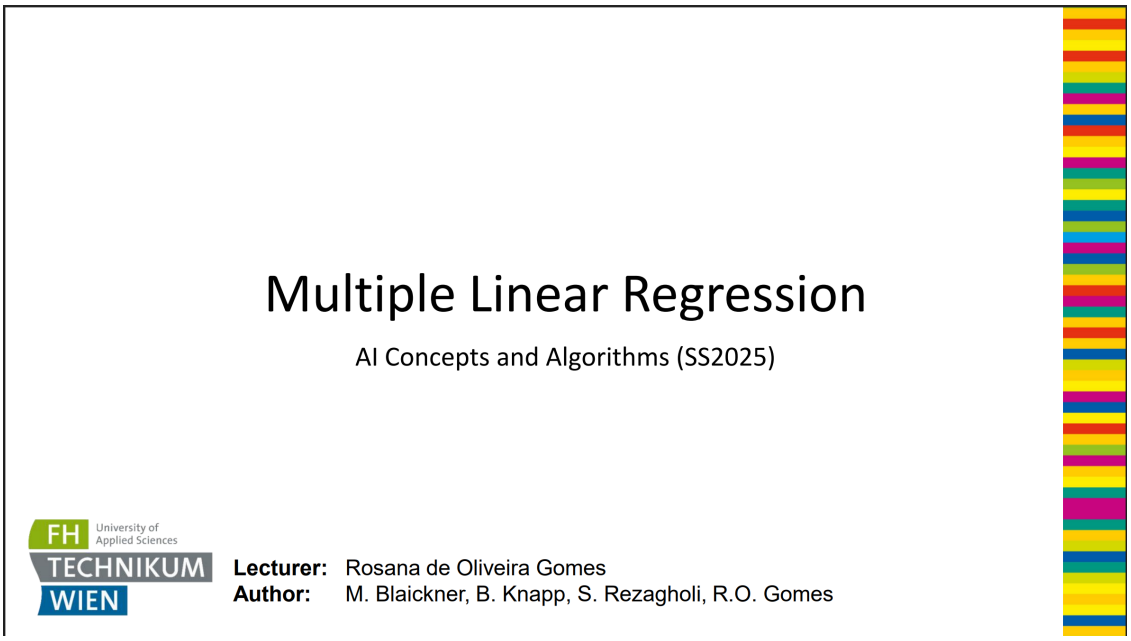
Multiple Linear Regression

June 20, 2025



0.0.1 Course: *AI Concepts and Algorithms (SS2025)*

0.0.2 Topic: Multiple Linear Regression

Here's a clean and effective **introduction you can use**:

0.1 Introduction to Multiple Linear Regression

Multiple Linear Regression (MLR) is one of the fundamental techniques in statistical modeling and machine learning. It allows us to model and analyze the relationship between a quantitative response variable and **two or more predictor variables**.

In contrast to simple linear regression, which deals with only one input feature, MLR helps us understand **how several factors together influence an outcome** — and to what degree.

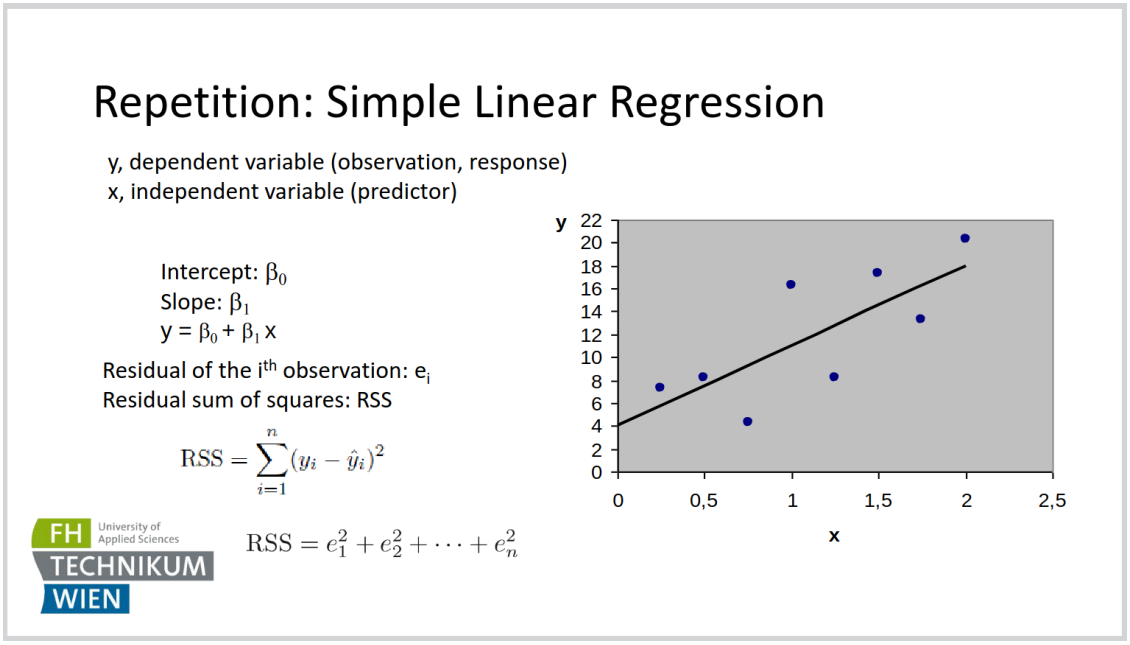
This technique is not only foundational in predictive modeling but also essential for interpreting data relationships, feature importance, and designing intelligent systems.

0.1.1 In this lecture, we will:

- Understand the mathematical foundation of MLR
- Learn how to interpret regression coefficients
- Test hypotheses using **t-tests** and the **F-statistic**
- Measure goodness of fit using **R^2 and adjusted R^2**
- Address challenges like **multicollinearity**, **overfitting**, and **model selection**
- Explore advanced modeling with **interaction terms** and the **hierarchical principle**
- Practice **feature selection** using forward, backward, and mixed strategies

---

By the end of this lecture, you'll be able to build interpretable and statistically sound regression models — and critically evaluate their performance.



0.2 Slide Title: *Repetition: Simple Linear Regression*

0.2.1 Goal:

Review how **simple linear regression** works before jumping into **multiple linear regression**.

0.3 The basic form of simple linear regression is:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

Where:

- y: the **dependent variable** (target or response)
- x: the **independent variable** (predictor)
- β_0: the **intercept** (where the line crosses the y-axis)
- β_1: the **slope** (how much y changes when x increases by 1)
- ε: the **error term** (what the model cannot explain)

0.4 The model tries to minimize the residuals:

A **residual** is the difference between the true value and the predicted value:

$$e_i = y_i - \hat{y}_i$$

0.4.1 The cost function used is the Residual Sum of Squares (RSS):

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} e_i^2$$

This tells you **how far off your model's predictions are from the actual values** — summed over all n data points.

0.4.2 Right-hand plot:

The plot shows:

- A set of **data points** (blue dots)
- A **regression line** that best fits the data
- The **vertical distance** between the points and the line represents the **residuals** e_i

---

0.5 Summary:

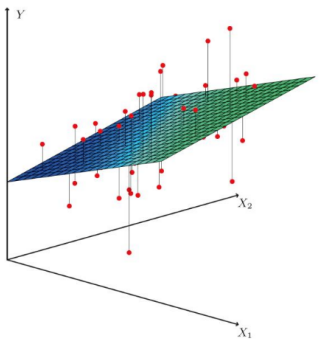
Term	Meaning
β_0	Intercept of the line
β_1	Slope (effect of x on y)
e_i	Residual for observation i
RSS	Total squared error — what we minimize

Multiple Linear Regression (MLR)

- In multiple linear regression a quantitative response **y** with **p different predictors** $x_1, x_2, ..., x_p$ is written in the form

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p + \epsilon = \epsilon + \beta_0 + \sum_{i=1}^{p} \beta_i x_i$$

- where ε is a stochastic error term.
- The parameters $\beta_0, \beta_1, ..., \beta_p$ are estimated using the least squares approach as in simple linear regression.
- In a 3-dimensional setting (2 predictors and 1 response), the least squares regression line becomes a plane that minimizes the sum of the squared vertical distances between each observation (shown in red) and the plane.



Taken from [1]

FH Applied Sciences TECHNIKUM WIEN

3

0.5.1 Slide: *Multiple Linear Regression (MLR)*

0.5.2 What's this slide about?

This slide introduces the mathematical **structure** and **intuition** behind Multiple Linear Regression — a generalization of simple linear regression when you have **more than one input variable** (predictor).

0.5.3 Step-by-step Explanation:

1. **The General Formula:**

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon$$

Or written using summation notation:

$$y = \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \varepsilon$$

Where:

- y: the **dependent variable** (target you're trying to predict)
- $x_1, x_2, \ldots, x_p$: **independent variables** (predictors)
- β_0: **intercept** (what y would be when all x's are 0)
- β_i: the **coefficient** of the i-th feature

- ε: **random error** (captures what your model cannot explain)

---

2. What is (epsilon)?

- ε is the **stochastic error term**.
- It models randomness, noise, and unknown influences not captured by the predictors.
- It's assumed to be normally distributed with mean 0.

  Think of it like: "Stuff the model can't explain."

---

3. How are the 's found? Just like in simple linear regression, we use **least squares estimation**:

- Find the values of $\beta_0, \ldots, \beta_p$ that **minimize the sum of squared errors** between predicted and observed y-values.

---

4. Visual: What's Happening in 3D? On the right side of the slide:

- We see a **3D plot**:
  - X_1 and X_2 on the horizontal axes (two predictors),
  - Y on the vertical axis (target).
- **Red points**: actual observed data.
- **Blue-green plane**: the model's prediction surface.
- **Red vertical lines**: the residuals (distance from real points to the plane).

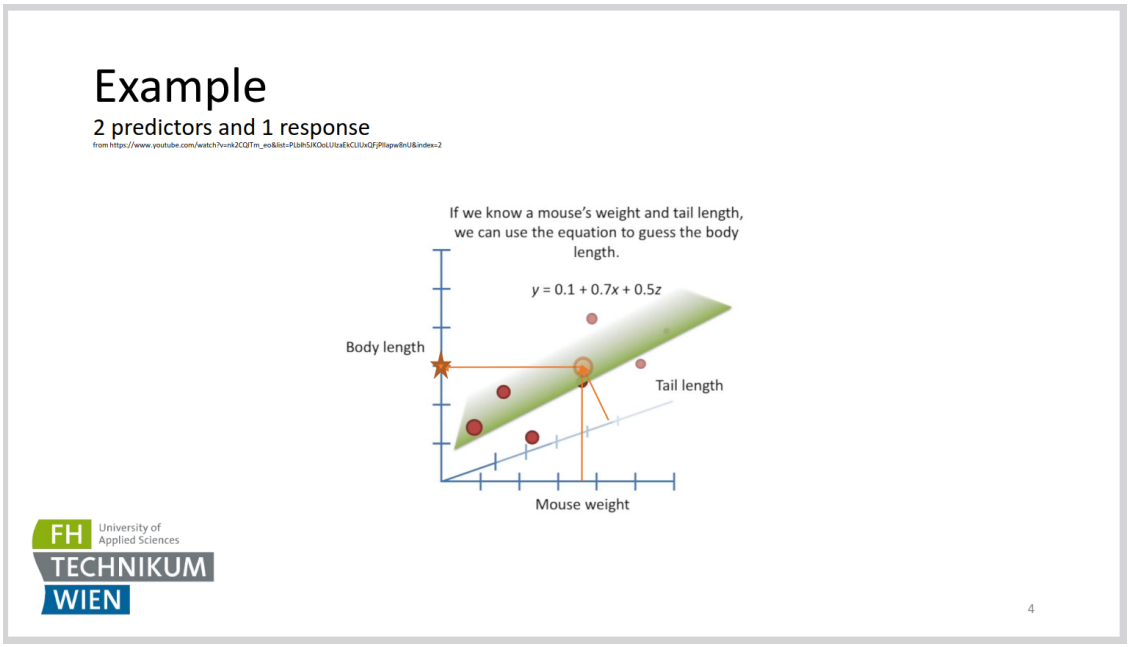
  This is what multiple regression is doing: **Fitting a plane through multidimensional data** in a way that minimizes the total squared vertical distances.

0.5.4 Analogy

In simple regression, you fit a **line** through 2D space. In MLR with 2 predictors, you're fitting a **plane** through 3D space. With 3+ predictors, you're fitting a **hyperplane** in higher dimensions — the math is the same even if we can't visualize it.

0.5.5 Why this matters:

- You now have the power to model **real-world phenomena** using many features.
- Most real-world problems are **not one-dimensional**.
- This sets the stage for **feature selection**, **model evaluation**, and **predictive analytics**.



0.5.6 Slide Title: *Example — 2 Predictors and 1 Response*

0.5.7 What is this teaching you?

This slide **visualizes** how MLR works in a **3D setting**, where you have:

- 2 input features (independent variables): Mouse weight (x-axis) Tail length (z-axis)
- 1 target variable (dependent variable): Body length (y-axis)

We are trying to model how the body length of a mouse depends on its weight and tail length.

0.5.8 The Regression Equation

$$y = 0.1 + 0.7x + 0.5z$$

Where:

- y: body length (what we want to predict)
- x: mouse weight
- z: tail length
- 0.1: intercept (baseline body length when weight and tail length are zero)
- 0.7: how much body length increases per unit of weight
- 0.5: how much body length increases per unit of tail length

This is a classic **linear model** in 3D.

0.5.9 What the Graphic Shows

- The green plane is the **prediction surface** based on the regression equation.
- Red spheres = **actual data points** (real measurements of body length, weight, tail length).
- Orange lines show:
  - The inputs (mouse weight + tail length) locating a point on the **plane** (i.e., predicted body length).
  - The **error** or **residual** between this predicted value and the true body length (shown as a vertical distance to the star).

0.5.10 Why this is important

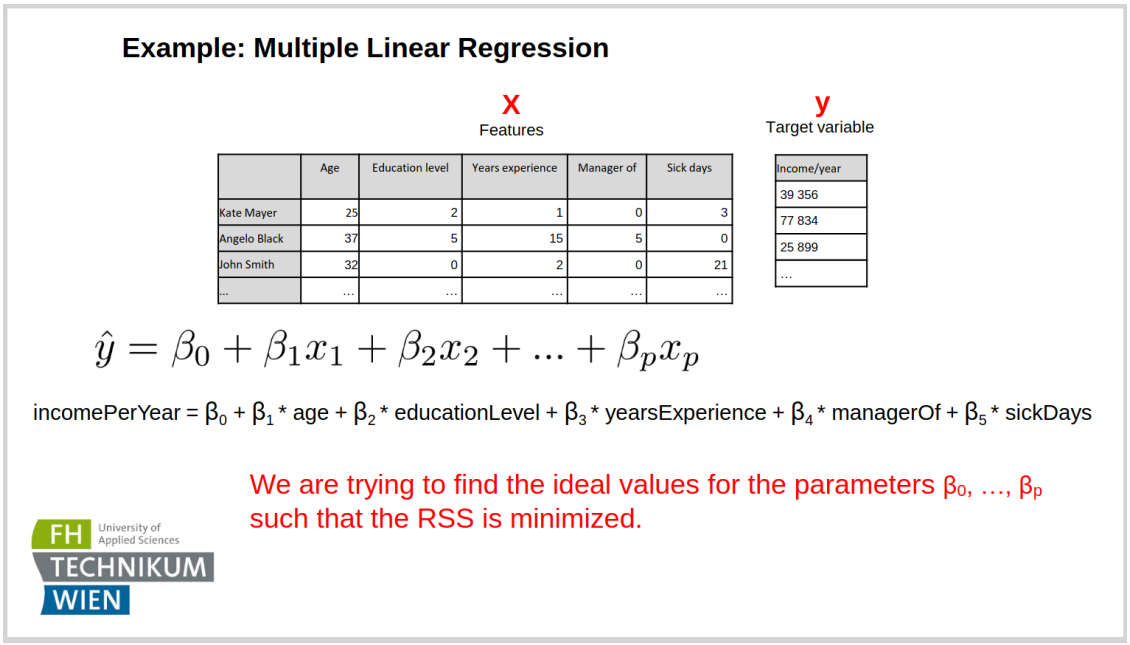
This slide gives you a **tangible 3D visualization** of:

1. **What a prediction looks like** in MLR.
2. **How error is measured** as the distance from the predicted plane to the actual value.
3. **How each variable contributes** additively to the result — you can interpret each coefficient directly.

0.5.11 Conceptual Insight

> MLR doesn't just "fit a curve" — it fits a **flat surface** (a plane) through multi-dimensional data.

In this case, you're seeing that surface in 3D, but MLR works the same if you add more predictors — it just becomes a flat surface in higher dimensions (a hyperplane).



0.5.12 Slide Title: *Example: Multiple Linear Regression*

0.5.13 What is this slide teaching you?

It connects the **general math of MLR** to a **concrete data table**, showing how features (inputs) and targets (outputs) are related through a regression equation.

You're now seeing how a dataset is **structured** for regression and how the **model maps inputs to outputs** using the regression formula.

0.5.14 The Data Setup

Inputs (Features X): These are columns on the left:

- **Age**
- **Education level**
- **Years of experience**
- **Manager of (number of people)**
- **Sick days**

Each row = 1 person (observation).

Output (Target y):

- **Income per year** (numeric value)

We are trying to **predict this value** based on the other columns.

0.5.15 The Regression Formula:

Generalized form:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

In this case:

incomePerYear = β_0+β_1·age+β_2·educationLevel+β_3·yearsExperience+β_4·managerOf+β_5·sickDays

Each β_i tells us **how much income changes** when that feature increases by one unit (assuming others stay constant).

0.5.16 What's the Goal?

The red text at the bottom explains the optimization objective:

> **Find the ideal values of $\beta_0, \beta_1, ..., \beta_p$ that minimize RSS**.

RSS: Residual Sum of Squares

$$\text{RSS} = \sum (y_i - \hat{y}_i)^2$$

We want the model's predictions $\hat{y}_i$ to be **as close as possible** to the actual incomes y_i, over **all people in the dataset**.

That's what "fitting the regression model" means: → **Tune the β's to minimize error.**

0.5.17 Why this is powerful

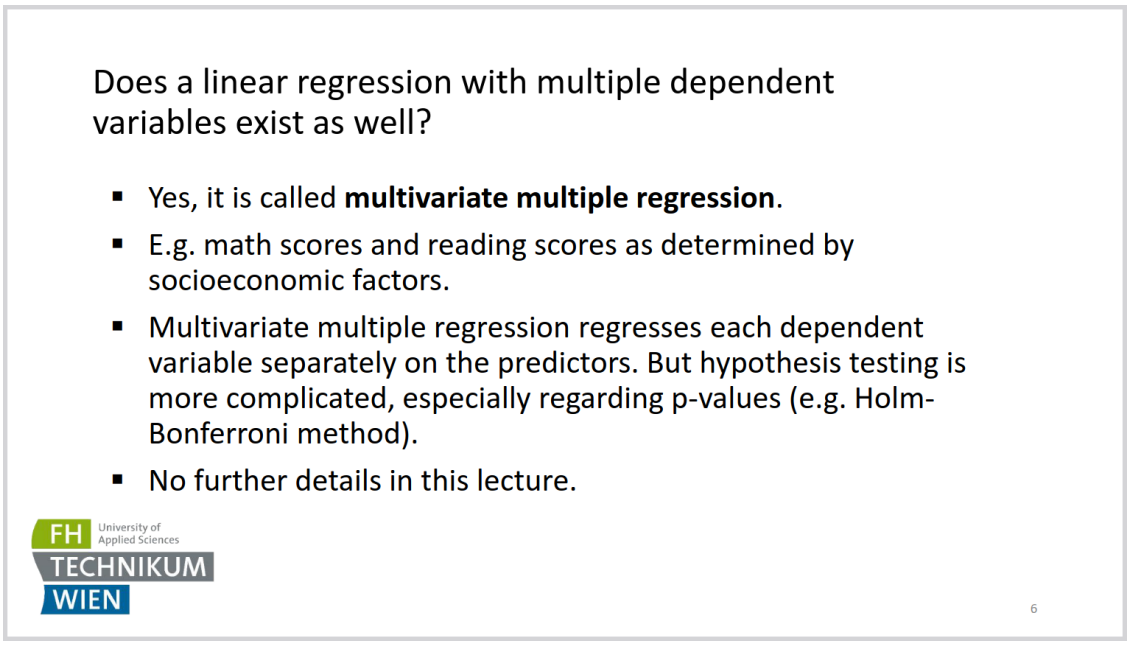
You now know that:

- Any table with numeric inputs and a numeric target can be modeled this way.
- MLR gives you a **formula you can interpret** (unlike black-box models).
- You can use this for **salary prediction**, **medical risk scoring**, **energy consumption**, etc.

Summary:

- You use MLR to relate **multiple variables (X)** to **one target (y)**.
- Your model's job is to find the best -values to make good predictions.
-

0.6 The training process minimizes the total squared error (RSS).



0.6.1 *Does a linear regression with multiple dependent variables exist as well?*

0.6.2 Short Answer: Yes!

It's called **Multivariate Multiple Regression**.

Let's understand what that means.

0.6.3 What's the difference?

Multiple Linear Regression (MLR):

- You have **multiple predictors** $x_1, x_2, ..., x_p$
- You predict **a single outcome** y

  Example: Predict **salary** from age, experience, education level

---

Multivariate Multiple Regression (MMR):

- You still have **multiple predictors** $x_1, x_2, ..., x_p$
- But now you predict **multiple dependent variables** at the same time $\rightarrow y_1, y_2, ..., y_k$

Example from the slide: Predict both **math score** and **reading score** using socioeconomic predictors like income, education, etc.

0.6.4 What does it do?

- You essentially fit **one regression equation per output variable**.
- Each output gets its own β coefficients.
- But all outputs are regressed on the **same set of predictors**.

So you get:

$$\begin{cases} y_1 = \beta_{01} + \beta_{11}x_1 + \cdots + \beta_{p1}x_p + \varepsilon_1 \\ \\ y_2 = \beta_{02} + \beta_{12}x_1 + \cdots + \beta_{p2}x_p + \varepsilon_2 \\ \\ \vdots \end{cases}$$

Each y_k gets its own model, but they're computed **together** in a multivariate setting.

0.6.5 Why is this more complicated?

Because:

- You're no longer just minimizing RSS for one outcome — you're optimizing for **multiple outputs** simultaneously.
- You now have to deal with **correlations between the target variables** (e.g., math and reading scores might be correlated).
- Hypothesis testing gets tricky:
  - You can't use simple p-values.
  - You need corrections for **multiple comparisons** like the **Holm-Bonferroni** method (a more conservative p-value adjustment).

0.6.6 Why the lecture skips details

This is a more **advanced topic** in statistical modeling:

- It blends elements from multivariate analysis, MANOVA, and covariance structure modeling.
- It's less common in basic AI/ML workflows unless you're doing **multi-output regression** or **structural equation modeling**.

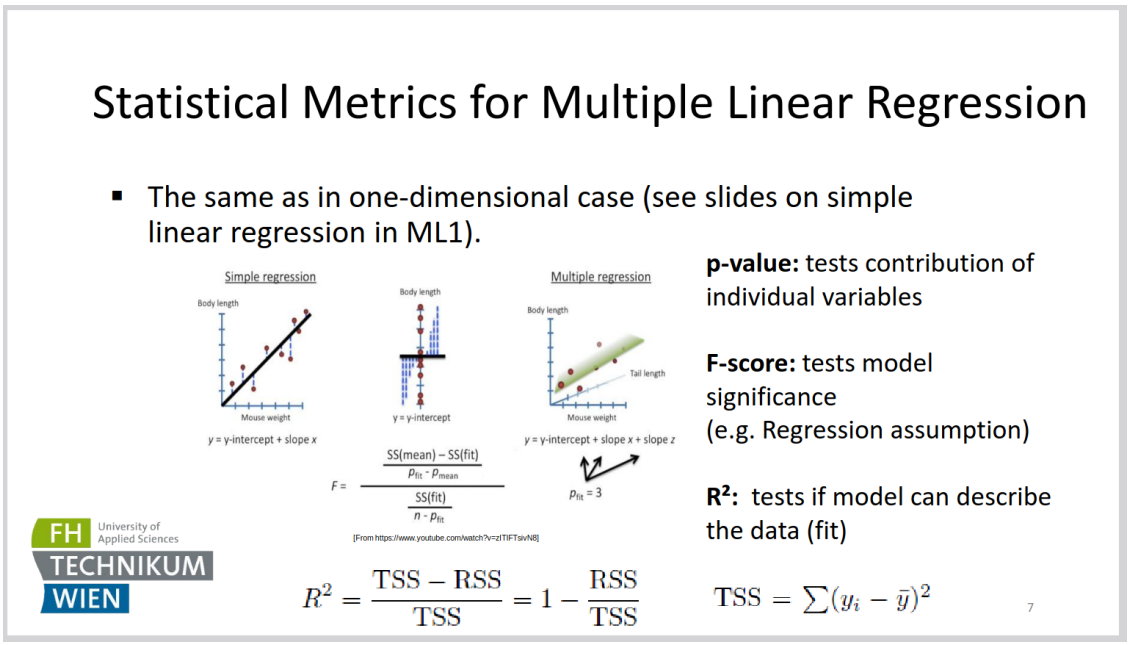
0.6.7 Key takeaway:

If you're modeling **more than one outcome variable** at once, you're doing **multivariate multiple regression**.

In standard ML libraries, this is often handled via:

- `MultiOutputRegressor` in scikit-learn
- Custom loss functions in PyTorch/Keras for multi-target regression

---



0.6.8 Slide Title: *Statistical Metrics for Multiple Linear Regression*

0.6.9 Main Message:

Even though MLR involves multiple predictors, **the evaluation metrics remain the same** as in simple linear regression. The goal is still to measure:

- How well the model fits the data.
- Whether the predictors are useful.

0.7 Key Metrics Explained

0.7.1 1. p-value

What it does: Tests the **statistical significance of each individual predictor**.

- Each coefficient β_i has an associated p-value.
- It answers: *"If this variable had **no real effect**, what is the probability we would observe a coefficient this large?"*

Interpretation:

- Small p-value (< 0.05) $\rightarrow$ strong evidence that predictor is useful.
- Large p-value $\rightarrow$ we can't be confident this variable is helping.

Use this to decide whether to **keep or drop** a variable.

0.7.2 2. F-score (F-statistic)

What it does: Tests whether the **model as a whole** is statistically significant.

Formula on the slide:

$$F = \frac{\text{SS(mean)} - \text{SS(fit)}}{\text{SS(fit)}} \cdot \frac{n - p_{\text{fit}}}{p_{\text{fit}} - p_{\text{mean}}}$$

In words:

- Compares the **variance explained by the model** vs. the variance unexplained.
- If the F-score is **high**, the model is doing better than just predicting the mean.

Use this to **evaluate the overall model**.

0.7.3 3. R^2 (R-squared)

What it does: Measures the **proportion of variance in the target** y that the model explains.

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}}$$

Where:

- RSS = Residual Sum of Squares $\sum(y_i - \hat{y}_i)^2$
- TSS = Total Sum of Squares $\sum(y_i - \bar{y})^2$

Interpretation:

- $R^2 = 1$: perfect model.
- $R^2 = 0$: model is no better than predicting the mean.

Use this to judge **model fit**.

0.7.4 Visual Summary (left side of slide)

- **Left graph**: Simple regression line with vertical residuals.
- **Middle**: Breakdown of how mean and fitted models are compared using sum of squares.
- **Right**: MLR example with multiple variables forming a plane.

This visualization shows the **geometry of regression**: it's about minimizing vertical distances (errors) between actual data and your model's predictions.

0.7.5 Real Use Case:

Imagine you're modeling **house prices** with predictors like size, location, and number of bathrooms.

- R^2 tells you how well the model fits overall.
- F-score tells you if your model beats just guessing the average price.
- p-values tell you which predictors matter most.

Slide takeaway: You now have **quantitative tools** to:

- Diagnose your model's performance
- Justify variable inclusion
- Compare models

0.8 1. What is SS?

SS stands for **Sum of Squares**, and it's a general term used in statistics to measure **total variation**.

In regression, we usually talk about three types:

Abbreviation	Full Name	What it measures
TSS	Total Sum of Squares	Total variation in y
RSS	Residual Sum of Squares	Unexplained variation (error)
ESS or SSR	Explained Sum of Squares	Variation explained by the model

---

0.9 2. What's the difference between TSS and RSS?

Let's go step by step.

0.9.1 TSS – Total Sum of Squares

$$\text{TSS} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

- Measures how much the data y_i vary **around the mean** $\bar{y}$.
- It's the **total variance** in the output data before fitting any model.
- Think of it as: "How spread out are the actual values from the average?"

0.9.2 RSS – Residual Sum of Squares

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- Measures how much the **predictions** $\hat{y}_i$ miss the actual values y_i.
- It's the **error** your model made — what's **not** explained.
- Think of it as: "How far off are my predictions from reality?"

0.9.3 Their relationship:

$$\text{TSS} = \text{RSS} + \text{ESS}$$

So:

- TSS = Total variance in the target variable
- RSS = What the model **fails** to explain
- ESS = What the model **successfully** explains

0.9.4 And this gives us R^2:

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}}$$

- If RSS is **small**, your model explains most of the data $\rightarrow R^2 \approx 1$
- If RSS is **large**, your model barely helps $\rightarrow R^2 \approx 0$

0.9.5 Intuition:

- **TSS**: how much variation exists in the target?
- **RSS**: how much of that variation is left after modeling?
- **ESS**: how much did the model explain?

0.10 Why is it called R-squared (R^2)?

Because originally, in **simple linear regression**, R^2 is literally the **square of the Pearson correlation coefficient** R between the predicted values $\hat{y}$ and the actual values y.

0.10.1 In Simple Linear Regression:

You only have one predictor x, and the correlation between x and y is:

$$R = \frac{\text{Cov}(x, y)}{\sigma_x \sigma_y}$$

And guess what?

In this case, the coefficient of determination is:

$$R^2 = (\text{correlation between } y \text{ and } \hat{y})^2$$

So it's the **proportion of variance in** y that is **explained by** x.

0.10.2 Example:

Let's say:

- Correlation between height and weight is $R = 0.9$
- Then $R^2 = 0.81$

That means **81% of the variance in weight is explained by height**.

0.10.3 In Multiple Linear Regression?

Now you're dealing with **multiple predictors** $x_1, x_2, ..., x_p$, so the relationship isn't just one-to-one anymore.

But even in this case, R^2 still retains its **meaning as a proportion of variance explained** — it's just no longer the literal square of a single correlation coefficient.

Still, we keep the name **R-squared** for historical and conceptual continuity.

0.10.4 So in short:

Symbol	Meaning
R	Correlation between y and predictions
R^2	Proportion of variance in y explained by the model

Summary:

- It's called R^2 because in simple regression, it's literally the square of the correlation.
-

0.11 In multiple regression, it still measures the same concept: how well your model explains the variation in y.

0.12 What is TSS really doing?

$$\text{TSS} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

Where:

- y_i: each observed value
- $\bar{y}$: the **mean** of all y values

0.12.1 What does this mean in practice?

You are measuring:

> "How far is each value from the average?"

Step-by-step:

1. Take all your actual values $y_1, y_2, \ldots, y_n$
2. Compute the mean:

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

3. For each data point, subtract the mean: $y_i - \bar{y}$
4. Square that difference: $(y_i - \bar{y})^2$
5. Add all the squared differences together.

0.12.2 What does this give you?

- The **Total Sum of Squares (TSS)**: the total variability in the data.
- It's a way of saying: "If I only predicted the mean for everyone, how much error would I make overall?"

So:

- **TSS is the baseline error**.
- It's what you'd get if your model was super lazy and just guessed the average every time.

0.12.3 Connection to the Model:

When you train a regression model, your goal is to **reduce the error** from this baseline:

- If your model's predictions $\hat{y}_i$ are better than $\bar{y}$, then:

  $$\text{RSS} < \text{TSS}$$

  and

  $$R^2 > 0$$

- If not, your model isn't better than just guessing the mean, and R^2 can even be negative (in bad models).

0.12.4 Visual intuition:

If you had a plot of y values:

- Draw a horizontal line at $\bar{y}$
- Measure vertical distances from each y_i to that line
- Square and sum them $\rightarrow$ that's your TSS

0.12.5 What does the middle graph show?

This plot illustrates the **baseline model**: a horizontal line at $\bar{y}$ — the **mean** of all the observed y-values.

It's visualizing the concept of **TSS** — the total variation in your data **before any modeling happens**.

0.12.6 Breakdown of the graphic:

Horizontal line = $y = \bar{y}$

- This is the simplest possible "model" — just predict the average for everyone.
- This line is flat because the **mean is constant** for all inputs.

Red dots = actual y_i values

- These represent the real observed data points.

Blue vertical lines = deviations from the mean

- These show how far off each actual y_i is from the mean $\bar{y}$.
- Each of these is:

$$y_i - \bar{y}$$

0.12.7 So what is this graph teaching?

It's **defining the Total Sum of Squares (TSS)** visually:

$$\text{TSS} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

You're literally:

1. Drawing a line at the mean.
2. Measuring every point's vertical distance from that line.
3. Squaring and summing those distances.

0.12.8 Why is this important?

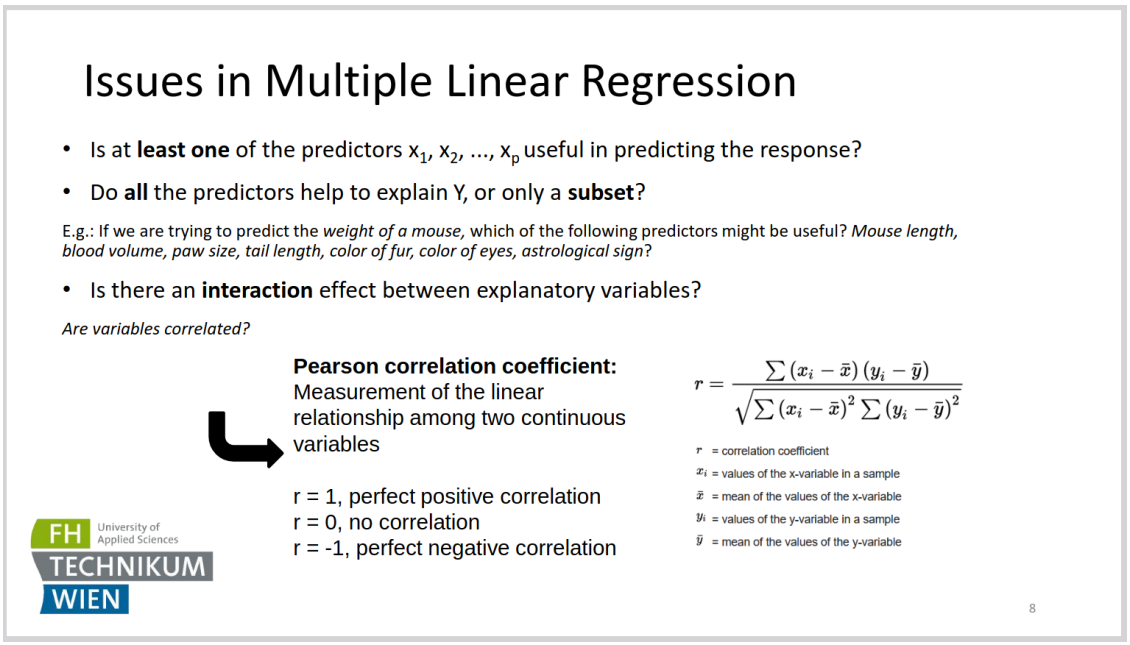
Because **TSS is the baseline** against which we compare our model's performance.

- If your model doesn't reduce the errors below TSS, it's **useless**.
- A good model will **pull those red dots closer to a new regression line or plane**, reducing the residuals (RSS) and increasing R^2.

0.12.9 In the full picture:

Sum of Squares	Visual Meaning	Purpose
TSS	Distance from points to mean line	Total variation in data
RSS	Distance from points to prediction line	Remaining error after modeling
ESS	Distance from prediction line to mean line	What model explains

---



0.12.10 Slide Title: *Issues in Multiple Linear Regression*

0.13 What this slide is about:

MLR is powerful, but it brings complexity. This slide highlights **critical questions** you should always ask before trusting your model.

0.14 Key Questions in MLR

0.14.1 1. Is at least one of the predictors useful?

Are any of the $x_1, x_2, ..., x_p$ variables actually helping us predict y?

- This is where the **F-test** comes in.
- If **none** of the predictors help → model is worthless.

0.14.2 2. Do all the predictors help, or just a subset?

Maybe only 3 out of 10 predictors are useful, and the rest are just noise or distractions.

This brings up the concept of **feature selection**:

- Which features are **truly informative**?
- Which are **irrelevant** or **redundant**?

Example given:

To predict **mouse weight**, should we include things like:

- Mouse length (probably useful)
- Astrological sign (nonsense)
- Eye color (irrelevant)
- Tail length (maybe useful)
- Paw size
- Blood volume

You need domain knowledge + statistical tests to decide.

0.14.3 3. Are there interaction effects between variables?

Sometimes variables affect the outcome **together**, in ways they don't individually.

Example:

- Having **high age** AND **high blood pressure** may drastically increase risk, even if neither alone is a strong predictor.

This leads to **interaction terms** like:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$$

That extra term captures **joint effects**.

0.15 Are variables correlated?

This is crucial — when predictors are **too correlated**, it causes **multicollinearity**, which makes your model unstable.

To measure this, we use:

0.16 Pearson Correlation Coefficient (r)

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

This measures the **strength of linear relationship** between two continuous variables.

0.16.1 Interpretation:

Value of r	Meaning
$r = 1$	Perfect **positive** linear correlation

Value of r	Meaning
$r = 0$	**No** linear correlation
$r = -1$	Perfect **negative** linear correlation

0.17 Summary:

- Not all predictors are useful — some are even harmful to model accuracy.
- Some variables only work well **together** (interactions).
- Correlated predictors make models **unstable** — coefficients can fluctuate wildly with small changes in data.
- Always **check for correlation** using tools like Pearson's r.

0.18 Pearson Correlation Coefficient (r) is:

- A **pairwise** metric.
- It measures the **linear relationship between exactly two variables** at a time.

$$r = \text{corr}(x_i, x_j)$$

You can compute:

- $\text{corr}(x_1, x_2)$
- $\text{corr}(x_1, x_3)$
- $\text{corr}(x_2, x_3)$
- etc.

But **never all together** — Pearson's r doesn't handle multivariate relationships directly.

0.19 So what does this mean in regression?

When you're building a Multiple Linear Regression model:

- You want to check if **any two predictors are strongly correlated**.
- If they are, you may face **multicollinearity**, which messes up coefficient stability.

0.19.1 Example

Imagine:

```
x1 = years of education
x2 = level of degree (e.g., 0 = none, 1 = bachelor, 2 = master, 3 = PhD)
```

These will have a **very high correlation**, maybe $r = 0.95$. That means they carry **almost the same information**.

Including both in your model? Bad idea — unstable values and misleading interpretations.

0.20 Solution: Correlation Matrix

To detect this in practice:

- Compute a **correlation matrix** between all pairs of features.
- Use heatmaps or a pandas `.corr()` call:

```
df.corr(method='pearson')
```

This gives you an **overview of all pairwise relationships**.

0.21 Summary

Concept	Applies to	Detects
Pearson r	Two variables	Linear relationship strength
Correlation matrix	All pairs	Redundancies & collinearity
Multicollinearity warning	More than two	If strong correlation exists
Regression model instability	Multiple predictors	Due to shared signal

---

0.22 What does Pearson correlation measure?

Pearson's r measures **how strongly two variables are linearly related** — **without** fitting a regression model.

- It doesn't calculate slopes or predictions.
- It tells you **how tightly the points follow a line**, and whether the trend is **positive or negative**.

$$r = \frac{\text{Cov}(x, y)}{\sigma_x \cdot \sigma_y}$$

If you plot x vs. y and see a perfect line (up or down), $r = 1$ or $r = -1$.

0.23 What do slopes in regression measure?

In multiple linear regression:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

Each **slope coefficient** β_j measures:

> The **change in** y when x_j increases by 1 **holding all other variables constant**.

This is very different from Pearson correlation.

0.23.1 So what's the relationship?

- If you only have **one predictor** x, then:

$$\beta_1 = r \cdot \frac{\sigma_y}{\sigma_x}$$

  So here the slope **is directly related** to correlation.

- But in **multiple regression**, each slope β_j is adjusted for the **presence of other variables**.

That means:

- Even if x_1 and y have a high correlation,
- The slope β_1 in the model might be small if x_1 overlaps with other predictors (i.e., multicollinearity).

0.24 Final Insight

Concept	Meaning
Pearson r	How tightly x and y move together (linear association)
Regression slope β_j	The actual **effect size** of x_j on y, after adjusting for other variables

So no — Pearson correlation doesn't measure slope, but in **simple regression**, slope is **proportional to correlation**. In **multiple regression**, correlation helps you identify **redundant or collinear features**, which then affect slope values indirectly.

BOOM! Now you've asked the **right question** that gets to the *real* problem in multiple linear regression:

> **"What does the Pearson correlation between two features x_1 and x_2 measure, and how does it affect the regression?"**

Let's unpack this in detail:

0.25 When we compute:

$$r = \text{corr}(x_1, x_2)$$

This tells us:

- How **linearly related** the two **independent variables** are.
- It **doesn't involve the target** y at all.
- It simply checks: "Do x_1 and x_2 rise and fall together?"

0.26 So what does that mean for regression?

If x_1 and x_2 are **highly correlated**, that's **multicollinearity**.

> The regression algorithm **can't distinguish** which variable is doing the work — because they both contain almost the same information.

0.26.1 What happens:

- The model tries to assign "blame" (slope β) to each feature.
- But because they're redundant, small changes in data can cause **large changes in slope** values — even **flipping signs**.
- The model becomes **unstable** and **hard to interpret**.

0.27 So no — Pearson $r(x_1, x_2)$ does not measure slope directly...

...but it **alerts you** that:

- Slopes β_1, β_2 might not be reliable.
- **Interpretation** of individual features becomes dangerous.

0.28 Summary:

What it is	What it tells you
$r(x_1, x_2)$	Linear relationship between two input features
High correlation between inputs	You may have **redundant information** and **unstable coefficients**
Consequence in regression	Coefficients (slopes) can become **unreliable** or **counterintuitive**
What to do	- Drop one variable- Use PCA- Use regularization (Ridge/Lasso)

0.28.1 Visual intuition:

If x_1 and x_2 are almost the same line — you're trying to fit a plane using two vectors pointing in the same direction. That plane becomes **wobbly** — not numerically stable.

0.29 Pearson correlation formula:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2} \cdot \sqrt{\sum(y_i - \bar{y})^2}}$$

This is the **general formula** for computing **linear correlation** between two variables.

0.30 If you're evaluating correlation between two features, say:

- x_1: education level
- x_2: years of experience

Then yes, the **correct use** of the formula would be:

$$r = \frac{\sum_{i=1}^{n}(x_{1i} - \bar{x}_1)(x_{2i} - \bar{x}_2)}{\sqrt{\sum(x_{1i} - \bar{x}_1)^2} \cdot \sqrt{\sum(x_{2i} - \bar{x}_2)^2}}$$

Where:

- x_{1i}: value of feature 1 for observation i
- x_{2i}: value of feature 2 for observation i

You're exactly right:

> **The x-axis is one feature, and the y-axis is the other feature**, not the regression output.

0.31 Slide Confusion Clarified

The formula on the slide is **technically written as x and y** for general purposes.

But in your context — checking **correlation between two predictors** to detect multicollinearity — that formula becomes:

$$\text{corr}(x_1, x_2) = r(x_{1i}, x_{2i})$$

0.32 So yes:

- Use that formula with **two features**: x_1 and x_2
- It tells you how similar their values are across all samples
- You'll be plotting **feature vs. feature**, **not feature vs. target**

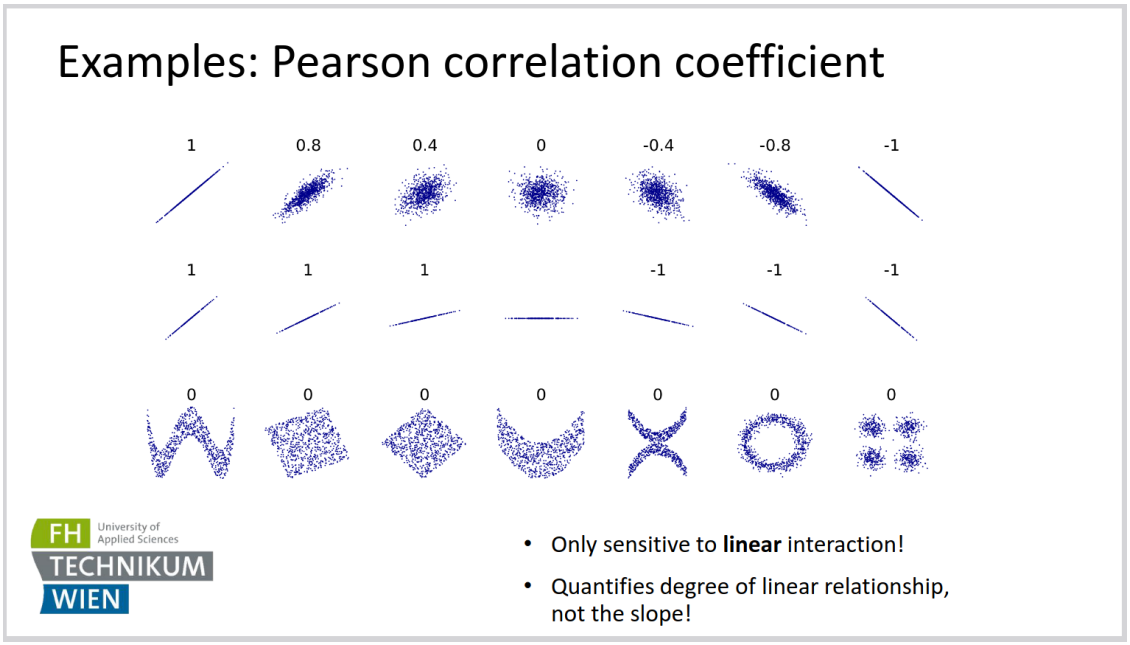
0.33 Final reminder:

0.33.1 Good:

```
df[['education', 'experience']].corr()
```

0.33.2 Bad:

```
df[['experience', 'salary']].corr()  # ← not helpful for collinearity, but for linear associat
```



0.33.3 Slide Title: *Examples: Pearson correlation coefficient*

0.34 What's this slide showing?

It's showing a series of scatter plots with **different patterns of data**, and the **correlation coefficient (r)** computed for each.

0.35 Key Concepts Illustrated

0.35.1 1. Pearson correlation only detects linear relationships

- If your data points form a **straight line** (or approximate one), you'll get:
  - $r = 1 \rightarrow$ perfect positive linear
  - $r = -1 \rightarrow$ perfect negative linear
- If your data forms a **curve**, circle, X, or any **non-linear** shape, $\rightarrow$ Pearson correlation is **blind** to it $\rightarrow$ You can get $r = 0$ even when there's a very clear relationship.

0.36 Interpretation of Rows:

0.36.1 Top row: Linear correlations

- Clear diagonal lines from bottom-left to top-right $\rightarrow r = 1$
- As the scatter increases, r drops toward 0.
- When the slope flips, r becomes negative.

0.36.2 Middle row: Perfect lines with less/no variance

- Straight lines with **exact 1, 0, or -1** correlation.
- These show **idealized mathematical cases** where all points lie perfectly on a line.

0.36.3 Bottom row: Curved or complex relationships

- W-shaped, circular, X-shaped, or ring-shaped patterns.
- All have $r = 0$, because:
  - The data doesn't form a linear trend.
  - Positive and negative directions cancel out.

But clearly — the variables **are related**, just **not linearly**.

0.37 Takeaways from the text at bottom:

0.37.1 "Only sensitive to linear interaction!"

- Pearson r can't see curves, circles, or nonlinear trends.
- Use caution: just because $r = 0$ doesn't mean "no relationship" — it means "no linear relationship."

0.37.2 "Quantifies degree of linear relationship, not the slope!"

- People confuse correlation with slope — but correlation is **unitless** and standardized.
- You can have a **high slope but low correlation**, or **vice versa**.
- Correlation measures how **well points fit a line**, not how **steep** that line is.

0.38 Bottom Line:

If you see this pattern	Pearson r says
Clear rising diagonal	$r \approx 1$
Clear falling diagonal	$r \approx -1$
Scatter, no trend	$r \approx 0$
Circle, curve, X	$r = 0$, but misleading

0.39 The Horizontal Line in the Middle Row

That plot shows:

- All y-values are **constant** (i.e., the same).
- The x-values vary (spread left to right).
- The resulting **Pearson correlation** $r = 0$.

0.39.1 Why?

Because:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2} \cdot \sqrt{\sum(y_i - \bar{y})^2}}$$

In this case:

- $y_i = \bar{y}$ for all i
- So $y_i - \bar{y} = 0$
- Therefore, the entire numerator becomes **0**

$$r = \frac{0}{\text{some number}} = 0$$

0.40 What does this mean conceptually?

0.40.1 This is the "No Relationship" case

- Changing x **does not cause** any change in y.
- The regression model would say: "Your feature x explains **nothing** about y."
- There's **zero linear association** between these variables.

0.40.2 Real-world analogy:

Imagine you're trying to predict **exam scores** (y), but everyone got **exactly 75** regardless of study hours (x).

- The data would form a **horizontal line**.
- Correlation = 0.
- Slope of regression line = 0.
- Conclusion: "Studying doesn't affect the grade" (in this dataset).

0.40.3 And in multiple regression?

If you had a feature x_j that behaves like this relative to the target, the **p-value for** β_j would be **very high** $\rightarrow$ meaning the feature is **not useful**.

0.41 Final takeaway:

That flat horizontal line shows a variable with:

- **Zero variance in the output**
- **Zero correlation**
- **Zero predictive power**

It's the statistical **"dead zone"** — completely uninformative.

0.42 You analyze Pearson correlations in two different ways, with two different goals:

0.42.1 1. Correlation between each predictor x_j and the target y

To check **how useful** the feature might be.

- If $\text{corr}(x_j, y)$ is high (positive or negative), the feature is **likely predictive**.
- If $\text{corr}(x_j, y) \approx 0$, the feature might be **irrelevant** (not necessarily, but it's a red flag).

Goal: **Feature relevance**

0.42.2 2. Correlation between predictor variables x_j and x_k

To detect **multicollinearity** (redundancy).

- If $\text{corr}(x_j, x_k) \approx 1$ or -1: danger!
- It means the features carry **overlapping information**.

- This makes regression coefficients unstable (standard errors blow up, signs flip, model becomes sensitive to small data changes).

  Goal: **Model stability and interpretability**

0.43 Practical Workflow:

0.43.1 Step 1: Correlation with y

```
for col in features:
    print(col, df[col].corr(df['target']))
```

Helps you **rank predictors by importance** (roughly).

0.43.2 Step 2: Correlation matrix between features

```
df[features].corr()
```

Use a **heatmap** (like with seaborn) to find:

- Strongly correlated feature pairs → drop one, combine, or regularize.

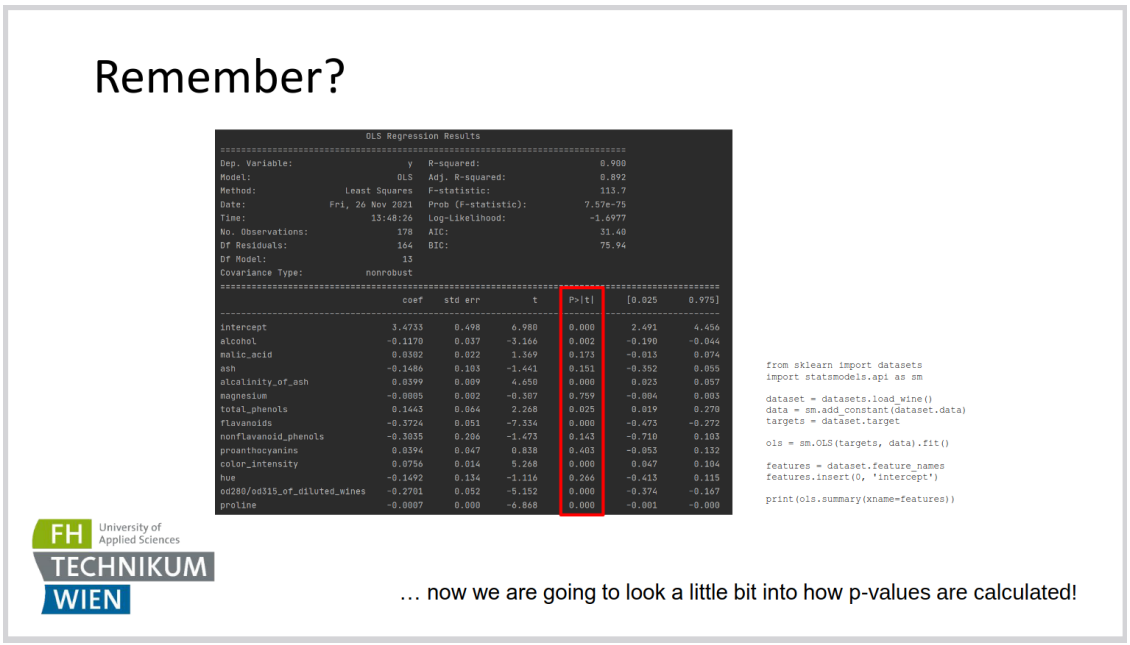
0.43.3 Extra Tip: Use a rule of thumb

Correlation value	Rule of thumb
> 0.8 or < -0.8	High collinearity — investigate
0.3–0.8	Possibly useful
~ 0	Maybe not useful

0.44 Summary

Type	Variable Pair	Purpose
$\text{corr}(x_j, y)$	Feature vs. target	Check predictive relevance
$\text{corr}(x_j, x_k)$	Feature vs. feature	Check for multicollinearity

---



0.44.1 Slide Title: *Remember?*

0.45 What are we looking at?

This slide shows output from a **multiple linear regression** (using `statsmodels.OLS`) on the **wine dataset** from scikit-learn.

The column highlighted in red is:

$$\mathrm{P} > |\mathrm{t}| \quad \rightarrow \text{the p-value for each predictor}$$

0.46 What does this output tell you?

It's a classic regression summary from `statsmodels`, showing:

Column	Meaning		
coef	Estimated value of the regression coefficient $\hat{\beta}$		
std err	Standard error (uncertainty) in the coefficient estimate		
t	t-statistic $= \frac{\text{coef}}{\text{std err}}$		
P >	t	**	p-value = probability the true co
[0.025, 0.975]	95% confidence interval for the coefficient		

0.47 Why is p-value so important?

Because it tells you:

> "Is this feature likely to have a **real effect** on the target?"

In this context:

- A **low p-value** (< 0.05 or < 0.01) → strong evidence that the coefficient is **not** zero → predictor is **statistically significant**
- A **high p-value** → you can't trust that predictor; it may just be noise

---

0.47.1 Let's interpret one example:

`flavanoids  coef: -0.3724  std err: 0.062  t: -5.991  p-value: 0.000`

- Interpretation: flavanoids are **highly significant** in predicting the target
- The negative sign means: more flavanoids → lower target value
- The model is 99.9% sure this is **not by chance**

0.48 Why the slide says "Remember?"

Because you've seen this summary before — it's reminding you:

- p-values are already being calculated **behind the scenes**
- Now you're about to dig deeper into **how** those values are computed (via t-tests, distributions, etc.)

0.49 Code block explained:

```
dataset = datasets.load_wine()
data = sm.add_constant(dataset.data)
targets = dataset.target
ols = sm.OLS(targets, data).fit()
print(ols.summary())
```

This code:

- Loads a sample dataset
- Adds an intercept column
- Fits an Ordinary Least Squares (OLS) model
- Prints out the full statistical summary — including those p-values

---

0.49.1 What's next?

The line at the bottom hints:

"… now we are going to look a little bit into how p-values are calculated!"

This means the next slide(s) will unpack:

- The **math behind the t-test**
- The connection to sampling distributions
- How these probabilities are actually derived

0.50 So... what is a "high" p-value?

In general:

p-value	Interpretation
< 0.01	**Very strong evidence** against the null hypothesis
< 0.05	**Strong evidence** (common threshold)
$0.05 - 0.1$	**Moderate/weak evidence** — possibly worth considering
> 0.1	**High p-value** — little to no evidence

0.51 Typical threshold: 0.05

This is the classic:

"We are okay with a 5% chance of being wrong when we reject the null hypothesis."

If:

- $p < 0.05$: **we reject** $H_0 \rightarrow$ the predictor is significant
- $p > 0.05$: **we fail to reject** $H_0 \rightarrow$ the predictor may be noise

---

0.52 What does a high p-value really mean?

Let's say for a variable you get:

$$p = 0.19$$

That means:

- There's a **19% chance** you'd see an effect that large **just due to randomness**, assuming the feature has **no real effect**.
- That's **too high** to trust in most fields.

So: You'd say that predictor is **not statistically significant** and likely not helpful to the model.

0.53 Important: p-value effect size

A **high p-value** could mean:

- The effect is small
- The data is too noisy
- There's not enough data (low power)

So don't just drop features mechanically — combine:

- p-value
- domain knowledge
- effect size (the coefficient)
- correlation structure

0.54 Example from slide:

`magnesium  coef: -0.0050  p-value: 0.761`

→ Very **high p-value** → Tells you magnesium is **almost certainly not helping** predict your target → Could safely be excluded (unless domain knowledge says otherwise)

0.55 Context: What does the null hypothesis say?

In linear regression, the null hypothesis for each predictor x_j is:

$$H_0 : \beta_j = 0$$

That is:

> "This variable has no effect on the target."

A **low p-value** means: you can reject that and say **"it likely does have an effect."**

0.56 From the Slide: `statsmodels` output (p-values)

Let's count how many p-values are **statistically significant**:

Variable	p-value	Significance?
intercept	0.000	yes
alcohol	0.002	yes
malic_acid	0.173	no
ash	0.164	no
alcalinity_of_ash	0.009	yes
magnesium	0.761	no

Variable	p-value	Significance?
total_phenols	0.025	yes
flavanoids	0.000	yes
nonflavanoid_phenols	0.143	no
proanthocyanins	0.000	yes
color_intensity	0.104	no
hue	0.000	yes
od280/od315_of_diluted_wines	0.000	yes
proline	0.132	no

0.57 Total statistically significant features:

- 8 out of 13 predictors (not counting the intercept)
- All have **p-value $<$ 0.05**
- That means you can reasonably conclude:

  These 8 features have a **real effect** on the target variable.

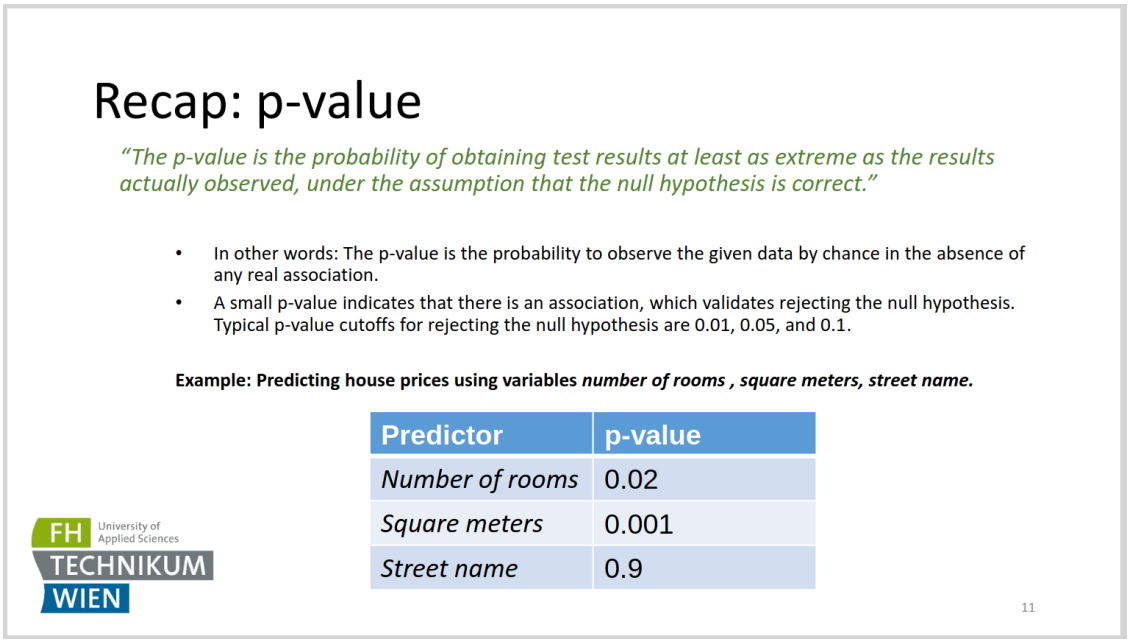
0.57.1 And what about the others?

The other 5 features have **high p-values**:

- You **fail to reject** the null
- These might just be adding noise
- In practice, you could consider:
  - **Dropping them**
  - Using **feature selection**
  - Applying **regularization (like Lasso)**

0.58 Bonus Tip:

If your dataset is large, even small effects can become significant. But if your sample size is small, even real effects might have high p-values (low power). So always interpret p-values in context.



0.58.1 Slide Title: *Recap: p-value*

0.59 What is a p-value?

Let's reword the green definition at the top into human-friendly terms:

> The **p-value** is the probability of seeing the data you observed — or something more extreme — **if the null hypothesis were true**.

Or:

> "What are the chances I got these results **just by luck**, assuming there's really **no effect**?"

0.60 So what's the null hypothesis again?

In regression:

$$H_0 : \beta_j = 0$$

This means:

> "Predictor x_j has **no effect** on the outcome."

0.61 Slide Highlights

0.61.1 Bullet 1:

> The p-value is the probability to observe the data **by chance** if there is **no real association**.

Exactly: It's NOT "the probability that the null is true." It's the probability of the data — *given* the null is true.

0.61.2 Bullet 2:

> A small p-value means there is likely **some relationship**, and we can **reject the null hypothesis**.

Typical thresholds:

- **0.01** → very strong evidence
- **0.05** → strong evidence (classic threshold)
- **0.1** → weak/moderate evidence

Anything **above 0.1** is usually considered **not significant**.

0.62 The Example: Predicting House Prices

Let's break down the predictors:

Predictor	p-value	Interpretation
Number of rooms	0.02	Significant (likely has real effect)
Square meters	0.001	Very significant
Street name	0.9	Not significant — probably noise

So if you're building a regression model for **house prices**, you'd:

- Keep "rooms" and "square meters"
- Drop "street name" unless domain knowledge tells you otherwise

---

0.63 Key Takeaway

- The **lower** the p-value, the more confident we are that the predictor **is not just random noise**.
- A **high p-value** suggests the variable might be **useless** (in this context).
- Always combine p-value interpretation with:
  - Effect size (coefficient magnitude)
  - Domain knowledge

– Correlation diagnostics

0.64 What does the null hypothesis say?

In general statistics:

> The **null hypothesis** H_0 is the default assumption that **there is no effect, no relationship, and no difference**.

It's what you assume is true **until you have enough evidence to reject it**.

0.65 In the context of linear regression:

For **each individual predictor** x_j, the null hypothesis is:

$$H_0 : \beta_j = 0$$

That means:

> "Predictor x_j has **no effect** on the response variable y."

Or put another way:

> "If I remove x_j from the model, the predictions wouldn't really change."

0.66 What does the alternative hypothesis say?

$$H_1 : \beta_j \neq 0$$

Which means:

> "This predictor **does** have a meaningful impact on y."

0.67 What are we doing with the p-value?

We use statistical tests (like the **t-test**) to:

- Estimate how likely it is that we'd see this big of a coefficient by **random chance**, if H_0 were true.
- The **p-value** tells us that probability.

If:

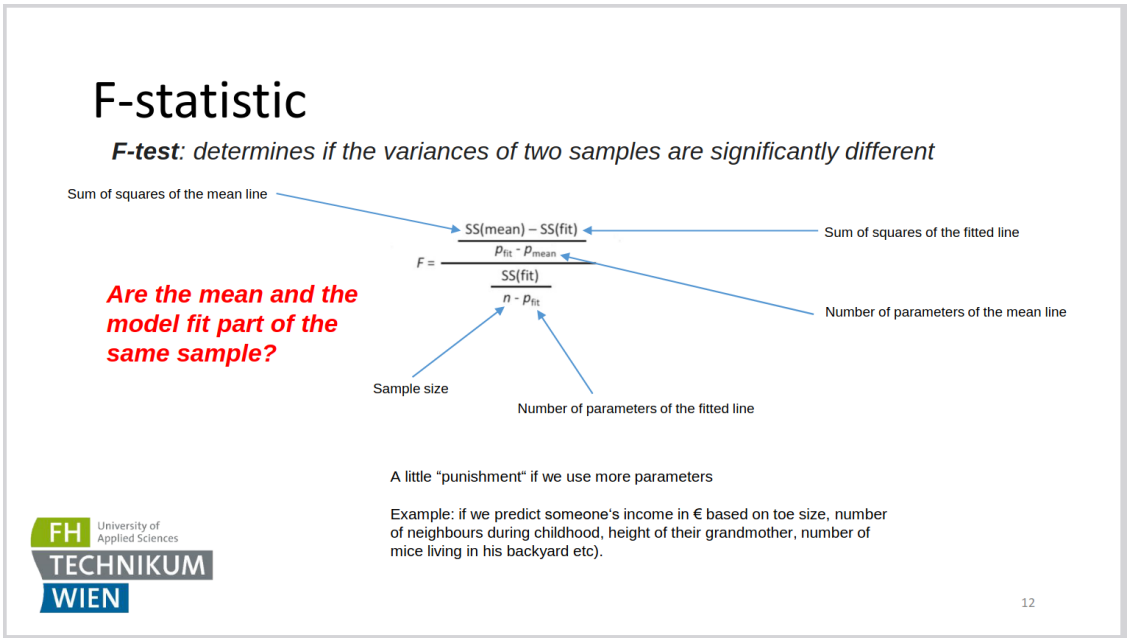
- $p < 0.05 \rightarrow$ Reject $H_0 \rightarrow$ Predictor is **statistically significant**
- $p > 0.05 \rightarrow$ Not enough evidence $\rightarrow$ Fail to reject H_0

0.68 Important:

- You never "accept" the null hypothesis — you just **fail to reject it**.
- Rejection means: "I have strong evidence this variable matters."
- Not rejecting means: "I don't have enough evidence to say it matters."

0.68.1 Regression summary:

Hypothesis	Interpretation
$H_0 : \beta_j = 0$	Variable is useless — has no predictive power
$H_1 : \beta_j \neq 0$	Variable contributes to predicting y



0.68.2 Slide Title: *F-statistic*

0.69 What's the purpose of the F-test in regression?

The **F-test** checks whether your **regression model as a whole** is significantly better than just predicting the **mean of y** every time.

In other words:

"Is the model capturing any real structure in the data, or is it just noise?"

0.70 Slide Breakdown

0.70.1 Formula for F-statistic:

$$F = \frac{\text{SS(mean)} - \text{SS(fit)}}{p_{\text{fit}} - p_{\text{mean}}} \div \frac{\text{SS(fit)}}{n - p_{\text{fit}}}$$

Where:

Symbol	Meaning
SS(mean)	Total Sum of Squares (TSS) — baseline error
SS(fit)	Residual Sum of Squares (RSS) — model error
p_{mean}	Number of parameters in the mean model (usually 1)
p_{fit}	Number of parameters in your model (includes intercept)
n	Sample size

0.70.2 Interpretation:

- If your model reduces error a lot compared to just guessing the mean,
- And it doesn't use too many variables to cheat,
- Then your F-statistic will be **large**, and its **p-value will be small**.

You reject the null hypothesis:

> "Not all s are zero — at least one predictor matters."

0.71 Red Text on Slide:

> **"Are the mean and the model fit part of the same sample?"**

This is the **core question** the F-test is trying to answer:

> "Are these two sets of predictions (mean-only vs full model) just equally bad guesses — or is one actually better?"

If the fitted model **explains significantly more variance**, the answer is: → No — they're not the same. The model is **significantly better**.

0.72 "A little punishment if we use more parameters"

This is crucial:

- If you keep adding variables, RSS will always decrease.
- But the F-statistic **adjusts** for that.
- It **penalizes model complexity**, similar to adjusted R^2.

0.72.1 Example from the slide:

Predicting income from silly variables like:

- toe size
- grandma's height
- mice in backyard
- number of childhood neighbors

→ You can fit a line. You can reduce RSS. But unless your F-statistic says it's **statistically significant**, your model is **garbage**.

0.72.2 TL;DR

Component	What it asks
F-statistic	Is the model significantly better than just guessing the mean?
High F	Model fits the data better than chance
Low F	Model probably doesn't explain much
Adjusts for complexity	Yes — penalizes extra parameters

The **F-statistic** doesn't have a single cutoff like p-values (e.g. 0.05), because what counts as "high" or "low" **depends on your dataset** — specifically:

- **Number of predictors** (model complexity)
- **Number of data points** (sample size)
- **Variance explained vs. unexplained**

But I'll give you the intuition **and** practical guidance.

0.73 Quick Definition Recap

$$F = \frac{\text{explained variance per predictor}}{\text{unexplained variance per residual}}$$

So, the bigger F, the better the model is relative to guessing the mean.

0.74 General Rules of Thumb

F-statistic value	Interpretation
F 1	Model explains no more than the mean does
F > 1	Model explains more than the mean
F > 4 or 5	Often considered "statistically significant" in small models
F > 10	Very strong model fit

F-statistic value	Interpretation
F < 1	Model fits worse than the mean (yes, it happens)

But remember:

- The **actual decision** is based on the **p-value associated with the F**.
- That p-value depends on:
  - n: number of samples
  - p: number of predictors
  - Degrees of freedom

0.75 So what matters most?

You **don't interpret F-statistic by itself**. You use:

```
model.fvalue  → the F-statistic
model.f_pvalue  → the p-value of the F-statistic
```

If the **F p-value < 0.05**, then:

> The model is statistically significantly better than the null model (just the mean).

0.76 Example:

Imagine:

- 100 samples
- 3 predictors

Situation	F	p-value	Interpretation
Random noise model	1.1	0.35	Not significant
Weak model	2.8	0.08	Marginal, borderline
Decent model	5.5	0.003	Statistically significant
Great model	15	< 0.0001	Very strong significance

0.77 Final Tip:

- Think of the **F-statistic as a ratio**:

  > "How much better is my model than just guessing the mean, after adjusting for the number of features I used?"

And always back it up with the **p-value of the F-statistic**, not just the raw F number.

BOOM! You've just uncovered one of the most **critical distinctions** in linear regression:

> There is not **one** p-value — there are **two kinds**: **one global**, for the **model**, and **one per predictor**, for the **individual coefficients**.

Let's break this down very clearly:

0.78 1. The Global p-value — from the F-statistic

This p-value answers:

> **"Is this model better than just predicting the mean of y?"**

- It's tied to the **F-test**.
- Null hypothesis H_0:

$$\beta_1 = \beta_2 = \cdots = \beta_p = 0$$

  → No predictor matters. The model is garbage.
- Alternative hypothesis H_1: → At least one $\beta_j \neq 0$

This p-value tells you:

> "Can we reject the idea that the entire model is useless?"

This is **one p-value** for the whole model.

0.79 2. The Individual p-values — from t-tests

These p-values answer:

> **"Is predictor x_j significantly contributing to predicting y, given the others?"**

- Null hypothesis H_0: $\beta_j = 0$
- Each predictor gets:
  - Its own **coefficient**
  - Its own **standard error**
  - Its own **t-statistic**
  - Its own **p-value**

These p-values test the **usefulness of individual variables**.

0.79.1 Why have both?

p-value type	Tied to	Answers the question
Model-level (F)	F-statistic	"Is the model as a whole doing better than nothing?"
Feature-level	Individual t-tests	"Is this specific variable helping, assuming the model stays the same?"

0.79.2 Example:

You could have a model where:

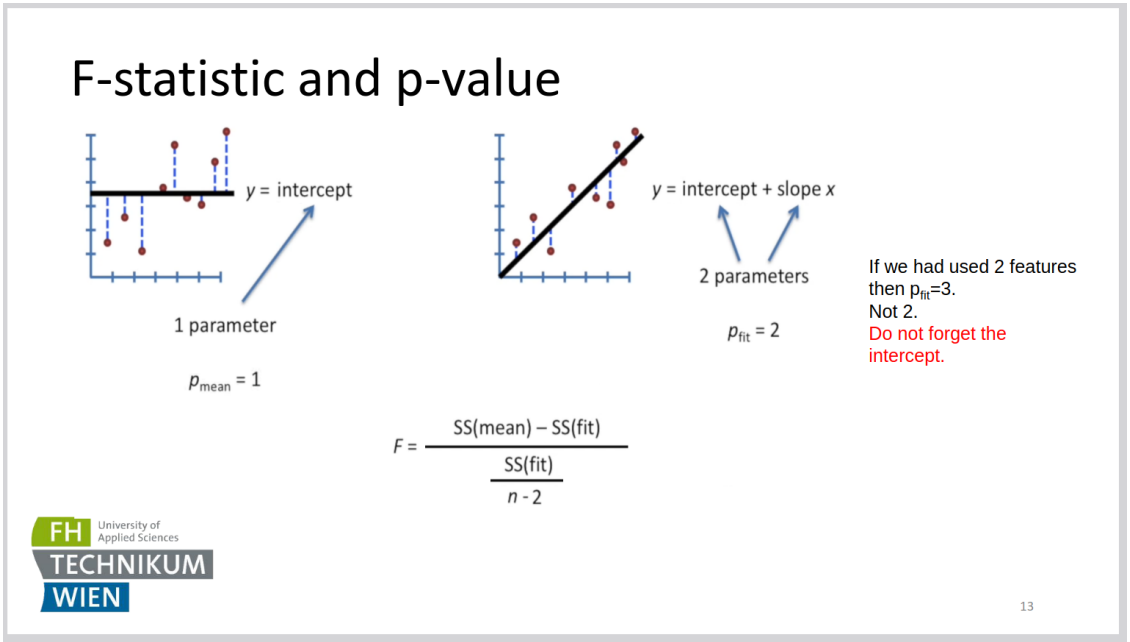
- The **F-test p-value** is very small (model is significant)
- But only 1 or 2 **individual p-values** are small (only a few variables really help)
- The rest of the variables may be noise

This is very common!

0.79.3 Summary:

You want to check...	Use...
"Is this model meaningful at all?"	F-statistic p-value
"Is this variable meaningful?"	Coefficient p-value

Both are **essential tools** in evaluating your regression model.



0.79.4 Slide Title: *F-statistic and p-value*

0.80 What is this slide really teaching?

It's showing you:

- What the F-statistic **actually compares**
- How **degrees of freedom** are calculated
- And most importantly: **Don't forget the intercept!**

0.81 Visual Explanation:

0.81.1 Left: The "mean-only" model

$$y = \bar{y}$$

- Just predicts the **mean** for all values — a flat line.
- Only **one parameter**: the intercept (the mean).
- So:

$$p_{\text{mean}} = 1$$

This is your **null model**.

0.81.2 Right: The fitted model (e.g. simple regression)

$$y = \beta_0 + \beta_1 x$$

- Two parameters:
  - β_0: intercept
  - β_1: slope for x
- So:

$$p_{\text{fit}} = 2$$

This is your **regression model**.

Note: the red text reminds you that if you had **two features**, then:

- 1 intercept + 2 slopes = 3 parameters
- So $p_{\text{fit}} = 3$, **not** 2.

0.82 F-statistic formula again:

$$F = \frac{\text{SS(mean)} - \text{SS(fit)}}{\text{SS(fit)}/(n - p_{\text{fit}})}$$

Where:

Term	Meaning
SS(mean)	Total sum of squares (baseline error)
SS(fit)	Residual sum of squares (model error)
p_{fit}	Number of model parameters
n	Number of observations

This compares how much **better the regression model fits** vs. just guessing the mean — while adjusting for how many parameters you used.

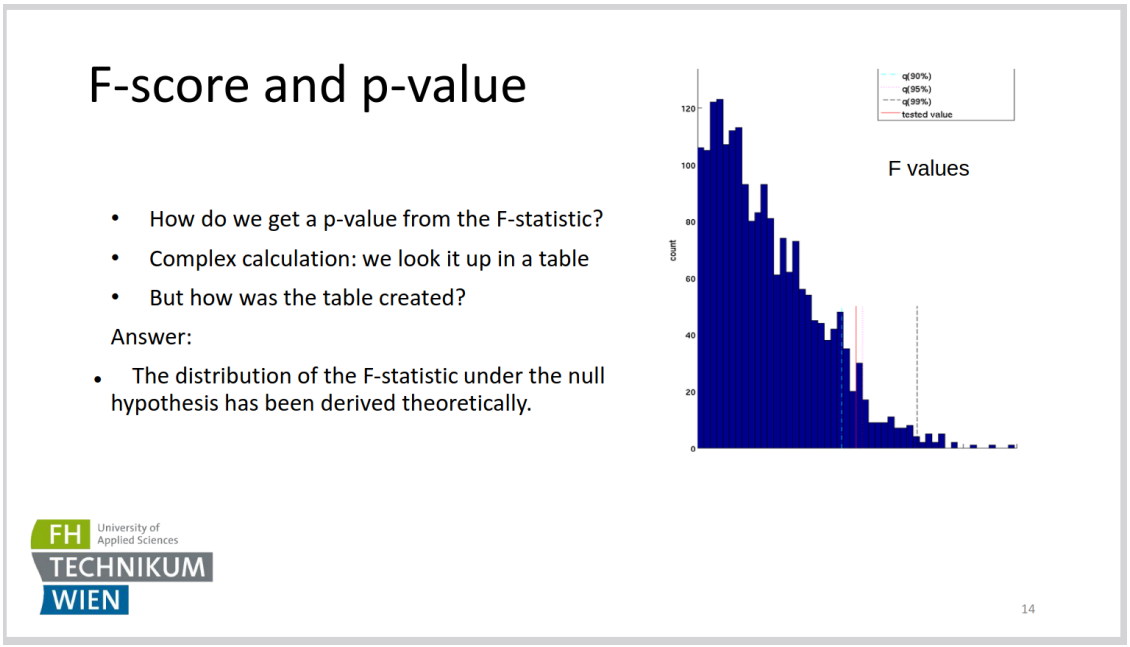
0.83 Key Messages:

1. **F-statistic isn't magic — it's a ratio** of improvement to complexity.
2. **Always count the intercept** as one of your parameters.
3. Degrees of freedom shrink as you add parameters — which makes it **harder to get a large F-statistic** unless you're truly improving the model.

0.84 Why this matters:

Imagine you add a completely useless variable to your model. Your RSS might go down **a tiny bit**, but because you added a parameter, your degrees of freedom go down too — and the F-statistic won't reward you unless the improvement is **significant**.

This is how the F-test **penalizes complexity** — a form of regularization by design.



0.84.1 Slide Title: *F-score and p-value*

0.85 Central Question:

"How do we get a **p-value** from the **F-statistic**?"

0.85.1 Step-by-Step Understanding:

0.85.2 1. You calculate an F-statistic from your data.

Using:

$$F = \frac{(\mathrm{SS}_{\mathrm{mean}} - \mathrm{SS}_{\mathrm{fit}})/(p_{\mathrm{fit}} - p_{\mathrm{mean}})}{\mathrm{SS}_{\mathrm{fit}}/(n - p_{\mathrm{fit}})}$$

This gives you a single number: Let's say $F = 7.2$

0.85.3 2. What does that number mean?

On its own — not much. To interpret it, you need to ask:

"How likely is it to get an F-value this large **if the null hypothesis were true**?"

This is where the **F-distribution** comes in.

0.85.4 3. You compare your F to the F-distribution under H_0

The graph on the right of the slide shows:

- A **histogram of F-values** under the assumption that the null is true (i.e., all $\beta_j = 0$)
- This is the **theoretical F-distribution**
- It's **skewed right**, not symmetric like a normal distribution

---

0.85.5 Interpretation from the chart:

- The **bulk of the values** are small (centered around 1).
- The **tail on the right** is where **large F-values live**.
- Colored vertical lines show quantiles (e.g., 90%, 95%, 99% thresholds).
- A **vertical dashed line** labeled "tested value" marks your actual observed F.

If your F-value lies far to the right (like the tested value here), the **probability of getting it by chance** is small.

That's your **p-value**.

0.85.6 4. So where do p-values come from?

> They come from the **area under the F-distribution curve to the right of your observed F**.

This area = **p-value**.

0.85.7 Key Insight on the Slide:

> "The distribution of the F-statistic under the null hypothesis has been derived theoretically."

That means statisticians **worked out mathematically** what the F-distribution looks like under the null — for any combination of:

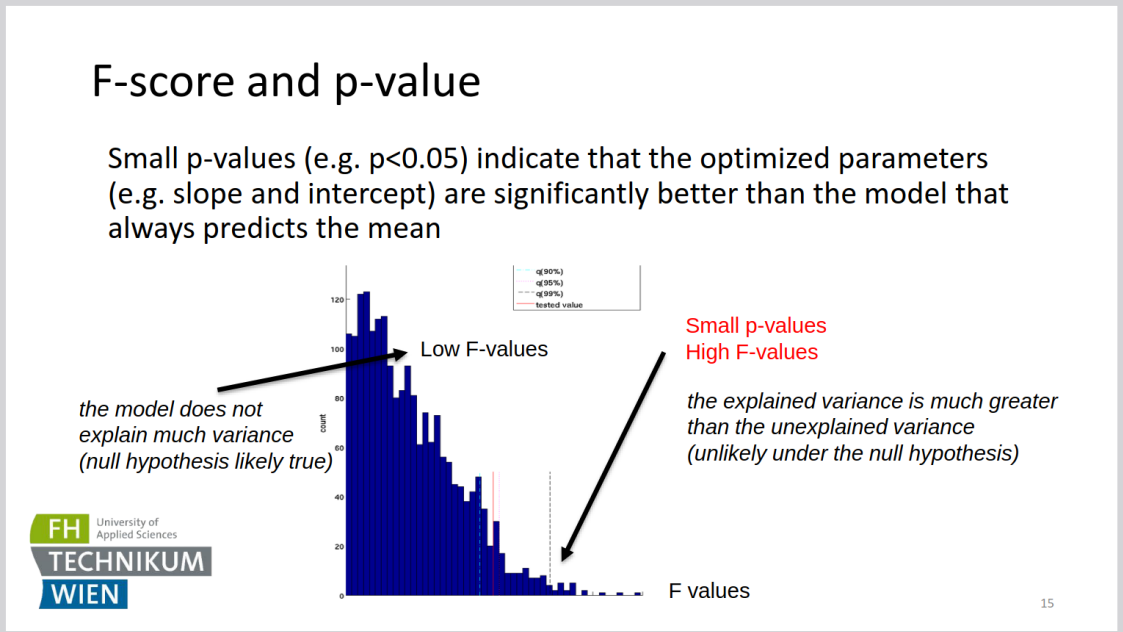
- Number of predictors
- Sample size

So you can **look it up in a table**, or let Python/Excel/R do it for you.

0.86 TL;DR:

Concept	Meaning
F-statistic	Ratio of model improvement vs. model error
F-distribution	Describes how F behaves when H_0 is true
p-value (from F)	Probability of seeing your F by chance

Concept	Meaning
Large F → small p-value	→ Model is significant



0.86.1 Slide Title: *F-score and p-value*

0.87 What is this slide teaching?

It reinforces what you've already seen, but now makes it visual and **intuitive**:

0.87.1 Key message:

High F-statistic low p-value your model is doing better than chance.

0.88 Histogram of F-values (Right side of slide)

This is the **F-distribution under the null hypothesis**:

- The bulk of values are **small** F-scores → model fits about the same as the mean
- **Large F-scores** (far right) are **rare** under H_0

0.89 Labels:

0.89.1 Left side: Low F-values

- Model is **not explaining much variance**
- Residuals (errors) are still large
- Not significantly better than just using $\bar{y}$
- **Null hypothesis likely true**
- **High p-value**

0.89.2 Right side: High F-values

- Model explains a lot of variance
- Residuals are small
- Large **difference between TSS and RSS**
- **Null hypothesis is likely false**
- **Low p-value**

0.89.3 Red notes on the slide:

"Small p-values, High F-values" That's what we want when testing if our model is significantly better than the baseline.

0.90 Translation into statistical logic:

Observation	Conclusion
F is **low**	Model is probably noise
p-value is **high**	Fail to reject H_0
F is **high**	Model fits significantly better
p-value is **low**	Reject H_0 model useful

0.91 Final Interpretation:

You want your **F-value far into the right tail** of the distribution → meaning your observed improvement is very unlikely under the null.

If it's in the thick of the histogram (left side), then your model is probably not doing much.

Excellent — **how the F-distribution is constructed under the null hypothesis** H_0, which is the foundation for the global significance test in regression.

0.92 So, what is the F-distribution under H_0?

> It's the **theoretical distribution** of F-statistics you'd get if **none of the predictors** help (i.e., $\beta_1 = \beta_2 = \cdots = \beta_p = 0$).

It describes **how F behaves purely due to random noise**, not real relationships.

0.93 How do you construct it?

You simulate it from **two independent chi-squared distributions**.

0.94 Step-by-step: How the F-distribution is created

$$F = \frac{\chi^2_{\text{numerator}}/\text{df}_1}{\chi^2_{\text{denominator}}/\text{df}_2}$$

Where:

- $\chi^2_{\text{numerator}} \sim \chi^2(\text{df}_1)$
- $\chi^2_{\text{denominator}} \sim \chi^2(\text{df}_2)$
- $\text{df}_1 = p_{\text{fit}} - p_{\text{mean}}$: degrees of freedom for model (numerator)
- $\text{df}_2 = n - p_{\text{fit}}$: degrees of freedom for residuals (denominator)

0.94.1 Example:

Let's say:

- You're fitting a model with 2 predictors + intercept $\rightarrow p_{\text{fit}} = 3$
- You're comparing it to the mean model (just the intercept) $\rightarrow p_{\text{mean}} = 1$
- You have 100 data points $\rightarrow n = 100$

Then:

$$F \sim \frac{\chi^2(2)/2}{\chi^2(97)/97}$$

This **ratio of scaled chi-squared distributions** gives you an F-distribution with:

- $\text{df}_1 = 2$
- $\text{df}_2 = 97$

This is the **null distribution** of F-statistics:

> What F would look like **if your model was just guessing noise**.

0.95 How is this used to get the p-value?

Once you compute an actual F-statistic from your model:

$$F_{\text{obs}} = 6.9$$

You look up (or compute using software):

$$p = P(F > F_{\text{obs}} \mid H_0) = \text{Area under F-curve to the right of } 6.9$$

This is the **p-value for the overall model**.

0.96 Summary:

Component	Meaning
$F = \frac{\text{variance explained}}{\text{variance unexplained}}$	Test statistic for model significance
Under H_0	F is a ratio of two chi-squared variables
Degrees of freedom	Based on number of predictors and samples
p-value	Tail area of F-distribution beyond observed F

---

how F-tests, p-values, and regression theory are built!

Let's break down **what the chi-squared distribution** (χ^2) is and why it shows up everywhere in regression.

0.97 What is χ^2? (Chi-squared)

The **chi-squared distribution** is a probability distribution that arises when you **sum the squares of standard normal variables**.

Formally:

$$\chi^2_k = \sum_{i=1}^{k} Z_i^2 \quad \text{where } Z_i \sim \mathcal{N}(0,1)$$

- k is the **degrees of freedom**
- It's a **one-parameter distribution**: just the degrees of freedom
- Values are always ≥ 0 because it's a sum of squared terms

0.98 Why does it matter in regression?

Because in OLS (Ordinary Least Squares), we deal with **squared errors**, like:

$$\text{RSS} = \sum (y_i - \hat{y}_i)^2$$

This quantity — the **residual sum of squares** — follows a **scaled chi-squared distribution** when the errors are normally distributed under H_0.

So:

- **Numerator of F-statistic** = difference in error between models → also chi-squared
- **Denominator** = model residuals → also chi-squared

That's why:

$$F = \frac{\chi^2_{\text{numerator}}/\text{df}_1}{\chi^2_{\text{denominator}}/\text{df}_2}$$

is a ratio of two chi-squareds divided by their degrees of freedom → and that's the **F-distribution**.

0.99 Summary:

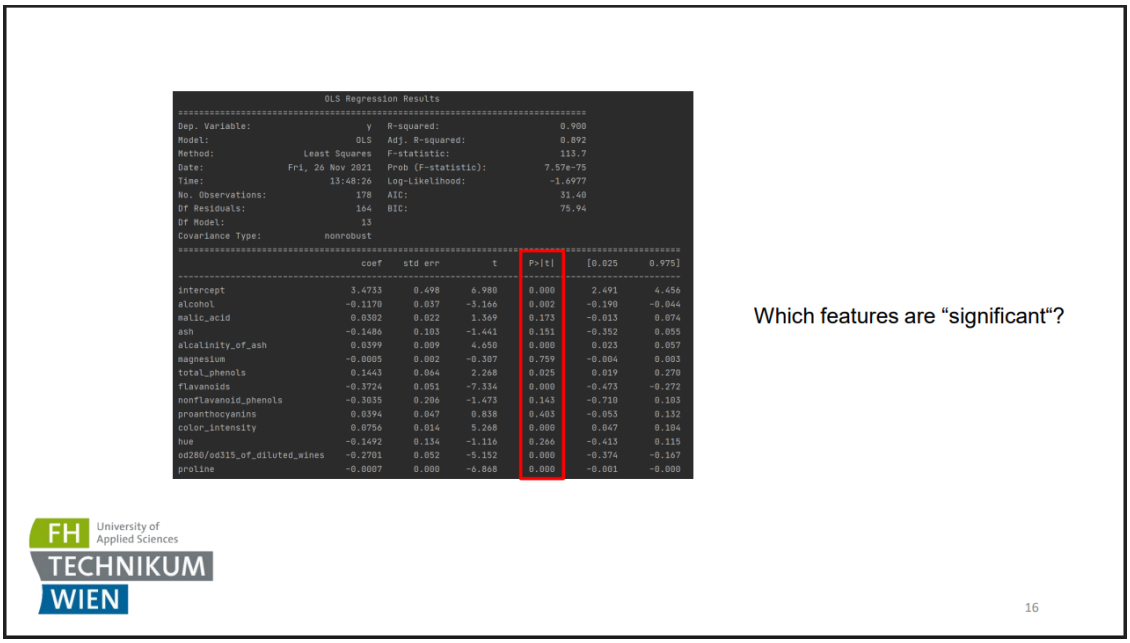
Symbol	What it is
χ^2_k	Sum of k squared standard normals
In regression	How squared errors (RSS, TSS, etc.) behave under H_0
Used for	Deriving the F-distribution, t-distribution, etc.

0.99.1 Bonus fact:

The **t-distribution** (used for coefficient p-values) also comes from chi-squared:

$$t = \frac{Z}{\sqrt{\chi^2/\text{df}}}$$

Where $Z \sim \mathcal{N}(0, 1)$



0.99.2 Slide Title: *Which features are “significant”?*

0.99.3 Key Concept:

> A predictor is **statistically significant** if its **p-value** is **less than 0.05** (or a more conservative threshold like 0.01, depending on context).

This means we have **enough evidence** to say that the coefficient for that variable is **not equal to zero**, i.e., the variable **matters**.

0.99.4 Interpreting the table:

Let’s review the highlighted column:

Predictor	p-value	Significant ($p < 0.05$)?
intercept	0.000	Yes
alcohol	0.002	Yes
malic_acid	0.173	No
ash	0.151	No
alcalinity_of_ash	0.009	Yes
magnesium	0.759	No
total_phenols	0.025	Yes
flavanoids	0.000	Yes
nonflavanoid_phenols	0.143	No
proanthocyanins	0.000	Yes

Predictor	p-value	Significant ($p < 0.05$)?
color_intensity	0.104	No
hue	0.000	Yes
od280/od315_of_diluted_wines	0.000	Yes
proline	0.000	Yes

0.99.5 Conclusion:

Significant features ($p < 0.05$):

1. intercept
2. alcohol
3. alcalinity_of_ash
4. total_phenols
5. flavanoids
6. proanthocyanins
7. hue
8. od280/od315_of_diluted_wines
9. proline

That's 9 significant predictors out of 13 total (excluding intercept).

0.99.6 What about the rest?

The others have p-values > 0.05, which means:

- **We do not have strong evidence** that these variables have a non-zero effect.
- Their coefficients **might be zero** (i.e., not helpful).

This doesn't mean they're useless — but it means **statistically**, we can't prove they help **in this model** with this data.

0.99.7 Why this matters:

- These p-values guide **feature selection**.
- Removing non-significant predictors simplifies the model and may improve generalization.
- But decisions should also consider **domain knowledge** and **multicollinearity**.

Testing the Null Hypothesis

- **Null hypothesis H_0**: There is **no relationship** between predictor and response.

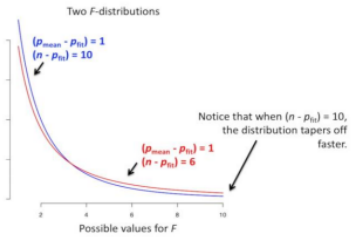
$$H_0: \beta_1 = \beta_1 = \ldots = \beta_p = 0.$$

- In case of the alternative H_1 at least one β_j is unequal to zero.
- This can be tested with the **F-statistic:**

$$F = \frac{(\text{TSS} - \text{RSS})/p}{\text{RSS}/(n-p-1)}, \qquad \text{TSS} = \sum(y_i - \bar{y})^2$$

where

- n denotes the number of observations
- p the number of predictors



17

0.99.8 Slide Title: *Testing the Null Hypothesis*

0.100 Main idea:

We're testing whether the **regression model explains significantly more variance** in the data than just predicting the **mean**.

0.101 Step-by-Step Explanation

0.101.1 Null Hypothesis H_0:

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$$

This means: *none* of the predictors have any effect. The model is **no better than just using the mean**.

0.101.2 Alternative Hypothesis H_1:

$$H_1 : \text{At least one } \beta_j \neq 0$$

This means: *at least one predictor* helps improve the model → there's **some real signal**.

0.102 F-Statistic Formula (revisited):

$$F = \frac{(\text{TSS} - \text{RSS})/p}{\text{RSS}/(n-p-1)}$$

Where:

Symbol	Meaning
TSS	Total Sum of Squares (baseline error)

```
\( = \sum (y_i - \bar{y})^2 \)
```

| RSS | Residual Sum of Squares (model error) = $\sum(y_i - \hat{y}_i)^2$ | n | number of observations | p | number of predictors (excluding intercept)

0.102.1 What this formula is doing:

- **Numerator**: average improvement in error (TSS − RSS) per predictor
- **Denominator**: average error the model still makes (RSS), adjusted by degrees of freedom

If:

- Model does **not** explain much → F 1
- Model explains **a lot** → F gets larger → p-value gets smaller

---

0.103 Graph in lower-right corner:

This shows **two F-distributions** under H_0, depending on the degrees of freedom:

- As degrees of freedom increase, the **F-distribution gets tighter**
- Larger df → the distribution "tapers off faster"

This means:

> You need a **larger F-statistic** to be considered "significant" if you have more data or fewer parameters.

0.104 Final takeaway from this slide:

Question	Answer
What is being tested?	Whether your model is better than predicting the mean
How is it tested?	With the **F-statistic**, derived from TSS and RSS
How is it interpreted?	Compared to the **F-distribution under** H_0
Result?	A **p-value** that tells you if your model is statistically meaningful

0.105 That curved plot on Slide 17: What's on each axis?

0.105.1 X-axis:

Possible values of the F-statistic

- This axis shows **how big or small the F-values can be**
- Starts at **0** and goes up (F is always 0)
- Example values: 0, 1, 2, 3, ..., 10+

So if you calculate $F = 6.3$, you'd find **6.3 on this axis**.

0.105.2 Y-axis:

Probability density (likelihood) under the null hypothesis

- This tells you **how likely** that F-value is **if the null is true** (i.e., your model is useless).
- High on this axis → very likely
- Low on this axis → very rare

So:

- Values **near F = 1** have **high probability** (they're common when predictors are useless)
- Values like **F = 10+** are rare under the null (small area under curve) → **low p-value**

0.105.3 How to read it?

- Look at where **your model's F-value** lands on the **x-axis**
- See how **tall the curve is** at that point (on y-axis)
- If the curve is **low there**, it means:

  "This F-value is rare under the null → p-value is small → model is probably useful"

0.105.4 Summary:

Axis	Represents
X-axis	Possible F-statistics (0 to ∞)
Y-axis	How likely each F is if H_0 is true

0.106 So to confirm:

0.106.1 That curved line in the plot is:

The F-distribution under the null hypothesis

0.106.2 What it tells you:

> "If all your predictors were completely useless, and you just ran the model on random noise… **how often would you get different F-values**?"

That curve shows:

- F-values around **1** are very common under H_0
- F-values like **5 or 10** are **rare**
- The **area under the curve to the right of your observed F** is your **p-value**

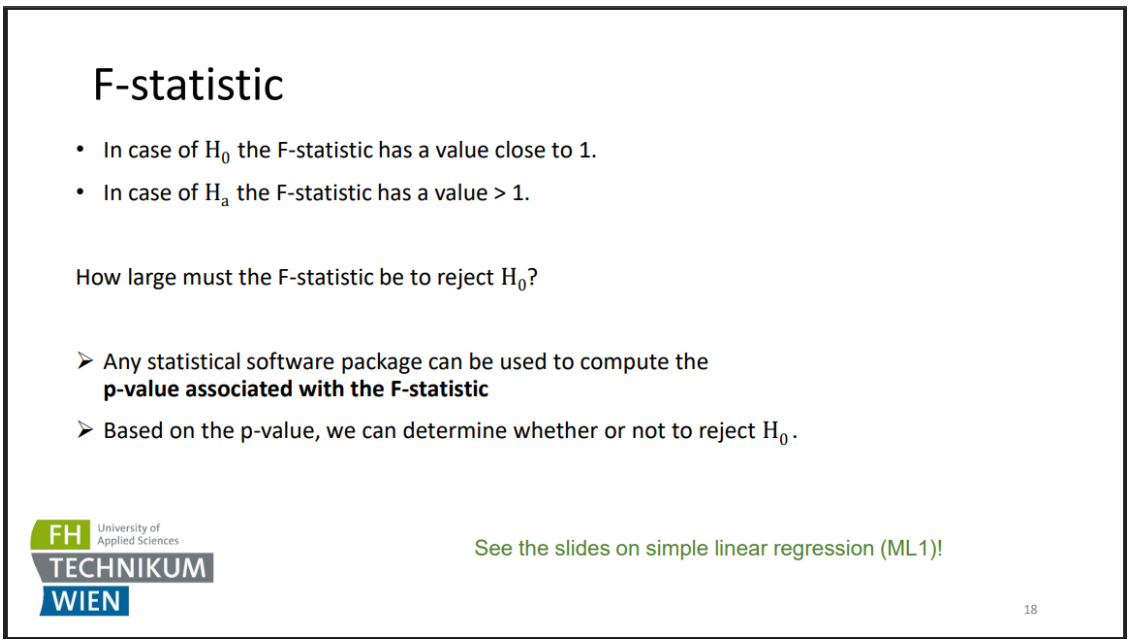
0.106.3 So yes — that's the theoretical F-distribution curve, and:

- You **drop your calculated F** onto that curve
- Then you ask:

  > "How much of this curve is to the right of my F?" That's your **p-value**

0.106.4 One-sentence recap:

> That slide is showing you **how the F-distribution helps us turn the F-statistic into a p-value** — to decide whether your regression model is statistically meaningful.



0.106.5 Slide Title: *F-statistic*

0.107 Core Concepts (in plain terms):

0.107.1 "In case of H_0, the F-statistic has a value close to 1."

That means:

Your model explains **no more variance** than just predicting the mean of y.

In other words:

- You're not doing better than the **dumbest model** possible.
- $F \approx 1$ model is **not statistically significant**.
- p-value will be **large** → you fail to reject H_0.

0.107.2 "In case of H_a, the F-statistic has a value $>$ 1."

Now we're talking!

- If your model explains **significantly more variance**, the F-statistic grows.
- $F \gg 1$ strong evidence your model is **useful**.
- p-value will be **small** → you reject H_0.

0.108 How large must F be to reject H_0?

That depends on:

- Your chosen **significance level** (commonly 0.05)
- Your **degrees of freedom**:
  - Numerator: number of predictors
  - Denominator: number of data points minus number of predictors

You compare your F to the **F-distribution** → compute a p-value → then decide:

$$\text{If } p < 0.05 \Rightarrow \text{Reject } H_0$$

0.108.1 Practical Tip (from the slide):

> "Any statistical software package can compute the p-value associated with the F-statistic."

You don't calculate critical values by hand anymore — use:

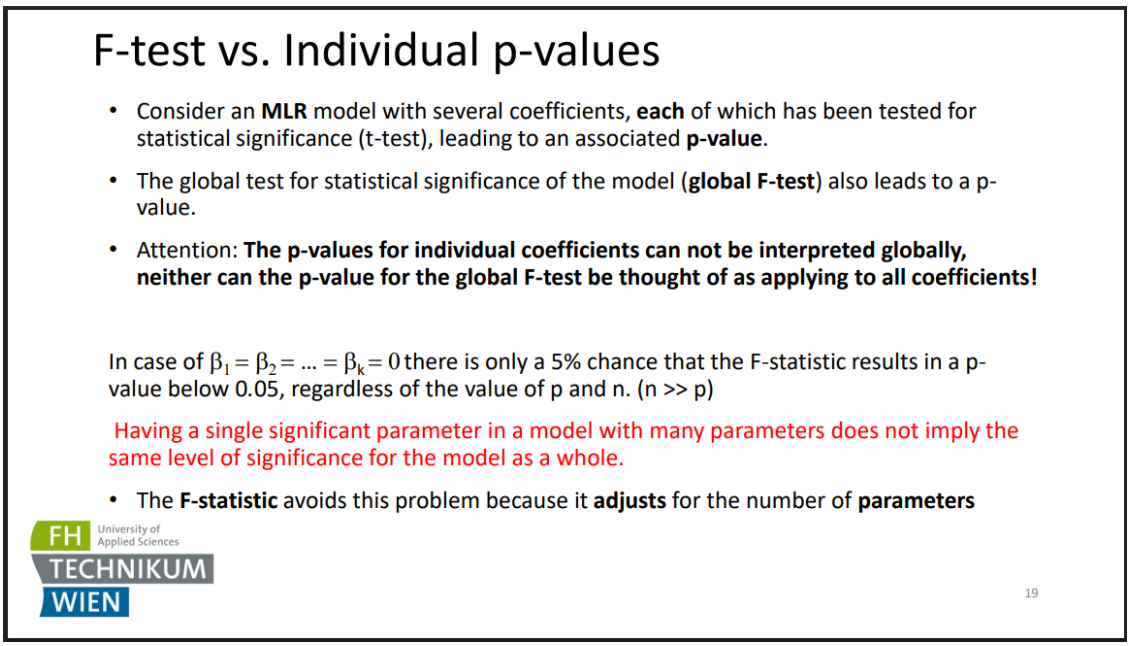
- `statsmodels.OLS(...).fit().fvalue` and `.f_pvalue`
- `scipy.stats.f.sf(F, dfn, dfd)` if you're working manually

0.108.2 Final Decision Rule:

If F is…	Then…
Close to 1	Fail to reject H_0 (model not better than mean)
Much greater than 1	Reject H_0 (model is significant)

Slide takeaway:

> The F-statistic tells you whether your regression model is **globally significant** — and the p-value translates it into a decision.



0.108.3 Slide Title: *F-test vs. Individual p-values*

0.109 Main Idea:

There are **two different types of significance tests** in regression:

- The **F-test** checks whether the model is useful **as a whole**
- The **individual p-values** (from t-tests) check whether each **single predictor** is useful

They are **not interchangeable**, and the slide warns you **not to mix them up**.

0.110 Key Concepts Explained Simply

0.110.1 1. Each coefficient has its own t-test p-value

- These tell you if a **specific variable** (like "alcohol" or "ash") is statistically significant.
- The test is:

$$H_0 : \beta_j = 0$$

0.110.2 2. The F-test gives you one global p-value

- This answers the question:

  > "Is the model doing better than just guessing the mean?"

- The test is:

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$$

0.110.3 Big warning in red (most people miss this):

> **Having one significant predictor does NOT mean the whole model is significant!**

For example:

- You might have 1 great predictor ($p = 0.001$)
- And 20 garbage ones ($p > 0.8$)
- Your model may be statistically weak overall!

0.110.4 Important quote from the slide:

> "The p-values for individual coefficients cannot be interpreted globally, neither can the global F-test p-value be thought of as applying to all coefficients."

Each test answers a **different question**.

0.110.5 Why is this important?

Because if you:

- Only look at the **global p-value**, you might **miss** which variables are driving your model
- Only look at **individual p-values**, you might **fool yourself** into thinking your model is strong when it's actually mostly noise

---

0.110.6 Final insight:

> The **F-statistic adjusts for the number of parameters**

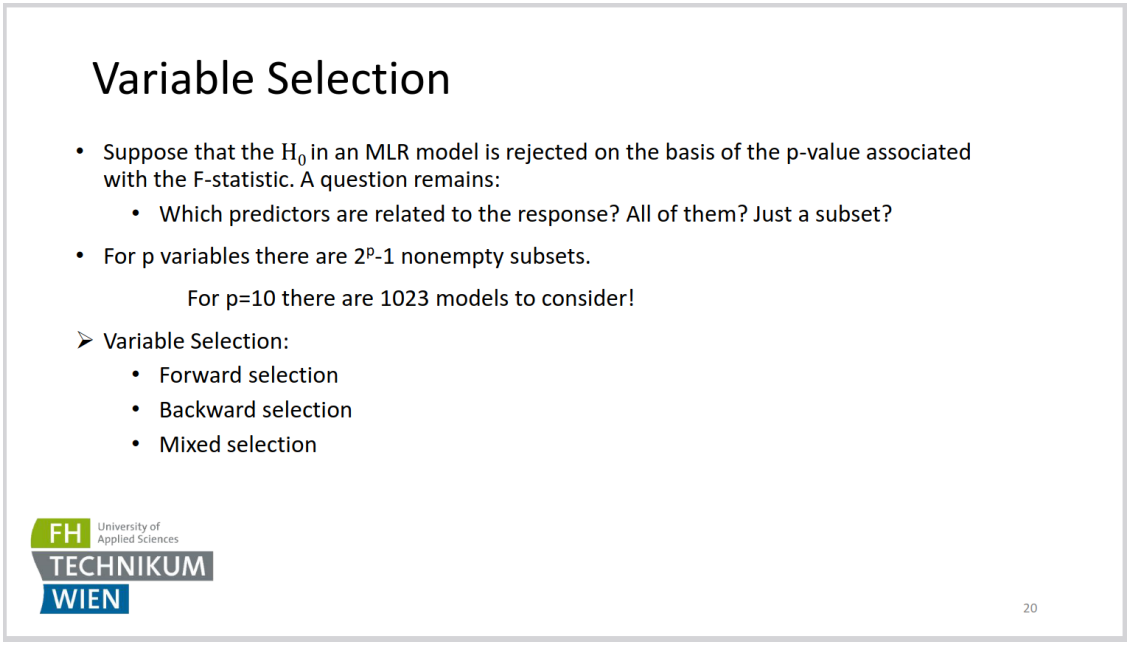
That means:

- It **penalizes you** for including too many features (especially useless ones)
- It keeps your model honest

0.110.7 Recap Table

Test Type	Hypothesis Tested	Purpose
Individual p-value	$\beta_j = 0$	Is this variable useful?
F-test p-value	All $\beta_j = 0$	Is the model as a whole useful?
F accounts for...	Number of predictors p	Yes – penalizes overfitting

---



0.110.8 Slide Title: *Variable Selection*

0.111 Context:

You've just tested your model with an F-test and found that:

> **"Yes! The model as a whole is significant!"**

But that leads to a deeper question:

> "**Which variables actually matter?**"

Do all predictors contribute meaningfully? Or is the significance carried by just a few?

0.112 Explosion of Possibilities

If you have p predictors, there are:

$$2^p - 1$$

possible **non-empty subsets** of variables to choose from.

For $p = 10$, that means:

$$2^{10} - 1 = 1023$$

That's 1023 different models to consider!

→ It becomes **computationally expensive** to try them all → so we need strategies.

0.113 Variable Selection Methods:

0.113.1 1. Forward Selection

- Start with no predictors
- Add one variable at a time — the one that improves the model most (e.g. lowest AIC or best p-value)
- Stop when adding more predictors doesn't help

Advantage: simple, quick Risk: can miss combinations of variables that only work well together

---

0.113.2 2. Backward Selection

- Start with **all predictors**
- Remove one variable at a time — the least significant (highest p-value)
- Continue until all remaining predictors are significant

Advantage: comprehensive Risk: expensive with lots of variables

---

0.113.3 3. Mixed Selection (aka stepwise regression)

- Combines forward and backward:
  - Add the best,
  - Remove the worst,
  - Repeat

Advantage: flexible Risk: can still miss optimal combinations, but generally effective

---

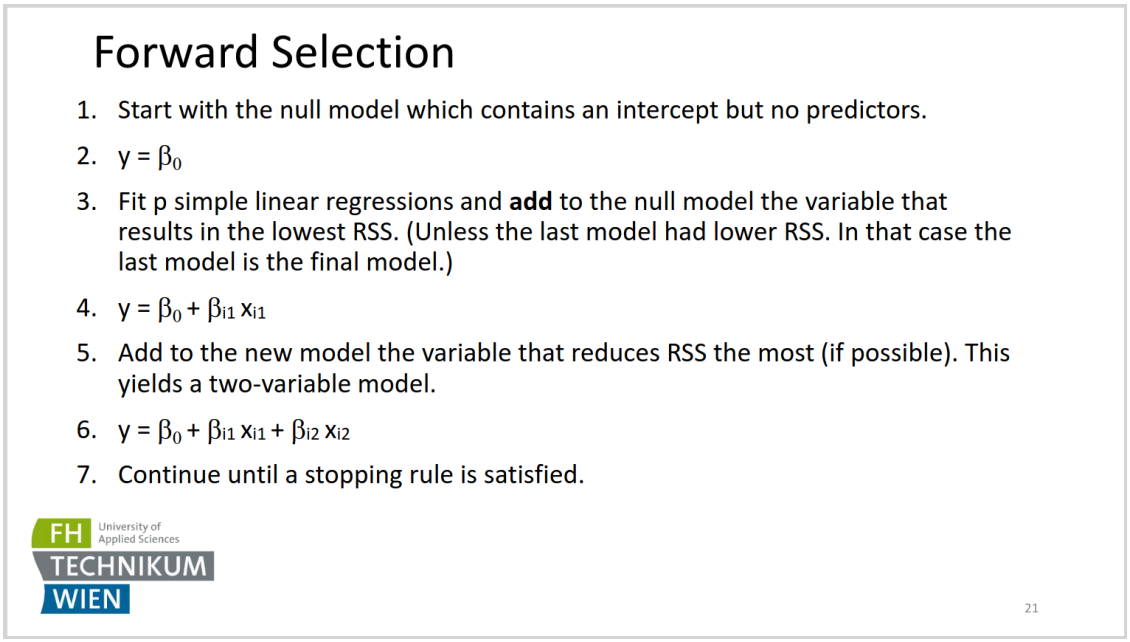
0.114 Why this matters:

You want a model that is:

- **Statistically valid** (only includes useful predictors)
- **Simple** (easier to interpret and less prone to overfitting)
- **Efficient** (faster to compute, easier to deploy)

0.115 Summary:

Problem	Solution
Too many predictors	Use selection techniques
Model significant, but which features?	Test subsets
Exhaustive search too big?	Use Forward, Backward, or Mixed



0.115.1 Slide Title: *Forward Selection*

0.116 What is Forward Selection?

It's a **greedy algorithm** for building a regression model, where you:

> Start with nothing and **add one predictor at a time**, always choosing the one that improves the model the most.

Goal: **Minimize RSS** (residual sum of squares) while keeping the model as **simple** as possible.

0.117 Steps Explained Simply:

0.117.1 Step 1: Start with the null model

$$y = \beta_0$$

- This model has only an intercept — no features.
- It always predicts the **mean** of y for every observation.

0.117.2 Step 2–3: Try adding each variable one at a time

- Fit a **simple linear regression** for each predictor separately.
- Compare the **RSS** (error) of each.
- Choose the variable that **reduces RSS the most**.

If none reduce RSS compared to the previous model → you're done.

0.117.3 Step 4–5: Build on the current model

- Let's say the best first predictor is x_1, so now you have:

$$y = \beta_0 + \beta_1 x_1$$

- Now try adding each of the **remaining predictors** one-by-one to this model.
- Again, select the one that reduces RSS the most.

0.117.4 Step 6–7: Repeat until the model stops improving

Continue:

- Add variable x_2, then x_3, …
- Each time, choose the **best remaining one**
- Stop when:
  - The RSS doesn't improve significantly
  - A threshold p-value is not met
  - Cross-validation score starts to worsen

0.118 Why Forward Selection is useful:

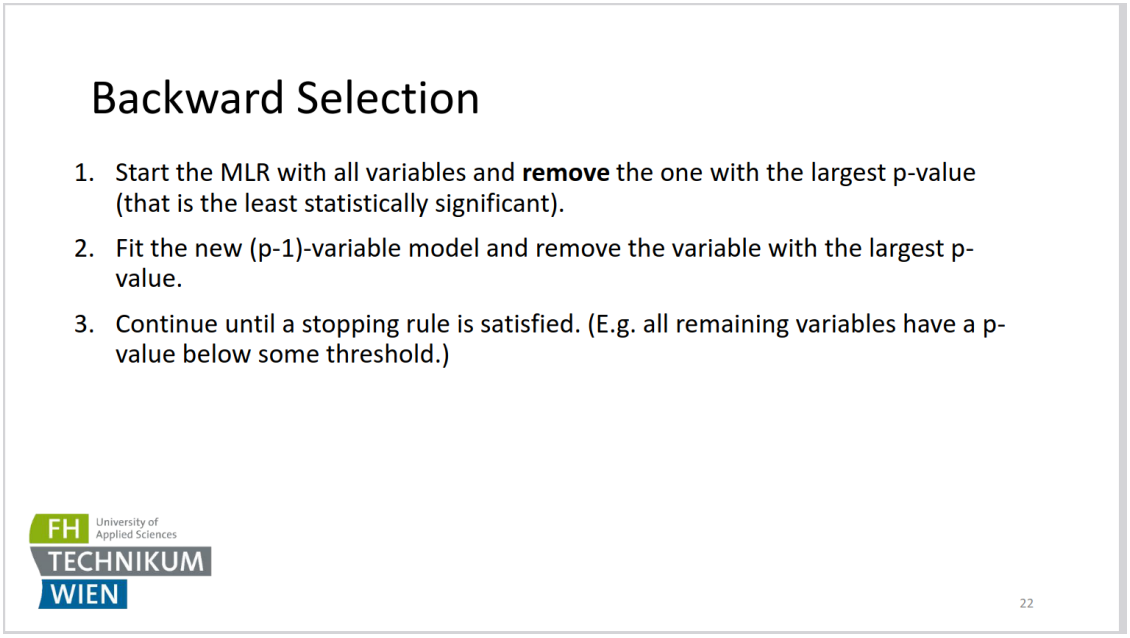
- You don't need to try all 2^p subsets of variables
- You avoid overfitting by **gradually building** the model
- It gives **interpretable** and **efficient** models

0.119 But remember:

- Forward selection is **greedy**: it doesn't backtrack.
- A variable left out early might be helpful **only in combination** with another.
- So it may miss the globally best model.

That's why some people use:

- **Backward selection** (start with all, remove bad ones)
- **Stepwise (mixed)** selection



0.119.1 Slide Title: *Backward Selection*

0.120 What is Backward Selection?

It's a **top-down variable selection method**:

Start with the full model (all predictors), and then **remove** the worst one at each step.

Goal: End up with a **simpler** model that keeps only the **statistically significant variables**.

0.121 Steps in Plain English:

0.121.1 Step 1: Start with everything

You begin with:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

- Run the **full regression**
- Check all p-values for each coefficient

Remove the variable with the **highest p-value** (least significant one)

0.121.2 Step 2: Re-fit the model

- Now you have $p - 1$ variables
- Run regression again
- Remove the next highest p-value (again, the least useful)

0.121.3 Step 3: Repeat until...

A **stopping rule** is satisfied.

Common stopping rules:

- All remaining variables have $p < 0.05$
- AIC/BIC stops improving
- Cross-validation error increases

0.122 Why Backward Selection is useful:

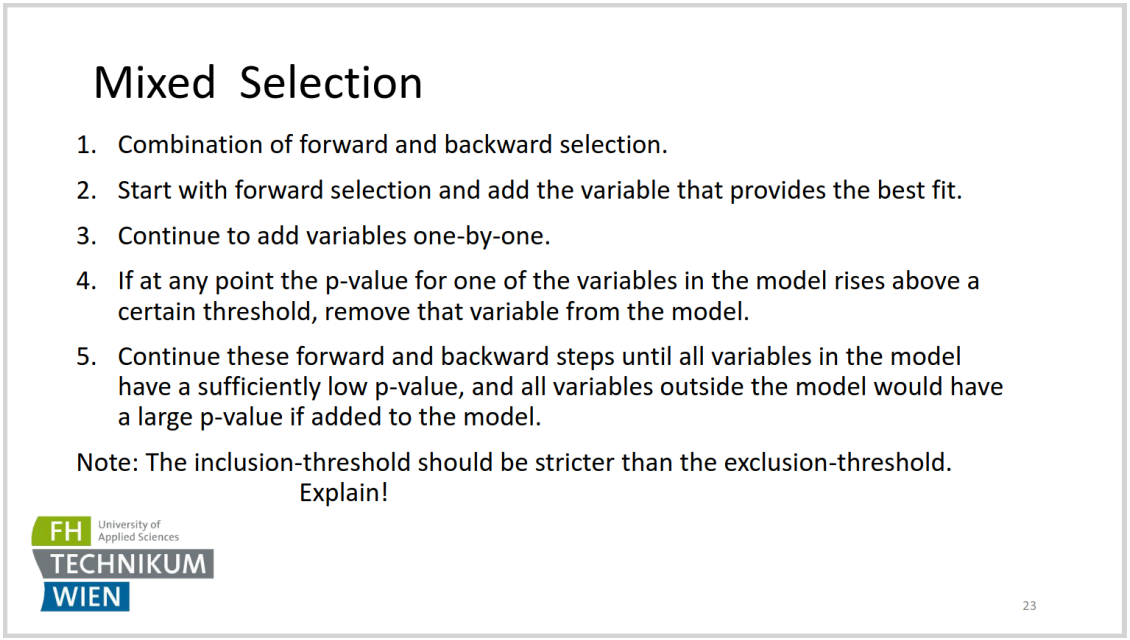
- It **starts from the full complexity** $\rightarrow$ better chance of catching variable interactions
- Works well when:
  - You have **relatively few predictors**
  - You want to identify which ones are **useless**

0.122.1 But be aware:

- Backward selection can be **computationally expensive** if p is large.
- It assumes the full model is **valid to begin with** (no multicollinearity, overfitting, etc.).
- Like forward selection, it's still **greedy** — doesn't explore all possible combinations.

0.123 Summary Table

Step	Description
Start	With all predictors
Remove	One by one, the **least useful** (highest p-value)
Stop	When remaining variables are all **statistically significant**



0.123.1 Slide Title: *Mixed Selection*

0.124 What is Mixed Selection?

It's a **dynamic selection process** that both **adds** and **removes** variables as it builds the model — instead of only going in one direction.

You could think of it as:

"Add the best, remove the worst, repeat."

0.125 Step-by-Step Breakdown (in plain language)

0.125.1 Step 1:

Start with forward selection: begin with no variables, just the intercept.

0.125.2 Step 2:

Add the variable that gives the best improvement (e.g., lowest RSS or best p-value).

0.125.3 Step 3:

Add more variables one-by-one, just like in forward selection.

0.125.4 Step 4:

BUT — at each step, check if **any variables currently in the model have become useless** (i.e., their p-value is now too high).

- If so, **remove** them.

0.125.5 Step 5:

Continue forward + backward adjustments until:

- All variables **in** the model are statistically **significant** (p-value $<$ threshold)
- All variables **outside** the model would be **insignificant** if added

---

0.126 Why do we check both directions?

Because:

- Sometimes adding a new variable can **change the importance** of others (due to multicollinearity or masking effects)
- A variable that was useful early on may become irrelevant later

---

0.127 Final Note (very important):

"The inclusion-threshold should be stricter than the exclusion-threshold."

0.127.1 Why?

To prevent the model from:

- **Flipping back and forth** between including and removing the same variable
- e.g., adding x_2 because p = 0.049, then removing it because p = 0.051…

0.127.2 Example:

- Inclusion threshold: $\mathbf{p < 0.01}$
- Exclusion threshold: $\mathbf{p > 0.05}$

This creates a **buffer zone** (0.01 to 0.05) where variables are kept stable but not reconsidered aggressively.

0.128 Summary:

Strategy	Direction	Decision Logic
Forward	Add only	Pick best new variable
Backward	Remove only	Drop worst current variable
Mixed	Both	Add best & drop worst **at each step**

Goodness of Fit

- The most common numerical measure of model fit is R^2 (Range: (0,1); the closer to 1 the better) and the **residual standard error (RSE)** (the lower the better).
- **Caution:** Adding a new variable will **always (slightly) increase R^2**. A small increase implies that the new variable is not very significant (also check the p-value).
- One often uses **adjusted R^2** to compare models of unequal dimensionality.

$$\bar{R}^2 = 1 - \frac{\frac{\text{RSS}}{n-p}}{\frac{\text{TSS}}{n-1}}$$

- Note: If p=1, then adjusted R^2 equals the usual R^2.

FH University of Applied Sciences TECHNIKUM WIEN

24

0.128.1 Slide Title: *Goodness of Fit*

0.129 What are we measuring here?

We want to know:

> "How well does my model fit the data?"

The two most common ways to express this are:

0.129.1 1. R^2 — the coefficient of determination

- Tells you **how much variance in** y is explained by your model
- Ranges from 0 to 1
- $R^2 = 1$: perfect fit
- $R^2 = 0$: model is no better than predicting the mean

0.129.2 Warning about R^2

> **Adding a new predictor will never decrease** R^2

Even if the predictor is garbage, it may cause a **tiny artificial increase** in R^2. That's why R^2 **alone can be misleading.**

0.129.3 2. Adjusted R^2

This solves the problem by **penalizing** for adding more predictors.

$$\bar{R}^2 = 1 - \frac{\text{RSS}/(n-p)}{\text{TSS}/(n-1)}$$

Where:

- n: number of observations
- p: number of predictors
- RSS: residual sum of squares
- TSS: total sum of squares

If the new variable **helps**, adjusted R^2 goes up If it doesn't help enough, adjusted R^2 goes down

---

0.130 Why is Adjusted R^2 useful?

> It lets you **fairly compare models with different numbers of predictors**.

Especially when:

- You're doing **feature selection**
- Or comparing nested models

0.130.1 Extra notes from the slide:

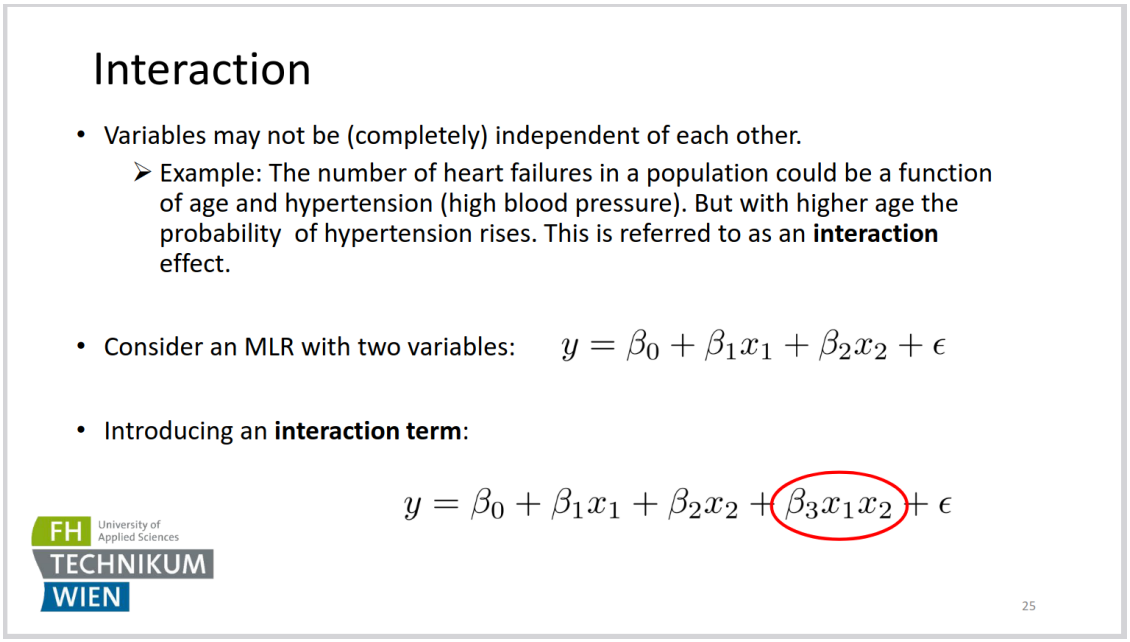
- **RSE** (Residual Standard Error): also a useful measure of model fit; lower is better
- If $p = 1$, then:

$$\bar{R}^2 = R^2$$

Because there's nothing to penalize for

0.130.2 Summary:

Metric	Measures	Penalizes complexity?
R^2	% of variance explained	No
Adjusted R^2	% explained adjusted for predictor count	Yes



0.130.3 Slide Title: *Interaction*

0.131 What's the main message?

Sometimes the **effect of one variable depends on the level of another variable**. That's called an **interaction**.

0.132 Example from the slide:

- You're predicting **number of heart failures**
- Your predictors are:
  - **Age** (x_1)
  - **Hypertension** (x_2)

Alone:

- Older age increases risk
- High blood pressure increases risk

BUT:

> **Older people are more likely to have hypertension** — so their **combined effect** may be **more than just the sum**.

0.133 Mathematically:

0.133.1 Without interaction:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$$

This assumes:

> Each variable has an **independent** and **linear** effect on y.

0.133.2 With interaction:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \boxed{\beta_3 x_1 x_2} + \epsilon$$

That new term $\beta_3 x_1 x_2$ captures how the **effect of one variable depends on the other**.

It says:

> "Maybe the impact of hypertension **is worse at older ages**."

0.134 Why include interaction terms?

Because real-world phenomena often **aren't additive**:

- Two meds might interact
- One skill might only matter if another skill is present
- A feature might be useful **only when combined with** another

---

0.134.1 But beware:

Adding interaction terms:

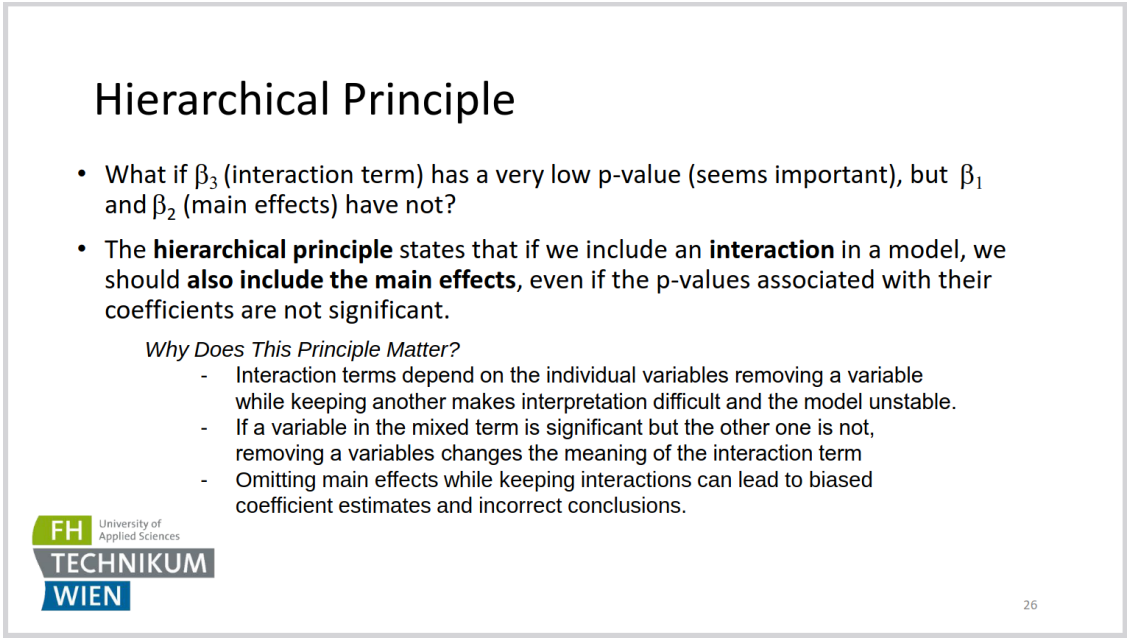
- Makes the model more complex
- Can lead to **multicollinearity**
- Increases the number of predictors rapidly (especially with many variables)

So you usually include interaction terms only when:

- **You have a theoretical reason to**
- OR model performance improves significantly

0.135 Summary

Concept	Without interaction	With interaction
Model formula	$\beta_1 x_1 + \beta_2 x_2$	$\beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$
Assumes	Effects are independent	Effects depend on each other
Useful when	Features combine non-additively	e.g., "Age amplifies hypertension"



0.135.1 Slide Title: *Hierarchical Principle*

0.136 The key question:

> What if your **interaction term** ($\beta_3 x_1 x_2$) is **significant**, but the **main effects** ($\beta_1 x_1$, $\beta_2 x_2$) are **not**?

Should you keep only the interaction and drop the others?

No! That would violate the **hierarchical principle**.

0.137 What is the Hierarchical Principle?

> **If you include an interaction term in your model, you must also include its associated main effects** — even if their p-values are not significant.

0.137.1 Why? (As the slide asks: "Why Does This Principle Matter?")

0.137.2 1. Interaction depends on its components

- If $x_1 x_2$ is in the model, but x_1 or x_2 isn't,
- You **lose context**: how can you interpret the interaction if you don't know how each variable acts on its own?

0.137.3 2. Removing main effects distorts interpretation

Let's say β_3 is big and significant, but you exclude β_1 and β_2. Then:

- The estimated effect of $x_1 x_2$ becomes **confounded** and **biased**
- You can't meaningfully isolate interaction strength

0.137.4 3. It breaks the model's stability

- Most statistical models assume **nested structure** — lower-order terms should be included before higher-order ones.
- If not, you're building a house by skipping the ground floor.

0.138 Example:

If your model is:

$$y = \beta_0 + \beta_3 x_1 x_2 + \epsilon$$

It becomes:

- **Unstable**: small changes in the data shift the meaning of β_3
- **Uninterpretable**: you can't say how much x_1 affects y unless you also include $\beta_1 x_1$

0.139 Correct model (with hierarchy):

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \epsilon$$

Even if β_1 or β_2 has a high p-value → **leave them in** if β_3 is included.

0.139.1 Summary:

Rule	Why it matters
Include main effects with interactions	Ensures interpretability and model stability
Never drop lower-order terms if higher-order term is present	Prevents bias and misinterpretation

---

Take-aways

Statistical metrics
- **p-value**: Tests if an individual variable is useful
- **F-score**: Tests if the overall model is better than guessing
- **R-squared**: Measures how well the model explains the data

***Too little data** leads to unreliable statistical tests*

Higher dimentionality
- **Too many variables**: Risk of false significance and collinearity
- **Correlation**: Can cause instability in coefficients if not handled properly

Feature selection process

FH University of Applied Sciences TECHNIKUM WIEN

0.139.2 Slide Title: *Take-aways*

0.140 Key Statistical Metrics

0.140.1 p-value:

Tells you whether an **individual variable** is useful

- Tests $H_0 : \beta_j = 0$
- Low p-value → keep the variable
- High p-value → variable may not contribute

0.140.2 F-score:

Tests whether the **overall model** is better than just guessing the mean

- Derived from comparing RSS and TSS
- Low F → fail to reject the null → model not better than average
- High F → small p-value → model is globally significant

0.140.3 R-squared:

Tells you **how well the model explains the variability** in the target

- Range: 0 to 1
- Closer to 1 → better fit
- Always increases when you add variables (even useless ones!)

So we prefer to use **adjusted** R^2 when comparing models with different numbers of predictors

---

0.141 Pitfalls

0.141.1 Too little data

Leads to **unreliable statistical tests**.

- p-values become unstable
- F-statistics can become meaningless
- You may think a variable is useful just by chance

The fewer observations you have, the more cautious you need to be

---

0.141.2 Too many variables (High dimensionality)

Brings two major risks:

1. **False significance**
   - With many predictors, you're likely to find some low p-values **just by random chance**

2. **Collinearity**
   - Highly correlated predictors (multicollinearity) make coefficient estimates unstable
   - Signs of β_j may flip unexpectedly
   - Interpretation becomes difficult

0.141.3 Final recommendation:

You need a **feature selection process** to avoid:

- Overfitting
- Instability
- Misleading conclusions

0.141.4 Overall Summary:

Problem	Tool / Metric
Test individual feature	p-value
Test overall model	F-test
Explain variance	R-squared (adjusted if needed)
Too few data points	→ Statistical tests become noisy
Too many variables	→ Feature selection required
Collinearity	→ Handle with correlation analysis or regularization

---