STAT 213
Multicollinearity

Colin Reimer Dawson

Oberlin College

July 12, 2021

Outline

Correlated Predictors

Diagnosis and Remediation

Simpson's Paradox

Added Variable Plots

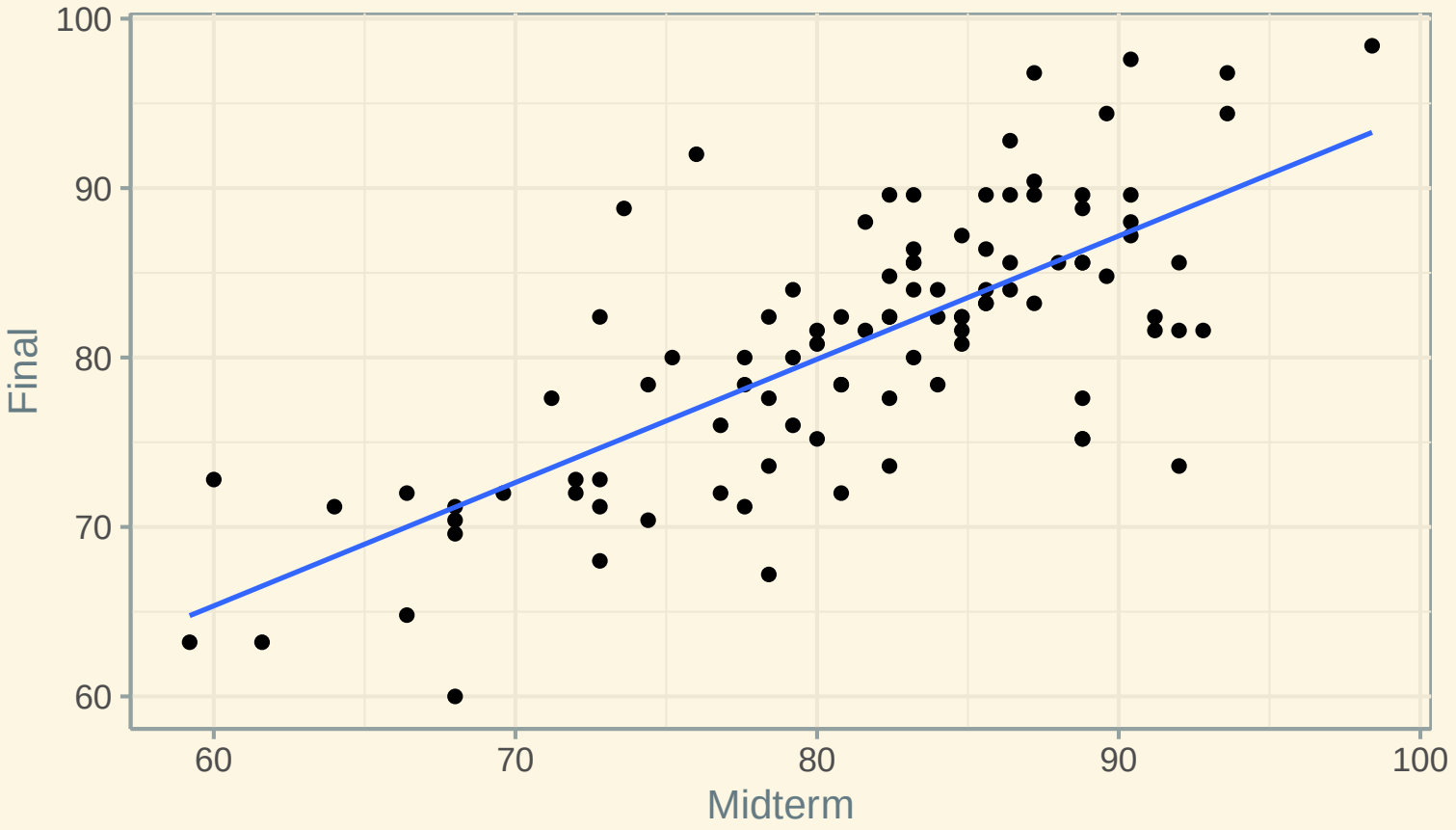
Predicting Final Exam Scores

A lazy stats instructor wants to leave early for vacation, and so they decide that rather than actually grade the final exam they will simply use a regression model to predict final exam scores using midterm and quiz scores. To estimate the coefficients they'll use the grades from last semester.[1]

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

Midterm and Final

```
gf_point(Final ~ Midterm, data = Scores) %>% gf_lm()
```



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
```

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

```
m_midterm %>% confint() %>% round(2)

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

For each additional point earned on the midterm, the predicted final exam score should go up by between 0.59 and 0.86 points.

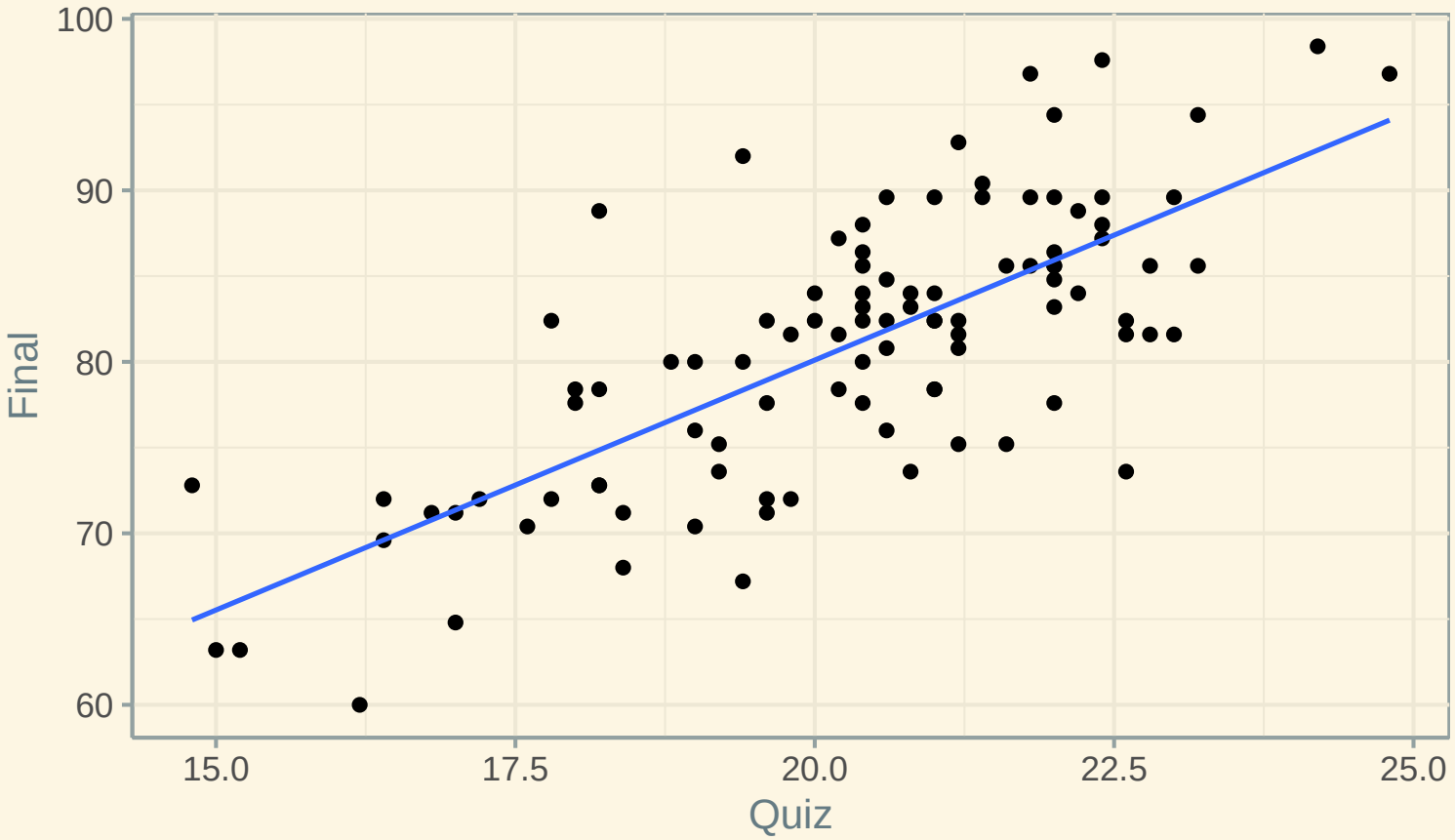
SLR Model: Midterm Only

```
m_midterm %>% rsquared() %>% round(2)

     [1] 0.54
```

In fact, 54% of the variability in the final exam scores is predictable using the midterm score alone!

Quiz and Final

```
gf_point(Final ~ Quiz, data = Scores) %>% gf_lm()
```



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
```

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

```
m_quiz %>% confint() %>% round(2)

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

For each additional point earned on the quiz, the predicted final exam score should go up between 2.38 and 3.45 points.

SLR Model: Quiz Only

```
m_quiz %>% rsquared() %>% round(2)

     [1] 0.55
```

In fact, 55% of the variability in final exam scores is predictable from quiz score

A Two Predictor Model

- 54% of the variability in final exam scores is predictable using the model that has `Midterm` as the sole predictor
- 55% of the variability in final exam scores is predictable using the model that has `Quiz` as the sole predictor
- **Question:** How much variability should we expect to be accounted for by a model that uses both `Midterm` and `Quiz` as predictors?

A Two Predictor Model

```
m_both <- lm(Final ~ Midterm + Quiz, data = Scores)
m_both %>% rsquared() %>% round(2)

     [1] 0.55
```

Apparently.... still 55%...

A Two Predictor Model

```
m_both %>% summary() %>% coef() %>% round(2)

                  Estimate Std. Error t value Pr(>|t|)
     (Intercept)    21.09       5.54    3.81     0.00
     Midterm         0.25       0.30    0.82     0.41
     Quiz            1.95       1.20    1.63     0.11
```

Neither predictor has a significant t-test in the two-predictor model!

```
m_both %>% confint() %>% round(2)

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

And the confidence intervals for the individual coefficients both include positive and negative values (and zero)...

What's Going On?

Outline

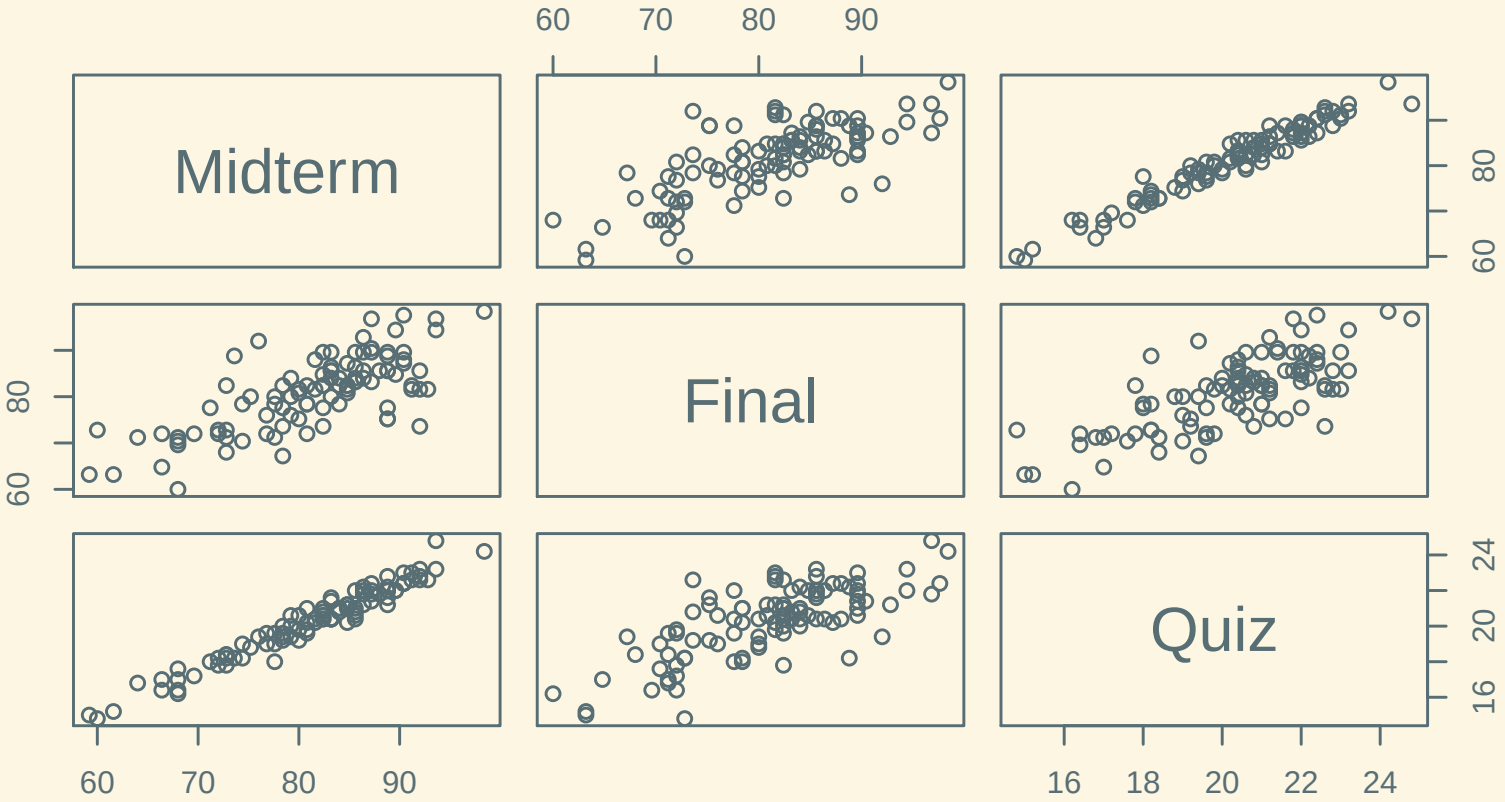
Correlated Predictors

Diagnosis and Remediation

Simpson's Paradox

Added Variable Plots

Correlated Predictors



Correlated Predictors

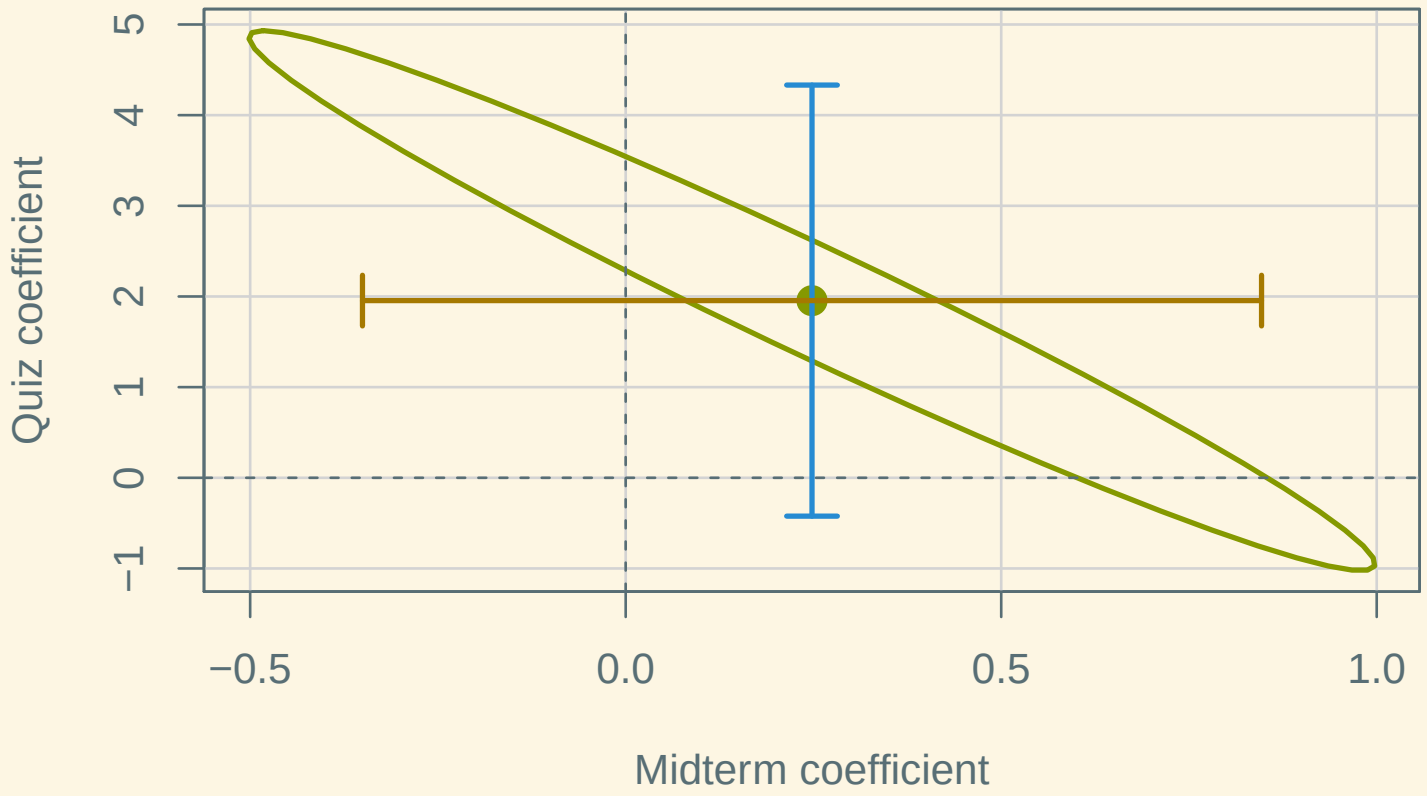
```
cor(Scores) %>% round(digits = 2)

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

Both `Midterm` and `Quiz` are correlated with `Final`, but they are *also* **strongly correlated with each other**!

Redundant Information

- Since `Midterm` and `Quiz` are both so highly predictable from the other (that is, they are **collinear**), we don't gain much **new information** 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



Since higher `Midterm` scores are strongly associated with higher `Quiz` scores, we obtain nearly the same predicted values for the actual data by increasing the `Midterm` coefficient and decreasing the `Quiz` coefficient — they offset each other

Outline

Correlated Predictors

Diagnosis and Remediation

Simpson's Paradox

Added Variable Plots

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:

$$\hat{X}_{ik} = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_{k-1} X_{i,k-1} + \beta_{k+1} X_{i,k+1} + \cdots + \beta_K X_{iK}$$

Rough rule: If the R^2 **from the above model** is > 0.80 (or so), tests and confidence intervals for individual coefficients are **unstable**

Equivalently: **VIF** (Variance Inflation Factor) $:= \frac{1}{1-R^2} > 5$ (or so), **Note:** each predictor has its own VIF, based on the R^2 to predict it from the others

Inter-Predictor R^2s

```
m_midterm_from_quiz <- lm(Midterm ~ Quiz, data = Scores)
m_midterm_from_quiz %>% rsquared() %>% round(3)

     [1] 0.95
```

95% of the variability in midterm scores is predictable from quiz scores

```
m_quiz_from_midterm <- lm(Quiz ~ Midterm, data = Scores)
m_quiz_from_midterm %>% rsquared() %>% round(3)

     [1] 0.95
```

95% of the variability in quiz scores is predictable from midterm scores

Variance Inflation Factor

```
library(car) # needed for vif()
vif(m_both) %>% round(1)

     Midterm    Quiz
        19.9    19.9
```

Since `Quiz` is **highly** predictable from `Midterm`, we have **very high** VIFs for both predictors

Note that $19.9 \approx \frac{1}{1-0.95}$ (differs only due to rounding in the R^2 values)

Can We Just Look at Correlations?

```
library(Stat2Data); data(Pulse)
Pulse2 <- Pulse %>%
    mutate(BMI = Wgt / Hgt^2 * 703)
```

A MLR Model

```
m_rest_hgt_bmi <-
    lm(Active ~ Rest + Hgt + Wgt + BMI, data = Pulse2)
m_rest_hgt_bmi %>% summary() %>% coefficients() %>% round(3)

                Estimate Std. Error t value Pr(>|t|)
     (Intercept)  249.020    130.396   1.910    0.057
     Rest           1.111      0.102  10.902    0.000
     Hgt           -3.623      1.874  -1.933    0.055
     Wgt            0.704      0.400   1.759    0.080
     BMI           -4.126      2.754  -1.498    0.135
```

None of `Height`, `Weight`, or `BMI` yield significant t-tests.

Pairwise R^2

```
correlations <- Pulse2 %>% select(Active, Rest, Hgt, Wgt, BMI) %>% cor()
correlations^2 %>% round(2)

            Active Rest  Hgt  Wgt  BMI
     Active   1.00 0.37 0.03 0.00 0.00
     Rest     0.37 1.00 0.06 0.03 0.01
     Hgt      0.03 0.06 1.00 0.57 0.10
     Wgt      0.00 0.03 0.57 1.00 0.73
     BMI      0.00 0.01 0.10 0.73 1.00
```

None of the squared correlations (R^2 to predict each predictor from each other) are above our "problematic" threshold for collinearity

Variance Inflation Factors

```
m_rest_hgt_bmi %>% vif()

            Rest        Hgt        Wgt        BMI
        1.073961  51.335068 169.792713  80.859987
```

However, the VIFs for all three predictors are off the charts...
Pairwise correlations don't tell the whole story!

Any two of `Height`, `Weight`, and `BMI` let us fill in the third one. They are **multicollinear**.

Multicollinearity: Remedies

If we find that some predictors suffer from high multicollinearity (guide: $\mathrm{VIF} > 5$), 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.

Outline

Correlated Predictors

Diagnosis and Remediation

Simpson's Paradox

Added Variable Plots

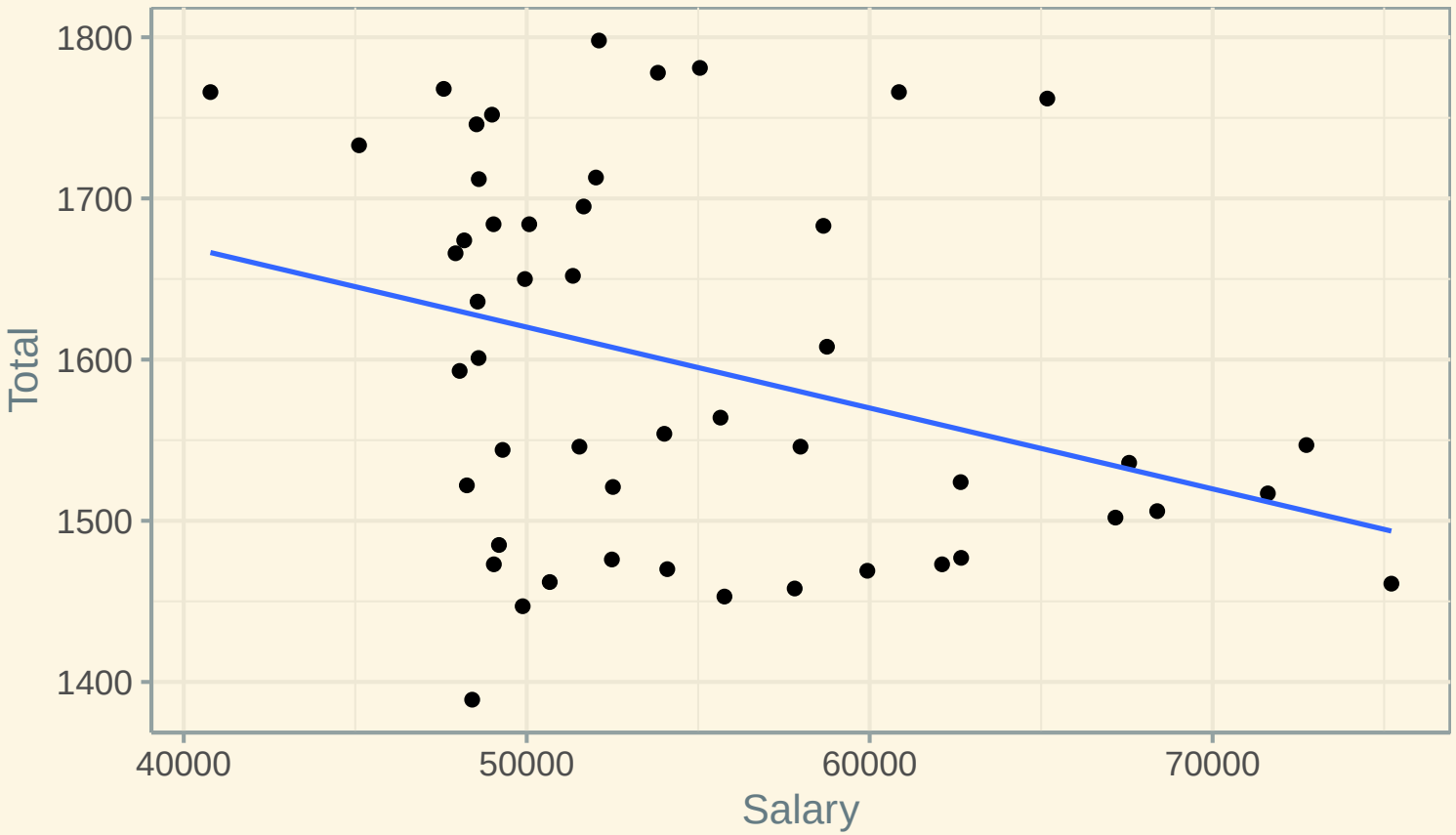
Collinearity and Simpson's Paradox

```
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 Median Teacher Salary

```
gf_point(Total ~ Salary, data = SATdata) %>% gf_lm()
```



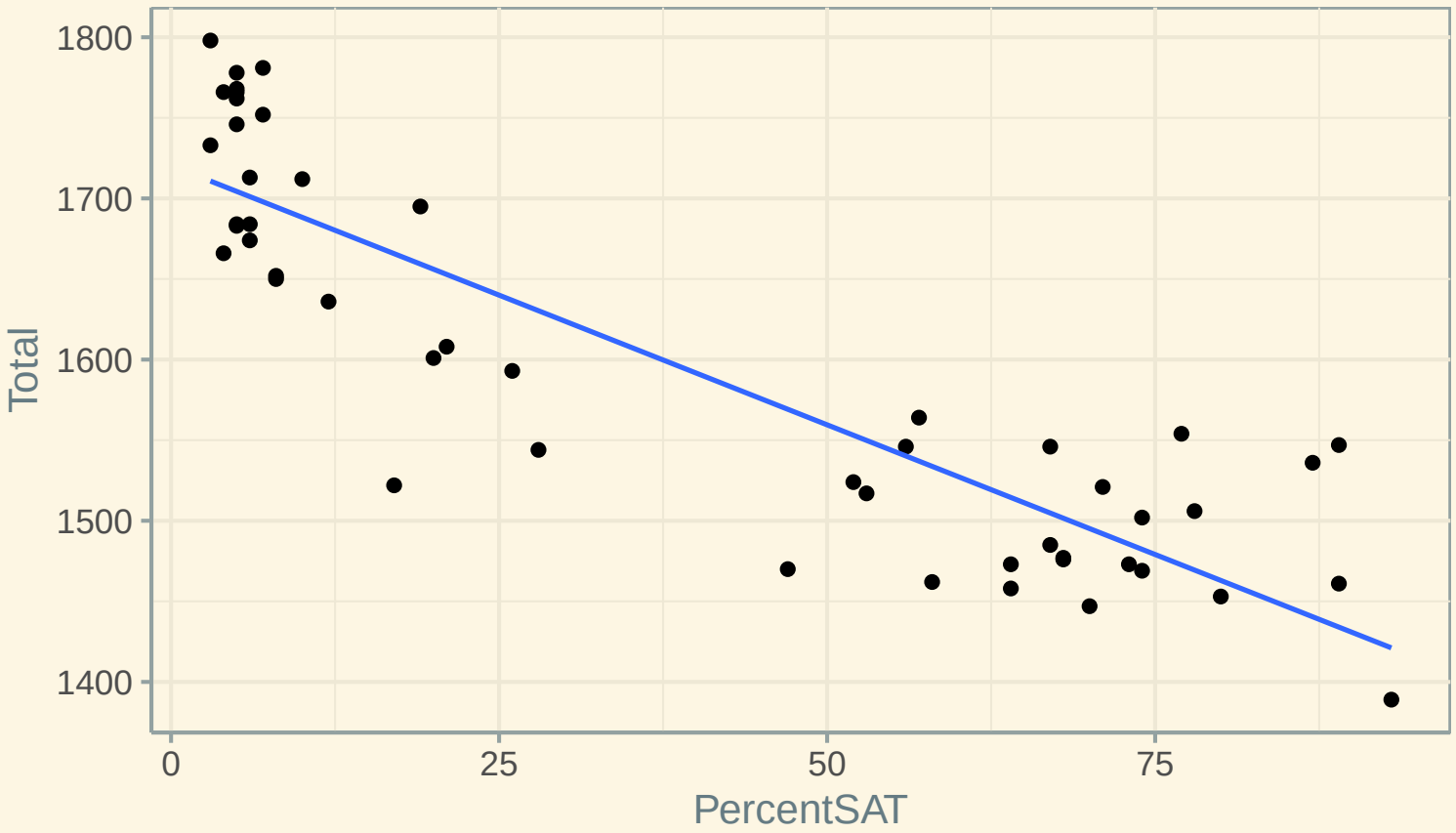
SATs and Teacher Salary

```
m_salary <- lm(Total ~ Salary, data = SATdata)
m_salary %>% summary() %>% coef() %>% 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

```
gf_point(Total ~ PercentSAT, data = SATdata) %>% gf_lm()
```



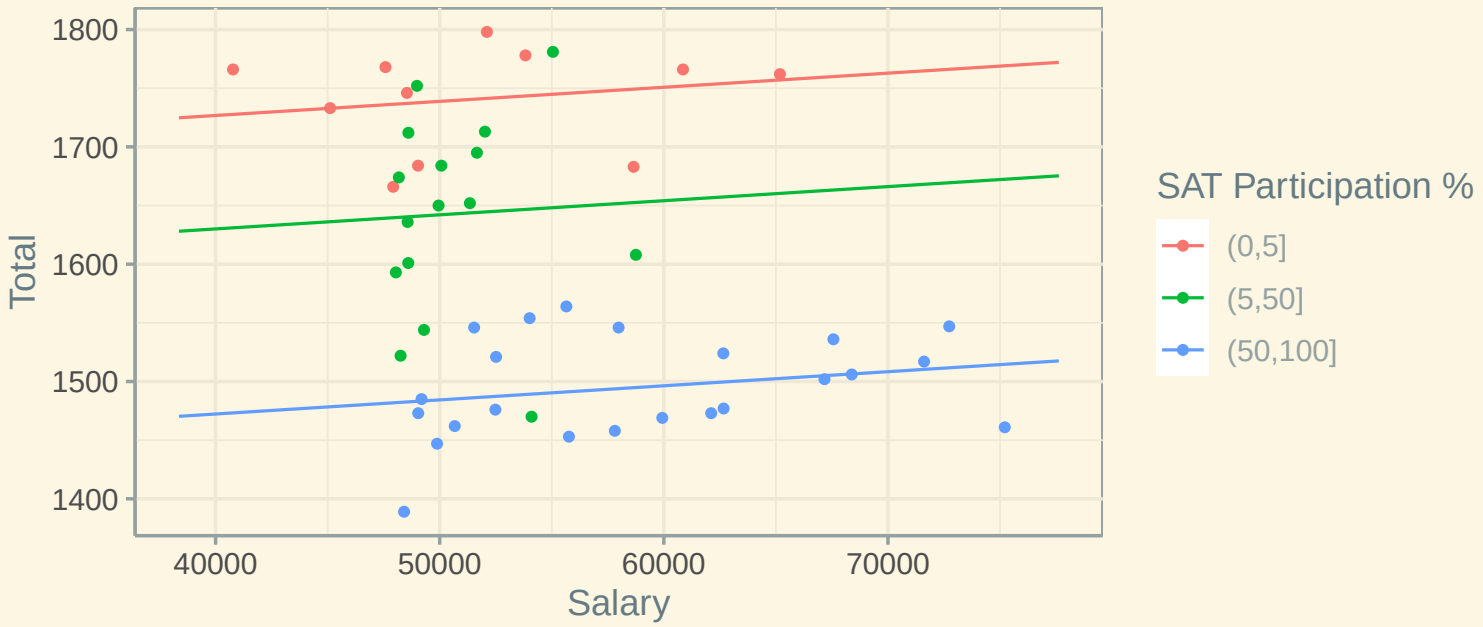
SATs and Participation Rate

```
m_participation <- lm(Total ~ PercentSAT, data = SATdata)
m_participation %>% 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 <- SATdata %>%
    mutate(
        ParticCategory = cut(PercentSAT, breaks = c(0,5,50,100)))
m_salary_particcat <- lm(Total ~ Salary + ParticCategory, data = SATdata)
plotModel(m_salary_particcat) +
    scale_color_discrete(name = "SAT Participation %")
```



Within each participation group, the more teachers are paid, the higher the mean score.

Controlling for Participation

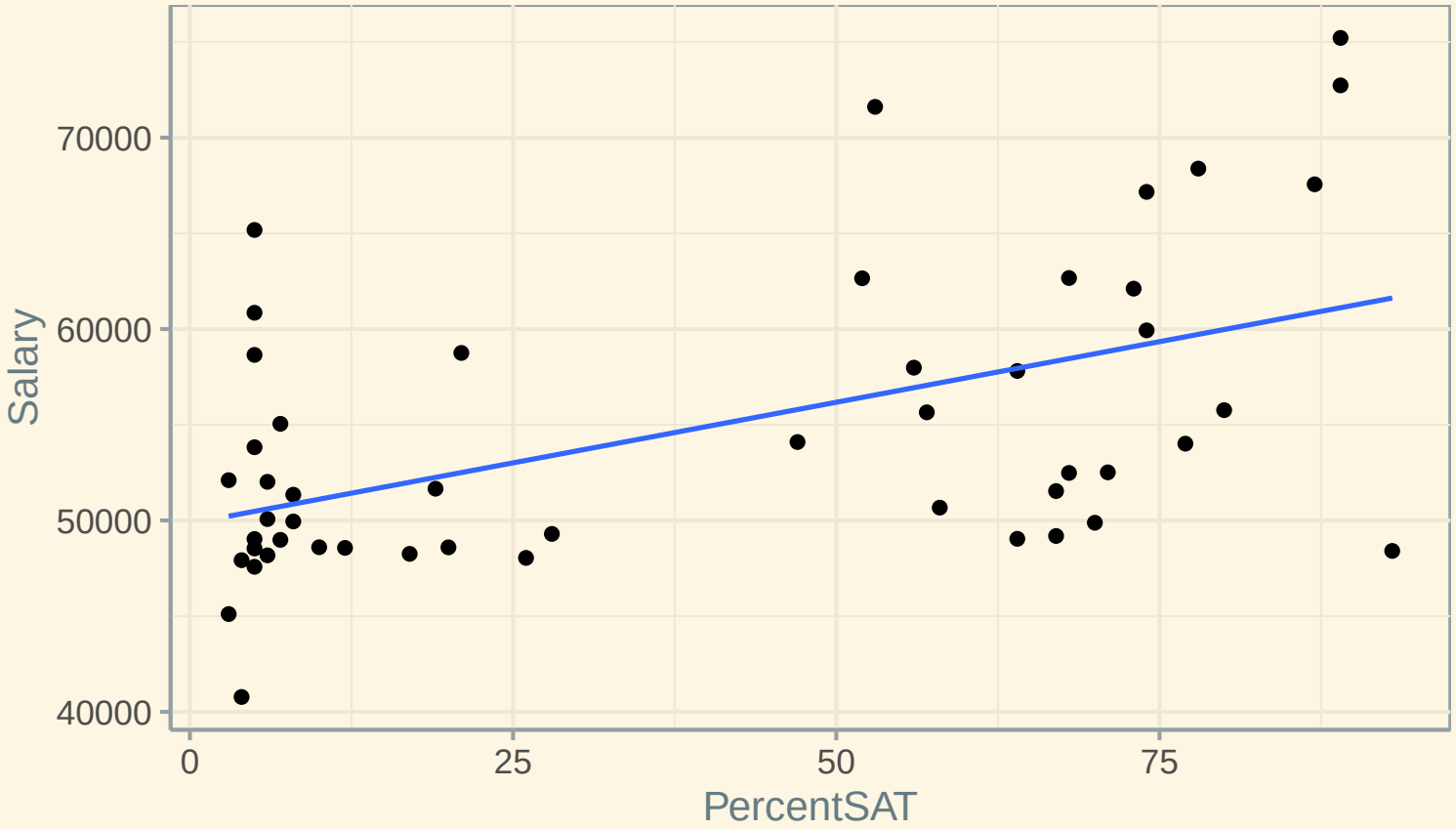
```
m_salary_partic <- lm(Total ~ Salary + PercentSAT, data = SATdata)
m_salary_partic %>% summary() %>% coefficients() %>% round(3)

                 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, and the evidence is significant that it is a useful and **positive** predictor when controlling for participation rate!

This reversal of direction when controlling for another variable is an instance of **Simpson's Paradox**

What's Going On?

```
gf_point(Salary ~ PercentSAT, data = SATdata) %>% gf_lm()
```



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!

Outline

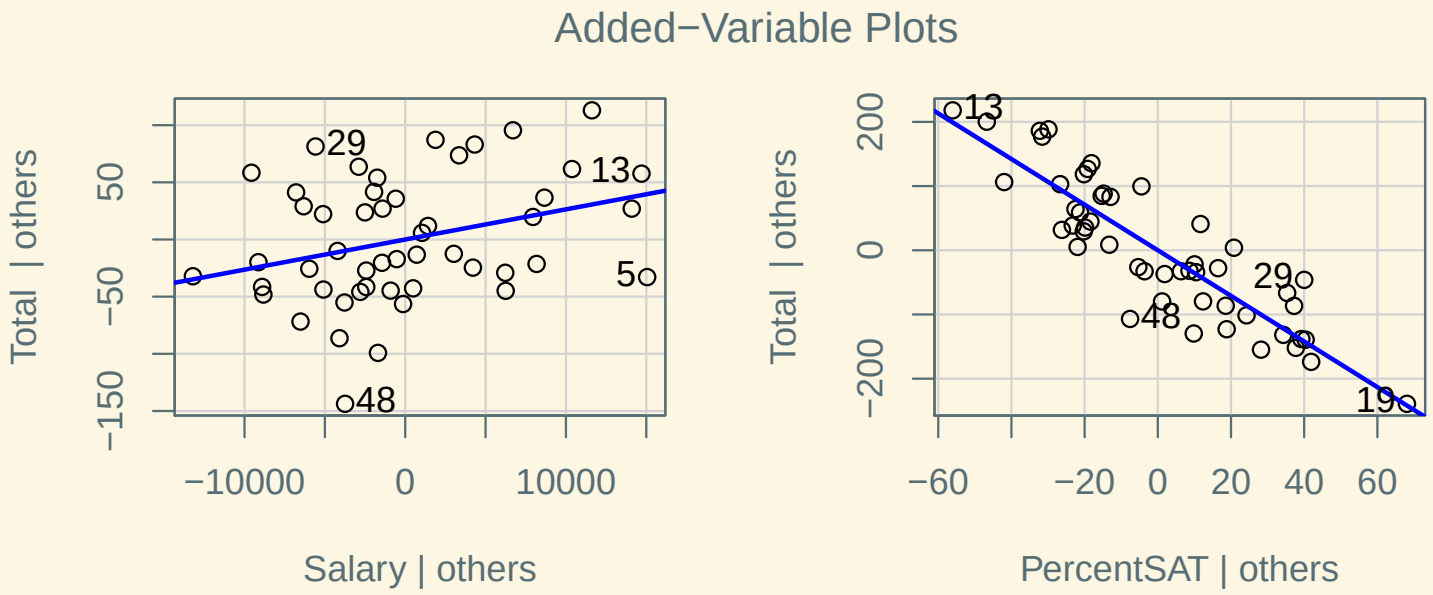
Correlated Predictors

Diagnosis and Remediation

Simpson's Paradox

Added Variable Plots

```
library(car) # same package that provides vif()
avPlots(m_salary_partic)
```



To see the unique contribution of a predictor, X_k, we can plot **two sets of residuals** against each other

1. Residuals from model predicting Y from other predictors (excluding X_k)
2. Residuals from model predicting X_k from other predictors

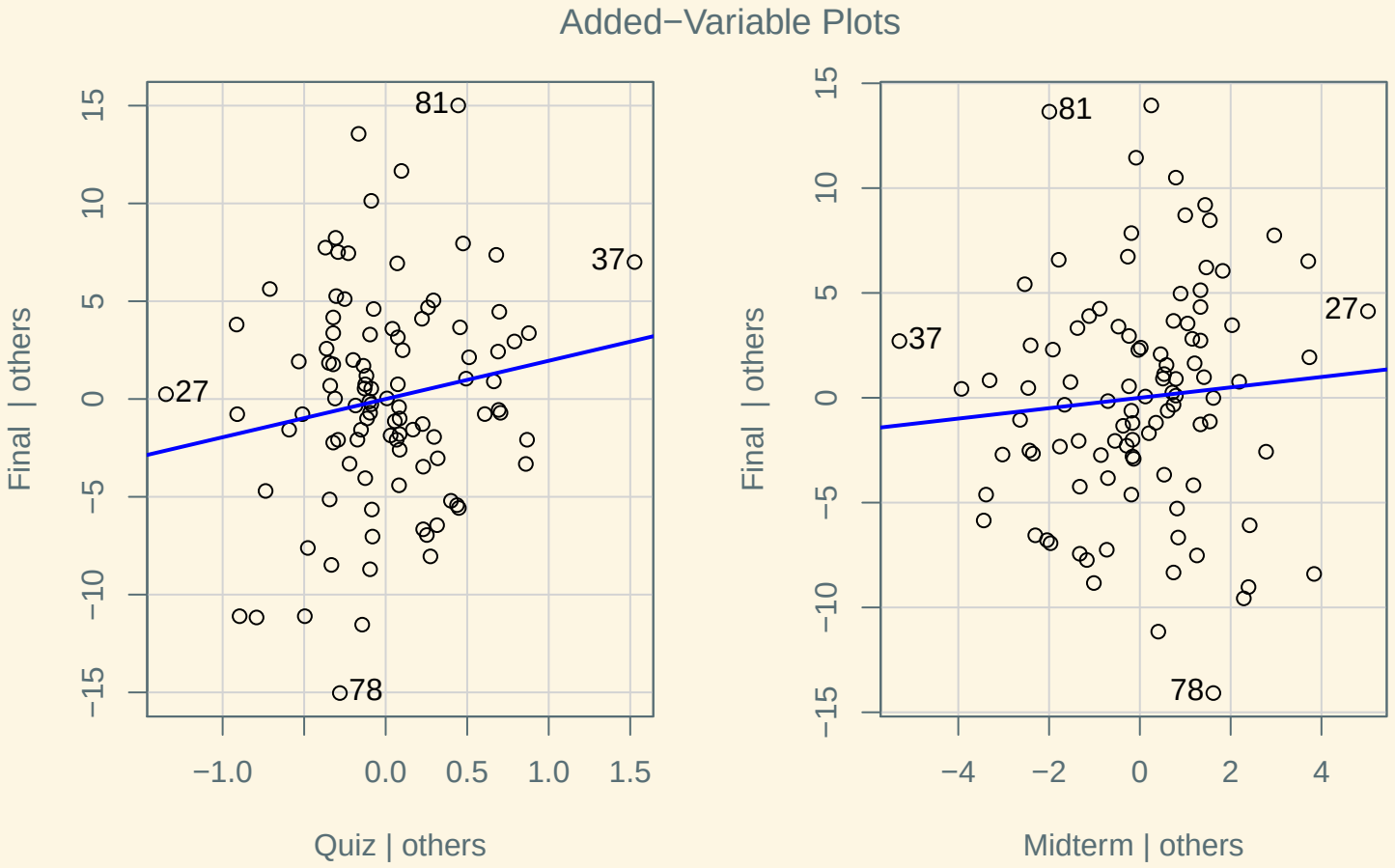
Added Variable Plots

- Also called "partial regression plots"
- Idea: Capture **nonredundant information** provided by X_k
- **How much information does X_k provide about Y that we couldn't have already predicted?**
- Slope of this line is the β for the predictor in question
- R^2 of this fit is the proportion of previously **unexplained** variability we can "soak up" by adding X_k to the model

$$R^2_{av} = \frac{R^2_{\text{full}} - R^2_{\text{reduced}}}{1 - R^2_{\text{reduced}}}$$

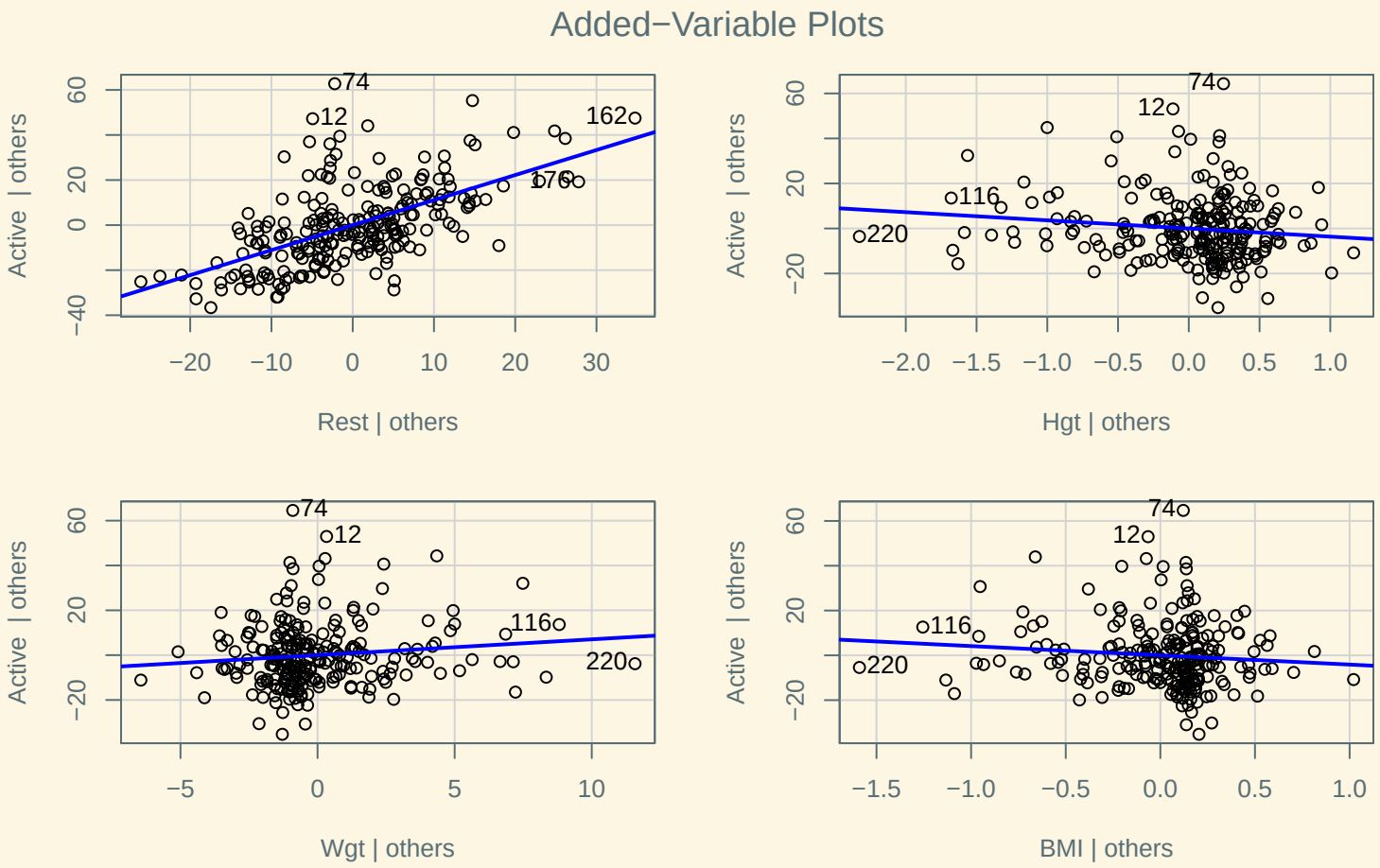
Example: Quiz and Exam Scores

```
m_both <- lm(Final ~ Quiz + Midterm, data = Scores)
avPlots(m_both, alpha = 0.3)
```



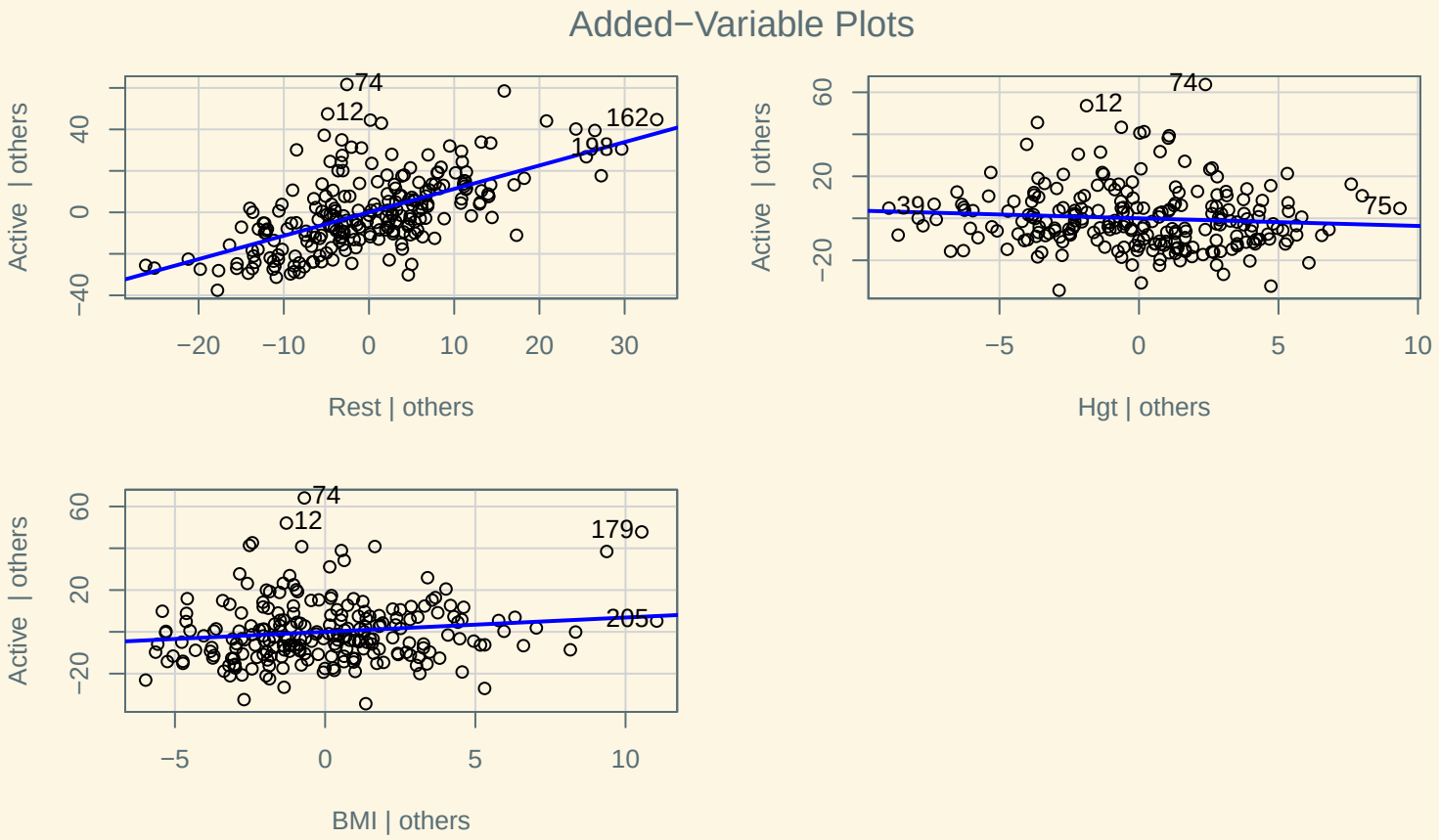
Example: Pulse Rates

```
m_rest_hgt_wgt_bmi <- lm(Active ~ Rest + Hgt + Wgt + BMI, data = Pulse2)
avPlots(m_rest_hgt_wgt_bmi)
```



Example: Pulse Rates (Removing Wgt)

```
m_rest_hgt_bmi <- lm(Active ~ Rest + Hgt + BMI, data = Pulse2)
avPlots(m_rest_hgt_bmi)
```



Note the differences in x axis range!