# Analysis of Covariance

The ASTA team

## Contents



# 1 The regression problem

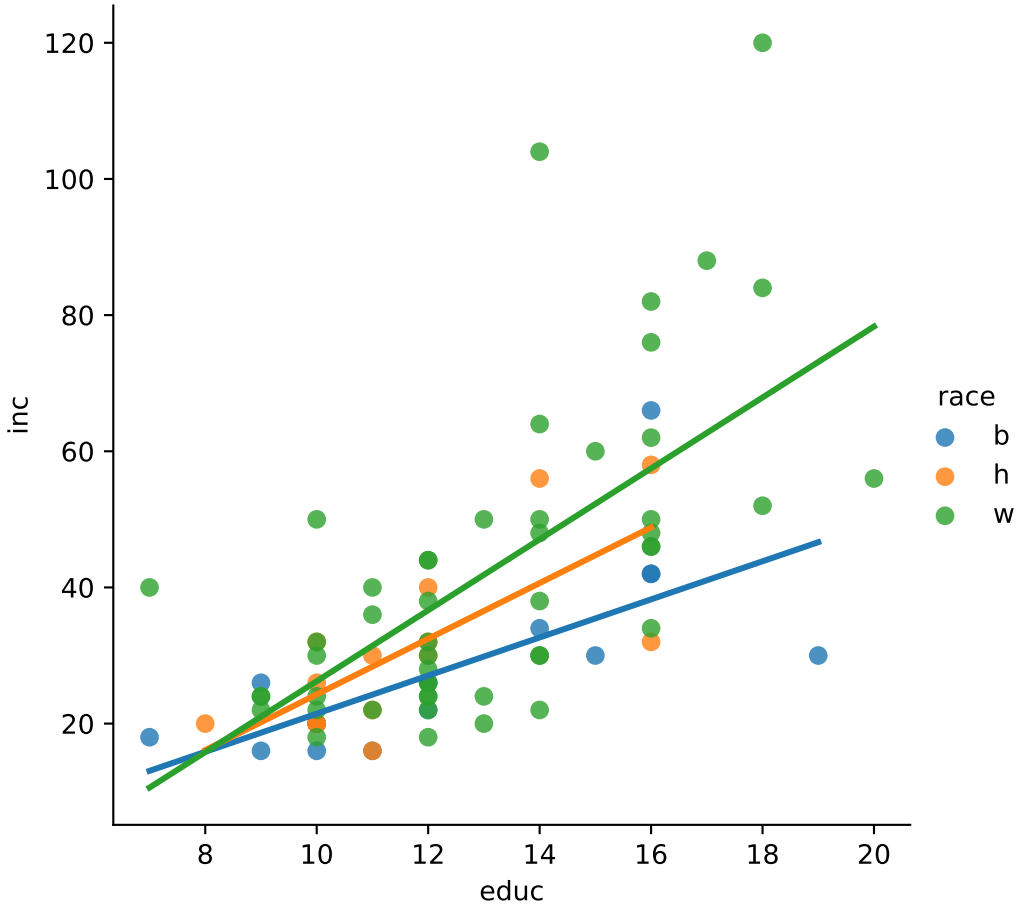
## 1.1 Example

- We will study the dataset in Agresti Table 13.1 available as `Income.txt` on the course website. We read in data in RStudio

```
import pandas as pd

Income = pd.read_csv("https://asta.math.aau.dk/datasets?file=Income.txt", sep='\t')
```

- We have a sample with measurements of 3 variables:
  - `y=income`: Quantitative variable, which is yearly income. This will be our response.
  - `x=education`: Quantitative predictor, which is the number of years of education.
  - `z=race`: Explanatory factor with levels `b`(black), `h`(hispanic) and `w`(white).
- We always start with some graphics:

```
import seaborn as sns

p = sns.lmplot(x='educ', y='inc', hue='race', data=Income, ci=None)
```




- An unclear picture, but a tendency to increasing income with increasing education.
- The trend lines for the three races are different. But is the difference significant? Or can the difference be explained by sampling variation?
- Such a regression with both qualitative and quantitative predictors is called an analysis of covariance (ANCOVA). When the model only contains qualitative predictors, the problem is known as analysis of variance (ANOVA) which is the topic of the next lecture.

# 2 Dummy coding

## 2.1 Dummy coding

- First, we will look at the model **without interaction**, i.e. the effect of `education` is the same for all races, which corresponds to parallel lines.
- We also have to introduce dummy coding of the factor $z$:
  - $z_1 = 1$ if `race=b` and zero otherwise
  - $z_2 = 1$ if `race=h` and zero otherwise

- This determines the regression model:

$$E(y|x,z) = \alpha + \beta x + \beta_1 z_1 + \beta_2 z_2$$

  which corresponds to **parallel** regressions lines for each race.

- `w`: $(z_1 = 0, z_2 = 0)$ $E(y|x) = \alpha + \beta x$
- `b`: $(z_1 = 1, z_2 = 0)$ $E(y|x) = \alpha + \beta_1 + \beta x$.
- `h`: $(z_1 = 0, z_2 = 1)$ $E(y|x) = \alpha + \beta_2 + \beta x$.
- $\beta_1$ is the difference in `Intercept` between black and white.
- $\beta_2$ is the difference in `Intercept` between Hispanic and white.

## 2.2 Example

- We want to tell the software that we want `race` to be a factor (grouping variable) and we want `w` as reference level for race (default is lexicographical ordering, i.e. `(b, h, w)` and `b` would then be the reference):

```
Income['race'] = Income['race'].astype('category').cat.reorder_categories(
    ['w', 'b', 'h'],
    ordered = True
)
```

- Then we use `+` in the model formula to only have additive effects of `educ` and `race`, i.e. a model without interaction:

```
import statsmodels.formula.api as smf

model1 = smf.ols('inc ~ educ + race', data=Income).fit()
model1.summary(slim = True)
```

```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                    inc   R-squared:                       0.462
## Model:                            OLS   Adj. R-squared:                  0.441
## No. Observations:                  80   F-statistic:                     21.75
## Covariance Type:            nonrobust   Prob (F-statistic):           2.85e-10
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept    -15.6635      8.412     -1.862      0.066     -32.418       1.091
## race[T.b]    -10.8744      4.473     -2.431      0.017     -19.783      -1.966
## race[T.h]     -4.9338      4.763     -1.036      0.304     -14.421       4.553
## educ           4.4317      0.619      7.158      0.000       3.199       5.665
## ==============================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## """
```

- The common slope to `educ` is estimated to be $\widehat{\beta} = 4.4316685$, with corresponding p-value=$4.42 \times 10^{-10}$ which is significantly different from zero.
- There is a clear positive effect of `educ` on `income`.

- The estimate for `w`-intercept is $\widehat{\alpha} = -15.6635$, which isn't significantly different from zero if we test at level 5% (this test is not really of interest).
- **The difference** between `b`- and `w`-intercept (`raceb`) is $\widehat{\beta}_1 = -10.8744$, which is significant with p-value=1.74%.
- There is no significant difference between `h`- and `w`-intercept.

## 2.3 Example: Prediction equations

```
model1.summary(slim = True)
```

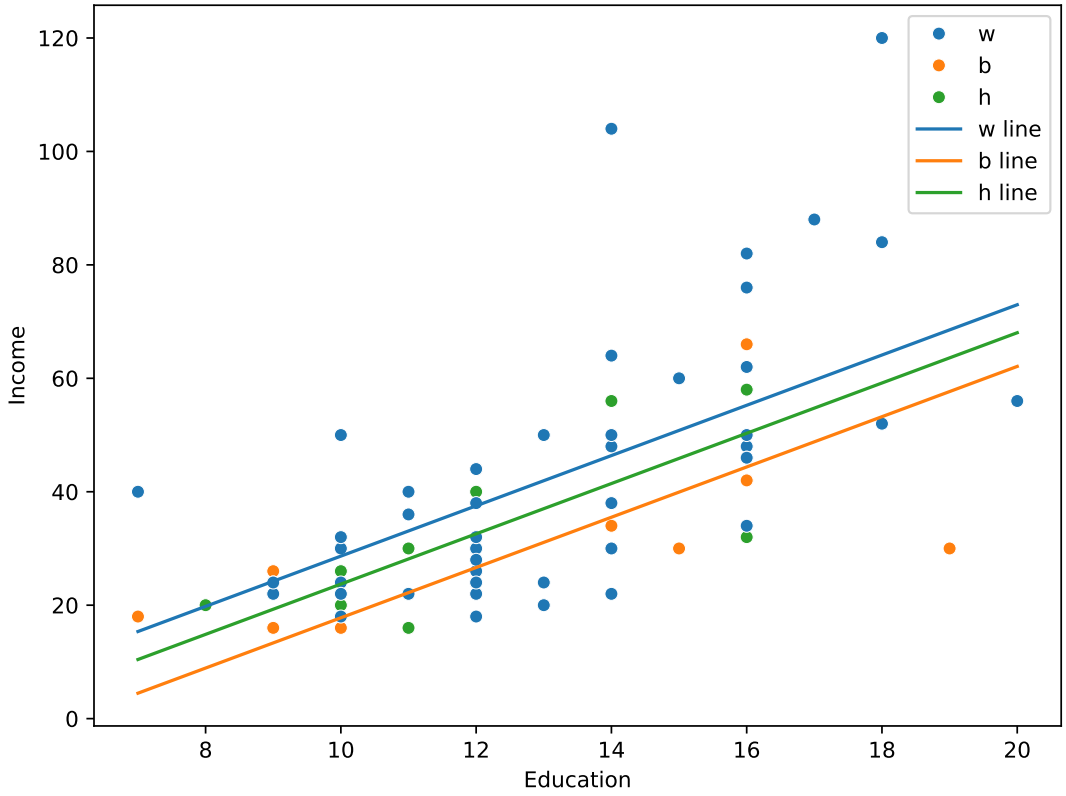
```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                    inc   R-squared:                       0.462
## Model:                            OLS   Adj. R-squared:                  0.441
## No. Observations:                  80   F-statistic:                     21.75
## Covariance Type:            nonrobust   Prob (F-statistic):           2.85e-10
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept    -15.6635      8.412     -1.862      0.066     -32.418       1.091
## race[T.b]    -10.8744      4.473     -2.431      0.017     -19.783      -1.966
## race[T.h]     -4.9338      4.763     -1.036      0.304     -14.421       4.553
## educ           4.4317      0.619      7.158      0.000       3.199       5.665
## ==============================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## """
```

- Reference/baseline group (white):
$$\widehat{y} = -15.66 + 4.43x$$
- Black:
$$\widehat{y} = -15.66 - 10.87 + 4.43x = -26.54 + 4.43x$$
- Hispanic:
$$\widehat{y} = -15.66 - 4.93 + 4.43x = -20.60 + 4.43x$$

## 2.4 Example: Plot

```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

coef = model1.params
educ_vals = np.linspace(Income['educ'].min(), Income['educ'].max(), 100)

plt.figure(figsize=(8,6))
sns.scatterplot(x='educ', y='inc', hue='race', data=Income)

for r in Income['race'].cat.categories:
    intercept = coef['Intercept'] + coef.get(f'race[T.{r}]', 0)
    slope = coef['educ']
```

```
    plt.plot(educ_vals, intercept + slope * educ_vals, label=f'{r} line')

plt.xlabel("Education")
plt.ylabel("Income")
plt.legend()
plt.show()
```



## 2.5 Agresti – summary

**TABLE 13.4:** Summary of Regression Equations and Parameters for Model with No Interaction, when Categorical Predictor Has Three Categories

| Category | $y$-Intercept | Slope | Mean $E(y)$ at Fixed $x$ | Difference From Mean of Category 3, Controlling for $x$ |
|---|---|---|---|---|
| 1 | $\alpha + \beta_1$ | $\beta$ | $(\alpha + \beta_1) + \beta x$ | $\beta_1$ |
| 2 | $\alpha + \beta_2$ | $\beta$ | $(\alpha + \beta_2) + \beta x$ | $\beta_2$ |
| 3 | $\alpha$ | $\beta$ | $\alpha + \beta x$ | 0 |

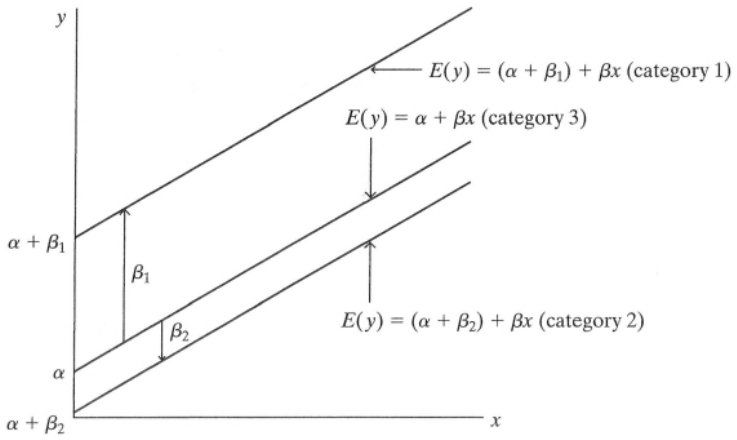



**FIGURE 13.5:** Graphic Portrayal of a Model with No Interaction, when the Categorical Predictor Has Three Categories

# 3 Model with interaction

## 3.1 Interaction

- In the following we will expand the model to include interaction between the effects of race and education on income. Before proceeding, let us recall what interaction means (and doesn't mean) in this context:
- Interaction between the effects of race and education on income does **not** mean that the values of education and race themselves are related or affect each other.
- Interaction between the effects of race and education on income means that the relationship between education and income depends on the value of race. I.e. for each fixed value of race the slope of the line relating education and income may have a different value.
- Often we just refer to this as "interaction between education and race" when it really should read "interaction between the effects of race and education on income".

## 3.2 Interaction

- We will expand the regression model, so we include interaction between $x$ and $z_1$ respectively $z_2$:

$$E(y|x,z) = \alpha + \beta x + \beta_1 z_1 + \beta_2 z_2 + \beta_3 z_1 x + \beta_4 z_2 x.$$

- This yields a regression line for each race:
- `w` ($z_1 = 0, z_2 = 0$): $E(y|x) = \alpha + \beta x$
- `b` ($z_1 = 1, z_2 = 0$): $E(y|x) = \alpha + \beta_1 + (\beta + \beta_3)x$.
- `h` ($z_1 = 0, z_2 = 1$): $E(y|x) = \alpha + \beta_2 + (\beta + \beta_4)x$.

- $\beta_1$ is **the difference** in `Intercept` between black and white, while $\beta_3$ is **the difference** in `slope` between black and white.
- $\beta_2$ is **the difference** in`Intercept` between Hispanic and white, while $\beta_4$ is the difference in `slope` between Hispanic and white.

## 3.3 Example: Prediction equations

- When we use `*` in the model formula we include interaction between `educ` and `race`:

```
model2 = smf.ols('inc ~ educ * race', data=Income).fit()
model2.summary(slim = True)
```

```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                    inc   R-squared:                       0.482
## Model:                            OLS   Adj. R-squared:                  0.448
## No. Observations:                  80   F-statistic:                     13.80
## Covariance Type:            nonrobust   Prob (F-statistic):           1.62e-09
## ===================================================================================
##                       coef    std err          t      P>|t|      [0.025      0.975]
## -----------------------------------------------------------------------------------
## Intercept        -25.8688     10.498     -2.464      0.016     -46.787      -4.951
## race[T.b]         19.3333     18.293      1.057      0.294     -17.116      55.782
## race[T.h]          9.2640     24.280      0.382      0.704     -39.114      57.642
## educ               5.2095      0.783      6.655      0.000       3.650       6.769
## educ:race[T.b]    -2.4107      1.418     -1.700      0.093      -5.236       0.414
## educ:race[T.h]    -1.1208      2.006     -0.559      0.578      -5.118       2.876
## ===================================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## """
```

- Reference/baseline group (white):

$$\widehat{y} = -25.87 + 5.21x$$

- Black:

$$\widehat{y} = -25.87 + 19.33 + (5.21 - 2.41)x = -6.54 + 2.80x$$

- Hispanic:

$$\widehat{y} = -25.87 + 9.26 + (5.21 - 1.12)x = -16.60 + 4.09x$$

## 3.4 Example: Individual tests

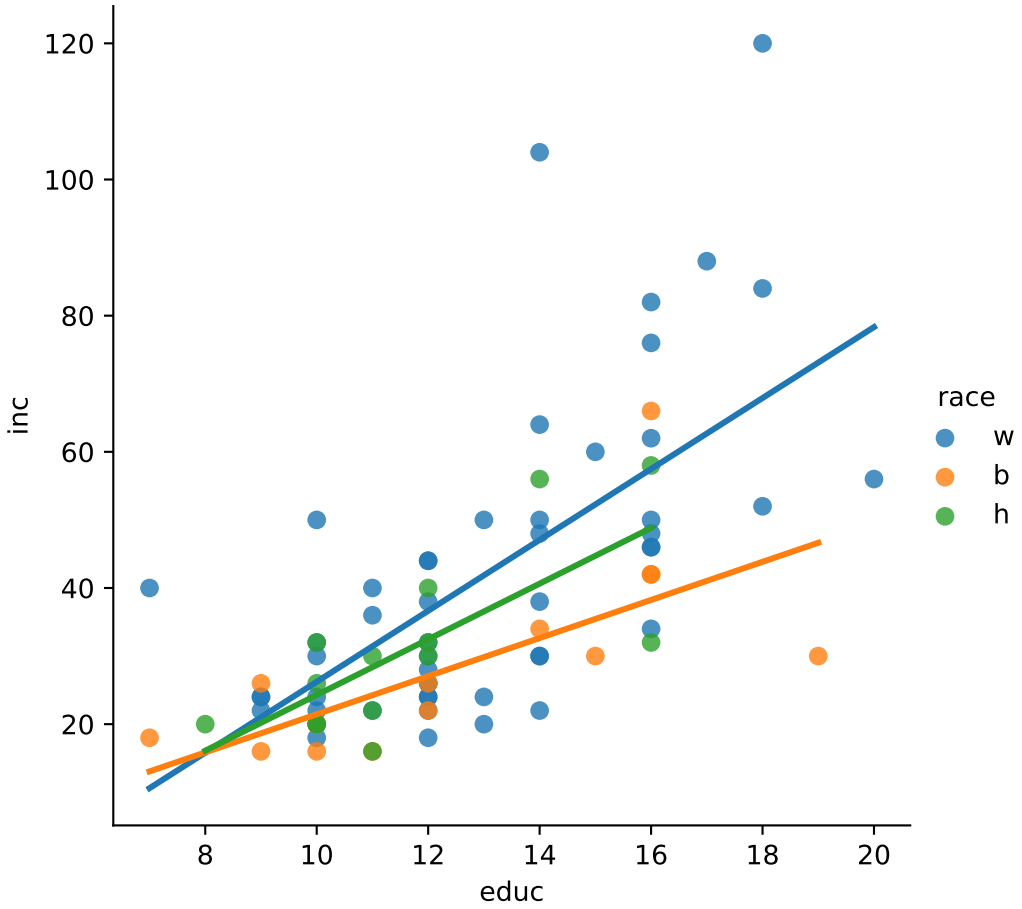
```
model2.summary(slim = True)
```

```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                    inc   R-squared:                       0.482
## Model:                            OLS   Adj. R-squared:                  0.448
## No. Observations:                  80   F-statistic:                     13.80
## Covariance Type:            nonrobust   Prob (F-statistic):           1.62e-09
```

```
## ==============================================================================
##                      coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept        -25.8688     10.498     -2.464      0.016     -46.787      -4.951
## race[T.b]         19.3333     18.293      1.057      0.294     -17.116      55.782
## race[T.h]          9.2640     24.280      0.382      0.704     -39.114      57.642
## educ               5.2095      0.783      6.655      0.000       3.650       6.769
## educ:race[T.b]    -2.4107      1.418     -1.700      0.093      -5.236       0.414
## educ:race[T.h]    -1.1208      2.006     -0.559      0.578      -5.118       2.876
## ==============================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## """
```

- **The difference** in slope between `b` and `w` (`educ:raceb`) is estimated to $\widehat{\beta}_3 = -2.4107$. With p-value=9.33% there is no significant difference.
- Furthermore, there isn't any significant difference of slope between `h` and `w`. In other words there is probably not interaction between `educ` and `race`.

```
p = sns.lmplot(x='educ', y='inc', hue='race', data=Income, ci=None)
```



# 4 Test for no interaction

## 4.1 Test for no interaction

```
model1.rsquared
```

```
## np.float64(0.46199055232513464)
```

```
model2.rsquared
```

```
## np.float64(0.4824821580587537)
```

- Is `model2` significantly better than `model1`? I.e. is $R^2$ significantly higher for `model2`?

## 4.2 Hypothesis and test statistic

- The simpler `model1` is obtained from the more complicated `model2` by setting $\beta_3 = 0$ and $\beta_4 = 0$, so the null hypothesis "the simpler additive model describes data sufficiently well compared to the complicated interaction model" is really the simple mathematical hypothesis:

$$H_0 : \beta_3 = 0, \beta_4 = 0.$$

- We will look at the difference between $R^2$ for the two models, but as before (for multiple linear regression) we have to convert this to an $F$ statistic which we can then calculate a $p$-value for.
- Formula for $F_{obs}$ (no need to learn this by heart):

$$F_{obs} = \frac{(R_2^2 - R_1^2)/(\mathrm{df}_1 - \mathrm{df}_2)}{(1 - R_2^2)/\mathrm{df}_2}$$

  where $\mathrm{df}_1$ and $\mathrm{df}_2$ are $n$ minus the number of model parameters for the two models (i.e. 80-4=76 and 80-6=74 in our case).
- The formula for $F_{obs}$ can be rewritten in terms of sums of squared errors (SSE) for each model (no need to memorize it):

$$F_{obs} = \frac{(SSE_1 - SSE_2)/(\mathrm{df}_1 - \mathrm{df}_2)}{(SSE_2)/\mathrm{df}_2}.$$

- In the literature SSE is sometimes denoted by RSS for **Residual Sums of Squares**; i.e SSE = RSS.

## 4.3 Test for no interaction in Python

- In Python the calculations are done using `anova_lm`:

```
from statsmodels.stats.anova import anova_lm

anova_lm(model1, model2)
```

```
##    df_resid           ssr  df_diff     ss_diff        F   Pr(>F)
## 0      76.0  18164.248072      0.0         NaN      NaN      NaN
## 1      74.0  17472.411504      2.0  691.836568  1.46505  0.23769
```

- The F-test for dropping the interaction `educ:race` has F-value=1.465, which in no way is significant with p-value=23.77%.

# 5 Hierarchy of models

## 5.1 Hierarchy of models

- `Interaction:` The most general model with main effects `educ` and `race` and interaction `educ:race`:

```
Interaction = smf.ols('inc ~ educ * race', data=Income).fit()
```

- `MainEffects`: The model where there are additive effects of `educ` and `race`.

```
MainEffects = smf.ols('inc ~ educ + race', data=Income).fit()
```

- `educEff`: Model where there only is an effect of `educ` (simple lin. reg.).

```
educEff = smf.ols('inc ~ educ', data=Income).fit()
```

- `raceEff`: Model where there only is an effect of `race` (a different mean for each group – more on this in the ANOVA lecture).

```
raceEff = smf.ols('inc ~ race', data=Income).fit()
```

- We can, corresponding to Agresti Table 13.10, make F-tests for 3 pairwise comparisons of models.

## 5.2 Example

- Comparing `MainEffects` and `Interaction` is what we have already done.

```
anova_lm(MainEffects, Interaction)
```

```
##    df_resid           ssr  df_diff     ss_diff        F   Pr(>F)
## 0      76.0  18164.248072      0.0         NaN      NaN      NaN
## 1      74.0  17472.411504      2.0  691.836568  1.46505  0.23769
```

- We recognize $F = 1.465$ with p-value=23.77%, i.e. `model2` isn't significantly better than `model1`. So no `educ:race` interaction.
- In the same manner we can compare `educEff` and `MainEffects`. I.e. we investigate whether the effect of `race` can be left out.

```
anova_lm(educEff, MainEffects)
```

```
##    df_resid           ssr  df_diff      ss_diff         F    Pr(>F)
## 0      78.0  19624.832018      0.0          NaN       NaN       NaN
## 1      76.0  18164.248072      2.0  1460.583947  3.055573  0.052922
```

- If any, the effect of `race` is weak with p-value=5.292%.
- Finally, we compare `raceEff` and `MainEffects`. Clearly `educ` cannot be left out (P-value=$4.422 \times 10^{-10}$).

```
anova_lm(raceEff, MainEffects)
```

```
##    df_resid           ssr  df_diff       ss_diff         F        Pr(>F)
## 0      77.0  30409.480000      0.0           NaN       NaN           NaN
## 1      76.0  18164.248072      1.0  12245.231928  51.23458  4.422192e-10
```

## 5.3 Example

- The methods generalize to models with more than 2 predictors.
- We return to the dataset `Ericksen`, where we study the response `crime`:

```
Ericksen = pd.read_csv("https://asta.math.aau.dk/datasets?file=Ericksen.txt", sep='\t')
model = smf.ols('crime ~ city * highschool + city * poverty', data=Ericksen).fit()
```

- The variables are:
  - `crime`: Quantitative variable
  - `city`: `city` or `state`
  - `highschool`: Quantitative variable

  - `poverty`: Quantitative variable
- The model has 3 predictors with main effects and includes
  - interaction between `city` and `highschool`
  - interaction between `city` and `poverty`.

```
model.summary(slim = True)
```

```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                   crime   R-squared:                       0.658
## Model:                             OLS   Adj. R-squared:                  0.629
## No. Observations:                   66   F-statistic:                     23.06
## Covariance Type:            nonrobust   Prob (F-statistic):           7.75e-13
## ================================================================================================
##                                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------------------------
## Intercept                     61.1456     18.125      3.373      0.001      24.889      97.402
## city[T.state]                 18.1526     20.413      0.889      0.377     -22.680      58.985
## highschool                    -1.5711      0.606     -2.592      0.012      -2.784      -0.358
## city[T.state]:highschool       0.7025      0.733      0.959      0.342      -0.763       2.168
## poverty                        5.3105      1.433      3.705      0.000       2.443       8.178
## city[T.state]:poverty         -5.1862      1.662     -3.121      0.003      -8.510      -1.862
## ================================================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## """
```

- There isn't significant (p-value=34.1523%) interaction between `city` and `highschool`.
- I.e. the effect of `highschool` on crime is the same in metropolitan areas (`city=city`) and the non-metropolitan areas (`city=state`).
- There is clearly (p-value=0.2773%) interaction between `city` and `poverty`.
- I.e. the effect of `poverty` on crime is different in metropolitan and non-metropolitan areas.
- For `city=state`, the effect of `poverty` (on crime) is smaller than in the major cities.
- Hence, poverty has larger effect on crime in the major cities than in the states outside the major cites.

## 5.4 Multicollinearity and variance inflation factors

- Ideally the predictors in linear regression should be **uncorrelated**, which is almost never the case.
- The consequence of the two predictors being correlated (**collinear**), is that the uncertainty of the parameter estimates increase (because the squared standard error increases) by a factor commonly called the variance inflation factor (VIF).
- If multiple pairs of predictors are collinear, we say that the model suffers from **multicollinearity**.
- If we have a model with $p$ predictors, then the VIF of $x_j$ is:

$$\text{VIF}_j = \frac{1}{1 - R_j^2},$$

  where $R_j^2$ is the multiple $R^2$ value of a model using $x_j$ as a response and the remaining $p-1$ predictors as explanatory variables.
- The larger $\text{VIF}_j$ is, the higher the collinearity between $x_j$ and the remaining predictors is.
- **Rule of thumb:** If a VIF is larger than 10 the collinearity is too high.