ASTA

The ASTA team

Contents

1 The regression problem

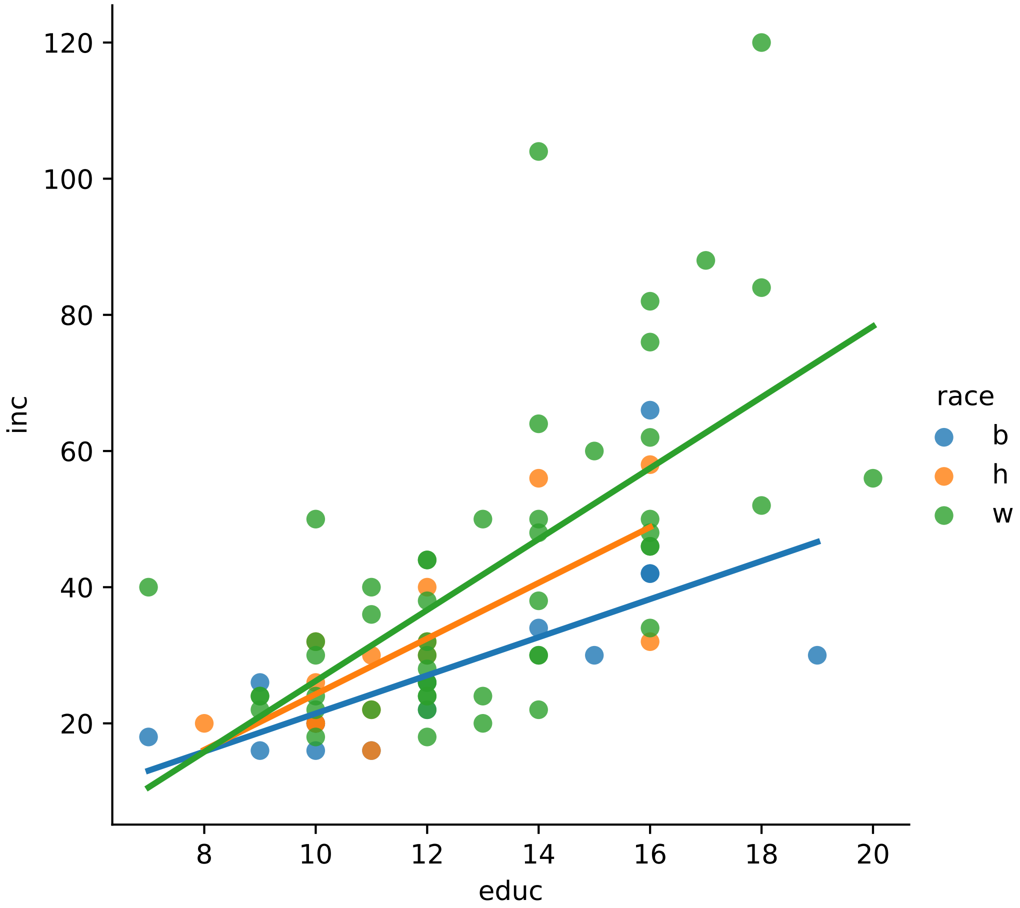
1.1 Example

- 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: $(z_1 = 0, z_2 = 1)$ $E(y|x) = \alpha + \beta_2 + \beta x$.
- β_1 is the difference in `Intercept` between black and white.
- β_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=4.42×10^{-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

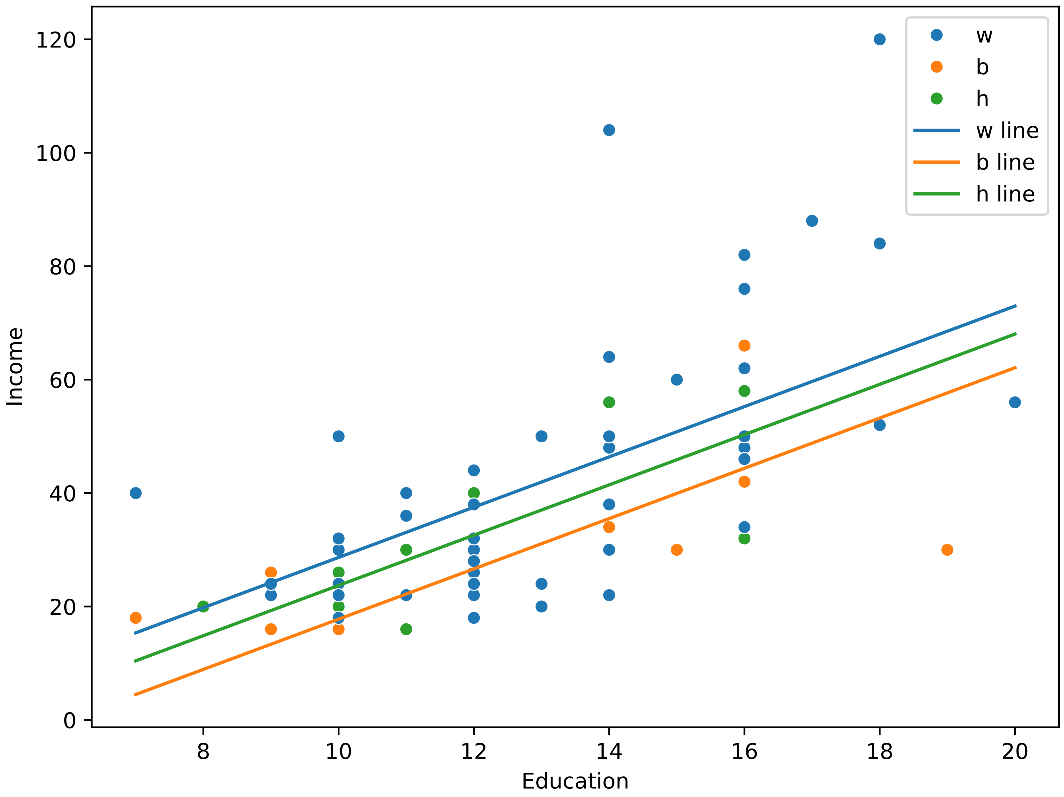
```
model1.summary(slim = True)
```

```
## <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

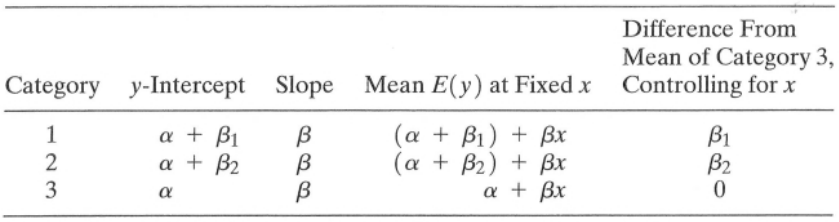
```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

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()
```



120
100
80
60
40
20
0
Income
8
10
12
14
16
18
20
Education
w
b
h
w line
b line
h line

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	$\alpha + \beta_1$	β	$(\alpha + \beta_1) + \beta x$	β_1
2	$\alpha + \beta_2$	β	$(\alpha + \beta_2) + \beta x$	β_2
3	α	β	$\alpha + \beta x$	0

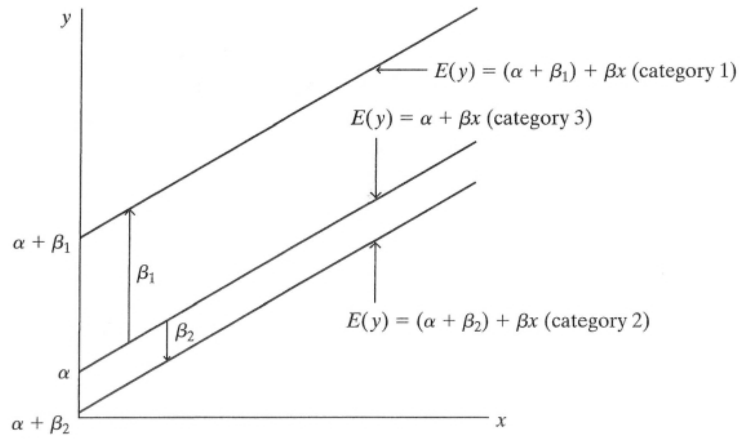


FIGURE 13.5: Graphic Portrayal of a Model with No Interaction, when the Categorical Predictor Has Three Categories

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` ($z_1 = 0, z_2 = 1$): $E(y|x) = \alpha + \beta_2 + (\beta + \beta_4)x$.

- β_1 is **the difference** in `Intercept` between black and white, while β_3 is **the difference** in `slope` between black and white.
- β_2 is **the difference** in`Intercept` between Hispanic and white, while β_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:
$$\widehat{y} = -25.87 + 9.26 + (5.21 - 1.12)x = -16.60 + 4.09x$$

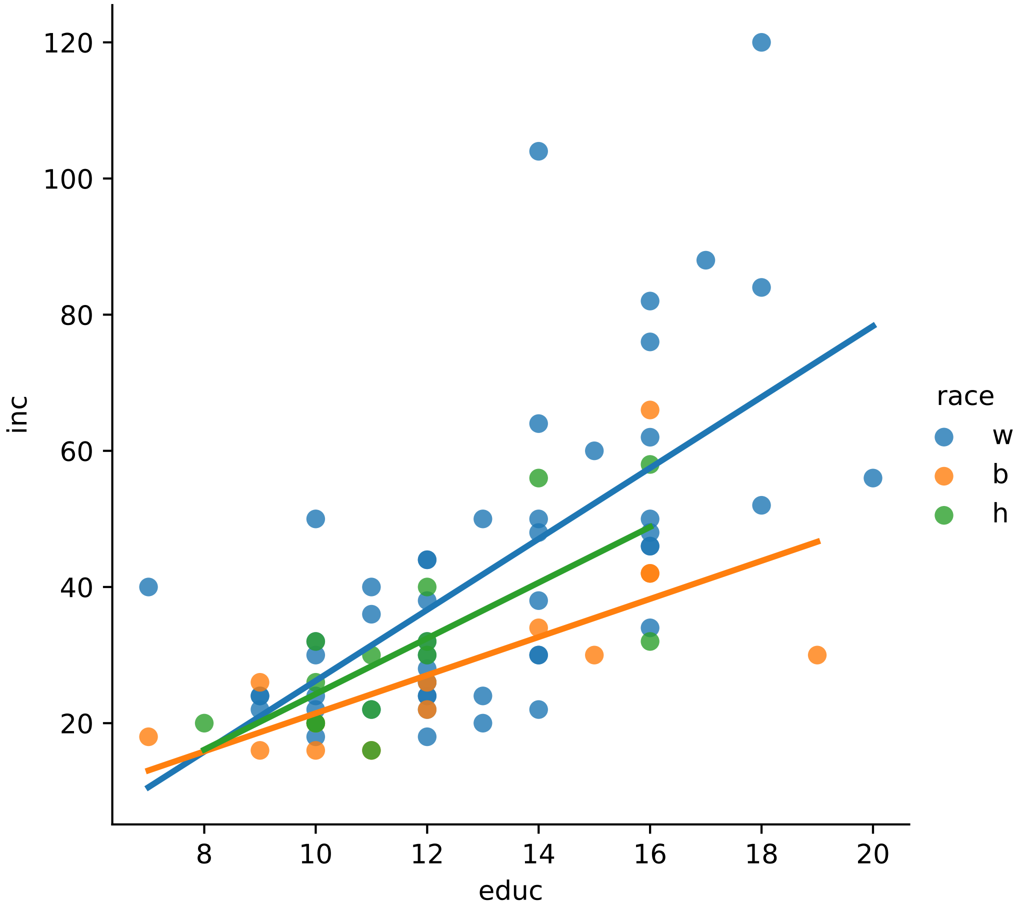
3.4 Example: Individual tests

```
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 df_1 and 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=4.422×10^{-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:

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

  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 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.

6 One way analysis of variance

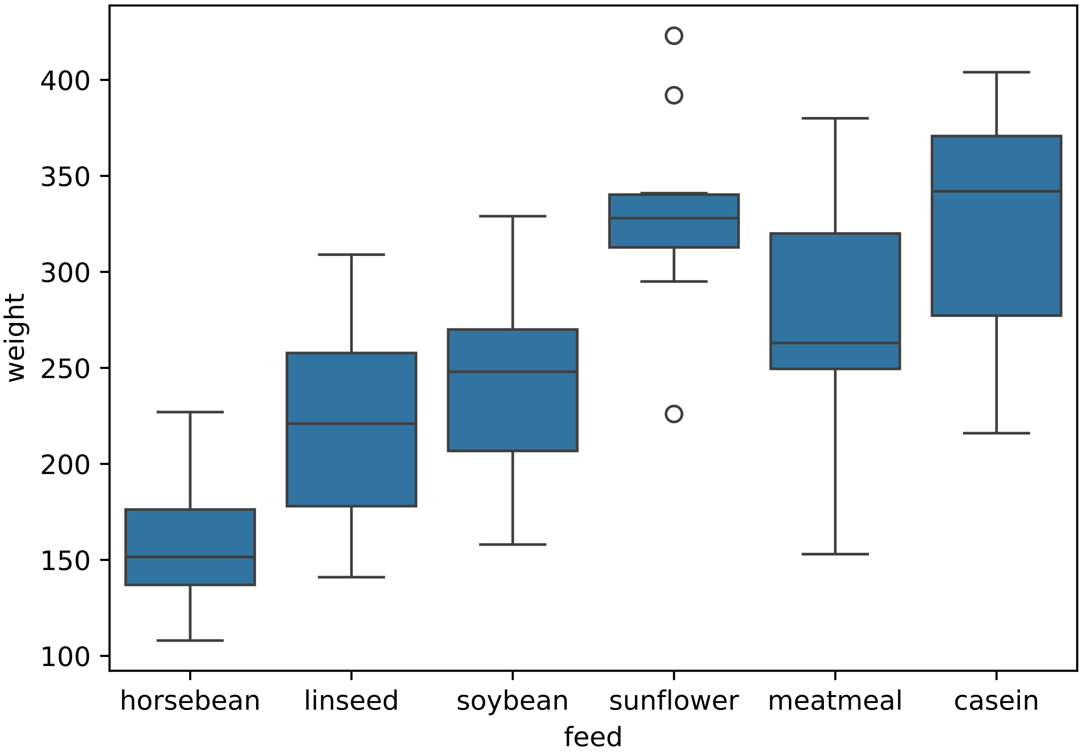
6.1 Example

- The data set `chickwts` is available and on the course webpage.
- 71 newly hatched chicks were randomly allocated into six groups, and each group was given a different feed supplement.
- Their weights in grams after six weeks are given along with feed types, i.e. we have a sample with corresponding measurements of 2 variables:
  - `weight`: a numeric variable giving the chick weight.
  - `feed`: a factor giving the feed type.
- Always start with some graphics:

```
import pandas as pd

chickwts = pd.read_csv("https://asta.math.aau.dk/datasets?file=chickwts.txt", sep='\t')
chickwts.head(3)
```

```
##    weight       feed
## 0     179  horsebean
## 1     160  horsebean
## 2     136  horsebean
```

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

p = sns.boxplot(x='feed', y='weight', data=chickwts)
```



6.2 The ANOVA Model

- We measure the response y which in this case is `weight`.
- We want to study the effect of the factor x on y. In this case $x =$`feed` and divides the sample in $g = 6$ groups.
- The mean responses within the groups are denoted $\mu_1, \mu_2, \ldots, \mu_g$.
- We will assume that
  - $y = \mu_x + \epsilon$, when y is a response in group x
  - ϵ are a sample from a population with mean zero and standard deviation σ.
  - The standard deviation for the population in each group is the same and equals σ
  - The response variable, y, is normal distributed within each group.
- The ANOVA test is a *test of equal means* for the different groups.

7 Estimation of mean values

7.1 Estimates

- Least squares estimates for population means $\widehat{\mu}_x$ is given by the average of the response measurements in group x.
- For a given measured response y we let $\widehat{y}$ denote the model's prediction of y, i.e.

$$\widehat{y} = \widehat{\mu}_x$$

  if y is a response for an observation in group x.
- We use `mean` to find the mean, for each group:

```
chickwts.groupby('feed')['weight'].mean()
```

```
## feed
## casein       323.583333
## horsebean    160.200000
## linseed      218.750000
## meatmeal     276.909091
## soybean      246.428571
## sunflower    328.916667
## Name: weight, dtype: float64
```

- We can e.g. see that $\widehat{y} = 323.6$, when `feed=casein` but $\widehat{y} = 160.2$, when `feed=horsebean`.
- Is it a significant difference ?

7.2 Contrast coding

- In many cases there is a group corresponding to "no treatment" and we are interested in the effect of different treatments.
- In this example we only have different `feeds`, which are sorted in lexicographical order by `R`, so `casein` is the reference.
- We can specify the model via:
  - `Intercept` corresponding to the mean response for the reference (`casein`).
  - For each of the other groups we have a **contrast**, which measures **the difference** between the mean value for that group and the reference group.
- For a given contrast we can calculate standard error, t-score and p-value, and thereby investigate whether there is a difference between this group and the reference group.
- In Agresti this is referred to as using **dummy variables**.

7.3 Example

```
import statsmodels.formula.api as smf

model = smf.ols('weight ~ feed', data=chickwts).fit()
model.summary(slim = True)
```

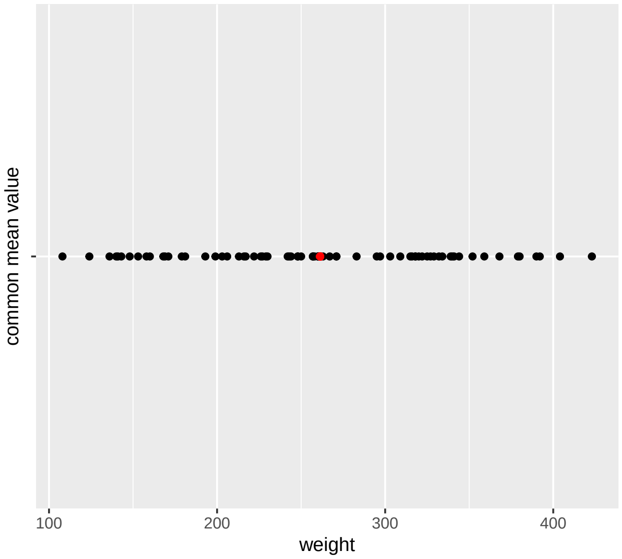
```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                 weight   R-squared:                       0.542
## Model:                            OLS   Adj. R-squared:                  0.506
## No. Observations:                  71   F-statistic:                     15.36
## Covariance Type:            nonrobust   Prob (F-statistic):           5.94e-10
## =====================================================================================
##                         coef    std err          t      P>|t|      [0.025      0.975]
## -------------------------------------------------------------------------------------
## Intercept           323.5833     15.834     20.436      0.000     291.961     355.206
## feed[T.horsebean]  -163.3833     23.485     -6.957      0.000    -210.287    -116.480
## feed[T.linseed]    -104.8833     22.393     -4.682      0.000    -149.554     -60.112
## feed[T.meatmeal]    -46.6742     22.896     -2.039      0.046     -92.400      -0.948
## feed[T.soybean]     -77.1548     21.578     -3.576      0.001    -120.249     -34.061
## feed[T.sunflower]     5.3333     22.393      0.238      0.812     -39.388      50.054
## =====================================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## """
```

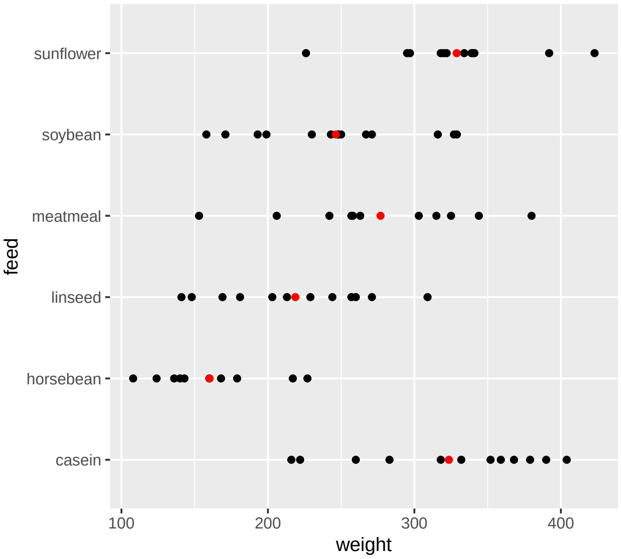
- We get information about contrasts and their significance:
- `Intercept` corresponding to `casein` has `weight` different from zero ($p < 2 \times 10^{-16}$) (of course, chickens grow a lot over 6 weeks)
- Weight difference between `casein` and `horsebean` is extremely significant (p=2×10^{-9}).
- There is no significant weight difference between `casein` and `sunflower` (p=81%).

8 Overall test for effect

8.1 Graphical representation of models

- We have two alternative explanations of the data.
- Simple model with one parameter (mean): “The feed type doesn’t matter. The weight is just random around a common mean value”.
- Complex model with six parameters (means): “The feed type is important. For each feed type we get a different mean value and the weights are random around these values.”





8.2 Hypotheses and test statistic

- Is the complex model significantly better (i.e. is there any effect of the explanatory grouping variable)? We can write the corresponding hypotheses in two different ways

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_g \quad \text{against} \quad H_a : \text{ At least 2 of the population means are different}$$

- Alternatively

$$H_0 : \text{ All contrasts are equal to zero.} \qquad H_a : \text{ At least one contrast is non-zero.}$$

- We will (indirectly) use R^2 to do the test. If it is large, the complex model has good predictive power compared to the simple model. To judge significance we use

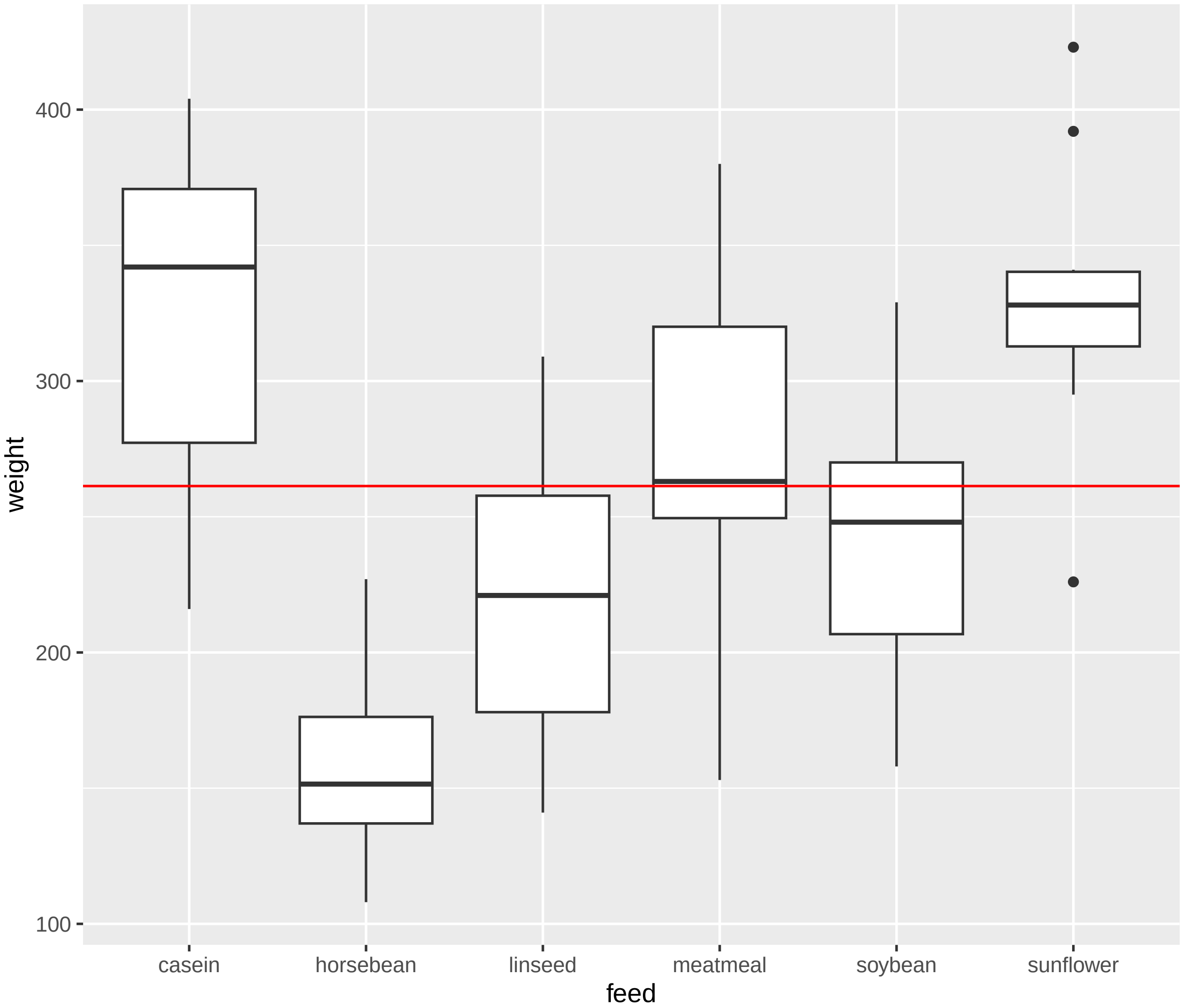
$$F_{obs} = \frac{(n-g)R^2}{(g-1)(1-R^2)} = \frac{(TSS - SSE)/(g-1)}{SSE/(n-g)}.$$

- Large values of R^2 implies large values of F_{obs}, which points to the alternative hypothesis.
- I.e. when we have calculated the observed value F_{obs}, then we have to find the probability that a new experiment would result in a larger value.
- TSS: error sum of squares if common mean. SSE: error sum of squares if different means.
- TSS-SSE: how much does error sum of squares increase if means are restricted to be equal.

8.3 Interpretation of F statistic - Variance between/within groups

- It can be shown that the numerator of F_{obs} is a measure of **the variance between the groups**, i.e. how much “boxes” vary around the total average (the red line).

- Likewise it can be shown the denominator of F_{obs} is a measure for **the variance within groups**, i.e. how “tall” the boxes in the boxplot are.



- The bigger deviations between the red line and the box means relative to the variation within boxes, the less we trust H_0. This is measured by the F-test statistic, which can be stated as

$$F_{obs} = \frac{\text{variance between groups}}{\text{variance within groups}}$$

8.4 Example

```
import statsmodels.formula.api as smf

model = smf.ols('weight ~ feed', data=chickwts).fit() # same as earlier
model.summary(slim = True)
```

```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                 weight   R-squared:                       0.542
## Model:                            OLS   Adj. R-squared:                  0.506
## No. Observations:                  71   F-statistic:                     15.36
## Covariance Type:            nonrobust   Prob (F-statistic):           5.94e-10
## ==================================================================================
##                           coef    std err          t      P>|t|      [0.025      0.975]
## ----------------------------------------------------------------------------------
## Intercept            323.5833     15.834     20.436      0.000     291.961     355.206
## feed[T.horsebean]   -163.3833     23.485     -6.957      0.000    -210.287    -116.480
## feed[T.linseed]     -104.8333     22.393     -4.682      0.000    -149.554     -60.112
## feed[T.meatmeal]     -46.6742     22.896     -2.039      0.046     -92.400      -0.948
## feed[T.soybean]      -77.1548     21.578     -3.576      0.001    -120.249     -34.061
## feed[T.sunflower]      5.3333     22.393      0.238      0.812     -39.388      50.054
## ==================================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## """
```

- The `F-statistic` gives us the value of $F_{obs} = 15.36$ and the corresponding p-value (5.9×10^{-10}). Clearly there is a significant difference between the types of `feed`.

9 Two way analysis of variance

9.1 Additive effects

- The data set `ToothGrowth` is available on the webpage.
- The data describes the tooth length in guinea pigs where some receive vitamin C treatment and others are given orange juice in different dosage.

```
ToothGrowth = pd.read_csv("https://asta.math.aau.dk/datasets?file=ToothGrowth.txt", sep='\t')
ToothGrowth['dose'] = pd.Categorical(
    ToothGrowth['dose'].map({0.5: 'LO', 1: 'ME', 2: 'HI'}),
    categories=['LO', 'ME', 'HI'],
    ordered=True
)
ToothGrowth.head(3)
```

```
##     len supp dose
## 0   4.2   VC   LO
## 1  11.5   VC   LO
## 2   7.3   VC   LO
```

- A total of 60 observations on 3 variables.
  - `len` The tooth length
  - `supp` The type of the supplement (`OJ` or `VC`)
  - `dose` The dosage (`LO`, `ME`, `HI`)
- We will study the response `len` with the predictors `supp` and `dose`.
- At first we look at the model with additive effects
  - `len`=μ + `"effect of supp"`+ `"effect of dose"` + `error`
- This is also called the main effects model since it does not contain interaction terms.
- The parameter μ corresponds to the `Intercept` and is the mean tooth length in the reference group (supp `OJ`, dose `LO`).
- The effect of `supp` is the difference in mean when changing from `OJ` to `VC`.
- The effect of `dose` is the difference in mean when changing from `LO` to either`ME` or `HI`.

9.2 Dummy coding

- Let us introduce dummy variables:
  - $s_C = 1$ if supp `VC` and zero otherwise.

  - $d_M = 1$ if dose is `ME` and zero otherwise.
  - $d_H = 1$ if dose is `HI` and zero otherwise.
- Then we state the model

$$\text{length} = \mu + \beta_1 s_C + \beta_2 d_M + \beta_3 d_H + \text{error}.$$

- Interpretation:
  - μ is the expected tooth length when supp is `OJ` and `dose` is `LO` ($s_C = d_M = d_H = 0$)).
  - β_1 is the effect of supplement `OJ` to `VC` ($s_C = 1$).
  - β_2 is the effect of increasing dosage from `LO` to `ME` ($d_M = 1$).
  - β_3 is the effect of increasing dosage from `LO` to `HI` ($d_H = 1$).
- As a two-way table:

	LO	ME	HI
OJ	μ	$\mu + \beta_2$	$\mu + \beta_3$
VC	$\mu + \beta_1$	$\mu + \beta_1 + \beta_2$	$\mu + \beta_1 + \beta_3$

9.3 Main effect model in R

- The main effects model is fitted by

```
MainEff = smf.ols('len ~ supp + dose', data=ToothGrowth).fit()
MainEff.summary(slim = True)
```

```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                    len   R-squared:                       0.762
## Model:                            OLS   Adj. R-squared:                  0.750
## No. Observations:                  60   F-statistic:                     59.88
## Covariance Type:            nonrobust   Prob (F-statistic):           1.78e-17
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept     12.4550      0.988     12.603      0.000      10.475      14.435
## supp[T.VC]    -3.7000      0.988     -3.744      0.000      -5.680      -1.720
## dose[T.ME]     9.1300      1.210      7.543      0.000       6.705      11.555
## dose[T.HI]    15.4950      1.210     12.802      0.000      13.070      17.920
## ==============================================================================
## 
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## """
```

- The model has 4 parameters.
- The F test at the end compares with the (null) model with only one overall mean parameter.

9.4 Testing effect of supp

- Alternative model without effect of supp:

```
doseEff = smf.ols('len ~ dose', data=ToothGrowth).fit()
```

- We can compare R^2 to see if `doseEff` (Model 1) is sufficient to explain the data compared to `MainEff`

(Model 2). This is done by converting to F-statistic:

$$F_{obs} = \frac{(R_2^2 - R_1^2)/(df_1 - df_2)}{(1 - R_2^2)/df_2} = \frac{(SSE_1 - SSE_2)/(df_1 - df_2)}{(SSE_2)/df_2}.$$

- $SSE_1 - SSE_2$: increase in error sum of square when using Model 1 instead of Model 2
- In **R** the calculations are done using `anova`:

```
from statsmodels.stats.anova import anova_lm

anova_lm(doseEff, MainEff)
```

```
##    df_resid       ssr  df_diff  ss_diff          F    Pr(>F)
## 0      57.0  1025.775      0.0      NaN        NaN       NaN
## 1      56.0   820.425      1.0   205.35  14.016638  0.000429
```

- p-value is 0.0004 hence we reject that `supp` does not have an effect. Thus we prefer Model 2 (`MainEff`).

9.5 Testing effect of dose

- Alternative model without effect of dose:

```
suppEff = smf.ols('len ~ supp', data=ToothGrowth).fit()
anova_lm(suppEff, MainEff)
```

```
##    df_resid          ssr  df_diff      ss_diff          F        Pr(>F)
## 0      58.0  3246.859333      0.0          NaN        NaN           NaN
## 1      56.0   820.425000      2.0  2426.434333  82.810935  1.871163e-17
```

- p-value is ≈ 0 hence we reject that `dose` does not have an effect. Thus we prefer Model 2 (`MainEff`).

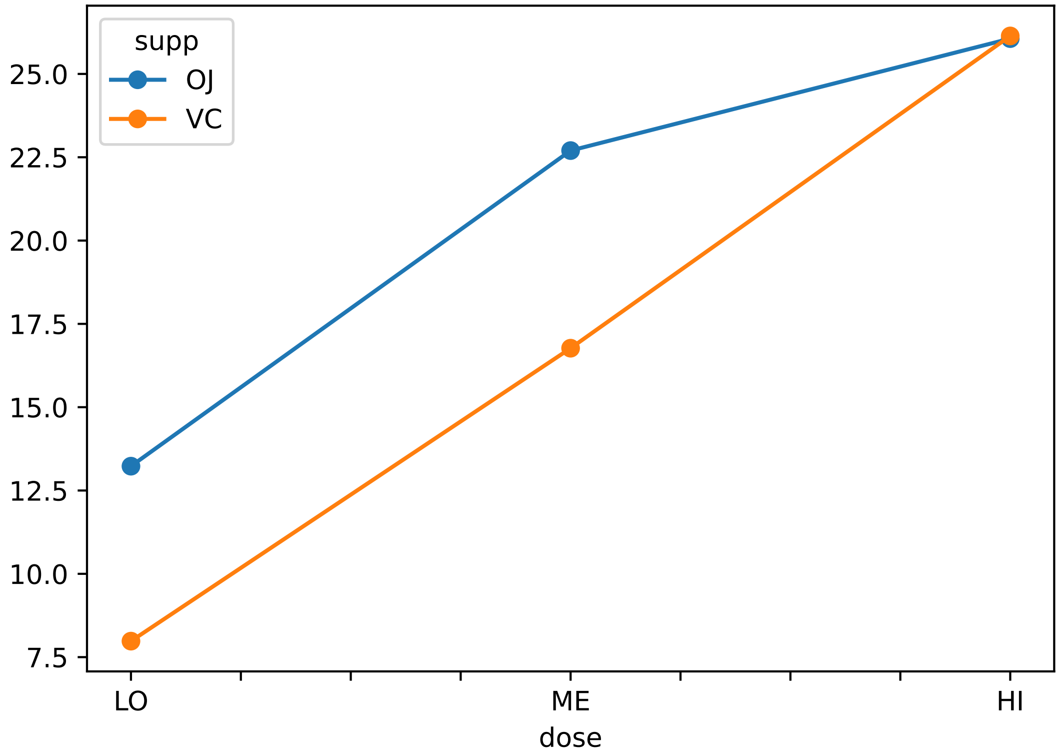
10 Interaction

10.1 Example

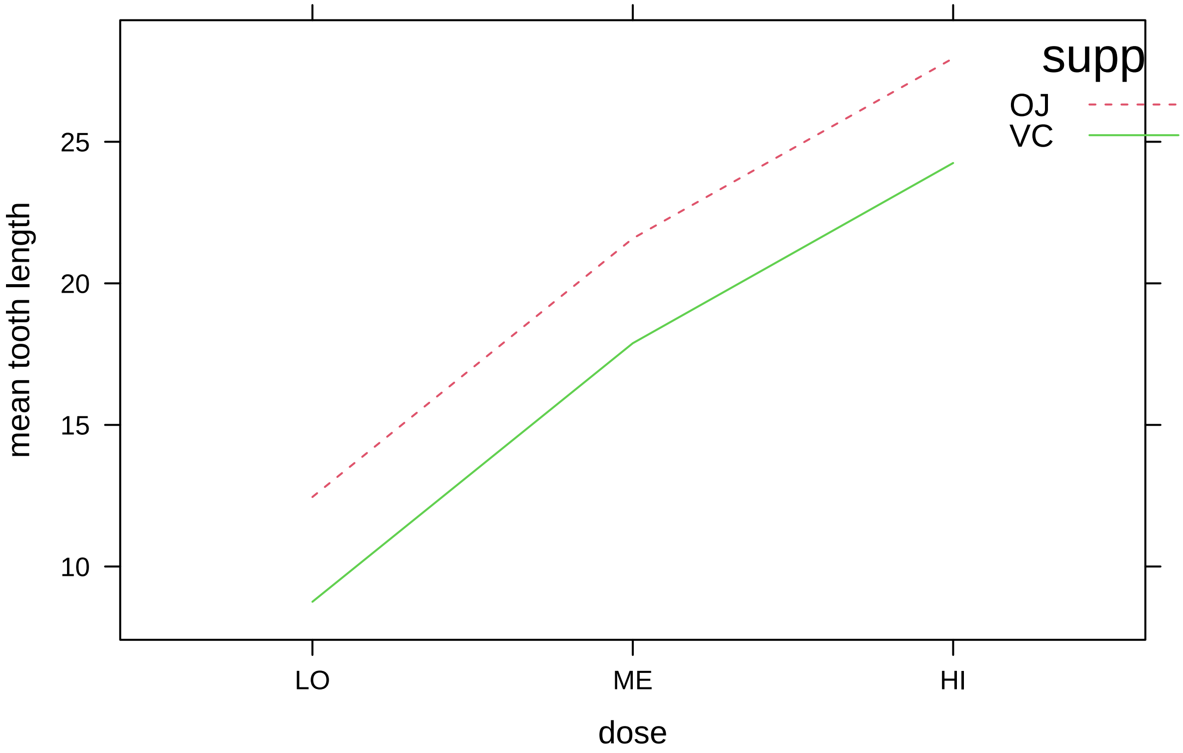
- We will extend the model by introducing an interaction between `supp` and `dose`.

- Interaction plot:

```
means = ToothGrowth.groupby(['dose', 'supp'], observed=False)['len'].mean().unstack()
means.plot(marker='o')
```



- For each of the supplement types we plot the average tooth length as a function of dosage.
- If the main effects model is correct then the difference between supplements is the same for all levels of dosage, i.e. the curves should be parallel - except for noise.
- This does not seem to be the case.
- This is how the plot *should* look *if* the main effects model (no interaction) is correct:



- Parallel lines mean that effect of supplement does not depend on dose !

10.2 Dummy coding

- The extended model can be formulated as

$$\texttt{length} = \mu + \beta_1 s_C + \beta_2 d_M + \beta_3 d_H + \beta_4 s_C d_M + \beta_5 s_C d_H + \texttt{error}$$

- Interpretation:
  - μ is the expected tooth length for `supp` `OJ` and `dose` `LO` ($s_C = d_M = d_H = 0$).
  - β_1 is the effect of changing from `supp` `OJ` to `VC`, `dose` is `LO` ($s_C = 1, d_M = d_H = 0$).
  - β_2 is the effect of increasing `dose` from `LO` to `ME`, when `supp` is `OJ` ($s_C = 0, d_M = 1$).
  - β_3 is the effect of increasing `dose` from `LO` to `HI`, when `supp` is `OJ` ($s_C = 0, d_H = 1$).
  - β_4 is an additional effect of both changing from `supp` `OJ` to `VC` and increasing `dose` from `LO` to `ME` ($s_C = 1, d_M = 1$)
  - β_5 is an additional effect of both changing from `supp` `OJ` to `VC` and increasing `dose` from `LO` to `HI` ($s_C = 1, d_H = 1$)
- As a two-way table:

$$\begin{array}{cccc} & LO & ME & HI \\ OJ & \mu & \mu + \beta_2 & \mu + \beta_3 \\ VC & \mu + \beta_1 & \mu + \beta_1 + \beta_2 + \beta_4 & \mu + \beta_1 + \beta_3 + \beta_5 \end{array}$$

- Further examples:
  - effect of changing from `supp` `OJ` to `VC` if `dose` is `LO` is $\mu + \beta_1 - \mu = \beta_1$
  - effect of changing from `supp` `OJ` to `VC` if `dose` is `ME` is $\mu + \beta_1 + \beta_2 + \beta_4 - \mu - \beta_2 = \beta_1 + \beta_4$
  - effect of changing from `supp` `OJ` to `VC` if `dose` is `HI` is $\mu + \beta_1 + \beta_3 + \beta_5 - \mu - \beta_3 = \beta_1 + \beta_5$
  - if $\beta_4 = 0$ and $\beta_5 = 0$ the effect of changing from `OJ` to `VC` does not depend on `dose`

10.3 Example

- We fit the interaction model by changing plus to multiply in the model expression from before:

```
Interaction = smf.ols('len ~ supp*dose', data=ToothGrowth).fit()
```

- Now we can think of an experiment with 6 groups corresponding to each combination of the predictors.
- Is added interaction significant ? - we compare main effects model and more complex interaction model using anova:

```
anova_lm(MainEff, Interaction)
```

```
##    df_resid      ssr  df_diff  ss_diff         F   Pr(>F)
## 0      56.0  820.425      0.0      NaN       NaN      NaN
## 1      54.0  712.106      2.0  108.319  4.106991  0.02186
```

- With a p-value of 2.186% there is a significant interaction `supp:dose`, i.e. the lack of parallel curves in the interaction plot is significant.

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

```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                    len   R-squared:                       0.794
## Model:                            OLS   Adj. R-squared:                  0.775
## No. Observations:                  60   F-statistic:                     41.56
## Covariance Type:            nonrobust   Prob (F-statistic):           2.50e-17
## =========================================================================================
##                             coef    std err          t      P>|t|      [0.025      0.975]
## -----------------------------------------------------------------------------------------
## Intercept                13.2300      1.148     11.521      0.000      10.928      15.532
## supp[T.VC]               -5.2500      1.624     -3.233      0.002      -8.506      -1.994
## dose[T.ME]                9.4700      1.624      5.831      0.000       6.214      12.726
## dose[T.HI]               12.8300      1.624      7.900      0.000       9.574      16.086
## supp[T.VC]:dose[T.ME]    -0.6800      2.297     -0.296      0.768      -5.285       3.925
## supp[T.VC]:dose[T.HI]     5.3300      2.297      2.321      0.024       0.725       9.935
## =========================================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## """
```

- Note the negative effect of changing from `OJ` to `VC` when `dose` is low is cancelled by the positive interaction parameter (β_5 for `suppVC:doseHI`) meaning almost no difference between `OJ` and `VC` when `dose` is high (compare with interaction plot)

10.4 Hierarchical principle

- In presence of interaction effect it does not make sense to make tests for absence of main effects ! Indeed each factor has an effect that just happens to vary depending on the other factor
- Hence start by investigating whether there is an interaction effect
- If yes: no further tests !
- If no: you may test main effects if relevant for your study