Analysis of Variance

The ASTA team

Contents

1 One way analysis of variance

1.1 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:

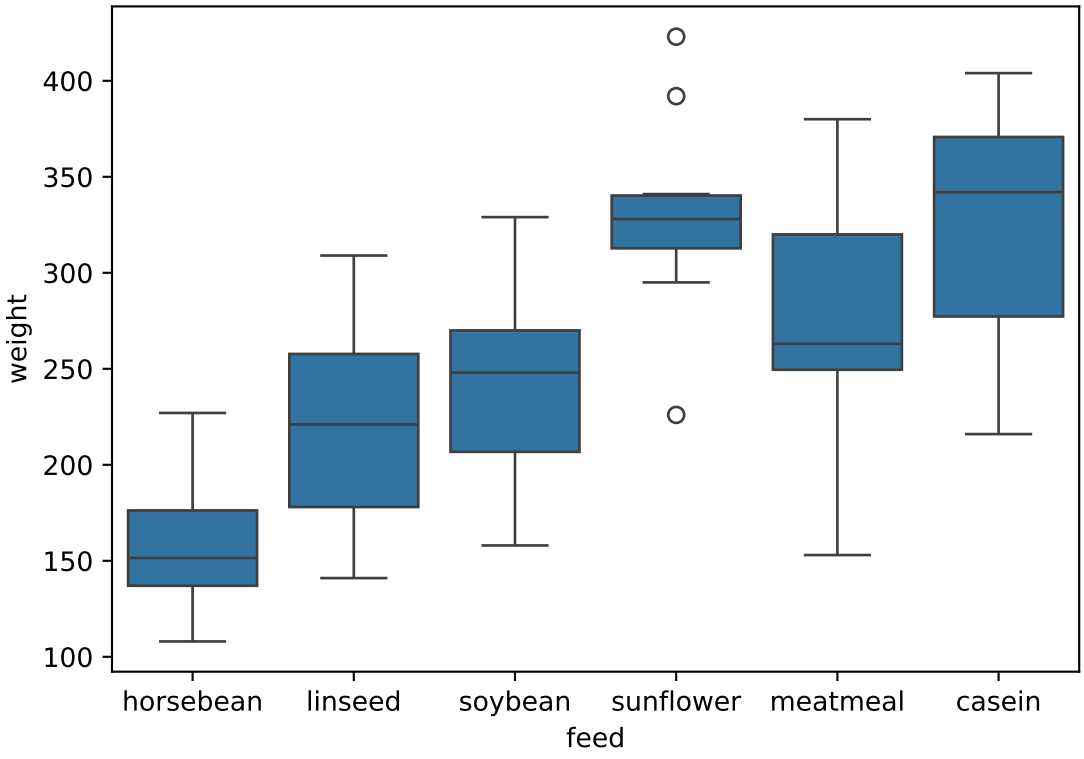
```
import pandas as pd

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)
```



1.2 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.

2 Estimation of mean values

2.1 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 ?

2.2 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**.

2.3 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
```

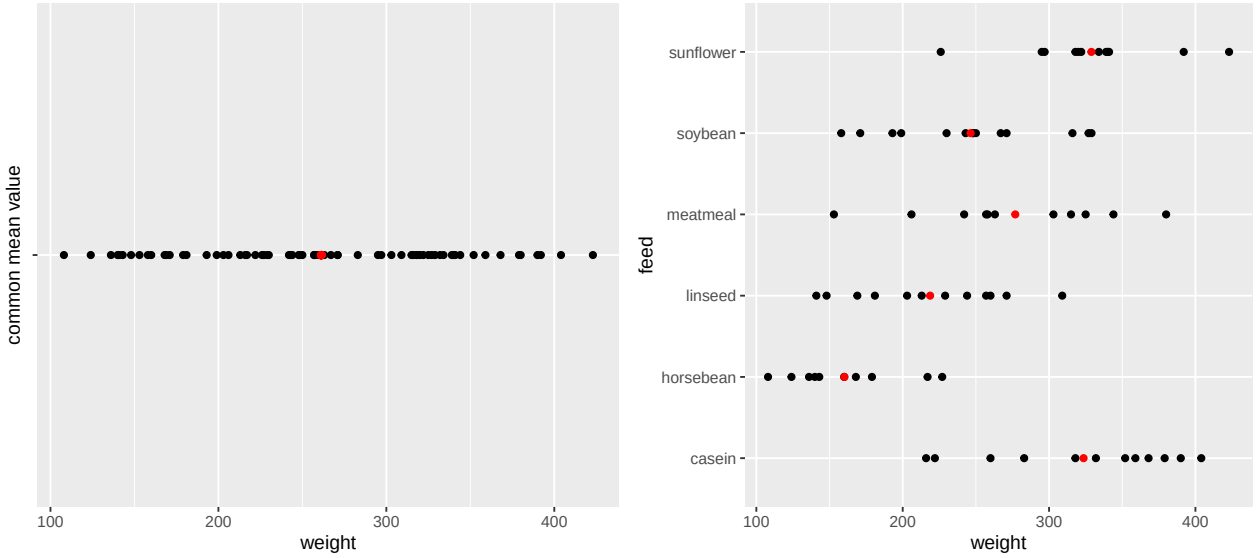
```
## ==============================================================================
##                          coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept           323.5833     15.834     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]    -104.8333     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.
## """
```

- 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%).

3 Overall test for effect

3.1 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."



3.2 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

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

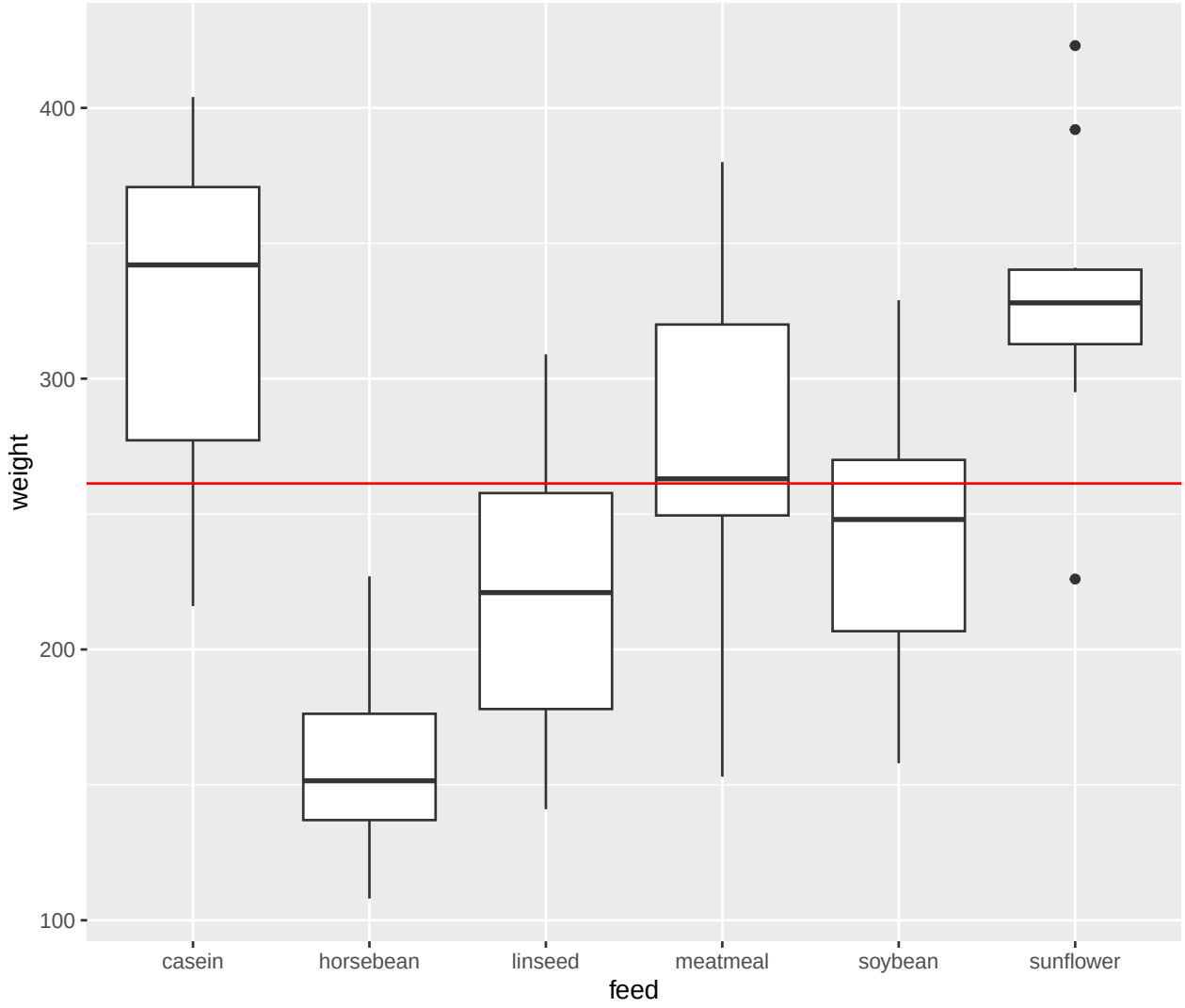
- We will (indirectly) use R^2 to do the test. If it is large, the complex model has good predictive power compared to the simple model. To judge significance we use

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

- 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.

3.3 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

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

3.4 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`.

4 Two way analysis of variance

4.1 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`.

4.2 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	$\mu + \beta_1$	$\mu + \beta_1 + \beta_2$	$\mu + \beta_1 + \beta_3$

4.3 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.

4.4 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`).

4.5 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
```

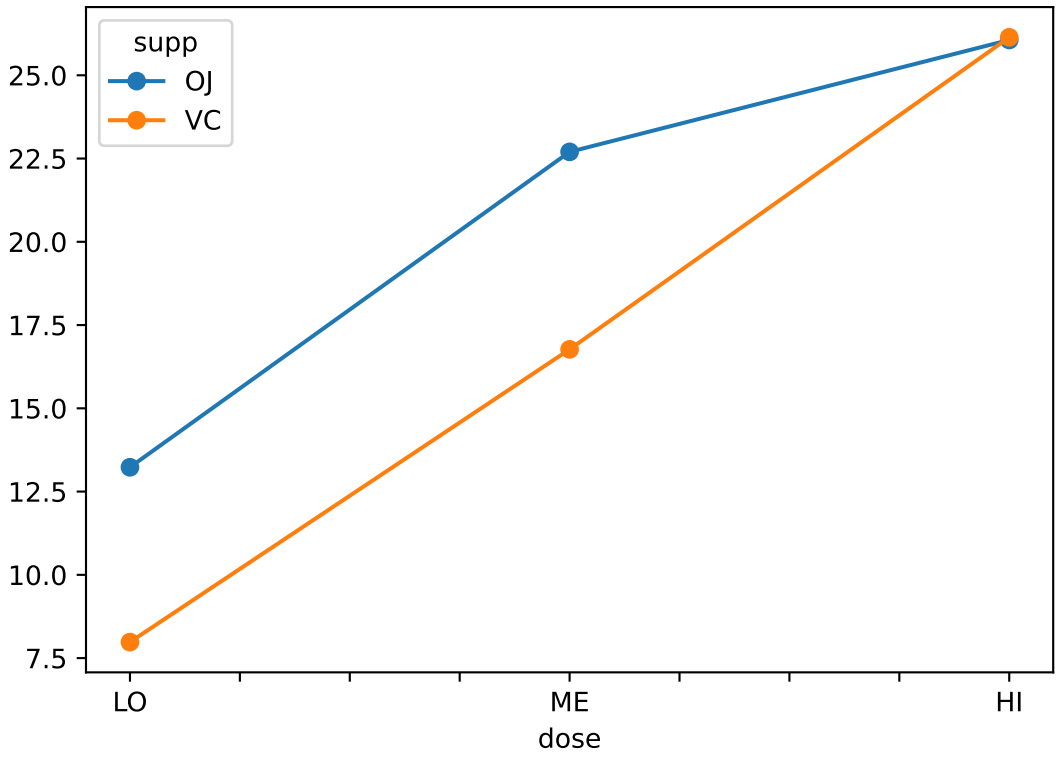
- p-value is ≈ 0 hence we reject that `dose` does not have an effect. Thus we prefer Model 2 (`MainEff`).

5 Interaction

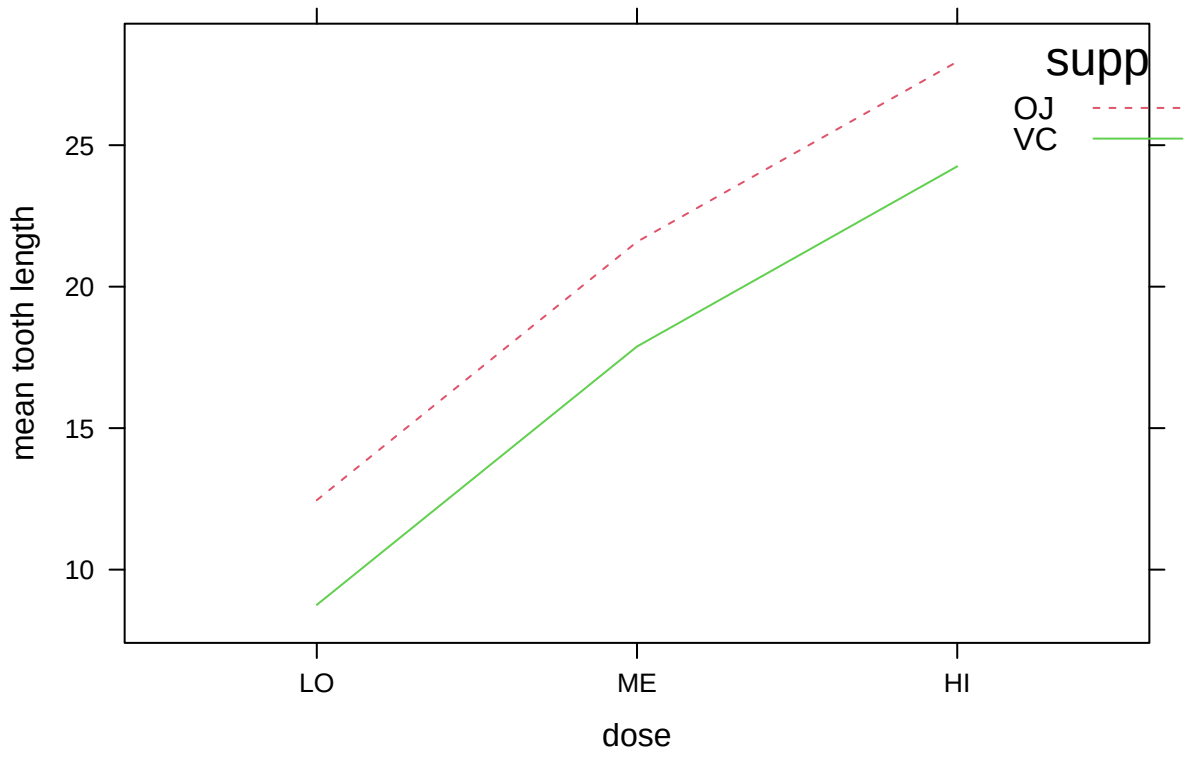
5.1 Example

- 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 !

5.2 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:

	LO	ME	HI
OJ	μ	$\mu + \beta_2$	$\mu + \beta_3$
VC	$\mu + \beta_1$	$\mu + \beta_1 + \beta_2 + \beta_4$	$\mu + \beta_1 + \beta_3 + \beta_5$

- 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`

5.3 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)

5.4 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