Regression Basics

Jesse Lecy

April 14, 2016

create some fake data

```
x1 <- 1:100

x2 <- -0.1*x1 + rnorm(100)

x3 <- 0.05*x2 + rnorm(100)

y <- 2*x1 + 10*rnorm(100) + 10*x2

dat <- data.frame( y, x1, x2, x3 )
```

descriptive statistics

```
library( pastecs )
```

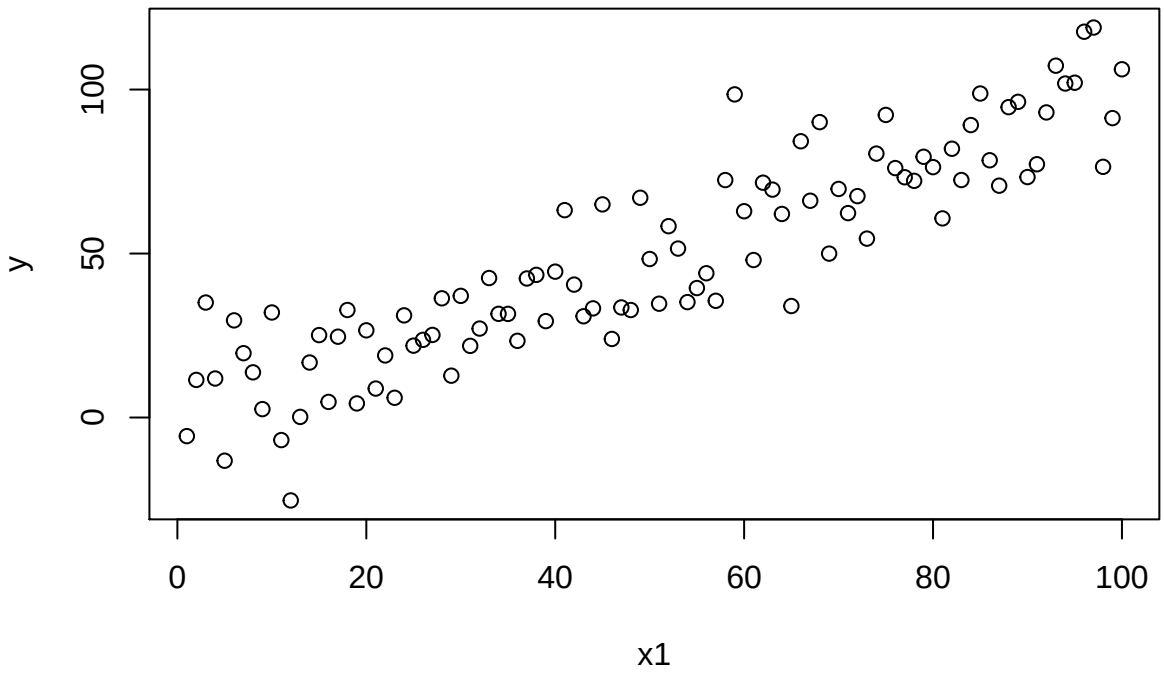
```
## Loading required package: boot
```

```
t( stat.desc( dat ) )
```

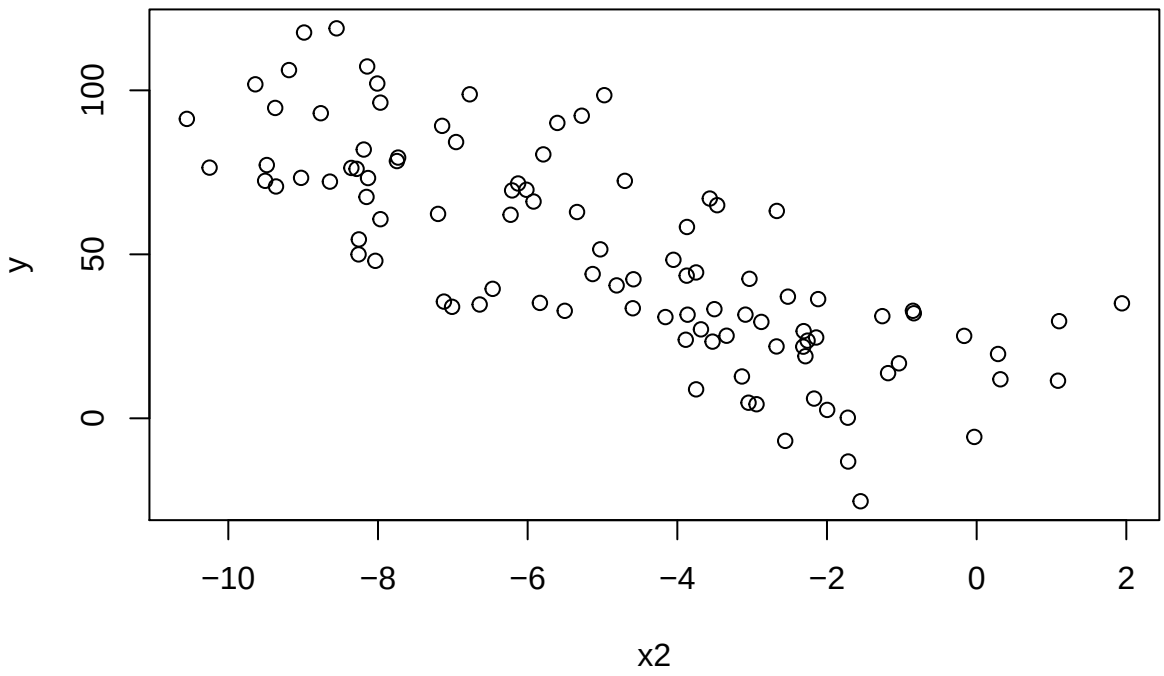
```
##    nbr.val nbr.null nbr.na        min        max      range         sum
## y      100        0      0 -25.279848 118.900147 144.179995 4959.017423
## x1     100        0      0   1.000000 100.000000  99.000000 5050.000000
## x2     100        0      0 -10.554231   1.941608  12.495839 -491.656891
## x3     100        0      0  -2.525308   2.348630   4.873938   -8.008704
##         median        mean   SE.mean CI.mean.0.95         var   std.dev
## y  43.74483334 49.59017423 3.2038440    6.3571215 1026.461609 32.038440
## x1 50.50000000 50.50000000 2.9011492    5.7565094  841.666667 29.011492
## x2 -4.75620811 -4.91656891 0.3000515    0.5953673    9.003091  3.000515
## x3 -0.09426656 -0.08008704 0.1068463    0.2120063    1.141614  1.068463
##       coef.var
## y    0.6460643
## x1   0.5744850
## x2  -0.6102864
## x3 -13.3412754
```

```
# To copy and paste into Excel:
#
# descriptives <- t( stat.desc(dat) )
#
# write.table( descriptives, "clipboard", sep="\t", row.names=TRUE )
```

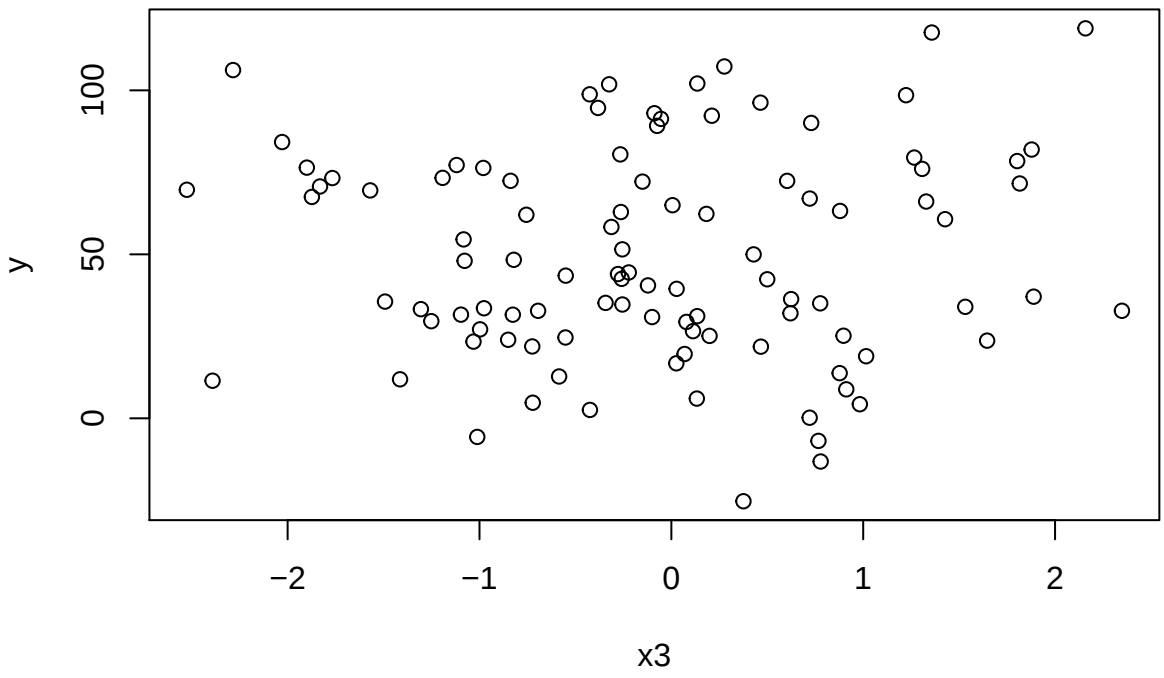
```
plot( x1, y )
```



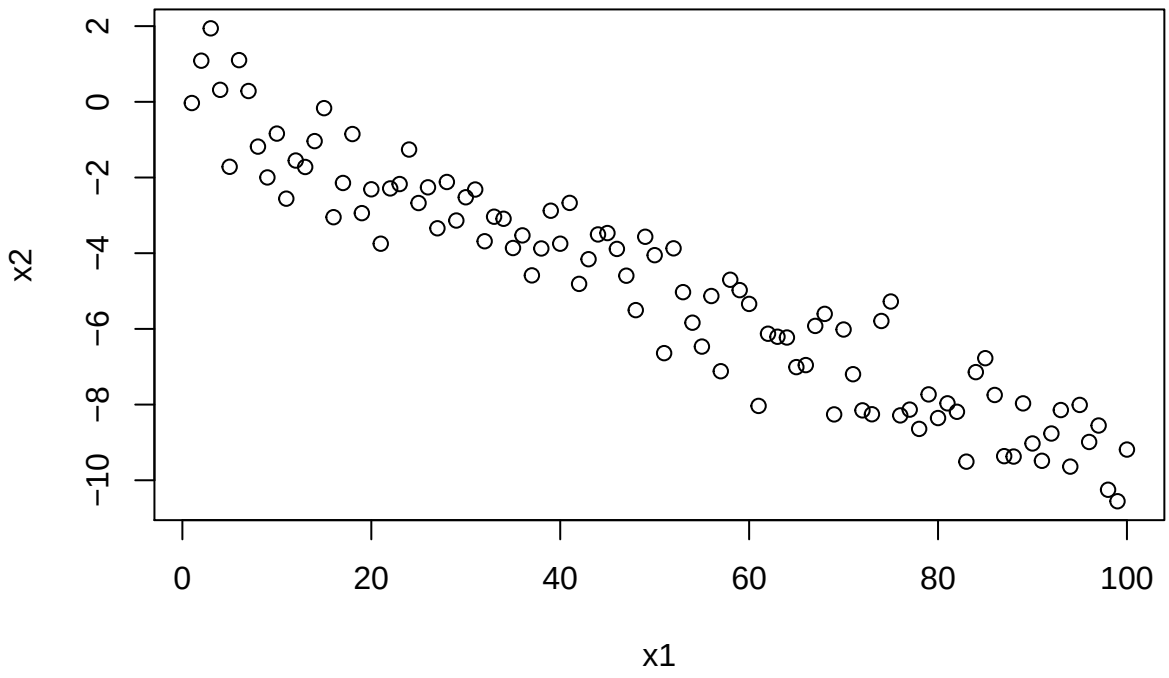

```
plot( x2, y )
```



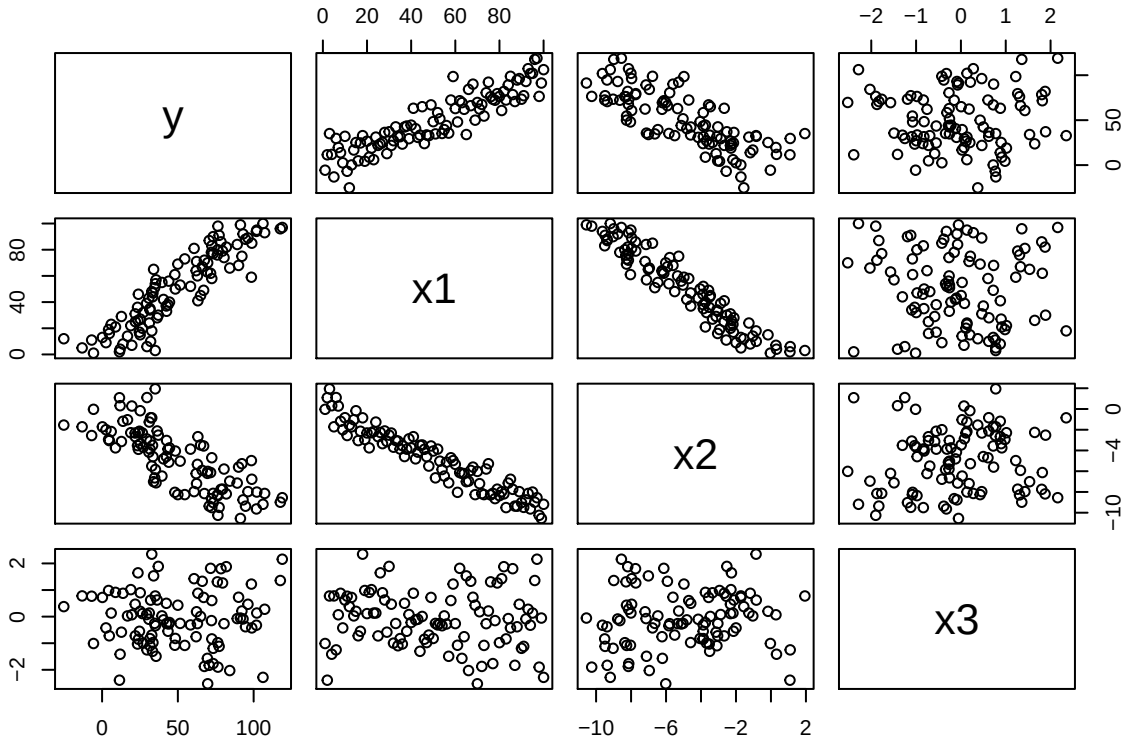

```
plot( x3, y )
```




```
plot( x1, x2 )
```

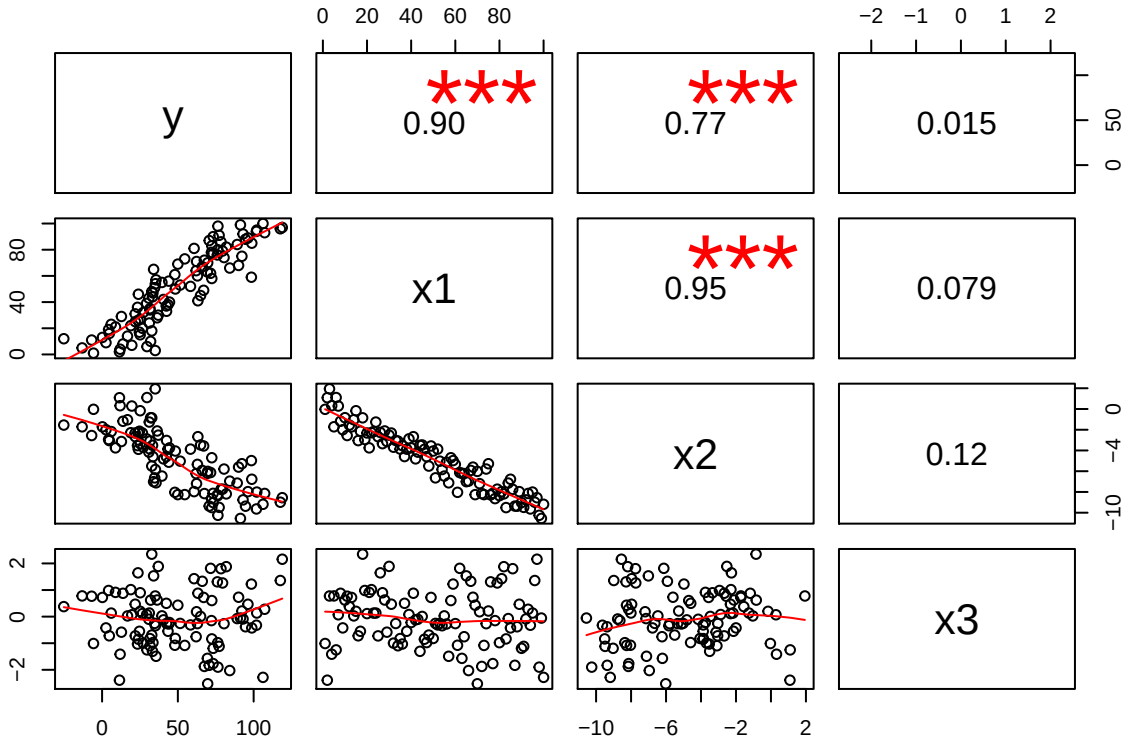



```
pairs( dat )
```



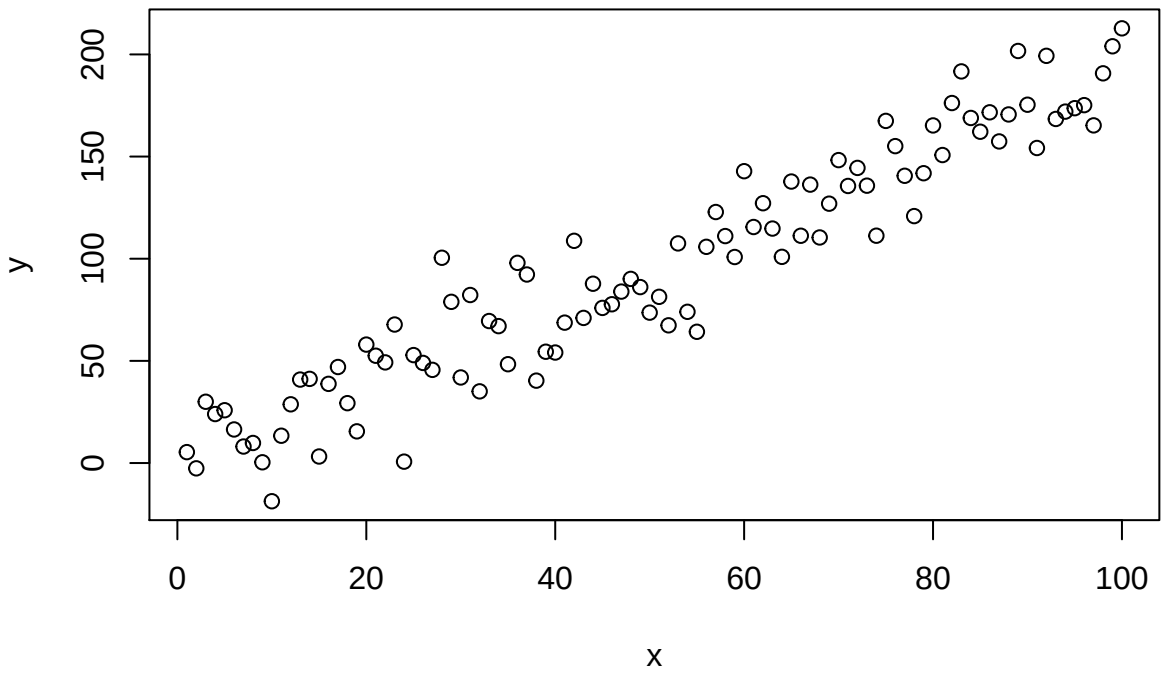
pretty pairs plot

```
panel.cor <- function(x, y, digits=2, prefix="", cex.cor)
{
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- abs(cor(x, y))
    txt <- format(c(r, 0.123456789), digits=digits)[1]
    txt <- paste(prefix, txt, sep="")
    if(missing(cex.cor)) cex <- 0.8/strwidth(txt)

    test <- cor.test(x,y)
    # borrowed from printCoefmat
    Signif <- symnum(test$p.value, corr = FALSE, na = FALSE,
                  cutpoints = c(0, 0.001, 0.01, 0.05, 0.1, 1),
                  symbols = c("***", "**", "*", ".", " "))

    text(0.5, 0.5, txt, cex = 1.5 )
    text(.7, .8, Signif, cex=cex, col=2)
}


pairs( dat, lower.panel=panel.smooth, upper.panel=panel.cor)
```



create some fake regression data

```
x <- 1:100
y <- 2*x + rnorm(100,0,20)

plot( x, y )
```




```
dum <- sample( c("NJ","NY","MA","PA"), 100, replace=T )
```

basic regression syntax

```
lm( y ~ x )
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##     -0.8393       1.9148
```

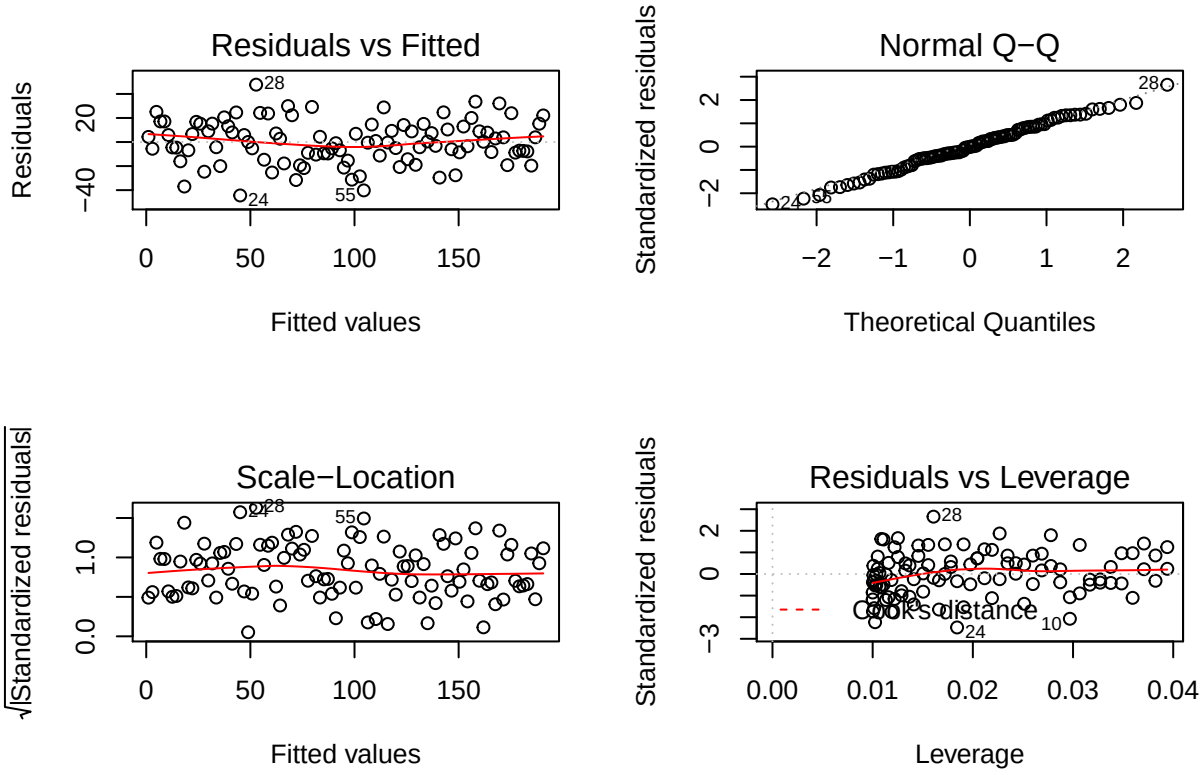
```
m.01 <- lm( y ~ x )

summary( m.01 )
```

```
##
## Call:
## lm(formula = y ~ x)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.433  -9.769  -0.195  13.403  47.705
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.83930    3.65219   -0.23    0.819
## x            1.91481    0.06279   30.50   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.12 on 98 degrees of freedom
## Multiple R-squared:  0.9047, Adjusted R-squared:  0.9037
## F-statistic: 930.1 on 1 and 98 DF,  p-value: < 2.2e-16
```

nice visual diagnostics

```
par( mfrow=c(2,2) )
plot( m.01 )
```



useful model fit functions

```
coefficients( m.01 ) # model coefficients
```

```
## (Intercept)           x
##  -0.8393018   1.9148119
```

```
confint( m.01, level=0.95) # CIs for model parameters
```

```
##                 2.5 %   97.5 %
## (Intercept) -8.086945 6.408341
## x            1.790213 2.039411
```

```
fitted( m.01 ) # predicted values
```

```
##          1          2          3          4          5          6
##   1.075510   2.990322   4.905134   6.819946   8.734758  10.649570
##          7          8          9         10         11         12
##  12.564382  14.479194  16.394006  18.308817  20.223629  22.138441
##         13         14         15         16         17         18
##  24.053253  25.968065  27.882877  29.797689  31.712501  33.627313
##         19         20         21         22         23         24
##  35.542125  37.456937  39.371749  41.286561  43.201372  45.116184
##         25         26         27         28         29         30
##  47.030996  48.945808  50.860620  52.775432  54.690244  56.605056
##         31         32         33         34         35         36
##  58.519868  60.434680  62.349492  64.264304  66.179116  68.093928
##         37         38         39         40         41         42
##  70.008739  71.923551  73.838363  75.753175  77.667987  79.582799
##         43         44         45         46         47         48
##  81.497611  83.412423  85.327235  87.242047  89.156859  91.071671
##         49         50         51         52         53         54
##  92.986483  94.901294  96.816106  98.730918 100.645730 102.560542
##         55         56         57         58         59         60
## 104.475354 106.390166 108.304978 110.219790 112.134602 114.049414
##         61         62         63         64         65         66
## 115.964226 117.879038 119.793850 121.708661 123.623473 125.538285
##         67         68         69         70         71         72
## 127.453097 129.367909 131.282721 133.197533 135.112345 137.027157
##         73         74         75         76         77         78
## 138.941969 140.856781 142.771593 144.686405 146.601216 148.516028
##         79         80         81         82         83         84
## 150.430840 152.345652 154.260464 156.175276 158.090088 160.004900
##         85         86         87         88         89         90
## 161.919712 163.834524 165.749336 167.664148 169.578960 171.493772
##         91         92         93         94         95         96
## 173.408583 175.323395 177.238207 179.153019 181.067831 182.982643
##         97         98         99        100
## 184.897455 186.812267 188.727079 190.641891
```

```r
residuals( m.01 ) # residuals
```

```
##           1           2           3           4           5
##  4.26437268 -5.58180953 25.08762706 17.15594141 17.10447527
##           6           7           8           9          10
##  5.78505228 -4.47905801 -4.66387337 -16.02412684 -37.00407796
##          11          12          13          14          15
## -6.85928252  6.61124503 16.80042617 15.23084997 -24.67561286
##          16          17          18          19          20
##  8.95843736 15.29583464 -4.31970061 -20.03016414 20.52026235
##          21          22          23          24          25
## 13.15462469  7.97575209 24.59196895 -44.43291214  5.80337080
##          26          27          28          29          30
##  0.04725515 -5.25453132 47.70525697 24.21675449 -14.72017748
##          31          32          33          34          35
## 23.71642149 -25.35022953  7.15265817  2.74876923 -17.81053326
##          36          37          38          39          40
## 29.88979552 22.27451630 -31.59652344 -19.35137110 -21.66448896
##          41          42          43          44          45
## -8.95350892 29.18060581 -10.46947806  4.35357396 -9.39690332
##          46          47          48          49          50
## -9.53524322 -5.24449096 -0.94634953 -6.87023446 -21.28598969
##          51          52          53          54          55
## -15.45706983 -31.32367136  6.85376699 -28.54605464 -40.23526225
##          56          57          58          59          60
## -0.55922175 14.55559103  0.86572862 -11.29474193 28.78771903
##          61          62          63          64          65
## -0.43728378  9.28653508 -5.02943353 -20.79407584 14.14955712
##          66          67          68          69          70
## -14.29175152  8.82310645 -18.95430573 -4.34390151 15.06892316
##          71          72          73          74          75
##  0.49430246  7.41510315 -3.18287528 -29.58955388 24.68807815
##          76          77          78          79          80
## 10.42590195 -6.01013153 -27.64765613 -8.58285693 12.89719650
##          81          82          83          84          85
## -3.51134416 20.02222168 33.58531805  8.89286093  0.23213854
##          86          87          88          89          90
##  7.81937576 -8.34659897  2.95560851 32.10280090  3.88250384
##          91          92          93          94          95
## -19.21796616 23.98062470 -8.83254553 -7.16077406 -7.38716433
##          96          97          98          99         100
## -7.81142122 -19.63356567  3.92576442 15.25282071 22.11250317
```

```r
anova( m.01 ) # anova table
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x          1 305512  305512  930.06 < 2.2e-16 ***
## Residuals 98  32191     328
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
influence( m.01 ) # regression diagnostics
```

```
## $hat
##          1          2          3          4          5          6
## 0.03940594 0.03822982 0.03707771 0.03594959 0.03484548 0.03376538
##          7          8          9         10         11         12
## 0.03270927 0.03167717 0.03066907 0.02968497 0.02872487 0.02778878
##         13         14         15         16         17         18
## 0.02687669 0.02598860 0.02512451 0.02428443 0.02346835 0.02267627
##         19         20         21         22         23         24
## 0.02190819 0.02116412 0.02044404 0.01974797 0.01907591 0.01842784
##         25         26         27         28         29         30
## 0.01780378 0.01720372 0.01662766 0.01607561 0.01554755 0.01504350
##         31         32         33         34         35         36
## 0.01456346 0.01410741 0.01367537 0.01326733 0.01288329 0.01252325
##         37         38         39         40         41         42
## 0.01218722 0.01187519 0.01158716 0.01132313 0.01108311 0.01086709
##         43         44         45         46         47         48
## 0.01067507 0.01050705 0.01036304 0.01024302 0.01014701 0.01007501
##         49         50         51         52         53         54
## 0.01002700 0.01000300 0.01000300 0.01002700 0.01007501 0.01014701
##         55         56         57         58         59         60
## 0.01024302 0.01036304 0.01050705 0.01067507 0.01086709 0.01108311
##         61         62         63         64         65         66
## 0.01132313 0.01158716 0.01187519 0.01218722 0.01252325 0.01288329
##         67         68         69         70         71         72
## 0.01326733 0.01367537 0.01410741 0.01456346 0.01504350 0.01554755
##         73         74         75         76         77         78
## 0.01607561 0.01662766 0.01720372 0.01780378 0.01842784 0.01907591
##         79         80         81         82         83         84
## 0.01974797 0.02044404 0.02116412 0.02190819 0.02267627 0.02346835
##         85         86         87         88         89         90
## 0.02428443 0.02512451 0.02598860 0.02687669 0.02778878 0.02872487
##         91         92         93         94         95         96
## 0.02968497 0.03066907 0.03167717 0.03270927 0.03376538 0.03484548
##         97         98         99        100
## 0.03594959 0.03707771 0.03822982 0.03940594
##
## $coefficients
##       (Intercept)             x
## 1    1.775723e-01 -2.637213e-03
## 2   -2.286299e-01  3.378081e-03
## 3    1.010565e+00 -1.485206e-02
## 4    6.794718e-01 -9.930988e-03
## 5    6.659178e-01 -9.677184e-03
## 6    2.213454e-01 -3.197491e-03
## 7   -1.683825e-01  2.417373e-03
## 8   -1.722244e-01  2.456632e-03
## 9   -5.810939e-01  8.233322e-03
## 10  -1.317431e+00  1.853602e-02
## 11  -2.396848e-01  3.347790e-03
## 12   2.266738e-01 -3.142013e-03
## 13   5.650179e-01 -7.769774e-03
```

```
## 14   5.022871e-01 -6.849796e-03
## 15  -7.976974e-01  1.078380e-02
## 16   2.837888e-01 -3.801481e-03
## 17   4.746494e-01 -6.297328e-03
## 18  -1.312585e-01  1.723944e-03
## 19  -5.957474e-01  7.741767e-03
## 20   5.971548e-01 -7.673571e-03
## 21   3.743890e-01 -4.754402e-03
## 22   2.219026e-01 -2.782937e-03
## 23   6.685388e-01 -8.273995e-03
## 24  -1.179688e+00  1.439637e-02
## 25   1.503999e-01 -1.808202e-03
## 26   1.194773e-03 -1.413762e-05
## 27  -1.295365e-01  1.506984e-03
## 28   1.146001e+00 -1.309217e-02
## 29   5.665273e-01 -6.347231e-03
## 30  -3.351304e-01  3.676838e-03
## 31   5.250964e-01 -5.632222e-03
## 32  -5.454267e-01  5.708851e-03
## 33   1.494316e-01 -1.523037e-03
## 34   5.571457e-02 -5.516294e-04
## 35  -3.499246e-01  3.356331e-03
## 36   5.686877e-01 -5.267308e-03
## 37   4.099878e-01 -3.653357e-03
## 38  -5.620068e-01  4.796917e-03
## 39  -3.322366e-01  2.702066e-03
## 40  -3.585700e-01  2.761265e-03
## 41  -1.426668e-01  1.032243e-03
## 42   4.469878e-01 -3.009423e-03
## 43  -1.539265e-01  9.525154e-04
## 44   6.133059e-02 -3.432189e-04
## 45  -1.266040e-01  6.267527e-04
## 46  -1.226136e-01  5.202839e-04
## 47  -6.422124e-02  2.225489e-04
## 48  -1.100827e-02  2.868230e-05
## 49  -7.570713e-02  1.249293e-04
## 50  -2.215261e-01  1.290193e-04
## 51  -1.514012e-01 -9.368886e-05
## 52  -2.876449e-01 -5.695938e-04
## 53   5.874503e-02  2.077264e-04
## 54  -2.272139e-01 -1.211346e-03
## 55  -2.956484e-01 -2.195409e-03
## 56  -3.767184e-03 -3.729886e-05
## 57   8.915243e-02  1.147507e-03
## 58   4.773109e-03  7.876418e-05
## 59  -5.536403e-02 -1.164837e-03
## 60   1.234985e-01  3.318912e-03
## 61  -1.608334e-03 -5.573435e-05
## 62   2.847091e-02  1.296695e-03
## 63  -1.233910e-02 -7.635579e-04
## 64  -3.827386e-02 -3.410542e-03
## 65   1.736849e-02  2.493496e-03
## 66  -8.774715e-03 -2.693229e-03
## 67  -6.661338e-17  1.770641e-03
```

```
## 68   1.164673e-02 -4.035996e-03
## 69   5.340678e-03 -9.782431e-04
## 70  -2.780295e-02  3.578597e-03
## 71  -1.216611e-03  1.234680e-04
## 72  -2.282488e-02  1.943505e-03
## 73   1.176319e-02 -8.735044e-04
## 74   1.276540e-01 -8.486194e-03
## 75  -1.217951e-01  7.386089e-03
## 76  -5.789938e-02  3.248480e-03
## 77   3.710888e-02 -1.947297e-03
## 78   1.879021e-01 -9.302085e-03
## 79   6.367830e-02 -2.994771e-03
## 80  -1.037350e-01  4.661361e-03
## 81   3.043740e-02 -1.313070e-03
## 82  -1.860973e-01  7.738698e-03
## 83  -3.332323e-01  1.340353e-02
## 84  -9.382534e-02  3.661210e-03
## 85  -2.595449e-03  9.850716e-05
## 86  -9.236184e-02  3.417244e-03
## 87   1.038703e-01 -3.753730e-03
## 88  -3.865577e-02  1.366894e-03
## 89  -4.402720e-01  1.525695e-02
## 90  -5.572031e-02  1.894922e-03
## 91   2.880859e-01 -9.626632e-03
## 92  -3.748388e-01  1.232143e-02
## 93   1.437325e-01 -4.652424e-03
## 94   1.211387e-01 -3.864710e-03
## 95   1.297386e-01 -4.083004e-03
## 96   1.422484e-01 -4.419461e-03
## 97   3.702855e-01 -1.136520e-02
## 98  -7.659682e-02  2.324081e-03
## 99  -3.075707e-01  9.230926e-03
## 100 -4.603923e-01  1.367502e-02
##
## $sigma
##        1        2        3        4        5        6        7        8
## 18.21197 18.20815 18.03143 18.13073 18.13135 18.20752 18.21145 18.21097
##        9       10       11       12       13       14       15       16
## 18.14222 17.81355 18.20361 18.20460 18.13507 18.14981 18.03973 18.19404
##       17       18       19       20       21       22       23       24
## 18.14941 18.21192 18.10089 18.09519 18.16727 18.19895 18.04203 17.63903
##       25       26       27       28       29       30       31       32
## 18.20762 18.21732 18.20938 17.55066 18.04798 18.15497 18.05510 18.03194
##       33       34       35       36       37       38       39       40
## 18.20264 18.21516 18.12617 17.95950 18.07464 17.92917 18.10980 18.08250
##       41       42       43       44       45       46       47       48
## 18.19437 17.97209 18.18595 18.21190 18.19206 18.19131 18.20946 18.21707
##       49       50       51       52       53       54       55       56
## 18.20383 18.08736 18.14891 17.93469 18.20389 17.98288 17.74848 18.21723
##       57       58       59       60       61       62       63       64
## 18.15664 18.21711 18.18079 17.97864 18.21727 18.19262 18.21008 18.09304
##       65       66       67       68       69       70       71       72
## 18.15986 18.15868 18.19499 18.11397 18.21191 18.15201 18.21725 18.20151
##       73       74       75       76       77       78       79       80
```

```
## 18.21441 17.96363 18.04099 18.18598 18.20691 17.99548 18.19605 18.16921
##       81       82       83       84       85       86       87       88
## 18.21376 18.10098 17.88777 18.19439 18.21731 18.19957 18.19707 18.21478
##       89       90       91       92       93       94       95       96
## 17.91487 18.21293 18.10930 18.04868 18.19451 18.20232 18.20134 18.19943
##       97       98       99      100
## 18.10383 18.21279 18.14875 18.07272
##
## $wt.res
##            1            2            3            4            5
##   4.26437268  -5.58180953  25.08762706  17.15594141  17.10447527
##            6            7            8            9           10
##   5.78505228  -4.47905801  -4.66387337 -16.02412684 -37.00407796
##           11           12           13           14           15
##  -6.85928252   6.61124503  16.80042617  15.23084997 -24.67561286
##           16           17           18           19           20
##   8.95843736  15.29583464  -4.31970061 -20.03016414  20.52026235
##           21           22           23           24           25
##  13.15462469   7.97575209  24.59196895 -44.43291214   5.80337080
##           26           27           28           29           30
##   0.04725515  -5.25453132  47.70525697  24.21675449 -14.72017748
##           31           32           33           34           35
##  23.71642149 -25.35022953   7.15265817   2.74876923 -17.81053326
##           36           37           38           39           40
##  29.88979552  22.27451630 -31.59652344 -19.35137110 -21.66448896
##           41           42           43           44           45
##  -8.95350892  29.18060581 -10.46947806   4.35357396  -9.39690332
##           46           47           48           49           50
##  -9.53524322  -5.24449096  -0.94634953  -6.87023446 -21.28598969
##           51           52           53           54           55
## -15.45706983 -31.32367136   6.85376699 -28.54605464 -40.23526225
##           56           57           58           59           60
##  -0.55922175  14.55559103   0.86572862 -11.29474193  28.78771903
##           61           62           63           64           65
##  -0.43728378   9.28653508  -5.02943353 -20.79407584  14.14955712
##           66           67           68           69           70
## -14.29175152   8.82310645 -18.95430573  -4.34390151  15.06892316
##           71           72           73           74           75
##   0.49430246   7.41510315  -3.18287528 -29.58955388  24.68807815
##           76           77           78           79           80
##  10.42590195  -6.01013153 -27.64765613  -8.58285693  12.89719650
##           81           82           83           84           85
##  -3.51134416  20.02222168  33.58531805   8.89286093   0.23213854
##           86           87           88           89           90
##   7.81937576  -8.34659897   2.95560851  32.10280090   3.88250384
##           91           92           93           94           95
## -19.21796616  23.98062470  -8.83254553  -7.16077406  -7.38716433
##           96           97           98           99          100
##  -7.81142122 -19.63356567   3.92576442  15.25282071  22.11250317
```

pretty output

```
# install.packages( "memisc" )

library( memisc )
```

```
## Loading required package: lattice
##
## Attaching package: 'lattice'
##
## The following object is masked from 'package:boot':
##
##     melanoma
##
## Loading required package: MASS
##
## Attaching package: 'memisc'
##
## The following objects are masked from 'package:stats':
##
##     contr.sum, contr.treatment, contrasts
##
## The following objects are masked from 'package:base':
##
##     as.array, trimws
```

```
m.02 <- lm( y ~ x + I(x^2) )  # quadratic term
m.03 <- lm( y ~ x - 1 )       # no intercept term


pretty.table <- mtable("Model 1"=m.01,"Model 2"=m.02,"Model 3"=m.03,
                  summary.stats=c("R-squared","F","p","N"))

pretty.table
```

```
##
## Calls:
## Model 1: lm(formula = y ~ x)
## Model 2: lm(formula = y ~ x + I(x^2))
## Model 3: lm(formula = y ~ x - 1)
##
## ============================================
##                Model 1   Model 2   Model 3
## --------------------------------------------
## (Intercept)   -0.839     5.706
##               (3.652)   (5.506)
## x              1.915***  1.530***  1.902***
##               (0.063)   (0.252)   (0.031)
## I(x^2)                   0.004
##                         (0.002)
## --------------------------------------------
```

```
## R-squared       0.905     0.907     0.974
## F             930.064   473.369  3763.377
## p               0.000     0.000     0.000
## N             100       100       100
## ========================================
```

specification

```
summary( lm( y ~ x1 + x2 + x3 ) )
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.544  -9.404  -0.020  12.648  48.194
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.789562   3.689871  -0.214    0.831
## x1           1.830865   0.209591   8.735 7.72e-14 ***
## x2          -0.852292   2.033850  -0.419    0.676
## x3           0.009895   1.741971   0.006    0.995
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.29 on 96 degrees of freedom
## Multiple R-squared:  0.9049, Adjusted R-squared:  0.9019
## F-statistic: 304.3 on 3 and 96 DF,  p-value: < 2.2e-16
```

```
# add different functional forms

summary( lm( y ~ x1^2 + x2 + x3 ) )  # not right
```

```
##
## Call:
## lm(formula = y ~ x1^2 + x2 + x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.544  -9.404  -0.020  12.648  48.194
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.789562   3.689871  -0.214    0.831
## x1           1.830865   0.209591   8.735 7.72e-14 ***
## x2          -0.852292   2.033850  -0.419    0.676
## x3           0.009895   1.741971   0.006    0.995
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.29 on 96 degrees of freedom
## Multiple R-squared:  0.9049, Adjusted R-squared:  0.9019
## F-statistic: 304.3 on 3 and 96 DF,  p-value: < 2.2e-16
```

```
summary( lm( y ~ I(x1^2) + x2 + x3 ) )  # like this
```

```
##
## Call:
## lm(formula = y ~ I(x1^2) + x2 + x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.423 -12.682   0.363  12.163  54.002
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 22.503229   4.238992   5.309 7.11e-07 ***
## I(x1^2)      0.011954   0.001631   7.329 7.25e-11 ***
## x2          -6.703759   1.650013  -4.063 9.91e-05 ***
## x3           0.645330   1.863111   0.346     0.73
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.63 on 96 degrees of freedom
## Multiple R-squared:  0.8905, Adjusted R-squared:  0.8871
## F-statistic: 260.2 on 3 and 96 DF,  p-value: < 2.2e-16
```

```
summary( lm( y ~ log(x1) + x2 + x3 ) )
```

```
##
## Call:
## lm(formula = y ~ log(x1) + x2 + x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -58.718 -15.133   0.106  13.589  62.191
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -14.870     13.417  -1.108   0.2705
## log(x1)       10.197      5.477   1.862   0.0657 .
## x2           -14.989      1.706  -8.789 5.93e-14 ***
## x3             0.738      2.326   0.317   0.7517
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.08 on 96 degrees of freedom
## Multiple R-squared:  0.8352, Adjusted R-squared:   0.83
## F-statistic: 162.1 on 3 and 96 DF,  p-value: < 2.2e-16
```

```r
# interactions

summary( lm( y ~ x1 + x2 ) )
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.544  -9.416  -0.013  12.640  48.200
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.7890     3.6695  -0.215    0.830
## x1            1.8310     0.2074   8.829 4.51e-14 ***
## x2           -0.8507     2.0051  -0.424    0.672
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.2 on 97 degrees of freedom
## Multiple R-squared:  0.9049, Adjusted R-squared:  0.9029
## F-statistic: 461.2 on 2 and 97 DF,  p-value: < 2.2e-16
```

```r
summary( lm( y ~ x1 + x2 + I(x1*x2) ) )
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + I(x1 * x2))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -43.64 -11.51  -1.10  11.96  49.12
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.91146    5.22828   0.939    0.350
## x1           1.62866    0.24521   6.642 1.86e-09 ***
## x2           0.58021    2.20271   0.263    0.793
## I(x1 * x2)  -0.03497    0.02299  -1.521    0.132
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.08 on 96 degrees of freedom
## Multiple R-squared:  0.9071, Adjusted R-squared:  0.9042
## F-statistic: 312.4 on 3 and 96 DF,  p-value: < 2.2e-16
```

```r
summary( lm( y ~ x1*x2 ) ) # shortcut
```

```
##
## Call:
## lm(formula = y ~ x1 * x2)
```

```
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -43.64 -11.51  -1.10  11.96  49.12
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.91146    5.22828   0.939    0.350
## x1           1.62866    0.24521   6.642 1.86e-09 ***
## x2           0.58021    2.20271   0.263    0.793
## x1:x2       -0.03497    0.02299  -1.521    0.132
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.08 on 96 degrees of freedom
## Multiple R-squared:  0.9071, Adjusted R-squared:  0.9042
## F-statistic: 312.4 on 3 and 96 DF,  p-value: < 2.2e-16
```

```
# dummy variables

summary( lm( y ~ x1 + x2 + x3 + dum ) ) # drop one level
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + dum)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.470 -10.882   0.351  10.218  43.816
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.9449     5.0615   0.187    0.852
## x1            1.8075     0.2104   8.591 1.95e-13 ***
## x2           -1.1387     2.0584  -0.553    0.581
## x3           -0.3666     1.7637  -0.208    0.836
## dumNJ        -5.7247     5.5681  -1.028    0.307
## dumNY        -6.8535     5.1482  -1.331    0.186
## dumPA         2.9278     5.0815   0.576    0.566
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.1 on 93 degrees of freedom
## Multiple R-squared:  0.9098, Adjusted R-squared:  0.904
## F-statistic: 156.4 on 6 and 93 DF,  p-value: < 2.2e-16
```

```
summary( lm( y ~ x1 + x2 + x3 + dum - 1) ) # keep all, drop intercept
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + dum - 1)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -43.470 -10.882   0.351  10.218  43.816
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## x1      1.8075     0.2104   8.591 1.95e-13 ***
## x2     -1.1387     2.0584  -0.553    0.581
## x3     -0.3666     1.7637  -0.208    0.836
## dumMA   0.9449     5.0615   0.187    0.852
## dumNJ  -4.7798     5.6585  -0.845    0.400
## dumNY  -5.9085     4.6037  -1.283    0.203
## dumPA   3.8727     4.4608   0.868    0.388
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.1 on 93 degrees of freedom
## Multiple R-squared:  0.9758, Adjusted R-squared:  0.9739
## F-statistic: 534.9 on 7 and 93 DF,  p-value: < 2.2e-16
```

standardized regression coefficients (beta)

```
# install.packages( "lm.beta" )

coefficients( m.01 )
```

```
## (Intercept)           x
##  -0.8393018   1.9148119
```

```
library( lm.beta )

m.01.beta <- lm.beta( m.01 )
print( m.01.beta )
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Standardized Coefficients::
## (Intercept)           x
##   0.0000000   0.9511441
```

```
summary( m.01.beta )
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -44.433  -9.769  -0.195  13.403  47.705
##
## Coefficients:
##             Estimate Standardized Std. Error t value Pr(>|t|)
## (Intercept) -0.83930      0.00000    3.65219   -0.23    0.819
## x            1.91481      0.95114    0.06279   30.50   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.12 on 98 degrees of freedom
## Multiple R-squared:  0.9047, Adjusted R-squared:  0.9037
## F-statistic: 930.1 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
coef( m.01.beta )
```

```
## (Intercept)           x
##   0.0000000   0.9511441
```

```
# note the standard error is for the normal slope coefficients

summary( m.01 )
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.433  -9.769  -0.195  13.403  47.705
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.83930    3.65219   -0.23    0.819
## x            1.91481    0.06279   30.50   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.12 on 98 degrees of freedom
## Multiple R-squared:  0.9047, Adjusted R-squared:  0.9037
## F-statistic: 930.1 on 1 and 98 DF,  p-value: < 2.2e-16
```

or just use the formula:

```
lm.beta <- function( my.mod )
{
    b <- summary(my.mod)$coef[-1, 1]
    sx <- sd( my.mod$model[,-1] )
    sy <- sd( my.mod$model[,1] )
    beta <- b * sx/sy
    return(beta)
```

```
}


coefficients( m.01 )
```

```
## (Intercept)           x
##  -0.8393018   1.9148119
```

```
lm.beta( m.01 )
```

```
## [1] 0.9511441
```

robust standard errors

```
# install.packages( "sandwhich" )
# install.packages( "lmtest" )

library(sandwich)
library(lmtest)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
m.01 <- lm( y ~ x )

summary( m.01 )                                    # non-robust
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.433  -9.769  -0.195  13.403  47.705
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.83930    3.65219   -0.23    0.819
## x            1.91481    0.06279   30.50   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.12 on 98 degrees of freedom
## Multiple R-squared:  0.9047, Adjusted R-squared:  0.9037
## F-statistic: 930.1 on 1 and 98 DF,  p-value: < 2.2e-16
```

```r
# reproduce the Stata default
coeftest( m.01, vcov=vcovHC(m.01,"HC1") )      # robust; HC1 (Stata default)
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.839302   3.683322 -0.2279   0.8202
## x            1.914812   0.058823 32.5519   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# check that "sandwich" returns HC0
coeftest(m.01, vcov = sandwich)                # robust; sandwich
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.839302   3.646303 -0.2302   0.8184
## x            1.914812   0.058232 32.8824   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
coeftest(m.01, vcov = vcovHC(m.01, "HC0"))     # robust; HC0
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.839302   3.646303 -0.2302   0.8184
## x            1.914812   0.058232 32.8824   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# check that the default robust var-cov matrix is HC3
coeftest(m.01, vcov = vcovHC(m.01))            # robust; HC3
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.839302   3.734206 -0.2248   0.8226
## x            1.914812   0.059813 32.0135   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
coeftest(m.01, vcov = vcovHC(m.01, "HC3"))     # robust; HC3 (default)
```

```
##
```

```
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.839302   3.734206 -0.2248   0.8226
## x            1.914812   0.059813 32.0135   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```