Betas: Standardized Variables in Regression

Paul E. Johnson[1] [2]

[1]Department of Political Science

[2]Center for Research Methods and Data Analysis, University of Kansas

2014

Outline

Outline

Problem

- Regression pops out slope estimates
- How can we make sense of them?
- Can an "automatic" standardization of variables help?

Outline

Get Existential: What is Regression?

- You theorize:

$$y_i = \beta_0 + \beta_1 x1_i + \beta_2 x2_i + ... + \beta_k xk_i + e_i \quad i = 1, ..., N \tag{1}$$

- and through _____ procedure, you make estimates $\hat{\beta}_j$ with which to calculate predicted values:

$$\widehat{y}_i = \widehat{\beta}_0 + \widehat{\beta}_1 x1_i + \widehat{\beta}_2 x2_i + ... + \widehat{\beta}_k xk_i \quad i = 1, ..., N \tag{2}$$

- Everything else we do should be understood through this lens.

Yes, But What Do You DO with a Regression?

- Compare 2 cases, with inputs $(x1_0, x2_0, \ldots, xk_0)$ and $(x1_1, x2_1, \ldots, xk_1)$
- The predicted values $\widehat{y_0}$ and $\widehat{y_1}$ are different, some of the $x's$ matter
- The focus is on developing substantively interesting comparisons!
- We'd like to narrow our attention down, to concentrate on one predictor at a time.
  - $(x1_0, x2_0, \ldots, xk_0)$ and $(x1_0, x2_1, \ldots, xk_0)$
  - They only differ on $x2$, so the difference between predictions must be attributable to the change from $x2_0$ to $x2_1$.

Substantively Interesting $x2_0$ and $x2_1$

- $\hat{\beta}_j$ are "partial regression coefficients".
- Linear formula: "other things equal, a 1 unit increase in $x2_i$ causes an estimated $\hat{\beta}_2$ unit increase in the predicted value of y_i".
- No reason to say researcher can only compare variables by changing "one unit at a time"
- Know the problem's context, pick interesting values of $x2_0$ and $x2_1$ for comparison.
  - x2 represents "last year school", $x2_0$ =8th grade, $x2_1$ =high school
  - x2 represents income, $x2_0$ =\$10,000, $x2_1$ = \$100,000

Linear and Continuous X's: $\hat{\beta}$

- Maybe the calculus says it best:

$$\frac{\partial y}{\partial x2} = \hat{\beta}_2$$

- But there's no "absolute scale" for $\hat{\beta}_2$.
  - If $x's$ or y are numerically re-scaled, then the coefficients will change too.

Outline

Recall Effect of Fiddling with X's

- If one re-scales $x2_i$, replacing it with $k \cdot x2_i$, then the regression coefficient is re-scaled to $\frac{1}{k}\hat{\beta}_2$.
- If one adds or subtracts from $x2_i$, $\hat{\beta}_2$ is not changed, but the intercept $\hat{\beta}_0$ does change.
- Both multiplication and addition are apparently "harmless".

Consider Fiddling with y

- What happens if one multiplies y_i by 2?
  - doubles all the $\hat{\beta}'s$. That seems obvious.
- What happens if y_i has something added or subtracted?
  - $\hat{\beta}_0$ changes

Occupational Prestige Data from car

```
library(car)
Prestige$income <- Prestige$income/10
presmod1 <- lm(prestige ~ income + education +
    women, data = Prestige)
```

My Professionally Acceptable Regression Table

	M1	
	Estimate	(S.E.)
(Intercept)	-6.794*	(3.239)
income	0.013***	(0.003)
education	4.187***	(0.389)
women	-0.009	(0.030)
N	102	
RMSE	7.846	
R^2	0.798	
adj R^2	0.792	

$*p \leq 0.05$** $p \leq 0.01$***$p \leq 0.001$

- We are superficial, don't know much about the "Prestige" dataset
- How do we know what the slopes for income or women mean?
- Can they be compared?

predictOMatic (mostly defaults)

```
predictOMatic(presmod1, predVals = "margins",
    divider = "quantile")
```

```
$income
    income education    women      fit
1   61.100  10.73804 28.97902 38.70646
2  410.600  10.73804 28.97902 43.29736
3  593.050  10.73804 28.97902 45.69395
4  818.725  10.73804 28.97902 48.65833
5 2587.900  10.73804 28.97902 71.89751

$education
    income education    women      fit
1 679.7902    6.3800 28.97902 28.58780
2 679.7902    8.4450 28.97902 37.23321
3 679.7902   10.5400 28.97902 46.00421
```

predictOMatic (mostly defaults) ...

```
4 679.7902    12.6475 28.97902 54.82755
5 679.7902    15.9700 28.97902 68.73766

$women
    income education   women      fit
1 679.7902  10.73804  0.0000 47.09140
2 679.7902  10.73804  3.5925 47.05940
3 679.7902  10.73804 13.6000 46.97029
4 679.7902  10.73804 52.2025 46.62652
5 679.7902  10.73804 97.5100 46.22305
```

```
predictOMatic(presmod1, predVals = "margins",
    divider = "std.dev.")
```

predictOMatic (mostly defaults) ...

```
$income
   income education    women      fit
1 -169.39  10.73804 28.97902 35.67884
2  255.20  10.73804 28.97902 41.25608
3  679.79  10.73804 28.97902 46.83333
4 1104.38  10.73804 28.97902 52.41058
5 1528.97  10.73804 28.97902 57.98782

$education
    income education    women      fit
1 679.7902      5.28 28.97902 23.98250
2 679.7902      8.01 28.97902 35.41202
3 679.7902     10.74 28.97902 46.84154
4 679.7902     13.47 28.97902 58.27106
5 679.7902     16.20 28.97902 69.70058
```

predictOMatic (mostly defaults) ...

```
$women
    income education  women      fit
1 679.7902  10.73804 -34.46 47.39827
2 679.7902  10.73804  -2.74 47.11580
3 679.7902  10.73804  28.98 46.83332
4 679.7902  10.73804  60.70 46.55085
5 679.7902  10.73804  92.42 46.26838
```

```
predictOMatic(presmod1, predVals = "auto", divider
    = "quantile")
```

predictOMatic (mostly defaults) ...

```
[[1]]
      income education    women      fit
1     61.100    6.3800   0.0000 20.71900
2    410.600    6.3800   0.0000 25.30989
3    593.050    6.3800   0.0000 27.70648
4    818.725    6.3800   0.0000 30.67086
5   2587.900    6.3800   0.0000 53.91004
6     61.100    8.4450   0.0000 29.36440
7    410.600    8.4450   0.0000 33.95530
8    593.050    8.4450   0.0000 36.35189
9    818.725    8.4450   0.0000 39.31627
10  2587.900    8.4450   0.0000 62.55545
11    61.100   10.5400   0.0000 38.13541
12   410.600   10.5400   0.0000 42.72630
13   593.050   10.5400   0.0000 45.12289
14   818.725   10.5400   0.0000 48.08727
```

predictOMatic (mostly defaults) ...

```
15  2587.900   10.5400  0.0000 71.32645
16    61.100   12.6475  0.0000 46.95875
17   410.600   12.6475  0.0000 51.54964
18   593.050   12.6475  0.0000 53.94623
19   818.725   12.6475  0.0000 56.91061
20  2587.900   12.6475  0.0000 80.14979
21    61.100   15.9700  0.0000 60.86885
22   410.600   15.9700  0.0000 65.45974
23   593.050   15.9700  0.0000 67.85633
24   818.725   15.9700  0.0000 70.82071
25  2587.900   15.9700  0.0000 94.05989
26    61.100    6.3800  3.5925 20.68701
27   410.600    6.3800  3.5925 25.27790
28   593.050    6.3800  3.5925 27.67449
29   818.725    6.3800  3.5925 30.63887
30  2587.900    6.3800  3.5925 53.87805
31    61.100    8.4450  3.5925 29.33241
```

predictOMatic (mostly defaults) ...

32	410.600	8.4450	3.5925	33.92330
33	593.050	8.4450	3.5925	36.31990
34	818.725	8.4450	3.5925	39.28427
35	2587.900	8.4450	3.5925	62.52346
36	61.100	10.5400	3.5925	38.10342
37	410.600	10.5400	3.5925	42.69431
38	593.050	10.5400	3.5925	45.09090
39	818.725	10.5400	3.5925	48.05528
40	2587.900	10.5400	3.5925	71.29446
41	61.100	12.6475	3.5925	46.92675
42	410.600	12.6475	3.5925	51.51765
43	593.050	12.6475	3.5925	53.91424
44	818.725	12.6475	3.5925	56.87862
45	2587.900	12.6475	3.5925	80.11780
46	61.100	15.9700	3.5925	60.83686
47	410.600	15.9700	3.5925	65.42775
48	593.050	15.9700	3.5925	67.82434

predictOMatic (mostly defaults) ...

49	818.725	15.9700	3.5925	70.78872
50	2587.900	15.9700	3.5925	94.02790
51	61.100	6.3800	13.6000	20.59789
52	410.600	6.3800	13.6000	25.18878
53	593.050	6.3800	13.6000	27.58537
54	818.725	6.3800	13.6000	30.54975
55	2587.900	6.3800	13.6000	53.78893
56	61.100	8.4450	13.6000	29.24329
57	410.600	8.4450	13.6000	33.83419
58	593.050	8.4450	13.6000	36.23078
59	818.725	8.4450	13.6000	39.19516
60	2587.900	8.4450	13.6000	62.43434
61	61.100	10.5400	13.6000	38.01430
62	410.600	10.5400	13.6000	42.60519
63	593.050	10.5400	13.6000	45.00178
64	818.725	10.5400	13.6000	47.96616
65	2587.900	10.5400	13.6000	71.20534

predictOMatic (mostly defaults) ...

```
66    61.100   12.6475 13.6000 46.83764
67   410.600   12.6475 13.6000 51.42853
68   593.050   12.6475 13.6000 53.82512
69   818.725   12.6475 13.6000 56.78950
70  2587.900   12.6475 13.6000 80.02868
71    61.100   15.9700 13.6000 60.74774
72   410.600   15.9700 13.6000 65.33863
73   593.050   15.9700 13.6000 67.73522
74   818.725   15.9700 13.6000 70.69960
75  2587.900   15.9700 13.6000 93.93878
76    61.100    6.3800 52.2025 20.25413
77   410.600    6.3800 52.2025 24.84502
78   593.050    6.3800 52.2025 27.24161
79   818.725    6.3800 52.2025 30.20599
80  2587.900    6.3800 52.2025 53.44517
81    61.100    8.4450 52.2025 28.89953
82   410.600    8.4450 52.2025 33.49043
```

predictOMatic (mostly defaults) ...

83	593.050	8.4450	52.2025	35.88702
84	818.725	8.4450	52.2025	38.85139
85	2587.900	8.4450	52.2025	62.09058
86	61.100	10.5400	52.2025	37.67054
87	410.600	10.5400	52.2025	42.26143
88	593.050	10.5400	52.2025	44.65802
89	818.725	10.5400	52.2025	47.62240
90	2587.900	10.5400	52.2025	70.86158
91	61.100	12.6475	52.2025	46.49387
92	410.600	12.6475	52.2025	51.08477
93	593.050	12.6475	52.2025	53.48136
94	818.725	12.6475	52.2025	56.44574
95	2587.900	12.6475	52.2025	79.68492
96	61.100	15.9700	52.2025	60.40398
97	410.600	15.9700	52.2025	64.99487
98	593.050	15.9700	52.2025	67.39146
99	818.725	15.9700	52.2025	70.35584

predictOMatic (mostly defaults) ...

100	2587.900	15.9700	52.2025	93.59502
101	61.100	6.3800	97.5100	19.85066
102	410.600	6.3800	97.5100	24.44155
103	593.050	6.3800	97.5100	26.83814
104	818.725	6.3800	97.5100	29.80252
105	2587.900	6.3800	97.5100	53.04170
106	61.100	8.4450	97.5100	28.49606
107	410.600	8.4450	97.5100	33.08695
108	593.050	8.4450	97.5100	35.48355
109	818.725	8.4450	97.5100	38.44792
110	2587.900	8.4450	97.5100	61.68711
111	61.100	10.5400	97.5100	37.26707
112	410.600	10.5400	97.5100	41.85796
113	593.050	10.5400	97.5100	44.25455
114	818.725	10.5400	97.5100	47.21893
115	2587.900	10.5400	97.5100	70.45811
116	61.100	12.6475	97.5100	46.09040

predictOMatic (mostly defaults) ...

```
117   410.600    12.6475 97.5100 50.68130
118   593.050    12.6475 97.5100 53.07789
119   818.725    12.6475 97.5100 56.04227
120  2587.900    12.6475 97.5100 79.28145
121    61.100    15.9700 97.5100 60.00051
122   410.600    15.9700 97.5100 64.59140
123   593.050    15.9700 97.5100 66.98799
124   818.725    15.9700 97.5100 69.95237
125  2587.900    15.9700 97.5100 93.19155
```

```
predictOMatic(presmod1, predVals = "auto", divider
    = "std.dev.")
```

predictOMatic (mostly defaults) ...

```
[[1]]
      income education  women      fit
1    -169.39      5.28 -34.46 13.39294
2     255.20      5.28 -34.46 18.97019
3     679.79      5.28 -34.46 24.54743
4    1104.38      5.28 -34.46 30.12468
5    1528.97      5.28 -34.46 35.70193
6    -169.39      8.01 -34.46 24.82246
7     255.20      8.01 -34.46 30.39971
8     679.79      8.01 -34.46 35.97695
9    1104.38      8.01 -34.46 41.55420
10   1528.97      8.01 -34.46 47.13145
11   -169.39     10.74 -34.46 36.25198
12    255.20     10.74 -34.46 41.82923
13    679.79     10.74 -34.46 47.40647
14   1104.38     10.74 -34.46 52.98372
```

predictOMatic (mostly defaults) ...

15	1528.97	10.74	−34.46	58.56097
16	−169.39	13.47	−34.46	47.68150
17	255.20	13.47	−34.46	53.25875
18	679.79	13.47	−34.46	58.83599
19	1104.38	13.47	−34.46	64.41324
20	1528.97	13.47	−34.46	69.99049
21	−169.39	16.20	−34.46	59.11102
22	255.20	16.20	−34.46	64.68827
23	679.79	16.20	−34.46	70.26551
24	1104.38	16.20	−34.46	75.84276
25	1528.97	16.20	−34.46	81.42001
26	−169.39	5.28	−2.74	13.11047
27	255.20	5.28	−2.74	18.68772
28	679.79	5.28	−2.74	24.26496
29	1104.38	5.28	−2.74	29.84221
30	1528.97	5.28	−2.74	35.41946
31	−169.39	8.01	−2.74	24.53999

predictOMatic (mostly defaults) ...

32	255.20	8.01	−2.74	30.11724
33	679.79	8.01	−2.74	35.69448
34	1104.38	8.01	−2.74	41.27173
35	1528.97	8.01	−2.74	46.84898
36	−169.39	10.74	−2.74	35.96951
37	255.20	10.74	−2.74	41.54676
38	679.79	10.74	−2.74	47.12400
39	1104.38	10.74	−2.74	52.70125
40	1528.97	10.74	−2.74	58.27850
41	−169.39	13.47	−2.74	47.39903
42	255.20	13.47	−2.74	52.97628
43	679.79	13.47	−2.74	58.55352
44	1104.38	13.47	−2.74	64.13077
45	1528.97	13.47	−2.74	69.70801
46	−169.39	16.20	−2.74	58.82855
47	255.20	16.20	−2.74	64.40580
48	679.79	16.20	−2.74	69.98304

predictOMatic (mostly defaults) ...

49	1104.38	16.20	−2.74	75.56029
50	1528.97	16.20	−2.74	81.13753
51	−169.39	5.28	28.98	12.82800
52	255.20	5.28	28.98	18.40525
53	679.79	5.28	28.98	23.98249
54	1104.38	5.28	28.98	29.55974
55	1528.97	5.28	28.98	35.13698
56	−169.39	8.01	28.98	24.25752
57	255.20	8.01	28.98	29.83477
58	679.79	8.01	28.98	35.41201
59	1104.38	8.01	28.98	40.98926
60	1528.97	8.01	28.98	46.56650
61	−169.39	10.74	28.98	35.68704
62	255.20	10.74	28.98	41.26428
63	679.79	10.74	28.98	46.84153
64	1104.38	10.74	28.98	52.41878
65	1528.97	10.74	28.98	57.99602

predictOMatic (mostly defaults) ...

66	−169.39	13.47	28.98	47.11656
67	255.20	13.47	28.98	52.69380
68	679.79	13.47	28.98	58.27105
69	1104.38	13.47	28.98	63.84830
70	1528.97	13.47	28.98	69.42554
71	−169.39	16.20	28.98	58.54608
72	255.20	16.20	28.98	64.12332
73	679.79	16.20	28.98	69.70057
74	1104.38	16.20	28.98	75.27782
75	1528.97	16.20	28.98	80.85506
76	−169.39	5.28	60.70	12.54553
77	255.20	5.28	60.70	18.12277
78	679.79	5.28	60.70	23.70002
79	1104.38	5.28	60.70	29.27727
80	1528.97	5.28	60.70	34.85451
81	−169.39	8.01	60.70	23.97505
82	255.20	8.01	60.70	29.55229

predictOMatic (mostly defaults) ...

83	679.79	8.01	60.70	35.12954
84	1104.38	8.01	60.70	40.70679
85	1528.97	8.01	60.70	46.28403
86	−169.39	10.74	60.70	35.40457
87	255.20	10.74	60.70	40.98181
88	679.79	10.74	60.70	46.55906
89	1104.38	10.74	60.70	52.13631
90	1528.97	10.74	60.70	57.71355
91	−169.39	13.47	60.70	46.83409
92	255.20	13.47	60.70	52.41133
93	679.79	13.47	60.70	57.98858
94	1104.38	13.47	60.70	63.56583
95	1528.97	13.47	60.70	69.14307
96	−169.39	16.20	60.70	58.26361
97	255.20	16.20	60.70	63.84085
98	679.79	16.20	60.70	69.41810
99	1104.38	16.20	60.70	74.99535

predictOMatic (mostly defaults) ...

100	1528.97	16.20	60.70	80.57259
101	−169.39	5.28	92.42	12.26306
102	255.20	5.28	92.42	17.84030
103	679.79	5.28	92.42	23.41755
104	1104.38	5.28	92.42	28.99479
105	1528.97	5.28	92.42	34.57204
106	−169.39	8.01	92.42	23.69258
107	255.20	8.01	92.42	29.26982
108	679.79	8.01	92.42	34.84707
109	1104.38	8.01	92.42	40.42431
110	1528.97	8.01	92.42	46.00156
111	−169.39	10.74	92.42	35.12210
112	255.20	10.74	92.42	40.69934
113	679.79	10.74	92.42	46.27659
114	1104.38	10.74	92.42	51.85383
115	1528.97	10.74	92.42	57.43108
116	−169.39	13.47	92.42	46.55162

predictOMatic (mostly defaults) ...

```
117   255.20      13.47   92.42 52.12886
118   679.79      13.47   92.42 57.70611
119  1104.38      13.47   92.42 63.28335
120  1528.97      13.47   92.42 68.86060
121  -169.39      16.20   92.42 57.98114
122   255.20      16.20   92.42 63.55838
123   679.79      16.20   92.42 69.13563
124  1104.38      16.20   92.42 74.71287
125  1528.97      16.20   92.42 80.29012
```

One School of Thought: Get Inside the Data

- Generally preferred by economists or political scientists (possibly statisticians)
- Why are you trying to compare the effects of "women" and "income"?
- Learn More About Your Data, look for meaningful comparison cases
- Make a predicted value table. rockchalk::predictOMatic does that

Outline

Another School of Thought: Try to Convert Variables to a Common Metric

- Preferred by psychologists (and many sociologists)
- A standardized variable is calculated like so:

$$standardized\ y_i = \frac{y_i - Observed\ mean(y_i)}{Observed\ Std.Dev.(y_i)}$$

- I'm not calling that a "Z score" because Z score presumes we know the TRUE mean and standard deviation
- By definition, all standardized variables have a mean of 0 and a standard deviation of 1. See why?
- What is the common metric with standardized variables? (I'm asking, seriously)

Use Standardized Variables in Regression

- Replace y_i and $X1_i$ and $X2_i$ and $X3_i$ by standardized variables
- A standardized regression is like so:

$$\left(\frac{y_i - \bar{y}}{s_y}\right) = \beta_1^{st}\left(\frac{X1_i - \overline{X1}}{s_{X1}}\right) + \beta_2^{st}\left(\frac{X2_i - \overline{X2}}{s_{X2}}\right) + \beta_3^{st}\left(\frac{X3_i - \overline{X3}}{s_{X3}}\right) + u_i \quad (3)$$

- The estimated coefficients β^{st} are called "standardized regression coefficients"
- Coefficients we discussed until now are un-standardized parameter estimates, which in past I have labeled as b_j, just to avoid confusion with "Betas" slang
- If ALL variables are standardized, then the intercept is 0, I didn't even bother to write it in

Standardize the Numeric Data

- Unlike SPSS, R does not make standardization easy or automatic (not an oversight, probably).

```
stPrestige <- Prestige
stPrestige$income <- scale(stPrestige$income)
stPrestige$education <- scale(stPrestige$education
    )
stPrestige$women <- scale(stPrestige$women)
stPrestige$prestige <- scale(stPrestige$prestige)
presmod1st <- lm(prestige ~ income + education +
    women, data = stPrestige)
summary(presmod1st)
```

Standardize the Numeric Data ...

```
Call:
lm(formula = prestige ~ income + education + women,
    data = stPrestige)

Residuals:
     Min       1Q   Median       3Q      Max
-1.15229 -0.30999 -0.00793  0.29984  1.01744

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.822e-17  4.516e-02   0.000     1.00
income       3.242e-01  6.855e-02   4.729 7.58e-06
    ***
education    6.640e-01  6.164e-02  10.771  < 2e-16
    ***
women       -1.642e-02  5.607e-02  -0.293     0.77
```

Standardize the Numeric Data ...

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
    ' 0.1 ' ' 1

Residual standard error: 0.4561 on 98 degrees of
    freedom
Multiple R^2:  0.7982,  Adjusted R^2:  0.792
F-statistic: 129.2 on 3 and 98 DF,  p-value: < 2
    .2e-16
```

standardize function in rockchalk will automate this

Recall: presmod1 <- lm(prestige ~ income + education + women, data = Prestige)
standardize() will scan the model, rescale the variables, and give back what you want.

```
pres1st <- standardize(presmod1)
summary(pres1st)
```

```
All variables in the model matrix and the dependent variable
were centered. The centered variables have the letter "s" appended
    to their
non-centered counterparts, even constructed
variables like `x1:x2` and poly(x1,2). We agree, that's probably
ill-advised, but you asked for it by running standardize().

The rockchalk function meanCenter is a smarter option, probably.

The summary statistics of the variables in the design matrix.
             mean std.dev.
prestiges       0        1
incomes         0        1
educations      0        1
womens          0        1
```

standardize function in rockchalk will automate this ...

```
Call:
lm(formula = prestiges ~ incomes + educations + womens, data =
    stddat)

Residuals:
     Min       1Q   Median       3Q      Max
-1.15229 -0.30999 -0.00793  0.29984  1.01744

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.822e-17  4.516e-02   0.000     1.00
incomes      3.242e-01  6.855e-02   4.729 7.58e-06 ***
educations   6.640e-01  6.164e-02  10.771  < 2e-16 ***
womens      -1.642e-02  5.607e-02  -0.293     0.77
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4561 on 98 degrees of freedom
Multiple R^2:  0.7982,  Adjusted R^2:  0.792
F-statistic: 129.2 on 3 and 98 DF,  p-value: < 2.2e-16
```

Side By Side: UnStandardized and Standardized Regression Estimates

	Unstandardized		Standardized	
	Estimate	(S.E.)	Estimate	(S.E.)
(Intercept)	-6.794*	(3.239)	0.000	(0.045)
income	0.013***	(0.003)	0.324***	(0.069)
education	4.187***	(0.389)	0.664***	(0.062)
women	-0.009	(0.030)	-0.016	(0.056)
N	102		102	
RMSE	7.846		0.456	
R^2	0.798		0.798	
adj R^2	0.792		0.792	

$*p \leq 0.05$** $p \leq 0.01$***$p \leq 0.001$

Force yourself to stop and try to interpret those parameters

Notice something interesting about the t statistics

	Estimate	t value	Estimate	t value
(Intercept)	−6.79	−2.10	0.00	0.00
income	0.01	4.73	0.32	4.73
education	4.19	10.77	0.66	10.77
women	−0.01	−0.29	−0.02	−0.29

The estimated t values are identical, unstandardized on left and standardized on the right.

Why Do Some People Like Standardized Coefficients?

I'm an outsider, looking in. It seems like
They seek an easy comparison, like "a one standard deviation rise in X1 causes a $\widehat{\beta}_1^{st}$-standard-deviation-increase in y."
So, if X1 is measured in "dollars" and y is measured in pounds of elephant fat per cubic yard of shipping container, or "bushels of wheat per year", the standardization TRIES to make them comparable.

Translate between β and β^{st}

- How does the beta, say β_1^{st} differ from the unstandardized coefficient, β_1?
- Answer: its a rescaled value (recall my theme on rescaled predictors?)

$$\beta_1^{st} = \frac{s_{X1}}{s_y}\hat{\beta}_1$$

You can prove this to yourself by multiplying 3 by s_y

$$(y_i - \bar{y}) = \beta_1 \left[\frac{s_y}{s_{X1}}\right](X1_i - \overline{X1}) + \beta_2 s_y \left(\frac{X2_i - \overline{X2}}{s_{X2}}\right) + \beta_3 s_y \left(\frac{X3_i - \overline{X3}}{s_{X3}}\right) + u_i \quad (4)$$

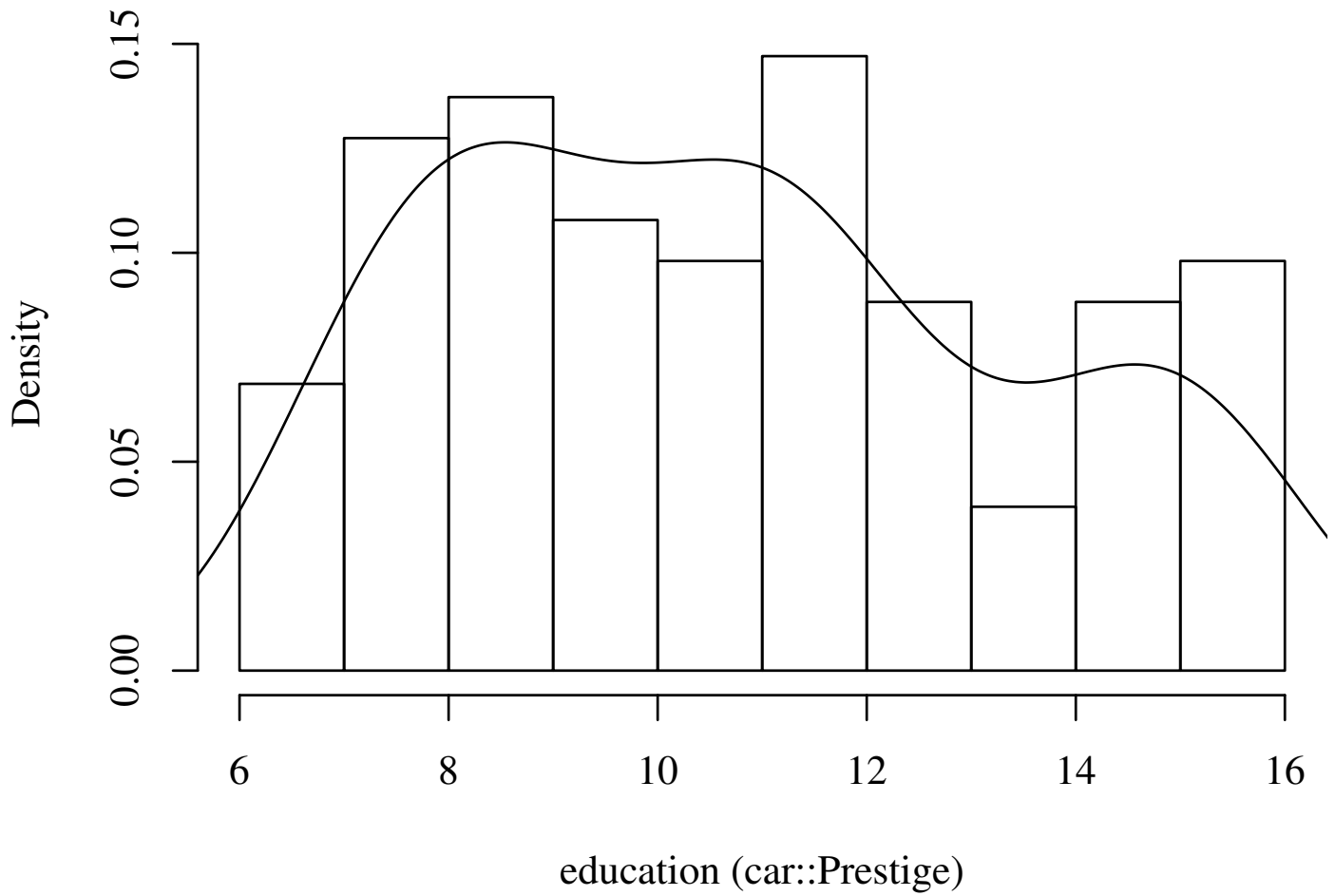
Does Standardization Make education, income, and women Comparable?

	education	income	women	prestige
Min.	6.380	61.1	0.000	14.80
1st Qu	8.445	410.6	3.592	35.23
Median	10.540	593.0	13.600	43.60
Mean	10.738	679.8	28.979	46.83
3rd Qu	12.648	818.7	52.203	59.27
Max	15.970	2587.9	97.510	87.20
Std. Dev.	2.73	424.59	31.72	17.20

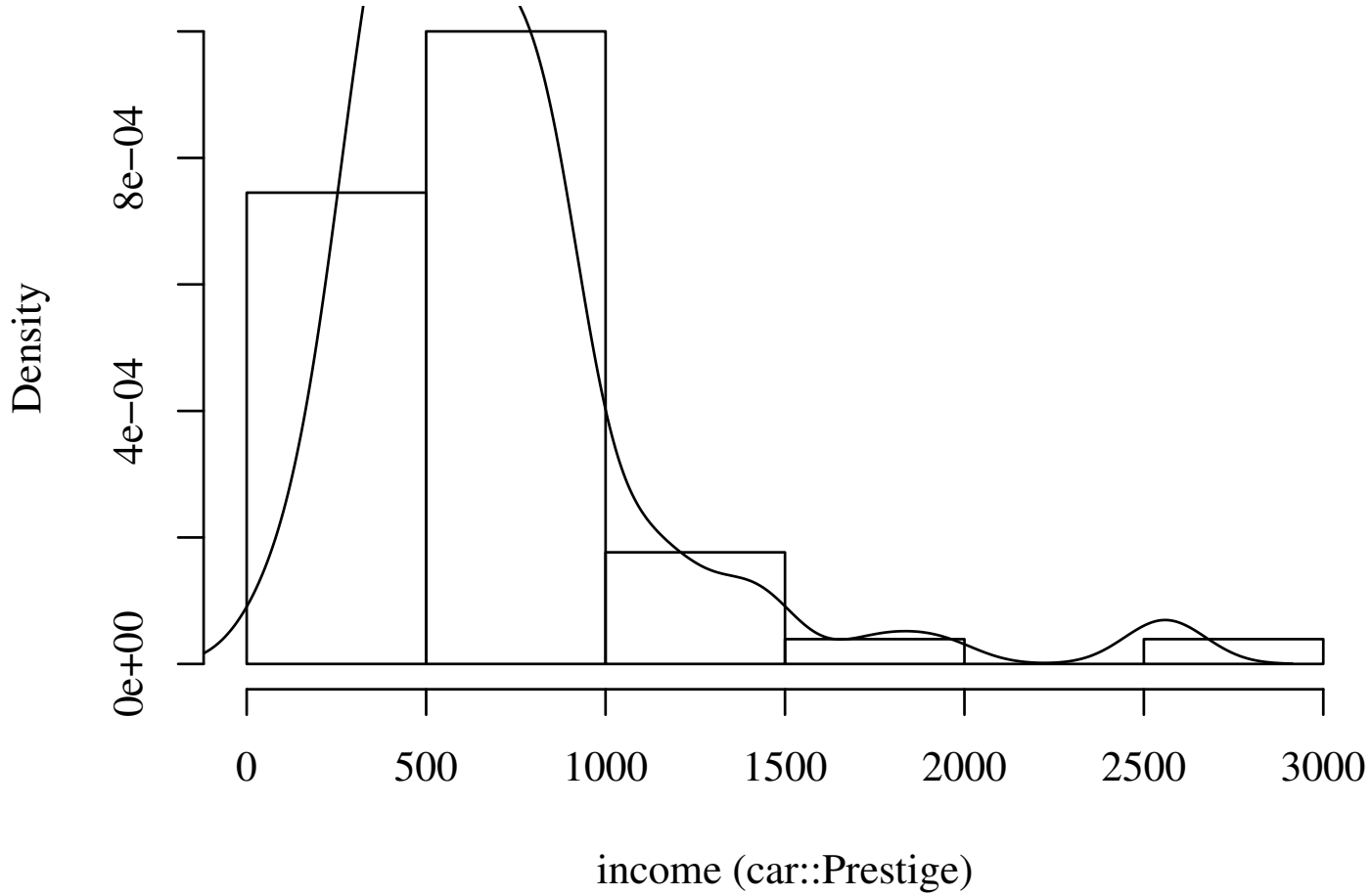
What about non-normal variables?

- Part of the motivation for standardization is the "normality" of many observed variables.
- We develop an intuition for the mean as a center point, and that a standard deviation is a step across "about" 34% of the observations.
- A two standard deviation change in a variable would be a huge step, from average to the edge.

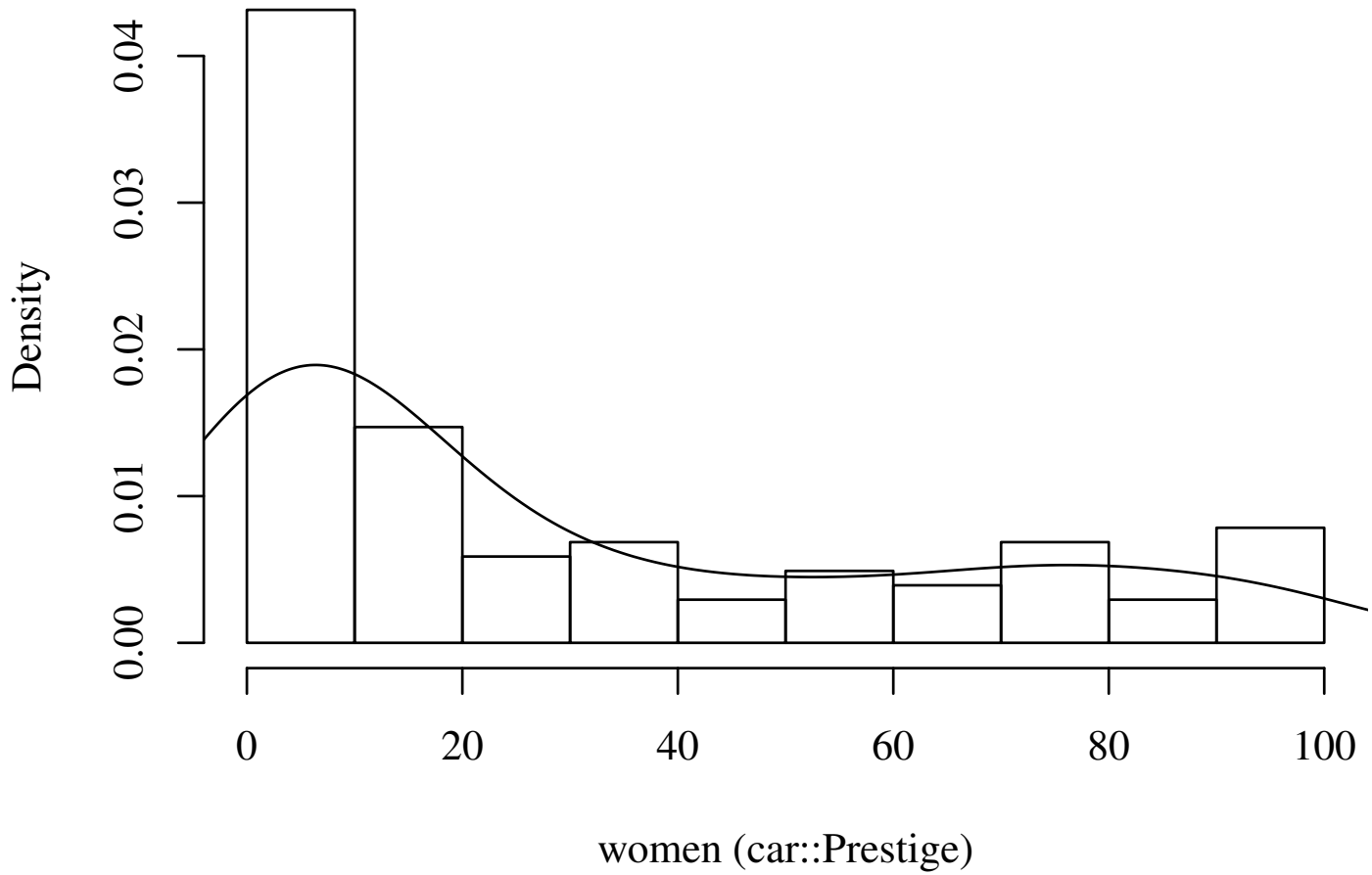
Review education



Review income



Review women



Suppose there's a Categorical Predictor "type"

- Recall that R creates "dummy variables"
- A 3 category predictor {bc, prof, wc} will be converted to dummy variables
- When we standardize education and income, should we standardize typeprof and typewc as well?

Suppose there's a Categorical Predictor "type"

- Step 1. Imagine fitting the model with unstandardized coefficients.

```
presmod2 <- lm(prestige ~ income + education +
    women + type, data=stPrestige)
```

- Step 2. Standardize. If we want to "norm" the coefficients to become comparable, should we Standardize
  - all of the variables,
  - or just the numeric ones?
- SPSS historically standardized all of the variables, even 0, 1 variables like "male" or "female".
- If we must standardize, lets only bother with numeric variables.

In rockchalk, meanCenter can be used

The ordinary, nothing standardized regression is:

```
presmod1 <- lm(prestige ~ income + education +
    women + type, data = Prestige)
```

- This use of the meanCenter function will standardize all numeric predictors and re-fit the regression

```
presmod2mc <- meanCenter(presmod1, centerDV =
    TRUE, centerOnlyInteractors = FALSE,
    standardize = TRUE)
```

Compare the factor's estimates with the Standardized Numeric Variables

	Unstandardized		Partly Standardized	
	Estimate	(S.E.)	Estimate	(S.E.)
(Intercept)	-0.814	(5.331)	-0.061	(0.108)
income	0.010***	(0.003)	0.257***	(0.065)
education	3.662***	(0.646)	0.581***	(0.102)
women	0.006	(0.030)	0.012	(0.056)
typeprof	5.905	(3.938)	0.343	(0.229)
typewc	-2.917	(2.665)	-0.170	(0.155)
N	98		98	
RMSE	7.132		0.415	
R^2	0.835		0.835	
adj R^2	0.826		0.826	

$*p \leq 0.05$** $p \leq 0.01$***$p \leq 0.001$

If you really want to Standardize everything

- R will resist you when you want to convert the model and get standardized coefficients. Its not easy to get the dummy variables out and smooth them over.
- Persuading R to do this is tough, so I wrote standardize() in rockchalk can handle it. Note the output scolds you for doing this.

```
presmod3st <- standardize(presmod2)
summary(presmod3st)
```

```
All variables in the model matrix and the dependent variable
were centered. The centered variables have the letter "s" appended
    to their
non-centered counterparts, even constructed
variables like `x1:x2` and poly(x1,2). We agree, that's probably
ill-advised, but you asked for it by running standardize().

The rockchalk function meanCenter is a smarter option, probably.

The summary statistics of the variables in the design matrix.
           mean std.dev.
prestiges     0        1
```

If you really want to Standardize everything ...

```
incomes        0         1
educations     0         1
womens         0         1
typeprofs      0         1
typewcs        0         1

Call:
lm(formula = prestiges ~ incomes + educations + womens + typeprofs +
    typewcs, data = stddat)

Residuals:
     Min       1Q   Median       3Q      Max
-0.86274 -0.26217  0.01824  0.30698  1.08206

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.318e-16  4.214e-02   0.000 1.000000
incomes      2.579e-01  6.487e-02   3.976 0.000139 ***
educations   5.889e-01  1.039e-01   5.671 1.63e-07 ***
womens       1.183e-02  5.577e-02   0.212 0.832494
typeprofs    1.615e-01  1.077e-01   1.500 0.137127
typewcs     -7.269e-02  6.642e-02  -1.094 0.276626
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If you really want to Standardize everything ...

```
Residual standard error: 0.4172 on 92 degrees of freedom
Multiple R^2:  0.8349,  Adjusted R^2:  0.826
F-statistic: 93.07 on 5 and 92 DF,  p-value: < 2.2e-16
```

Standardized Categorical Predictors Too

	Unstandardized		Standardized (except type)		All Standardized	
	Estimate	(S.E.)	Estimate	(S.E.)	Estimate	(S.E.)
(Intercept)	-0.814	(5.331)	-0.061	(0.108)	0.000	(0.042)
income	0.010***	(0.003)	0.257***	(0.065)	.	
education	3.662***	(0.646)	0.581***	(0.102)	.	
women	0.006	(0.030)	0.012	(0.056)	.	
typeprof	5.905	(3.938)	0.343	(0.229)	.	
typewc	-2.917	(2.665)	-0.170	(0.155)	.	
incomes	.		.		0.258***	(0.065)
educations	.		.		0.589***	(0.104)
womens	.		.		0.012	(0.056)
typeprofs	.		.		0.161	(0.108)
typewcs	.		.		-0.073	(0.066)
N	98		98		98	
RMSE	7.132		0.415		0.417	
R^2	0.835		0.835		0.835	
adj R^2	0.826		0.826		0.826	

$*p \leq 0.05** \; p \leq 0.01***p \leq 0.001$

Standardized Categorical Predictors Too ...

Note the summary stats in the stantardize output

- And the musical question is, DO YOU GAIN INSIGHT BY STANDARDIZING the categorical variables?
- Do you really think there is any way to formalize a comparison of $Sex \in \{0, 1\}$ and income in dollars?

Here's my answer

1. Consider standardizing a dichotomous variable. What does "the mean" mean?
   Run this in R to test your understanding. Create a variable "male" equal to 0 or 1

```
male <- rbinom(1000, 1, p = 0.55)
mean(male)
sd(male)
```

   When I ran that, I got male as a string of 0's and 1's with a mean of male is 0.542 and the standard deviation of 0.49.
   If you like standardized variables, tell me what a one standard deviation in male means to you?
2. "A one standard deviation increase in male raises the "average male" from 0.542 to 1.04."
3. "A two standard deviation increase in male results in change from 0.542 to 1.53"

Here's my answer ...

4. Can you then put that to use in interpreting a regression model?

More Problems: unknown σ.

Gary King's fine essay "How not to lie with statistics" explores many other flaws in the use of standardized coefficients. I'll summarize a couple of the points I found most persuasive.

1. Problem: We estimate by the sample standard deviation, s_{X1}, s_{X2} . But we act "as if" they were "true" values. (We don't know σ_{X1} , σ_{X2}, ...)
   1. Suppose unstandardized $\beta_1 = \beta_2$. Two variables have same effect. And they are measured on the same scale.
   2. If observed std.dev. are different, $s_{x1} \neq s_{x2}$, that will cause β_1^{st} and β_2^{st} to differ.
2. Along those lines, take a subset of the data. Even if the relationship is the same, the β^{st} will flop about because estimated standard deviations change..
   betas are not comparable across regressions. and they are not comparable within regressions.

Different y variances are a Problem Too

- Suppose we have two groups of respondents, and the same slopes apply to both

$$group\,1: \; y_i = \beta_0 + \beta_1 x1_i + \beta_2 x2_i + e1_i, \; e1_i \sim N(0, \sigma^2_{e1}) \quad (5)$$
$$group\,2: \; y_i = \beta_0 + \beta_1 x1_i + \beta_2 x2_i + e2_i, \; e2_i \sim N(0, \sigma^2_{e2}) \quad (6)$$

- This is a case of "Heteroskedasticity".
- Note only the error variances differ, so we expect the regression coefficients should be similar. Standardization of y_i has a multiplier effect across the whole line, so all of the coefficients will shrink or expand
- If we standardize the y data, we will cause the β^{st} estimates to flop about.
- Standardization complicates problem of comparing coefficients across groups.

Outline

Standardized Regression Coefficients

1. Take any "real life" data set you want that has (at least) 3 numeric variables. For ease of exposition, I will call the DV y and the IV x1, x2, and so forth, but you of course can use the "real names" when you describe the model.
    1. Regress y on $x1$. Do the usual chores: Create a scatterplot, draw the regression line, write a sentence to describe the estimated relationship. From the line you drew, pick 2 interesting values of $x1$ and write a sentence comparing the predicted values.
    2. Create histograms for y and $x1$ and super-impose the kernel density curves in order to get a mental image of the distributions. Calculate the mean and standard deviations.
    3. Create standardized variables yst and $x1st$. Run the regression of yst on $x1st$. Create a scatterplot of yst on $x1st$, draw the predicted line. For the 2 interesting values of $x1$ from the previous case, calculate the corresponding values of xst and figure out what the predicted value of yst is for those particular values. Then write a sentence comparing the predicted values of yst for those two cases.

Standardized Regression Coefficients ...

4. In your opinion, did standardization improve your ability to interpret the effect of $x1$ and $x1st$?

2. Repeat the same exercise, except this time include two or more numeric predictors. When you conduct part a), pick interesting values for all of your IV's, and make a predicted value table of this sort (I've included example "interesting values" for x1 and x2).

	value combinations		
	x1	x2	predicted y
	9	3.2	?
	9	4.6	?
	32	3.2	?
	32	4.7	?

I could show you how to make a 3D scatterplot (see the Multicollinearity lecture), but it is probably not worth your effort.

Standardized Regression Coefficients ...

3. Find a dataset with a dichotomous predictors. Or create your own dichotomous predictor by categorizing a numeric variable (In R I use the "cut" function for that). Conduct the same exercise again. Try to describe the regression model with unstandardized data, and then conduct the standardized model.
4. Let's concentrate on categorical predictors with many categories. We need data with a numeric variable for y and multi-category predictor. If x1 is type of profession, for example, then when R fits the regression of y on x1, R will create the "dummy variables" for g-1 categories when it fits a regression. You can create your own dummy variables if you want, but in R there is an easier way because you can ask the regression model to keep the data for you after it is done fitting. So instead of just running

   mod1 <- lm(y ~ x, data=dat)

   run this

   mod2 <- lm(y1~x2, data=dat, x=T, y=T)

Standardized Regression Coefficients ...

After that, the dependent variable will be saved in the model object as mod2$y and the matrix of input variables will be saved as mod2$x. So you can grab those into a new data frame like so

myNewDF <- data.frame(mod2$y, mod2$x)

Here's a "real life" example I just ran to make sure that works.

library(car)

mod1 <- lm(prestige ~ type, data=Prestige, x=TRUE, y=TRUE)

dat2 <- data.frame(mod1$y, mod1$x)

In dat2, the variables now are:

mod1.y X.Intercept. typeprof typewc

But I can beautify the names like so

colnames(dat2) <- c("prestige", "int", "prof", "wc")

The "baseline" value of the "type" is "bc", but that variable disappeared into the intercept, but we can re-create it easily.

dat2$bc <- dat2$int - dat2$prof - dat2wc

See what I mean? bc is what remains after you remove the prof and wc.

Standardized Regression Coefficients ...

After that, you can create standardized variables for “prestige”, ’bc” “prof” and “wc” and then run a regression with them.

I’m a little worried that the separate standardization of the dummy variables prof and wc throws away the information that flows from the fact that they are indicators for the same variable. Do you know what I mean? When they are “bc” “prof” and “wc”, we know that they are 0 or 1 in a logical pattern. I’ll have to think harder on that when I get some free time. Or else, you will work it out for me and then I’ll not have to do any hard thinking.