Lecture 19 Variale Selection Examples

Election data

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

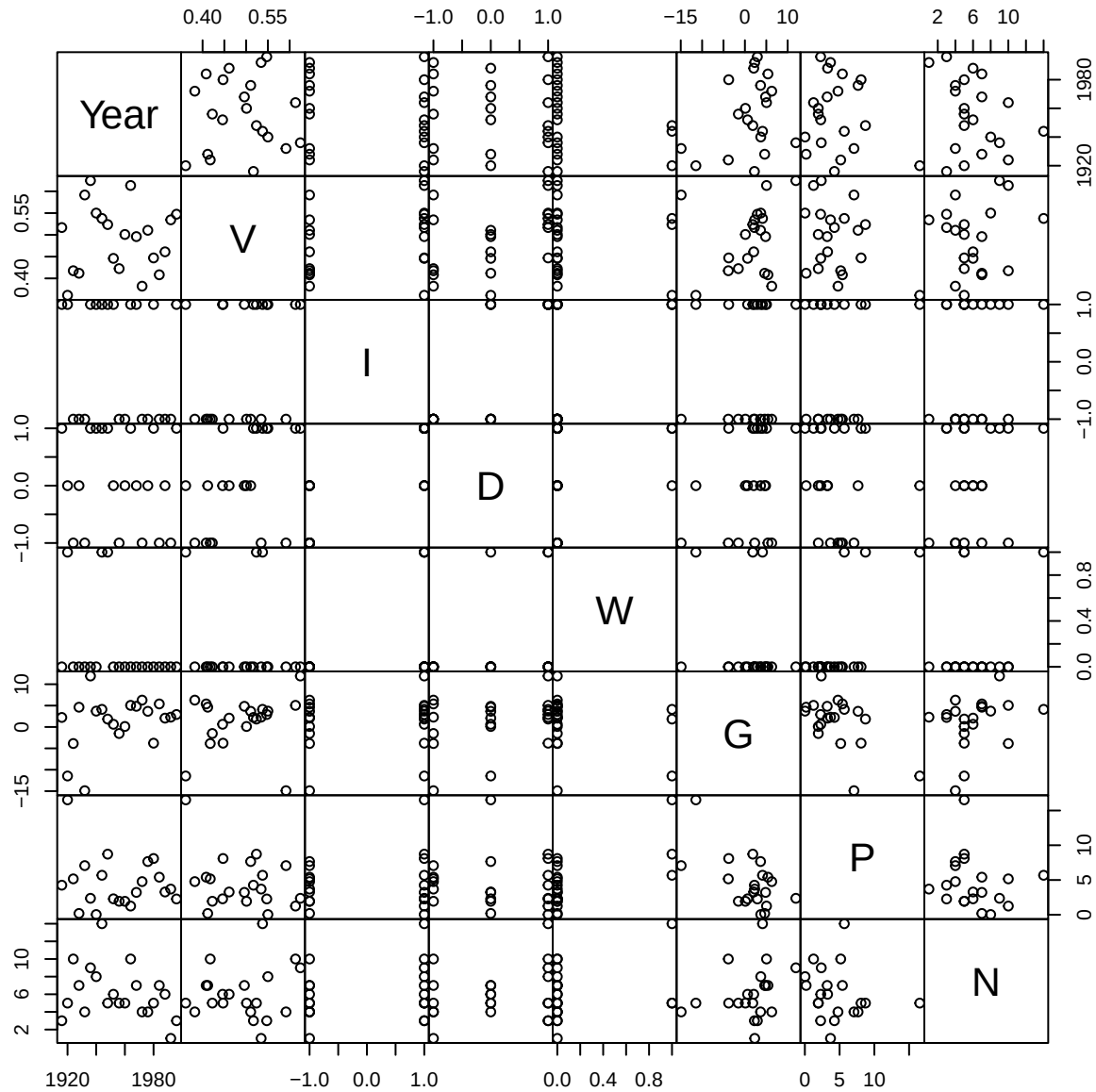
Here is a dataset from the book (p. 160) that we will use to demonstrate the FS, BE, and SW model selection algorithms.

- 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)
- 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
- 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

If $D = 1$ or -1, then $D = I$
If $D = 0$, I can be 1 or -1

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

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



Backward Elimination (BE)

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=-105
## V ~ I + D + W + G + P + N
##
##         Df Sum of Sq    RSS  AIC F value Pr(>F)
## - N      1   0.00001 0.0727 -107    0.00   0.97
## - I      1   0.00004 0.0727 -107    0.01   0.93
## - W      1   0.00009 0.0728 -107    0.02   0.90
## - G      1   0.00162 0.0743 -106    0.31   0.59
## - P      1   0.00442 0.0771 -106    0.85   0.37
## <none>               0.0727 -105
## - D      1   0.01010 0.0828 -104    1.95   0.18
##
## Step:  AIC=-107
## V ~ I + D + W + G + P
##
##         Df Sum of Sq    RSS  AIC F value Pr(>F)
## - I      1   0.00004 0.0728 -109    0.01   0.93
## - W      1   0.00014 0.0729 -109    0.03   0.87
## - G      1   0.00165 0.0744 -108    0.34   0.57
## - P      1   0.00488 0.0776 -108    1.01   0.33
## <none>               0.0727 -107
## - D      1   0.01015 0.0829 -106    2.09   0.17
##
## Step:  AIC=-109
## V ~ D + W + G + P
##
##         Df Sum of Sq    RSS  AIC F value Pr(>F)
## - W      1   0.00016 0.0729 -111    0.03  0.855
## - G      1   0.00162 0.0744 -110    0.36  0.559
## - P      1   0.00504 0.0778 -110    1.11  0.308
## <none>               0.0728 -109
## - D      1   0.02452 0.0973 -105    5.39  0.034
##
## Step:  AIC=-110.9
## V ~ D + G + P
##
##         Df Sum of Sq    RSS  AIC F value Pr(>F)
## - G      1   0.00178 0.0747 -112    0.42  0.528
## <none>               0.0729 -111
## - P      1   0.01107 0.0840 -110    2.58  0.127
```

```
## - D     1   0.02709 0.1000 -106    6.32  0.022
##
## Step:  AIC=-112.4
## V ~ D + P
##
##        Df Sum of Sq    RSS  AIC F value Pr(>F)
## <none>               0.0747 -112
## - P     1   0.00992 0.0846 -112    2.39  0.139
## - D     1   0.02556 0.1002 -108    6.16  0.023
##
## Call:
## lm(formula = V ~ D + P, data = p160)
##
## Coefficients:
## (Intercept)            D            P
##     0.51402      0.04313     -0.00602
```

```r
summary(lm(V ~ D+P, data=p160))
```

```
##
## Call:
## lm(formula = V ~ D + P, data = p160)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.10112 -0.03684 -0.00699  0.01903  0.16325
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.51402    0.02279   22.55  1.2e-14
## D            0.04313    0.01738    2.48    0.023
## P           -0.00602    0.00389   -1.55    0.139
##
## Residual standard error: 0.0644 on 18 degrees of freedom
## Multiple R-squared:  0.337,  Adjusted R-squared:  0.264
## F-statistic: 4.58 on 2 and 18 DF,  p-value: 0.0247
```

Forward Selection (FS)

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=-107.8
## V ~ 1
##
##         Df Sum of Sq    RSS  AIC F value Pr(>F)
## + D      1   0.02808 0.0846 -112    6.31  0.021
## + I      1   0.01353 0.0992 -108    2.59  0.124
## + P      1   0.01245 0.1002 -108    2.36  0.141
## <none>               0.1127 -108
## + G      1   0.00607 0.1066 -107    1.08  0.311
## + N      1   0.00242 0.1103 -106    0.42  0.526
## + W      1   0.00095 0.1117 -106    0.16  0.692
##
## Step:  AIC=-111.8
## V ~ D
##
##         Df Sum of Sq    RSS  AIC F value Pr(>F)
## + P      1   0.00992 0.0747 -112    2.39   0.14
## <none>               0.0846 -112
## + W      1   0.00681 0.0778 -112    1.58   0.23
## + I      1   0.00129 0.0833 -110    0.28   0.60
## + G      1   0.00063 0.0840 -110    0.14   0.72
## + N      1   0.00000 0.0846 -110    0.00   0.98
##
## Step:  AIC=-112.4
## V ~ D + P
##
##         Df Sum of Sq    RSS  AIC F value Pr(>F)
## <none>               0.0747 -112
## + G      1  0.001781 0.0729 -111    0.42   0.53
## + W      1  0.000319 0.0744 -110    0.07   0.79
## + N      1  0.000185 0.0745 -110    0.04   0.84
## + I      1  0.000026 0.0747 -110    0.01   0.94
##
## Call:
## lm(formula = V ~ D + P, data = p160)
##
## Coefficients:
## (Intercept)            D            P
##     0.51402      0.04313     -0.00602
```

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=-112.4
## V ~ D + P
##
##         Df Sum of Sq    RSS  AIC F value Pr(>F)
## <none>               0.0747 -112
## + G      1  0.001781 0.0729 -111    0.42   0.53
## + W      1  0.000319 0.0744 -110    0.07   0.79
## + N      1  0.000185 0.0745 -110    0.04   0.84
## + I      1  0.000026 0.0747 -110    0.01   0.94
##
## Call:
## lm(formula = V ~ D + P, data = p160)
##
## Coefficients:
## (Intercept)            D            P
##     0.51402      0.04313     -0.00602
```

Stepwise Selection (SW)

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=-111.6
## V ~ D + W
##
##         Df Sum of Sq    RSS  AIC F value Pr(>F)
## - W      1    0.0068 0.0846 -112    1.58  0.225
## <none>               0.0778 -112
## + P      1    0.0034 0.0744 -110    0.78  0.388
## + N      1    0.0004 0.0774 -110    0.08  0.778
## + I      1    0.0002 0.0776 -110    0.04  0.846
## + G      1    0.0000 0.0778 -110    0.00  0.962
## - D      1    0.0339 0.1117 -106    7.85  0.012
##
## Step:  AIC=-111.8
## V ~ D
##
##         Df Sum of Sq    RSS  AIC F value Pr(>F)
## + P      1   0.00992 0.0747 -112    2.39  0.139
## <none>               0.0846 -112
## + W      1   0.00681 0.0778 -112    1.58  0.225
## + I      1   0.00129 0.0833 -110    0.28  0.604
## + G      1   0.00063 0.0840 -110    0.14  0.717
## + N      1   0.00000 0.0846 -110    0.00  0.979
## - D      1   0.02808 0.1127 -108    6.31  0.021
##
## Step:  AIC=-112.4
## V ~ D + P
##
##         Df Sum of Sq    RSS  AIC F value Pr(>F)
## <none>               0.0747 -112
## - P      1   0.00992 0.0846 -112    2.39  0.139
## + G      1   0.00178 0.0729 -111    0.42  0.528
## + W      1   0.00032 0.0744 -110    0.07  0.790
## + N      1   0.00018 0.0745 -110    0.04  0.840
## + I      1   0.00003 0.0747 -110    0.01  0.939
## - D      1   0.02556 0.1002 -108    6.16  0.023
##
## Call:
## lm(formula = V ~ D + P, data = p160)
##
## Coefficients:
## (Intercept)            D            P
##     0.51402      0.04313     -0.00602
```

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=-113
## V ~ (I + D + P)^2 + G + W + N
##
##         Df Sum of Sq    RSS  AIC F value Pr(>F)
## - D:P    1   0.00169 0.0390 -114    0.50  0.495
## - I:D    1   0.00175 0.0391 -114    0.52  0.487
## - N      1   0.00276 0.0401 -114    0.81  0.386
## - G      1   0.00277 0.0401 -114    0.82  0.386
## <none>               0.0373 -113
## - W      1   0.00562 0.0430 -112    1.66  0.225
## - I:P    1   0.01598 0.0533 -108    4.71  0.053
##
## Step:  AIC=-114
## V ~ I + D + P + G + W + N + I:D + I:P
##
##         Df Sum of Sq    RSS  AIC F value Pr(>F)
## - I:D    1   0.00137 0.0404 -115    0.42  0.529
## - N      1   0.00198 0.0410 -115    0.61  0.450
## <none>               0.0390 -114
## - G      1   0.00396 0.0430 -114    1.22  0.291
## - W      1   0.00480 0.0438 -114    1.47  0.248
## - I:P    1   0.02446 0.0635 -106    7.52  0.018
##
## Step:  AIC=-115.3
## V ~ I + D + P + G + W + N + I:P
##
##         Df Sum of Sq    RSS  AIC F value Pr(>F)
## - N      1    0.0022 0.0426 -116    0.70 0.4191
## <none>               0.0404 -115
## - G      1    0.0042 0.0446 -115    1.35 0.2664
## - W      1    0.0060 0.0464 -114    1.92 0.1889
## - D      1    0.0123 0.0527 -112    3.95 0.0683
## - I:P    1    0.0323 0.0727 -105   10.39 0.0067
##
## Step:  AIC=-116.2
## V ~ I + D + P + G + W + I:P
##
##         Df Sum of Sq    RSS  AIC F value Pr(>F)
## - W      1   0.00392 0.0465 -116    1.29 0.2751
## <none>               0.0426 -116
```

```
## - G     1   0.00446 0.0470 -116    1.47 0.2461
## - D     1   0.01258 0.0551 -113    4.14 0.0614
## - I:P   1   0.03014 0.0727 -107    9.91 0.0071
##
## Step:  AIC=-116.4
## V ~ I + D + P + G + I:P
##
##        Df Sum of Sq    RSS  AIC F value Pr(>F)
## - G     1   0.00314 0.0496 -117    1.01  0.330
## <none>              0.0465 -116
## - D     1   0.01439 0.0609 -113    4.64  0.048
## - I:P   1   0.02636 0.0729 -109    8.51  0.011
##
## Step:  AIC=-117
## V ~ I + D + P + I:P
##
##        Df Sum of Sq    RSS  AIC F value Pr(>F)
## <none>              0.0496 -117
## - D     1    0.0119 0.0616 -114    3.85  0.067
## - I:P   1    0.0250 0.0747 -110    8.07  0.012
##
## 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

$$\text{AIC} = n\log_e(\text{SSE}_p/n) + 2p,$$
$$\text{BIC} = n\log_e(\text{SSE}_p/n) + p\log_e(n),$$

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=-102.5
## V ~ (I + D + P)^2 + G + W + N
##
##         Df Sum of Sq    RSS    AIC F value Pr(>F)
## - D:P    1   0.00169 0.0390 -104.6    0.50  0.495
## - I:D    1   0.00175 0.0391 -104.6    0.52  0.487
## - N      1   0.00276 0.0401 -104.1    0.81  0.386
## - G      1   0.00277 0.0401 -104.1    0.82  0.386
## - W      1   0.00562 0.0430 -102.6    1.66  0.225
## <none>               0.0373 -102.5
## - I:P    1   0.01598 0.0533  -98.1    4.71  0.053
##
## Step:  AIC=-104.6
## V ~ I + D + P + G + W + N + I:D + I:P
##
##         Df Sum of Sq    RSS    AIC F value Pr(>F)
## - I:D    1   0.00137 0.0404 -107.0    0.42  0.529
## - N      1   0.00198 0.0410 -106.6    0.61  0.450
## - G      1   0.00396 0.0430 -105.7    1.22  0.291
## - W      1   0.00480 0.0438 -105.3    1.47  0.248
## <none>               0.0390 -104.6
## - I:P    1   0.02446 0.0635  -97.5    7.52  0.018
##
## Step:  AIC=-107
## V ~ I + D + P + G + W + N + I:P
##
##         Df Sum of Sq    RSS    AIC F value Pr(>F)
## - N      1    0.0022 0.0426 -108.9    0.70 0.4191
## - G      1    0.0042 0.0446 -107.9    1.35 0.2664
## - W      1    0.0060 0.0464 -107.1    1.92 0.1889
## <none>               0.0404 -107.0
## - D      1    0.0123 0.0527 -104.4    3.95 0.0683
## - I:P    1    0.0323 0.0727  -97.7   10.39 0.0067
```

```
##
## Step:  AIC=-108.9
## V ~ I + D + P + G + W + I:P
##
##        Df Sum of Sq    RSS  AIC F value Pr(>F)
## - W     1   0.00392 0.0465 -110    1.29 0.2751
## - G     1   0.00446 0.0470 -110    1.47 0.2461
## <none>              0.0426 -109
## - D     1   0.01258 0.0551 -106    4.14 0.0614
## - I:P   1   0.03014 0.0727 -101    9.91 0.0071
##
## Step:  AIC=-110.1
## V ~ I + D + P + G + I:P
##
##        Df Sum of Sq    RSS  AIC F value Pr(>F)
## - G     1   0.00314 0.0496 -112    1.01  0.330
## <none>              0.0465 -110
## - D     1   0.01439 0.0609 -108    4.64  0.048
## - I:P   1   0.02636 0.0729 -104    8.51  0.011
##
## Step:  AIC=-111.8
## V ~ I + D + P + I:P
##
##        Df Sum of Sq    RSS  AIC F value Pr(>F)
## <none>              0.0496 -112
## - D     1    0.0119 0.0616 -110    3.85  0.067
## - I:P   1    0.0250 0.0747 -106    8.07  0.012
##
## 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=-133.4
## Y ~ X1 + X2
##
##         Df Sum of Sq  RSS    AIC F value Pr(>F)
## <none>               0.00 -133.4
## - X1     1      7.31 7.31   -6.8   63662 <2e-16
## - X2     1      9.01 9.01   -3.6   78444 <2e-16
##
## Call:
## lm(formula = Y ~ X1 + X2, data = hamilton)
##
## Coefficients:
## (Intercept)           X1           X2
##       -4.52         3.10         1.03
```

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.

```
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       1.7 7.31 -6.78    3.02   0.11
## <none>               9.01 -5.65
## + X1     1       0.0 9.01 -3.65    0.00   0.99
##
## Step:  AIC=-6.78
## Y ~ X2
##
##         Df Sum of Sq  RSS    AIC F value Pr(>F)
```

```
## + X1     1      7.31 0.00 -133.4   63662 <2e-16
## <none>               7.31   -6.8
##
## Step:  AIC=-133.4
## Y ~ X2 + X1
##
## Call:
## lm(formula = Y ~ X2 + X1, data = hamilton)
##
## Coefficients:
## (Intercept)           X2           X1
##       -4.52         1.03         3.10
```

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            P
##    0.504796    -0.073085     0.085849     0.000487     0.001672     0.003171
##           N          I:G          I:N          D:P
##   -0.008353     0.008939     0.009014    -0.007037
```

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
## -0.04349 -0.01140 -0.00358  0.00567  0.04043
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  0.504796   0.021360   23.63  8.9e-11
## W            0.000487   0.030371    0.02   0.9875
## I           -0.073085   0.017230   -4.24   0.0014
## G            0.001672   0.001473    1.13   0.2806
## N           -0.008353   0.002529   -3.30   0.0070
## D            0.085849   0.020348    4.22   0.0014
## P            0.003171   0.002990    1.06   0.3117
## I:G          0.008939   0.001276    7.01  2.2e-05
## I:N          0.009014   0.002433    3.71   0.0035
## D:P         -0.007037   0.003420   -2.06   0.0641
##
## Residual standard error: 0.0259 on 11 degrees of freedom
## Multiple R-squared:  0.934,  Adjusted R-squared:  0.881
## F-statistic: 17.4 on 9 and 11 DF,  p-value: 0.0000276
```

and the AIC of this model is

```
extractAIC(lm(V ~ W + I*G + I*N + D*P, data=p160))
## [1]   10 -147
```

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            N
##     0.50463     -0.07314      0.08579      0.00167      0.00320     -0.00834
##         I:G          I:N          D:P
##     0.00894      0.00903     -0.00702
```

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
## -0.04349 -0.01140 -0.00360  0.00576  0.04036
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  0.50463    0.01787   28.24  2.4e-12
## I           -0.07314    0.01617   -4.52  0.00070
## G            0.00167    0.00141    1.19  0.25817
## N           -0.00834    0.00218   -3.82  0.00242
## D            0.08579    0.01917    4.48  0.00076
## P            0.00320    0.00237    1.35  0.20249
## I:G          0.00894    0.00122    7.35  8.8e-06
## I:N          0.00903    0.00220    4.10  0.00146
## D:P         -0.00702    0.00302   -2.32  0.03866
##
## Residual standard error: 0.0248 on 12 degrees of freedom
## Multiple R-squared:  0.934,  Adjusted R-squared:  0.891
## F-statistic: 21.4 on 8 and 12 DF,  p-value: 0.00000562
```

and the AIC of this model is

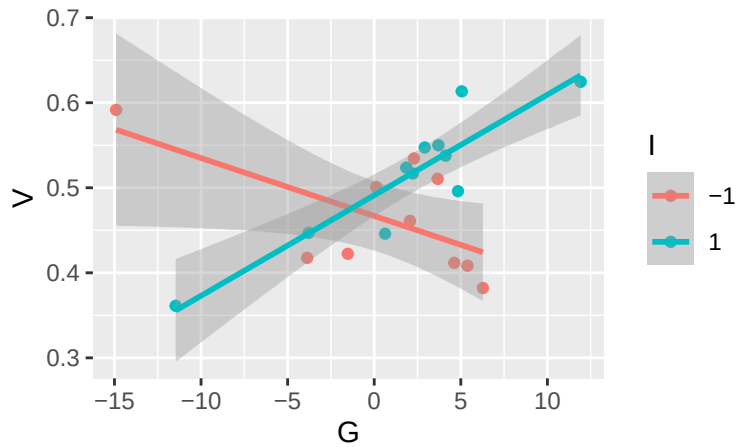
```
extractAIC(lm(V ~ I*G + I*N + D*P, data=p160))
## [1]    9 -149
```

```
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
## -0.04349 -0.01140 -0.00360  0.00576  0.04036
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.50463    0.01787   28.24  2.4e-12
## I           -0.07314    0.01617   -4.52  0.00070
## G            0.00167    0.00141    1.19  0.25817
## N           -0.00834    0.00218   -3.82  0.00242
## D            0.08579    0.01917    4.48  0.00076
## P            0.00320    0.00237    1.35  0.20249
## I:G          0.00894    0.00122    7.35  8.8e-06
## I:N          0.00903    0.00220    4.10  0.00146
## D:P         -0.00702    0.00302   -2.32  0.03866
##
## Residual standard error: 0.0248 on 12 degrees of freedom
## Multiple R-squared:  0.934,  Adjusted R-squared:  0.891
## F-statistic: 21.4 on 8 and 12 DF,  p-value: 0.00000562
```

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

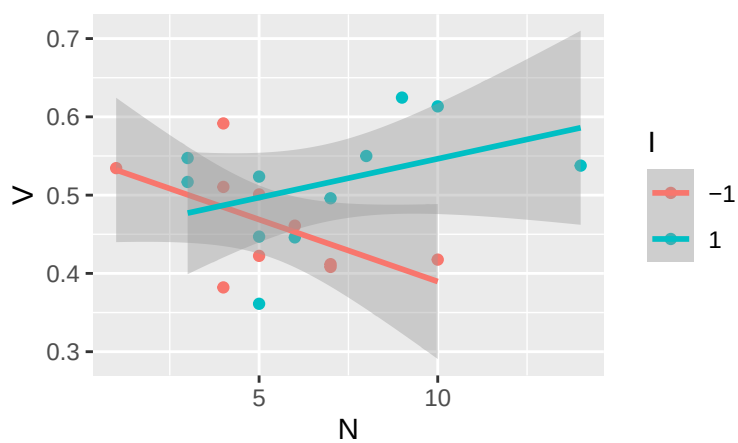
I*G interactions

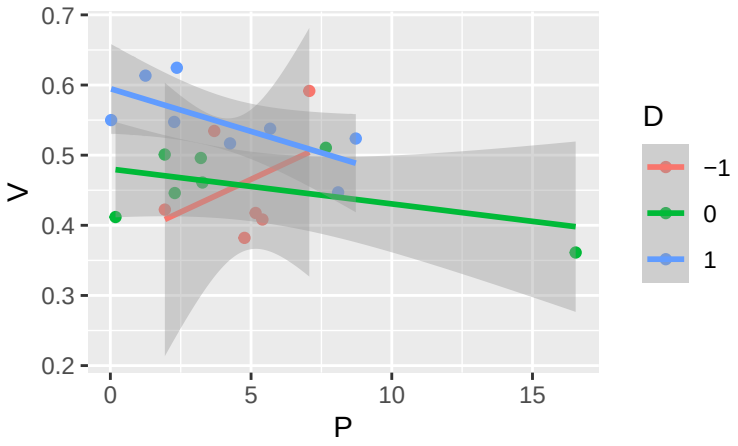
```
library(ggplot2)
p160$I = as.factor(p160$I)
ggplot(p160, aes(x=G, y=V, col=I))+geom_point()+
        geom_smooth(method='lm', formula='y~x')
```



- When the Incumbent was a Democrat, the higher the GDP growth rate, the more votes went to the Democrat candidate
- When the Incumbent 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')
```



- N = number of quarters in which GDP growth rate $> 3.2\%$ in the previous 4 years
- When the Incumbent was a Democrat, the more quarters the GDP growth rate were above 3.2% (N), the more votes went to the Democrat candidate
- When the Incumbent was a Republican the more quarters the GDP growth rate were above 3.2% (N), the fewer votes went to the Democrat candidate

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