Lab 3: Multiple Regression

PSC 103B - Winter 2024

```
library(ggplot2)
```

```
Warning: package 'ggplot2' was built under R version 4.3.2
```

Read in Data

```
reading <-
 ↪  read.csv("https://raw.githubusercontent.com/marwincarmo/phd/main/24-Winter/psc103b-wq24/
```

This data was generated based on information provided in: Weigel et al. (2006). Contributions of the home literacy environment to preschool-aged children's emerging literacy and language skills.

The dataset contains the following variables:

- `ParentChildAct`: A composite measure of how often parents spend time with reading-related activities to their children, such as reading aloud to them. This was measured in minutes per day.
- `ChildAge`: The child's age (in months)
- 5 different measures of literacy and language skills: Print knowledge, reading interest, emergent writing, expressive language, and receptive language.
- `OverallLiteracy`: A composite measure created by averaging the 5 measures of literacy/language. We're going to think of this as the number of points on some test (to help our interpretations).

Simple linear regression recap

Regression examines the relation between an outcome variable and a set of q predictor variables. In simple linear regression there is only one predictor:

$$y_i = b_0 + b_1 x_{1_i} + \epsilon_i$$

where, y_i is the outcome variable for individual $i = 1, 2, \ldots, n$, x_i is the vector of the single predictor variable, b_0 represents the intercept, b_1 is the slope, and ϵ_i is the vector of residuals for each individual – it represents the difference between the observed value of y_i and the value predicted by the model.

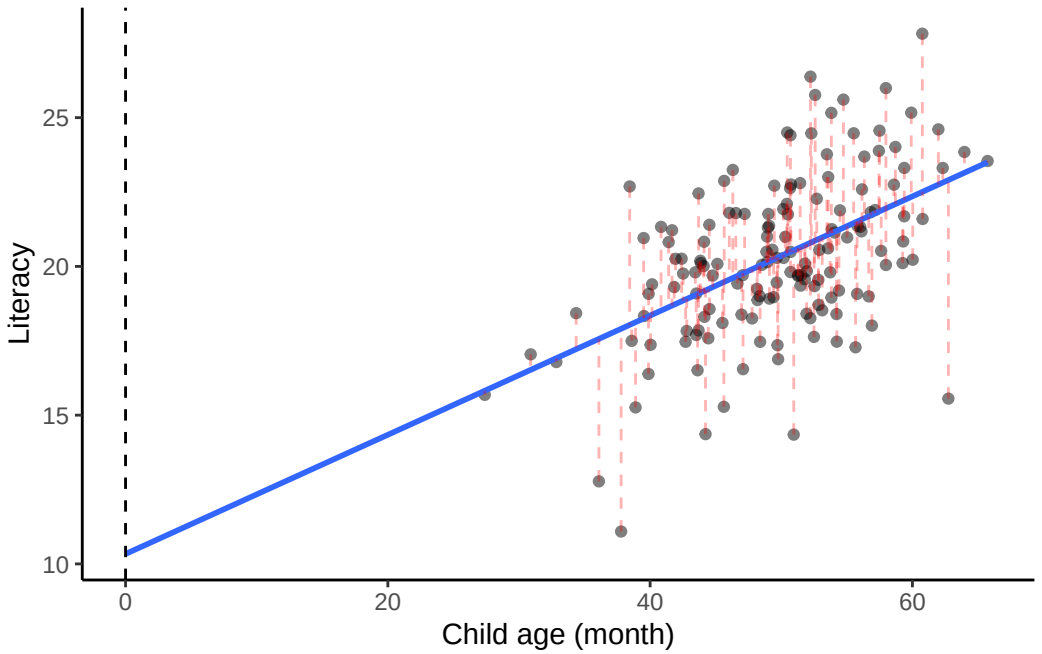


Figure 1: Scatterplot of Overall Literacy and Child age. The blue line represents the linear regression slope and the red dotted lines, the residual for each observation.

In R we use the `lm()` function to estimate a regression model. To estimate a model predicting Literacy from child age, we would write our model as:

```
literacy_age <- lm(OverallLiteracy ~ ChildAge, data = reading)
summary(literacy_age)
```

```
Call:
lm(formula = OverallLiteracy ~ ChildAge, data = reading)

Residuals:
    Min      1Q  Median      3Q     Max
-7.3463 -1.3542 -0.0035  1.5401  5.5831

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.33375    1.40062   7.378 1.07e-11 ***
ChildAge     0.20033    0.02792   7.175 3.23e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.342 on 148 degrees of freedom
Multiple R-squared:  0.2581,    Adjusted R-squared:  0.2531
F-statistic: 51.48 on 1 and 148 DF,  p-value: 3.226e-11
```

This output tells us that when a child is 0 months old, their expected literacy score is 10.33. The overall literacy score is expected to increase by 0.20 points for each increase in one month of age. Here is how we could write our results in a paper:

We fitted a linear regression model to predict Overall Literacy with Child Age. The model statistically significantly explains 26% of the variance in Overall Literacy ($R^2 = 0.26$, $F(1, 148) = 51.48$, $p < .001$). The effect of Child Age is statistically significant and positive ($b = 0.20$, 95% CI [0.15, 0.26], $t(148) = 7.18$, $p < .001$).

Confidence intervals for the estimates

To exemplify how we obtain confidence intervals in R, let's build a new regression model, this time predicting Overall Literacy from how often parents spend time with reading-related activities to their children (`ParentChildAct`):

```r
literacy_parent <-  lm(OverallLiteracy ~ ParentChildAct, data = reading)
summary(literacy_parent)
```

```
Call:
lm(formula = OverallLiteracy ~ ParentChildAct, data = reading)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-7.6483 -1.7220  0.0194  1.7186  6.8192

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     13.2682     1.6681   7.954 4.29e-13 ***
ParentChildAct   0.2096     0.0494   4.243 3.88e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.568 on 148 degrees of freedom
Multiple R-squared:  0.1084,    Adjusted R-squared:  0.1024
F-statistic:    18 on 1 and 148 DF,  p-value: 3.877e-05
```

We can obtain the 95% CI using the `confint()` function:

```
confint(object = literacy_parent, level = 0.95)
```

```
                   2.5 %     97.5 %
(Intercept)    9.9718817 16.5645063
ParentChildAct 0.1119778  0.3072325
```

That means that if we were to collect another sample size of 150, measure Parent-child activity and a Literacy score, and estimate b_1 repeatedly, 95% of the intervals would cover the true value – and 5% would miss it. Here's how we could write up our results:

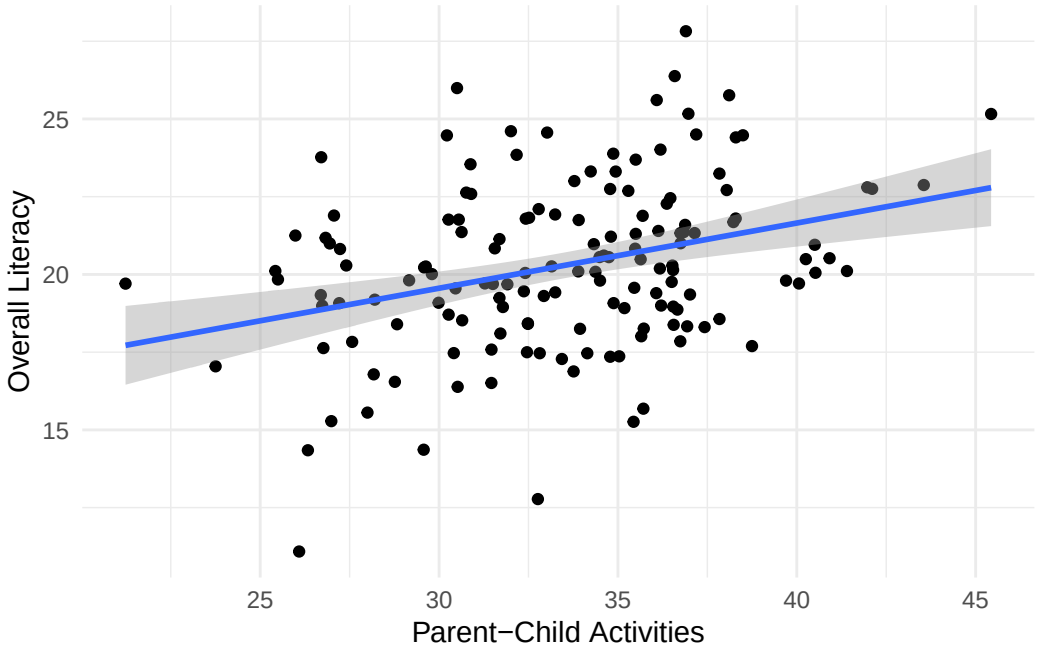
We ran a simple linear regression to determine whether Parent-Child activity predicts child reading literacy. The effect of Parent and Child activity is statistically significant and positive ($b = 0.21$, 95% CI [0.11, 0.31], $t(148) = 4.24$, $p < .001$). Parent-child activity explained 11% of the total variability in child reading literacy.

Confidence Limits on Y

Calculating confidence limits on Y is analogous to calculating a confidence interval for each of the predicted values $\hat{Y}_i$. We can also call them **prediction intervals**.

Given our model, where would we expect to see future observations fall?

```
ggplot(data = reading, aes(y = OverallLiteracy, x = ParentChildAct)) +
  geom_point() +
  theme_minimal() +
  xlab('Parent-Child Activities') +
  ylab('Overall Literacy') +
  geom_smooth(method = 'lm', se = TRUE)
```



Let's say that we want to calculate the prediction interval for $\widehat{Literacy}_i$ for a child whose parents spend 20 minutes each day reading with them.

$$\hat{Y}_i = 13.27 + 0.21 \times 20$$

Step 1: calculate $\hat{Y}^*$

```
pred_literacy <- 13.27 + 0.21*20
pred_literacy
```

```
[1] 17.47
```

Step 2: calculate $s_{\hat{Y}^*}$

- s_ϵ:

```
sqrt( (1/(nrow(reading) - 2)) * sum((reading$OverallLiteracy -
 ↪  predict(literacy_parent))^2) )
```

```
[1] 2.56781
```

```
# OR
s_epsilon <- summary(literacy_parent)$sigma
```

- SS_X:

```
ss_x <- sum((reading$ParentChildAct - mean(reading$ParentChildAct))^2)
```

- $s_{\hat{Y}^*} = s_\epsilon \sqrt{1 + \frac{1}{N} + \frac{(X^* - \bar{X})^2}{SS_X}}$

```
s_y <- s_epsilon * sqrt( 1+(1/nrow(reading)) + (( (20 -
 ↪  mean(reading$OverallLiteracy))^2 ) / ss_x))
```

Step 3: Calculate the 95% Prediction Interval for $\hat{Y}^*$

```
t_crit <- qt(.025, nrow(reading) - 2, lower.tail = FALSE)
```

Lower confidence limit:

$\hat{Y}^* - t_{crit}(s_{\hat{Y}^*})$

```
pred_literacy - (t_crit*s_y)
```

```
[1] 12.37873
```

Upper confidence limit:

$\hat{Y}^* + t_{crit}(s_{\hat{Y}^*})$

```
pred_literacy + (t_crit*s_y)
```

```
[1] 22.56127
```

Instead of doing all these calculations by hand, we can use R's `predict()` function and ask for the prediction interval for the new value that we want.

```
predict(object = literacy_age, newdata = data.frame( ChildAge=36), level =
 ↪  0.95, interval = "predict")
```

```
       fit      lwr      upr
1 17.54548 12.84008 22.25088
```

Note that these values don't exactly match our manual calculations because we rounded the numbers entered in the formula.

We specify our `lm` object in the `object` argument. Because we want to find the predicted value for a new observation, we can pass our new data in the optional argument `newdata`. It is an optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used. By default, this function uses 0.95 as the confidence `level`, but we could have specified any other value we wanted. Lastly, we request the prediction interval by setting `interval = "predict"`. If we wished to the Confidence Interval, we could put this argument equal to `"confidence"`.

Multiple Regression

Recall the R^2 value we obtained in our first example predicting literacy from a child's age. The R^2 value of 0.26 means that the child's age explains 26% of the variation in literacy scores. But that means that there is still 74% that is **not** explained by the child's age, or in other words, one predictor was not sufficient for explaining a decent amount of the variability in our outcome. We need to find other variables that could help predict our outcome.

This is where multiple regression comes in: we want to predict or explain a single outcome variable using 2 or more predictors.

So if this is our simple linear regression model is

$$y_i = b_0 + b_1 x_{1_i} + \epsilon_i$$

Multiple regression just adds on more predictors so our model is now:

$$y_i = b_0 + b_1 x_{1_i} + b_2 x_{2_i} + ... + b_q x_{q_i} + \epsilon_i$$

In our case, let's add on just one more predictor of the amount of time parents spend doing reading-related activities with their child.

Before, our model was: $OverallLiteracy_i = b_0 + b_1 \times ChildAge_i + \epsilon_i$, and now our model is: $OverallLiteracy_i = b_0 + b_1 \times ChildAge_i + b_2 \times ParentChildAct_i + \epsilon_i$.

We just have to add the predictor to our model, and luckily, it's also pretty easy to add it in R.

```
literacy_multiple <- lm(OverallLiteracy ~ ChildAge + ParentChildAct, data =
 ↪  reading)
summary(literacy_multiple)
```

```
Call:
lm(formula = OverallLiteracy ~ ChildAge + ParentChildAct, data = reading)

Residuals:
    Min      1Q  Median      3Q     Max
-6.1942 -1.5785  0.1858  1.5431  4.9392

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     3.36632    1.90794   1.764   0.0797 .
ChildAge        0.19992    0.02591   7.717 1.68e-12 ***
ParentChildAct  0.20860    0.04182   4.988 1.70e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.174 on 147 degrees of freedom
Multiple R-squared:  0.3655,    Adjusted R-squared:  0.3568
F-statistic: 42.34 on 2 and 147 DF,  p-value: 3.018e-15
```

Now we have more terms: we have the intercept and the effect of age, like before and now we also have the slope associated with the time parents spend doing reading-related activities.

How can we interpret these terms? Like before, the intercept is the expected value of our outcome when *all* predictors are 0. So when a child is 0 months old, and a parent spends 0 minutes per day doing reading-related activities, the expected literacy score is 3.37.

What has changed is the interpretation of the slope. The slope is now the expected change in Y for a 1-unit increase in X, **holding the other variable constant**.

So the slope for age means that if we keep the amount of time on parent-child activities constant, then increasing the child's age by one month will increase the literacy score by 0.20.

Similarly, the slope for parent-child activities means that if we hold the child's age constant, then increasing the time on reading activities by 1 minute is expected to increase the child's score by 0.21 points.

Are these slopes significant? Yes, both slopes are significant, meaning that they each have a unique effect on the outcome variable. And we can get the confidence intervals for each of the terms in our model:

```
confint(literacy_multiple)
```

```
                    2.5 %    97.5 %
(Intercept)    -0.4042156 7.1368459
ChildAge        0.1487244 0.2511233
ParentChildAct  0.1259563 0.2912465
```

As a side note, this is why you might read research articles that talk about "controlling for" or "adjusting for" other variables like gender or SES. All they're doing is running a multiple regression model with gender or SES as another predictor. This is because multiple regression lets us examine the unique effect of each variable in our model, given all the other variables; therefore, if the slope is still significant, it means that there is something unique about our predictor that is predicting the outcome.

What is our R^2 now? Before, in simple regression, it was 26%. Now, it has increased to 37% But remember: every time you add predictors to a model, the R^2 will generally increase. This is why, in multiple regression, you report the adjusted R^2, which penalizes you for having too many predictors in your model.

Our adjusted R2 isn't too different – 36%. This means that 36% of the variation in literacy scores is explained by our predictors (notice that we can't separate out the percentage due to each predictor, but just an overall value).

Cenering predictors

You might have noticed that the intercept doesn't always make sense. In our simple regression: Are we ever expected to have a child who is 0 months old? No! That child isn't even born yet!

So, the intercept value isn't very useful in this case. This is why you will often see researchers saying that they centered their predictors before doing the regression. Centering shifts all the values of your variable so that the overall distribution stays the same, but the mean of that distribution is now 0.

Commonly, we center our values using the average so that a value of 0 in the centered distribution corresponds to the average value of our original distribution.

Let's see this. Here is a distribution of the child's age:

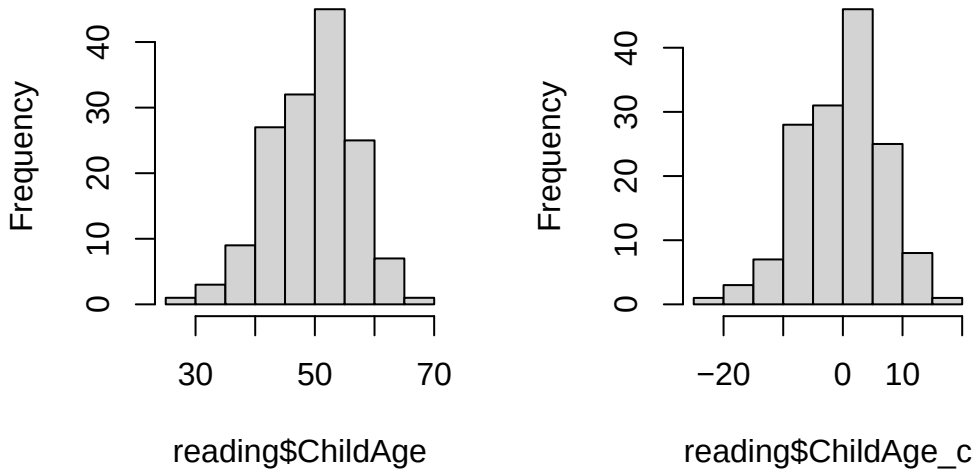
```
hist(reading$ChildAge)
```

You can see the mean is around 55.

Now we can center this by subtracting the mean from all values:

```
reading$ChildAge_c <- reading$ChildAge - mean(reading$ChildAge)
```

```
par(mfrow = c(1, 2))
hist(reading$ChildAge, main = NULL)
hist(reading$ChildAge_c, main = NULL)
```




```
dev.off()
```

```
null device
          1
```

Notice that the shape of the distribution has stayed the same, but the center is no longer at around 55 months, but is now at 0.

Let's also center our parent-child activities variable:

```
reading$ParentChildAct_c <- reading$ParentChildAct -
 ↪  mean(reading$ParentChildAct)
```

Let's re-run our regression to see what changes.

```
multiple_reg_centered <- lm(OverallLiteracy ~ ChildAge_c + ParentChildAct_c,
                            data = reading)
summary(multiple_reg_centered)
```

```
Call:
lm(formula = OverallLiteracy ~ ChildAge_c + ParentChildAct_c,
    data = reading)

Residuals:
    Min      1Q  Median      3Q     Max
-6.1942 -1.5785  0.1858  1.5431  4.9392

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      20.28922    0.17747 114.322  < 2e-16 ***
ChildAge_c        0.19992    0.02591   7.717 1.68e-12 ***
ParentChildAct_c  0.20860    0.04182   4.988 1.70e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.174 on 147 degrees of freedom
Multiple R-squared:  0.3655,    Adjusted R-squared:  0.3568
F-statistic: 42.34 on 2 and 147 DF,  p-value: 3.018e-15
```

Let's compare this with our previous model. Notice that our slopes have not changed: the effect of a 1-unit change in a variable, holding the other constant, stays the same.

What has changed is the intercept. The intercept still represents the expected literacy when all our predictors are 0. But now our predictors are centered, so instead of

$$Y_i = b_0 + b_1 X_{1_i} + b_2 X_{2_i} + \epsilon_i$$

we have,

$$Y_i = b_0 + b_1 (X_{1_i} - \bar{X_1}) + b_2 (X_{2_i} - \bar{X_2}) + \epsilon_i$$

So the predictors are only 0 when the variables are equal to the average value.

This makes the intercept more useful: it is the expected literacy when we have a child who is the average age and whose parents spend the average amount of time doing reading related activities with them.

Interactions

Interactions are how we can determine whether there is a moderation effect Moderation between 2 predictor variables is like saying "depends on" – the effect of one predictor variable "depends on" what value the other predictor takes.

This is often seen with categorical variables like biological sex (e.g., the effect of age depends on whether you're biologically male or female). But can be done with numeric variables too.

For our example, perhaps the effect of parent-child activities depends on the child's age. Maybe it matters that parents spend time doing these activities with their children when their children are a little bit older, versus when they're younger and parent-child activities has no effect. Since the effect depends on how old the child is, that is a moderation, or there is an interaction between child's age and parent-child activities.

To test an interaction in R, you just need to multiply your 2 predictors together instead of adding So it would be like `lm(outcome ~ pred1 * pred2)`, where `pred1 * pred2` is shorthand for `pred1 + pred2 + pred1:pred2`.

In our case:

```
interaction_model <- lm(OverallLiteracy ~ ChildAge_c * ParentChildAct_c, data
 ↪  = reading)
summary(interaction_model)
```

```

Call:
lm(formula = OverallLiteracy ~ ChildAge_c * ParentChildAct_c,
    data = reading)

Residuals:
    Min      1Q  Median      3Q     Max
-5.8585 -1.5671  0.1194  1.5494  4.8791

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                20.288743   0.177639 114.213  < 2e-16 ***
ChildAge_c                  0.204457   0.026470   7.724 1.65e-12 ***
ParentChildAct_c            0.211065   0.041958   5.030 1.42e-06 ***
ChildAge_c:ParentChildAct_c 0.005310   0.006219   0.854    0.395
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.176 on 146 degrees of freedom
```

```
Multiple R-squared:  0.3686,    Adjusted R-squared:  0.3557
F-statistic: 28.41 on 3 and 146 DF,  p-value: 1.568e-14
```

What does our output tell us:

Is our interaction term significant? No. So that means the effect of parent-child activities does not depend on how old the child is – it's equally important at all ages. If it was a significant effect, we could conclude that the effect of parent-child activities on child literacy depends on how old a child is.

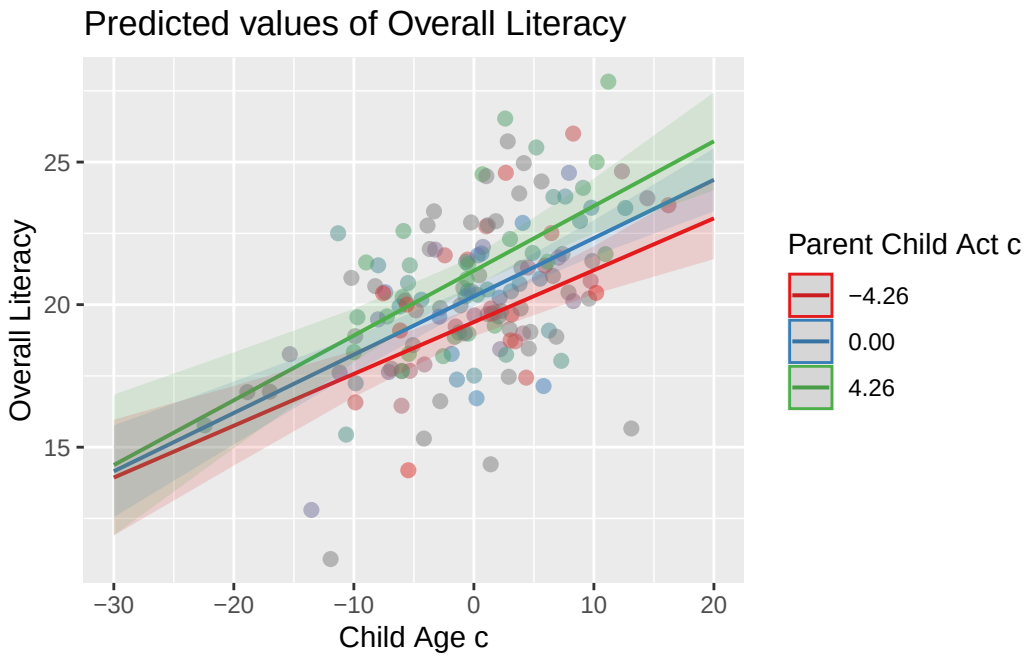
The effect wasn't significant, but I'll show you what to do if you *do* have a significant interaction. In that case, you want to understand this effect better, and it's usually better to graph the regression line between the outcome and one predictor at different values of the other predictor. If your "other predictor" is numeric, this is usually done at three values: 1 SD below the mean, 1 SD above the mean, and the mean.

Let's visualize the effect of parent-child activities on literacy at different values of children's ages. An easy way to do this is with a function from the `sjPlot` package:

```
# install.packages("sjPlot")
# un-comment the line above to install the sjPlot package if you haven't
 ↪  already.
library(sjPlot)
```

The `plot_model()` function will plot the regression lines for you, including the interaction term. Basically, all you have to do is specify `type = "int"` and R will plot with the first term in your interaction model on the x-axis, and the second term as a grouping factor. You can choose to plot at different values of your grouping variable, so we will specify the mean and +/- 1 SD using `mdrt.values = "meansd"`.

```
plot_model(interaction_model, type = "int", jitter= TRUE,
           mdrt.values = "meansd", show.data = TRUE)
```



You can see here that the slopes of all 3 lines are roughly the same, which is further proof that there is no interaction between our predictors. If there was an interaction, you might see something where as the child's age increases, you get different slopes for the effect of parent-child activities (e.g, maybe the slope also increases).