# Residuals and mean centering

Jeanette Mumford

10/13/2020

Hi! I decided to stick the code I mentioned into a markdown, so you can see the output without having to run the code. Some of this code is what I used to generate the images and examples for lecture and some was used to answer your questions.

## Mean centering

Just to review, there's nothing wrong with mean centering other than it is often done when not needed or folks think it is doing something that it isn't. Here I focus on the idea that mean centering regressors is important when you have interaction terms because it reduces collinearity. Does it reduce collinearity? Yes, it does! Is that collinearity a "problem"? No it isn't. Collinearity is only an issue if you're interpretation of that parameter is done without needing other parameters in the model. In this example, we cannot fully interpret an interaction effect without taking into account the main effects as well, so we're okay if the collinearity is only arising between main effects and the interaction effect. I will go into more detail below in the code. On the other hand, if I have a model with two regressors that are unrelated, theoretically, that I'm interpreting independently but are highly correlated, that's a collinearity problem we MUST fix... although fixing is often impossible. I'll include an example of that as well.

First, the interaction model, where the collinearity is not an issue in terms of interpretation, etc.

```
library(HH)  # This has a vif function in it
library(MASS)
library(viridis)
library(gtools)
library(ggplot2)

set.seed(12345)
x1 <- rnorm(100,20,2)
x2 <- rnorm(100,10,2)

y <- 1 + 2*x1 + 3*x2 + .5*x1*x2 + rnorm(100, sd = 3)

mod <- lm(y ~ x1*x2)
vif(mod)  #oh no!! (not really)  Goal is <5
```

```
##        x1        x2     x1:x2
##  32.92364 117.68148 162.30062
```

```
summary(mod)
```

```
##
## Call:
## lm(formula = y ~ x1 * x2)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -6.5408 -1.8346  0.1768  1.9949  7.0609
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.16030   15.11214   1.069   0.2876
## x1           1.34600    0.72565   1.855   0.0667 .
## x2           1.53098    1.51232   1.012   0.3139
## x1:x2        0.56237    0.07234   7.774 8.54e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.805 on 96 degrees of freedom
## Multiple R-squared:  0.9927, Adjusted R-squared:  0.9925
## F-statistic:  4377 on 3 and 96 DF,  p-value: < 2.2e-16
```

```
x1_mc <- x1 - mean(x1)
x2_mc <- x2 - mean(x2)

mod_mc <- lm(y ~ x1_mc*x2_mc)
vif(mod_mc) # all "fixed"
```

```
##       x1_mc       x2_mc x1_mc:x2_mc
##    1.026128    1.039259    1.039115
```

```
summary(mod_mc)  # but nothing really changes (at all)
```

```
##
## Call:
## lm(formula = y ~ x1_mc * x2_mc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5408 -1.8346  0.1768  1.9949  7.0609
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 175.46335    0.28255 621.000  < 2e-16 ***
## x1_mc         7.02060    0.12811  54.802  < 2e-16 ***
## x2_mc        13.05419    0.14212  91.854  < 2e-16 ***
## x1_mc:x2_mc   0.56237    0.07234   7.774 8.54e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.805 on 96 degrees of freedom
## Multiple R-squared:  0.9927, Adjusted R-squared:  0.9925
## F-statistic:  4377 on 3 and 96 DF,  p-value: < 2.2e-16
```

If you compare the model output above you can see the model fits are exactly the same and the only parameter for which you should be looking at the inference (the interaction) everything matches. Generally, you don't worry about significance of main effects or lower level terms if the higher level terms are significant. You still need them in your model whether or not they're significant.

So, why isn't collinearity an issue in this case? Because all of the covariates that are involved in the collinearity must be used together to properly interpret the model. By this I mean I cannot interpret the interaction *without* also looking at the main effects. For example, what if I want to illustrate my interaction by plotting

the relationship of y and x1 for the mean level of x2? Here's how I'd compute that.

```
# first I need the mean of x2
mn_x2 <-  mean(x2)
mn_x2
```

```
## [1] 10.09047
```

```
# I'll also plot +/- 1 SD from the mean
# be careful if you do this that +/- 1 SD is actually within the range of your
# regressor (x1)
# Sometimes, with skewed variables it can pop out of the range of your data.
mn_plus_sd_x2 <- mean(x2) + sd(x2)
mn_minus_sd_x2 <- mean(x2) - sd(x2)

# Then I need the coefficients, I'll start using the uncentered regressor model
mod_coeff <- mod$coeff
mod_coeff
```

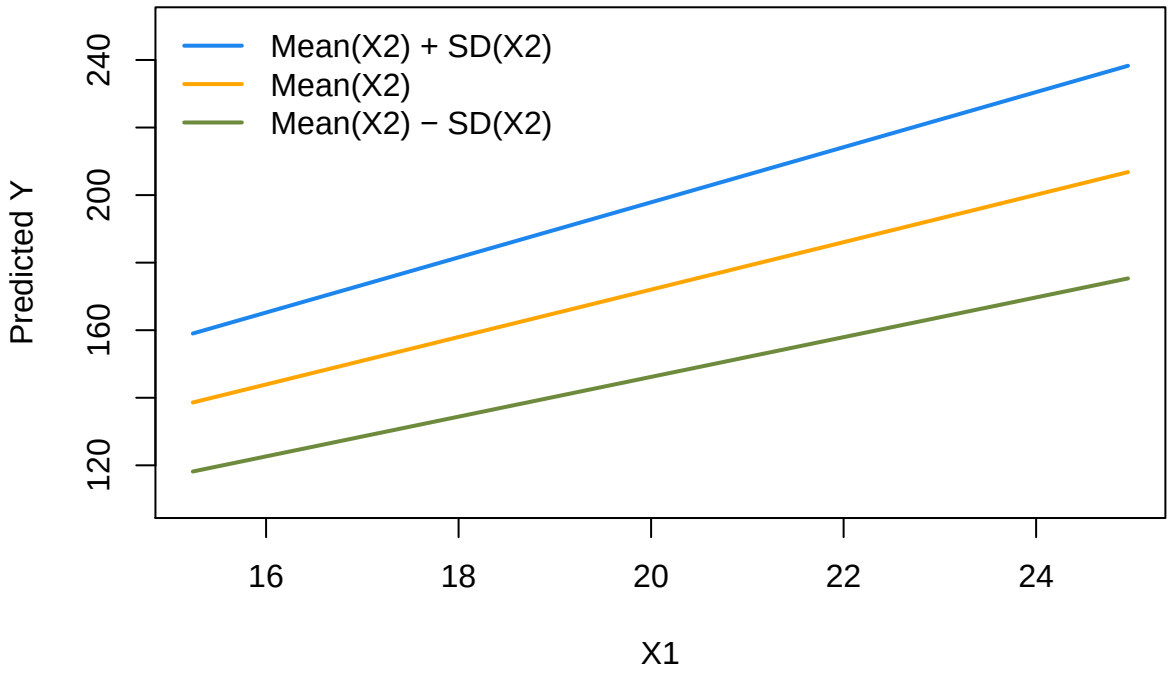
```
## (Intercept)          x1          x2       x1:x2
##  16.1603047   1.3460038   1.5309756   0.5623717
```

```
# Now I can generate a predicted y based on a range of x1 values and my mean
# x2 value for a plot.
# I'm just going to use a range of values from the min to max of x1.
# These are just estimated predictions based on our model fit.

x1_vals = seq(min(x1), max(x1), length = 50)
x1_vals
```

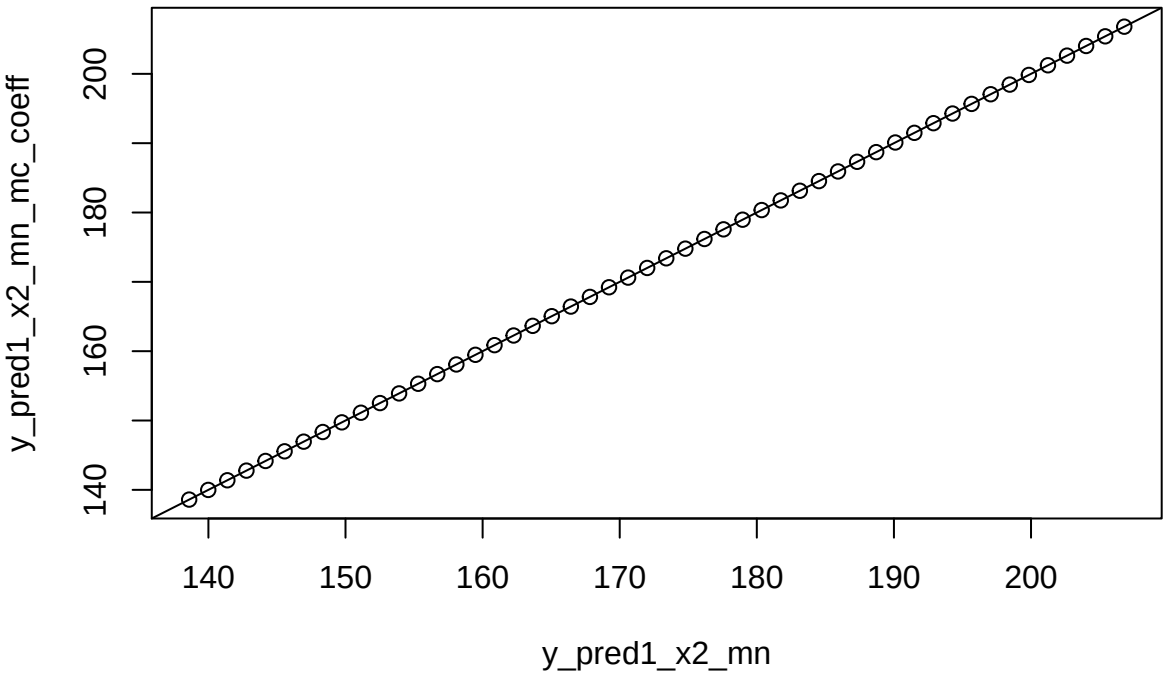
```
##  [1] 15.23928 15.43755 15.63581 15.83408 16.03234 16.23060 16.42887 16.62713
##  [9] 16.82540 17.02366 17.22192 17.42019 17.61845 17.81672 18.01498 18.21324
## [17] 18.41151 18.60977 18.80804 19.00630 19.20456 19.40283 19.60109 19.79936
## [25] 19.99762 20.19588 20.39415 20.59241 20.79068 20.98894 21.18721 21.38547
## [33] 21.58373 21.78200 21.98026 22.17853 22.37679 22.57505 22.77332 22.97158
## [41] 23.16985 23.36811 23.56637 23.76464 23.96290 24.16117 24.35943 24.55769
## [49] 24.75596 24.95422
```

```
# be careful when doing this that you are using the coeff's in the proper order

y_pred1_x2_mn = mod_coeff[1] + mod_coeff[2]*x1_vals + mod_coeff[3]*mn_x2 +
  mod_coeff[4]*mn_x2*x1_vals
y_pred1_x2_mn_plus_sd = mod_coeff[1] + mod_coeff[2]*x1_vals +
  mod_coeff[3]*mn_plus_sd_x2 + mod_coeff[4]*mn_plus_sd_x2*x1_vals
y_pred1_x2_mn_minus_sd = mod_coeff[1] + mod_coeff[2]*x1_vals +
  mod_coeff[3]*mn_minus_sd_x2 + mod_coeff[4]*mn_minus_sd_x2*x1_vals

#plot
# Just using a simple plot, but ggplot is great too!

plot(x1_vals, y_pred1_x2_mn, type = 'l', col = 'orange', lwd = 2, xlab = "X1",
     ylab = "Predicted Y",
     ylim = c(110, 250))
lines(x1_vals, y_pred1_x2_mn_plus_sd, lwd = 2, col = 'dodgerblue2')
lines(x1_vals, y_pred1_x2_mn_minus_sd, lwd = 2, col = 'darkolivegreen4')
legend('topleft', c("Mean(X2) + SD(X2)", "Mean(X2)", "Mean(X2) - SD(X2)"),
       lwd = 2,
```

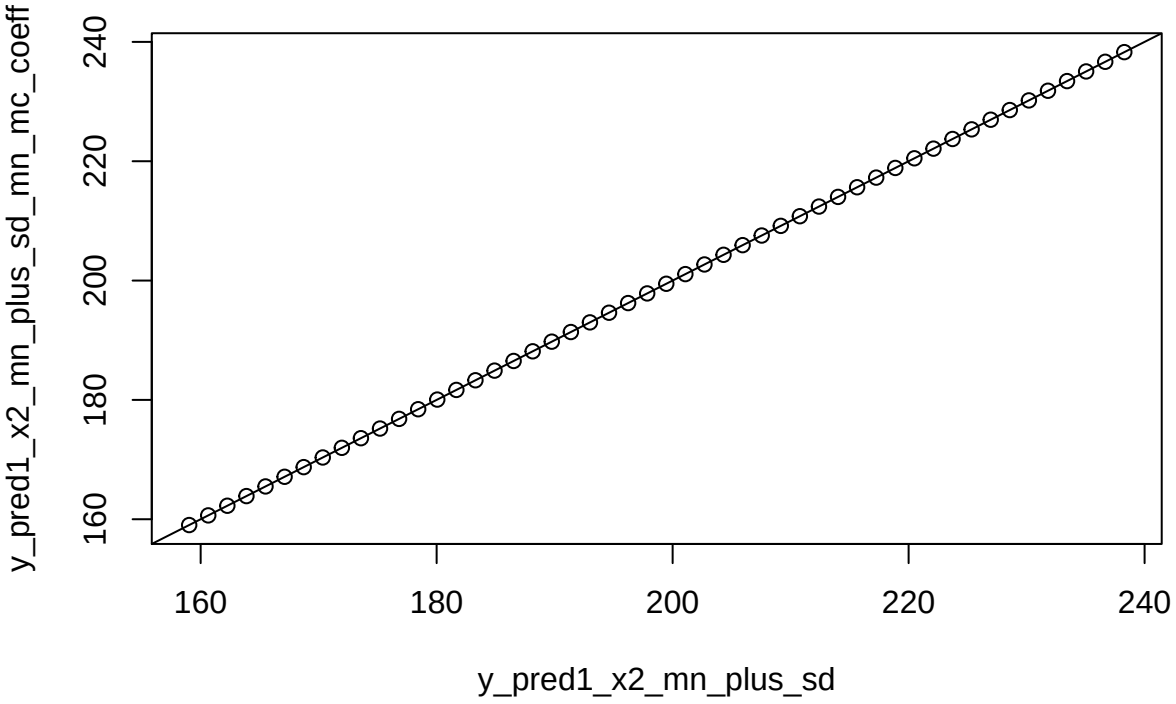
```
  col = c("dodgerblue2", "orange", "darkolivegreen4"), bty = 'n')
```




```
# If I use the coefficients from the second model it doesn't matter, I get the
# exact same values.  For simplicity, I'll illustrate with just the mean(x2)

#notably, x2 has a different meaning in this model, as does x1, so I must use
# 0 for the mean of x2 and my x1 range is different
x1_vals_dm = seq(min(x1_mc), max(x1_mc), length = 50)
mod_mc_coeff <- mod_mc$coeff
y_pred1_x2_mn_mc_coeff = mod_mc_coeff[1] + mod_mc_coeff[2]*x1_vals_dm +
  mod_mc_coeff[3]*0 + mod_mc_coeff[4]*0*x1_vals_dm

# If the values match they'll fall on a line with a slope of 1 and
# intercept of 0
plot(y_pred1_x2_mn, y_pred1_x2_mn_mc_coeff)
abline(a=0, b = 1)
```




```
# the values match exactly!  You can show this for mean - sd too

x2_dm_mean_plus_sd = 0 + sd(x2_mc)

y_pred1_x2_mn_plus_sd_mn_mc_coeff = mod_mc_coeff[1] +
  mod_mc_coeff[2]*x1_vals_dm + mod_mc_coeff[3]*x2_dm_mean_plus_sd +
  mod_mc_coeff[4]*x2_dm_mean_plus_sd*x1_vals_dm
plot(y_pred1_x2_mn_plus_sd, y_pred1_x2_mn_plus_sd_mn_mc_coeff)
abline(a=0, b = 1)
```



```
#So predictions match for mean + sd as well.  Hopefully that convinces you that
# it doesn't matter if I mean center or not.
```

On the other hand, what if I'm looking at the reaction time as a function of age and caffeine intake and I have an extremely strong correlation between age and caffeine? These are simulated data and different from the data I used in the slide for the example of adjusting covariates.

```
# Igore this, just making up some fake data that will have a collinearity issue!
mu_vec = c(82, 50, 1.5)
s_mat = matrix(c(30^2, -290, 6,
                 -290, 10^2, -3,
                 6, -3, 2.75^2),
               nrow = 3)
s_mat
```

```
##      [,1] [,2]    [,3]
## [1,]  900 -290  6.0000
## [2,] -290  100 -3.0000
## [3,]    6   -3  7.5625
```

```
cov2cor(s_mat)
```

```
##             [,1]       [,2]        [,3]
## [1,]  1.00000000 -0.9666667  0.07272727
## [2,] -0.96666667  1.0000000 -0.10909091
## [3,]  0.07272727 -0.1090909  1.00000000
```

```
set.seed(2010)
dat = data.frame(mvrnorm(n=200, mu = mu_vec, s_mat))
names(dat) = c("caffeine", "age", "rt")

mod = lm(rt ~ caffeine + age, dat)
vif(mod)  # the VIF is over 5, not good
```

```
## caffeine      age
## 15.50268 15.50268
```

```
summary(mod)
```

```
##
## Call:
## lm(formula = rt ~ caffeine + age, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5917 -1.4058 -0.0596  1.9658  6.5138
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19.55955    5.70812   3.427 0.000744 ***
## caffeine    -0.07112    0.02508  -2.835 0.005060 **
## age         -0.24688    0.07391  -3.340 0.001001 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.636 on 197 degrees of freedom
## Multiple R-squared:  0.06452,    Adjusted R-squared:  0.05502
```

```
## F-statistic: 6.794 on 2 and 197 DF,  p-value: 0.001402
```

In the above you'll see the VIF is quite high. In this case we're interested in interpreting each parameter estimate, alone. In other words, to fully interpret caffeine, I don't need to combine the caffeine parameter with any other parameter in the model. Compare this to the interaction model, where I had to use the main effects along with the interaction effect to describe what was happening. In that case, the variables involved in the collinearity are all combined to understand the effect. In this case we study each one, separately, albeit it is adjusted for the other variables in the model. In this case we should not proceed with this model. There's no fix for collinearity in this model due to the high correlation between caffeine and age. They're basically doing the same thing in the model and don't have enough unique variability to tell their own story. An anaolgy I often use is gathering information from eyewitnesses in an accident. If they all are telling the same story, you don't need multiple eyewitnesses, just one. Here you'd need to cut your loses and report that age and caffeine were highly correlated and so you have no way to tease apart each variable's unique contribution.

In the case of fMRI models with collinearity, sometimes we can reparamterize and fix it, but it is typically an issue that should be carefully thought through when designing the experiment.

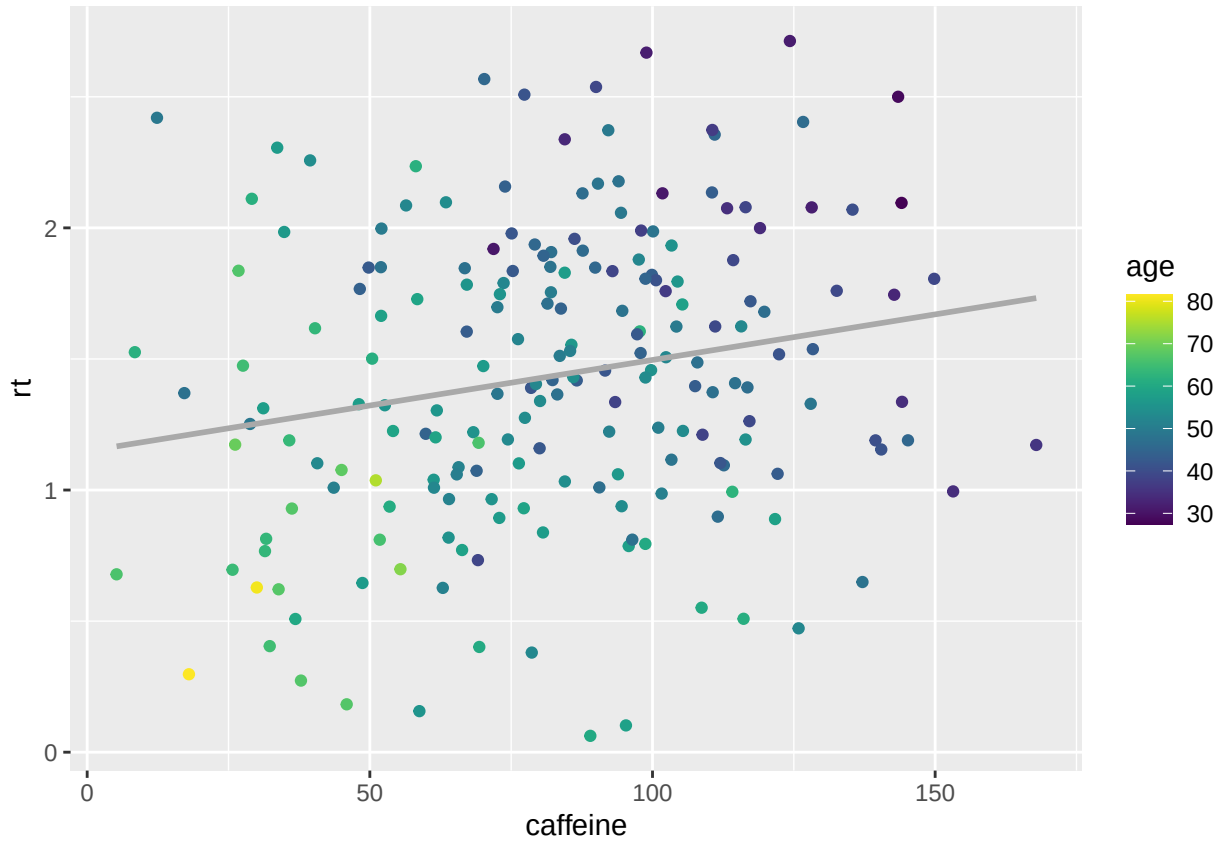
If interested, see my paper on orthogonalization. I have a video on it as well on MumfordBrainStats. Folks often think this fixes collinearity in fMRI but leads to HUGE misinterpretations. I've seen plenty. In fact I was asked to write this paper so a friend would have a reference to point to when reviewing papers making this error :)

## Adjusted regression

This is the exact code I used to make the residual plots. Since I set a seed for this one you'll get the exact plots I used in my slides. I'm including this for those who were interested to see how I derived the residuals for the plot.

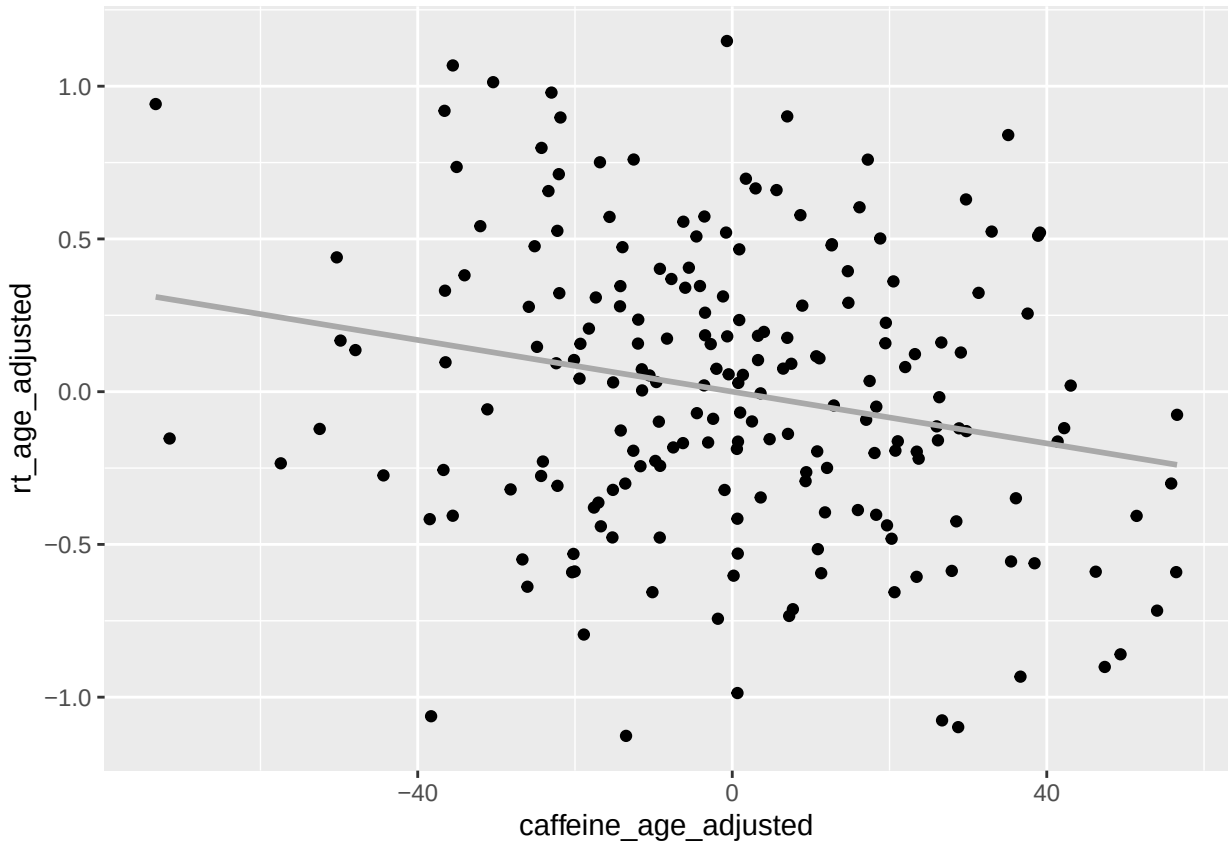
```
#ignore, I'm just making up some fake data...
mu_vec = c(82, 50, 1.5)
s_mat = matrix(c(30^2, -200, 5,
                 -200, 10^2, -3,
                 5, -3, .5^2),
               nrow = 3)
set.seed(1980)
dat = data.frame(mvrnorm(n=200, mu = mu_vec, s_mat))
names(dat) = c("caffeine", "age", "rt")


ggplot(dat, aes(x = caffeine, y = rt, color = age)) +
  geom_point()+ scale_color_viridis() +
  geom_smooth(method = lm, se = FALSE, col = 'darkgray')
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# Remove age effect, use the residual.  Note the intercept is in these
# models as well.
dat$rt_age_adjusted = lm(rt ~ age, dat)$resid
dat$caffeine_age_adjusted = lm(caffeine ~ age, dat)$resid
ggplot(dat, aes(x = caffeine_age_adjusted, y = rt_age_adjusted)) +
  geom_point()+
  geom_smooth(method = lm, se = FALSE, col = 'darkgray')
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# For funsies, you can see the parameter estimate using the residuals is
# exactly the same as the original model
# the only reason the se and p-value are off is because this model doesn't
# realize I already spent 2DF estimating the mean and age effects!

#unadjusted model
summary(lm(rt ~ caffeine, dat))
```

```
##
## Call:
## lm(formula = rt ~ caffeine, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.39558 -0.40661 -0.01608  0.40545  1.22847
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.148583   0.106948  10.740  < 2e-16 ***
## caffeine    0.003477   0.001204   2.887  0.00433 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5472 on 198 degrees of freedom
## Multiple R-squared:  0.04039,    Adjusted R-squared:  0.03554
## F-statistic: 8.333 on 1 and 198 DF,  p-value: 0.004325
```

```r
#original model
summary(lm(rt ~ caffeine + age, dat))
```

```
##
## Call:
## lm(formula = rt ~ caffeine + age, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.22452 -0.34426  0.00372  0.30714  1.14506
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.777672   0.301570  12.527  < 2e-16 ***
## caffeine    -0.004231   0.001318  -3.211  0.00154 **
## age         -0.039475   0.004322  -9.133  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4598 on 197 degrees of freedom
## Multiple R-squared:  0.3258, Adjusted R-squared:  0.319
## F-statistic: 47.61 on 2 and 197 DF,  p-value: < 2.2e-16
```

```r
# residual model
summary(lm(rt_age_adjusted ~ caffeine_age_adjusted, dat))
```

```
##
## Call:
## lm(formula = rt_age_adjusted ~ caffeine_age_adjusted, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.22452 -0.34426  0.00372  0.30714  1.14506
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             8.423e-18  3.243e-02   0.000   1.0000
## caffeine_age_adjusted -4.231e-03  1.314e-03  -3.219   0.0015 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4587 on 198 degrees of freedom
## Multiple R-squared:  0.04974,    Adjusted R-squared:  0.04494
## F-statistic: 10.36 on 1 and 198 DF,  p-value: 0.001503
```