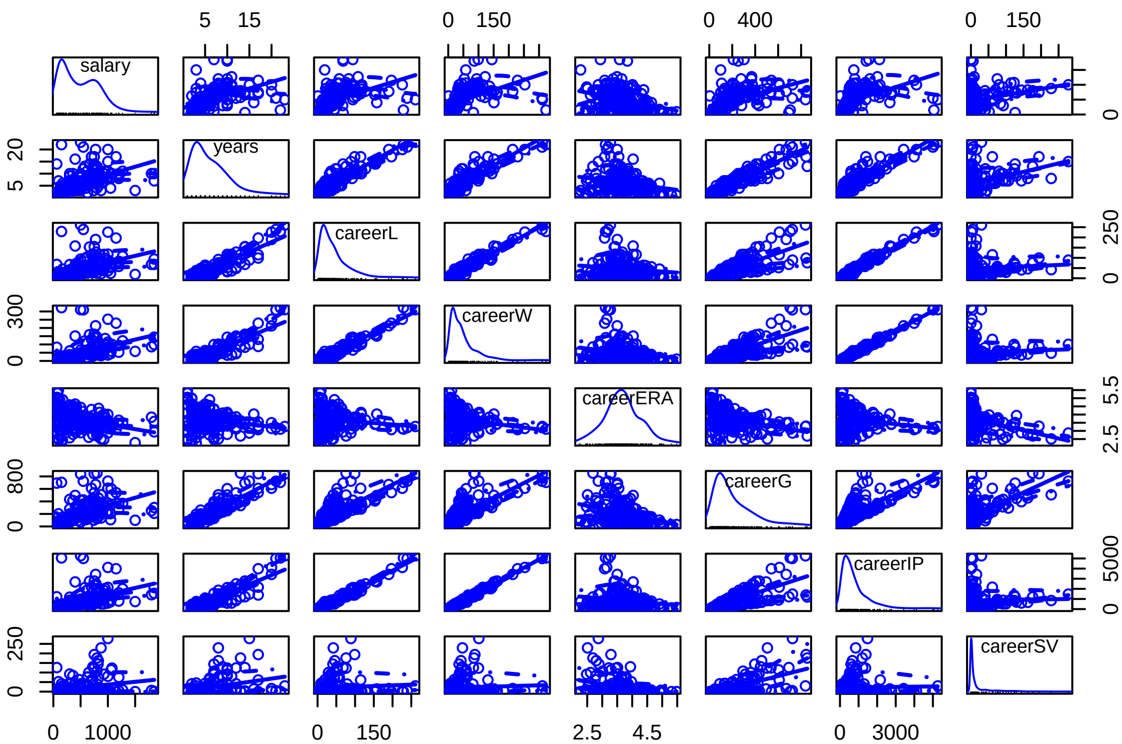
```
# So. no real outliers, but datapoint 60 is very comparitively influential.
```

Based on the above analyses, it seems that the best model for explaining the variance seen in *HT18* is $lm(HT18 \sim HT9 * I(1/LG9)$. Diagnostic tests revealed that no polynomial term was necessary for this model, the variance can be treated as constant, and finally that there were no outliers in this data - save for entry 60 which was found to be highly influential with a relatively high cook's distance of around 1.5.

3. Use the baseball pitchers data to answer the following questions.

a. Employing one or more of the methods of model selection described in the course, develop a model to predict pitchers' salaries. Be sure to explore the data and think about variables to use as predictors before specifying candidate models. How good is the model, and does it make sense?

```
baseball <- read.table("BaseballPitchers.txt", header = TRUE)

scatterplotMatrix(~salary + years + careerL + careerW + careerERA + careerG +
    careerIP + careerSV, smoother = FALSE, data = baseball)
```




```
# The above variables seem to have the strongest relationship with salary

baseball$careerSV2 <- baseball$careerSV + 0.001

bc1 = powerTransform(cbind(years, careerL, careerW, careerERA, careerG, careerIP,
    careerSV2) ~ 1, baseball)
summary(bc1)
```

```
## bcPower Transformations to Multinormality
##           Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## years        0.4395        0.50       0.3510       0.5280
## careerL      0.3427        0.33       0.2856       0.3998
## careerW      0.3529        0.33       0.2993       0.4064
## careerERA    0.6458        1.00       0.0701       1.2215
## careerG      0.3522        0.33       0.2981       0.4064
## careerIP     0.3423        0.33       0.2948       0.3898
## careerSV2    0.2747        0.27       0.2420       0.3074
##
## Likelihood ratio test that transformation parameters are equal to 0
##  (all log transformations)
##                                    LRT df       pval
## LR test, lambda = (0 0 0 0 0 0 0) 396.913  7 < 2.22e-16
##
## Likelihood ratio test that no transformations are needed
##                                     LRT df       pval
## LR test, lambda = (1 1 1 1 1 1 1) 1758.888  7 < 2.22e-16
```

```
bc2 = powerTransform(salary ~ sqrt(years) + I(careerL^(1/3)) + I(careerW^(1/3)) +
    careerERA + I(careerG^(1/3)) + I(careerIP^(1/3)) + I(careerSV2^(1/3)), baseball)
summary(bc2)
```

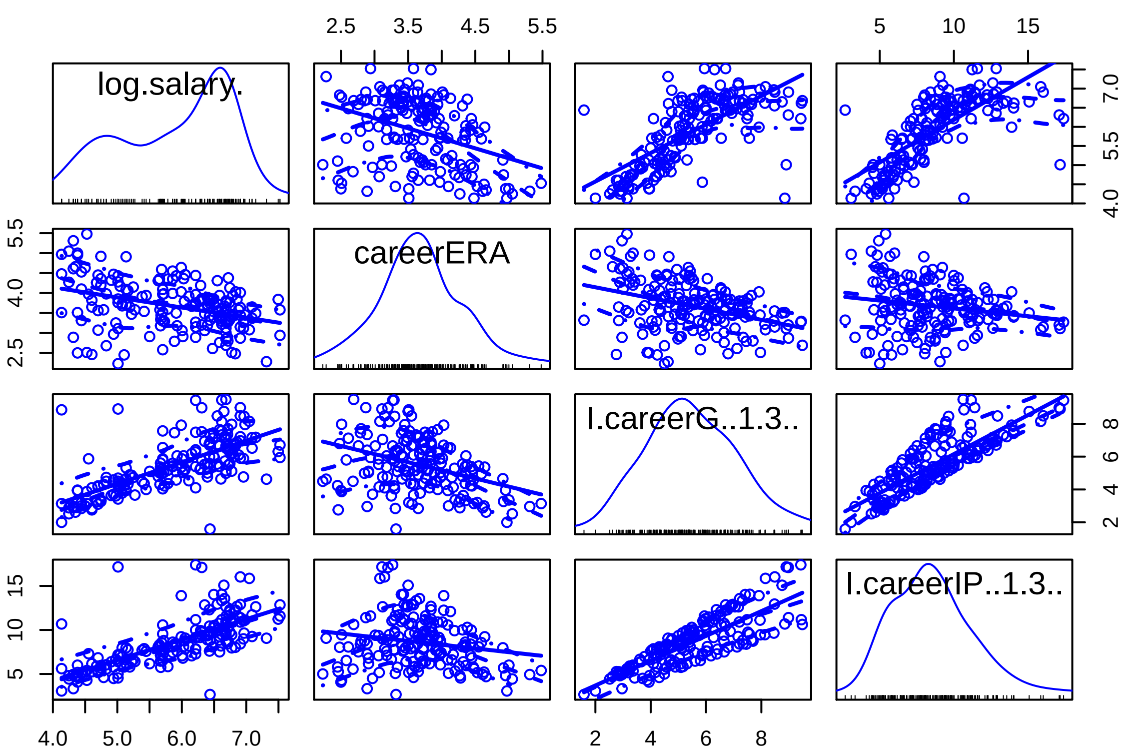
```
## bcPower Transformation to Normality
##    Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## Y1    0.1736        0.33       0.0155       0.3317
##
## Likelihood ratio test that transformation parameter is equal to 0
##  (log transformation)
##                            LRT df     pval
## LR test, lambda = (0) 4.630849  1 0.031402
##
## Likelihood ratio test that no transformation is needed
##                            LRT df       pval
## LR test, lambda = (1) 98.06735  1 < 2.22e-16
```

```r
# So, with these regressors, salary needs a cube root transformation, but it
# is really close to a log transformation, and that is easier to interpret,
# so I will use that.

m.base.full <- lm(log(salary) ~ sqrt(years) + I(careerL^(1/3)) + I(careerW^(1/3)) +
    careerERA + I(careerG^(1/3)) + I(careerIP^(1/3)) + I(careerSV2^(1/3)), baseball)

m.bck = step(m.base.full, scope = list(lower = ~1, upper = m.base.full), direction = "backward")
```

```
## Start:  AIC=-182.16
## log(salary) ~ sqrt(years) + I(careerL^(1/3)) + I(careerW^(1/3)) +
##     careerERA + I(careerG^(1/3)) + I(careerIP^(1/3)) + I(careerSV2^(1/3))
##
##                       Df Sum of Sq    RSS     AIC
## - I(careerW^(1/3))     1   0.01663 57.104 -184.11
## - I(careerL^(1/3))     1   0.02274 57.110 -184.09
## - I(careerSV2^(1/3))   1   0.03043 57.118 -184.06
## - I(careerG^(1/3))     1   0.25101 57.339 -183.39
## - I(careerIP^(1/3))    1   0.26780 57.355 -183.33
## - sqrt(years)          1   0.36045 57.448 -183.05
## <none>                             57.088 -182.16
## - careerERA            1   2.43881 59.526 -176.79
##
## Step:  AIC=-184.11
## log(salary) ~ sqrt(years) + I(careerL^(1/3)) + careerERA + I(careerG^(1/3)) +
##     I(careerIP^(1/3)) + I(careerSV2^(1/3))
##
##                       Df Sum of Sq    RSS     AIC
## - I(careerL^(1/3))     1   0.01695 57.121 -186.05
## - I(careerSV2^(1/3))   1   0.03918 57.143 -185.99
## - I(careerG^(1/3))     1   0.24483 57.349 -185.35
## - sqrt(years)          1   0.39715 57.501 -184.89
## <none>                             57.104 -184.11
## - I(careerIP^(1/3))    1   1.00286 58.107 -183.04
## - careerERA            1   2.66031 59.765 -178.09
##
## Step:  AIC=-186.05
## log(salary) ~ sqrt(years) + careerERA + I(careerG^(1/3)) + I(careerIP^(1/3)) +
##     I(careerSV2^(1/3))
##
##                       Df Sum of Sq    RSS     AIC
## - I(careerSV2^(1/3))   1    0.0431 57.164 -187.92
```

```
## - I(careerG^(1/3))    1    0.2668 57.388 -187.23
## - sqrt(years)         1    0.3979 57.519 -186.83
## <none>                            57.121 -186.05
## - I(careerIP^(1/3))   1    3.1380 60.259 -178.64
## - careerERA           1    3.2488 60.370 -178.32
##
## Step:  AIC=-187.92
## log(salary) ~ sqrt(years) + careerERA + I(careerG^(1/3)) + I(careerIP^(1/3))
##
##                     Df Sum of Sq    RSS     AIC
## - sqrt(years)        1    0.3838 57.548 -188.74
## <none>                           57.164 -187.92
## - I(careerG^(1/3))   1    1.7672 58.932 -184.56
## - careerERA          1    3.4608 60.625 -179.58
## - I(careerIP^(1/3))  1    8.5505 65.715 -165.39
##
## Step:  AIC=-188.74
## log(salary) ~ careerERA + I(careerG^(1/3)) + I(careerIP^(1/3))
##
##                     Df Sum of Sq    RSS     AIC
## <none>                           57.548 -188.74
## - I(careerG^(1/3))   1    1.7353 59.284 -185.51
## - careerERA          1    4.2774 61.826 -178.12
## - I(careerIP^(1/3))  1   11.4809 69.029 -158.73
```

```r
# So, this suggests that the model should be based on careerERA, careerG^1/3
# and careerIP^1/3.

scatterplotMatrix(~log(salary) + careerERA + I(careerG^(1/3)) + I(careerIP^(1/3)),
    smoother = FALSE, data = baseball)
```



```r
# Looks good enough! The curving of the data for careerG and IP is a little
# concerning, but it seems straight enough for my current purposes.


m.pitch.final <- lm(log(salary) ~ careerERA + I(careerG^(1/3)) + I(careerIP^(1/3)),
    baseball)

summary(m.pitch.final)
```

```
##
## Call:
## lm(formula = log(salary) ~ careerERA + I(careerG^(1/3)) + I(careerIP^(1/3)),
##     data = baseball)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.75505 -0.34531  0.07592  0.37316  1.84521
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        4.91523    0.37065  13.261  < 2e-16 ***
## careerERA         -0.27805    0.07777  -3.576 0.000454 ***
## I(careerG^(1/3))   0.11199    0.04918   2.277 0.023994 *
## I(careerIP^(1/3))  0.15840    0.02704   5.858 2.33e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5784 on 172 degrees of freedom
##   (30 observations deleted due to missingness)
## Multiple R-squared:  0.5857, Adjusted R-squared:  0.5784
## F-statistic: 81.04 on 3 and 172 DF,  p-value: < 2.2e-16
```

This chosen model, with *careerERA*, and the cube roots of *careerG* and *careerIP* as the regressors explains roughly 58% of the variance seen in the (log treansformed) salary variable. This seems to be pretty good for this rather messy data - and it also makes a good amount of sense. Pitchers who do a better job striking out batters (lower ERA), and who have been playing in the league longer and so have more experience, end up with higher salaries. So, more experienced better (by one important, individual, metric) pitchers make more money.

b. Repeat part a but divide the data randomly into two subsamples, applying one or more methods of model selection to the first subsample. Then evaluate the selected models on the second subsample.

```r
set.seed(103)  # Set Seed so that same sample can be reproduced in future also
# Now Selecting 50% of data as sample from total 'n' rows of the data
sample <- sample.int(n = nrow(baseball), size = floor(0.5 * nrow(baseball)),
    replace = F)
train <- baseball[sample, ]
test <- baseball[-sample, ]


# Repeating transformation stuff with the training set.

bc1 = powerTransform(cbind(years, careerL, careerW, careerERA, careerG, careerIP,
```

```r
    careerSV2) ~ 1, train)
summary(bc1)
```

```
## bcPower Transformations to Multinormality
##            Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## years         0.4494        0.50       0.3175       0.5814
## careerL       0.2783        0.33       0.1993       0.3573
## careerW       0.3201        0.33       0.2465       0.3938
## careerERA     0.5518        1.00      -0.2347       1.3382
## careerG       0.3127        0.33       0.2392       0.3861
## careerIP      0.2999        0.33       0.2356       0.3642
## careerSV2     0.2653        0.27       0.2152       0.3153
##
## Likelihood ratio test that transformation parameters are equal to 0
##  (all log transformations)
##                                         LRT df       pval
## LR test, lambda = (0 0 0 0 0 0 0) 164.7615  7 < 2.22e-16
##
## Likelihood ratio test that no transformations are needed
##                                         LRT df       pval
## LR test, lambda = (1 1 1 1 1 1 1) 923.2698  7 < 2.22e-16
```

```r
bc2 = powerTransform(salary ~ sqrt(years) + I(careerL^(1/3)) + I(careerW^(1/3)) +
    careerERA + I(careerG^(1/3)) + I(careerIP^(1/3)) + I(careerSV2^(1/3)), train)
summary(bc2)
```

```
## bcPower Transformation to Normality
##    Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## Y1    0.1638           0      -0.0606       0.3882
##
## Likelihood ratio test that transformation parameter is equal to 0
##  (log transformation)
##                             LRT df    pval
## LR test, lambda = (0) 2.029669  1 0.15425
##
## Likelihood ratio test that no transformation is needed
##                             LRT df       pval
## LR test, lambda = (1) 51.57954  1 6.8756e-13
```

```r
# So, rounded powers all seem to be the same.

m.train.full <- lm(log(salary) ~ sqrt(years) + I(careerL^(1/3)) + I(careerW^(1/3)) +
    careerERA + I(careerG^(1/3)) + I(careerIP^(1/3)) + I(careerSV2^(1/3)), train)

m.bck = step(m.train.full, scope = list(lower = ~1, upper = m.train.full), direction = "backward")
```

```
## Start:  AIC=-78.8
## log(salary) ~ sqrt(years) + I(careerL^(1/3)) + I(careerW^(1/3)) +
##     careerERA + I(careerG^(1/3)) + I(careerIP^(1/3)) + I(careerSV2^(1/3))
##
##                       Df Sum of Sq    RSS     AIC
## - sqrt(years)          1   0.00002 32.108 -80.800
## - I(careerW^(1/3))     1   0.02155 32.129 -80.739
## - I(careerIP^(1/3))    1   0.03304 32.141 -80.707
## - I(careerG^(1/3))     1   0.06119 32.169 -80.627
```

```
## - I(careerL^(1/3))    1   0.07439 32.182 -80.590
## - I(careerSV2^(1/3))  1   0.09222 32.200 -80.539
## <none>                            32.108 -78.800
## - careerERA           1   2.75020 34.858 -73.322
##
## Step:  AIC=-80.8
## log(salary) ~ I(careerL^(1/3)) + I(careerW^(1/3)) + careerERA +
##     I(careerG^(1/3)) + I(careerIP^(1/3)) + I(careerSV2^(1/3))
##
##                      Df Sum of Sq    RSS     AIC
## - I(careerW^(1/3))    1   0.02247 32.130 -82.737
## - I(careerIP^(1/3))   1   0.03498 32.143 -82.701
## - I(careerG^(1/3))    1   0.06357 32.171 -82.620
## - I(careerL^(1/3))    1   0.07457 32.182 -82.589
## - I(careerSV2^(1/3))  1   0.09410 32.202 -82.534
## <none>                            32.108 -80.800
## - careerERA           1   2.77239 34.880 -75.264
##
## Step:  AIC=-82.74
## log(salary) ~ I(careerL^(1/3)) + careerERA + I(careerG^(1/3)) +
##     I(careerIP^(1/3)) + I(careerSV2^(1/3))
##
##                      Df Sum of Sq    RSS     AIC
## - I(careerG^(1/3))    1   0.05686 32.187 -84.576
## - I(careerL^(1/3))    1   0.05955 32.190 -84.568
## - I(careerSV2^(1/3))  1   0.08240 32.213 -84.503
## - I(careerIP^(1/3))   1   0.27109 32.401 -83.972
## <none>                            32.130 -82.737
## - careerERA           1   2.86511 34.995 -76.964
##
## Step:  AIC=-84.58
## log(salary) ~ I(careerL^(1/3)) + careerERA + I(careerIP^(1/3)) +
##     I(careerSV2^(1/3))
##
##                      Df Sum of Sq    RSS     AIC
## - I(careerSV2^(1/3))  1   0.02992 32.217 -86.491
## - I(careerL^(1/3))    1   0.08621 32.273 -86.332
## - I(careerIP^(1/3))   1   0.50354 32.691 -85.163
## <none>                            32.187 -84.576
## - careerERA           1   3.00631 35.193 -78.450
##
## Step:  AIC=-86.49
## log(salary) ~ I(careerL^(1/3)) + careerERA + I(careerIP^(1/3))
##
##                     Df Sum of Sq    RSS     AIC
## - I(careerL^(1/3))   1    0.0586 32.276 -88.326
## <none>                           32.217 -86.491
## - I(careerIP^(1/3))  1    0.7451 32.962 -86.410
## - careerERA          1    3.6839 35.901 -78.639
##
## Step:  AIC=-88.33
## log(salary) ~ careerERA + I(careerIP^(1/3))
##
##                     Df Sum of Sq    RSS     AIC
```

```
## <none>                              32.276 -88.326
## - careerERA           1    4.1315 36.407 -79.365
## - I(careerIP^(1/3))  1   30.8998 63.175 -29.210
```

```r
# interestingly, this training set points to a model without careerG like in
# part a, here just with careerERA and careerIP. Though the overall AIC is
# much higher than in part a, so maybe this model won't be as good?

m.test <- lm(log(salary) ~ careerERA + I(careerIP^(1/3)), test)

summary(m.test)
```

```
##
## Call:
## lm(formula = log(salary) ~ careerERA + I(careerIP^(1/3)), data = test)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8034 -0.3342  0.0708  0.3695  1.1396
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         5.50415    0.49483  11.123  < 2e-16 ***
## careerERA          -0.36545    0.10797  -3.385  0.00109 **
## I(careerIP^(1/3))  0.20082    0.02191   9.164 3.39e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5728 on 82 degrees of freedom
##   (18 observations deleted due to missingness)
## Multiple R-squared:  0.5957, Adjusted R-squared:  0.5858
## F-statistic: 60.41 on 2 and 82 DF,  p-value: < 2.2e-16
```

Interstingly, even though the AIC arrived at using the backward elimination method on the training set was less negative than in part a when the full dataset was used, when the final model identified by the training set was evaluated on the validation subsample - the amount of variation explained by the model was slightly higher than in part a - 59% vs 58%. This is also despite the model chosen in part b having one less regressor. So at the very worst, the method of subsetting the data to choose a model used here is equivalent to using backward elimination on the entire dataset - at least in this particular case.