Lab 4 Instructions

BSTA 512/612

Nicky Wakim

> Caution
>
> Needs to be worked on!

Directions

Directions

Please turn in your `.html` file on Sakai. Please let me know if you greatly prefer to submit a physical copy.

You can download the `.qmd` file for this lab here. Please use the linked qmd file and not this one! (This is specifically the instructions.)

The rest of this lab's instructions are embedded into the lab activities.

> Caution
>
> This is the **instructions** file. The link above will take you to the **editing** file where you can add your work and turn it in!! Please do not remove anything from the editing file!!

Purpose

The main purpose of this lab is to perform model selection, identify one or more potential final models, and start our interpretation of our main relationship.

Grading

This lab is graded out of 12 points. Nicky will use the following rubric displayed on the Project page.

Lab activities

Before starting this lab, you should go back to Lab 2, save a new `.rda` file that contains all the new variables from that Lab. Then you can load it here!

Restate your research question

> **!** Task
>
> Please restate your research question below using the provided format. It's repetitive, but it helps me contextualize my feedback as I look through your lab.

How is implicit anti-fat bias, as measured by the IAT score, associated with "insert main independent variable here"?

Step 1: Simple linear regressions / analysis

We have done most of this step through visualizations in Lab 2 and 3. Now, we will quickly run a simple linear regression model for each covariate against the IAT score (outcome). Remember, the goal of this is to see if each covariate explains enough variation of the outcome, IAT score. You should have at least 9 simple linear regression models and their results. Results include the F-statistic and p-value from the test if each covariate explains enough variation of the outcome. Please revisit the slides from Lesson 5 (SLR: More inference + Evaluation) for more help with this test.

> ⚠ VERY IMPORTANT FOR VARIABLES WE ORDERED USING FACTOR!!
>
> I asked that you order variables to make plots more interpretable. However, for the `lm()`, R reads the ordered variables in an unexpected way. For these variables to run correctly in R, we need to unorder the variables. We can also set a reference level that makes sense.
> For example, I may want to unorder my variable `iam_001` and set the reference to `Neither underweight nor overweight`. I can do this with:

```
iat_2021_new = iat_2021_old %>%
  mutate(iam_unordered = factor( iam_ordered, ordered = FALSE ) %>%
           relevel( ref = "Neither underweight nor overweight"))
```

Recall, we mentioned 3 options to running and outputting the results of

1. We can run `lm()` for each covariate *in separate lines of code*, and use something like `summary()` or `anova()` to look at the results of each. (More time consuming to write, but less complicated coding)
2. We can use `lapply()` to run `lm()` and display the `anova()` on each covariate *in one line of code*. (Less time consuming to write, but more complicated coding, and more prone to errors that may not be apparent from output)
3. We can use `sapply()` to run `lm()`, `anova()`, and display the p-value for each covariate *in one line of code*. (Less time consuming to write, but more complicated coding, more prone to errors that may not be apparent from output, and no sense of what's going on in the regression)

Please take a note for yourself if your dataset contains the original numeric versions of variables that we created factors for. I am not saying that you should take them out. They might be useful if our sample is not big enough to handle all the categorical covariates that we've included, but I think our sample is large enough.

! Tasks

1. Run a simple linear regression model for each covariate against the IAT score (outcome).
2. Display results from the test if each covariate explains enough variation of the outcome. This may be from three options in the instructions: `summary()/anova()` only, `lapply()`, or `sapply()`

Interpretation of the results will be in the next step.

Step 2: Preliminary variable selection

Using the previous p-values from the F-test on each covariate's SLR, decide which covariates will be included in the initial model. Recall the decision rule: we keep covariates that explain enough variation using p-value < 0.25. Note that because our sample size is so large, the p-values might be really small. For now, that's okay, but this means we may want to alter our Step 3 a little bit.

Once you have decided on the covariates, run the model and display the regression table.

! Tasks

1. Decide which covariates will be included in the initial model and list them.
2. Run the initial model and display the regression table.

No need to write out the model, but you may *in addition* to the list.

Step 3: Assess change in coefficient

Now that all the selected variables are in one initial model, we can start considering the effect of each variable (outside of our main research question).

Remember our general rule: We can remove a variable if (1) p-value > 0.05 for the F-test to include or exclude the variable and (2) change in coefficient ($\Delta\%$) of our explanatory variable is $< 10\%$. *Please remember that the p-values for the F-test for a multi-level categorical variable must be calculated by creating a reduced and full model.*

It might be helpful to copy your list of covariates here and make note of the ones that you are removing. It was hard for me to keep track of all the variables when our dataset contains sooo many categorical covariates, and the regression table is so long.

Since our sample size is quite large, most (if not all) of the F-tests will conclude that the variable should be kept in the model. At this point, I advise that you turn to some common sense and the change in coefficients.

1. For **common sense**, you may notice that some of your covariates are essentially measuring the same thing. If there is clinical relevance to having both in the model, then keep them in, but if not, you will have to decide which is more interpretable/relevant/aligned with your research question. For example, if you chose variables involving attitudes and beliefs that are measuring similar things, then you might exclude one. There are measurements like "I am …" with relative weight groups and "Compared to most…" with relative weight groups. These two might capture a lot of the same information, so we may chose one. (Additionally, this might create issues with multicollinearity, which we will discuss on the last day, so just keep that in mind!) Another example is if you used gender identity, this might be a good time to throw out sex assigned at birth. Remember, my reasoning for using SAB was that (1) lab work has been extensive and I wanted to give you an option to avoid multi-selection variables, and (2) it *might* capture some of the differences around fat attitudes tied with gender. If you included gender identity in your work, then sex assigned at birth could be superfluous.
2. For **change in coefficients**, focus on the variable of your research question. Does the removal of variables change the coefficients for your explanatory variable? Remember what we discussed with change in coefficients when our explanatory variable is a multi-level categorical variable (Lesson 11.2 Interactions continued slides 26-28). You may find

these changes small, which tracks with a lot of our plots in Lab 3. Nothing seemed to have such a big effect on IAT score, and as a consequence it's hard to see big changes for a potential confounder.

Note that I put common sense first. The change in coefficients may not be very large, and may lead you to think we don't need a lot of the variables in our model. However, I would let common sense override the change in coefficients if your reasoning is well justified.

psst... There might be some code in Step 4 that might help you get started in this step.

> ! Tasks
>
> Remove variables from the initial model based on your common sense, change in coefficient, and/or p-values of the F-tests.
> **You do NOT need to show all your work here.** You just need to include:
>
> 1. A brief explanation of what variables were dropped and why (a sentence per variable), and
> 2. An example of your process with one variable is enough (including code that you ran)

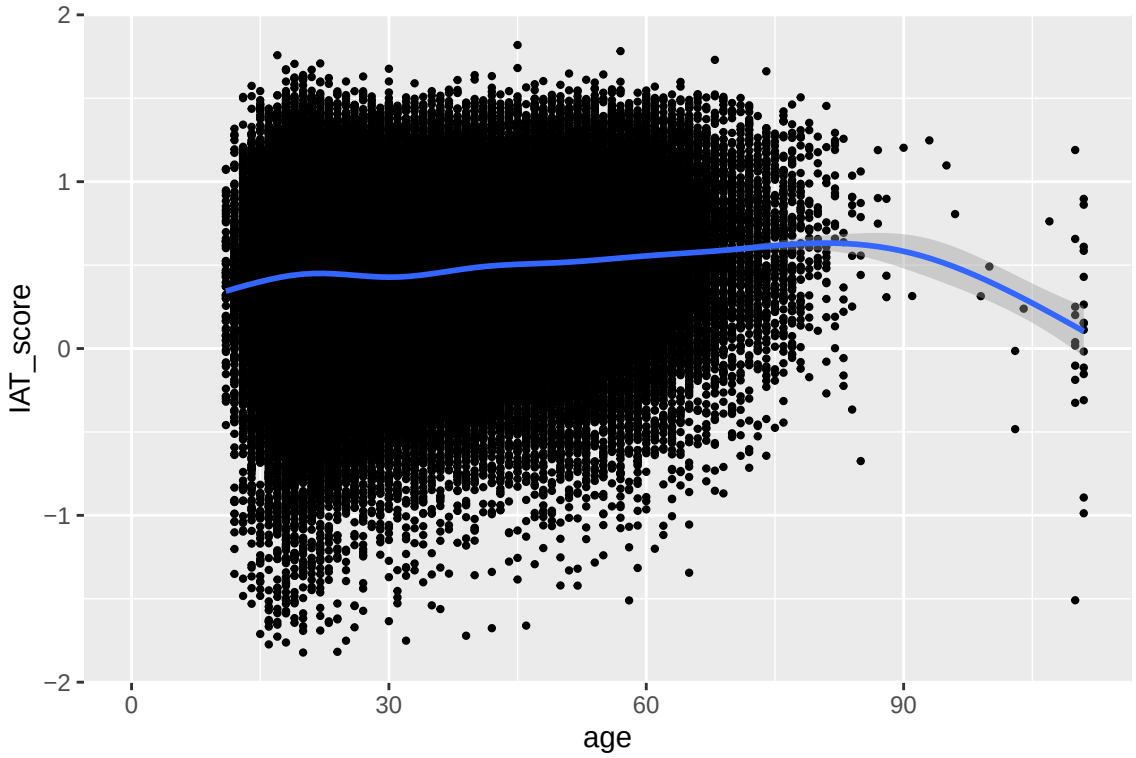
Step 4: Assess scale for continuous variables

There is one variable in our model (unless you removed it) that is continuous: age. We need to assess the scale for age. **In this step we will have ZERO delivarables.** To save you time, I will walk you through my thought process, and why I determined age is fine as is. If you still want to try something else out with age, then you can!

First, we can start with a scatterplot of IAT score and age. Your plot may look a little different than mine.

```
ggplot(data = iat, aes(x = age, y = IAT_score)) +
  geom_point(size = 0.8) + geom_smooth() + xlim(0, 111)
```

```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



In the above scatterplot, it looks like the relationship is mostly linear (and increasing) until we get to approximately 90 years old. At that point IAT score decreases with age. Let's say we 100% believe there is suddenly more responders around 110 years old than 90-100 year olds. I'm already skeptical of this since we did a quality control in Lab 3. We'll play it out because it's not worth making judgement calls on what we consider "admissable" data.

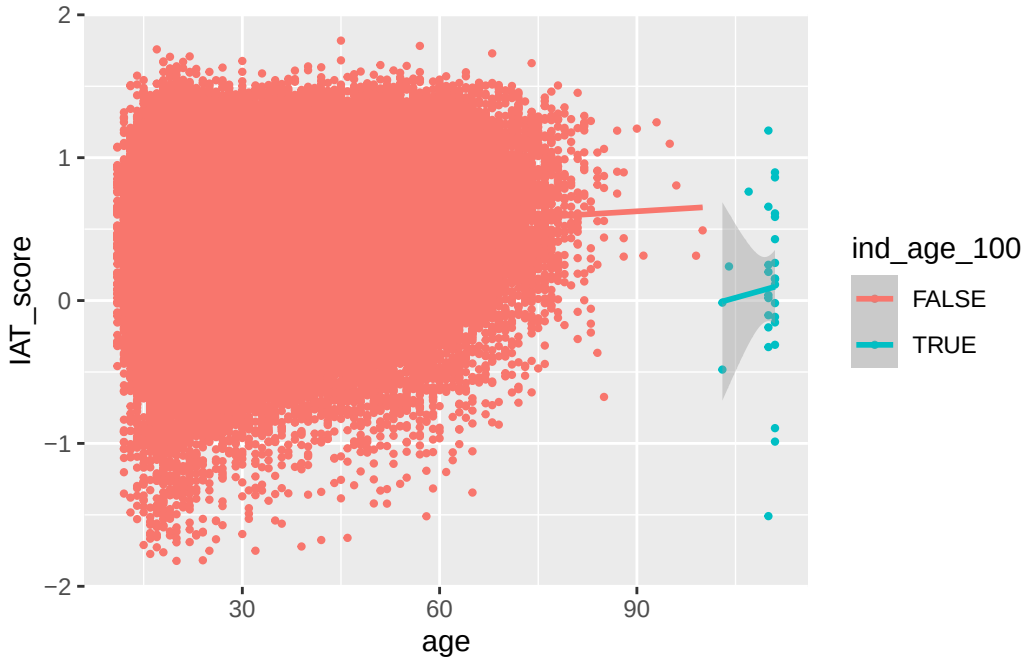
We could do quantiles, splines, or polynomials, but those approaches will either make more categorical variables or make the relationship between age and IAT score harder to interpret. We have a pretty linear relationship up until the higher ages!

I wanted to investigate the linearity a little more so I created an indicator for individuals who are 100 years or older:

```
iat1 = iat %>% mutate(ind_age_100 = ifelse(age > 100, "TRUE", "FALSE"))
```

Now I can see if the linearity differs between the two groups of ages:

```
ggplot(iat1, aes(x = age, y = IAT_score, color = ind_age_100)) +
  geom_point(size = 0.8) + geom_smooth(method = "lm")
```



I am happy to see that both groups' IAT score are increasing with age. It actually looks like my indicator might be a confounder… In that case, we only need to include the indicator in the model so that the relationship between age and IAT is adjusted for the indicator. I can test to see if the indicator is a big enough confoudner using the change in coefficient of age and my explanatory variable.

Here's the model without the indicator:

```
prelim_model = lm(IAT_score ~ iam_unordered + identfat + comptomost +
                    ind_m + ind_f + ind_tmm + ind_twf + ind_gqnc +
                      ind_other +
                    race +
                    ethn +
                    edu_14_f +
                    age, data = iat1)
```

And we'll take a look at the coefficients for the model:

```
prelim_model$coefficients[c(2:6, 46)] # by using c(2:6, 46) I am telling R to
```

```
      iam_unorderedVery underweight iam_unorderedModerately underweight
                        -0.061151601                         -0.015513320
  iam_unorderedSlightly underweight    iam_unorderedSlightly overweight
```

```
                        0.006954580                         -0.023369363
iam_unorderedModerately overweight                                  age
                       -0.056237038                          0.003857134
```

```
                                    # only print certain variables' coefficients
```

Then we can run the model with the indicator, then look at the coefficients:

```
prelim_model2 = lm(IAT_score ~ iam_unordered + identfat + comptomost +
                   ind_m + ind_f + ind_tmm + ind_twf + ind_gqnc + ind_other +
                   race +
                   ethn +
                   edu_14_f +
                   age + ind_age_100,
                  data = iat1)
prelim_model2$coefficients[c(2:6, 46)] # by using c(2:6, 46) I am telling R to
```

```
      iam_unorderedVery underweight iam_unorderedModerately underweight
                        -0.060385515                         -0.015352043
  iam_unorderedSlightly underweight    iam_unorderedSlightly overweight
                         0.006817642                         -0.023677757
iam_unorderedModerately overweight                                  age
                        -0.056762301                          0.003918822
```

```
                                    # only print certain variables' coefficients
```

We can check the % change in the coefficients between the models.

Recall,

$$\Delta\% = 100\% \cdot \frac{\hat{\beta}_{FLR,full} - \hat{\beta}_{FLR,red}}{\hat{\beta}_{FLR,full}}$$

Here's how I quickly do it with the coefficients:

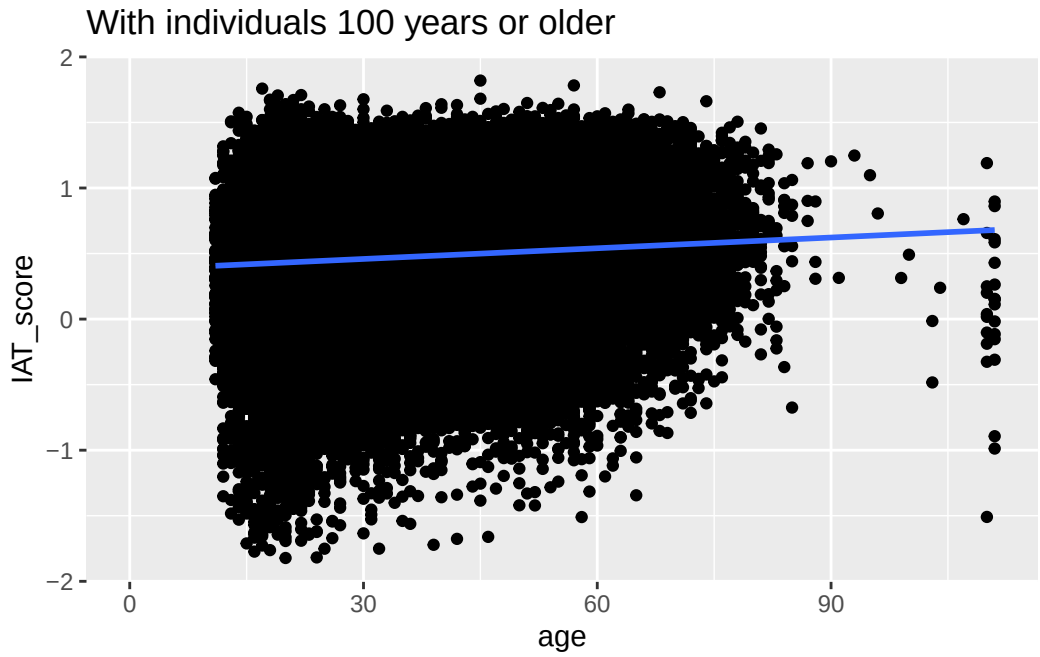
```
100 * ( prelim_model2$coefficients[c(2:6, 46)] - prelim_model$coefficients[c(2:6, 46)] ) /
  prelim_model2$coefficients[c(2:6, 46)]
```

```
      iam_unorderedVery underweight iam_unorderedModerately underweight
                         -1.2686585                          -1.0505274
  iam_unorderedSlightly underweight    iam_unorderedSlightly overweight
                         -2.0085770                           1.3024639
iam_unorderedModerately overweight                                  age
                          0.9253735                           1.5741387
```

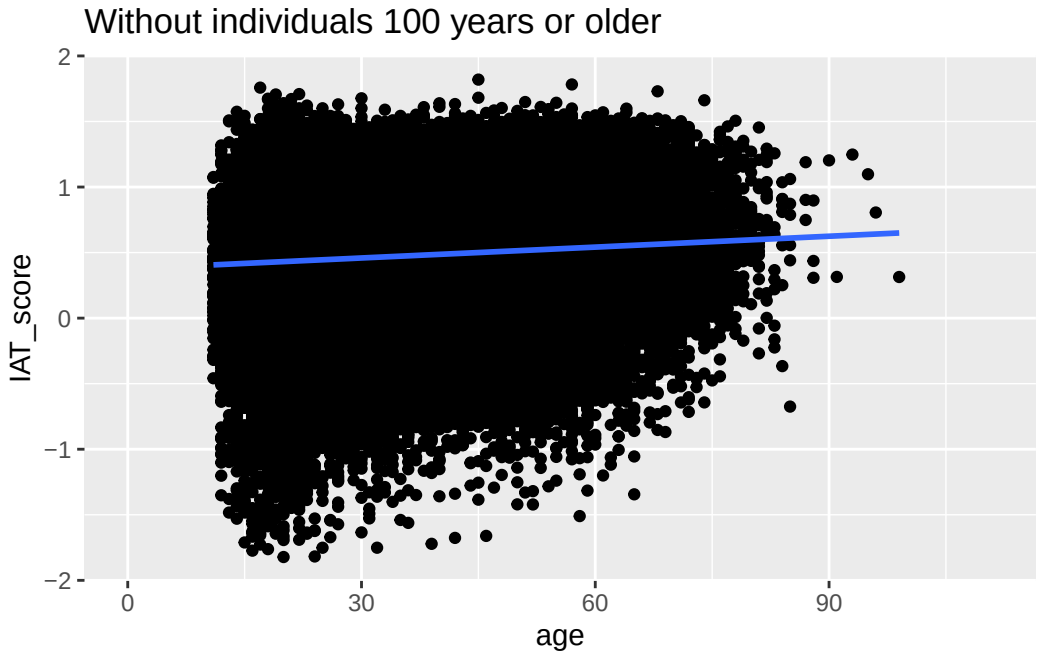
Based on %'s above, it doesn't look like the indicator makes much of a difference in my model. It is likely because there are only 29 individuals over the age of 100 and 201,031 individuals under the age of 100 (In my dataset). Those 29 individuals will not have a big impact on the linear relationship between age and IAT, even though the first smoothed scatterplot made it look like it does.

To bring this point home, I can plot age and IAT with and without the individuals that are 100 years or older. Let me know if you find a better way to overlay these plots! (I have been a little stressed on time, and couldn't find a quick answer.)

```
ggplot(iat1, aes(x = age, y = IAT_score)) +
  geom_point() + geom_smooth(method = "lm") + xlim(0, 111) +
  labs(title = "With individuals 100 years or older")
```




```
ggplot(iat1 %>% filter(age < 100), aes(x = age, y = IAT_score)) +
  geom_point() + geom_smooth(method = "lm") + xlim(0, 111) +
  labs(title = "Without individuals 100 years or older")
```



I see no difference. Thus, I think it's okay to leave age as is!!

> **!** Tasks
>
> No tasks here! If you want to try out what I did above, you can!

Step 5: Check for interactions

Now we're going to check if there are any interactions. I will walk you through a streamlined way to check for interactions between your explanatory variable and all the other variables in the model.

First, I want you to revisit your work in Lab 3. Remind yourself of the variables that you identified as possible effect modifiers.

As you check for interactions, don't forget to make your decisions **based on your discussion/hypotheses in Lab 3**. Always prioritize investigation of interactions that are justified clinically before investigating interactions only based on statistical significance.

```
vars = names(model.frame(prelim_model))[-1]                    ①

.env <- environment()
```

```
interactions <- combn(vars, 2, function(x) paste(x, collapse=" * ")) %>%    ②
    grep(., pattern = "iam_unordered", value = T)                           ③
```

① Create a vector of the variable names that are in your preliminary model. Note I use `[-1]` to remove `IAT_score` from my list. Please make sure to change `prelim_model` to the name of your model at this point.
② Here we are just combining all our covariates into interactions that R can understand. This makes it so we don't have to write it all ourselves.
③ Make sure to change the `pattern = "iam_unordered"` to be `pattern =` to your explanatory variable.

Now that we've created the set up for all the possible interactions, we can run them through the `lm()` function and see the summary of the models. In the following code I use the `lappy()` function to fit an individual model for the main effects + each interaction listed in `interactions`.

> **Note**
>
> Please note that this code takes a while to run. Once you run it and take note of the results, you can comment out or add `#| eval: false` to prevent it from running every time you render. You don't need to show the results for this in your submitted work, but I want to see the code, and read about your decisions about from results.

```
summary = lapply(interactions,
             function(int) summary(lm(reformulate(c(vars, int), "IAT_score", env=.env),
                                      data = iat)))
summary
```

You can alse go straight to using the `anova()` function to compare the preliminary model.

```
anova_res = lapply(interactions,
             function(int) anova(lm(reformulate(c(vars, int), "IAT_score", env=.env),
                                      data = iat),
                                 prelim_model))                             ①
anova_res[1]
g = anova_res[[1]]
g$F
```

① You will to change this name for the preliminary model if you called it something different.

```
[[1]]
Analysis of Variance Table
```

```
Model 1: IAT_score ~ iam_unordered + identfat + comptomost + ind_m + ind_f +
    ind_tmm + ind_twf + ind_gqnc + ind_other + race + ethn +
    edu_14_f + age + iam_unordered * identfat
Model 2: IAT_score ~ iam_unordered + identfat + comptomost + ind_m + ind_f +
    ind_tmm + ind_twf + ind_gqnc + ind_other + race + ethn +
    edu_14_f + age
  Res.Df   RSS  Df Sum of Sq     F    Pr(>F)
1 200990 31337
2 201014 31346 -24   -9.4223 2.518 5.558e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

[1]       NA 2.518048
```

! Tasks

Using your discussion in Lab 3 and the results from the F-test on interactions:

1. Create a list of the interactions that you will include in your model.
2. Run the preliminary final model that includes the main effects and interactions.

Step 6: Assess model fit

At this point we may want to compare different models. While Steps 1-5 have been directing us towards a single model, you may have been interested in other models along the way. Maybe there were some interactions that you thought were interesting, but didn't think of before. Maybe you would like to combine different groups for categorical variables.

If you are completely happy with your model, then you don't have to do this step.

You might create a table like such:

```r
sum = summary(prelim_model)
model_fit_stats = data.frame(Model = "Preliminary main effects model",
                             Adjusted_R_sq = sum$adj.r.squared,
                             AIC = AIC(prelim_model),
                             BIC = BIC(prelim_model))

model_fit_stats
```

```
                          Model Adjusted_R_sq      AIC      BIC
1 Preliminary main effects model    0.04511326 197006.6 197486.5
```

> **!** Tasks
>
> **Optional:** Create a table that displays some fo the model fit statistics to compare preliminary final models.

Create a forest plot of your coefficient estimates

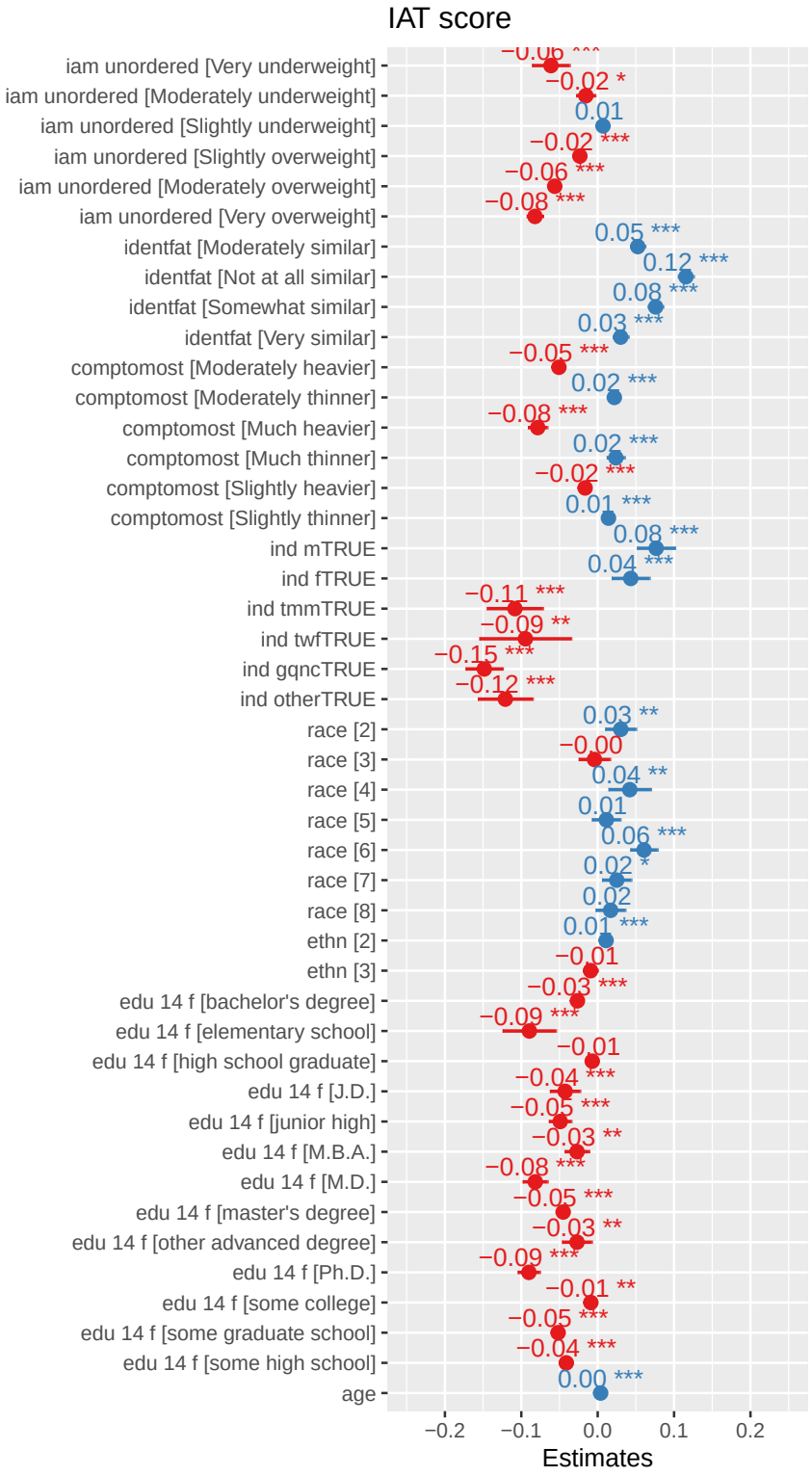
It's often helpful to have a visualization of coefficient estimates. Forest plots are a nice way to show all the values together. Below I have started a forest plot using my `prelim_model`. You can make the plot with your final model.

I used the `plot_model()` function to make the plot, and here's a site that discusses some of it's capabilities. The below plot is just a starting point!! You'll need to clean up the variables, title, etc.

You may use another function to make the plots. I chose this one since it can handle the model as input.

```
plot_model(prelim_model, show.values = TRUE, value.offset = 0.5) + ylim(-0.25, 0.25)
```

```
Scale for y is already present.
Adding another scale for y, which will replace the existing scale.
```


IAT score
iam unordered [Very underweight]
iam unordered [Moderately underweight]
iam unordered [Slightly underweight]
iam unordered [Slightly overweight]
iam unordered [Moderately overweight]
iam unordered [Very overweight]
identfat [Moderately similar]
identfat [Not at all similar]
identfat [Somewhat similar]
identfat [Very similar]
comptomost [Moderately heavier]
comptomost [Moderately thinner]
comptomost [Much heavier]
comptomost [Much thinner]
comptomost [Slightly heavier]
comptomost [Slightly thinner]
ind mTRUE
ind fTRUE
ind tmmTRUE
ind twfTRUE
ind gqncTRUE
ind otherTRUE
race [2]
race [3]
race [4]
race [5]
race [6]
race [7]
race [8]
ethn [2]
ethn [3]
edu 14 f [bachelor's degree]
edu 14 f [elementary school]
edu 14 f [high school graduate]
edu 14 f [J.D.]
edu 14 f [junior high]
edu 14 f [M.B.A.]
edu 14 f [M.D.]
edu 14 f [master's degree]
edu 14 f [other advanced degree]
edu 14 f [Ph.D.]
edu 14 f [some college]
edu 14 f [some graduate school]
edu 14 f [some high school]
age
−0.02 *
0.01
−0.02 ***
−0.06 ***
−0.08 ***
0.05 ***
0.12 ***
0.08 ***
0.03 ***
−0.05 ***
0.02 ***
−0.08 ***
0.02 ***
−0.02 ***
0.01 ***
0.08 ***
0.04 ***
−0.11 ***
−0.09 **
−0.15 ***
−0.12 ***
0.03 **
−0.00
0.04 **
0.01
0.06 ***
0.02 *
0.02
0.01 ***
−0.01
−0.03 ***
−0.09 ***
−0.01
−0.04 ***
−0.05 ***
−0.03 **
−0.08 ***
−0.05 ***
−0.03 **
−0.09 ***
−0.01 **
−0.05 ***
−0.04 ***
0.00 ***
−0.2
−0.1
0.0
0.1
0.2
Estimates

Here are some other packages for forest plots:

- https://cran.r-project.org/web/packages/forestploter/vignettes/forestploter-intro.html
- https://larmarange.github.io/ggstats/articles/ggcoef_model.html

! Tasks

Create a forest plot to visualize the coefficient estimates.