An alternative modeling strategy: Partial Least Squares

Dr. Jon Starkweather, Research and Statistical Support consultant.

Partial Least Squares (PLS) modeling is often used as an alternative to traditional modeling techniques. Unlike traditional modeling techniques which rely upon covariance decomposition, PLS is a variance based (or components based) technique and does not carry with it many of the assumptions of covariance methods (i.e. distributional assumptions). It is sometimes considered an analysis of last resort because large samples are not *as* necessary with it, and PLS is less sensitive to multicollinearity. However, PLS is primarily descriptive when used with small samples and is still constrained with respect to making inferences about parameters when sample sizes are small. The benefit of having the ability to do descriptive analysis with small samples is that PLS can fit models with non-linear relationships and non-Gaussian distributions among the variables in addition to the traditional linear and Gaussian situations.

PLS is also quite versatile; it can be used as a regression technique, a principal components technique, a canonical correlation technique, or a path modeling (or structural equation modeling) technique. It is well documented that PLS is biased because the optimization is local rather than global level; however, as sample size increases PLS becomes less bias. PLS can be used to make inferences about parameters when sample sizes are large. PLS is often used when other methods fail (i.e. a slightly biased estimate is better than no estimate).

As an example, we will first model a simulated data set using traditional modeling techniques using a popular method and package. John Fox's (2010) package 'sem' is one of the more established modeling packages in R and will be used here to demonstrate how certain data sets do not converge on a specified model.

Example

First, import the data from the internet and run the ubiquitous 'head' function to get a look at the data. The example data contains 20 variables (v1 – v20) and 1000 cases. Here we will name the data 'pls.data'.

```
R Console
File  Edit  Misc  Packages  Windows  Help

R version 2.13.0 (2011-04-13)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: i386-pc-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> pls.data <- read.table("http://www.unt.edu/rss/class/Jon/R_SC/Module8/PLSdata001.txt",
+    header=TRUE, sep=",", na.strings="NA", dec=".", strip.white=TRUE)
> cov.m <- cov(pls.data[,4:23])
> head(pls.data)
  id sex age        v1       v2       v3       v4       v5       v6       v7       v8
1  1   1  30 131.11743 59.51358 25.94516 73.32716 57.45567 46.63123 44.55832 63.80995
2  2   2  27  87.99215 48.39060 31.23116 92.53837 79.06543 48.73282 38.40678 60.67488
3  3   2  36  94.41560 52.77620 21.57114 63.59395 45.64394 29.87916 23.38897 52.36779
4  4   2  27  80.61290 46.07760 30.65801 71.62971 65.66272 33.55456 35.87245 44.09138
5  5   1  27 107.70663 58.61991 25.66450 86.68273 74.67492 44.21711 42.31820 54.76483
6  6   1  35 129.69572 60.33568 27.04439 72.06893 71.33984 54.55930 36.40072 54.02113
        v9      v10      v11       v12      v13      v14      v15      v16       v17
1 24.72150 24.46817 60.57330 101.03496 50.77800 22.99464 23.08530 50.23240  97.58174
2 24.98940 22.37192 51.07190  96.25460 61.19011 22.65629 23.96417 48.62368  93.85276
3 16.15046 18.29899 42.41373  62.57547 37.41148 14.95853 18.57941 35.64330  63.71728
4 17.38695 16.32546 36.46037  88.30067 40.23851 20.07501 20.10772 42.83935  69.43062
5 22.91687 20.71210 49.89569  90.71575 60.32451 22.93967 22.01327 47.12148 100.03622
6 25.82321 25.31235 59.92794  71.97986 58.56063 23.23412 23.16468 55.03165  94.96685
       v18      v19      v20
1 56.05315 22.10798 60.62849
2 64.78140 23.94896 45.84800
3 38.33542 15.77232 36.45215
4 46.78583 16.48520 35.31118
5 61.12789 23.67255 45.03366
6 61.39238 25.06159 52.53633
> |
```

Next, create a covariance matrix object which will be passed on to the ‘sem’ function. The covariance object is named ‘cov.m’ (some of the matrix in the image below is not shown).

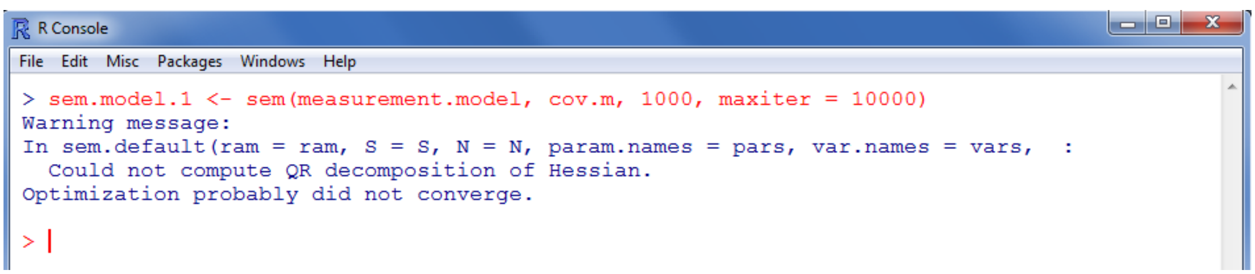
```
R Console
File  Edit  Misc  Packages  Windows  Help
> cov.m <- cov(pls.data[,4:23])
> cov.m
             v1         v2         v3         v4         v5        v6        v7        v8
v1  225.0000000 82.4458932  0.6275042  -4.162772  0.3626107 51.509785 38.632938 50.610755
v2   82.4458932 56.2500000  0.7103732  -2.017296  0.3207956 22.841821 16.155788 23.143522
v3    0.6275042  0.7103732 14.0625000  18.687207 21.1108057  8.221973  5.876501  9.405605
v4   -4.1627721 -2.0172958 18.6872073 126.562500 55.3054256 18.944608 15.544449 23.921660
v5    0.3626107  0.3207956 21.1108057  55.305426 81.0000000 24.485383 18.944014 26.426809
v6   51.5097852 22.8418206  8.2219733  18.944608 24.4853834 36.000000 18.037645 24.398476
v7   38.6329385 16.1557877  5.8765015  15.544449 18.9440143 18.037645 27.562500 18.649779
v8   50.6107546 23.1435219  9.4056053  23.921660 26.4268095 24.398476 18.649779 68.062500
v9   28.4324332 12.8101526  4.3032578  10.776141 13.8760229 13.117043  9.745600 12.898916
v10  25.4395190 11.2221589  3.7345258  10.504763 12.4496514 11.540798  8.725929 11.730134
v11  49.0496078 21.3374561  7.3816034  19.197224 23.9037955 22.998952 17.137271 23.077786
v12  45.2919591 19.2186731 18.5226206  52.908010 59.9266838 33.763836 26.395675 34.575677
v13  33.4545605 14.9415087 17.3464097  45.163672 52.9947690 29.338479 22.189366 29.937668
v14  13.4705705  5.5531698  6.2699226  16.294190 19.8686515 10.878815  8.473193 11.089990
v15  12.0616551  5.3743438  5.6847087  15.137249 17.7541376 10.094192  7.445060  9.969569
v16  40.6980194 18.0731626 10.7690379  26.946218 34.4143007 23.076664 17.203529 24.729194
v17  93.8815359 41.6372802 24.2968475  62.508273 75.3421656 53.322918 40.600596 55.096656
v18  53.2273603 23.4027734 14.4123724  37.815029 45.2801612 30.935149 23.954879 32.875793
v19  19.7264396  8.3296843  5.3515471  12.940903 16.2033235 11.117326  8.455361 11.211828
v20  41.0924684 17.4708487 10.8094127  27.265314 32.5463932 23.337671 18.631685 25.108573
           v9       v10       v11      v12      v13       v14       v15      v16
v1  28.432433 25.439519 49.049608 45.29196 33.45456 13.470570 12.061655 40.69802
v2  12.810153 11.222159 21.337456 19.21867 14.94151  5.553170  5.374344 18.07316
v3   4.303258  3.734526  7.381603 18.52262 17.34641  6.269923  5.684709 10.76904
```

Next, load the ‘sem’ package by typing: `library(sem)` in the R console. Then, specify the sem measurement model (i.e. confirmatory factor model). The model specification syntax is given below (not in an image) due to its length.

```
measurement.model <- specify.model()
F1 -> v1, lam11, NA
F1 -> v2, lam12, NA
F2 -> v3, lam21, NA
F2 -> v4, lam22, NA
F2 -> v5, lam23, NA
F3 -> v6, lam31, NA
F3 -> v7, lam32, NA
F3 -> v8, lam33, NA
F3 -> v9, lam34, NA
F3 -> v10, lam35, NA
F3 -> v11, lam36, NA
F4 -> v12, lam41, NA
F4 -> v13, lam42, NA
F4 -> v14, lam43, NA
F4 -> v15, lam44, NA
F5 -> v16, lam51, NA
F5 -> v17, lam52, NA
F5 -> v18, lam53, NA
F5 -> v19, lam54, NA
F5 -> v20, lam55, NA
v1 <-> v1, var1, NA
v2 <-> v2, var2, NA
v3 <-> v3, var3, NA
v4 <-> v4, var4, NA
v5 <-> v5, var5, NA
v6 <-> v6, var6, NA
```

```
v7 <-> v7, var7, NA
v8 <-> v8, var8, NA
v9 <-> v9, var9, NA
v10 <-> v10, var10, NA
v11 <-> v11, var11, NA
v12 <-> v12, var12, NA
v13 <-> v13, var13, NA
v14 <-> v14, var14, NA
v15 <-> v15, var15, NA
v16 <-> v16, var16, NA
v17 <-> v17, var17, NA
v18 <-> v18, var18, NA
v19 <-> v19, var19, NA
v20 <-> v20, var20, NA
F1 <-> F2, cov1, NA
F1 <-> F3, cov2, NA
F1 <-> F4, cov3, NA
F1 <-> F5, cov4, NA
F2 <-> F3, cov5, NA
F2 <-> F4, cov6, NA
F2 <-> F5, cov7, NA
F3 <-> F4, cov8, NA
F3 <-> F5, cov9, NA
F4 <-> F5, cov10, NA
F1 <-> F1, NA, 1
F2 <-> F2, NA, 1
F3 <-> F3, NA, 1
F4 <-> F4, NA, 1
F5 <-> F5, NA, 1
```

Next, we run the measurement model; but unfortunately, it does not converge.



So, we detach the 'sem' package using the following command: `detach("package:sem")` and decide to use a PLS strategy. The 'plspm' package (PLS Path Modeling; [Sanchez & Trinchera, 2010]) provides functions for conducting and graphing a variety of PLS techniques; such as PLS regression with a single outcome, PLS canonical correlation, PLS regression with multiple outcomes (similar to canonical correlation, but with directionality implied between the two composite variates), PLS principal components analysis, and PLS path modeling (i.e. SEM).

PLS Path Modeling

Load the package (which three dependencies [amap, diagram, shape]).

```
R Console
File  Edit  Misc  Packages  Windows  Help  Vignettes
> library(plspm)
Loading required package: amap
Loading required package: diagram
Loading required package: shape
>
```

First, we must create a matrix which expresses the *inner* (structural) model; this model simply shows the relationships among the latent variables; where the column variable 'causes' the row variable(s) if a 'one' is in the intersecting cell (e.g. f1 and f2 cause f3 --> columns 1 and 2 cause row 3).

```
R Console
File  Edit  Misc  Packages  Windows  Help  Vignettes
> inner.matrix <- matrix(c(0, 0, 0, 0, 0,
+                          0, 0, 0, 0, 0,
+                          1, 1, 0, 0, 0,
+                          0, 1, 1, 0, 0,
+                          0, 0, 1, 1, 0), 5, 5, byrow = TRUE)
> dimnames(inner.matrix) <- list(c("f1", "f2", "f3", "f4", "f5"),
+                                c("f1", "f2", "f3", "f4", "f5"))
> inner.matrix
   f1 f2 f3 f4 f5
f1  0  0  0  0  0
f2  0  0  0  0  0
f3  1  1  0  0  0
f4  0  1  1  0  0
f5  0  0  1  1  0
>
```

Next, create the list which expresses the outer (measurement) model; this model simply shows the relationships between the manifest variables and the latent variables (e.g. variables v1 and v2 are related to the first factor [f1]). Although we create a *list* object in R, this is often referred to as the outer *matrix* in the PLS literature.

```
R Console
File  Edit  Misc  Packages  Windows  Help  Vignettes
> outer.list <- list(c(1,2), c(3,4,5), c(6,7,8,9,10,11), c(12,13,14,15), c(16,17,18,19,20$
> outer.list
[[1]]
[1] 1 2

[[2]]
[1] 3 4 5

[[3]]
[1]  6  7  8  9 10 11

[[4]]
[1] 12 13 14 15

[[5]]
[1] 16 17 18 19 20

>
```

Next, create a vector which identifies the "mode" of indicators which were used (i.e. "A" for reflective measurement or "B" for formative measurement). Recall, 'Reflective' measurement is said to occur when each manifest variable is "caused by" a latent variable and 'Formative' measurement is said to occur when each manifest variable "causes" the latent variable. Below, all 5 latent variables in our model are "reflectively" measured (i.e. each latent causes the observed scores on the manifest variables).

```
R Console
File  Edit  Misc  Packages  Windows  Help  Vignettes
> mode.vec <- c("A", "A", "A", "A", "A")
> mode.vec
[1] "A" "A" "A" "A" "A"
> |
```

Finally, we can run the Partial Least Squares Path Model. One of the benefits of using the ‘plspm’ package rather than one of the other PLS packages available in R, is that the ‘plspm’ package offers some very easy to use and interpret output. Each function provides a description of the function’s output items and shows how to extract or reference them.

```
R Console
File  Edit  Misc  Packages  Windows  Help  Vignettes
> pls.model.1 <- plspm(x = pls.data[,4:23], inner = inner.matrix, outer = outer.list, mod$
+    scheme = "factor", scaled = TRUE, plsr = TRUE, tol = 0.00001, iter = 100)
> pls.model.1

PARTIAL LEAST SQUARES PATH MODELING (PLS-PM)
----------------------------------------------
Results available in the following objects:

     Name              Description
1    "$outer.mod"      "outer model"
2    "$inner.mod"      "inner model"
3    "$latents"        "scaled LVs"
4    "$scores"         "LVs for scaled=FALSE"
5    "$out.weights"    "outer weights"
6    "$loadings"       "loadings"
7    "$path.coefs"     "path coefficients matrix"
8    "$r.sqr"          "R-squared"
9    "$outer.cor"      "outer correlations"
10   "$inner.sum"      "summary inner model"
11   "$effects"        "total effects"
12   "$unidim"         "unidimensionality"
13   "$gof"            "goodnes-of-fit"
14   "$data"           "data matrix"
----------------------------------------------
You can also use the function 'summary'

> |
```

Using the ‘summary’ function on a ‘plspm’ object provides a well-documented and indexed summary of the analysis’ output. Below you can see that the current summary provides a very thorough summary with labels for each element which makes interpretation very straighforward. In fact, the output (from the ‘summary’) is so large that is necessitates four screen capture images to display it all here.

```
> summary(pls.model.1)

PARTIAL LEAST SQUARES PATH MODELING (PLS-PM)

----------------------------------------------------------
MODEL SPECIFICATION
1   Number of Cases      1000
2   Latent Variables     5
3   Manifest Variables   20
4   Scale of Data        Standardized Data
5   Weighting Scheme     factor
6   Tolerance Crit       1e-05
7   Max Num Iters        100
8   Convergence Iters    3
9   Paths by PLS-R       TRUE
10  Bootstrapping        FALSE
11  Bootstrap samples    NULL

----------------------------------------------------------
BLOCKS DEFINITION
    Block         Type   NMVs         Mode
1      f1    Exogenous      2   Reflective
2      f2    Exogenous      3   Reflective
3      f3   Endogenous      6   Reflective
4      f4   Endogenous      4   Reflective
5      f5   Endogenous      5   Reflective

----------------------------------------------------------
BLOCKS UNIDIMENSIONALITY
    Type.measure  MVs  C.alpha  DG.rho  eig.1st  eig.2nd
f1    Reflective    2    0.846   0.928     1.73    0.267
f2    Reflective    3    0.778   0.871     2.08    0.566
f3    Reflective    6    0.885   0.913     3.83    0.618
f4    Reflective    4    0.890   0.924     3.02    0.472
f5    Reflective    5    0.918   0.939     3.78    0.431

----------------------------------------------------------
OUTER MODEL
       weights  std.loads  communal  redundan
f1
  v1     0.570      0.940     0.883     0.000
  v2     0.504      0.922     0.849     0.000
f2
  v3     0.374      0.826     0.682     0.000
  v4     0.328      0.761     0.579     0.000
  v5     0.489      0.902     0.814     0.000
f3
  v6     0.222      0.846     0.716     0.531
  v7     0.192      0.760     0.578     0.429
  v8     0.168      0.673     0.453     0.336
  v9     0.251      0.896     0.803     0.596
  v10    0.221      0.838     0.703     0.521
  v11    0.190      0.759     0.576     0.427
```

Continued below.

R Console

File Edit Misc Packages Windows Help Vignettes

```
  v9     0.251      0.896     0.803     0.596
  v10    0.221      0.838     0.703     0.521
  v11    0.190      0.759     0.576     0.427
f4
  v12    0.238      0.781     0.611     0.495
  v13    0.328      0.923     0.852     0.690
  v14    0.305      0.905     0.820     0.664
  v15    0.274      0.856     0.733     0.594
f5
  v16    0.206      0.814     0.663     0.577
  v17    0.263      0.943     0.888     0.772
  v18    0.248      0.910     0.828     0.720
  v19    0.222      0.860     0.740     0.643
  v20    0.208      0.813     0.661     0.575

------------------------------------------------------------
CORRELATIONS BETWEEN MVs AND LVs
            f1       f2     f3     f4     f5
f1
  v1    0.9395  -0.0026  0.665  0.322  0.518
  v2    0.9216   0.0039  0.587  0.279  0.452
f2
  v3    0.0191   0.8256  0.414  0.609  0.550
  v4   -0.0261   0.7608  0.353  0.539  0.465
  v5    0.0039   0.9025  0.542  0.795  0.710
f3
  v6    0.5821   0.4506  0.846  0.664  0.739
  v7    0.4865   0.3942  0.760  0.579  0.648
  v8    0.4216   0.3724  0.673  0.489  0.563
  v9    0.6471   0.4993  0.896  0.756  0.845
  v10   0.5736   0.4519  0.838  0.664  0.734
  v11   0.4886   0.3846  0.759  0.568  0.637
f4
  v12   0.2511   0.5540  0.565  0.781  0.639
  v13   0.3034   0.7905  0.768  0.923  0.869
  v14   0.2950   0.7269  0.722  0.905  0.812
  v15   0.2732   0.6579  0.647  0.856  0.726
f5
  v16   0.4091   0.5527  0.674  0.693  0.814
  v17   0.5305   0.6952  0.872  0.874  0.943
  v18   0.4794   0.6670  0.818  0.829  0.910
  v19   0.4365   0.5974  0.735  0.738  0.860
  v20   0.4053   0.5397  0.694  0.690  0.813

------------------------------------------------------------
INNER MODEL
$f3
    concept  value
1        R2 0.7419
2 Intercept 0.0000
3   path_f1 0.6743
4   path_f2 0.5355
```

Continued below.

R Console

File Edit Misc Packages Windows Help Vignettes

```
$f3
    concept  value
1        R2 0.7419
2 Intercept 0.0000
3   path_f1 0.6743
4   path_f2 0.5355

$f4
    concept  value
1        R2 0.8105
2 Intercept 0.0000
3   path_f2 0.5238
4   path_f3 0.5036

$f5
    concept  value
1        R2 0.8695
2 Intercept 0.0000
3   path_f3 0.4773
4   path_f4 0.5099

-----------------------------------------------------------
CORRELATIONS BETWEEN LVs
       f1      f2     f3     f4     f5
f1  1.0000  0.0005  0.675  0.324  0.523
f2  0.0005  1.0000  0.536  0.794  0.706
f3  0.6746  0.5359  1.000  0.784  0.877
f4  0.3243  0.7936  0.784  1.000  0.884
f5  0.5231  0.7060  0.877  0.884  1.000

-----------------------------------------------------------
SUMMARY INNER MODEL
    LV.Type  Measure  MVs  R.square  Av.Commu  Av.Redun    AVE
f1   Exogen    Rflct    2     0.000     0.866     0.000  0.866
f2   Exogen    Rflct    3     0.000     0.692     0.000  0.692
f3  Endogen    Rflct    6     0.742     0.638     0.473  0.638
f4  Endogen    Rflct    4     0.810     0.754     0.611  0.754
f5  Endogen    Rflct    5     0.870     0.756     0.657  0.756

-----------------------------------------------------------
GOODNESS-OF-FIT
         GoF   value
1   Absolute  0.7632
2   Relative  0.9754
3  Outer.mod  0.9996
4  Inner.mod  0.9758

-----------------------------------------------------------
TOTAL EFFECTS
    relationships  dir.effects  ind.effects  tot.effects
1          f1->f2        0.000        0.000        0.000
2          f1->f3        0.674        0.000        0.674
3          f1->f4        0.000        0.340        0.340
```

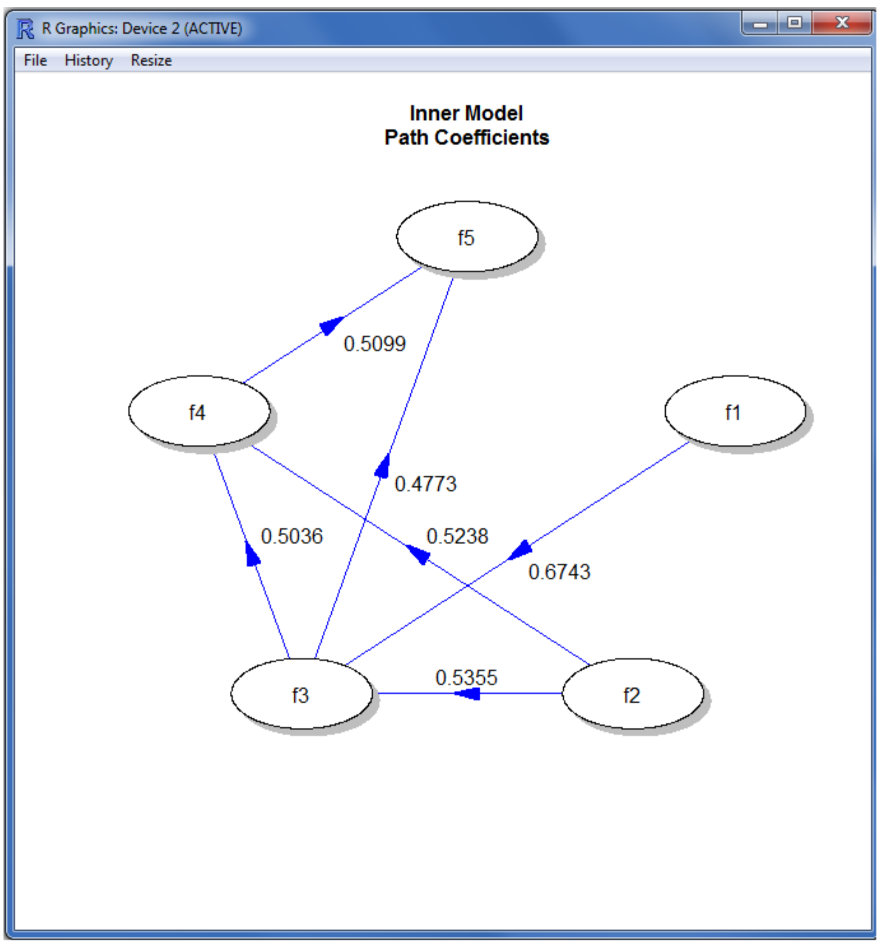
Continued below.

```
3   Outer.mod   0.9996
4   Inner.mod   0.9758

----------------------------------------------------------
TOTAL EFFECTS
    relationships  dir.effects  ind.effects  tot.effects
1          f1->f2        0.000        0.000        0.000
2          f1->f3        0.674        0.000        0.674
3          f1->f4        0.000        0.340        0.340
4          f1->f5        0.000        0.495        0.495
5          f2->f3        0.536        0.000        0.536
6          f2->f4        0.524        0.270        0.793
7          f2->f5        0.000        0.660        0.660
8          f3->f4        0.504        0.000        0.504
9          f3->f5        0.477        0.257        0.734
10         f4->f5        0.510        0.000        0.510
> 
```

Another big advantage to using the ‘plspm’ package (rather than others available for PLS modeling) is the ability to produce a path diagram based on the model fitted.

```
R Console
File  Edit  Misc  Packages  Windows  Help  Vignettes
> plot(pls.model.1)
> 
```



Another advantage to using the ‘plspm’ package is the ability to conduct bootstrapped validation of a PLS path model using the ‘boot.val’ optional argument to the ‘plspm’ function.

```
R Console
File  Edit  Misc  Packages  Windows  Help  Vignettes
> pls.model.2 <- plspm(x = pls.data[,4:23], inner = inner.matrix, outer = outer.list, mod$
+    scheme = "factor", scaled = TRUE, plsr = TRUE, boot.val = TRUE, br = 200, tol = 0.00$
> pls.model.2

PARTIAL LEAST SQUARES PATH MODELING (PLS-PM)
----------------------------------------------
Results available in the following objects:

     Name              Description
1    "$outer.mod"      "outer model"
2    "$inner.mod"      "inner model"
3    "$latents"        "scaled LVs"
4    "$scores"         "LVs for scaled=FALSE"
5    "$out.weights"    "outer weights"
6    "$loadings"       "loadings"
7    "$path.coefs"     "path coefficients matrix"
8    "$r.sqr"          "R-squared"
9    "$outer.cor"      "outer correlations"
10   "$inner.sum"      "summary inner model"
11   "$effects"        "total effects"
12   "$unidim"         "unidimensionality"
13   "$gof"            "goodnes-of-fit"
14   "$boot"           "bootstrap results"
15   "$data"           "data matrix"
----------------------------------------------
You can also use the function 'summary'

> |
```

Notice in the above table, there is a “$boot” element in the output. The rest of the output is identical to what was displayed above. The “$boot” element contains the cross validation output, which is the only part of the output displayed below.

R Console

File Edit Misc Packages Windows Help Vignettes

```
10         f4->f5        0.510        0.000        0.510

-------------------------------------------------------
BOOTSTRAP VALIDATION
weights
     Original  Mean.Boot  Std.Error  perc.05  perc.95
v1      0.570      0.571    0.00873    0.557    0.586
v2      0.504      0.503    0.00764    0.491    0.516
v3      0.374      0.375    0.00976    0.358    0.390
v4      0.328      0.327    0.01060    0.310    0.343
v5      0.489      0.490    0.01127    0.472    0.509
v6      0.222      0.222    0.00372    0.216    0.228
v7      0.192      0.192    0.00401    0.186    0.198
v8      0.168      0.168    0.00527    0.160    0.176
v9      0.251      0.251    0.00440    0.243    0.258
v10     0.221      0.221    0.00381    0.215    0.227
v11     0.189      0.190    0.00416    0.183    0.197
v12     0.238      0.238    0.00434    0.231    0.245
v13     0.328      0.328    0.00375    0.321    0.334
v14     0.306      0.305    0.00343    0.300    0.311
v15     0.274      0.275    0.00359    0.269    0.280
v16     0.206      0.206    0.00264    0.201    0.209
v17     0.263      0.263    0.00297    0.258    0.268
v18     0.248      0.248    0.00265    0.244    0.252
v19     0.222      0.221    0.00272    0.217    0.225
v20     0.208      0.208    0.00277    0.204    0.213

loadings
     Original  Mean.Boot  Std.Error  perc.05  perc.95
v1      0.939      0.939    0.00357    0.933    0.945
v2      0.922      0.921    0.00566    0.912    0.930
v3      0.826      0.826    0.01168    0.806    0.844
v4      0.761      0.759    0.01683    0.731    0.786
v5      0.902      0.903    0.00522    0.895    0.912
v6      0.846      0.846    0.00835    0.831    0.861
v7      0.760      0.760    0.01558    0.733    0.785
v8      0.673      0.673    0.02030    0.640    0.705
v9      0.896      0.896    0.00598    0.886    0.905
v10     0.838      0.838    0.01008    0.820    0.853
v11     0.759      0.758    0.01425    0.735    0.780
v12     0.781      0.782    0.01203    0.762    0.802
v13     0.923      0.923    0.00453    0.916    0.930
v14     0.905      0.905    0.00490    0.897    0.913
v15     0.856      0.857    0.00922    0.841    0.872
v16     0.815      0.814    0.01043    0.796    0.829
v17     0.943      0.943    0.00283    0.938    0.947
v18     0.910      0.909    0.00521    0.901    0.918
v19     0.860      0.859    0.00871    0.845    0.872
v20     0.813      0.813    0.01042    0.796    0.831

paths
        Original  Mean.Boot  Std.Error  perc.05  perc.95
f1->f3     0.674      0.673     0.0151    0.648    0.698
```

Continued below.

```
R Console
File  Edit  Misc  Packages  Windows  Help  Vignettes

loadings
     Original  Mean.Boot  Std.Error  perc.05  perc.95
v1      0.939      0.939    0.00357    0.933    0.945
v2      0.922      0.921    0.00566    0.912    0.930
v3      0.826      0.826    0.01168    0.806    0.844
v4      0.761      0.759    0.01683    0.731    0.786
v5      0.902      0.903    0.00522    0.895    0.912
v6      0.846      0.846    0.00835    0.831    0.861
v7      0.760      0.760    0.01558    0.733    0.785
v8      0.673      0.673    0.02030    0.640    0.705
v9      0.896      0.896    0.00598    0.886    0.905
v10     0.838      0.838    0.01008    0.820    0.853
v11     0.759      0.758    0.01425    0.735    0.780
v12     0.781      0.782    0.01203    0.762    0.802
v13     0.923      0.923    0.00453    0.916    0.930
v14     0.905      0.905    0.00490    0.897    0.913
v15     0.856      0.857    0.00922    0.841    0.872
v16     0.815      0.814    0.01043    0.796    0.829
v17     0.943      0.943    0.00283    0.938    0.947
v18     0.910      0.909    0.00521    0.901    0.918
v19     0.860      0.859    0.00871    0.845    0.872
v20     0.813      0.813    0.01042    0.796    0.831

paths
        Original  Mean.Boot  Std.Error  perc.05  perc.95
f1->f3     0.674      0.673     0.0151    0.648    0.698
f2->f3     0.536      0.536     0.0170    0.507    0.563
f2->f4     0.524      0.526     0.0152    0.501    0.551
f3->f4     0.504      0.502     0.0151    0.476    0.523
f3->f5     0.477      0.477     0.0178    0.448    0.503
f4->f5     0.510      0.510     0.0174    0.485    0.538

rsq
    Original  Mean.Boot  Std.Error  perc.05  perc.95
f3     0.742      0.741    0.01346    0.721    0.763
f4     0.811      0.811    0.00891    0.796    0.826
f5     0.869      0.869    0.00612    0.859    0.879

total.efs
        Original  Mean.Boot  Std.Error  perc.05  perc.95
f1->f2     0.000      0.000     0.0000    0.000    0.000
f1->f3     0.674      0.673     0.0151    0.648    0.698
f1->f4     0.340      0.338     0.0136    0.314    0.358
f1->f5     0.495      0.493     0.0147    0.469    0.515
f2->f3     0.536      0.536     0.0170    0.507    0.563
f2->f4     0.793      0.795     0.0112    0.776    0.814
f2->f5     0.660      0.661     0.0136    0.638    0.683
f3->f4     0.504      0.502     0.0151    0.476    0.523
f3->f5     0.734      0.733     0.0114    0.713    0.749
f4->f5     0.510      0.510     0.0174    0.485    0.538

> |
```

Interpretation was excluded from this article because the output of the functions covered is considered fairly intuitive. However, if one would like more information on interpreting PLS models, see Chin (2010).

Until next time, I'll drive my *Chevy to the leeve*..

References & Resources

Chin, W. W. (2010). How to write up and report PLS analyses. In Esposito, V., et al. (eds.), *Handbook of Partial Least Squares* (pp. 655 – 688). New York: Springer-Verlag.

Diamantopoulos, A., & Siguaw, J. A. (2006). Formative versus reflective indicators in organizational measure development: A comparison and empirical illustration. *British Journal of Management, 17*, 263 – 282.

Falk, R. F., & Miller, N. B. (1992). A primer for soft modeling. Akron, OH: University of Akron Press.

Garson, D. (2011). Partial Least Squares. Statnotes. Accessed May 9, 2011; from: http://faculty.chass.ncsu.edu/garson/PA765/pls.htm

Haenlein, M., & Kaplan, A. (2004). A beginner's guide to partial least squares analysis. Understanding Statistics, 3(4), 283 -- 297. Available at: http://www.stat.umn.edu/~sandy/courses/8801/articles/pls.pdf

Lohmoller, J. (1989). Latent variable path modeling with partial least squares. New York: Springer-Verlag.

Marcoulides, G. A., & Saunders, C. (2006). PLS: A silver bullet? *MIS Quarterly, 30*, iii – ix.

Sanchez, G. (2010). Package 'plspm'. Available at CRAN: http://cran.r-project.org/web/packages/plspm/index.html

Tenenhaus, M., Vinzi, V. E., Chatelin, Y., & Lauro, C. (2005). PLS path modeling. Computational Statistics & Data Analysis, 48, 159 -- 205. Available at: www.sciencedirect.com

Trinchera, L. (2007). Unobserved heterogeneity in structural equation models: A new approach to latent class detection in PLS path modeling. Doctoral dissertation. Available at: http://www.fedoa.unina.it/view/people/Trinchera,_Laura.html