Lecture 17 Variale Selection Examples

Election data

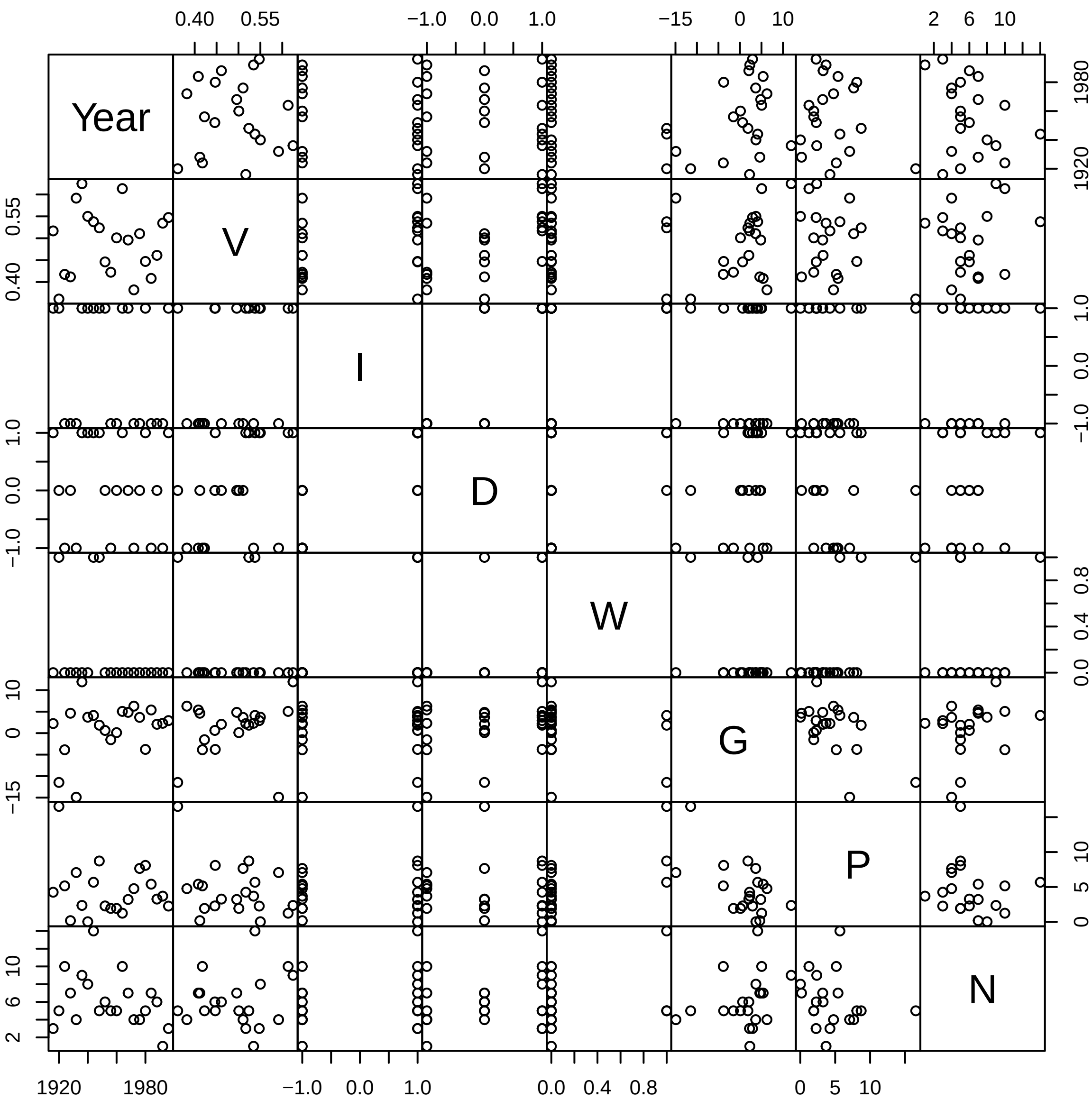
Data: http://www.stat.uchicago.edu/~yibi/s224/data/P160.txt

Here is a dataset from the book (p. 160) that we will use to demonstrate the FS, BE, and SW model selection algorithms.

- V: Proportion of votes to the Democrat candidate out of the total votes to the Dem + Rep candidates (i.e., votes to the 3rd or other candidates are not included)
- I: 1 if the incumbent is a Democrat at the time of the election, -1 if the incumbent is a Republican
- D: Democrat incumbent?
  - $D = 1$ if the Democrat candidate is incumbent
  - $D = -1$ if the Republican candidate is incumbent
  - $D = 0$ if neither candidate is incumbent
- W: war time election? ($1 =$ Yes, $0 =$ No)
- G: GDP growth rate in election year
- P: (absolute) GDP deflator growth rate
- N: number of quarters in which GDP growth rate $> 3.2\%$ in the previous 4 years

```
p160 = read.table("P160.txt", h=T)
```

```r
pairs(p160, gap=0,oma=c(2,2,2,2))
```



Backward Elimination

The R command step() can perform all BE, FS, and SW algorithm.

- By default, step() performs BE when applied to a model.
- test="F" means we want it to perform F-tests to compare the current model with a potential model. Without specifying test="F", only the AIC but not P-values are shown in the output
- By default, step() chooses models based on AIC, not P-values even if we specify test="F"

```
step(lm(V ~ I + D + W + G + P + N, data=p160), test="F")
## Start:  AIC=-104.98
## V ~ I + D + W + G + P + N
##
##         Df Sum of Sq      RSS     AIC F value Pr(>F)
## - N      1 0.0000079 0.072712 -106.98  0.0015 0.9694
## - I      1 0.0000400 0.072744 -106.97  0.0077 0.9313
## - W      1 0.0000894 0.072793 -106.96  0.0172 0.8975
## - G      1 0.0016214 0.074325 -106.52  0.3122 0.5851
## - P      1 0.0044157 0.077119 -105.75  0.8503 0.3721
## <none>               0.072704 -104.98
## - D      1 0.0101039 0.082808 -104.25  1.9456 0.1848
##
## Step:  AIC=-106.98
## V ~ I + D + W + G + P
##
##         Df Sum of Sq      RSS     AIC F value Pr(>F)
## - I      1 0.0000436 0.072755 -108.97  0.0090 0.9257
## - W      1 0.0001396 0.072851 -108.94  0.0288 0.8675
## - G      1 0.0016497 0.074361 -108.51  0.3403 0.5683
## - P      1 0.0048827 0.077594 -107.62  1.0073 0.3315
## <none>               0.072712 -106.98
## - D      1 0.0101469 0.082859 -106.24  2.0933 0.1685
##
## Step:  AIC=-108.97
```

```
## V ~ D + W + G + P
##
##         Df Sum of Sq      RSS     AIC F value  Pr(>F)
## - W     1 0.0001571 0.072912 -110.92  0.0346 0.85488
## - G     1 0.0016185 0.074374 -110.51  0.3559 0.55912
## - P     1 0.0050355 0.077791 -109.56  1.1074 0.30829
## <none>              0.072755 -108.97
## - D     1 0.0245242 0.097280 -104.87  5.3932 0.03373
##
## Step:  AIC=-110.92
## V ~ D + G + P
##
##         Df Sum of Sq      RSS     AIC F value  Pr(>F)
## - G     1 0.0017808 0.074693 -112.42  0.4152 0.52794
## <none>              0.072912 -110.92
## - P     1 0.0110706 0.083983 -109.95  2.5812 0.12655
## - D     1 0.0270882 0.100001 -106.29  6.3158 0.02234
##
## Step:  AIC=-112.42
## V ~ D + P
##
##         Df Sum of Sq      RSS     AIC F value  Pr(>F)
## <none>              0.074693 -112.42
## - P     1 0.0099223 0.084616 -111.80  2.3911 0.13943
## - D     1 0.0255565 0.100250 -108.24  6.1588 0.02317
##
## Call:
## lm(formula = V ~ D + P, data = p160)
##
## Coefficients:
## (Intercept)            D            P
##    0.514022     0.043134    -0.006017
```

Forward Selection

To perform FS using `step()`

- need to set `direction="forward"`
- need to specify the scope, i.e., the pool of candidate terms.
- We can include more terms in the initial model, not just the intercept.

```
step(lm(V~1, data=p160),
     scope = V ~ I + D + W + G + P + N,
     direction="forward", test="F")
## Start:  AIC=-107.78
## V ~ 1
##
##         Df Sum of Sq      RSS     AIC F value  Pr(>F)
## + D      1 0.0280805 0.084616 -111.80  6.3054 0.02124
## + I      1 0.0135288 0.099167 -108.47  2.5921 0.12389
## + P      1 0.0124463 0.100250 -108.24  2.3589 0.14106
## <none>               0.112696 -107.78
## + G      1 0.0060738 0.106622 -106.94  1.0824 0.31123
## + N      1 0.0024246 0.110271 -106.24  0.4178 0.52579
## + W      1 0.0009518 0.111744 -105.96  0.1618 0.69197
##
## Step:  AIC=-111.8
## V ~ D
##
##         Df Sum of Sq      RSS     AIC F value Pr(>F)
## + P      1 0.0099223 0.074693 -112.42  2.3911 0.1394
## <none>               0.084616 -111.80
## + W      1 0.0068141 0.077801 -111.56  1.5765 0.2253
## + I      1 0.0012874 0.083328 -110.12  0.2781 0.6044
## + G      1 0.0006325 0.083983 -109.95  0.1356 0.7170
## + N      1 0.0000033 0.084612 -109.80  0.0007 0.9793
##
## Step:  AIC=-112.42
## V ~ D + P
```

```
##
##        Df  Sum of Sq      RSS     AIC F value Pr(>F)
## <none>                0.074693 -112.42
## + G     1 0.00178078 0.072912 -110.92  0.4152 0.5279
## + W     1 0.00031940 0.074374 -110.51  0.0730 0.7903
## + N     1 0.00018496 0.074508 -110.47  0.0422 0.8397
## + I     1 0.00002633 0.074667 -110.42  0.0060 0.9392
##
## Call:
## lm(formula = V ~ D + P, data = p160)
##
## Coefficients:
## (Intercept)            D            P
##    0.514022     0.043134    -0.006017
```

We can include more terms in the initial model, not just the intercept.

```
step(lm(V~D+P, data=p160),
     scope = V ~ I + D + W + G + P + N,
     direction="forward", test="F")
## Start:  AIC=-112.42
## V ~ D + P
##
##        Df  Sum of Sq      RSS     AIC F value Pr(>F)
## <none>                0.074693 -112.42
## + G     1 0.00178078 0.072912 -110.92  0.4152 0.5279
## + W     1 0.00031940 0.074374 -110.51  0.0730 0.7903
## + N     1 0.00018496 0.074508 -110.47  0.0422 0.8397
## + I     1 0.00002633 0.074667 -110.42  0.0060 0.9392
##
## Call:
## lm(formula = V ~ D + P, data = p160)
##
## Coefficients:
## (Intercept)            D            P
##    0.514022     0.043134    -0.006017
```

Stepwise Selection

Set `direction="both"` within `step()` to perform the SW algorithm

```
step(lm(V ~ D+W, data=p160),
     scope = V ~ I + D + W + G + P + N,
     direction="both", test="F")
## Start:  AIC=-111.56
## V ~ D + W
##
##        Df Sum of Sq      RSS     AIC F value  Pr(>F)
## - W     1  0.006814 0.084616 -111.80  1.5765 0.22532
## <none>              0.077801 -111.56
## + P     1  0.003428 0.074374 -110.51  0.7835 0.38843
## + N     1  0.000374 0.077428 -109.66  0.0820 0.77802
## + I     1  0.000178 0.077623 -109.61  0.0391 0.84567
## + G     1  0.000011 0.077791 -109.56  0.0023 0.96213
## - D     1  0.033943 0.111744 -105.96  7.8529 0.01178
##
## Step:  AIC=-111.8
## V ~ D
##
##        Df Sum of Sq      RSS     AIC F value  Pr(>F)
## + P     1 0.0099223 0.074693 -112.42  2.3911 0.13943
## <none>              0.084616 -111.80
## + W     1 0.0068141 0.077801 -111.56  1.5765 0.22532
## + I     1 0.0012874 0.083328 -110.12  0.2781 0.60439
## + G     1 0.0006325 0.083983 -109.95  0.1356 0.71703
## + N     1 0.0000033 0.084612 -109.80  0.0007 0.97928
## - D     1 0.0280805 0.112696 -107.78  6.3054 0.02124
##
## Step:  AIC=-112.42
## V ~ D + P
##
##        Df Sum of Sq      RSS     AIC F value  Pr(>F)
## <none>              0.074693 -112.42
```

```
## - P      1 0.0099223 0.084616 -111.80  2.3911 0.13943
## + G      1 0.0017808 0.072912 -110.92  0.4152 0.52794
## + W      1 0.0003194 0.074374 -110.51  0.0730 0.79026
## + N      1 0.0001850 0.074508 -110.47  0.0422 0.83968
## + I      1 0.0000263 0.074667 -110.42  0.0060 0.93919
## - D      1 0.0255565 0.100250 -108.24  6.1588 0.02317
##
## Call:
## lm(formula = V ~ D + P, data = p160)
##
## Coefficients:
## (Intercept)            D            P
##    0.514022     0.043134    -0.006017
```

Backward Elimination w/ Interactions

The `step()` function automatically chooses model following the hierarchy principle below.

- an interaction is never added unless all the lower order effects in the interaction are already included.
- if an interaction is in the current model, none of its component variables or lower order interaction should be removed

```
step(lm(V ~ (I + D + P)^2 + G + W + N, data=p160), test="F")
## Start:  AIC=-112.97
## V ~ (I + D + P)^2 + G + W + N
##
##         Df Sum of Sq      RSS     AIC F value  Pr(>F)
## - D:P    1 0.0016907 0.039036 -114.04  0.4980 0.49505
## - I:D    1 0.0017542 0.039099 -114.01  0.5167 0.48725
## - N      1 0.0027613 0.040106 -113.48  0.8133 0.38646
## - G      1 0.0027717 0.040117 -113.47  0.8164 0.38559
## <none>               0.037345 -112.97
## - W      1 0.0056207 0.042966 -112.03  1.6556 0.22462
## - I:P    1 0.0159785 0.053323 -107.49  4.7065 0.05283
##
## Step:  AIC=-114.04
## V ~ I + D + P + G + W + N + I:D + I:P
##
##         Df Sum of Sq      RSS     AIC F value  Pr(>F)
## - I:D    1 0.0013687 0.040404 -115.32  0.4207 0.52878
## - N      1 0.0019804 0.041016 -115.00  0.6088 0.45035
## <none>               0.039036 -114.04
## - G      1 0.0039643 0.043000 -114.01  1.2187 0.29126
## - W      1 0.0047961 0.043832 -113.61  1.4744 0.24801
## - I:P    1 0.0244602 0.063496 -105.83  7.5193 0.01786
##
## Step:  AIC=-115.32
## V ~ I + D + P + G + W + N + I:P
```

```
##
##         Df Sum of Sq      RSS     AIC F value   Pr(>F)
## - N      1  0.002164 0.042568 -116.22  0.6962 0.419125
## <none>               0.040404 -115.32
## - G      1  0.004191 0.044595 -115.25  1.3484 0.266442
## - W      1  0.005976 0.046381 -114.42  1.9229 0.188857
## - D      1  0.012284 0.052688 -111.75  3.9524 0.068284
## - I:P    1  0.032299 0.072704 -104.98 10.3923 0.006657
##
## Step:  AIC=-116.22
## V ~ I + D + P + G + W + I:P
##
##         Df Sum of Sq      RSS     AIC F value   Pr(>F)
## - W      1 0.0039224 0.046491 -116.37  1.2900 0.275113
## <none>               0.042568 -116.22
## - G      1 0.0044563 0.047024 -116.13  1.4656 0.246085
## - D      1 0.0125772 0.055145 -112.79  4.1364 0.061374
## - I:P    1 0.0301435 0.072712 -106.98  9.9137 0.007111
##
## Step:  AIC=-116.37
## V ~ I + D + P + G + I:P
##
##         Df Sum of Sq      RSS     AIC F value  Pr(>F)
## - G      1 0.0031447 0.049635 -117.00  1.0146 0.32977
## <none>               0.046491 -116.37
## - D      1 0.0143950 0.060886 -112.71  4.6445 0.04781
## - I:P    1 0.0263606 0.072851 -108.94  8.5051 0.01064
##
## Step:  AIC=-117
## V ~ I + D + P + I:P
##
##         Df Sum of Sq      RSS     AIC F value  Pr(>F)
## <none>               0.049635 -117.00
## - D      1  0.011935 0.061570 -114.47  3.8472 0.06747
## - I:P    1  0.025032 0.074667 -110.42  8.0690 0.01181
```

```
##
## Call:
## lm(formula = V ~ I + D + P + I:P, data = p160)
##
## Coefficients:
## (Intercept)            I            D            P          I:P
##     0.48126      0.04517      0.05379      0.00224     -0.01243
```

BE, FS, SW algorithm Using BIC

To perform BE, FS, SW algorithms using BIC rather than AIC, where

$$
\begin{aligned}
\mathrm{AIC} &= n \log_e(\mathrm{SSE}_p/n) + 2p, \\
\mathrm{BIC} &= n \log_e(\mathrm{SSE}_p/n) + p \log_e(n),
\end{aligned}
$$

we need to specify `k=log(n)` in `step()`, where n = number of observations in the data. By default, the `step()` command assumes `k=2`, which corresponds to AIC.

There are $n = 21$ observations in the presidential data, obtained as follows.

```
dim(p160)
## [1] 21  8
```

We hence set `k=log(21)`.

```
step(lm(V ~ (I + D + P)^2 + G + W + N, data=p160), test="F", k=log(21))
## Start:  AIC=-102.53
## V ~ (I + D + P)^2 + G + W + N
##
##        Df Sum of Sq      RSS      AIC F value  Pr(>F)
## - D:P   1 0.0016907 0.039036 -104.643  0.4980 0.49505
## - I:D   1 0.0017542 0.039099 -104.609  0.5167 0.48725
## - N     1 0.0027613 0.040106 -104.075  0.8133 0.38646
## - G     1 0.0027717 0.040117 -104.070  0.8164 0.38559
## - W     1 0.0056207 0.042966 -102.629  1.6556 0.22462
## <none>              0.037345 -102.528
## - I:P   1 0.0159785 0.053323  -98.093  4.7065 0.05283
##
## Step:  AIC=-104.64
## V ~ I + D + P + G + W + N + I:D + I:P
##
##        Df Sum of Sq      RSS      AIC F value  Pr(>F)
## - I:D   1 0.0013687 0.040404 -106.964  0.4207 0.52878
## - N     1 0.0019804 0.041016 -106.648  0.6088 0.45035
```

```
## - G     1 0.0039643 0.043000 -105.656  1.2187 0.29126
## - W     1 0.0047961 0.043832 -105.254  1.4744 0.24801
## <none>              0.039036 -104.643
## - I:P   1 0.0244602 0.063496  -97.471  7.5193 0.01786
##
## Step:  AIC=-106.96
## V ~ I + D + P + G + W + N + I:P
##
##        Df Sum of Sq      RSS      AIC F value   Pr(>F)
## - N     1  0.002164 0.042568 -108.913  0.6962 0.419125
## - G     1  0.004191 0.044595 -107.936  1.3484 0.266442
## - W     1  0.005976 0.046381 -107.112  1.9229 0.188857
## <none>              0.040404 -106.964
## - D     1  0.012284 0.052688 -104.434  3.9524 0.068284
## - I:P   1  0.032299 0.072704  -97.672 10.3923 0.006657
##
## Step:  AIC=-108.91
## V ~ I + D + P + G + W + I:P
##
##        Df Sum of Sq      RSS     AIC F value   Pr(>F)
## - W     1 0.0039224 0.046491 -110.11  1.2900 0.275113
## - G     1 0.0044563 0.047024 -109.87  1.4656 0.246085
## <none>              0.042568 -108.91
## - D     1 0.0125772 0.055145 -106.52  4.1364 0.061374
## - I:P   1 0.0301435 0.072712 -100.71  9.9137 0.007111
##
## Step:  AIC=-110.11
## V ~ I + D + P + G + I:P
##
##        Df Sum of Sq      RSS     AIC F value  Pr(>F)
## - G     1 0.0031447 0.049635 -111.78  1.0146 0.32977
## <none>              0.046491 -110.11
## - D     1 0.0143950 0.060886 -107.49  4.6445 0.04781
## - I:P   1 0.0263606 0.072851 -103.72  8.5051 0.01064
##
```

```
## Step:  AIC=-111.78
## V ~ I + D + P + I:P
##
##         Df Sum of Sq      RSS     AIC F value  Pr(>F)
## <none>               0.049635 -111.78
## - D      1  0.011935 0.061570 -110.30  3.8472 0.06747
## - I:P    1  0.025032 0.074667 -106.25  8.0690 0.01181
##
## Call:
## lm(formula = V ~ I + D + P + I:P, data = p160)
##
## Coefficients:
## (Intercept)            I            D            P          I:P
##     0.48126      0.04517      0.05379      0.00224     -0.01243
```

FS, BE, SW Algorithms May Not Choose The Same Model

Recall Hamilton's Data on P.103 of the textbook

http://www.stat.uchicago.edu/~yibi/s224/data/P103.txt

```
hamilton = read.table("P103.txt", h = T)
```

Backward Elimination

The BE algorithm retains both `X1` and `X2` in the model.

```
step(lm(Y~X1+X2, data=hamilton), test="F")
## Start:  AIC=-133.43
## Y ~ X1 + X2
##
##        Df Sum of Sq    RSS      AIC F value    Pr(>F)
## <none>              0.0014 -133.429
## - X1    1    7.3099 7.3112   -6.780   63662 < 2.2e-16
## - X2    1    9.0072 9.0085   -3.648   78444 < 2.2e-16
##
## Call:
## lm(formula = Y ~ X1 + X2, data = hamilton)
##
## Coefficients:
## (Intercept)           X1           X2
##      -4.515        3.097        1.032
```

Forward Selection

The FS algorithm using AIC below also choose the Model `Y ~ X1 + X2`. However, if we choose using P-values, neither `X1` nor `X2` would be selected since their P-values are both over 0.05 in the first iteration.

```r
step(lm(Y~1, data=hamilton),
     scope = Y ~ X1 + X2,
     direction="forward", test="F")
## Start:  AIC=-5.65
## Y ~ 1
##
##        Df Sum of Sq    RSS     AIC F value Pr(>F)
## + X2    1   1.69736 7.3112 -6.7796  3.0181  0.106
## <none>              9.0086 -5.6481
## + X1    1   0.00006 9.0085 -3.6482  0.0001  0.993
##
## Step:  AIC=-6.78
## Y ~ X2
##
##        Df Sum of Sq    RSS     AIC F value    Pr(>F)
## + X1    1    7.3099 0.0014 -133.43   63662 < 2.2e-16
## <none>              7.3112   -6.78
##
## Step:  AIC=-133.43
## Y ~ X2 + X1
##
## Call:
## lm(formula = Y ~ X2 + X1, data = hamilton)
##
## Coefficients:
## (Intercept)           X2           X1
##      -4.515        1.032        3.097
```

Checking Interactions in the Presidential Election Data

As there might be interactions between the 6 predictors, we would want to include all $\binom{6}{2} = 15$ two-way interactions of the 6 predictors in the pool of variables being selected. Hence, we could do Backward Elimination starting from the largest model below.

```
lm(V ~ (I + D + W + G + P + N)^2, data=p160)
```

However, we cannot fit the model above since it involves

- 1 intercept
- 6 predictors
- $\binom{6}{2} = 15$ two-way interactions

However, we have only 21 observations, not enough to estimate the $1+6+15 = 22$ parameters.

Nonetheless, we could do Forward Selection or Stepwise Selection and using the model `V ~ (I + D + W + G + P + N)^2` as the `scope`. As we need to keep the lower-order components in the model if interaction terms might be present, we hence start from the model `V~I + D + W + G + P + N`.

In the following, I set `trace=0` so that only the last model selected is displayed, but no the intermediate steps.

Forward Selection

```
step(lm(V~I + D + W + G + P + N, data=p160),
     scope = V ~ (I + D + W + G + P + N)^2,
     direction="forward", test="F", trace=0)
##
## Call:
## lm(formula = V ~ I + D + W + G + P + N + I:G + I:N + D:P, data = p160
##
## Coefficients:
```

```
## (Intercept)            I            D            W            G
##   0.5047957   -0.0730853    0.0858492    0.0004869    0.0016718    0.
##           N          I:G          I:N          D:P
##  -0.0083533    0.0089385    0.0090142   -0.0070370
```

The model selected by Forward Selection is

```
summary(lm(V ~ W + I*G + I*N + D*P, data=p160))
##
## Call:
## lm(formula = V ~ W + I * G + I * N + D * P, data = p160)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.043491 -0.011401 -0.003583  0.005666  0.040428
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.5047957  0.0213603  23.632 8.86e-11
## W            0.0004869  0.0303705   0.016  0.98750
## I           -0.0730853  0.0172299  -4.242  0.00138
## G            0.0016718  0.0014732   1.135  0.28059
## N           -0.0083533  0.0025291  -3.303  0.00704
## D            0.0858492  0.0203483   4.219  0.00144
## P            0.0031706  0.0029900   1.060  0.31169
## I:G          0.0089385  0.0012757   7.007 2.25e-05
## I:N          0.0090142  0.0024327   3.706  0.00347
## D:P         -0.0070370  0.0034195  -2.058  0.06410
##
## Residual standard error: 0.02592 on 11 degrees of freedom
## Multiple R-squared:  0.9344, Adjusted R-squared:  0.8808
## F-statistic: 17.41 on 9 and 11 DF,  p-value: 2.758e-05
```

and the AIC of this model is

```
extractAIC(lm(V ~ W + I*G + I*N + D*P, data=p160))
## [1]   10.0000 -146.9932
```

Stepwise Selection

```
step(lm(V~I + D + W + G + P + N, data=p160),
     scope = V ~ (I + D + W + G + P + N)^2,
     direction="both", test="F", trace=0)
##
## Call:
## lm(formula = V ~ I + D + G + P + N + I:G + I:N + D:P, data = p160)
##
## Coefficients:
## (Intercept)            I            D            G            P
##    0.504629    -0.073140     0.085791     0.001670     0.003197    -0
##         I:G          I:N          D:P
##    0.008936     0.009027    -0.007016
```

The model selected by Stepwise Selection removes the insignificant `W`.

```
summary(lm(V ~ I*G + I*N + D*P, data=p160))
##
## Call:
## lm(formula = V ~ I * G + I * N + D * P, data = p160)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.043486 -0.011402 -0.003604  0.005756  0.040362
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.504629   0.017871  28.238 2.41e-12
## I           -0.073140   0.016171  -4.523 0.000698
## G            0.001670   0.001407   1.187 0.258170
```

```
## N             -0.008336   0.002180  -3.824 0.002423
## D              0.085791   0.019167   4.476 0.000758
## P              0.003197   0.002372   1.348 0.202486
## I:G            0.008936   0.001215   7.354 8.81e-06
## I:N            0.009027   0.002200   4.104 0.001462
## D:P           -0.007016   0.003022  -2.322 0.038656
##
## Residual standard error: 0.02482 on 12 degrees of freedom
## Multiple R-squared:  0.9344, Adjusted R-squared:  0.8907
## F-statistic: 21.37 on 8 and 12 DF,  p-value: 5.62e-06
```

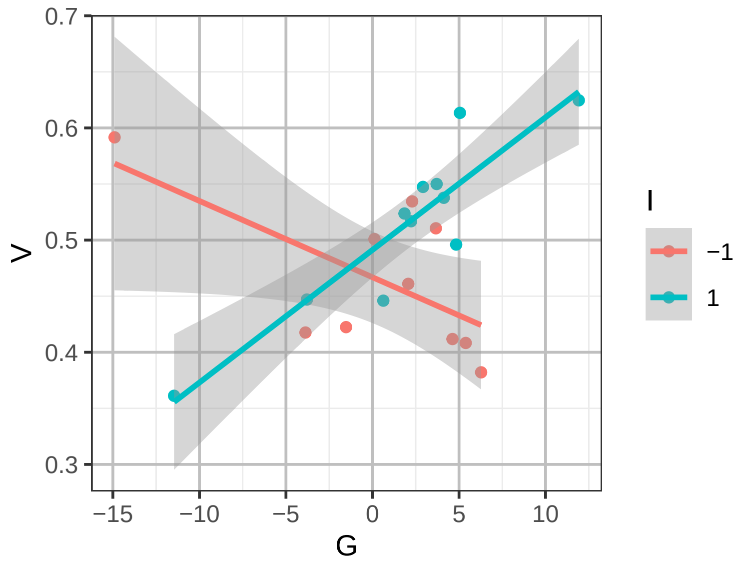
and the AIC of this model is

```
extractAIC(lm(V ~ I*G + I*N + D*P, data=p160))
## [1]    9.0000 -148.9927
```

We made a few plots to visualize the `I*G`, `I*N`, and `D*P` interactions.

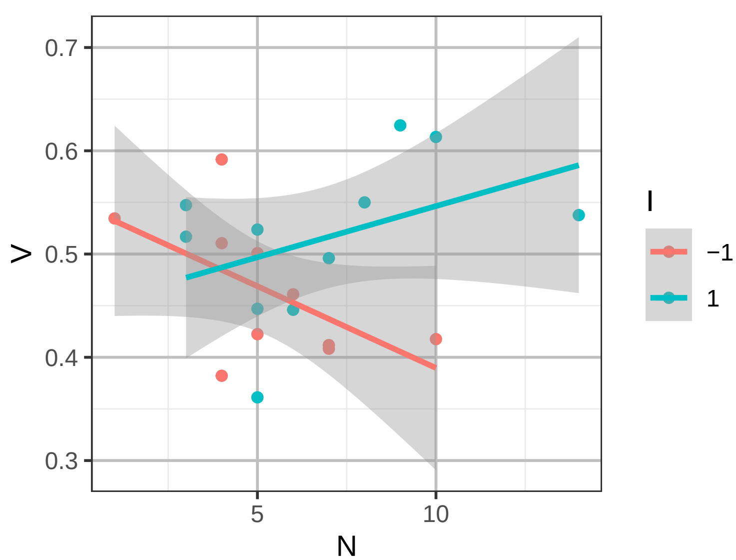
`I*G` interactions

```
library(ggplot2)
p160$I = as.factor(p160$I)
ggplot(p160, aes(x=G, y=V, col=I))+geom_point()+
        geom_smooth(method='lm', formula='y~x')
```



- When the Incumbent president was a Democrat the higher the GDP growth rate, the more votes went to the Democrat candidate
- When the Incumbent president was a Republican the higher the GDP growth rate, the fewer votes went to the Democrat candidate

```
ggplot(p160, aes(x=N, y=V, col=I))+geom_point()+
        geom_smooth(method='lm', formula='y~x')
```



```
p160$D = as.factor(p160$D)
ggplot(p160, aes(x=P, y=V, col=D))+geom_point()+
        geom_smooth(method='lm', formula='y~x')
```