STAT 215
Polynomials, Multicollinearity

Colin Reimer Dawson

Oberlin College

4 November 2016

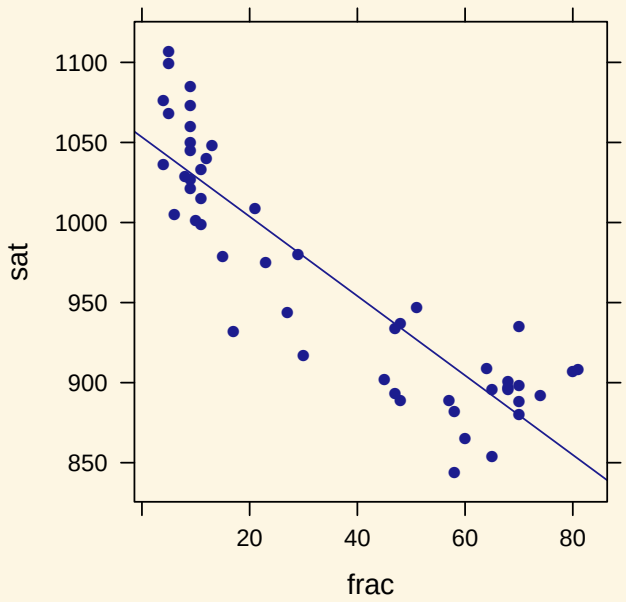
Outline

Example: State SAT Scores

```
library("mosaicData"); data("SAT")       ## sat = mean SAT score per state
slr.model <- lm(sat ~ frac, data = SAT) ## frac = % taking SAT
f.hat <- makeFun(slr.model)
```

```
xyplot(sat ~ frac, data = SAT)
plotFun(
    f.hat(frac) ~ frac, add = TRUE)
```

```
plot(slr.model, which = 1)
```



Polynomial Regression

We can create "new" predictors from old, e.g.:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_p X^p$$

$$p = \begin{cases} 1, & \text{linear} \\ 2, & \text{quadratic} \\ 3, & \text{cubic} \\ etc. & \end{cases}$$

R: Three Equivalent Methods

Method 1: Explicit Variable Creation

```
SAT.augmented <- mutate(SAT, frac.squared = frac^2)
quadratic.model <- lm(sat ~ frac + frac.squared, data = SAT.augmented)
```

Method 2: Inline transformation (note use of `I()`)

```
quadratic.model <- lm(sat ~ frac + I(frac^2), data = SAT.augmented)
```

Method 3: Using `poly()` to generate polynomials

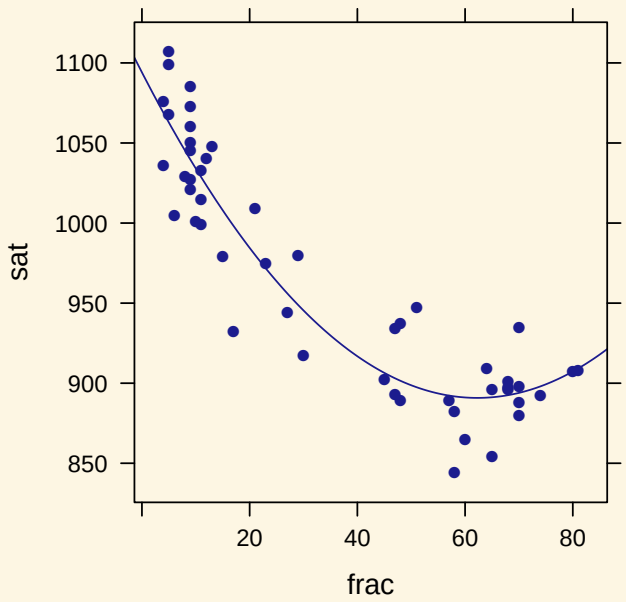
```
quadratic.model <- lm(sat ~ poly(frac, degree = 2, raw = TRUE),
                      data = SAT.augmented)
```

```
Call:
lm(formula = sat ~ frac + I(frac^2), data = SAT.augmented)

Coefficients:
(Intercept)         frac    I(frac^2)
 1094.09787     -6.52850      0.05242
```

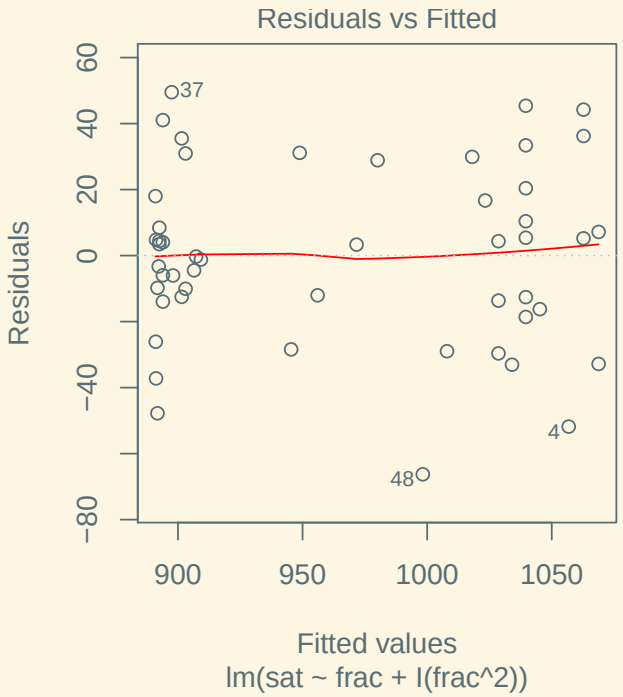
Example: State SAT Scores

```
f.hat <- makeFun(quadratic.model)
```

```
xyplot(sat ~ frac, data = SAT)
plotFun(f.hat(frac) ~ frac,
        add = TRUE)
```




```
plot(quadratic.model, which = 1)
```



ASSESS: Do we need the quadratic term?

```
summary(quadratic.model)
```

```
Call:
lm(formula = sat ~ frac + I(frac^2), data = SAT.augmented)

Residuals:
    Min      1Q  Median      3Q     Max
-66.262 -13.867   1.521  17.693  49.518

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.094e+03  9.644e+00 113.450  < 2e-16 ***
frac        -6.528e+00  7.306e-01  -8.935 1.06e-11 ***
I(frac^2)    5.242e-02  9.271e-03   5.654 8.96e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 27.2 on 47 degrees of freedom
Multiple R-squared:  0.8732,Adjusted R-squared:  0.8678
F-statistic: 161.8 on 2 and 47 DF,  p-value: < 2.2e-16
```

Selecting Polynomial Order

- Start with a higher-order model, then remove highest order term if not significant.
- Repeat until highest order term is significant.
- To be safe: nested F-test between final model and highest-order model.
- Don't remove lower order terms even if nonsignificant!

Interaction Terms and Second-Order Models

Consider the model:

$$\texttt{sat} = \beta_0 + \beta_1 \cdot \texttt{frac} + \beta_2 \cdot \texttt{expend} + \beta_3 \cdot \texttt{frac} \cdot \texttt{expend} + \varepsilon$$

where `expend` is state education expenditure per pupil.

How can we interpret β_3? Represents change in slope for `expend` for *each unit increase* in `frac` (or vice versa)

Interaction Visualization

Demo

So many models...

- How to decide among all these models?
  1. Understand the subject area! Build sensible models.
  2. Nested F-tests
  3. Other model selection techniques (next week)

The Economic Value of a College Degree

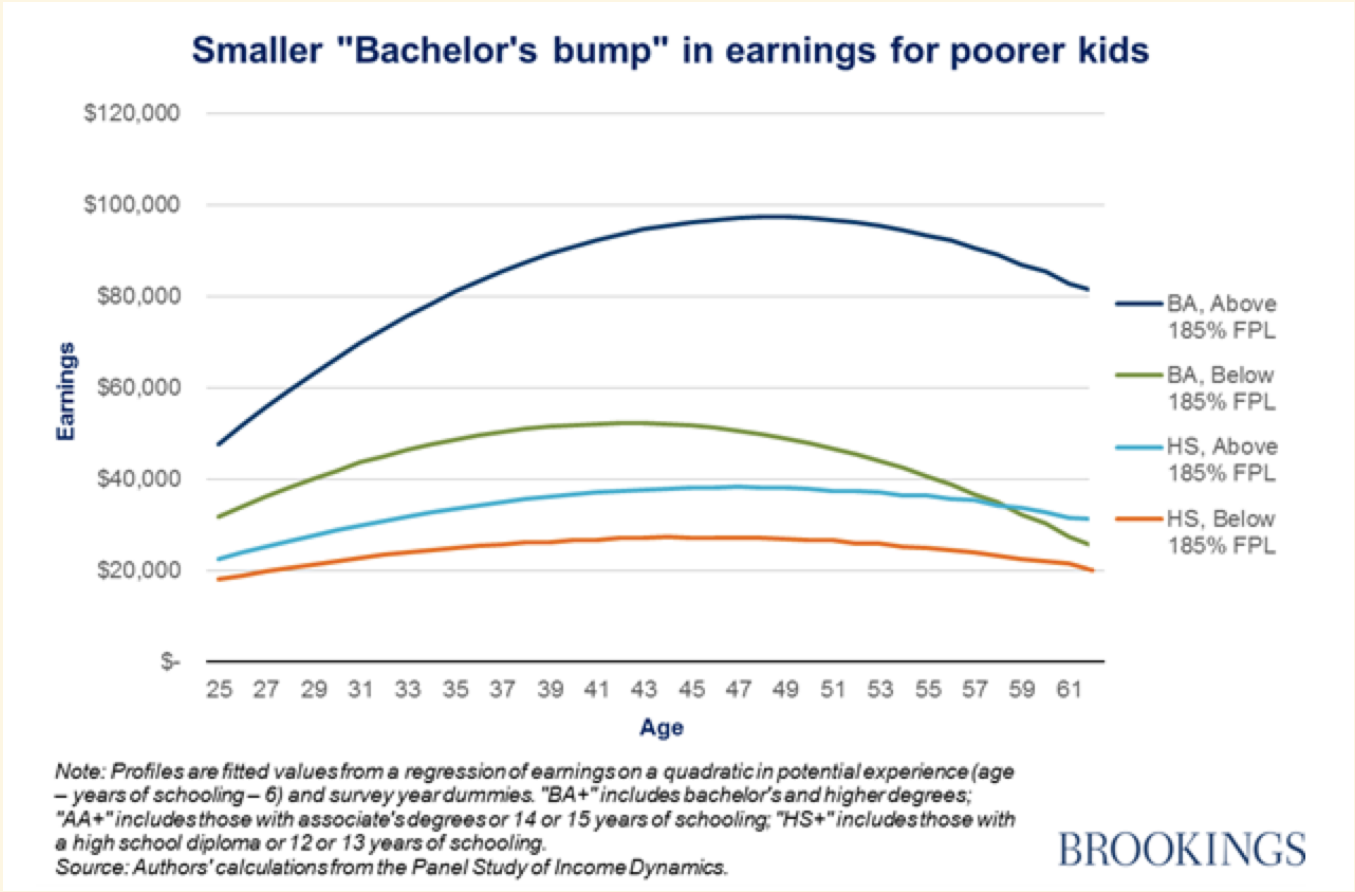


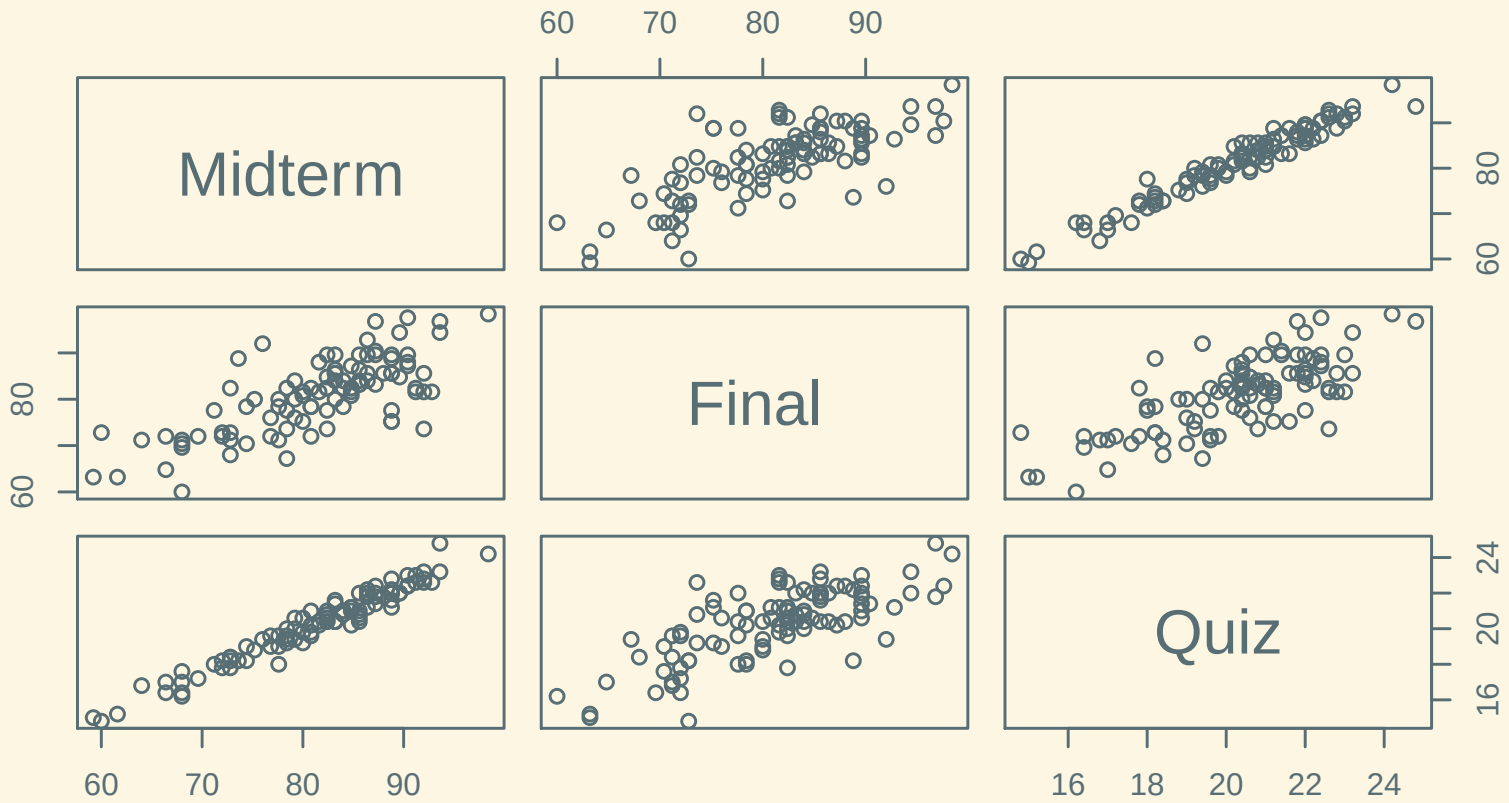
Figure: Source: `http://www.pbs.org/newshour/making-sense/if-you-grew-up-poor-your-college-degree-may-be-worth-less/`

Correlated Predictors

Worksheet

Correlated Variables

```
plot(Scores)
```



Correlated Variables

```
cor(Scores)
```

```
             Midterm     Final      Quiz
Midterm 1.0000000 0.7334905 0.9745957
Final   0.7334905 1.0000000 0.7397381
Quiz    0.9745957 0.7397381 1.0000000
```

SLR Model: Midterm Only

```
summary(m.midterm <- lm(Final ~ Midterm, data = Scores))
```

```
Call:
lm(formula = Final ~ Midterm, data = Scores)

Residuals:
     Min       1Q   Median       3Q      Max
-15.0320  -2.7025  -0.1945   3.3716  15.0110

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 21.68490    5.57328   3.891 0.000182 ***
Midterm      0.72769    0.06812  10.683  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.474 on 98 degrees of freedom
Multiple R-squared:  0.538,Adjusted R-squared:  0.5333
F-statistic: 114.1 on 1 and 98 DF,  p-value: < 2.2e-16
```

SLR Model: Quiz Only

```
summary(m.quiz <- lm(Final ~ Quiz, data = Scores))
```

```
Call:
lm(formula = Final ~ Quiz, data = Scores)

Residuals:
     Min       1Q   Median       3Q      Max
-14.0811  -2.8279   0.0806   3.3445  13.9445

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  21.8043     5.4604   3.993 0.000126 ***
Quiz          2.9149     0.2678  10.883  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.419 on 98 degrees of freedom
Multiple R-squared:  0.5472,Adjusted R-squared:  0.5426
F-statistic: 118.4 on 1 and 98 DF,  p-value: < 2.2e-16
```

MLR Model: Midterm and Quiz

```
summary(m.both <- lm(Final ~ Midterm + Quiz, data = Scores))
```

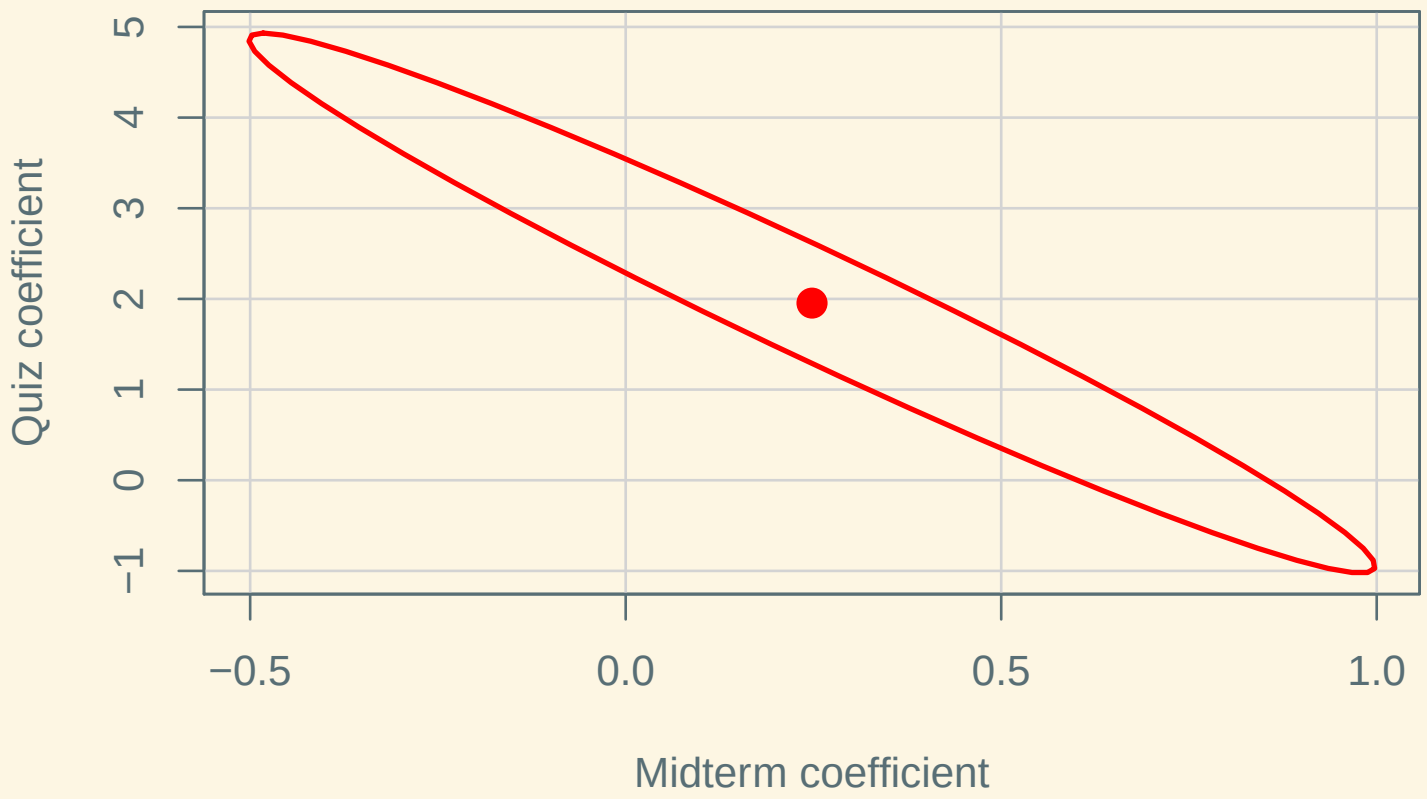
```
Call:
lm(formula = Final ~ Midterm + Quiz, data = Scores)

Residuals:
     Min       1Q   Median       3Q      Max
-14.4826  -2.9728   0.0513   3.1453  14.1414

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  21.0855     5.5388   3.807 0.000247 ***
Midterm       0.2481     0.3016   0.823 0.412717
Quiz          1.9545     1.1979   1.632 0.105993
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.428 on 97 degrees of freedom
Multiple R-squared:  0.5503,Adjusted R-squared:  0.5411
F-statistic: 59.36 on 2 and 97 DF,  p-value: < 2.2e-16
```

Confidence Intervals

```
confint(m.midterm)

                   2.5 %     97.5 %
(Intercept) 10.6249111 32.7448870
Midterm      0.5925106  0.8628613

confint(m.quiz)

                  2.5 %    97.5 %
(Intercept) 10.968290 32.640322
Quiz         2.383376  3.446427

confint(m.both)

                   2.5 %     97.5 %
(Intercept) 10.0924950 32.0784591
Midterm     -0.3504585  0.8466639
Quiz        -0.4229139  4.3319161
```

Confidence Ellipse

```
confidenceEllipse(m.both)
```



Elliptical Axes

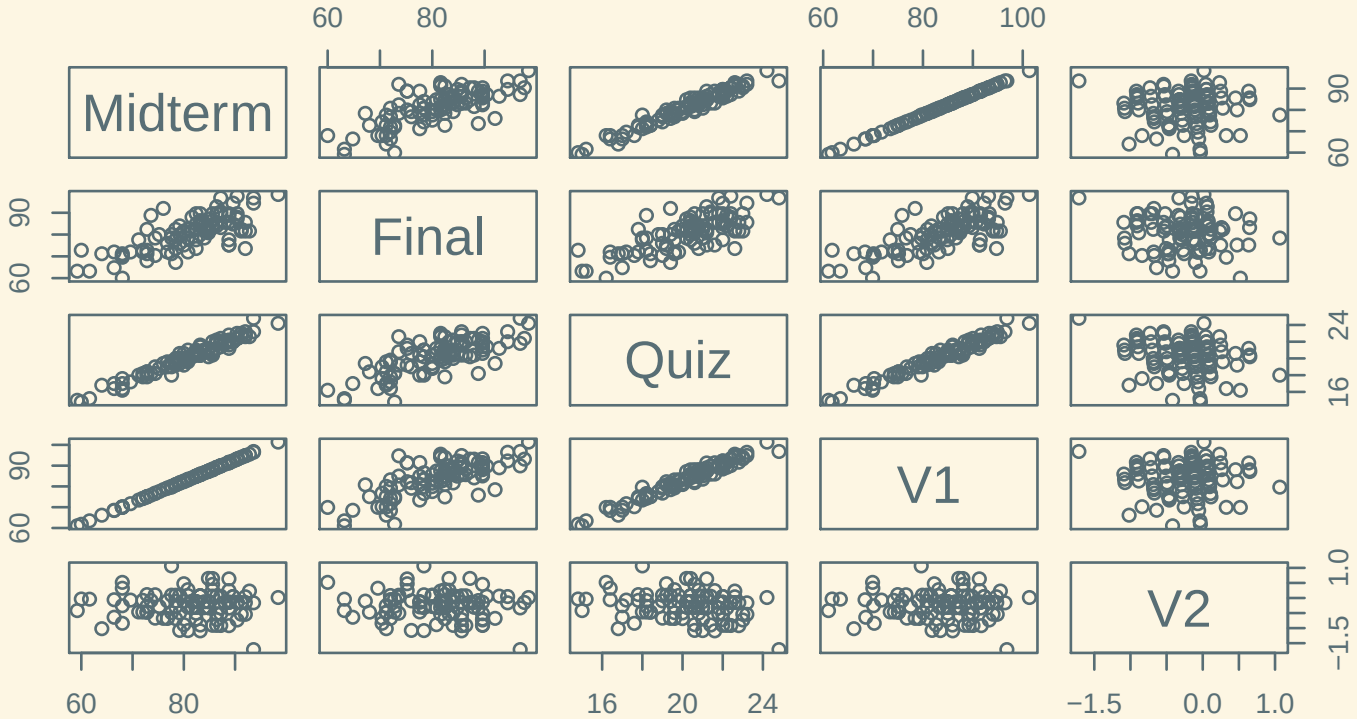
```
select(Scores, Midterm, Quiz) %>% cov() %>% eigen()
```

```
$values
[1] 69.161619  0.195581

$vectors
           [,1]       [,2]
[1,] -0.9710244  0.2389805
[2,] -0.2389805 -0.9710244
```

```
Scores.augmented <-
    mutate(Scores,
           V1 = 0.9710244 * Midterm + 0.2389805 * Quiz,
           V2 = 0.2389805 * Midterm - 0.9710244 * Quiz)
```

Elliptical Axes

```
plot(Scores.augmented)
```



Elliptical Axes

```
cor(Scores.augmented)
```

```
               Midterm      Final       Quiz            V1            V2
Midterm 1.00000000  0.7334905  0.9745957  9.999144e-01  1.308627e-02
Final   0.73349045  1.0000000  0.7397381  7.348815e-01 -1.014838e-01
Quiz    0.97459573  0.7397381  1.0000000  9.774433e-01 -2.111984e-01
V1      0.99991437  0.7348815  0.9774433  1.000000e+00 -3.036446e-07
V2      0.01308627 -0.1014838 -0.2111984 -3.036446e-07  1.000000e+00
```

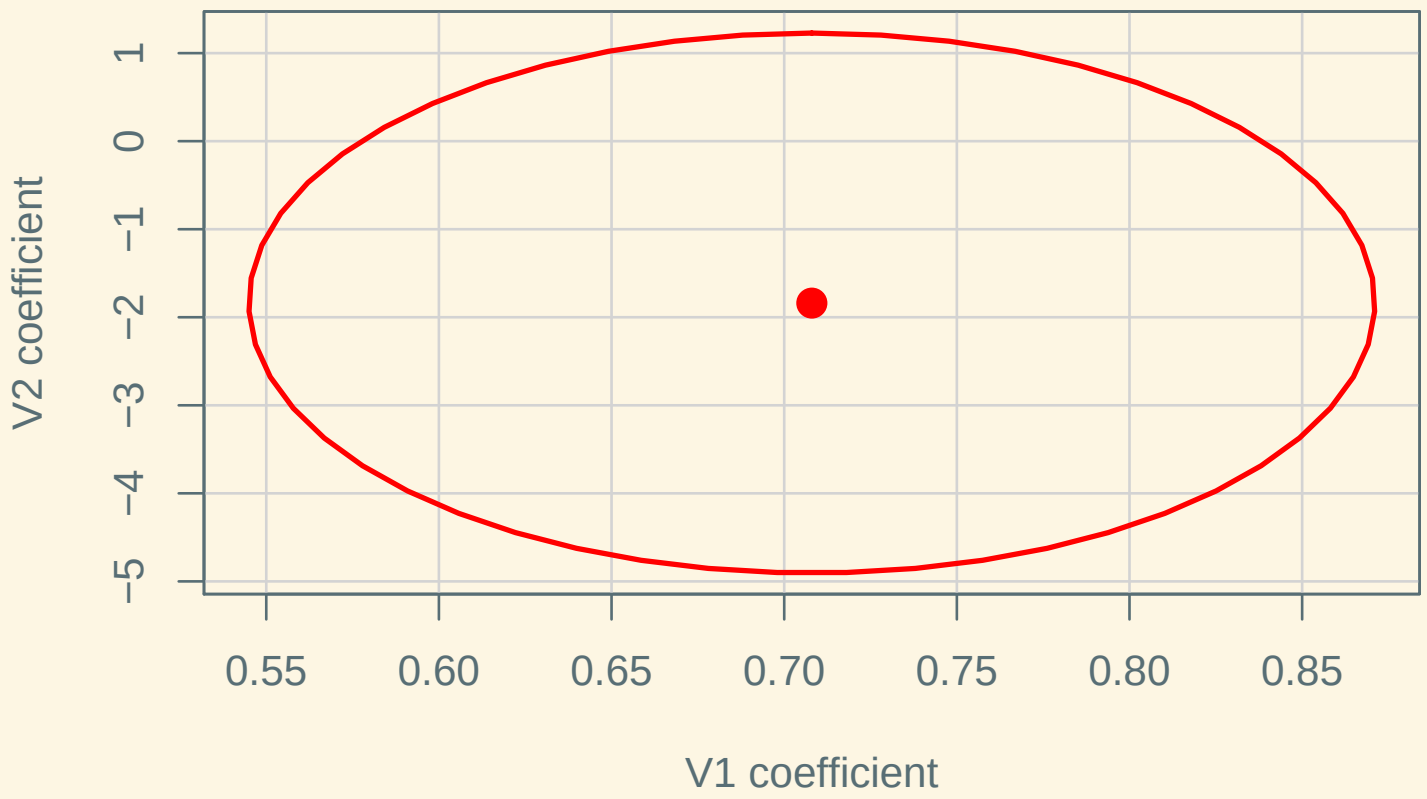
Orthogonal Predictors

```
m.rotated <- lm(Final ~ V1 + V2, data = Scores.augmented); summary(m.rotated)
```

```
Call:
lm(formula = Final ~ V1 + V2, data = Scores.augmented)

Residuals:
     Min       1Q   Median       3Q      Max
-14.4826  -2.9728   0.0513   3.1453  14.1414

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 21.08548    5.53880   3.807 0.000247 ***
V1           0.70800    0.06559  10.794  < 2e-16 ***
V2          -1.83858    1.23350  -1.491 0.139327
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.428 on 97 degrees of freedom
Multiple R-squared:  0.5503,Adjusted R-squared:  0.5411
F-statistic: 59.36 on 2 and 97 DF,  p-value: < 2.2e-16
```

Orthogonal Predictors

```
confidenceEllipse(m.rotated)
```



Multicollinearity

When one *predictor* is highly *predictable* from the other predictors, the model suffers from **multicollinearity**

One measure: R^2 from a model predicting X_j using $X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_k$.

Rough rule: If this R^2 is > 0.80, test/intervals for coefficients may not be meaningful.

Equivalently: VIF (Variance Inflation Factor) $= \frac{1}{1-R^2} > 5$

Variance Inflation Factor

```
m.midterm <- lm(Midterm ~ Quiz, data = Scores)
summary(m.midterm)$r.squared
```

```
[1] 0.9498368
```

```
m.quiz <- lm(Quiz ~ Midterm, data = Scores)
summary(m.quiz)$r.squared
```

```
[1] 0.9498368
```

```
vif(m.both)
```

```
 Midterm     Quiz
19.93495 19.93495
```

```
vif(m.rotated)
```

```
V1 V2
 1  1
```

Remedies for Multicollinearity

1. Remove redundant predictors
2. Combine predictors into a scale
3. Use the multicollinear model anyway, just don't use tests/intervals for individual coefficients.