Exercise 01 - Special predictors in linear models

Zoltan Kekecs

16 november 2021

Contents

1 Abstract

In the previous exercises we used continuous predictors and modeled simple linear relationships between the predictors and the outcome without too much concern about the relationship of the predictors on each others effect on the outcome. In this exercise we will expand the array of predictors to categorical variables, interaction terms, and higher order terms.

2 Data management and descriptive statistics

2.1 Load packages

```
library(tidyverse)
library(psych)
library(gridExtra)
```

2.2 Load the weight loss dataset

To explore some of the more advanced predictor types, we will need a new dataset. Let's download the weight_loss dataset.

```
data_weightloss =
↪  read_csv("https://raw.githubusercontent.com/kekecsz/SIMM61-Course-materials/main/Exercise_01%20-%20
```

This dataset contains simulated data. It is about a study where different types of interventions were tested to help overweight people to lose weight.

Variables:

- ID - participant ID
- Gender - gender
- Age - age
- BMI_baseline - Body mass index (BMI) measured before treatment
- BMI_post_treatment - Body mass index (BMI) measured after treatment
- treatment_type - The type of treatment in the group to which the participant was randomized to. Levels:
    1. no treatment
    2. pill - medication which lowers apetite
    3. psychotherapy - cognitive behavioral therapy
    4. combined - a third kind of treatment, where theboth pills and psychotherapy ws used
- got_pill - represent whether a person got pills as part of their treatment, if yes this is coded as 1, if no this is coded as 0
- got_psychotherapy - represents whether a person got psychotherapy as part of their treatment, if yes this is coded as 1, if no this is coded as 0
- motivation - self report motivation to lose weight (on a 0-10 scale from extremely low motivation to extremely high motivation)
- body_acceptance - how much the person feels that he or she is satisfied with his or her body. (on a scale of -7 to +7 from very unsatisfied to very satisfied)

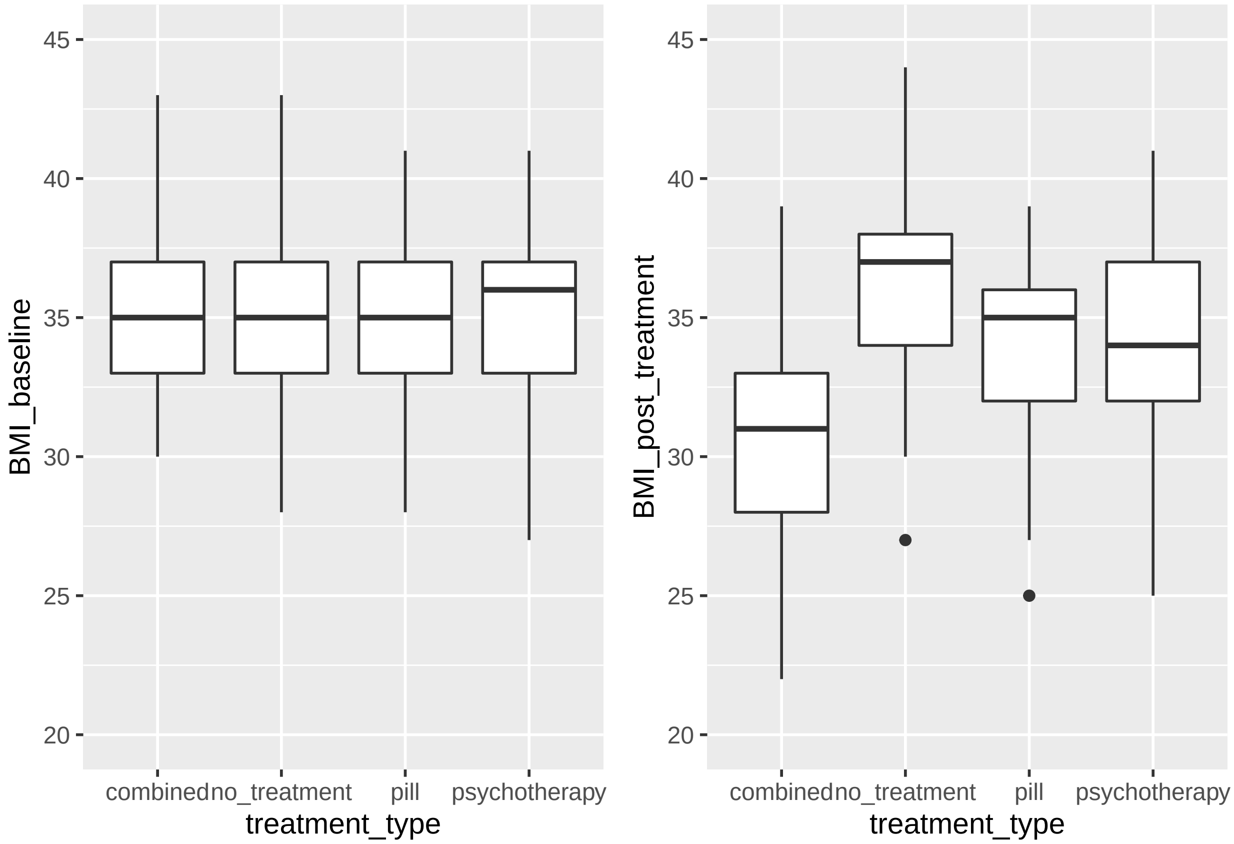
2.3 check data

Lets explore the dataset we will use

```
data_weightloss %>%
  summary()

describe(data_weightloss)
```

Now lets run some more focused exploratory analysis on the variables of interest. In this exercise we would like to understand the effect of the different treatment types on BMI.

```
fig_1 = data_weightloss %>%
  ggplot() +
  aes(y = BMI_baseline, x = treatment_type) +
  geom_boxplot()+
  ylim(c(20, 45))

fig_2 = data_weightloss %>%
  ggplot() +
  aes(y = BMI_post_treatment, x = treatment_type) +
  geom_boxplot()+
  ylim(c(20, 45))

grid.arrange(fig_1, fig_2, nrow=1)
```



```
data_weightloss %>%
  group_by(treatment_type) %>%
    summarize(mean_pre = mean(BMI_baseline),
              sd_pre = sd(BMI_baseline),
              mean_post = mean(BMI_post_treatment),
              sd_post = sd(BMI_post_treatment))
```

```
## # A tibble: 4 x 5
##   treatment_type mean_pre sd_pre mean_post sd_post
##   <chr>             <dbl>  <dbl>     <dbl>   <dbl>
## 1 combined           35.2   2.95      30.8    3.41
## 2 no_treatment       34.9   3.06      36.1    3.49
## 3 pill               35.0   2.50      34.0    2.95
## 4 psychotherapy      34.8   3.09      34.1    3.40
```

2.4 Categorcial predictors

Because it seems that the groups are comparable at baseline, lets focus on the post-treatment BMI.

The treatment_type is a categorical variable, while BMI is a numeric continuous variable. Thus, one of the ways in which the relationship of treatment_type and BMI can be explored is **to use a one-way ANOVA** (using the aov() function).

```
anova_model = aov(BMI_post_treatment ~ treatment_type, data = data_weightloss)
sum_aov = summary(anova_model)
sum_aov
```

```
##                 Df Sum Sq Mean Sq F value   Pr(>F)    
## treatment_type   3    877  292.33   26.51 8.17e-15 ***
## Residuals      236   2602   11.03                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The result of this test tell us that the mean post-treatment BMI is significantly different among groups (F (3, 236) = 26.51, p < .001).

In linear regression it is important that the predicted variable be a numeric continuous variable. However, the predictors are not constrained by this, so predictors can be categorical as well.

This means that **we could build** the above one-way ANOVA model **with lm() as well**.

Notice that the result of the full model **F-test is the same** as the result of the aov().

```
mod_1 = lm(BMI_post_treatment ~ treatment_type, data = data_weightloss)
summary(mod_1)
```

```
## 
## Call:
## lm(formula = BMI_post_treatment ~ treatment_type, data = data_weightloss)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -9.133 -2.133 -0.050  2.200  8.200 
## 
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                  30.8000     0.4287  71.846  < 2e-16 ***
## treatment_typeno_treatment    5.3333     0.6063   8.797 3.02e-16 ***
## treatment_typepill            3.2500     0.6063   5.361 1.97e-07 ***
## treatment_typepsychotherapy   3.3333     0.6063   5.498 9.94e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.321 on 236 degrees of freedom
## Multiple R-squared:  0.2521, Adjusted R-squared:  0.2425
## F-statistic: 26.51 on 3 and 236 DF,  p-value: 8.173e-15
```

The regression coefficient table looks different than usual, because instead of having a single row for our predictor, **we have multiple rows**, one for each treatment type (except for one).

2.5 Interpreting the coefficients table for categorical variables

Remember what was the interpretation for the regression coefficents and the intercept from the previous exercises.

The interpretation of the regression coefficients of the predictors is: this is the amount by which the outcome variable's estimate would change if the predictor's value is increased by 1.

Interpretation of the estimate of the intercept: if all the predictors in the model would have the value of zero (0), this would be the estimated value for the outcome variable.

This interpretation stays the same for every linear model.

2.5.1 Dummy coding

(This is done automatically by R so we do not usually have to do this manually, we do the dummy coding here **as a demonstration**.)

However, for nominal predictors (like treatment_type) we do not have a numerical value for the predictor levels by default. To assign numerical value to the different factor levels, we need to **dummy code the categorical predictor**. This basically means that we will **create separate variables representing each level of the categorical variable, except for the default level**. So

- we create a variable (pill_group) that will take the value of 1 if the person got the treatment "pill", and take the value of 0 every other time.
- we create a variable (psychotherapy_group) that will take the value of 1 if the person got the treatment "psychotherapy", and take the value of 0 every other time.
- we create a variable (combined_group) that will take the value of 1 if the person got the treatment "combined", and take the value of 0 every other time.
- we ususally only create one fewer dummy variables than the number of levels in the categorical predictor, and leave the "default level" of the predictor un-dummied. In our case the "default level" is "no_treatment", and we would like to compare the effect of each other level to this level.

Now if we **fit a regression model using these dummies**, we will see how the results of the previously seen regression output were generated.

```
data_weightloss = data_weightloss %>%
  mutate(
        pill_group = recode(treatment_type,
                            "no_treatment" = "0",
                            "pill" = "1",
                            "psychotherapy" = "0",
                            "combined" = "0"),
        psychotherapy_group = recode(treatment_type,
                            "no_treatment" = "0",
                            "pill" = "0",
                            "psychotherapy" = "1",
                            "combined" = "0"),
        combined_group = recode(treatment_type,
                            "no_treatment" = "0",
                            "pill" = "0",
                            "psychotherapy" = "0",
                            "combined" = "1")
        )

mod_2 = lm(BMI_post_treatment ~ pill_group + psychotherapy_group + combined_group, data =
↪  data_weightloss)
summary(mod_2)
```

```
##
## Call:
## lm(formula = BMI_post_treatment ~ pill_group + psychotherapy_group +
##     combined_group, data = data_weightloss)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.133 -2.133 -0.050  2.200  8.200
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           36.1333     0.4287  84.287  < 2e-16 ***
## pill_group1           -2.0833     0.6063  -3.436 0.000697 ***
## psychotherapy_group1  -2.0000     0.6063  -3.299 0.001121 **
## combined_group1       -5.3333     0.6063  -8.797 3.02e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.321 on 236 degrees of freedom
## Multiple R-squared:  0.2521, Adjusted R-squared:  0.2425
## F-statistic: 26.51 on 3 and 236 DF,  p-value: 8.173e-15
```

This model produces the **same results as the model above** where we entered the variable name of the categorical predictor directly into the model, because **the lm() function does all the dummy-ing automatically**. It is important to understand though **how the "default level" is selected** in this automatic process. It is selected to be the first level name in alphabetic order.

Now we can use the **same interpretation that we are used to for the regression coefficients** in the coefficient table: The intercept is the value of the outcome if every predictor's value is zero. This basically means that this is the predicted value of the outcome variable at the "default level" or "reference level". The coefficients of the predictors show the amount of change in the outcome variable's value if the given predictor's value increases by 1 point. This can happen for "pill_group" if the person got the treatment "pill". **This change is always in relation to the default level**.

So in our example:

- In case of "no_treatment" we can expect a post-test BMI of 36.13,
- if someone was in the "pill" group instead, we expect a BMI -2.08 difference compared to the intercept,
- if someone was in the "psychotherapy" group instead, we expect a BMI -2.00 difference compared to the intercept
- if someone was in the "combined" group instead, we expect a BMI -5.33 difference compared to the intercept

Practice

Open the house sale dataset from the previous exercise and build a regression model where we predict the sales price with sqm_living, grade, has_basement as predictors. has_basement is a categorical predictor with two levels: “has basement” and “no basement”. Interpret the regression coefficents based on the description above. Pay attention to what is the default level and why in order to be able to interpret the meaning of the values correctly.

How much more or less can a person expect to get for their apartment if it has a basement?

How would you interpret the intercept in this model?

```
data_house = read.csv("https://bit.ly/2DpwK0r")

data_house = data_house %>%
  mutate(sqm_living = sqft_living * 0.09290304,
         sqm_lot = sqft_lot * 0.09290304,
         sqm_above = sqft_above * 0.09290304,
         sqm_basement = sqft_basement * 0.09290304,
         sqm_living15 = sqft_living15 * 0.09290304,
         sqm_lot15 = sqft_lot15 * 0.09290304,
         has_basement = factor(has_basement))
```

2.6 Introducing interaction terms into the model

The **combined treatment** is a condition where the **person got both pill and psychotherapy** treatments at the same time. Whether the person got pill as part of their treatment, and whether the person got psychotherapy as part of their treatment is represented by the got_pill and the got_psychotherapy variables respectively.

Now we can answer the question whether there is an **interaction between the pill and the psychotherapy treatments**. That is, can we expect a **different effect of one of these predictors on the outcome depending on the value of the other predictor** (for example a multiplicative effect), or are the effects completely independent from each other (this would represent a simple additive effect).

We can **enter the interaction term into the model** by using a * instead of a + between the predictors we want to include the interaction of.

```
mod_3 = lm(BMI_post_treatment ~ got_pill * got_psychotherapy, data = data_weightloss)
summary(mod_3)
```

```
##
## Call:
## lm(formula = BMI_post_treatment ~ got_pill * got_psychotherapy,
##     data = data_weightloss)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.133 -2.133 -0.050  2.200  8.200
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 36.1333     0.4287  84.287  < 2e-16 ***
## got_pill                    -2.0833     0.6063  -3.436 0.000697 ***
## got_psychotherapy           -2.0000     0.6063  -3.299 0.001121 **
## got_pill:got_psychotherapy  -1.2500     0.8574  -1.458 0.146194
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.321 on 236 degrees of freedom
## Multiple R-squared:  0.2521, Adjusted R-squared:  0.2425
## F-statistic: 26.51 on 3 and 236 DF,  p-value: 8.173e-15
```

Now we have **a new predictor in our model**, which is basically the **product of** got_pill * got_psychotherapy. The interpretation of the regression coefficient for the interaction term is the same as before, that is: if the product of got_pill and got_psychotherapy increases by 1, we can expect this change in the value of the predicted variable.

We need to realize that this also means that **the value of got_pill and/or the value of got_psychotherapy also needs to change** (to produce a change in the product), so the effect of change in the original predictors,also called **the "main effect" is already factored in**, meaning that the coefficient of the interaction term can be **interpreted as the unique effect of the interaction** of the two predictors not including the "main effect" of the predictors alone.

In our example this means that:

- In case of "no_treatment" we can expect a post-test BMI of 36.13,
- The main effect of getting pill treatment on BMI is -2.08
- The main effect of getting psychotherapy treatment on BMI is -2.00
- the interaction effect of pill and psychotherapy is 1.25

_______________*Practice*_________________

Build a new model where you predict **BMI_post_treatment** with the predictors **motivation** and **body_acceptance**. Interpret the coefficients. How much change in BMI can a person expect if the level of motivation is increased by 1? How much change in BMI can a person expect if the level of body_acceptance is increased by 1? Is there a significant interaction between the two predictors? How is the coefficient of the interaction term interpreted?

2.7 Including higher order terms in regression models to model non-linear relationships

Let's build a linear regression model with body_acceptance as a predictor of post-treatment BMI the usual way.

```
mod_4 = lm(BMI_post_treatment ~ body_acceptance, data =  data_weightloss)
summary(mod_4)
```

```
##
## Call:
## lm(formula = BMI_post_treatment ~ body_acceptance, data = data_weightloss)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.6960  -2.2936   0.0052   2.5112   8.9136
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      32.6960     0.3613  90.490  < 2e-16 ***
## body_acceptance  -0.5976     0.1495  -3.996 8.59e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 3.701 on 238 degrees of freedom
## Multiple R-squared:  0.06287,    Adjusted R-squared:  0.05894
## F-statistic: 15.97 on 1 and 238 DF,  p-value: 8.595e-05
```

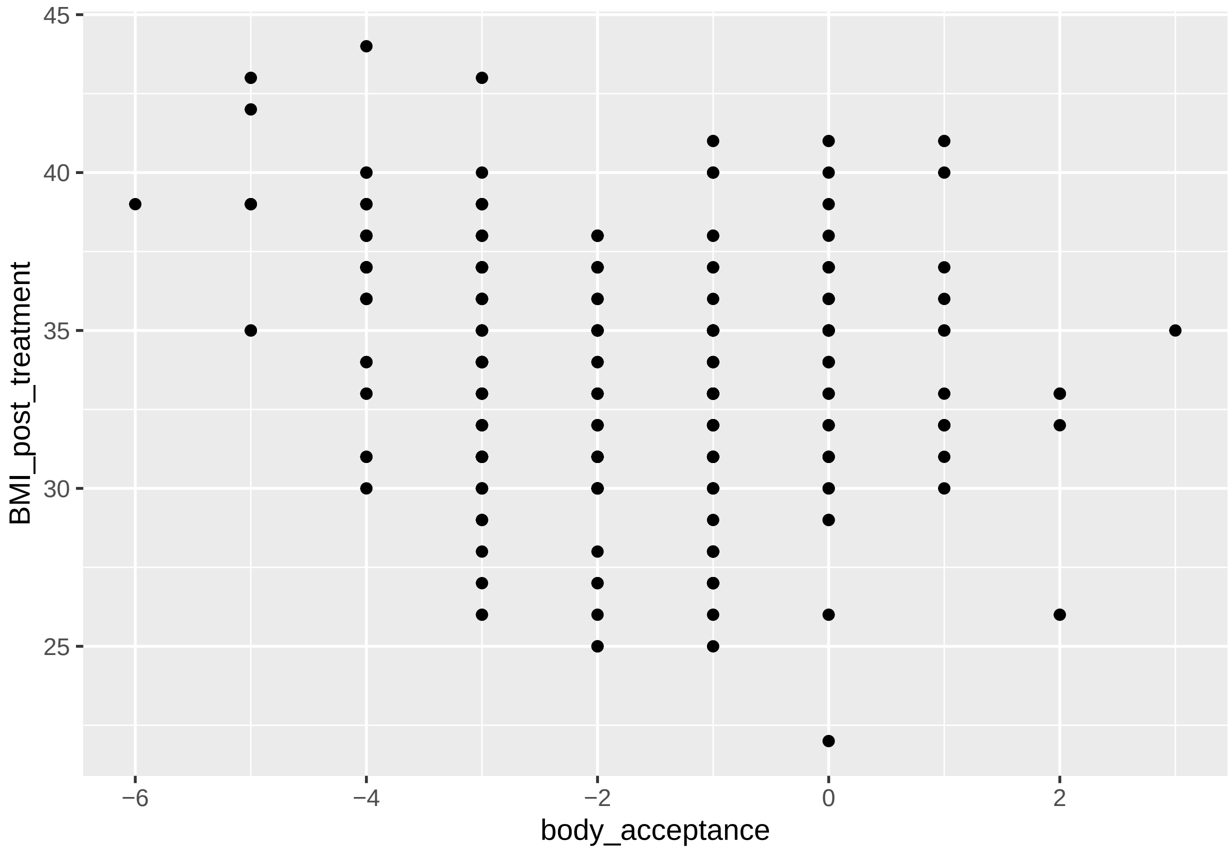
The coefficient table tells us that with every step up the value of body_acceptance, we can expect -0.60 change in BMI post treatment (so the more satisfied the person is with their body at baseline, the higher the BMI will be at post-test, note that this is probably because of the higher baseline BMI of those who are less satisfied with their body to begin with).

The output tells us that this model is significantly better than a null model (F (1, 238) = 15.97, p < .001, Adj. R^2 = 0.06, AIC = 1313.25). This means that taking into account body acceptance adds significant predictive power to the model (this being the only predictor).

However, the variance explained by this model is mediocre, explaining only 6% of the variance.

Let's explore this relationship with a scatterplot.

```
data_weightloss %>%
  ggplot() +
  aes(y = BMI_post_treatment, x = body_acceptance) +
  geom_point()
```



This scatterplot indicates that there might be **a non-linear relationship** between BMI and body acceptance.

Linear models are originally designed to model linear relationships between predictors and outcomes, but with a little matematical trick we can model non-linear relationships as well. In order to do this, we need to incluide the higher order terms of the predictor in the model as well.

This can be included in the model by **adding + I(body_acceptance^2)**.

Baed on the model summary and the model fit indices, this model fits the data better, and explains more of the variability of the predicted variable (BMI).

```
mod_5 = lm(BMI_post_treatment ~ body_acceptance + I(body_acceptance^2), data =
↪  data_weightloss)
summary(mod_5)
```

```
##
## Call:
## lm(formula = BMI_post_treatment ~ body_acceptance + I(body_acceptance^2),
##     data = data_weightloss)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7602  -2.2547   0.1633   2.3218   8.7453
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          32.76024    0.35018  93.552  < 2e-16 ***
## body_acceptance       0.37209    0.27684   1.344     0.18
## I(body_acceptance^2)  0.29008    0.07059   4.110 5.46e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.584 on 237 degrees of freedom
## Multiple R-squared:  0.1252, Adjusted R-squared:  0.1178
## F-statistic: 16.96 on 2 and 237 DF,  p-value: 1.305e-07
```

```
AIC(mod_4)
```

```
## [1] 1313.253
```

```
AIC(mod_5)
```

```
## [1] 1298.732
```

It is important to **include all of the lower order terms as well** in the model if we include higher order terms, for the model to work as intended.

```
mod_6 = lm(BMI_post_treatment ~ body_acceptance + I(body_acceptance^2)+
↪  I(body_acceptance^3), data =  data_weightloss)
summary(mod_6)
```

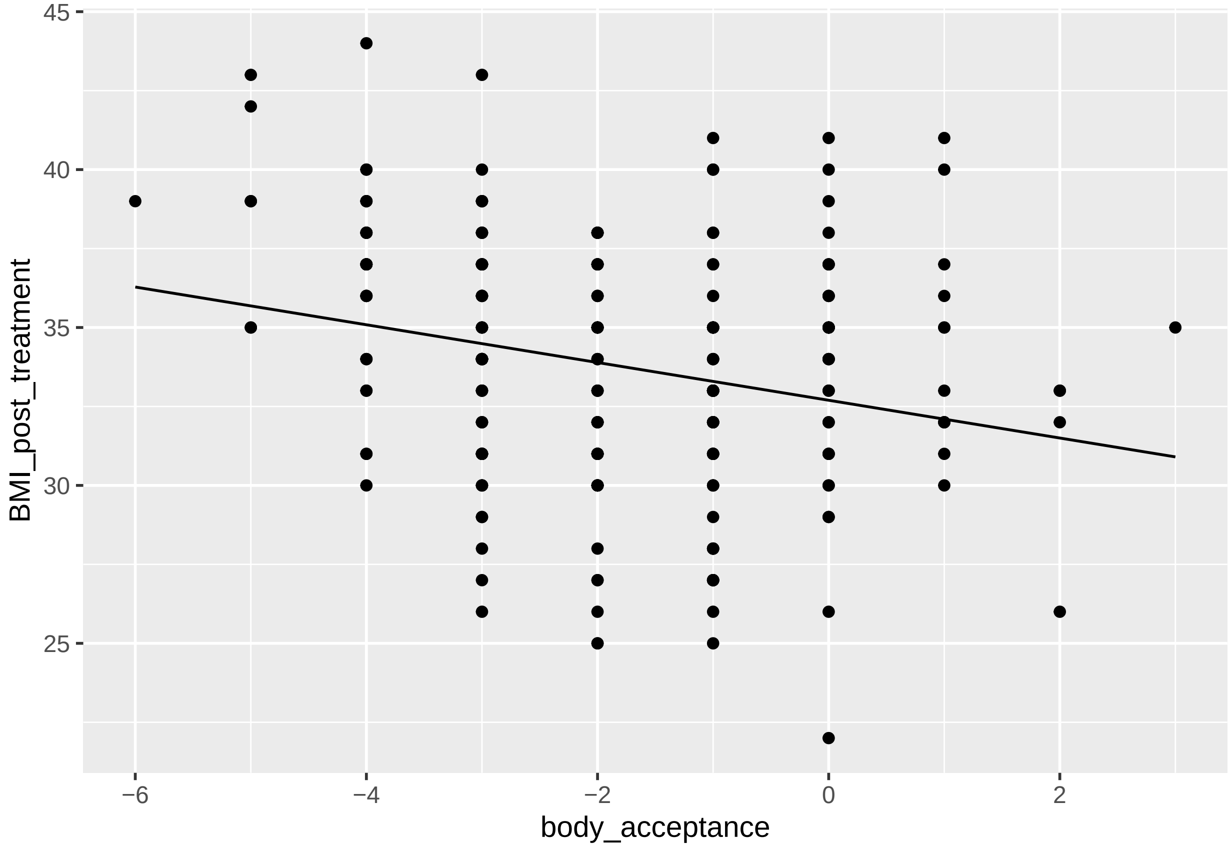
```
##
## Call:
## lm(formula = BMI_post_treatment ~ body_acceptance + I(body_acceptance^2) +
##     I(body_acceptance^3), data = data_weightloss)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0437  -2.0752   0.1402   2.1689   8.9924
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          33.04373    0.40618  81.352   <2e-16 ***
## body_acceptance       0.36393    0.27639   1.317    0.189
## I(body_acceptance^2)  0.11268    0.14740   0.764    0.445
## I(body_acceptance^3) -0.03858    0.02815  -1.370    0.172
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.577 on 236 degrees of freedom
## Multiple R-squared:  0.1321, Adjusted R-squared:  0.1211
## F-statistic: 11.98 on 3 and 236 DF,  p-value: 2.513e-07
```

```
AIC(mod_6)
```

```
## [1] 1298.83
```

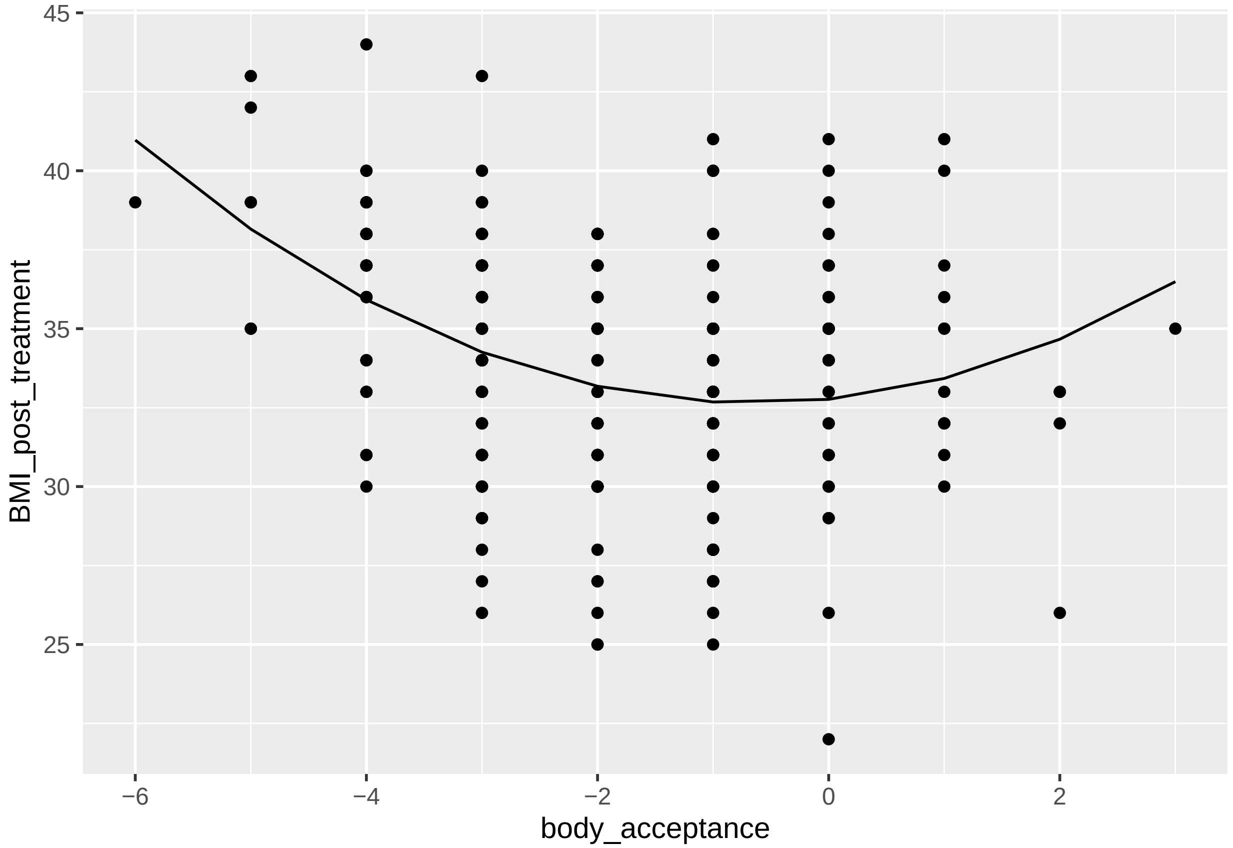
Here is the **regression line** for the model with the **first order term only**:

```
data_weightloss = data_weightloss %>%
  mutate(pred_mod_4 = predict(mod_4),
         pred_mod_5 = predict(mod_5),
         pred_mod_6 = predict(mod_6))

data_weightloss %>%
  ggplot() +
  aes(y = BMI_post_treatment, x = body_acceptance) +
  geom_point() +
  geom_line(aes(y = pred_mod_4))
```



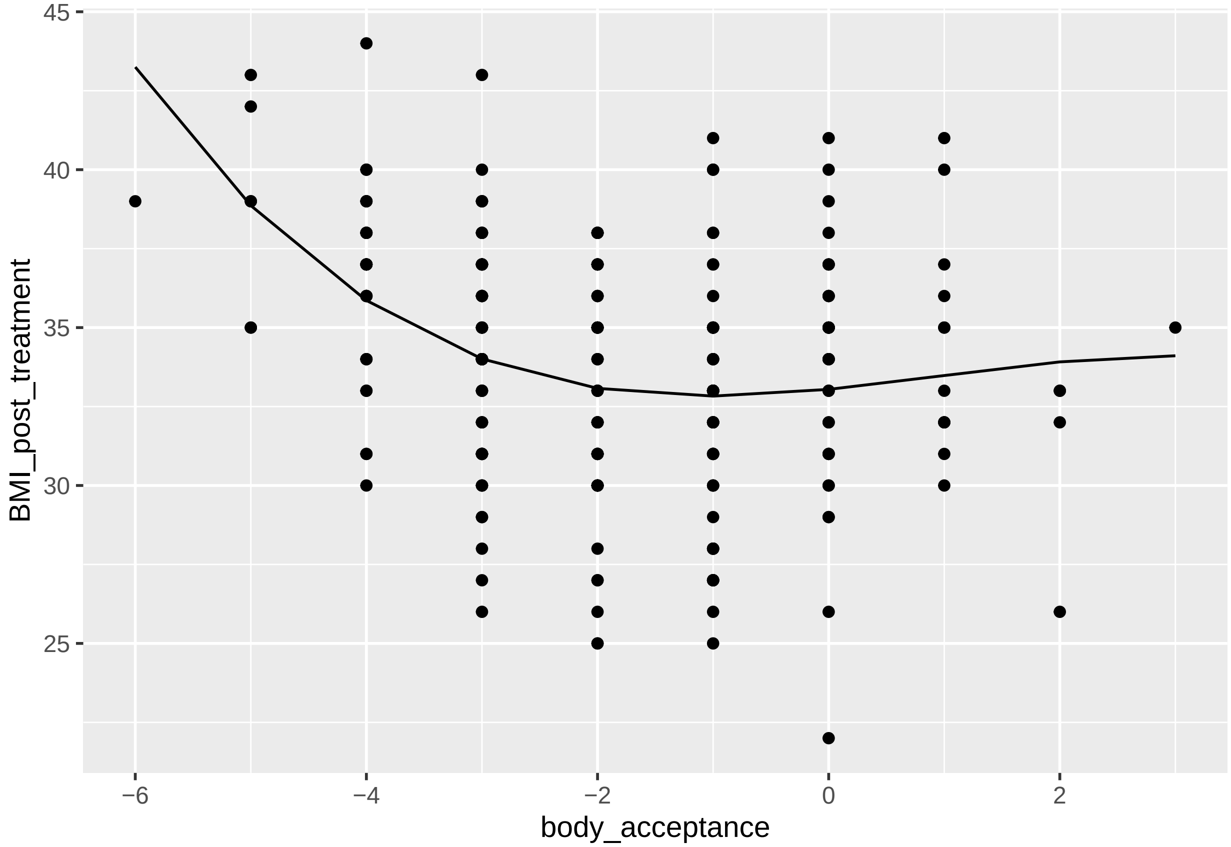
The model with the **first and the second order term**:

```
data_weightloss %>%
  ggplot() +
  aes(y = BMI_post_treatment, x = body_acceptance) +
  geom_point() +
  geom_line(aes(y = pred_mod_5))
```



The model with the **first, second, and third order terms**:

```
data_weightloss %>%
  ggplot() +
  aes(y = BMI_post_treatment, x = body_acceptance) +
  geom_point() +
  geom_line(aes(y = pred_mod_6))
```



It is apparent from the graphs that **the more higher order terms we include, the more flexibility we allow** for the regression line. Specifically, the we allow for one less inflection points for the line.

However, too much flexibility can be bad for our models performance on new data. **The more flexibility we allow, the higher the chance for "overfitting" the model** to the dataset the model is trained on, which makes it less effective in correctly esitmating the outcome on new datasets from the same population. For this reason, we usually don't use higher order terms unless there is a good theoretical grounding for there to be a non-linear effect, and even then, we usually do not include hiher order terms than three.

_______________*Practice*________________

Open the house sale dataset from the previous exercise. Experiment with different models based on your theories about what could influence housing prices.

Try to increase the adjusted R^2 above 54%.

If you want to get access to the whole dataset or get ideas on which model works best, go to Kaggle, check out the top kernels, and download the data. https://www.kaggle.com/harlfoxem/housesalesprediction/activity

```
data_house = read.csv("https://bit.ly/2DpwKOr")

data_house = data_house %>%
  mutate(sqm_living = sqft_living * 0.09290304,
         sqm_lot = sqft_lot * 0.09290304,
         sqm_above = sqft_above * 0.09290304,
         sqm_basement = sqft_basement * 0.09290304,
         sqm_living15 = sqft_living15 * 0.09290304,
         sqm_lot15 = sqft_lot15 * 0.09290304,
         has_basement = factor(has_basement))
```