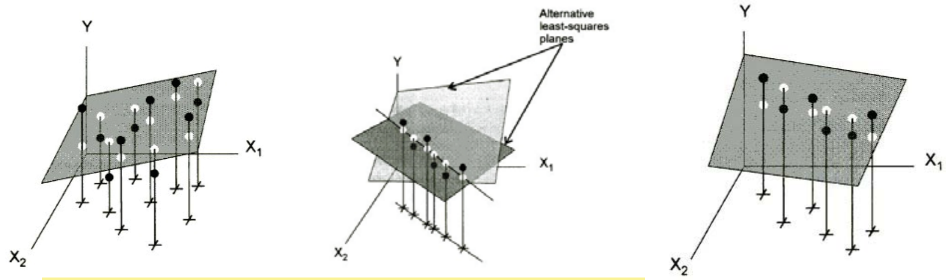
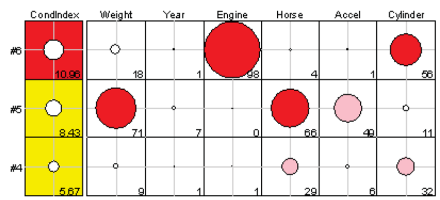



# Collinearity in regression

(the dreaded disease, and how to live with it)

Psychology 6140

| | |
|---|---|
| 15-30 | watch out |
| >30 | Trouble! |
| >100 | DISASTER |




# What is collinearity?

- If there is a *perfect* linear relation among the predictors:
  - $|\mathbf{X'X}|=0 \rightarrow (\mathbf{X'X})^{-1}$ does not exist
  - No *unique* solution for regression coeffs
  - Standard errors are infinite (why?)
- (Multi-) collinearity refers to the case when there are very high multiple correlations among Xs
  - i.e., $R^2$ ($x_i$ | other xs) > .90
  - Can't tell just by looking at simple correlations (why?)
  - (High simple $r_{ij}$ is sufficient, but not necessary)
  - $|\mathbf{X'X}| \approx 0 \rightarrow$ regression coeffs not well-determined



# What is collinearity?

- Consequences:
  - Estimated coefficients have large standard errors → small *t* statistics, large CIs
  - →Overall model may be highly significant, while no (or few) individual predictors are
  - May have poor numerical accuracy because $|\mathbf{X'X}| \approx 0$. (why?)
  - Partial regression coeffs (Δy/Δx, holding others **constant**) are estimating something that does not occur in the data. (why?)



# Collinearity: Practicalities

- Collinearity often occurs with time-series or region data, where different variables (wages, prices, GNP, mortality, …) tend to rise and fall together.
- Less common in cross-sectional social science studies, where variables are often weakly related.
- Perfect linear relations always arise when scores are *ipsatized* (individuals' % of total or dev. from mean)

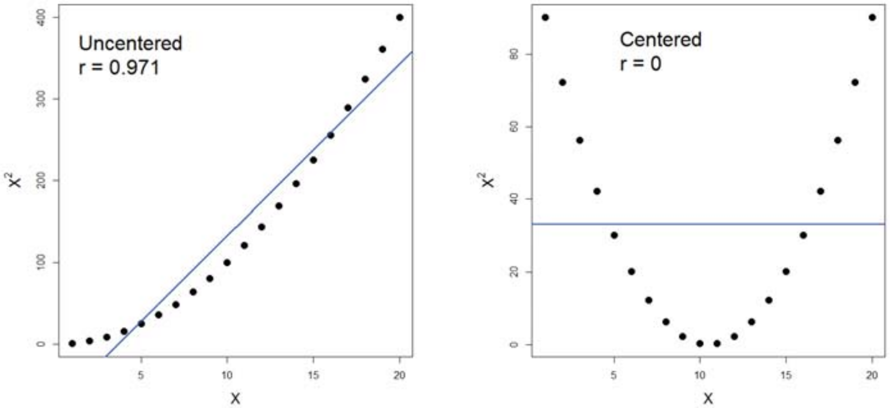
  %verbal + %math + %social + %perceptual = 100
- Also always in cases of wide data, $p > n$ [why? Think: R(**X**)]
- Common in models with interactions ($X_1$*$X_2$) or polynomial terms ($X^2$, $X^3$), *unless* these are centered using deviations from the mean
  - E.g., use

$$X_1X_2 = (x_1 - \bar{x}_1) \times (x_2 - \bar{x}_2) \qquad x, \ (x-\bar{x})^2, \ (x-\bar{x})^3, \ldots$$

# Why centering works

```
x <- 1:20
y <- x^2
```

```
x <- 1:20
y <- (x - mean(x))^2
```




Centering removes the necessary linear relation between X and $X^2$
NB: In R, **`poly(x, degree)`** in a model formula does the right thing.



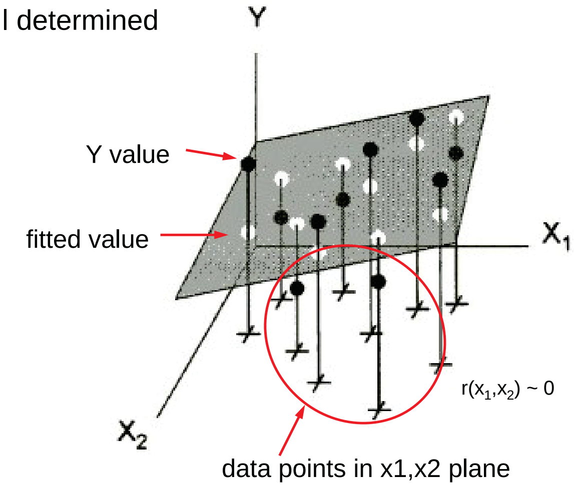
# Visualizing collinearity: Case 1

Small correlation between $x_1$ & $x_2$

Regression plane is well determined

We can see this in terms of how well the plane is supported.

A small change in one Y won't change things very much.




These figs originally from John Fox

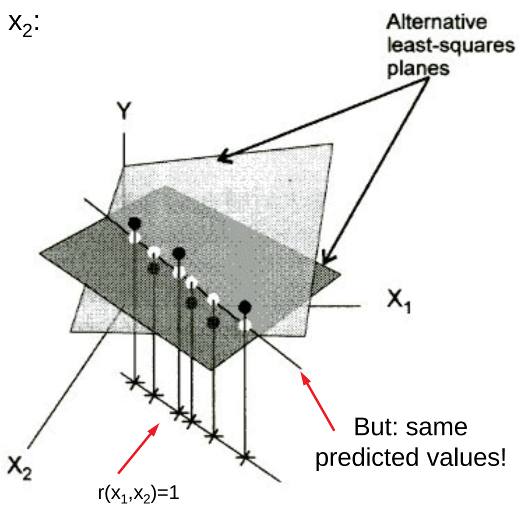


# Visualizing collinearity: Case 2

Perfect correlation between $x_1$ & $x_2$:
Regression plane is not unique

Note: if all we care about is in-sample **prediction**, no need to worry about collinearity.

We could use $x_1$, $x_2$, or both and get the same predicted values.






# Visualizing collinearity: Case 3

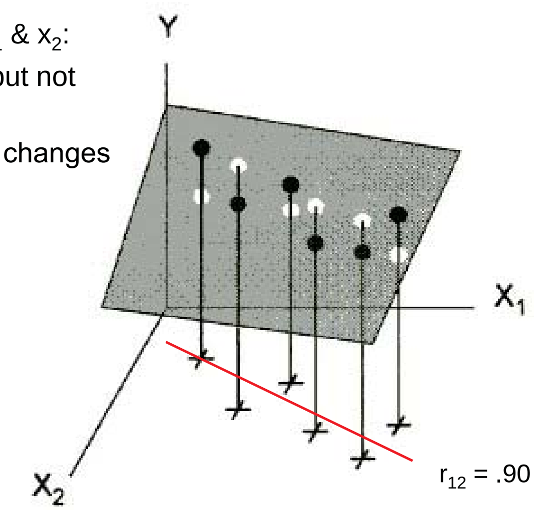
Strong correlation between $x_1$ & $x_2$:

Regression plane is unique, but not well determined

Small changes in Ys → large changes in coefficients

We can see this in that the plane is **not** well supported.

So, a small change in the data can make a large change in the coefficients



# Measuring collinearity

- Sampling variances: $s^2(\mathbf{b}) = \text{MSE } (\mathbf{X'X})^{-1}$
- For 2 predictors:

$$s^2(b_1) = \frac{MSE}{(n-1)s^2_{x1}} \times \frac{1}{1-r^2_{12}}$$

Variance inflation factor (VIF): $s^2(b) \rightarrow$ infinity as $r^2 \rightarrow 1$

- More generally:

$$s^2(b_i) = \underbrace{\frac{MSE}{(n-1)s^2_{x_i}}}_{\text{Std. err. when } R^2=0} \times \frac{1}{1-R^2_{i|others}}$$

$$\frac{1}{1-R_i^2 \mid others} = VIF_i$$

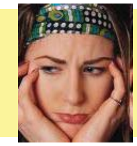
$$VIF_i = diag(R_{xx}^{-1})_i$$

$\sqrt{VIF_i}$ = multiplier of s.e.

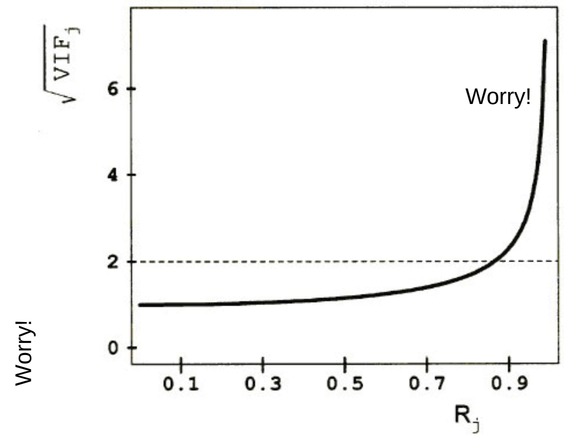
(more useful measure)



# When should I worry?



| $R_i$ | $R_i^2$ | VIF | $\sqrt{VIF}$ |
|---|---|---|---|
| 0 | 0 | 1.0 | 1.0 |
| .2 | .04 | 1.04 | 1.02 |
| .4 | .16 | 1.19 | 1.09 |
| .6 | .36 | 1.56 | 1.25 |
| .8 | .64 | 2.78 | 1.67 |
| .9 | .81 | 5.26 | 2.29 |
| .95 | .903 | 10.3 | 3.21 |
| .99 | .980 | 50.3 | 7.09 |
| 1.0 | 1.0 | $\infty$ | $\infty$ |






# Collinearity diagnostics

- VIF, or its inverse, TOLerance = $1-R^2_{i|others}$
- Condition #s, based on eigenvalues of $R_{xx}$:
  - #($\lambda_i \approx 0$) = # near linear dependencies
  - Scale relative to max λ to make scale free:

$$CN_i = \left(\frac{\lambda_{max}}{\lambda_i}\right)^{1/2}$$

| | |
|---|---|
| 15-30 | watch out |
| >30 | Trouble! |
| >100 | DISASTER |

  - Connection with eigenvalues of (X'X):

$$(\mathbf{X'X}) = \mathbf{V} \ \text{diag}(\lambda_i) \ \mathbf{V'} \ \rightarrow \ (\mathbf{X'X})^{-1} = \mathbf{V} \ \text{diag}(\frac{1}{\lambda_i}) \ \mathbf{V'}$$



# Collinearity diagnostics

- How to tell which variables are involved in each near-linear dependence?
  - Eigenvector proportions: % variance of each variable related to each small λ (large CN)
  - PROC REG: option COLLINOINT on MODEL statement
    - E.g., `proc reg; model y=x1-x5 / vif collinoint;`
  - Note: SAS (SPSS?) also has a less useful COLLIN that does *not* adjust for the intercept.
  - R: use `car::vif()` and `perturb::colldiag()`
    - `mymod <- lm(y ~ ., data=)`
    - `vif(mymod)`
    - `colldiag(mymod, center=TRUE)`

# Example: cars data

```
%include data(cars2);
proc reg data=cars2;
   model mpg = weight year engine horse accel cylinder;
run;
```

Standard output:

```
                     Analysis of Variance
                                Sum of           Mean
Source                  DF     Squares         Square    F Value    Pr > F

Model                    6       19054     3175.66762     269.59    <.0001
Error                  384     4523.41       11.77970
Corrected Total        390       23577

          Root MSE             3.43216    R-Square       0.8081

                       Parameter Estimates
                      Parameter       Standard
Variable      DF       Estimate          Error    t Value    Pr > |t|

Intercept      1      -14.63175        4.88451      -3.00      0.0029
Weight         1       -0.00678     0.00067704     -10.02      <.0001
Year           1        0.76205        0.05292      14.40      <.0001
Engine         1        0.00848        0.00747       1.13      0.2572
Horse          1       -0.00290        0.01411      -0.21      0.8375
Accel          1        0.06121        0.10366       0.59      0.5552
Cylinder       1       -0.34602        0.33313      -1.04      0.2996
```



```
*-- refit model, and request collinearity diagnostics;
proc reg data = cars2;
  model mpg = weight year engine horse accel cylinder / vif collinoint;
run;
```

VIF Output:

```
                           Parameter Estimates
                      Parameter     Standard                               Variance
Variable      DF       Estimate        Error    t Value    Pr > |t|       Inflation

Intercept      1      -14.63175      4.88451      -3.00      0.0029               0
Weight         1       -0.00678   0.00067704     -10.02      <.0001        10.85718
Year           1        0.76205      0.05292      14.40      <.0001         1.25307
Engine         1        0.00848      0.00747       1.13      0.2572        20.23415
Horse          1       -0.00290      0.01411      -0.21      0.8375         9.66219
Accel          1        0.06121      0.10366       0.59      0.5552         2.70928
Cylinder       1       -0.34602      0.33313      -1.04      0.2996        10.65789
```

- 4 of 6 predictors have dangerously high VIFs!
- How many near singularities?
- which predictors involved in each?



COLLINOINT Output:

*Collinearity Diagnostics (intercept adjusted)*

| | | | Proportion of Variation | | | | | |
|---|---|---|---|---|---|---|---|---|
| *Number* | *Eigenvalue* | *Condition Index* | *Weight* | *Year* | *Engine* | *Horse* | *Accel* | *Cylinder* |
| *1* | 4.25623 | 1.00000 | 0.0043 | 0.0097 | 0.0026 | 0.0052 | 0.0092 | 0.0046 |
| *2* | 0.83541 | 2.25716 | 0.0054 | 0.8562 | 0.0011 | 0.00004 | 0.0040 | 0.0030 |
| *3* | 0.68081 | 2.50034 | 0.0128 | 0.0536 | 0.0018 | 0.0024 | 0.4240 | 0.0052 |
| *4* | 0.13222 | 5.67358 | 0.0882 | 0.0058 | 0.0115 | 0.2917 | 0.0614 | 0.3172 |
| *5* | 0.05987 | 8.43157 | 0.7111 | 0.0688 | 0.00006 | 0.6602 | 0.4918 | 0.1110 |
| *6* | 0.03545 | 10.95701 | 0.1783 | 0.0059 | 0.983 | 0.0404 | 0.0096 | 0.5591 |

Look at large CN rows

(others don't matter)

See which predictors have large % variance in each

5: Weight & Horsepower
6: Engine size & cylinders



## Same output from R

```
library(car)        # for vif
library(perturb)    # for colldiag
cars.mod <- lm(mpg ~ weight + year + engine + horse + accel + cylinder,
               data=cars)
vif(cars.mod)
colldiag(cars.mod, center=TRUE)
```

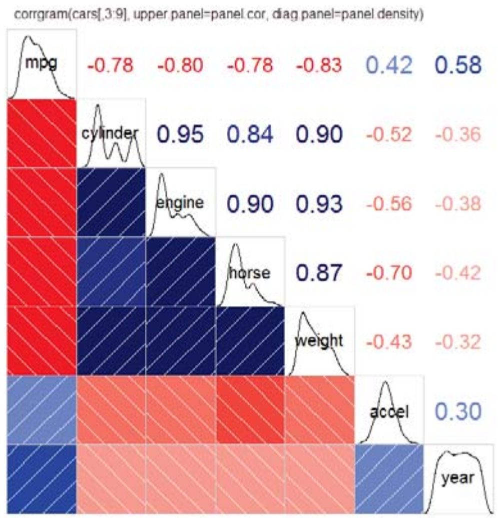
```
> vif(cars.mod)
 weight      year   engine     horse     accel cylinder
 10.732     1.245   19.642     9.398     2.626   10.633

> colldiag(cars.mod, center=TRUE)
Condition
Index   Variance Decomposition Proportions
          weight year  engine horse accel cylinder
1    1.000 0.004  0.010 0.003  0.005 0.009 0.005
2    2.252 0.007  0.787 0.002  0.000 0.022 0.004
3    2.515 0.010  0.142 0.001  0.002 0.423 0.004
4    5.660 0.087  0.005 0.014  0.306 0.063 0.309
5    8.342 0.715  0.052 0.000  0.654 0.469 0.115
6   10.818 0.176  0.004 0.981  0.032 0.013 0.563
```

# Visualizing correlations

High simple correlations $r_{ij}$ among predictors are sufficient for collinearity, but not necessary (because it depends on $R^2_{i|others}$)

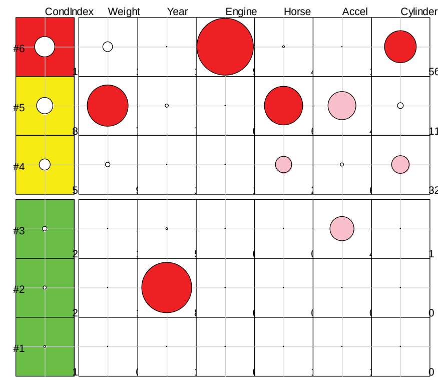
Nevertheless, high simple correlations signal a problem.

A corrgram reorders the variables to show patterns and can highlight large correlations






# Visualizing diagnostics: tableplots

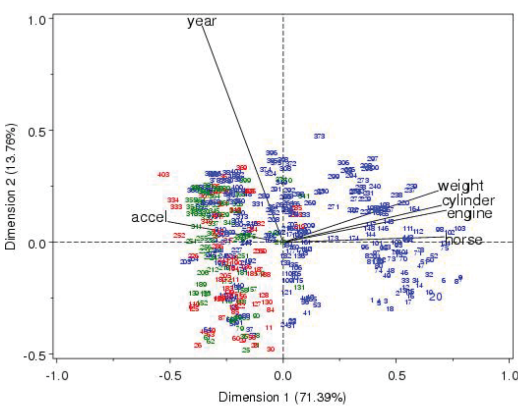



- Sort table in reverse order, by Condition Index
- Color code CondIndex by "danger"
- Variance proportions: ~ circle diameter
- Uses R tableplot package

See: Friendly & Kwan (2009), "Where's Waldo: Visualizing collinearity diagnostics", *The American Statistician*.



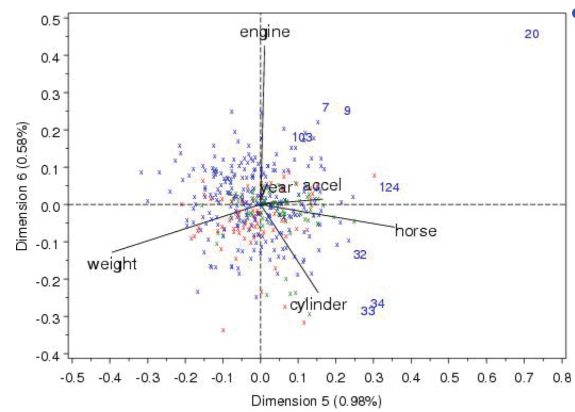
# Visualizing collinearity: biplots




- Standard biplot shows the data in the space of the largest dimensions
  - Largest eigenvalues
  - Smallest condition indices
  - Not useful for assessing collinearity



# Visualizing collinearity: biplots




- Collinearity biplot shows the data in the space of the *smallest* dimensions
  - Smallest eigenvalues
  - Largest condition indices
  - Shows collinearity directly
  - Also shows possible outliers

# Example: Acetylene production data

- Model: $y = x_1 + x_2 + x_3 + x_1x_2 + x_1^2$

```
data acetyl;
   input x1-x3 y @@;
   x1x2 = x1 * x2;
   x1x1 = x1 * x1;
  label x1   = 'Reactor temperature'
        x2   = 'H2 to n-heptone ratio'
        x3   = 'Contact time'
        y    = 'Conversion percentage'
        x1x2= 'Temp-ratio interaction'
        x1x1= 'Squared temperature';
   datalines;
1300  7.5 .012 49   1300  9   .012  50.2 1300 11 .0115 50.5
 ...
;
proc reg data=acetyl;
  model y=x1 x2 x3 x1x2 x1x1 /  VIF COLLINOINT;
  run;
```

Models with interactions and polynomial terms often result in high collinearity

Again, this is only a problem if we care about testing coefficients for individual terms



VIF Output:

Parameter Estimates

| Variable | DF | Parameter Estimate | Standard Error | t Value | Pr > \|t\| | Variance Inflation |
|---|---|---|---|---|---|---|
| Intercept | 1 | 390.53822 | 211.52287 | 1.85 | 0.0946 | 0 |
| x1 | 1 | -0.77676 | 0.32448 | -2.39 | 0.0377 | 7682.37019 |
| x2 | 1 | 10.17351 | 0.94301 | 10.79 | <.0001 | 320.02156 |
| x3 | 1 | -121.62608 | 69.01749 | -1.76 | 0.1085 | 53.52457 |
| x1x2 | 1 | -0.00805 | 0.00077209 | -10.43 | <.0001 | 344.54471 |
| x1x1 | 1 | 0.00039831 | 0.00012528 | 3.18 | 0.0098 | 6643.31989 |

COLLINOINT Output:

Collinearity Diagnostics (intercept adjusted)

| Number | Eigenvalue | Condition Index | Proportion of Variation x1 | x2 | x3 | x1x2 | x1x1 |
|---|---|---|---|---|---|---|---|
| 1 | 3.3204 | 1.000 | 0.0000103 | 0.0000867 | 0.0014 | 0.0001125 | 0.0000118 |
| 2 | 1.6176 | 1.433 | 0.0000061 | 0.0008279 | 0.0007648 | 0.0006342 | 0.0000071 |
| 3 | 0.0603 | 7.420 | 0.0002676 | 0.0001027 | 0.2085 | 0.0001889 | 0.0004914 |
| 4 | 0.0015 | 47.158 | 0.0003061 | 0.99890 | 0.0125 | 0.9990 | 0.0004257 |
| 5 | 0.0000696 | 218.335 | 0.9994 | 0.0000218 | 0.7767 | 0.00001123 | 0.9991 |

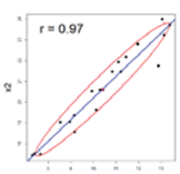
Two near linear dependencies, both fairly severe

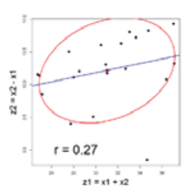


# Remedies for structural collinearity

- Collinearity often a data problem – no magic cure
- *Always* enter interactions using mean deviations
  - $x_1 * x_2 \rightarrow (x_1 - \bar{x}_1)(x_2 - \bar{x}_2)$
- Sometimes can redefine variables to reduce/remove high correlations
  - Divide by (adjust for): population, GNP, years in major leagues→ per capita measures, etc.
  - Sums & differences reduce correlations

$$\begin{matrix} x_1 \\ x_2 \end{matrix} \mapsto \begin{matrix} z_1 = x_1 + x_2 \\ z_2 = x_1 - x_2 \end{matrix}$$






# Example: Acetylene production data

```
*-- transform x1, x2 to deviations from mean;
proc standard data=acetyl out=acetyl1 m=0;
   var x1 x2;
*-- recompute powers and interactions using deviations;
data acetyl1;
    set acetyl1;
    x1x2 = x1 * x2;
    x1x1 = x1 * x1;
proc reg data=acetyl1;
    model y=x1 x2 x3 x1x2 x1x1 /  VIF COLLINOINT;
    run;
```

VIF Output:

Parameter Estimates

| Variable | DF | Parameter Estimate | Standard Error | t Value | Pr > \|t\| | Variance Inflation |
|---|---|---|---|---|---|---|
| Intercept | 1 | 39.35299 | 2.16281 | 18.20 | <.0001 | 0 |
| x1 | 1 | 0.08890 | 0.02431 | 3.66 | 0.0044 | 43.11271 |
| x2 | 1 | 0.40706 | 0.05459 | 7.46 | <.0001 | 1.07248 |
| x3 | 1 | -121.62608 | 69.01749 | -1.76 | 0.1085 | 53.52457 |
| x1x2 | 1 | -0.00805 | 0.00077209 | -10.43 | <.0001 | 1.09087 |
| x1x1 | 1 | 0.00039831 | 0.00012528 | 3.18 | 0.0098 | 4.68010 |

This removes the artificial collinearity in the interaction terms

COLLINOINT Output:

| Collinearity Diagnostics (intercept adjusted) | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | Proportion of Variation | | | | |
| Number | Eigenvalue | Condition Index | x1 | x2 | x3 | x1x2 | x1x1 |
| 1 | 2.3699 | 1.0000 | 0.00330 | 0.0232 | 0.0031 | 0.0174 | 0.015 |
| 2 | 1.0751 | 1.4847 | 0.00064 | 0.2678 | 0.000087 | 0.4871 | 0.017 |
| 3 | 0.8544 | 1.6654 | 0.0043 | 0.6113 | 0.0014 | 0.0949 | 0.032 |
| 4 | 0.6905 | 1.8526 | 0.0016 | 0.0974 | 0.0000015 | 0.3921 | 0.180 |
| 5 | 0.0100 | 15.3568 | 0.9902 | 0.000322 | 0.9954 | 0.0086 | 0.755 |

One near linear dependency remains

… involving x1 and x3



# Remedies

- Variable selection, model re-specification
  - Use of automatic, stepwise methods often misleading
    - Curing a collinearity-cold by risking pneumonia
  - Diagnostics + thought:
    - Redefine variables
    - Remove or average redundant ones
    - Force important predictors into model, use selection methods on remaining ones.



# Statistical remedies

- Transform $X_1 - X_p$ to principal components, $PC_1 - PC_p$
  - $PC_1 - PC_p$ are uncorrelated
  - Regress Y on $PC_1 - PC_p$
  - But: are the components interpretable?
  - Biplot of PCs with projected variable vectors can help!
- Incomplete principal components regression
  - Drop components associated with smallest eigenvalues (large condition #s)
  - Gives biased estimates, but with smaller std. errors
  - PROC REG: PCOMIT= option
  - In a way, this is similar to what we saw in biplots, looking at the smallest dimensions
- Good for prediction goal; less good for scientific explanation



# Example: fitness data

```
%include data(fitnessd);
proc reg data=fitness;
  model oxy = age weight runtime rstpulse runpulse maxpulse /
          / vif collinoint;
```

Parameter Estimates

| Variable | DF | Parameter Estimate | Standard Error | t Value | Pr > \|t\| | Variance Inflation |
|---|---|---|---|---|---|---|
| Intercept | 1 | 102.93448 | 12.40326 | 8.30 | <.0001 | 0 |
| age | 1 | -0.22697 | 0.09984 | -2.27 | 0.0322 | 1.51284 |
| weight | 1 | -0.07418 | 0.05459 | -1.36 | 0.1869 | 1.15533 |
| runtime | 1 | -2.62865 | 0.38456 | -6.84 | <.0001 | 1.59087 |
| rstpulse | 1 | -0.02153 | 0.06605 | -0.33 | 0.7473 | 1.41559 |
| runpulse | 1 | -0.36963 | 0.11985 | -3.08 | 0.0051 | 8.43727 |
| maxpulse | 1 | 0.30322 | 0.13650 | 2.22 | 0.0360 | 8.74385 |

We should have known that runpulse and maxpulse would be highly correlated

- Redefine these using sum and difference: both reasonably interpretable

```
*-- redefine pulse rate variables;
data fit2;
  set fitness;
  pulse = (runpulse + maxpulse);
  pdiff = (maxpulse - runpulse);

proc reg data=fit2;
  model oxy = age weight runtime rstpulse pulse pdiff/ vif;
run;
```

Parameter Estimates

| Variable | DF | Parameter Estimate | Standard Error | t Value | Pr > \|t\| | Variance Inflation |
|---|---|---|---|---|---|---|
| Intercept | 1 | 102.93448 | 12.40326 | 8.30 | <.0001 | 0 |
| age | 1 | -0.22697 | 0.09984 | -2.27 | 0.0322 | 1.51284 |
| weight | 1 | -0.07418 | 0.05459 | -1.36 | 0.1869 | 1.15533 |
| runtime | 1 | -2.62865 | 0.38456 | -6.84 | <.0001 | 1.59087 |
| rstpulse | 1 | -0.02153 | 0.06605 | -0.33 | 0.7473 | 1.41559 |
| pulse | 1 | -0.03321 | 0.02780 | -1.19 | 0.2439 | 1.57086 |
| pdiff | 1 | 0.33642 | 0.12540 | 2.68 | 0.0130 | 1.26394 |



Demonstration of PCA regression, and incomplete PCA regression

- Transform $X_1$-$X_p$ → $PC_1$-$PC_p$
- Use all or subset of $PC_1$-$PC_p$ as predictors

```
proc princomp data=fitness out=prin;
  var age weight runtime rstpulse runpulse maxpulse;
  run;

*-- Drop last component (biased, but no collinearity);
proc reg data=prin;
  model oxy = prin1-prin5 / vif;
  title2 'Incomplete PCA regression';
  run;
```

Parameter Estimates

| Variable | DF | Parameter Estimate | Standard Error | t Value | Pr > \|t\| | Variance Inflation |
|---|---|---|---|---|---|---|
| Intercept | 1 | 47.37581 | 0.45988 | 103.02 | <.0001 | 0 |
| Prin1 | 1 | -1.41517 | 0.29133 | -4.86 | <.0001 | 1.00000 |
| Prin2 | 1 | -3.32426 | 0.40570 | -8.19 | <.0001 | 1.00000 |
| Prin3 | 1 | -1.15396 | 0.48604 | -2.37 | 0.0256 | 1.00000 |
| Prin4 | 1 | -1.25553 | 0.54226 | -2.32 | 0.0291 | 1.00000 |
| Prin5 | 1 | 1.36099 | 0.76992 | 1.77 | 0.0893 | 1.00000 |



# Statistical remedies

- **Ridge regression**: purposely biased estimation
  - Trade a small amount of bias in *b* estimates for (hopefully) large reduction in sampling variances
  - **X'X** modified to (**X'X** + *k* **I**), where *k* is a 'ridge tuning constant'. As *k* increases:
    - ||**b**|| gets smaller (shrunk towards 0), bias increases
    - But: sampling variance of **b** decreases

$$Var(\mathbf{b}_k) = \sigma^2 \mathbf{G}_k (\mathbf{X}^T\mathbf{X})^{-1} \mathbf{G}_k^T \quad \text{where} \quad \mathbf{G}_k = [\mathbf{I} + k(\mathbf{X}^T\mathbf{X})^{-1}]^{-1}$$

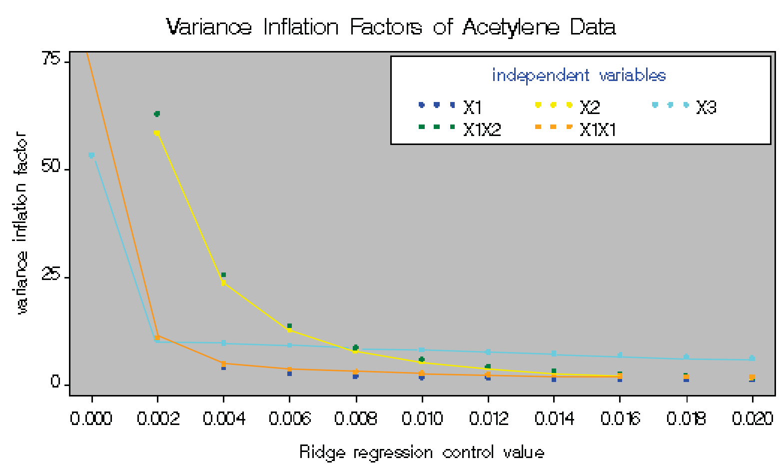
    - Goal: find a value of *k* making the trade-off most favorable
  - Probably best reserved for situations where other options don't work



# Example: Acetylene production data

Plot of VIF values vs. *k* for raw variables, just to illustrate how ridge regression decreases the effects of collinearity.

Even very small values of *k* are effective here.

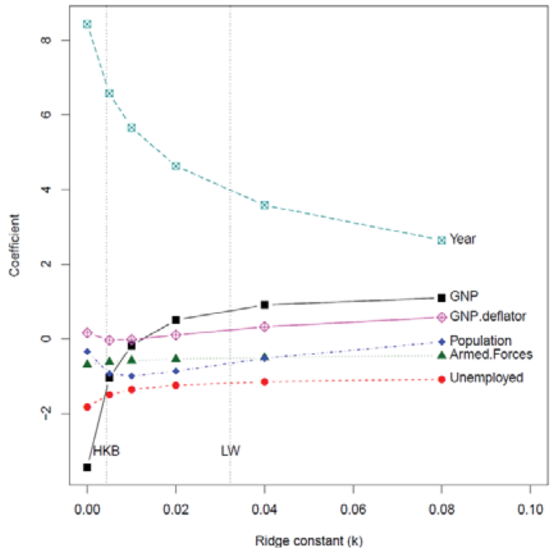



Note: the VIF at k=0 is 7682 for X1, 6643 for X1X1, 345 for X1X2, and 320 for X2

# Generalized ridge trace plots

The standard ridge trace plot shows bias, but not how shrinkage affects precision

In practice, people often rely on numerical criteria such as those due to Hoerl et al (HKB) and Lawless & Wang (LW) to choose the ridge constant, *k*.



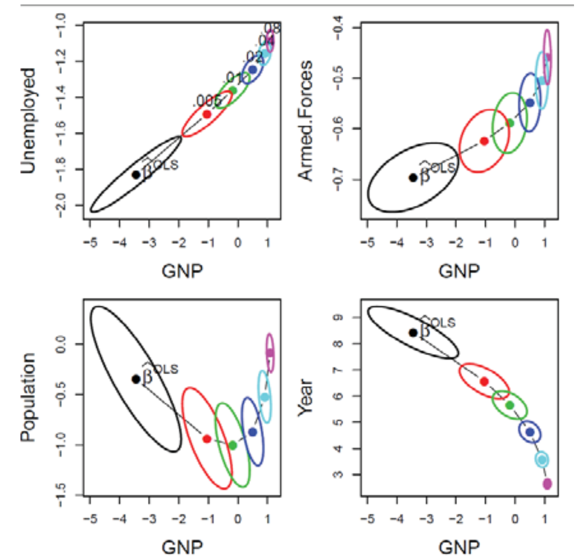



# Generalized ridge trace plots

The generalized ridge trace plot shows the covariance ellipse for pairs of coefficients

Can see directly how the changes in coefficients are related to decreases in variance

Graphs: R `genridge` package

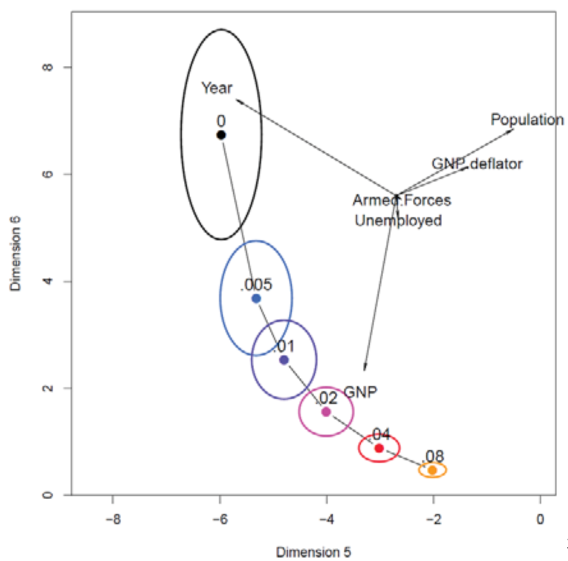





# Generalized ridge biplots

A biplot version shows the regression coefficients transformed to the space of the smallest principal components of **X'X**

Variable vectors show how these dimensions relate to the original variables






# Summary

- Collinearity is a *data* problem
  - Some predictors nearly linearly dependent
  - Consequences: large std. errors → large CIs (NS)
  - Not a problem if we are only interested in pure prediction
- Measuring & understanding collinearity:
  - VIF: $1/(1-R^2\ x_i|\text{others})$ – involvement of each variable
  - Variance proportions: how variables are involved
- Visualizing collinearity:
  - Tableplots: what information to pay attention to
  - Biplots: sources of collinearity among the small dimensions
- Remedies:
  - Re-express or re-define variables often helps
  - So too does thoughtful model selection
  - Statistical remedies (PCA regression, ridge-regression) cure the problem, but often at a cost of more difficult interpretation.