Data & Things

(Spring 26)

Wednesday February 11

Lecture 5: Regression

Jens Ulrik Hansen

Roskilde Universitet

Outline of this lecture

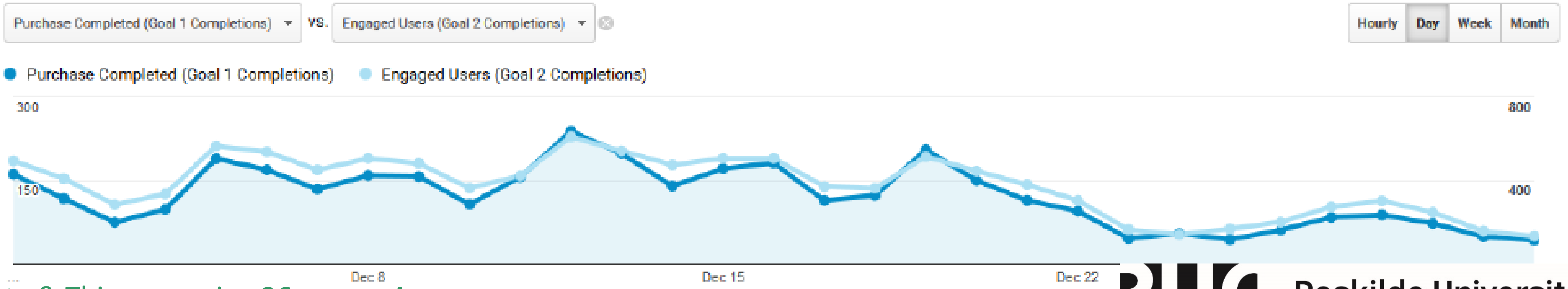
- Correlation and testing for relationship
- Simple linear regression
- Evaluations of regression models
- Multiple linear regression
- Exercises

Correlation and testing for relationship

- **Relationship between a numeric and categorical variable**
  - Visualization: Histogram or boxplot for each value of the categorical variable
    - See the notebook “Correlation and test of relationship.ipynb” or the notebook “Exploratory data analysis.ipynb” from a previous class.
  - Significance test: We can do the tests for comparison of groups we learned last time
    - See the notebook “Correlation and test of relationship.ipynb” or the notebook “Comparison of groups.ipynb” from a previous class.
- **Relationship between two categorical variables**
  - Visualization: Mosaic plot
    - See the notebook “Correlation and test of relationship.ipynb” or the notebook “Exploratory data analysis.ipynb” from a previous class.
  - Significance test: Use the Chi-square test, or if one of the combined groups has less than 5 datapoints, use the Fisher’s Exact test.
    - See the notebook “Correlation and test of relationship.ipynb”.

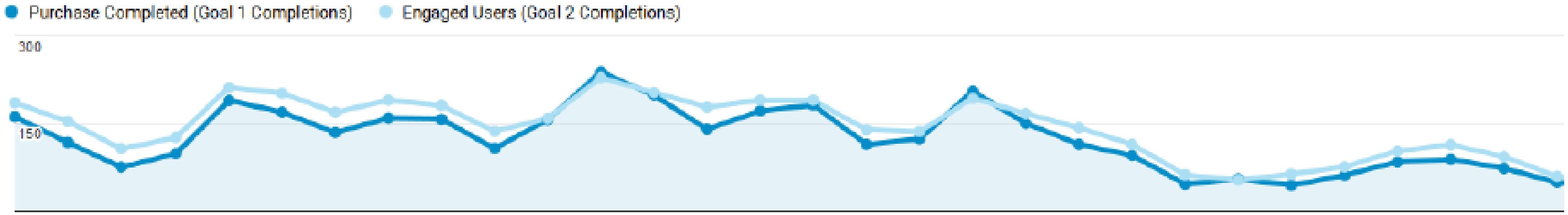
Correlation and testing for relationship

- **Relationship between two numeric variables – correlation**:
  - There is a tendency that the second goes up when the first one goes up, and there is a tendency that the second goes down when the first one goes down (positive correlation)
  - There is a tendency that the second goes down when the first one goes up, and there is a tendency that the second goes up when the first one goes down (negative correlation)
  - Examples of correlation: height and weight, engagement and sales, rain and sun



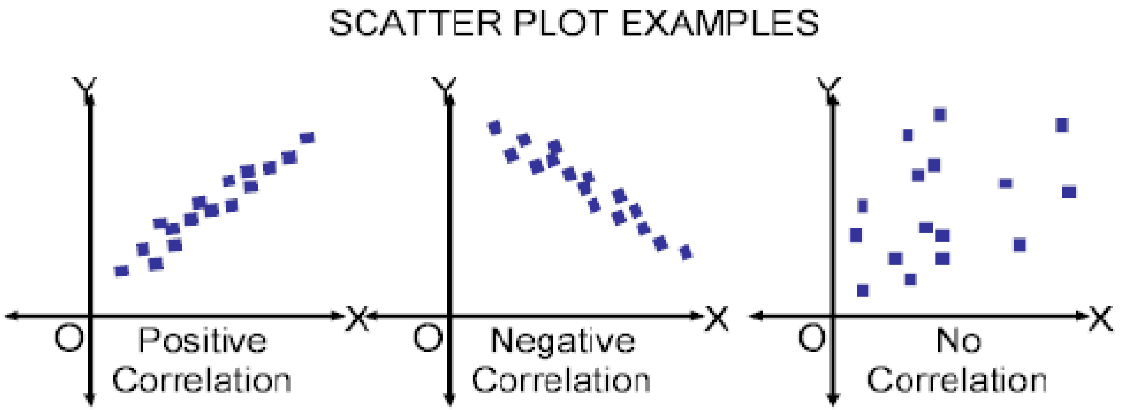
Roskilde Universitet

Correlation and testing for relationship



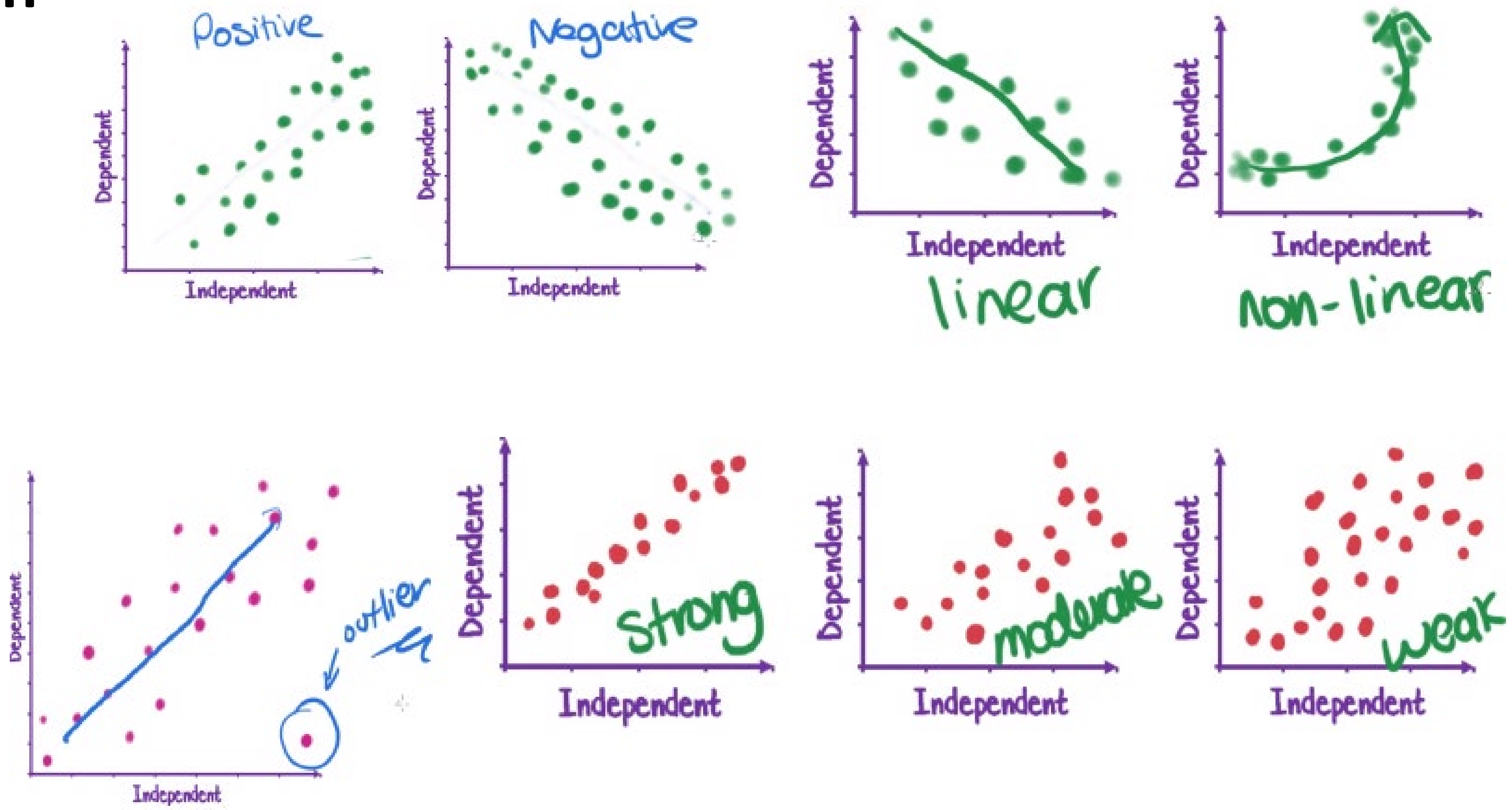
How to detect correlation (among two numerical variables)

- Visualizing correlation
  - ***Scatter plots***
- Quantifying the strength of correlation
  - ***(Pearson's) Correlation coefficient***
    - A number between -1 and 1. 1 is perfect positive correlation, -1 is perfect negative correlation, and 0 is no correlation.
    - <u>Only quantifies linear correlation</u>



Correlation and testing for relationship

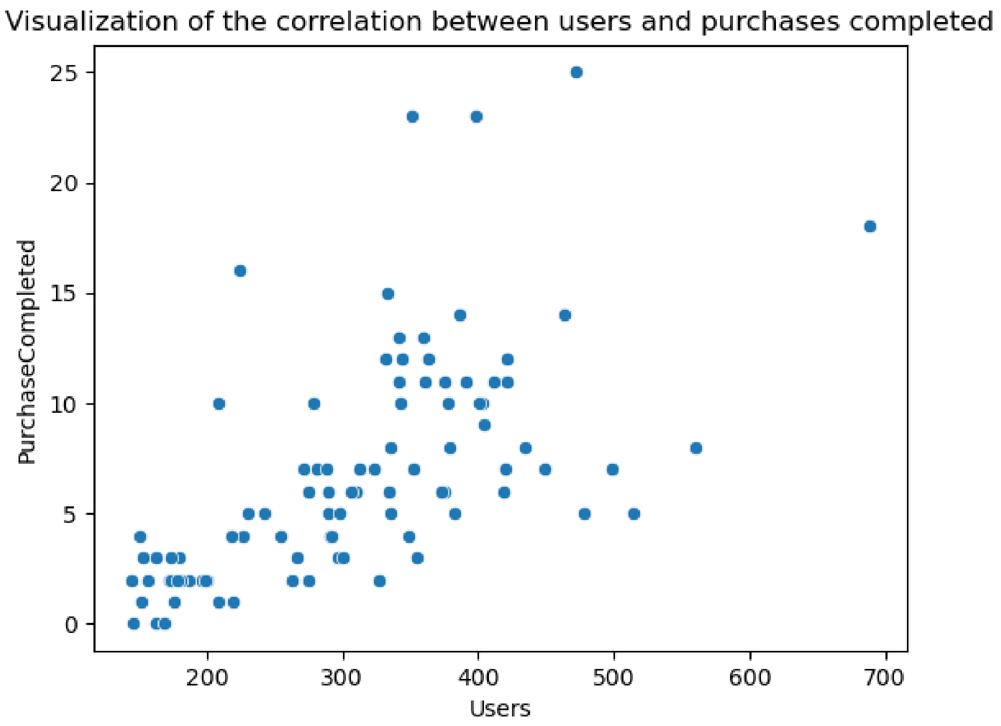
- **Types of Correlation**
  - ***Direction***
    - Positive
    - negative
  - ***Shape***
    - Linear
    - non-linear
  - ***Strength***
    - Weak
    - Moderate
    - strong
  - ***Outliers***



- See: https://www.youtube.com/watch?v=PE_BpXTyKCE

Correlation and testing for relationship

- **In Python**
  - Visualization: ***scatterplot***
    - sns.scatterplot(data = webdata, x = "Users", y = "PurchaseCompleted")
  - Descriptive statistics: ***Pearson correlations coefficient***:
    - Pandas *.corr* method:
      - ebdata["Users"].corr(webdata["PurchaseCompleted"])
    - SciPy's function *pearsonr*
      - stats.pearsonr(webdata["Users"], webdata["PurchaseCompleted"])

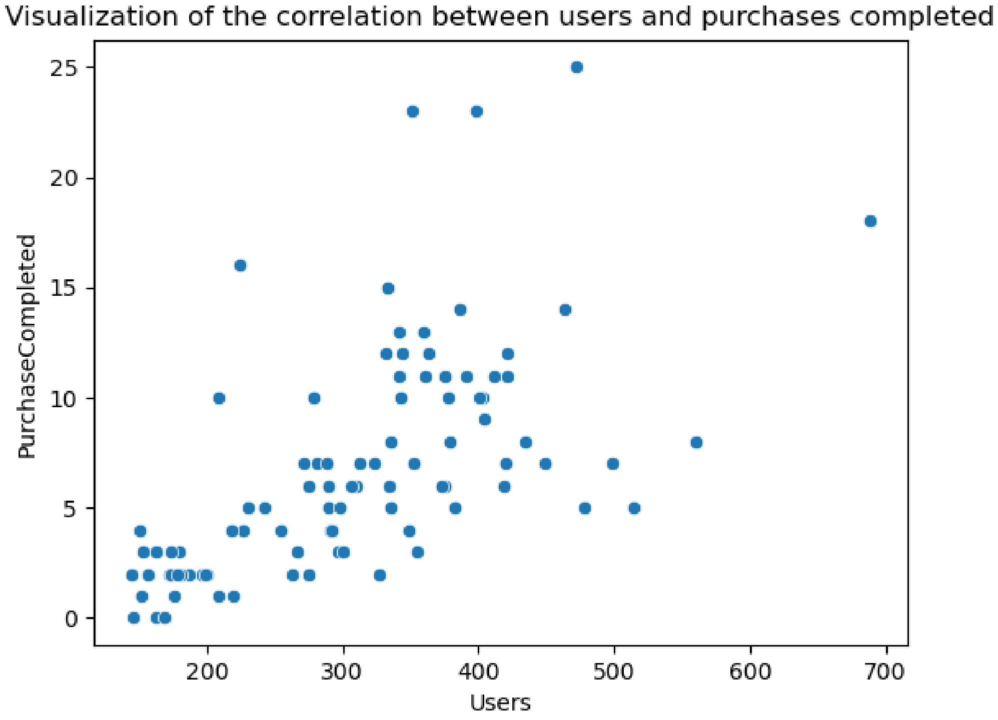



```
stats.pearsonr(webdata["Users"], webdata["PurchaseCompleted"])
```

```
PearsonRResult(statistic=0.6152012891837795, pvalue=6.80560196187495e-11)
```

Correlation and testing for relationship

- **Statistical testing for correlation (relationship) of two numeric variables**
  - The Pearson correlation coefficient tell us the strength of the linear relationship
  - To make sure the relationship is statistically significant (the correlation coefficient is truly different from 0) The *pearsonr* function from SciPy also give us a p-value




```
stats.pearsonr(webdata["Users"], webdata["PurchaseCompleted"])
```

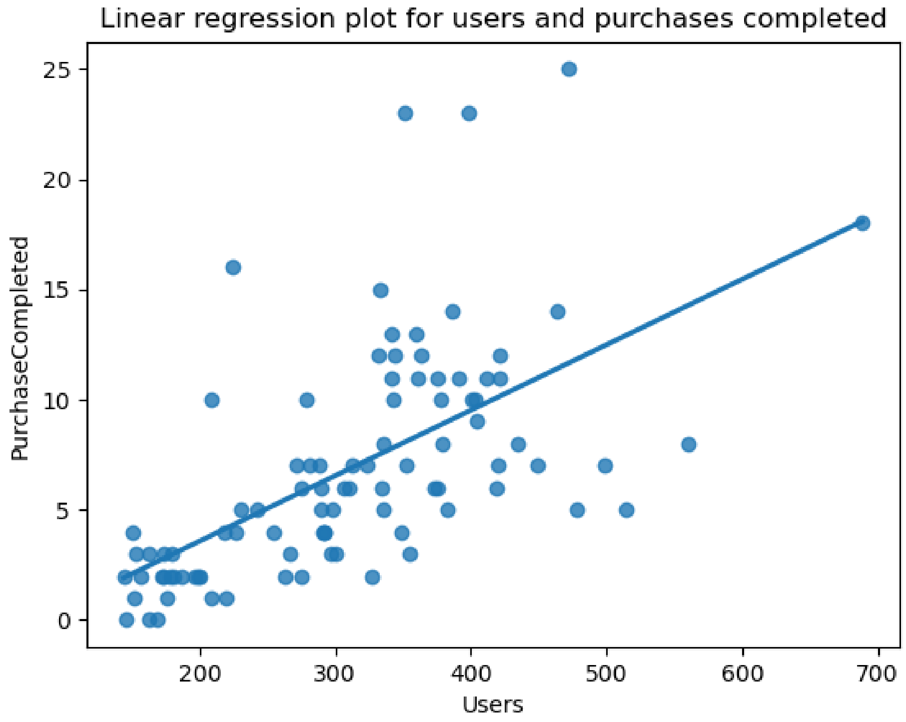
```
PearsonRResult(statistic=0.6152012891837795, pvalue=6.80560196187495e-11)
```

Outline of this lecture

- Correlation and testing for relationship
- Simple linear regression
- Evaluations of regression models
- Multiple linear regression
- Exercises

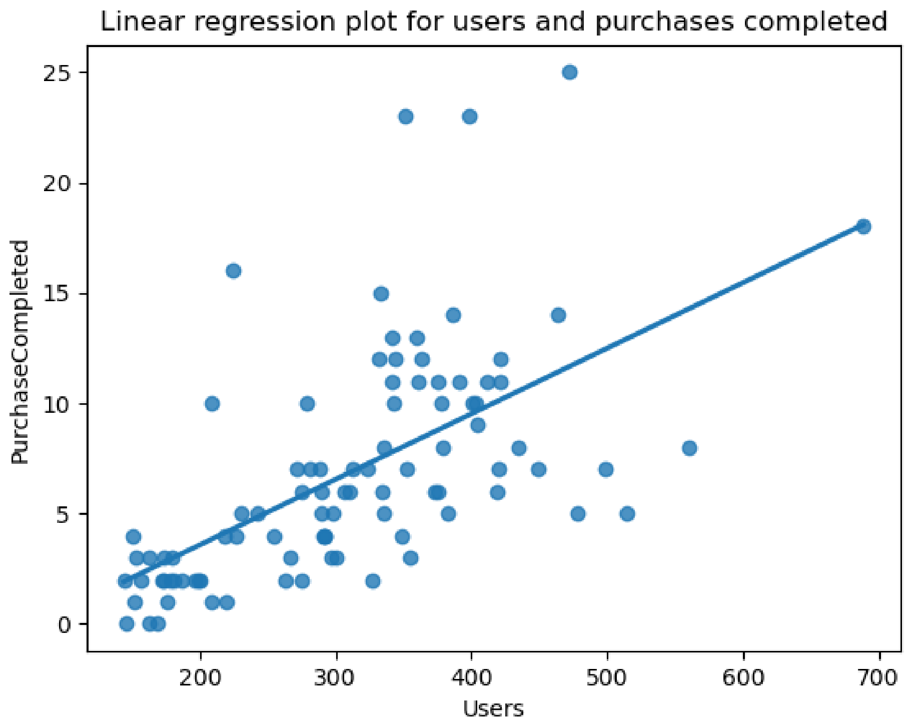
Simple linear regression

- **Correlation and linear regression**
  - We have seen how to measure the strength of a correlation and to test if it is statistically significant…
    - We have not yet seen how to quantify the relationship – if number of Users chances with a certain amount, how much exactly do the PurchaseComplete change?
  - Reformulated: Can we find the best linear line that fits the points?
    - Yes, that is what linear regression is all about
  - Can we use the line to predict y values from x values?
    - Yes, linear regression is the simplest form for predictive model for regression problems



Simple linear regression

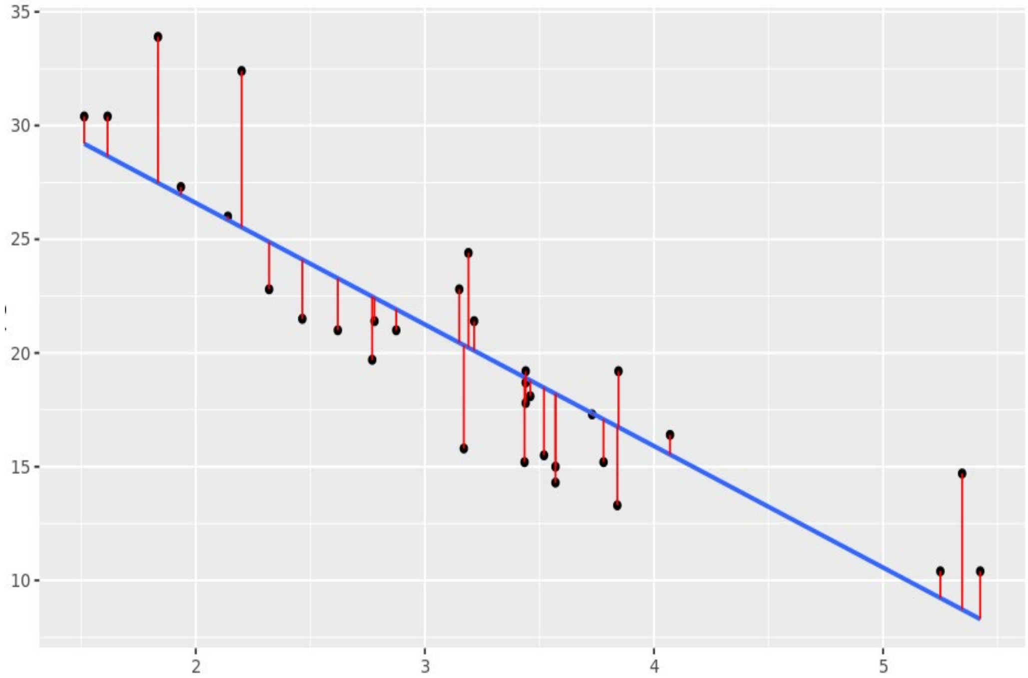
- **The simple linear regression model**
  - In a scatterplot, we characterize a linear line by the formula: $y = a + b*x$
  - How do we find parameters a and b that make the line fit the data points *"best"*?
  - One way is to minimize the sum of squared errors, also referred to as Ordinary Least Squares (OLS)…



Roskilde Universitet

Simple linear regression

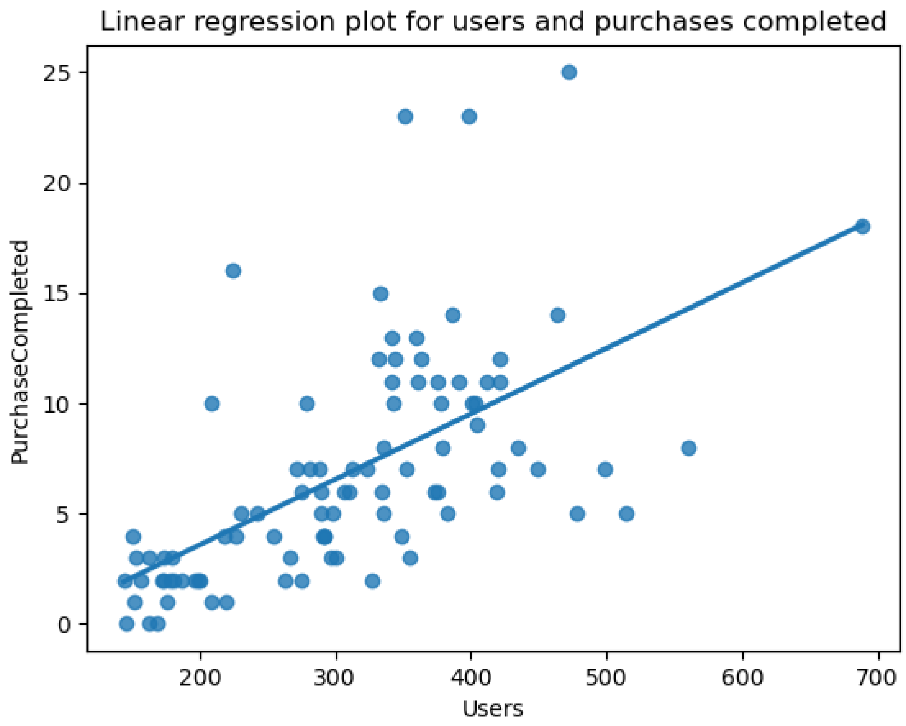
- **Ordinary Least Square (OLS)**
  - The simple linear regression formula:
    - *$y = a + b*x$*
  - Our dataset consists of pairs (x_i, y_i) and let $\hat{y}_i$ denote the predicted value for x_i, that is:
    - $\hat{y}_i = a + b*x_i$
  - The error in predicting y_i from x_i is thus:
    - $\hat{y}_i - y_i$
  - These are also referred to the ***residuals*** of the model (-the red line in the plot)
  - The ***sum of squared errors*** is thus:
    - Sum(($\hat{y}_i$ - y_i)^2)
  - *OLS* find the *a* and *b* that minimize the sum of squared errors (minimizes the sum of the square lengths of the red lines in the plot)
    - There are closed formulas for a and b, but one could also use approximation methods like gradient decent, generally used in machine learning



Roskilde Universitet

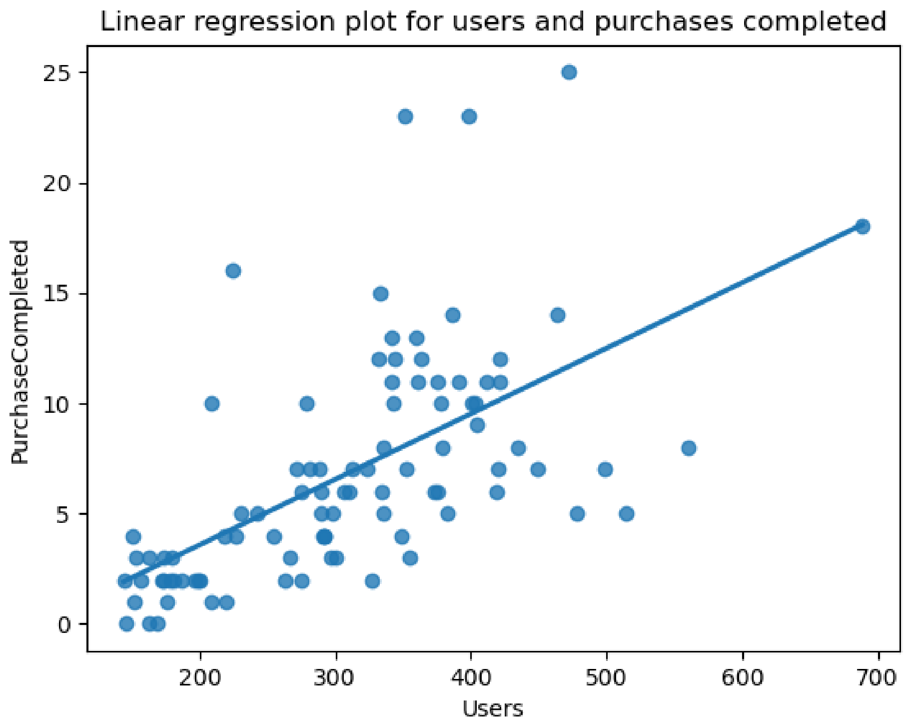
Simple linear regression

- **Interpretation of Simple Linear Regression models**
  - Given the linear line: *y = a + b*x*
  - *a* and *b* are also called the ***coefficients***
  - *a* is called the ***intercept*** and is where the line intersects the y-axis – it corresponds to the prediction of y if x is 0
  - *b* is called the ***slope*** and tell us how much the line increase in the direction of y given one unit of increase in x.
  - Thus, linear regression models are easy to interpret and very useful for making inference about the population from the sample (inferential statistics)



Simple linear regression

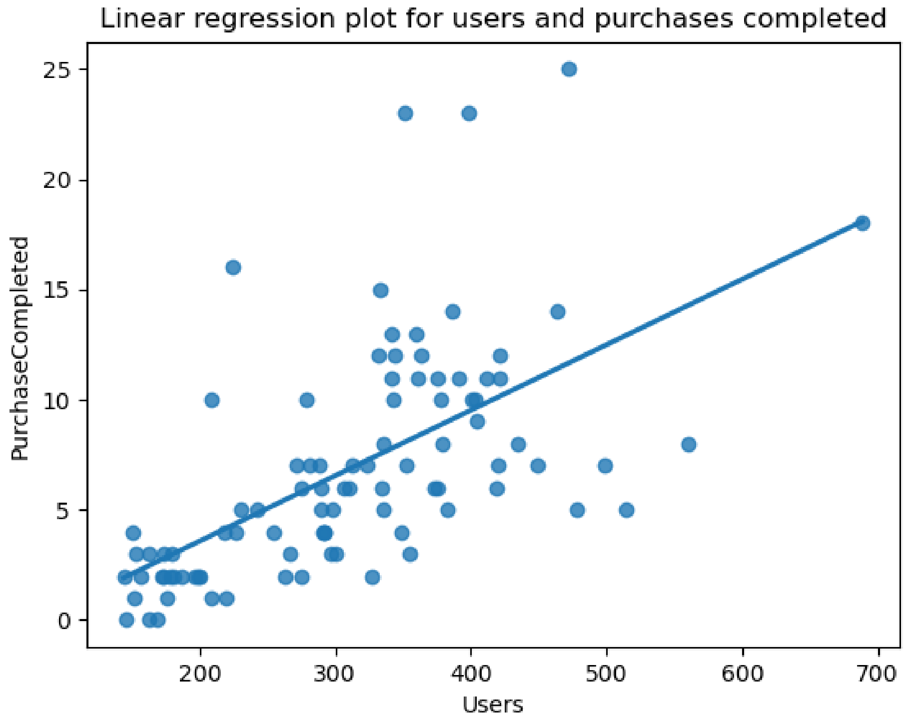
- **Difference between three important measures**
  - Given two numeric variables *y* and *x*.
  - *The* ***correlation coefficients*** between x and y tells us how strong a linear relationship (positive or negative) there is between x and y, that is how close to a straight line the points fall.
  - The associated ***p-value*** (returned by *personr* for instance) tell u whether this association is truly different from zero, that his whether the straight line is truly different from a horizontal line.
  - Finally, the ***slope*** or the ***coefficient of x*** (in *y = a + b*x)* tell us to what extent y changes as x changes, that is how steep the straight line is.
  - Note that we can have a very high correlation and a very small p-value, indicating a strong significant linear relations ship, and at the same time a very small coefficient of x indicating that y changes very little when x changes.



Roskilde Universitet

Simple linear regression

- **Assumptions and problems of simple linear regression**
  - We return to this when talking about multiple linear regression.



Simple linear regression

- Let us look at examples in Python in the notebook “Simple linear regression.ipynb”

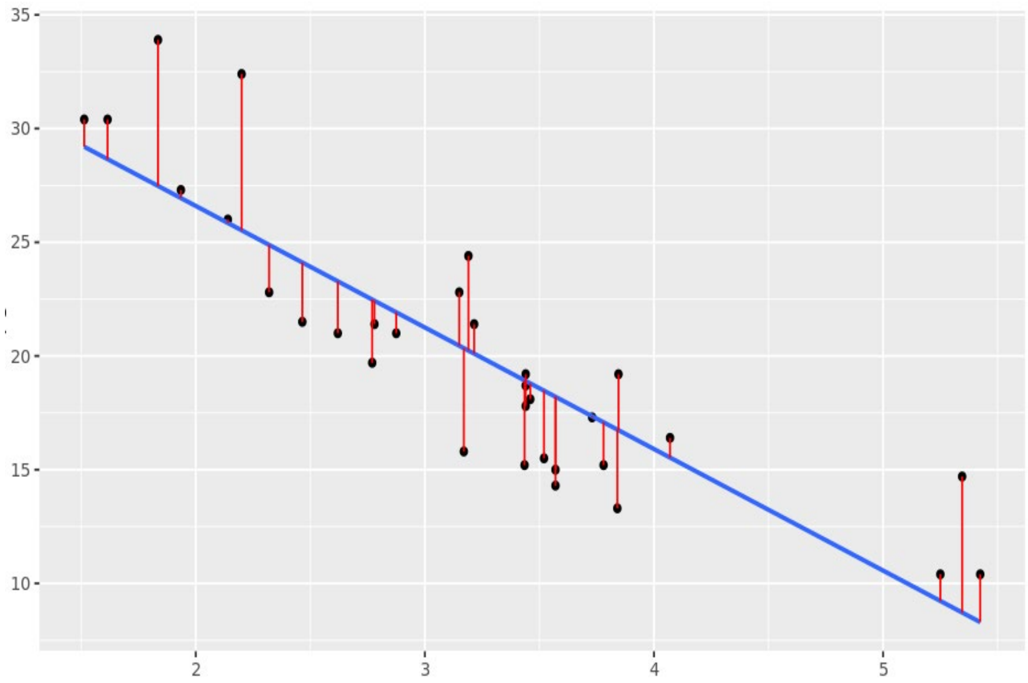
Roskilde Universitet

Outline of this lecture

- Correlation and testing for relationship
- Simple linear regression
- Evaluations of regression models
- Multiple linear regression
- Exercises

Evaluations of regression models

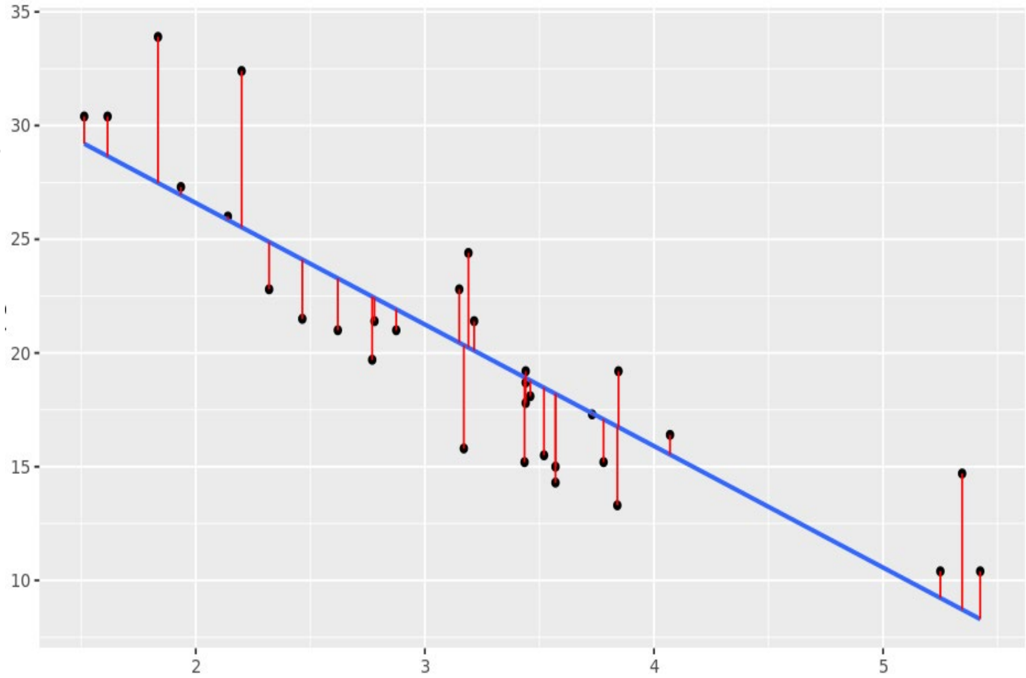
- **Evaluation of regression models**
  - How do we evaluate how good our line fit the points?
- **Error measures** (*smaller values are better)*
  - **MSE:** Mean Squared Error
    - MSE = mean(($\hat{y}_i$ - y_i)^2)
  - **RMSE:** Root Mean Squared Error
    - RMSE = sqrt(mean(($\hat{y}_i$ - y_i)^2))
    - On scale of the variable y.
    - "The accuracy of regression models"
  - **MAE:** Mean Absolute Error
    - MAE = mean(abs($\hat{y}_i$ - y_i))
    - "On everage how big are our errors when predicting y"



Roskilde Universitet

Evaluations of regression models

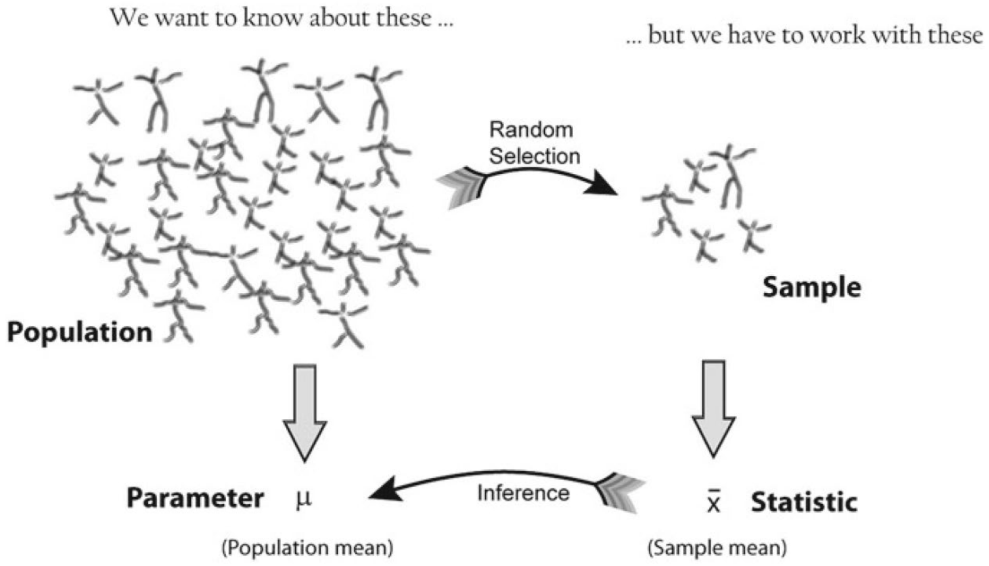
- **Evaluation of regression models**
  - ***R-squared (R^2) – coefficient of determination***
    - sum of squared errors / total sum of squares
    - sum(($\hat{y}_i$ - y_i)^2) / sum((y_i – mean(y_i))^2)
    - Always a value between 0 and 1
    - The fraction of variation in y explained by the variation in x
    - The higher value the better
    - For simple linear regression R^2 is indeed the Pearson correlation coefficient squared.
    - Different applications set different standards for what a good R^2 is. (Modeling a physical phenomena, we might want R^2 to be above 0.9, while an R^2 of 0.4 is really good if we are modeling human behavior.)



Roskilde Universitet

Evaluations of regression models

- **Simple linear regression in statistics and machine learning**
  - In ***statistics***, we want to ***infer*** knowledge about a population from a data sample of the population
    - Evaluating how good a model is in terms of inference can be done by *R-squared*
  - In ***machine learning***, we want to make ***predictions*** on a population from a model trained on a data sample (from the population)
    - Evaluating how good a model is in terms of prediction can be done by *RMSE* or *MAE*

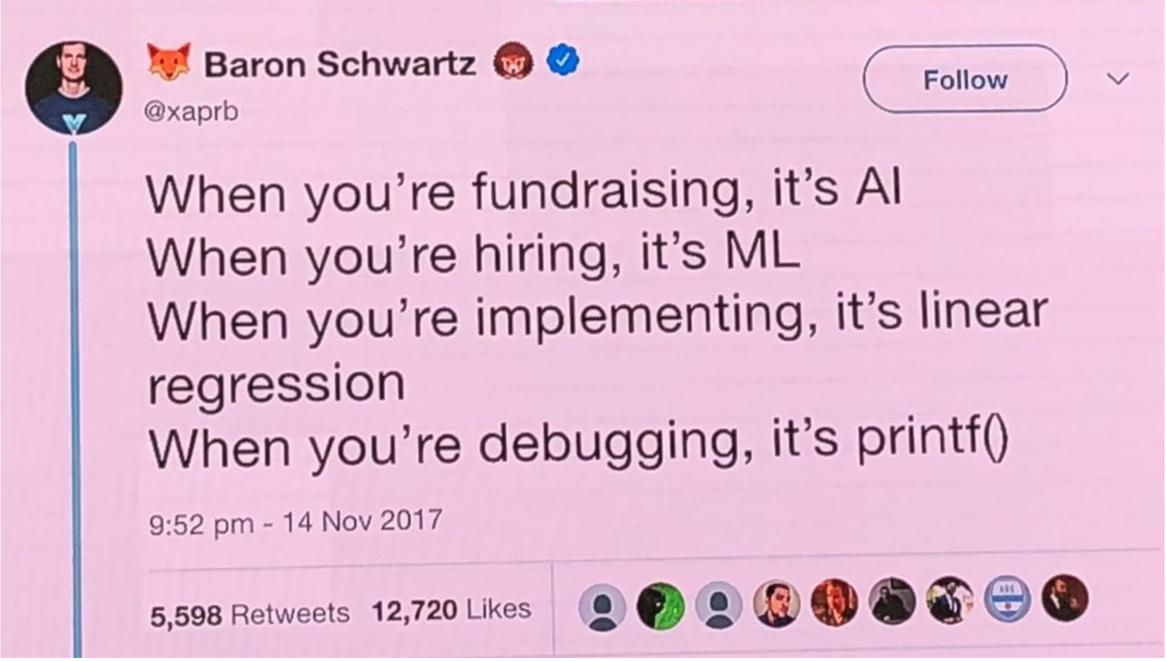


Haslwanter, T. (2022). *An Introduction to Statistics with Python - With Applications in the Life Sciences*. Springer, Cham.

  - Simple linear regression, showcase that the two tasks may overlap and can sometimes be done by the ***same underlying models***

Evaluations of regression models

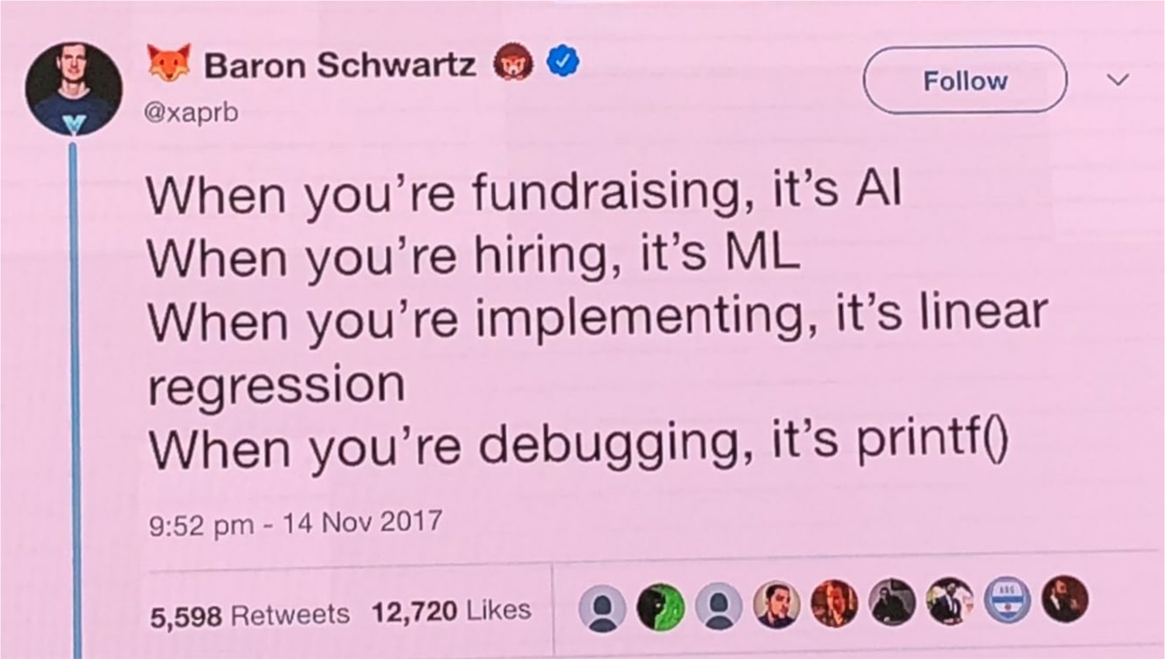
- **Linear regression in machine learning**
  - Linear regression is the easiest regression model to use and understand
  - Linear regression (and its extensions) is good enough for many real business problems
  - Linear regression models and predictions based on them can be explained and easily communicated
  - Linear regression provides a baseline to which more advanced and sophisticated regression models can be compared



Roskilde Universitet

Evaluations of regression models

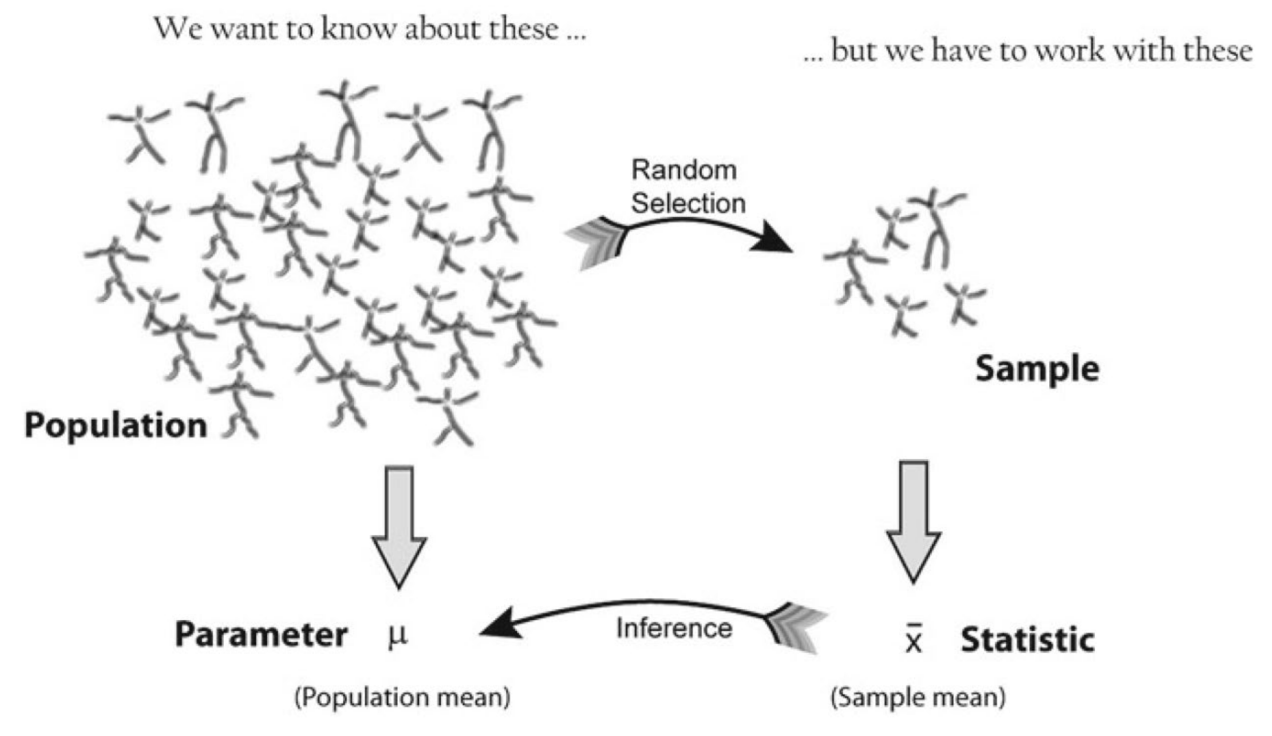
- **Machine learning beyond linear regression**
  - Linear regression models are extremely robust in the sense that training on different sub-samples of the entire dataset gives quite similar models – this is not the case for most other machine learning models.
  - For linear regression, if we satisfy the assumptions to be mentioned later, statistical theory can give use estimation of “goodness of fit” in terms of p-values for the coefficients and F-statistics – this is not the case for most other machine learning models.
  - Thus, we need another approach to evaluating machine learning models…



Roskilde Universitet

Evaluations of regression models

- In ***statistics***, we want to ***generalize*** from sample statistics to population statistics (from sample mean to population mean)
- In ***machine learning***, we want to build predictive models that make accurate predictions on the population – i.e. we want machine learning models that ***generalize*** well to the population
- In ***statistics***, we get an estimate about how good our generalization is by making ***assumptions about the data generation process*** (such as data comes from a normal distribution) that allow us to calculate things like p-values
- In ***machine learning***, we often make ***no assumption about the data generation process*** (we treat our predictive models as black boxes), so we need another way of measuring how well we generalize…

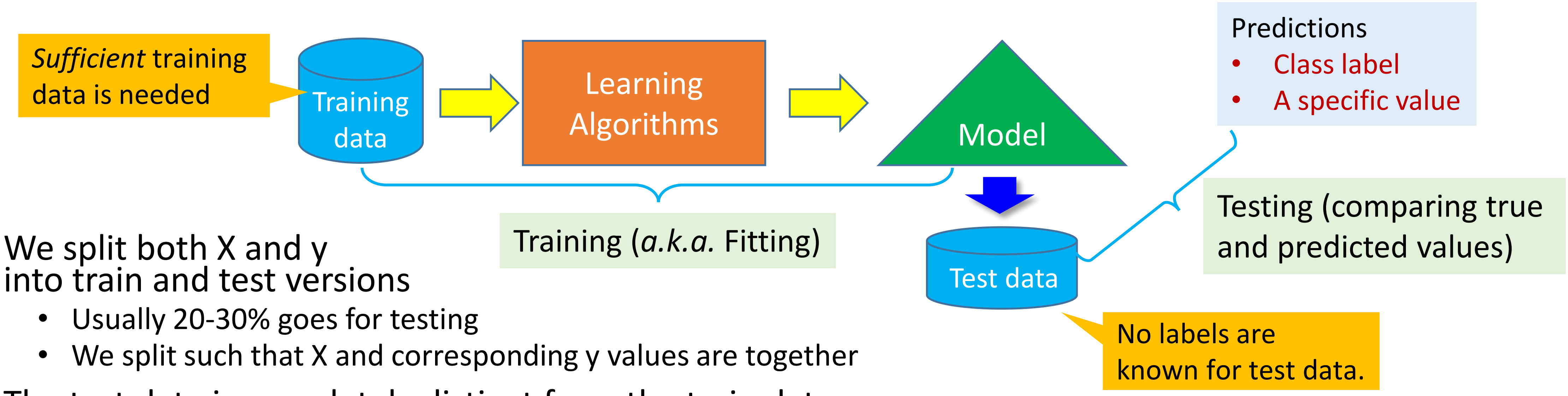


Haslwanter, T. (2022). *An Introduction to Statistics with Python - With Applications in the Life Sciences*. Springer, Cham.

Roskilde Universitet

Evaluations of regression models

- To be sure we generalize well in machine learning, we make a distinction between the data we ***train*** a model on and data we use to ***test/evaluate*** our model.



- We split both X and y into train and test versions
  - Usually 20-30% goes for testing
  - We split such that X and corresponding y values are together
- The test data is completely distinct from the train data
- Once the model is trained on the training data, we feed in the X part of the test data to make predictions $\hat{Y}$ on the test data.
  - We can then calculate our final evaluation of the model based on metrics (such as RMSE) that compare $\hat{Y}$ to the true Y from the test data.

Evaluations of regression models

- Overfitting: A model works well on the training data but generalizes poorly to unseen data – much better evaluation scores on the training set than on the test set. Can be due to:
  - Noise in the training data is learned as pattern.
  - Too many features are used in training.
  - The model type used is too complex.
  - Lack of proper variance in the training data
- Underfitting: A model even does not work well on the training data – poor evaluation scores on the training set. Can be due to:
  - Too few features are used in training.
  - The model type used is too simple.
  - The training dataset is too little, failing to contain sufficient variance.

Evaluations of regression models

Example: Leaf Detection/Classification

- Training data (leaf samples)

- Test data (unseen)

Include more training samples.
E.g., those without sawtooth.

An *overfitting* model might say ”Not a leaf”.
- The training data samples all have sawtooth
- The model thinks a leaf must have sawtooth.

Use more features or
a more complex model.

An *underfitting* model might say ”A leaf”.
- Only color is used as the feature.
- The model thinks everything green is a leaf.

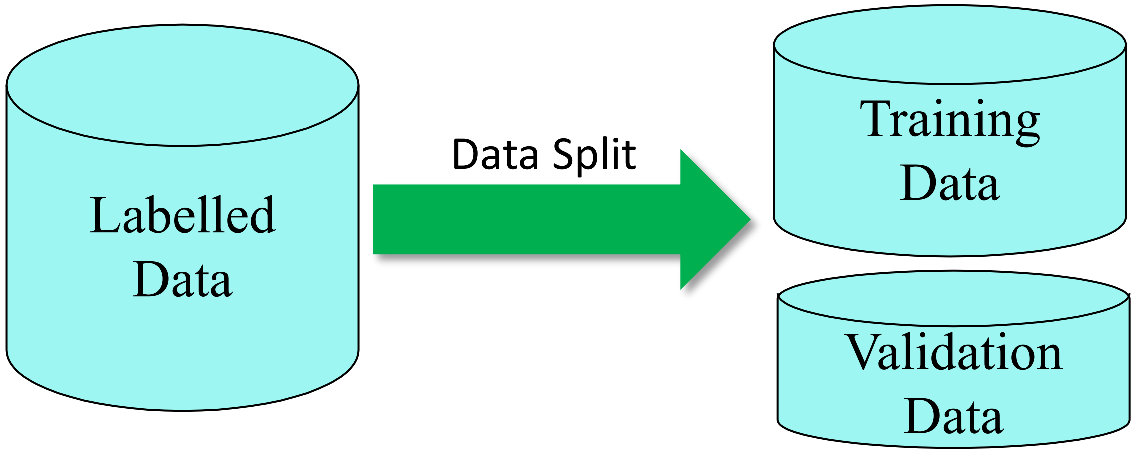
Include more training samples.
E.g., those in other colors.

An *overfitting* model might say ”Not a leaf”.
- The training data samples are all green.
- The model thinks a leaf must be green.

Leaf images from https://www.dreamstime.com/

Evaluations of regression models

- To see how to evaluate regression models and make train-test split in Python, let us look at the notebook "Evaluation of regression models.ipynb".



Roskilde Universitet

Outline of this lecture

- Correlation and testing for relationship
- Simple linear regression
- Evaluations of regression models
- Multiple linear regression
- Exercises

Multiple linear regression

- The multiple regression formula now looks like:
  - $y = a + b^1 * x^1 + b^2 * x^2 + \ldots + b^k * x^k$
  - *For some number of k features/predictors/independent variables*
- *Example*
  - *HousePrise = a + b¹*"size" + b²*"noRooms" + b³*"distToSchools"*
- Our dataset now consists of tuples/rows of the form $(x^1_i, x^2_i, \ldots x^k_i, y_i)$
- We still use $\hat{y}_i$ to denote the predicted value for the i'th datapoint/row, that is
  - $\hat{y}_i = a + b^1 * x^1_i + b^2 * x^2_i + \ldots + b^k * x^k_i$
- Residuals or errors are also still defined as
  - $\hat{y}_i - y_i$
- Sum of squared errors are defined in the same manner, and we can use OLS for fitting a multiple regression model, just as for simple linear regression

Roskilde Universitet

Multiple linear regression

- **Evaluation of multiple regression models**
  - **We use the same metrics as for simple linear regression:**
  - **MAE:** Mean Absolute Error
    - MEA = mean(abs($\hat{y}_i$ - y_i))
  - **MSE:** Mean Squared Error
    - MSE = mean(($\hat{y}_i$ - y_i)^2)
  - **RMSE:** Root Mean Squared Error
    - RMSE = sqrt(mean(($\hat{y}_i$ - y_i)^2))
  - ***R-squared (R²)*** / coefficient of determination
    - R^2 = sum(($\hat{y}_i$ - y_i)^2) / sum((y_i – mean(y_i))^2)
  - ***Adjusted R-squared (Adj. R²)***
    - *Adj.* R^2 = *1 – (1-* R^2*)**(n-1) / (n-p-1), where *n* is the number of rows and *p* is the number of columns in X.
    - Useful when adding new predictor variables. When addding new predictor variables R^2 will always increase, but the increase might be so small that it could be really no effect. Adj R^2 adjust for this.

Multiple linear regression

- **Dealing with categorical variables**
  - All categorical predictor/feature/independent variables need to be transformed into “dummy variables” as we learned to do when discussing Data Transformation.
  - Recall that one of the dummy variables will be dropped and acts as the reference variable. This is important when interpreting the regression coefficients for the dummy variables
  - See the notebook “Multiple linear regression.ipynb”

Multiple linear regression

- **Extending with interaction and polynomial features**
  - We could imagine adding an interacting term between two variables like x^3 and x^5. This corresponds to adding the term *$c*x^3*x^5$ to the multiple regression formula:*
    - $y = a + b^1*x^1 + b^2*x^2 + \ldots + b^k*x^k + c*x^3*x^5$
  - This can be done by adding a new column that is x^3*x^5. Following this, one can just conduct usual multiple linear regression and the coefficient for this new column will be *c* in the equation above.
  - We could imagine adding a polynomial transformation of one of the variables, such as (x^3)^2. This corresponds to adding the term *$d*(x^3)$^2 to the multiple regression formula:*
    - $y = a + b^1*x^1 + b^2*x^2 + \ldots + b^k*x^k + d*(x^3)^2$
  - This can be done by adding a new column that is (x^3)^2. Following this, one can just conduct usual multiple linear regression and the coefficient for this new column will be *d* in the equation above.

Roskilde Universitet

Multiple linear regression

- **Assumptions and problems for linear regression**
  - For inference, these might affect the validity of our inferences, such as the actual effect, significance of coefficients (p-values), and confidence intervals
  - For predictions, these might result in low predictive performance or biased errors

Multiple linear regression

- **Assumptions and problems for linear regression**
  - Linearity assumption: y varies linearly with x
    - Plot residuals vs predicted value $\hat{y}$
  - Correlation of error terms assumption: There are no correlation amount the residuals (e_i does not tell us anything about e_{i+1})
    - Plot residuals vs x (time)
  - Constant variance of error terms assumption: The variance of the residuals is constant (does not correlated with the predicted value $\hat{y}$)
    - Plot residuals vs predicted value $\hat{y}$
  - Outliers assumption: There are no outliers
    - Plot residuals vs predicted value $\hat{y}$
  - High-leverage points assumption: There are no leverage points
    - Plot leverage statistics
  - Collinearity assumption: None of the predictor variables are very strongly correlated
    - Look at correlation matrix of predictors

Multiple linear regression

- **Other regression models**
  - There are plenty of other models for regression that do not assume y is linear in the x variables
    - We will talk a bit about how tree-based models for regression, when we talk about tree-based models for classification.
    - We will also briefly see how neural networks can be used for regression, when we get to those
- **In Python**
  - Very similar to simple linear regression
  - We can both use statsmodels and scikit-learn
  - See the notebook "Multiple linear regression.ipynb"

Outline of this lecture

- Correlation and testing for relationship
- Simple linear regression
- Evaluations of regression models
- Multiple linear regression
- Exercises

Exercises

- Do the exercises in the notebook "Exercises in linear regression.ipynb"