STAT 213
Multicollinearity

Colin Reimer Dawson

Oberlin College

March 12, 2018

Outline

Correlated Predictors

Diagnosis and Remediation

Collinearity and Simpson's Paradox

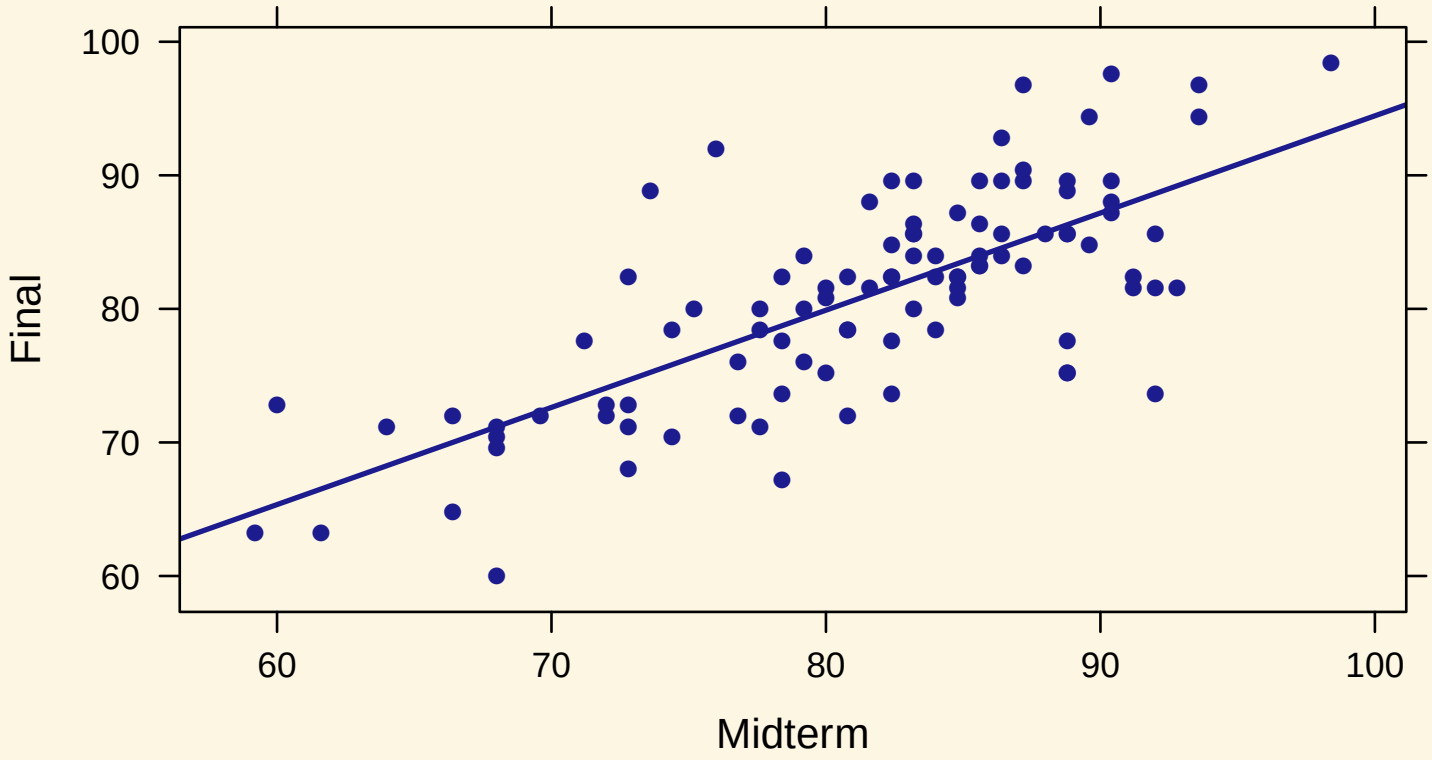
Predicting Final Exam Scores

A lazy stats instructor wants to leave early for summer vacation, and so decides to use a regression model fit on last semester's grades to predict final exam scores using midterm and quiz scores, instead of actually grading the final exam[1]

[1] Everything about this example is of course entirely fictional, including the data...

Midterm and Final

```
xyplot(Final ~ Midterm, data = Scores, type = c("p", "r"))
```



SLR Model: Midterm Only

```
m.midterm <- lm(Final ~ Midterm, data = Scores)
m.midterm %>% summary() %>% coefficients() %>% round(3)
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    21.685      5.573   3.891        0
Midterm         0.728      0.068  10.683        0
```

```
m.midterm %>% rsquared() %>% round(2)
```

```
[1] 0.54
```

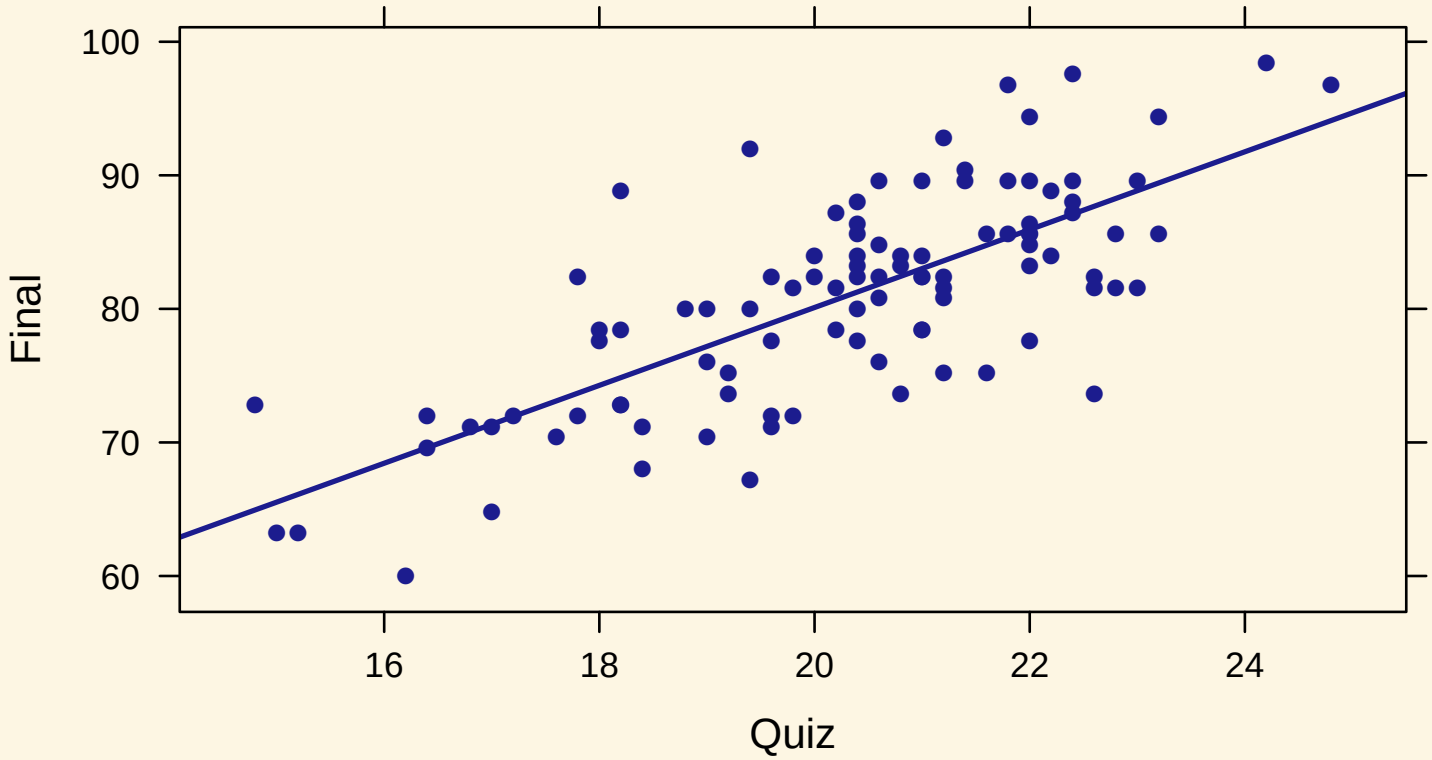
```
m.midterm %>% confint() %>% round(2)
```

```
             2.5 % 97.5 %
(Intercept) 10.62  32.74
Midterm      0.59   0.86
```

Strong evidence that midterm score helps predict final score...

Quiz and Final

```
xyplot(Final ~ Quiz, data = Scores, type = c("p", "r"))
```



SLR Model: Quiz Only

```
m.quiz <- lm(Final ~ Quiz, data = Scores)
m.quiz %>% summary() %>% coefficients() %>% round(3)
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    21.804      5.460   3.993        0
Quiz            2.915      0.268  10.883        0
```

```
m.quiz %>% rsquared() %>% round(2)
```

```
[1] 0.55
```

```
m.quiz %>% confint() %>% round(2)
```

```
             2.5 % 97.5 %
(Intercept) 10.97  32.64
Quiz         2.38   3.45
```

Strong evidence that quiz score helps predict final score...

MLR Model: Midterm and Quiz

```
m.both <- lm(Final ~ Midterm + Quiz, data = Scores)
m.both %>% summary() %>% coefficients() %>% round(3)
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    21.085      5.539   3.807    0.000
Midterm         0.248      0.302   0.823    0.413
Quiz            1.955      1.198   1.632    0.106
```

```
m.both %>% rsquared() %>% round(2)
```

```
[1] 0.55
```

```
m.both %>% confint() %>% round(2)
```

```
             2.5 % 97.5 %
(Intercept) 10.09  32.08
Midterm     -0.35   0.85
Quiz        -0.42   4.33
```

Neither is significant in the joint model...

Confidence Intervals

```
m.midterm %>% confint() %>% round(digits = 2)

                2.5 % 97.5 %
    (Intercept) 10.62  32.74
    Midterm      0.59   0.86

m.quiz %>% confint() %>% round(digits = 2)

                2.5 % 97.5 %
    (Intercept) 10.97  32.64
    Quiz         2.38   3.45

m.both %>% confint() %>% round(digits = 2)

                2.5 % 97.5 %
    (Intercept) 10.09  32.08
    Midterm     -0.35   0.85
    Quiz        -0.42   4.33
```

What's Going On?

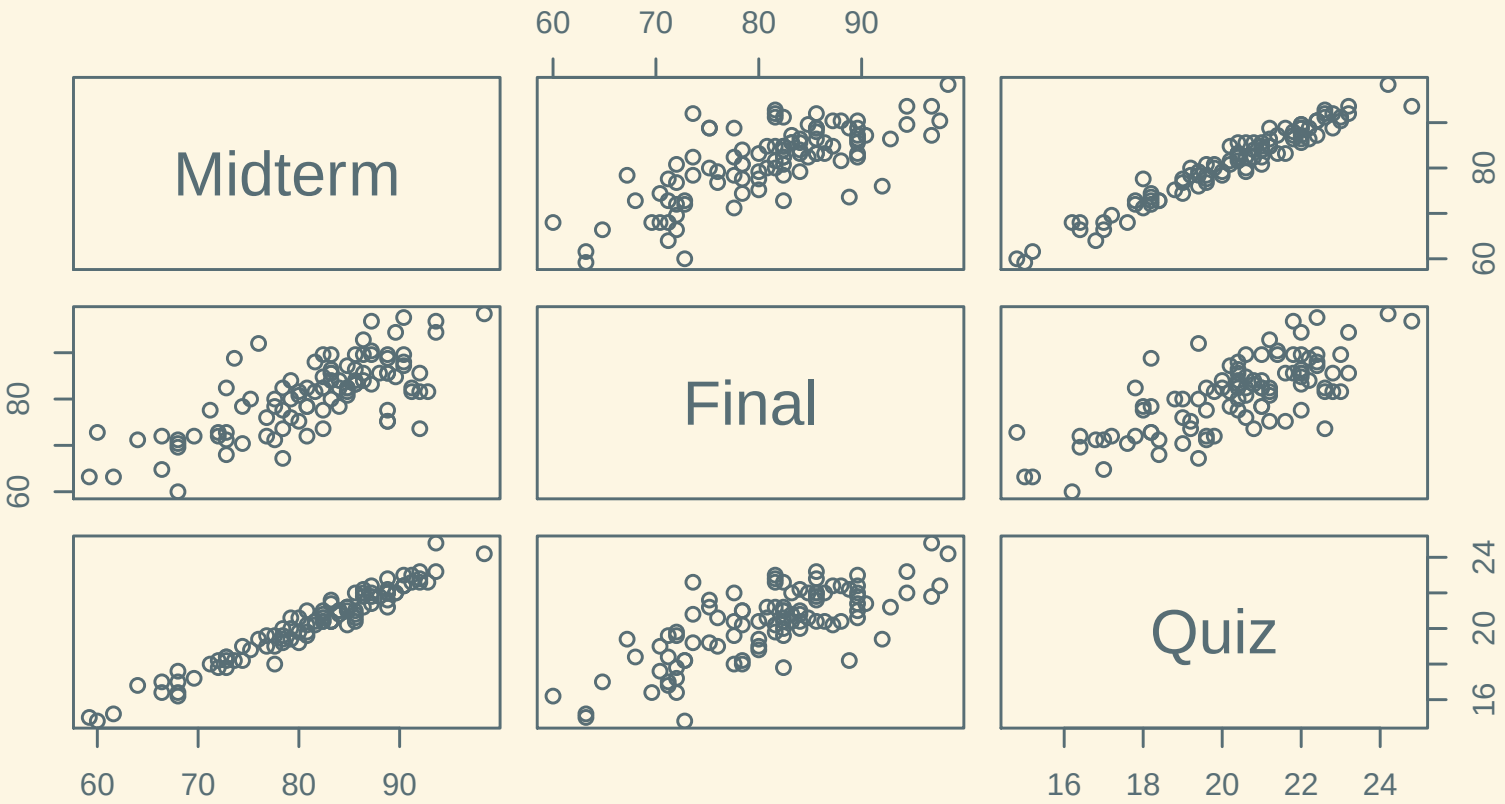
Outline

Correlated Predictors

Diagnosis and Remediation

Collinearity and Simpson's Paradox

Correlated Predictors



Correlated Predictors

```
cor(Scores) %>% round(digits = 2)
```

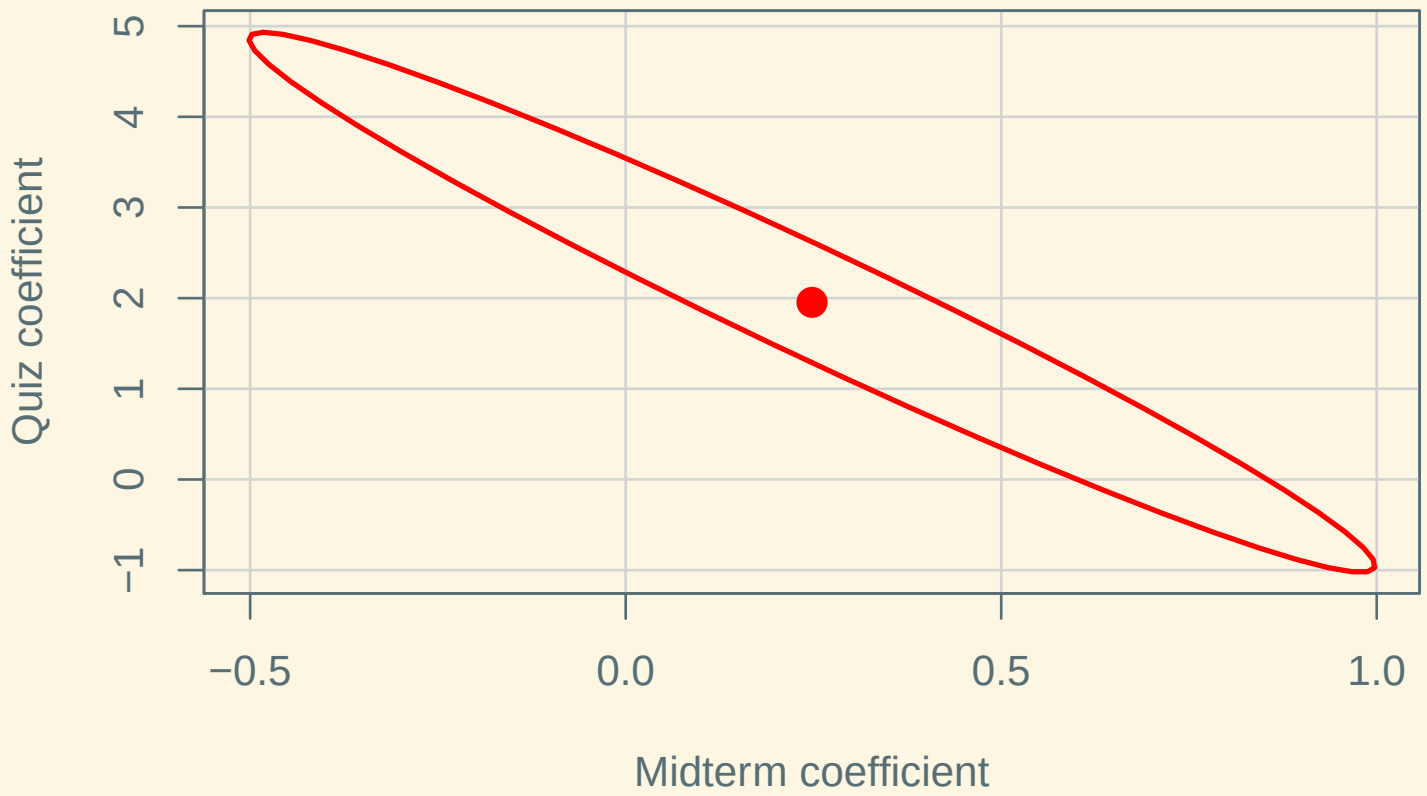
```
            Midterm Final Quiz
Midterm        1.00  0.73 0.97
Final          0.73  1.00 0.74
Quiz           0.97  0.74 1.00
```

Redundant Information

- Since `Midterm` and `Quiz` are both so highly predictable from the other (that is, they are **collinear**, we don't gain much by using both compared to using just one
- The t-tests and confidence intervals reflect this: either one could well have a coefficient of zero in the combined model

Confidence Ellipse

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



Outline

Correlated Predictors

Diagnosis and Remediation

Collinearity and Simpson's Paradox

Multicollinearity: Diagnosis

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

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

Rough rule: If this R^2 is > 0.80, interpret test and intervals for coefficients with great caution.

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

Variance Inflation Factor

```
m.midterm.from.quiz <- lm(Midterm ~ Quiz, data = Scores)
rsquared(m.midterm.from.quiz) %>% round(2)
```

```
[1] 0.95
```

```
m.quiz.from.midterm <- lm(Quiz ~ Midterm, data = Scores)
rsquared(m.quiz.from.midterm) %>% round(2)
```

```
[1] 0.95
```

```
library(car) # needed for vif()
vif(m.both) %>% round(2)
```

```
Midterm    Quiz
  19.93   19.93
```

Can We Just Look at Correlations?

```
library(Stat2Data); data(Pulse)
Pulse2 <-
    mutate(
        Pulse,
        InvActive = 1 / Active,
        InvRest = 1 / Rest,
        BMI = Wgt / Hgt^2 * 703)
```

A MLR Model

```
pulse.model <-
    lm(InvActive ~ InvRest + Hgt + Wgt + BMI + Exercise, data = Pulse2)
pulse.model %>% summary() %>% coefficients() %>% round(4)
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.0257     0.0156 -1.6470   0.1009
InvRest        0.6209     0.0645  9.6306   0.0000
Hgt            0.0004     0.0002  1.8996   0.0588
Wgt           -0.0001     0.0000 -1.6997   0.0906
BMI            0.0005     0.0003  1.4106   0.1597
Exercise       0.0001     0.0002  0.6165   0.5382
```

So, Weight, BMI and Exercise aren't useful predictors, if we know InvRest and Hgt, right?

Nested Test

```
reduced.model <- lm(InvActive ~ InvRest + Hgt, data = Pulse2)
anova(reduced.model, pulse.model)
```

```
Analysis of Variance Table

Model 1: InvActive ~ InvRest + Hgt
Model 2: InvActive ~ InvRest + Hgt + Wgt + BMI + Exercise
  Res.Df        RSS Df  Sum of Sq     F  Pr(>F)
1    229 0.00074515
2    226 0.00071617  3 2.8985e-05 3.049 0.02947 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Oh, I guess they are useful collectively.

Pairwise Correlations vs. VIF

```
Pulse2 %>% select(InvRest, Hgt, Wgt, BMI, Exercise) %>% cor() %>% round(2)
```

	InvRest	Hgt	Wgt	BMI	Exercise
InvRest	1.00	0.22	0.17	0.09	0.53
Hgt	0.22	1.00	0.75	0.31	0.18
Wgt	0.17	0.75	1.00	0.85	0.16
BMI	0.09	0.31	0.85	1.00	0.12
Exercise	0.53	0.18	0.16	0.12	1.00

```
vif(pulse.model) %>% round(2)
```

InvRest	Hgt	Wgt	BMI	Exercise
1.43	52.38	173.17	82.64	1.44

Pairwise correlations don't tell the whole story! Any *two* of Height, Weight, and BMI let us fill in the third one. They are **multi**collinear.

Multicollinearity: Remedies

If we find that some predictors suffer from high multicollinearity (guide: $VIF > 5$), what can we do?

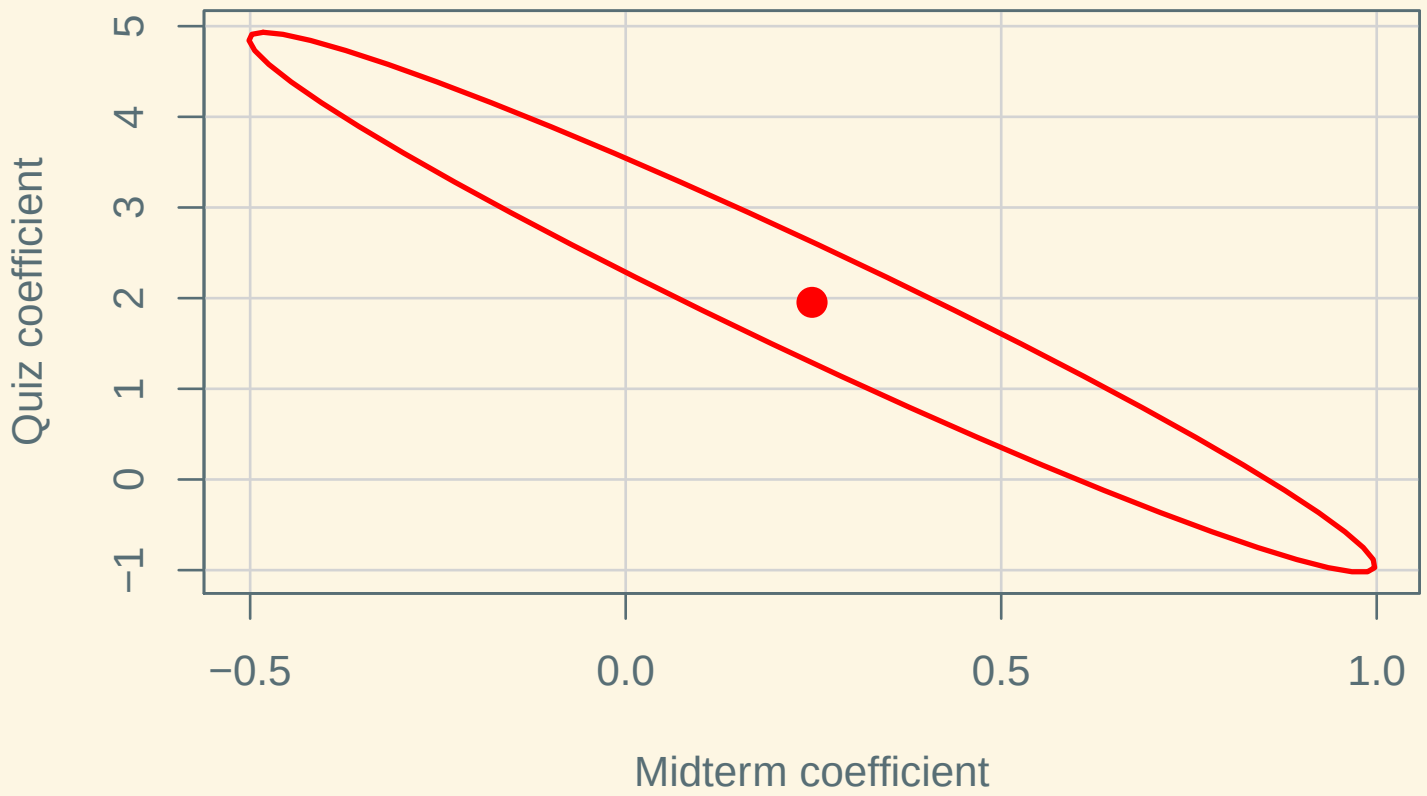
1. Remove redundant predictors
2. Combine predictors (e.g., taking a weighted average)
3. Use the multicollinear model anyway, just don't pay attention to tests/intervals for individual coefficients.

Removing Redundant Predictors

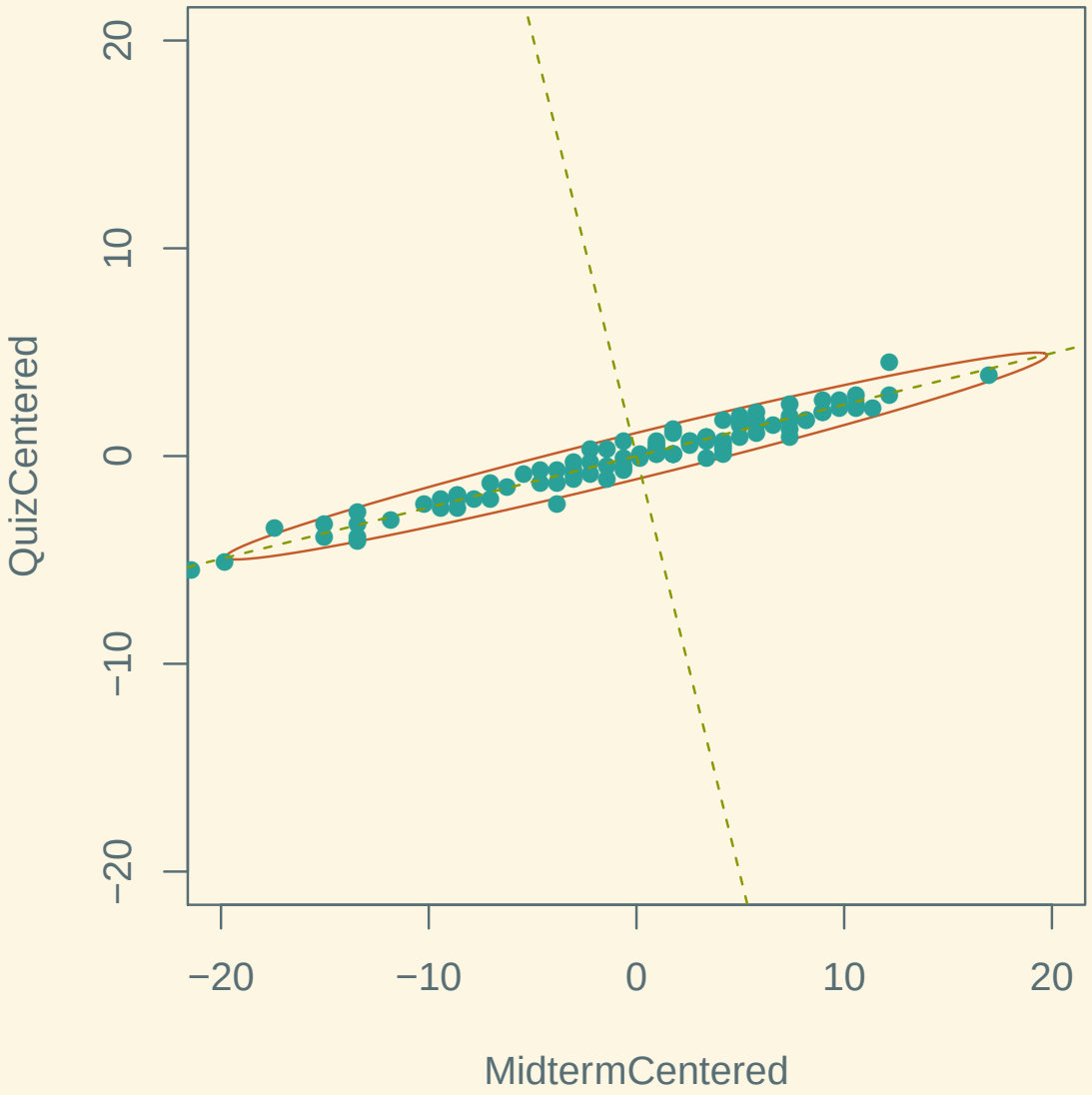
```
pulse.model2 <-
    lm(InvActive ~ InvRest + Exercise + Hgt + BMI, data = Pulse2)
pulse.model2 %>% summary() %>% coefficients() %>% round(4)

             Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.0006     0.0022  0.2522   0.8011
InvRest        0.6216     0.0647  9.6019   0.0000
Exercise       0.0002     0.0002  0.8689   0.3858
Hgt            0.0000     0.0000  1.4663   0.1440
BMI           -0.0001     0.0000 -2.3830   0.0180

pulse.model2 %>% vif() %>% round(2)

 InvRest Exercise      Hgt      BMI
    1.43     1.41     1.16     1.11
```

Bonus Material: Change of Coordinates

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



Bonus Material: Change of Coordinates



Bonus Material: Change of Coordinates

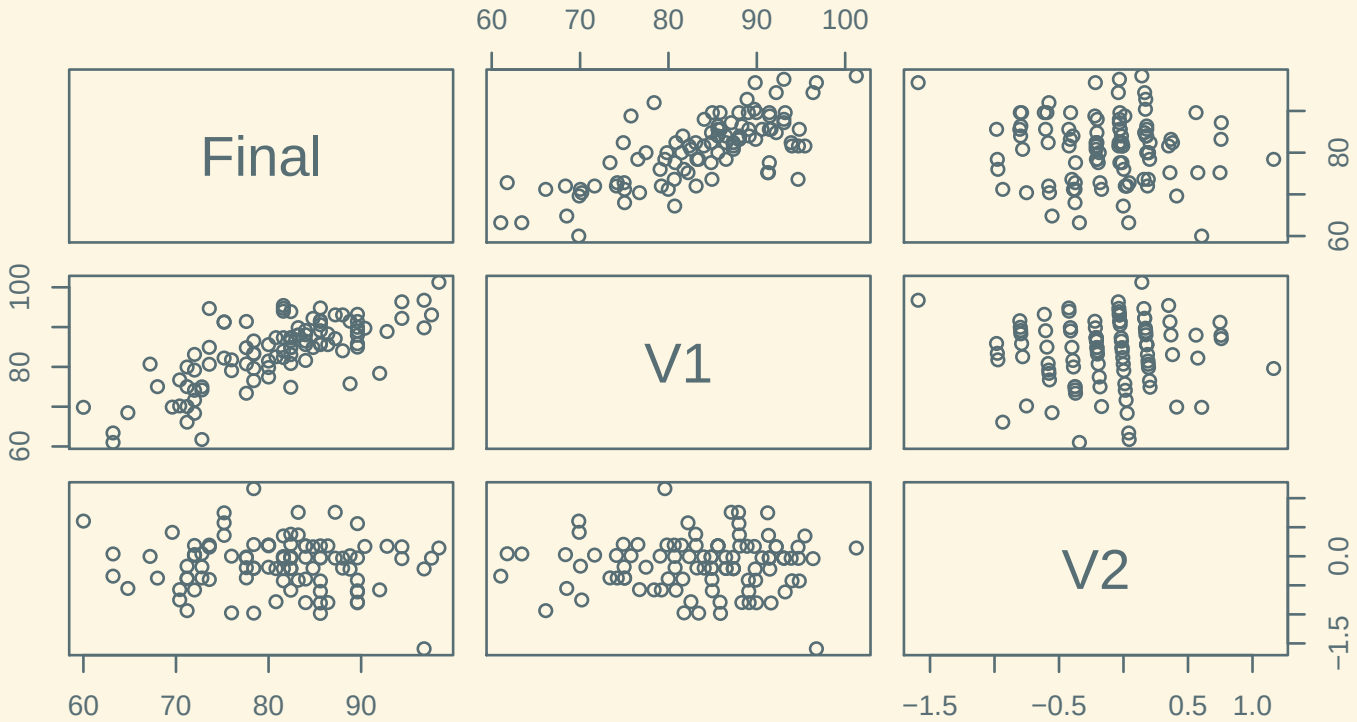
```
## Here I am pulling out the perpendicular directions in (Midterm,Quiz)
## space that align with the ellipse on the scatterplot.
## If you know some linear algebra:
## These are the eigenvectors of the covariance matrix
directions <- select(Scores, Midterm, Quiz) %>% cov() %>% eigen()
directions$vectors %>% round(digits = 2)
```

```
      [,1]  [,2]
[1,] -0.97  0.24
[2,] -0.24 -0.97
```

```
## Creating two new variables that are a weighted sum and weighted
## difference of the midterm and quiz score, with weights chosen so
## that the new variables are uncorrelated
Scores.augmented <-
    mutate(Scores,
           V1 = 0.97 * Midterm + 0.24 * Quiz,
           V2 = 0.24 * Midterm - 0.97 * Quiz)
```

Bonus Material: Change of Coordinates

```
select(Scores.augmented, Final, V1, V2) %>% plot()
```



Bonus Material: Change of Coordinates

```
select(Scores.augmented, Final, V1, V2) %>% cor() %>% round(digits = 2)
```

```
       Final   V1    V2
Final   1.00 0.73 -0.08
V1      0.73 1.00  0.02
V2     -0.08 0.02  1.00
```

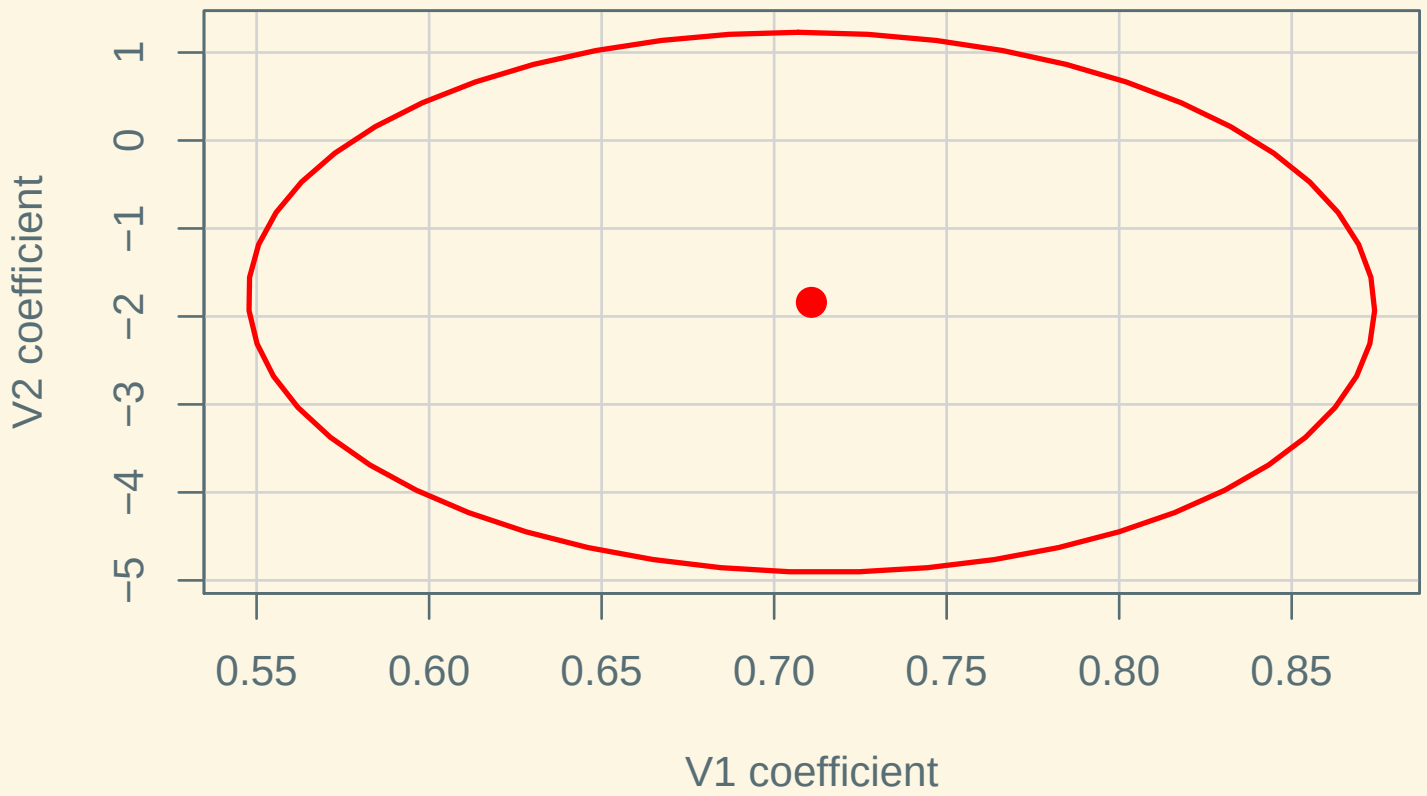
Bonus Material: Orthogonal Predictors

```
m.rotated <- lm(Final ~ V1 + V2, data = Scores.augmented)
m.rotated %>% summary() %>% coefficients() %>% round(3)

             Estimate Std. Error t value Pr(>|t|)
(Intercept)    21.085      5.539   3.807     0.00
V1              0.711      0.066  10.825     0.00
V2             -1.839      1.234  -1.490     0.14

m.rotated %>% rsquared()

[1] 0.5503497
```

The new model contains the same information as the old one, and makes the same predictions; but it has been **reparameterized**

Bonus Material: Orthogonal Predictors

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



Outline

Correlated Predictors

Diagnosis and Remediation

Collinearity and Simpson's Paradox

Collinearity and Simpson's Paradox

```
library("mosaic")
SATdata <- read.file("http://colindawson.net/data/SATS.csv")
head(SATdata)
```

```
           State Expenditure Ratio Salary Read Math Write Total PercentSAT
1    Alabama          10  15.3  49948  556  550   544  1650          8
2     Alaska          17  16.2  62654  518  515   491  1524         52
3    Arizona           9  21.4  49298  519  525   500  1544         28
4   Arkansas          10  14.1  49033  566  566   552  1684          5
5 California          10  24.1  71611  501  516   500  1517         53
6   Colorado          10  17.4  51660  568  572   555  1695         19
```

SATs and Teacher Salary

```
xyplot(Total ~ Salary, data = SATdata, type = c("p", "r"))
```

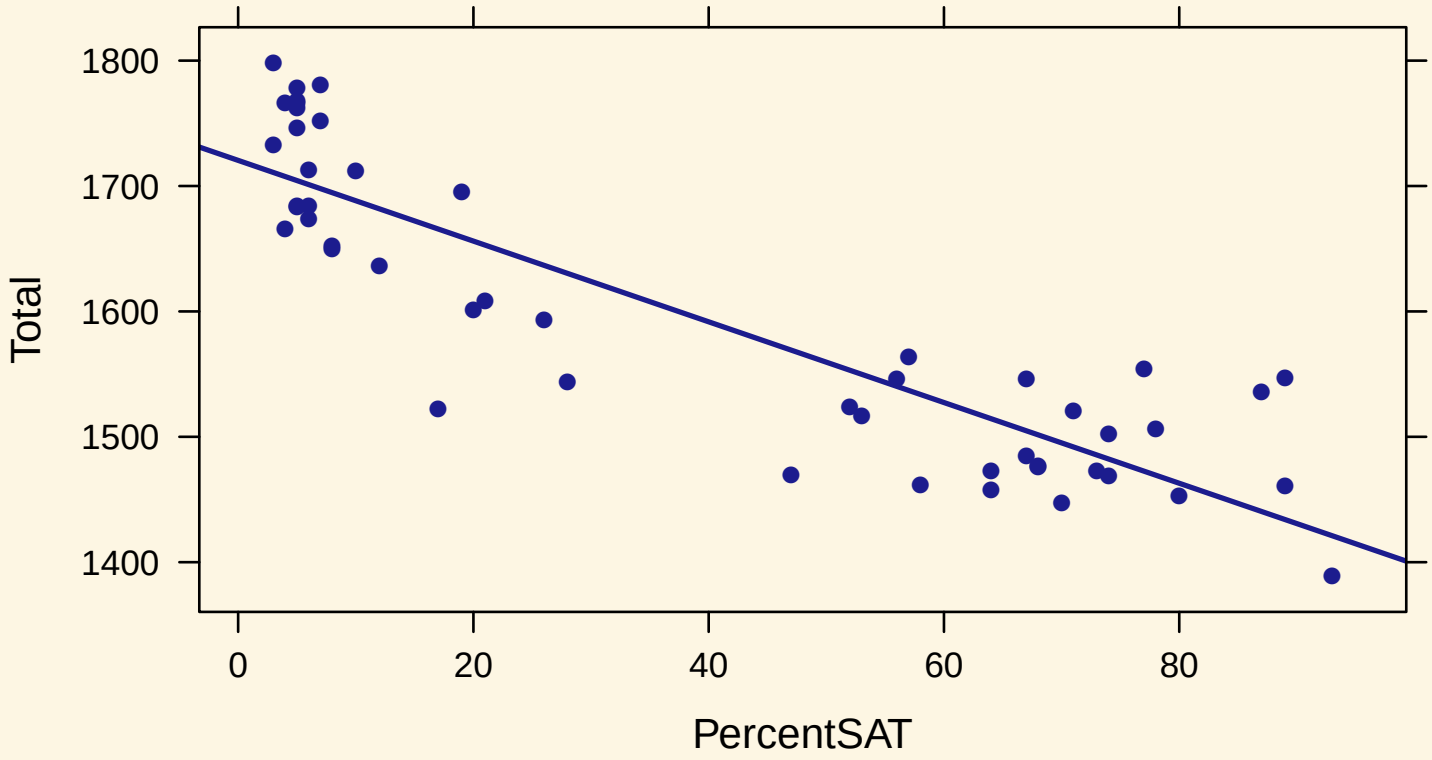
SATs and Teacher Salary

```
m.salary <- lm(Total ~ Salary, data = SATdata)
m.salary %>% summary() %>% coefficients() %>% round(3)

             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1871.104    113.141  16.538    0.000
Salary        -0.005      0.002  -2.451    0.018
```

Overall, states that pay teachers more have lower mean SAT scores.

SATs and Teacher Salary

```
xyplot(Total ~ PercentSAT, data = SATdata, type = c("p", "r"))
```



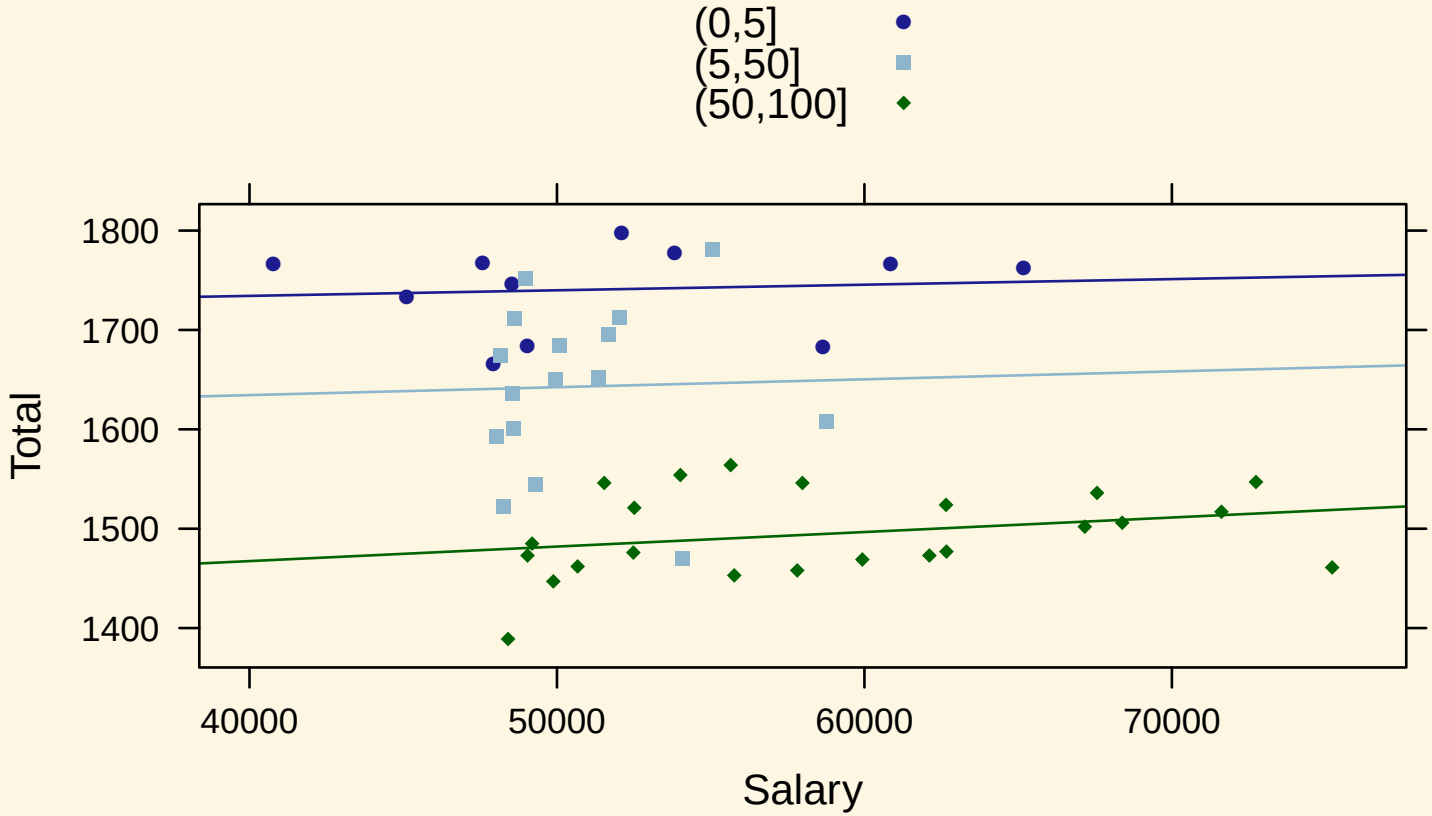
SATs and Participation Rate

```
m.percentSAT <- lm(Total ~ PercentSAT, data = SATdata)
m.percentSAT %>% summary() %>% coefficients() %>% round(3)

                Estimate Std. Error t value Pr(>|t|)
(Intercept)     1720.441     12.357 139.224        0
PercentSAT        -3.219      0.248 -12.989        0
```

Overall, states with higher participation rates have lower mean SAT scores.

Controlling for Participation

```
SATdata <-
    mutate(SATdata,
           ParticipationCategory = cut(PercentSAT, breaks = c(0,5,50,100)))
xyplot(Total ~ Salary, groups = ParticipationCategory, data = SATdata,
       type = c("p", "r"), auto.key = TRUE)
```



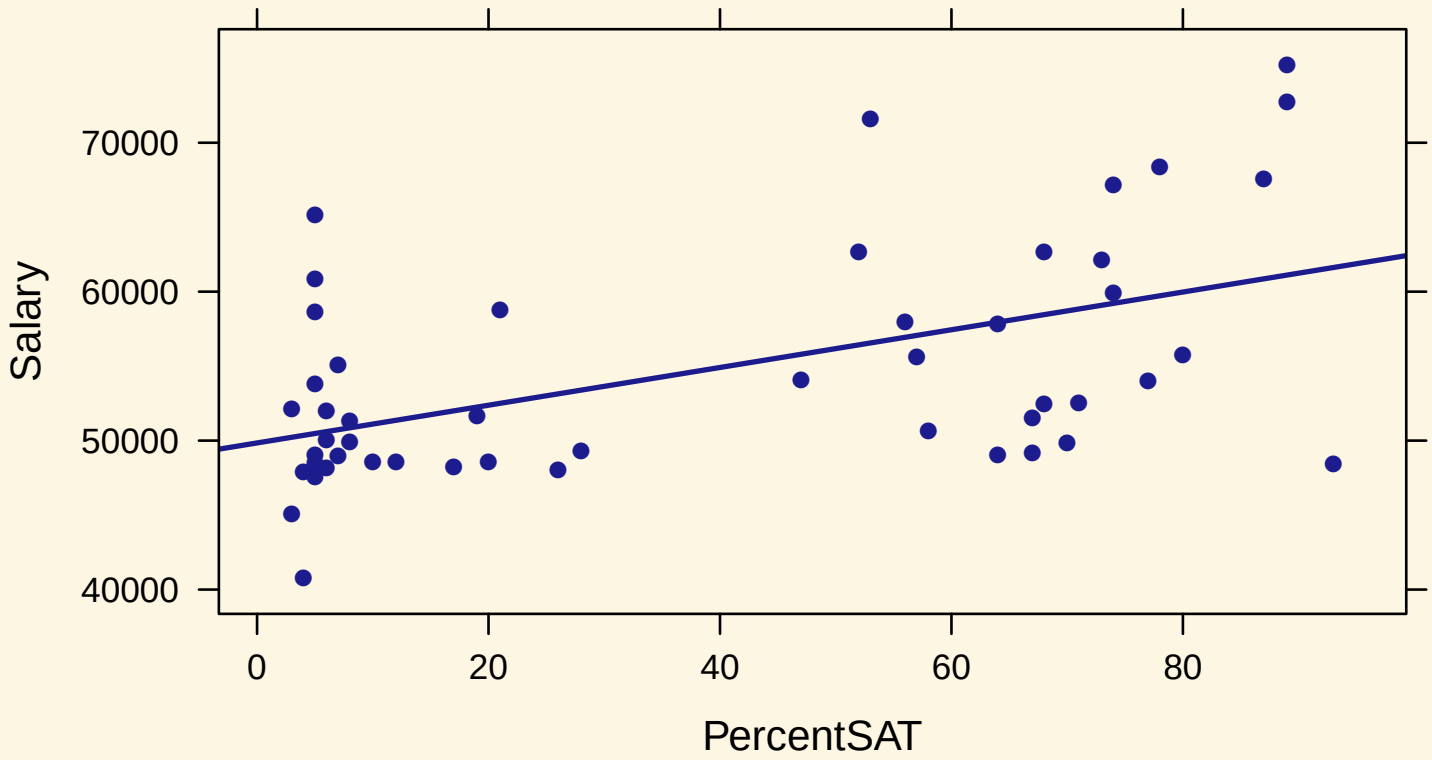
Controlling for Participation

```
m.salary.percent <- lm(Total ~ Salary + PercentSAT, data = SATdata)
m.salary.percent %>% summary() %>% coefficients() %>% round(3)

               Estimate Std. Error t value Pr(>|t|)
(Intercept) 1589.007     58.471  27.176    0.000
Salary         0.003      0.001   2.295    0.026
PercentSAT    -3.553      0.278 -12.756    0.000
```

In the full model the `Salary` coefficient has the opposite sign from the simple model! This is an instance of **Simpson's Paradox**

What's Going On?

```
xyplot(Salary ~ PercentSAT, data = SATdata, type = c("p", "r"))
```



PercentSAT was a **confounding variable**

Collinearity Isn't Always Bad

- Here is a case where, by adding a collinear predictor, we controlled for a confounding variable, removing (and in fact reversing!) a (likely) spurious association
- The difference: Here, we have evidence that we really need both predictors!