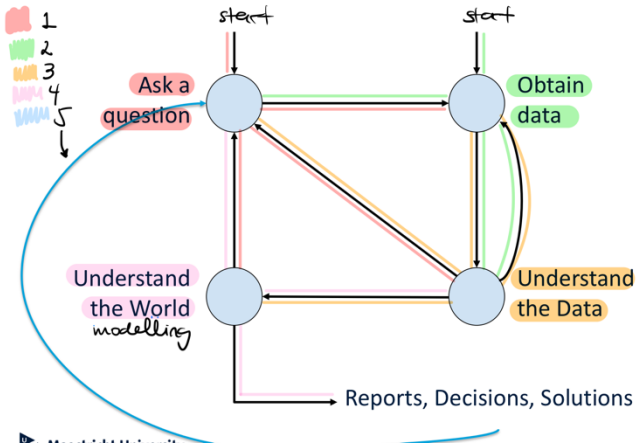
Table of Contents

Lecture 1.1: Exploratory Data analysis (EDA)

1. Introduction

→ Data Science lifecycle



1. Question/problem formulation
   - Bad: can I build… / What is the precision of…
   - *What do we want to know about the model?*
   - *What is the scientific goal?*
   - *What problem(s) are we trying to solve?*
   - *What are the hypotheses we want to test?*
   - *What are our metrics for success/evaluation?*
2. Data acquisition and cleaning
   - *What data do we have and what data do we need?*
   - *How were the data sampled? How will we sample more data?*
   - *Is our data representative of the population we want to study?*
   - *Are there privacy (or other) issues?*
3. Exploratory Data Analysis (EDA)
   - *How is our data organized and what does it contain?*
   - *Do we already have relevant data?*
   - *What are the biases, anomalies, or other issues with the data?*
   - *Are there patterns that I can detect in the data?*
   - *How do we transform the data to enable effective analysis?*
4. Prediction and inference
   - *What does the data say about the world?*
   - *Can I build, fit and validate a model about the world using this data?*
   - *How robust are our conclusions and can we trust the predictions?*
   - *What does accuracy of model mean? + what does missing accuracy mean?*
5. Communicate and visualize
   - *What did we learn?*
   - *Does it answer our questions or accurately solve the problem?*
   - *Do the results make sense?*
   - *Can we tell a story?*

2. Exploratory Data Analysis (EDA)

→ Key properties in EDA

1. Structure: shape of the data
   - Rectangular data: rows = samples, columns = features. Easy to manipulate.
   - Tables: different types, need a language.
   - Matrices: same type, need algebra.

2. Granularity: how fine is each data point
   - What does each record/sample represent?
   - If the sample is a group, how was it aggregated?

3. Scope: is the data incomplete? How much?
   - *Does my data cover my area of interest? Check for missing data.*

- *Is my data too expansive?*
- *Does my data cover the right time frame?*

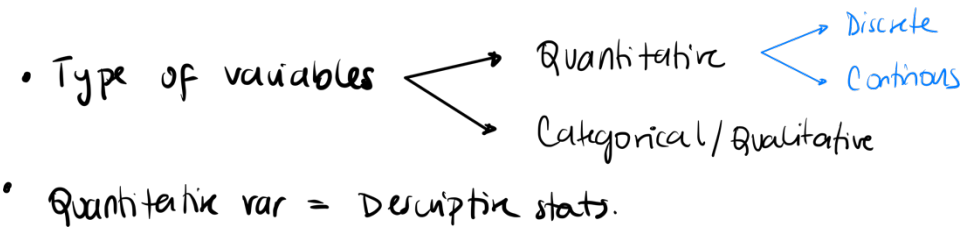
4. Temporality: how is the data situated in time?
   - When was the data collected?
   - What is the meaning of time? (When event happened or when data was collected)
   - Be careful with time zones → use *datetime* library in py.

5. Faithfulness: how well data captures reality?
   - Check if the data contains unrealistic or incorrect values
   - Check if the data violates dependences (ex: bday and age don't match)
   - Check for spelling errors
   - Check for falsification (repeated names, fake emails)
   - Missing/default values:
     - They look like: " ", 0 , 999, 12345, NaN, null, 1970, 1900, …
     - Check first the context (man wouldn't have a 'being pregnant feature')
     - How to handle missing values?
       - Ignore the records (bad when percentage is high)
       - Fill in missing values manually (tedious, infeasible)
       - Fill in missing values automatically:
         - A global constant (ex: unknown or new class)
         - The mean of all samples
         - The mean for all samples belonging to the same group
         - The most probable value (predict it by building a model)

→ Summary

1. Examine the data (date, size, organization, structure)
2. Examine each feature.
3. Examine correlations.
4. While doing all of this, also:

- Visualize/summarize the data
- Validate the assumptions
- Identify anomalies
- Apply data transformations and corrections
- Record what you do so you remember decisions



Lecture 1.2: Visualization

1. Effective Visualizations

→ What do visualizations help us with?

- Identify hidden patterns and trends
- Formulate/test hypotheses
- Communicate any modeling results
- Determine the next step in analysis/modelling

→ Properties

1. Graphical integrity

- Plot all data
- Be proportional and use appropriate scales
- Show data variation
- Detailed labeling
- **Area principle**: violated when in pie chart the areas are not proportional.

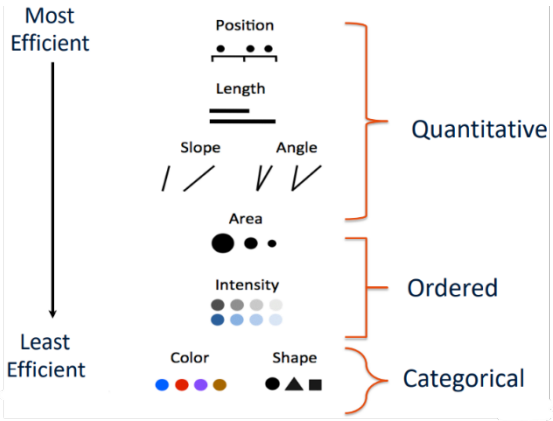
2. Keep it simple

- Make simple and clear plots
- **Lie factor**: size of the graph should be directly proportional to the numerical quantities
- **Jittering** is the act of adding random noise to data in order to prevent overplotting in statistical graphs. This only helps if you don't have too many samples.
- Maximize the **data-ink ratio**: encourages chart creators to examine if all elements (colors, legends, labels, images, annotations, effects) in the chart are relevant to the chart's message. If they are not, you should remove it.

$$\frac{data\text{–}ink}{total\text{–}ink} = \frac{Elements\ conveying\ data\ information}{All\ elements\ in\ the\ chart}$$

3. Use sensible design

- Use color strategically and only if it adds information
  - Quantitative variables: don't use color
  - Ordinal data: try varying luminance and saturation
  - Categorical data: don't use more than 5-8 colors.
- Avoid rainbow colors
- Consider color-blind people



4. Use right display

Depending on what you want to show, there are different plots to use

- Distribution: how a variable or variables in the dataset distribute over a range of possible values.
- Relationship: how the values of multiple variables in the dataset relate.
- Composition: how the dataset breaks down into subgroups
- Comparison: how trends in multiple variable or datasets compare

2. What type of visualization to use depending on type of variable?

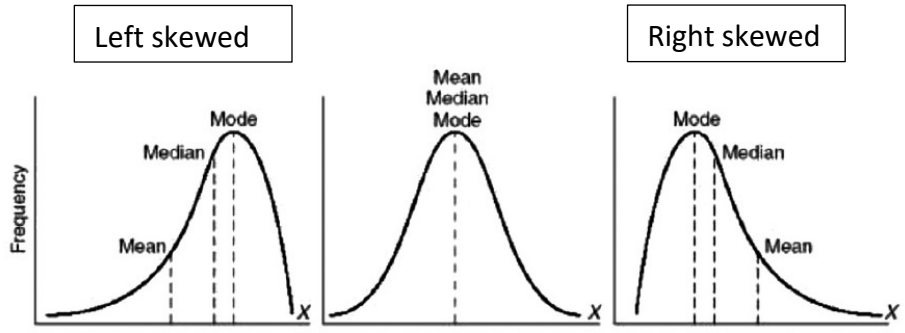
→ Quantitative variables

- Descriptive Statistics
  - min, max, median, and mean
  - Marginal distribution:
  - Five-number summary: min, max, median, Q1 and Q3.

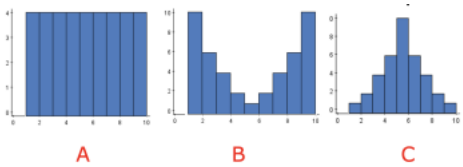
	Baseball	Basketball	Football	Total
Male	13	15	20	48
Female	23	16	13	52
Total	36	31	33	100

Marginal distribution of gender

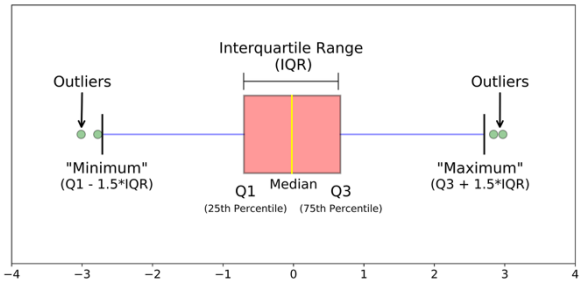
Marginal distribution of sports



- Histograms (NOT categorical variables)
  - Different bins tell different stories (one bin per value or bin as ranges of values)
  - Distribution
    - Skewed right: a lot of values on the left, median < mean.
    - Skewed left: a lot of values on the right, median > mean.
  - **Mode**: value that appears the most
    - A hump or a high-frequency bin (unimodal, bimodal, multimodal)
    - mean – mode = 3 * (mean – median)
  - Quantiles: describe what percentage of the observations in a sample have smaller value
    - 25th percentile = 25% of the data are under that value
  - Standard Deviation (std)
    - $S_x = \frac{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}}{n-1}$ n= data pints
    - quantify the amount of variation or dispersion of a set of data values.
    - If all values increase/decrease by same, std doesn't change.
    - C < A < B

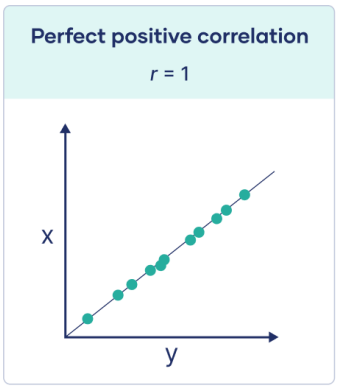


- Outlier detection
  - Histograms
  - Boxplots: Outside IQR = Q3 – Q1
    - Q1: the value under which 25% of data points are found when they are arranged in increasing order.
    - Q3: value under which 75% of data points are found when arranged in increasing order.



→ Covariance and Correlation

- Measure of how much two variables change together.
- Assumptions
  - there must be a true underlying linear relationship between the two variables.
  - the variables must be quantitative.
- Properties
  - outliers can strongly affect the correlation
  - interchanging x and y does not change the correlation neither changing x and y units
  - has no units



0

- Correlation formula:

$$r_{xy} = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Sigma(x_i - \bar{x})^2 \Sigma(y_i - \bar{y})^2}}$$

r_{xy} = correlation coefficient between x and y

x_i = the values of x within a sample

y_i = the values of y within a sample

$\bar{x}$ = the average of the values of x within a sample

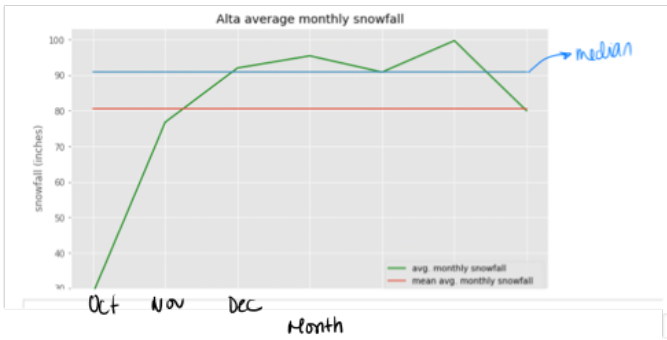
$\bar{y}$ = the average of the values of y within a sample

3. Plots for EDA based on type of variable

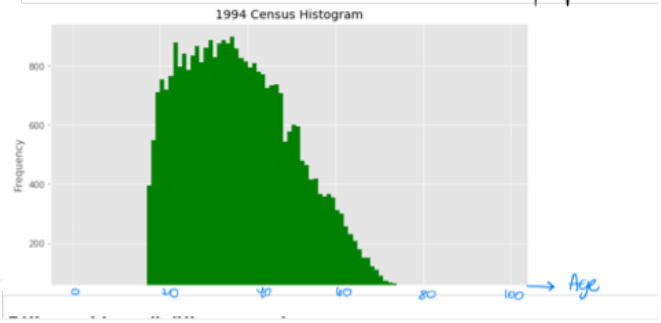
⇒ Quantitative variables

Ideally go for scatterplot, line plot assumes relationship.

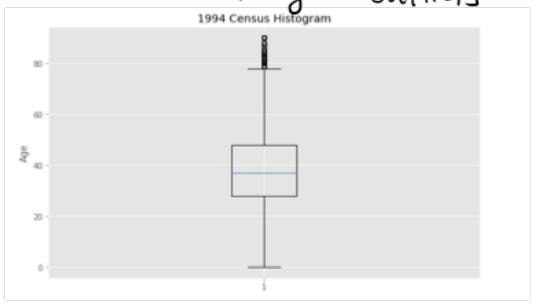
① Line plot



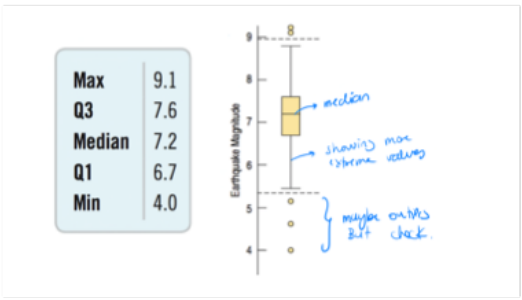
② Histogram : help to see Distribution, Mode : most freq. value



③ Box plot : shows the 5-point summary + outliers



④ →



⇒ Categorical variables

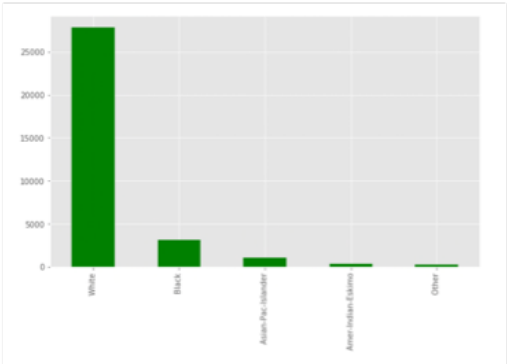
① Freq. tables : count occurances

White	27816
Black	3124
Asian-Pac-Islander	1039
Amer-Indian-Eskimo	311
Other	271

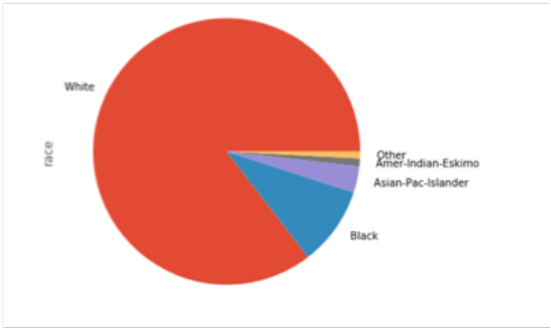
② Relative freq. tables occurances / length

White	0.854274
Black	0.095943
Asian-Pac-Islander	0.031909
Amer-Indian-Eskimo	0.009551
Other	0.008323

③ Bar chart

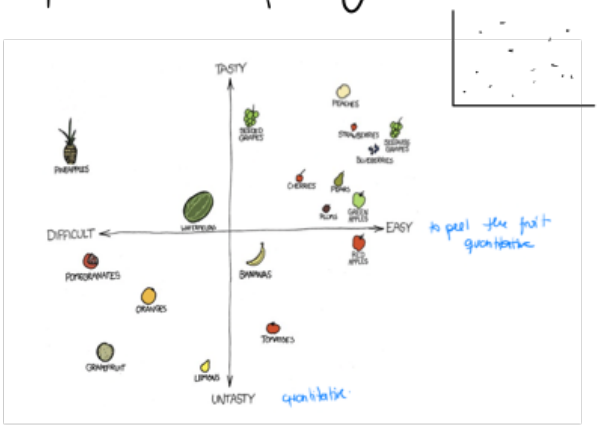


④ Pie chart



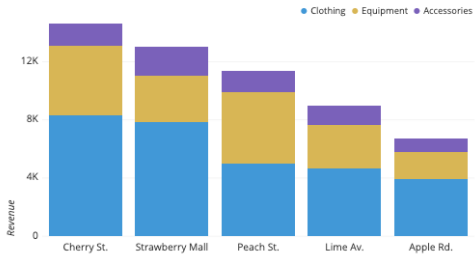
→ Comparing variables

④ Scatter plot : for 2 quantitative variables
careful with overplotting



if you want to add qualitative var. use colors !

For comparing categorical variables: stacked barplot

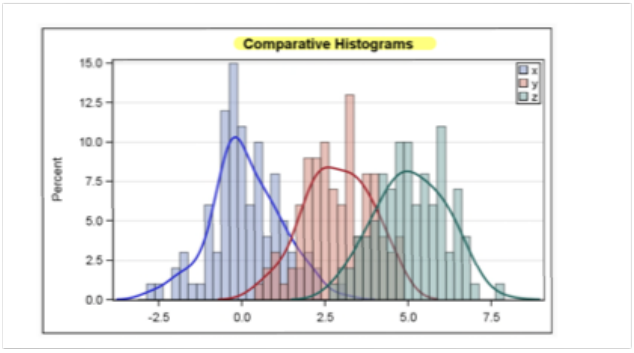


Correlation

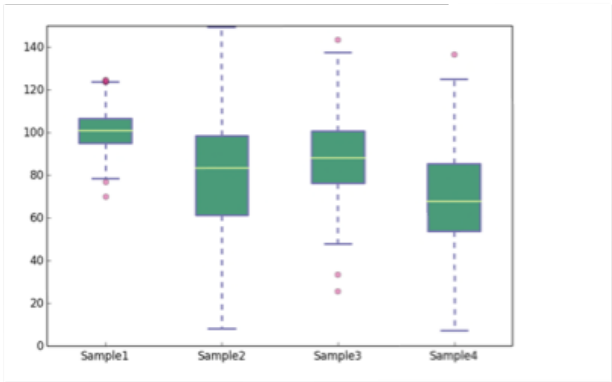
- To use *corr*, there must be a true underlying linear relationship between the two variables.
- The variables must be quantitative.
- Outliers can strongly affect the correlation. Look at the scatterplot to make sure that there are no strong outliers.
- $corr > 0$ --> positive association, $corr < 0$ --> negative association
- $-1 <= corr <= 1$ → grow together → decrease together
- Interchanging x and y does not change the correlation.
- *corr* has no units.

does not affect *corr* → Because it depends on mean

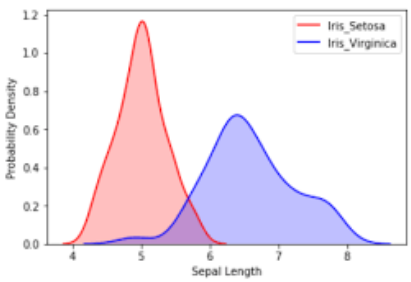
Multiple histograms



Boxplots

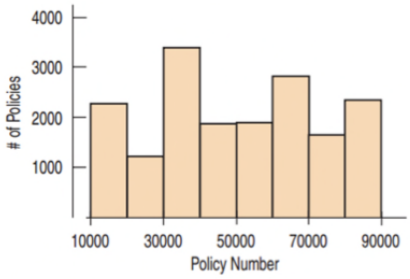


KDE plot →

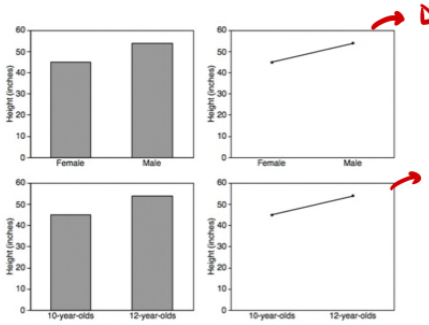


What can go wrong (and will)

- Don't make a histogram for categorical data!



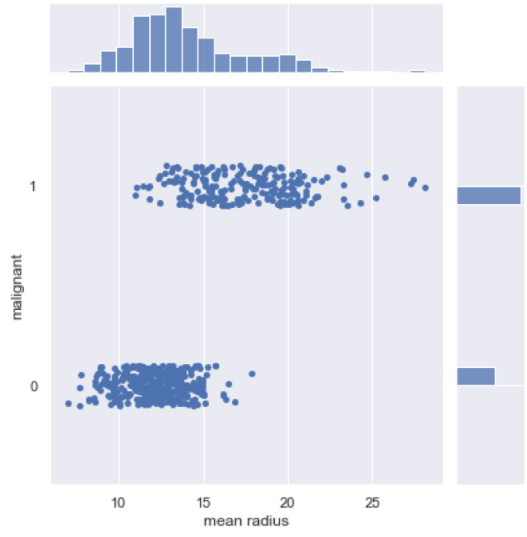
- Don't look for shape, center and spread in a bar chart
- Choose a histogram bin width appropriate for the data
- Do a reality check (don't blindly trust calculators or your computer ** A mean student age of 193 years old is nonsense. ** A (human) heart pulse cannot be 800 bpm
- Beware of outliers, the mean and standard deviation are sensitive to outliers.
- Use a histogram to ensure that the mean and standard deviation really do describe the data.
- Don't compute numerical summaries for a categorical variable (e.g. the mean Social Security number is meaningless)
- Don't do line plots when not appropriate



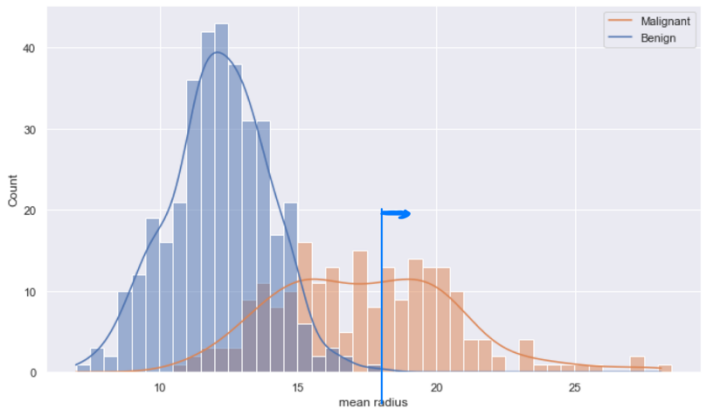
Don't do this
lines imply connection, when it doesn't make sense

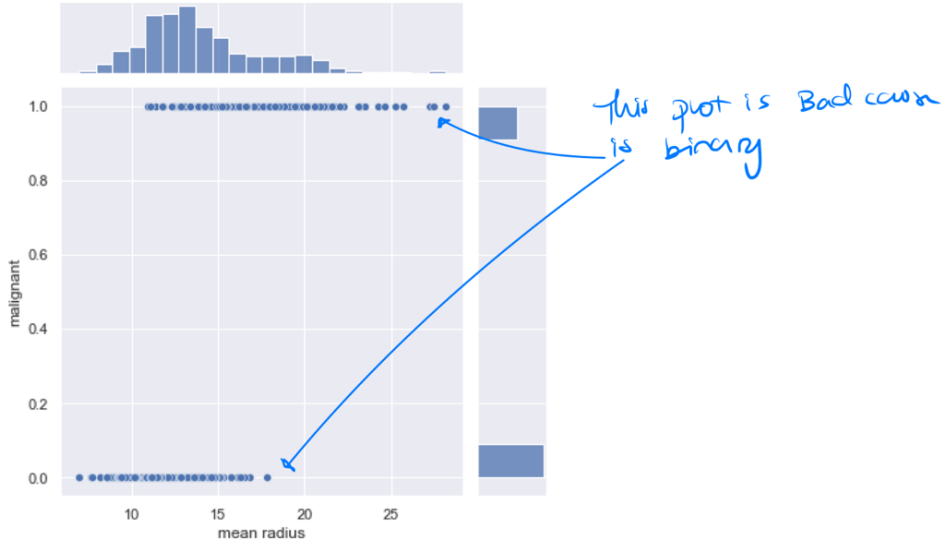
In [7]:

```
# manual to allow for jitter
g = sns.JointGrid(data = data, x = "mean radius", y = "malignant")
g.plot_marginals(sns.histplot)
g.plot_joint(sns.stripplot,
             orient='h', order=[1, 0],
             color=sns.color_palette()[0])
(g.ax_joint).set_xticks([10, 15, 20, 25])
plt.show()
```



Better





This is a clear example of over-plotting. We can improve the above plot by jittering the data:

Ex: how many people survived?

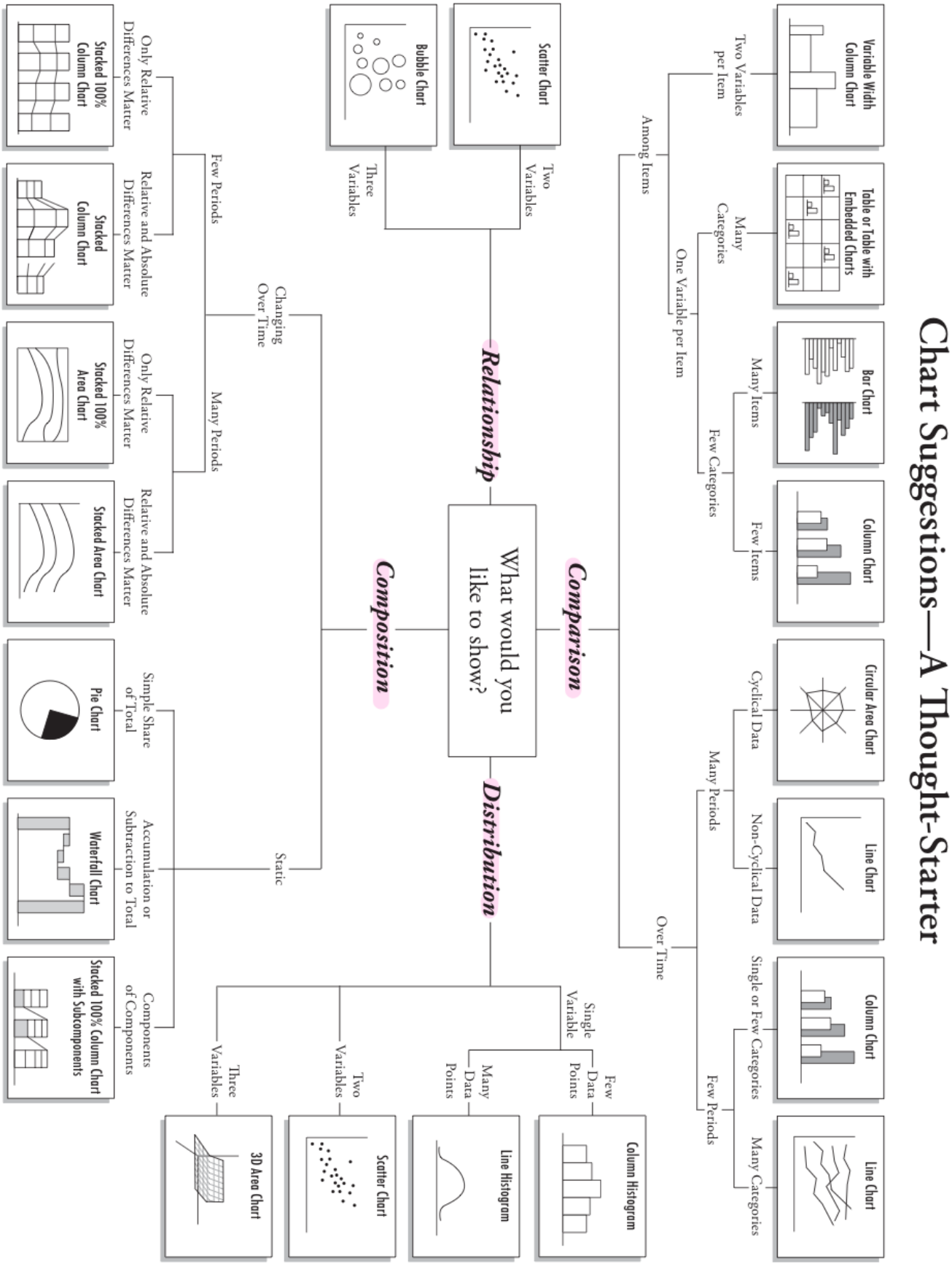
Out[6]:

38% survived

35% are female

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	0.352413	29.699118	0.523008	0.381594	32.204208

4. What type of plot/chart to use? Suggestions

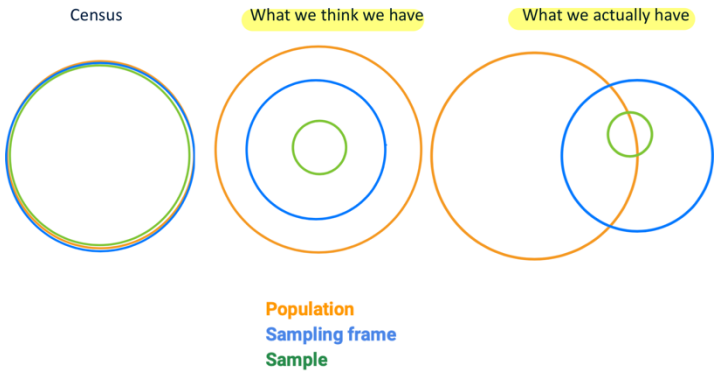


Lecture 2: Data Collection and Sampling

1. Sampling

→ Key Concepts

- Population: group from which you learn about.
- Sampling frame: list from which the sample is drawn (ex: people who completed the survey: participants of a class).
  - If it is the group of people that we want, then is equal to the population of interest.
- Sample: subset of the sampling frame
  - Often used to make inferences about population. However, the sample should be representative of the population.
  - How you draw the sample affects accuracy.
- Sources of error:
  - Chance error: when a random sample varies from what is expected.
  - Bias: systematic error in one direction.
- Types of Samples



	Convenience sample	Quota sample
What?	Whoever you can get ahold of	First specify your desired breakdown of various subgroups and then reach those targets
Example		Sample individuals in your town and you may want the age distribution of the sample to match that of your town's census results
Good?	No, is source of bias	

2. Bias

→ Types of bias

	Selection Bias	Response Bias	Non-response Bias
What is it?	Systematically excluding/favoring groups	When people don't respond truthfully	When people don't respond
How to avoid it	Examine the sampling frame and the method of sampling	Examine nature of questions and method of surveying	Keep surveys short and be persistent

3. Basics about Statistics

→ Statistics

1. Descriptive: describe or summarize features of a dataset.
2. Inferential: learn about the population from which the data was sampled by deducing properties of the underlying **probability distribution** from sampled data.

- Example: The week before an election, it is not possible to ask every voting person who they intend to vote for. Instead, a relatively small number of individuals are surveyed. The hope is that we can determine the population's vote from the surveyed results.
- It attempts to learn about the population from which data was sampled.

→ Data distribution and Random variables

1. Empirical distribution: based on observational data.
   - e.g. think about your answers to a poll. It shows us values and proportions.
2. Probability distribution: gives model for how the sample is generated.
   - e.g. think about what the polling of a student in class would be.
3. Random variable: variable that takes numerical values based on some probabilities.

→ Random Variables

1. *Discrete*

- Probability Mass Function (PMF)
- Distribution types

Distribution	$f(x)$	Support	Mean	Variance
Bernoulli (p)	$p^x(1-p)^{1-x}$	$0, 1$	p	$p(1-p)$
binomial(n, p)	$\binom{n}{x} p^x(1-p)^{n-x}$	$0, 1, \ldots, n$	np	$np(1-p)$

- **Bernulli**: probability distribution of a random variable which takes the value 1 (success) with probability p and the value 0 (failure) with probability q = 1-p.
- **Binomial**: discrete probability distribution "summarizing" the outcome of n Bernoulli random variables.
  - For simplicity, take p = 0.5 so that the Bernoulli distribution describes the outcome of a coin. For each flip, the probability of heads is p (so the probability of tails is q = 1-p). But we don't keep track of the individual flips. We only keep track of how many heads/tails there were in total.
  - So, the binomial distribution can be thought of as summarizing a bunch of (**independent**) Bernoulli random variables.

2. *Continuous*

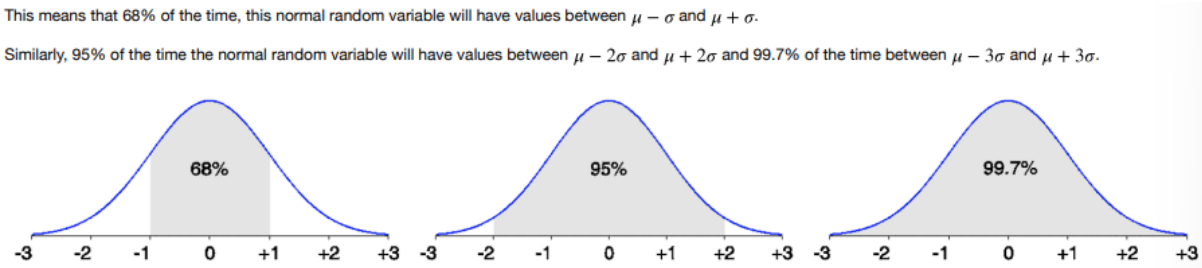
- Probability Density Function (PDF): cannot be less than 0.
- Cumulative distribution function (CDF): cannot be less than 0, nor bigger than 1.

  The probability that a normal random variable takes values in interval [a,b] is given by: $\int_a^b pdf$. Which gives as a result the area under the curve

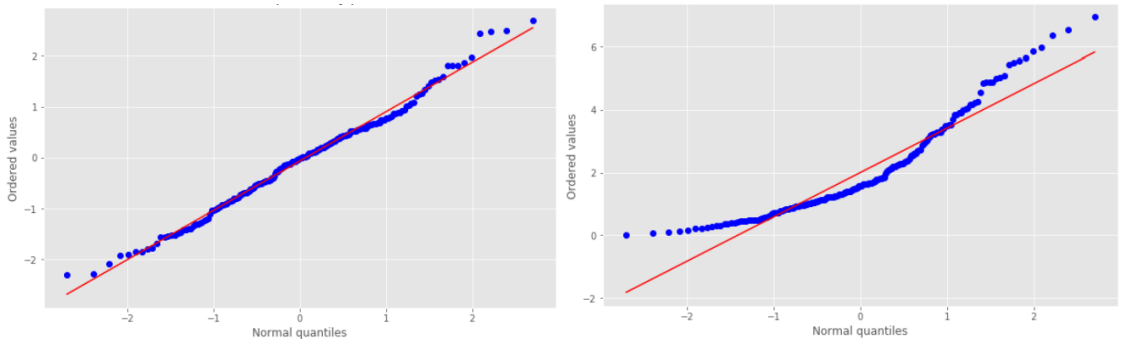
- Distribution types

Name	Probability density function, pdf	Mean	Variance
Uniform $a \le x \le b$	$f(x)=\begin{cases}\frac{1}{b-a}, & 0 \le x \le b \\ 0, & otherwise\end{cases}$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
Exponential $x \ge 0$	$f(x)=\begin{cases}\lambda e^{-\lambda x}, & x \ge 0 \\ 0, & otherwise\end{cases}$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$
Normal $-\infty \le x \le +\infty$	$f(x)=\frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}$	μ	σ^2

- **Normal**: a random variable can take any real value, but some numbers are more likely than others.
  - Computing the CDF for between $\mu - \sigma$ and $\mu + \sigma$, $\mu - 2\sigma$ and $\mu + 2\sigma$, and $\mu - 3\sigma$ and $\mu + 3\sigma$



  - ***Normal Probability Plot***
    - Used to check if a given sample data comes from a normal distribution.
    - the sorted data are plotted vs. values that are selected to make the points look close to a straight line if the data are approximately normally distributed.
    - Deviations from a straight line suggest departures from normality.



4. Hypothesis Testing

- P value: probability that a particular statistical measure will be greater than or equal to the observed results.
  - *Is the probability of the observed statistic given that the null hypothesis is true*.
- Confidence level: how confident we are that our conclusion is correct.
- Significance level: probability of rejecting the null hypothesis when is true.
  - If you reduce it, the region of acceptance gets bigger (reduce power of test), so you are less likely to reject H_0.

→ One Sample Hypothesis testing

- Checking if a random variable comes from a certain distribution?
- Performed when we have one population and a sample is drawn from that population to be tested.
- ***Central Limit Theorem (CLT)***

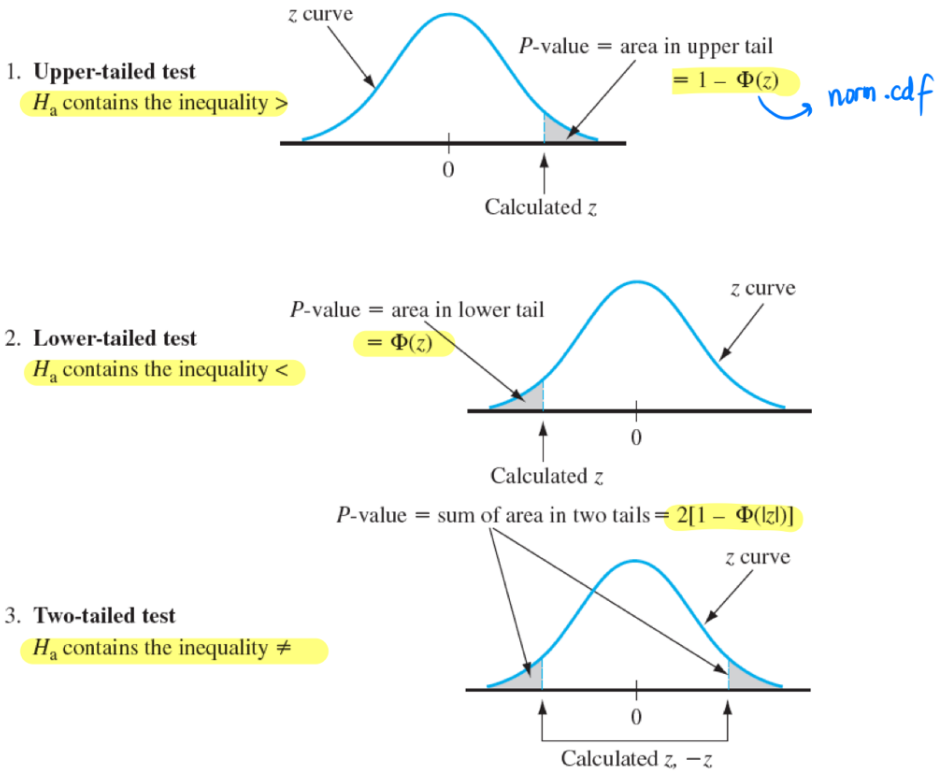
Central Limit Theorem. Let $\{X_1, \ldots, X_n\}$ be a sample of n random variables chosen identically and independently from a distribution with mean μ and finite variance σ^2. If n is 'large', then

- the sum of the variables $\sum_{i=1}^{n} X_i$ is also a random variable and is approximately **normally** distributed with mean $n\mu$ and variance $n\sigma^2$ and
- the mean of the variables $\frac{1}{n}\sum_{i=1}^{n} X_i$ is also a random variable and is approximately **normally** distributed with mean μ and variance $\frac{\sigma^2}{n}$.

- Used to approximate probabilities since any random variable sampled many enough times will be normally distributed.
- Used in hypothesis testing to check if sampled data belongs to a certain probability distribution
- Example:
    - H0: the coin is fair so p = 0.5
    - HA: the coin is unfair so p ≠ 0.5
    - Significance level: α =1%
    - Steps:
        1. Collect a random sample of data from the population of interest. In this case, flip the coin 1000 times and count the number of heads. It turns out to be 545.
        2. Assuming the null hypothesis is true, **compute how likely it is to see a number that is at least as extreme or more extreme as the number obtained from the random sample**. This is called the **p-value**. In the coin flipping we might say that any result >= 545 and <= 455 is 'as or more extreme'.
            a. P(455 < x < 545) = F(545) – F(455) = 99.5% so The probability of seeing 545 heads using a fair coin is 0.05.
        3. Check with significance level. Since 0.05 < 0.1, H0 is rejected and the coin is unfair.

- Hypothesis testing formulation
    - H_a: hypothesis that we want to validate.
    - ∝: Significance level
    - P_{value}: smaller the P, stronger H_0
    - $P_{value} < \propto$ → reject H_0

- One and two- sided hypothesis testing:



- ***Normal Probability plot***: sorted data are plotted vs calculated values that show the deviation of the data from the norm.

Deviation from a straight line in plot suggest departures from normality.
Higher z values represent more variation from norm.

$$z_i = \frac{x_i - \bar{x}}{s}.$$

- **Z-test:**
  - Same process as before (hypothesis testing using CLT) but for means. In particular, **it tests whether the sample mean is statistically different from the true mean of the population, given hypothesized value.**
  - Steps:
    - Identify the parameter of interest and describe it in the context of the problem.
    - Determine the null and alternative hypotheses.
    - Choose a significance level α.
    - Find the formula for the computed value of the test statistic, *e.g.*, $Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}$ (use the CLT).
    - Using the sampled data, compute the P-value, *e.g.*, $F(z)$
    - Compare the significance level to the P-value to decide whether or not the null hypothesis should be rejected and state the conclusion in the problem context. Report the P-value!

Ex: If Tesla has z-score = +2.2 means that Tesla achieves gains 2.2 standard deviations better than average car.

  - Example

Question: Does this data provide compelling evidence for concluding that the population mean zinc mass exceeds 2.0 grams, as reported by the company?

Let μ denote the true average zinc mass of such batteries. We take the null hypothesis to be

$$H_0 : \mu \leq 2.0$$

and that the alternative hypothesis is

$$H_a : \mu > 2.0.$$

We set a significance level of, say, $\alpha = 1\%$.

According to the CLT, the sample mean $\bar{X}$ has approximately a normal distribution with mean $\mu = 2.0$ (Assuming H_0) and standard deviation $S/\sqrt{n}$. We could proceed as previously but it is more standard to normalize the variable.

To proceed, we conduct a **z-test**, which is the same as we did in the previous example, except now we use the normalized Z-statistic,

$$Z = \frac{\bar{X} - 2.0}{S/\sqrt{n}}.$$

The Z-statistic is distributed according to the "standard" normal distribution with mean $\mu = 0$ and standard deviation $\sigma = 1$.

Assuming H_0 to be true, how unlikely is it that we would observe such a large value ($\bar{X} = 2.06$)?

$$P - \text{value} = P(\bar{X} \geq 2.06) = P(Z \geq 3.04) = \int_{3.04}^{\infty} f(x)dx = 1 - F(3.04). = .0012.$$

Because the P-value $= .0012 \leq .01 = \alpha$, the null hypothesis should be rejected at the chosen significance level. We conclude that the average zinc mass in the batteries exceeds 2.0.

- **Student's t-test**
  - When is small, the Central Limit Theorem can no longer be used. In this case, if the samples are drawn from an approximately normal distribution, then the correct distribution to use is called the Student's t distribution with n-1 degrees of freedom.
  - For a sample size greater than 20 (or 30 or 40), the normality assumption of the student's distribution is generally accepted as reasonable.

→ Two Sample Hypothesis testing

- Performed to compare two samples from two different populations (or treatment groups) and determine whether there is a statistically significant difference between the two populations or treatment groups A and B.
- Quantitative variables
  - Compare the means of the two populations.
  - **Two sample t-test**
    - When we have 2 *independent* samples, we compare the means with a two sample -test (this is more robust to small sample sizes than z-tests).
    - Assumptions
      - independent observations
      - normally distributed responses within each group (or sufficiently large sample size).
    - H0: $\mu_F = \mu_M$
    - HA: $\mu_F \neq \mu_M$

Then the t-statistic $t = \frac{\bar{X}_M - \bar{X}_F}{\sqrt{\left(\frac{s_M^2}{N_M} + \frac{s_F^2}{N_F}\right)}}$ is expected to follow a $t_{min}(n_F, n_M) - 1$ distribution so i can use that one to compute the p-value.

    - $df = n_M + n_f - 2$
    - Compare value in the t-distribution using alpha and df. If our t-statistic is smaller than this value then H0 is rejected
- Categorical variables
  - Compare the proportions of the two populations.
  - **Two sample *z*-test**
    - H0: $p_F = p_M$
    - HA: $p_F \neq p_M$
    - When we have 2 *independent* samples of a binary categorical variable, we compare the proportions (i.e. percents) with a two sample z-test.
    - Assumptions
      - independent observations
      - normally distributed responses within each group (or sufficiently large sample size).

We make the following definitions:

- N_A is the number of surveyed people from population A
- n_A is the number of successes from population A
- $p_A = n_A/N_A$ is the proportion of successes from population A

Similarly, we define

- N_B is the number of surveyed people from population B
- n_B is the number of successes from population B
- $p_B = n_B/N_B$ is the proportion of successes from population B

We make the null hypothesis:

H_0: p_A and p_B are the same, that is, $p_A - p_B = 0$.

That is, the proportion of successes in the two populations is the same.

    - Find $\hat{p} = \frac{N_A}{N_A+N_B}p_A + \frac{N_B}{N_A+N_B}p_B$, $\hat{q} = 1 - \hat{p}$, and $Z = \frac{p_A - p_B}{\sqrt{\hat{p}\hat{q}\left(\frac{1}{N_A}+\frac{1}{N_B}\right)}}$
    - Find the p-value = 1 – CDF of norm (z), if it is smaller than the significance level then the null hypothesis is rejected.

→ Other tests

- **A/B Testing**
  - A/B testing is a method of comparing two or more versions of an advertisement, webpage, app, etc.
  - We set up an experiment where the variants are shown to users and random and statistical analysis is used to determine which is best.
  - AB testing is the de facto test for many business decisions (e.g. if search results click through increases by having an underlined link).

- **χ^2 test for independence**
  - Correspondent of descriptive statistics for quantitative variables to measure the association between categorical variables.
  - H0: the two categorical variables are independent.
  - HA: the two categorical variables are not independent.

    Formally, the χ^2 test for independence can be calculated as follows:

    H_0: the 2 categorical variables are independent
    H_A: the 2 categorical variables are not independent

    The value of the statistic (based on the contingency matrix) is:

    $$\chi^2 = \sum_{\text{all cells}} \frac{(Obs - Exp)^2}{Exp}$$

    where:

    Obs is the observed cell count (as read from the matrix) and
    Exp is the expected cell count (iff independent), i.e. $Exp = \frac{\text{rowtotal} \times \text{columntotal}}{n}$

    The p-value can then be calculated based on a $\chi^2_{df=(r-1)\times(c-1)}$ distribution (r is the # of rows and c is the # of columns).

  - The larger the χ^2 value, the greater the probability that there really is a significant difference.

- **ANOVA test**
  - Used to compare more than 2 means of quantitative variables.

    H_0: the mean response is equal in all K groups.
    H_A: there is a difference in mean response somewhere among the group.

    $$F = \frac{\frac{\sum_{k=1}^{K} N_K(\bar{Y}_k - \bar{Y})^2}{K-1}}{\frac{\sum_{k=1}^{K}(n_k-1)\sigma_k^2}{n-K}}$$

    where:
    n_k is the sample size in group k,
    $\bar{Y}_k$ is the the mean response in group k,
    σ_k is the variance of responses in group k,
    n, Y, σ (without indices) are the same as above but for the overall sample

    The p-value can then be calculated based on a Fdf1=(K−1),df2=(n−K) distribution.

→ Summary of Hypothesis Testing methods

Variable type	# Groups	Classic approach
Quantitative	2	t-test
>>	3+	ANOVA
Binary (proportions)	2	z-test
>>	3+	χ^2 test
Categorical	2+	χ^2 test

Types of error in hypothesis testing

- *A type I error* is the incorrect rejection of a true null hypothesis (a "false positive").
- *A type II error* is incorrectly accepting a false null hypothesis (a "false negative"). Depending on the application, one type of error can be more consequential than the other.

Table of error types		Null hypothesis (H_0) is	
		Valid/True	Invalid/False
Judgment of Null Hypothesis (H_0)	Reject	Type I error (False Positive)	Correct inference (True Positive)
	Fail to reject	Correct inference (True Negative)	Type II error (False Negative)
Type I = True H_0 but reject it (False Positive) Type II = False H_0 but fail to reject it (False Negative)			

- The probability of making a type I (false positive) error is the significance level.
- The probability of making a type II (false negative) error is more difficult to calculate.
- Example
    - In vaccine testing, we take the null hypothesis (H0): "This vaccine has no effect on COVID19." A type I error detects an effect (the vaccine is effective against the disease) that is not present. A type II error fails to detect an effect (the vaccine is effective) that is present.

P hacking

- Doing enough hypothesis tests to eventually have a Type I (false positive) error but that proves the hypothesis.
- Can prevented using cross-validation.

Confidence interval

- A confidence interval is an interval estimate for an unknown population parameter. For example, we might use collected data to give an interval estimate for a population mean μ.
- Considering a sampled mean $\overline{X}$, the CLT tells us that the sample mean is normally distributed with expected value μ and standard deviation $\sigma/\sqrt{n}$.
- Then, the 95% CI $\bar{X} - 1.96\frac{\sigma}{\sqrt{n}} < \mu < \bar{X} + 1.96\frac{\sigma}{\sqrt{n}}$.

The 95% confidence interval means that if we were to repeat the same experiment many times, and compute the confidence interval using the same formula, 95% of the time it would contain the true value of μ.

Lecture 3: Modeling, Regression, Cross-Validation, Regularization

1. Introduction

→ Modeling Process

1. Choose a model	How should we represent the world?
2. Choose a loss function	How do we quantify prediction error?
3. Fit the model	How do we choose the best parameters of our model given our data? Find the params that model involve
4. Evaluate model performance	How do we evaluate whether this process gave rise to a good model?

2. Simple Linear Regression (SLR): one feature

- Regression line: Straight line that minimizes the mean squared error (MSE) of estimation among all straight lines.
- We want to find value that minimizes the MSE (simple: take the derivatives)

Model	*SLR*	$\hat{y} = a + bx$
Loss Function	L2	$L(y, \hat{y}) = (y - \hat{y})^2$ $R(a,b) = \frac{1}{n}\sum_{i=1}^{n}(y_i - (a + bx_i)^2$
Fit model	Calculus	$\hat{a} = \bar{y} - \hat{b}\bar{x}$ $\hat{b} = r\frac{\sigma_y}{\sigma_x}$
Evaluate model	R2, RMSE, Linearity checks	

→ Loss function

- They help to measure how good or bad our predictions are by quantifying how good or bad a single prediction is.
- It characterizes the cost, error, or fit from a choice of model or model parameters.
- Types for each data point:

  - Squared Loss (L2 Loss) $L(y, \hat{y}) = (y - \hat{y})^2$
  - Absolute Loss (L1 Loss) $L(y, \hat{y}) = |y - \hat{y}|$

  Which one to use depends on problem. But generally: squared penalizes more (makes errors weigh more). And if you have many outliers, squared is ore strict with it because it gives higher values.
  In SLR we normally use L2 loss

- Types for entire data set:
  - Average loss/Empirical Risk

Given data $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$:

$$R(\theta) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \hat{y}_i)$$

- Bad loss functions examples

Examples values

A ~~$y - \hat{y}$~~ $y: 1$ $\hat{y}: 0.9$

B ~~$\frac{1}{y - \hat{y}}$~~ $y: 0.9$ $\hat{y}: 1$

C $(y - \hat{y})^2$ $y: 2$ $\hat{y}: 3$

D ~~$e^{-(y-\hat{y})^2}$~~ $\frac{1}{e^{(y-\hat{y})^2}}$

  - B: if $y = \hat{y}$ then is undefined / it leads to bigger outputs when the difference is smaller
  - D: it outputs lower values if the difference is bigger
  - A: it might give + or – .

→ How to check for a good linear model? Model evaluation

1. *Performance measure: Metrics*
   - Lower values indicate more accurate predictions.
   - Sum of squared residuals (SSR): this gives any value, and we want smallest possible, so knowing how small is small is difficult.

$$SSR = \sum_{i=}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2.$$

   - Total sum of squares (TSS): the error that you have if you have a lazy model (simplest possible, that only uses the mean).
     It measures the total variance in data.

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

   - R^2: How much of the target variable's variance the model explains = 1-(SSR/TSS)

$$R^2 = 1 - \frac{SSR}{TSS}.$$

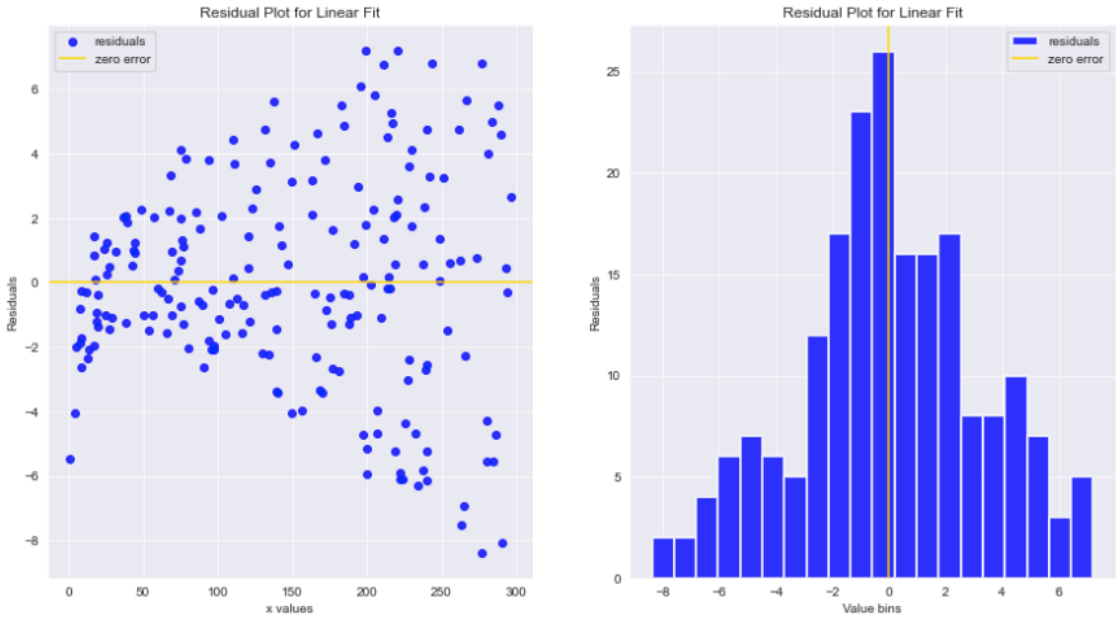
     - is a proportion so it takes values [0,1]
     - In linear regression this is the same as the ***correlation*** (squared).
     - Nearly 1: good model
     - If $R^2 = 0.612$ → the model explains 61% of the variability in the dependent variable.
   - Mean square error (MSE)
   - Root mean squared error (RMSE)

$$RMSE = \sqrt{\frac{\sum_{\text{houses in test set}} (\text{actual price of house} - \text{predicted price of house})^2}{\#\text{ of houses in test set}}}$$

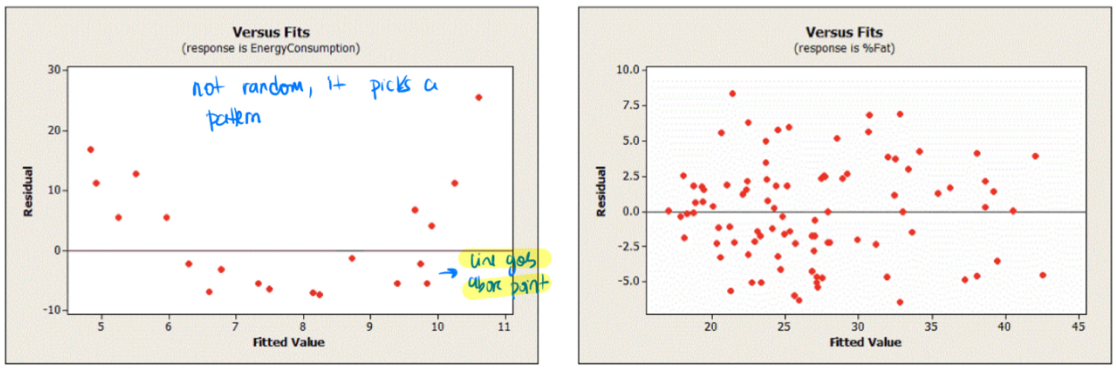
2. *Linearity check*
   - by plotting original data, computing correlation and checking for linearity.
   - The dependent variable (Y) is assumed to have linear relationship with the independent variable X. Easy to check with 1 variable but more complex with many.
   - Avoid ***multicollinearity*** among features: because the models assumes independence among variables, so having collinearity breaks the assumption.

3. *Residuals plot*

- Plot can help you identify patterns in the data that was hard to see in a original plot.
- Residuals are assumed to be normally distributed, independent and to have constant variance. These assumptions need to be verified with the data.
- We <u>cannot</u> talk about <u>outliers</u> from these plots.
- Looking into plot of $e_i = y_i - \hat{y}_i$
- Types of plots to create:
  - a plot of e_i with respect to x_i. This allows us to compare the distribution of the noise at different values of x_i.
  - a histogram of e_i. This allows us to explore the distribution of the noise independent of x_i, or a plot of e_i with respect to fitted values (y_i)
- You can plot the normal prob. plot of the residuals and see if residuals are normal. However, this only makes sense if you have many data points (CLT).
- Results:
  - Random distribution of dots around line → the model is good (ex: linear is good), but this doesn't mean good prediction.
    - *Randomness is good around 0 because being all in + would mean that model is constantly overshooting.*
  - Horizontal line of points at 0 → perfect predictive power



Which one is a better fit? Note that these are plotting residuals vs. fitted values.



A. Left
B. Right

The curvature in Left suggests that model is inappropriate. This could also be said as: the regression error is not constant, as is wider at both extremes.

→ Coefficients θ

- $f(x) = \theta_0$ then theta is the mean of the total because it decreases the error.

Suppose we fit the following linear regression models to predict `total`. Based on the data and visualizations shown in the block introduction, determine whether the fitted model weights are positive (+), negative (-), or exactly 0. The notation `meat=beef` refers to the one-hot encoded meat column with value 1 if the original value in the meat column was beef and 0 otherwise. In the next question shortly justify your answer.

- $f(\mathbf{x}) = \theta_0$
  Weight of θ_0 [+]

  (a)
  θ0 is the mean of total, which is positive

- $f(\mathbf{x}) = \theta_0 + \theta_1 \cdot veg^2$
  Weight of θ_0 [+]
  Weight of θ_1 [+]

  θ0 is the total when veg^2=0 is positive
  θ1 is positive as total increases when veg (or its square) increases

- $f(\mathbf{x}) = \theta_0 + \theta_1 \cdot (meat = beef) + \theta_2 \cdot (meat = chicken)$
  Weight of θ_0 [+]
  Weight of θ_1 [-]
  Weight of θ_2 [-]

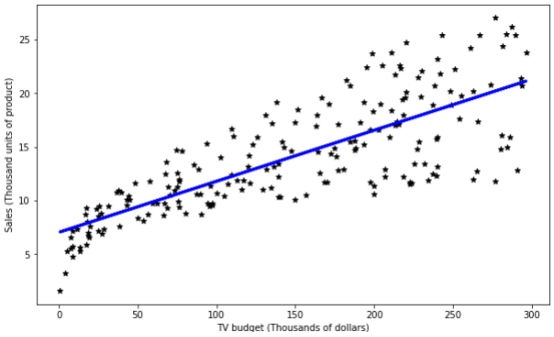
  θ0 is the total when meat=fish which is positive
  θ1 is the difference between total when meat=beef and total when meat=fish, which is negative
  θ2 is the difference between total when meat=chicken and total when meat=fish, which is negative

- $f(\mathbf{x}) = \theta_0 + \theta_1 \cdot (meat = beef) + \theta_2 \cdot (meat = chicken) + \theta_3 \cdot (meat = fish)$
  Weight of θ_0 [not enough info]
  Weight of θ_1 [not enough info]
  Weight of θ_2 [not enough info]
  Weight of θ_3 [not enough info]

  This model is an extension of the previous model, however θ3 is not needed. There are thus an infinite number of possible combinations of θ0, θ1, θ2, θ3. We cannot say for sure what they will be.

  Extra credit from P (in the exam):

- **Scatter plot with Regression Line Interpretation**
  - $target = \beta_0 + \beta_1 \times feature_1 + \cdots$
  - β_0 is the intercept and represents the effect on the target if there are no features
    - If no data is collected for when features=0, then β_0 doesn't mean anything and is just a calibration for the equation.
    - It shifts the line up or down.
    - If $\beta_0 = 0$ then line needs to pass throw origin.
  - β_1 is the coefficient of $feature_1$ and, multiplied by 10, represents how much the target changes if $feature_1$ increases one unit.



4. Multiple Linear Regression: more than one feature

Model	Multiple Linear Regression	$\hat{\mathbb{Y}} = \mathbb{X}\theta$
Loss Function	L2 Loss MSE	$R(\theta) = \frac{1}{n}\|\|\mathbb{Y} - \mathbb{X}\theta\|\|_2^2$
Fit model	1. Calculus 2. Numerical Methods	1. $\hat{\theta} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{Y}$ 2. $\vec{\theta}^{(t+1)} = \vec{\theta}^{(t)} - \alpha\nabla_{\vec{\theta}}L(\vec{\theta}, \mathbb{X}, \vec{y})$
Evaluate model	Adjusted R2, Visualize	

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ 1 & x_{31} & x_{32} & \dots & x_{3p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_p \end{bmatrix}$$

$$\hat{\mathbb{Y}} = \mathbb{X}\theta$$

Prediction vector $\mathbb{R}^n$ — Design matrix $\mathbb{R}^{n\times(p+1)}$ — Parameter vector $\mathbb{R}^{(p+1)}$

→ Interaction Term

- $target = \beta_0 + \beta_1 \times TV + \beta_2 \times Radio + \boldsymbol{\beta_3 \times Radio \times TV}$
- Interaction term is $\beta_3 \times Radio \times TV$, which express the combined effect of Radio and TV together
- Given that $\beta_3 = 0.0011$ and $\beta_1 = 0.019$, for every unit increase of TV, the target increases by $0.019 + 0.0011 \times Radio\_units$

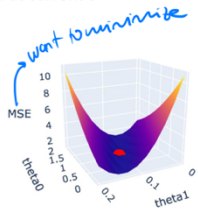
→ Gradient descent

- Algorithm to estimate parameters and find best solution (minimizing error).

gradient of the avg. loss function evaluated at current θ

$$\vec{\theta}^{(t+1)} = \vec{\theta}^{(t)} - \alpha \nabla_{\vec{\theta}} L(\vec{\theta}, \mathbb{X}, \vec{y})$$

Next value for **θ**



ϑ: Model weights
L: average loss function
α: Learning rate
y: True values from training data

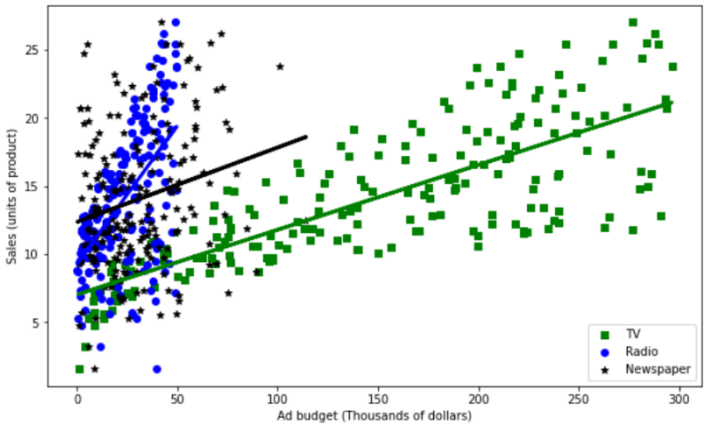
→ How to correct for adding to many predictors (features)?

$$\text{adj.}R^2 = 1 - \frac{(1 - R^2)(n - 1)}{n - p - 1}$$

This decreases if there is no significant gain in the performance after adding a predictor.
p: number of predictors
n: number of samples

→ Multiple Linear Regression Plot interpretation

- When features' relationships overlap, the p-value of the worse feature will be non-zero (so not significant), whereas the relevant feature will be significant.
- The model will perform better without this feature.
- In this case, Radio and Newspaper overlap and Newspaper is the redundant feature.




```
In [15]:
print(advert.corr())
pd.plotting.scatter_matrix(advert, figsize=(10, 10), diagonal='kde')
plt.show()
```

	TV	Radio	Newspaper	Sales
TV	1.000000	0.054809	0.056648	0.782224
Radio	0.054809	1.000000	0.354104	0.576223
Newspaper	0.056648	0.354104	1.000000	0.228299
Sales	0.782224	0.576223	0.228299	1.000000

5. Feature Engineering

- Process of transforming raw input into informative features that can be used in modeling or EDA.

→ One-hot encoding:

- When you have categorical variables (ex: gender, ethnicity, gender) and you want to include them in the regression model.
- Two categories (female, male):
  - you can add a new feature 'Gender_num'

$$\text{Gender\_num}_i = \begin{cases} 1 & \text{if } i\text{-th person is female} \\ 0 & \text{if } i\text{-th person is male} \end{cases}.$$

$$\text{Income} = \beta_0 + \beta_1 \times \text{Gender\_num}.$$

- If you have more options (Asian, Caucasian, African American):
  - Ordinal
    - Category1=0, Category2=1, Category3=2
  - Not ordinal
    - Asian = {0,1}, not_Asian = {0,1}, Caucasian = {0,1}, not_Caucasian = {0,1}
    - you should create a column per option -1 (you always need one less variable) as you use it as baseline.

$$\text{Asian}_i = \begin{cases} 1 & \text{if } i\text{-th person is Asian} \\ 0 & \text{otherwise} \end{cases}.$$

$$\text{Caucasian}_i = \begin{cases} 1 & \text{if } i\text{-th person is Caucasian} \\ 0 & \text{otherwise} \end{cases}.$$

The value with no dummy variable--African American--is called the *baseline*.

You still need to create the column, but you do it with inbuilt pd.get_dummies

6. Hypothesis testing

- Assuming that there is a known relationship β_0 and β_1 between the true target y and the true features x

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

- Considering our dataset as a sample from these true x and y, an error ε_i is introduced into the sample. Assuming that it comes from a normal random variable with zero mean and variance σ^2.

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i.$$

- The aim is to **test whether our "sampled" dataset have a relationship between x and y**
  - Hypothesis:

H_0 : There is no relationship between x and $y \iff \beta_1 = 0$ want to disprove

H_a : There is a relationship between x and $y \iff \beta_1 \neq 0$

- If we fail to reject H_0 then the variable has little influence.
- The test statistic is *t* is computed as:

$$t = \frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)}$$

- From the formulas:
  - SE: standard error
  - The bigger the noise (represented by σ^2), the bigger the SE
  - The bigger the variance (represented by $\sum_{i-1}^{n}(x_i - \bar{x})^2$), the smaller the SE

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \quad \text{and} \quad \hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$SE(\hat{\beta}_0)^2 = \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \right) \quad \text{and} \quad SE(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

- Under the assumptions of the null hypothesis, the test statistic is distributed according to the t-distribution with n-2 degrees of freedom. The p-value is computed as the probability of observing a value as extreme as |t|. A **small p-value** is interpreted to mean that **there is an association between the independent and dependent variables**

Question (poll):

$$SE(\hat{\beta}_0)^2 = \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \right) \quad \text{and} \quad SE(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

error contributes to this

Proposition 1: "The bigger the noise, the bigger the SE" True or False? error goes into σ^2

Proposition 2: "As x points are more spread, the bigger the SE" True or False?

smaller spread: there are multiple lines that I can draw so the SE will be bigger.

spread over x axis helps gives better approximation

1.4 Checks for a good linear model

Out[14]:

OLS Regression Results

Dep. Variable:	Sales	**R-squared:**	0.897
Model:	OLS	**Adj. R-squared:**	0.896
Method:	Least Squares	**F-statistic:**	570.3
Date:	Thu, 16 Feb 2023	**Prob (F-statistic):**	1.58e-96
Time:	10:37:00	**Log-Likelihood:**	-386.18
No. Observations:	200	**AIC:**	780.4
Df Residuals:	196	**BIC:**	793.6
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	2.9389	0.312	9.422	0.000	2.324	3.554
TV	0.0458	0.001	32.809	0.000	0.043	0.049
Radio	0.1885	0.009	21.893	0.000	0.172	0.206
Newspaper	-0.0010	0.006	-0.177	0.860	-0.013	0.011

Omnibus:	60.414	**Durbin-Watson:**	2.084
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	151.241
Skew:	-1.327	**Prob(JB):**	1.44e-33
Kurtosis:	6.332	**Cond. No.**	454.

Better model, it explains almost 90% of variability

small = good
Big = Bad

P>|t| is p value

[0.25 0.975] refer to 95% CI

newspaper is not significant anymore
But look at P value

negative value means that spending more would give losses

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Interpretation

Spending an additional \$1,000 on radio advertising results in an increase in sales by 189 units **iff all other spending remains the same (i.e. TV and newspapers)**. Radio is the most effective at method of advertising.

In multilinear regression, the F-test is a way to test the null hypothesis,

$$H_0 = \text{all coefficients are zero.}$$

In this case, we see that the p-value for the F-statistic is vanishingly small - indicating that our model is significant.

0.0458 0.1885

TV: 0.04
R: 0.2

Now let's consider the individual coefficients in the model and their p-values. Note that the coefficients for TV and Radio are approximately the same as for simple linear regression. The coefficient for Newspaper changed significantly. Furthermore, note that the p-value is now very large $p = 0.86$. There is not sufficient evidence to reject the null hypothesis that the Newspaper and Sales variables have no relationship.

So then why did the simple linear regression give that there is a relationship between Newspaper and Sales Variables? *Newspaper is actually a confounder!* (Think about spurious correlations here). Let's look at the correlations between the four variables. Recall that correlation between two variables is given by

$$r_{x,y} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{s_x s_y}.$$

Plotted below is also a scatter plot matrix, which is a good way of visualizing the correlations.

The correlation between Newspaper and Radio is 0.35, which implies that in markets where the company advertised using Radio, they also advertised using newspaper. Thus, the influence of Radio on Sales can be incorrectly attributed to Newspaper advertisements!

This leads us to the following linear regression model, where we forget about Newspaper advertisements:

$$\text{Sales} = \beta_0 + \beta_1 \times \text{TV\_budget} + \beta_2 \times \text{Radio\_budget}$$

this gives same R^2

A2. Considerations in the linear model: Interaction terms

We can consider the interaction between TV and Radio advertising in the model, by taking

$$\text{Sales} = \beta_0 + \beta_1 \times \text{TV\_budget} + \beta_2 \times \text{Radio\_budget} + \beta_3 \text{TV\_budget} \times \text{Radio\_budget}.$$

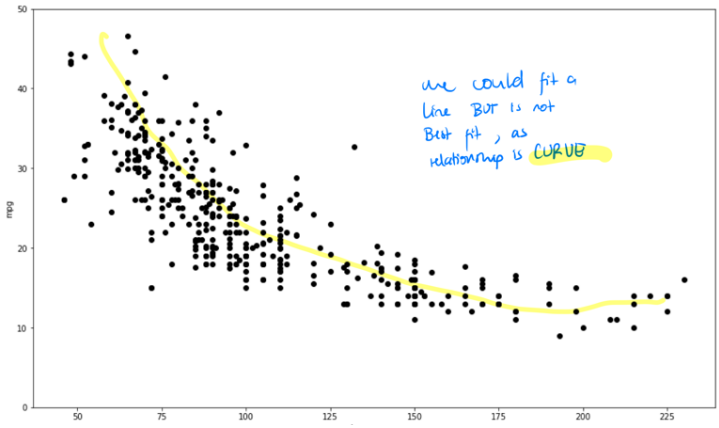
The rational behind the last term is that perhaps spending x on television advertising and y on radio advertising leads to more sales than simply $x + y$. In marketing this is known as the *synergy effect* and in statistics it is known as the *interaction effect*.

Note: even though the relationship between the independent and dependent variables is nonlinear, the model is still linear.

Another note: If we include an interaction in a model, we should also include the main effects, even if the p-values associated with their coefficients are not significant (*hierarchy principle*). The rationale for this principle is that interactions are hard to interpret withouth the main effect.

7. Non-linear in Linear Regression model

- Non-linearity can be introduced in a Linear Regression Model by increasing the power of the features



We consider the linear model

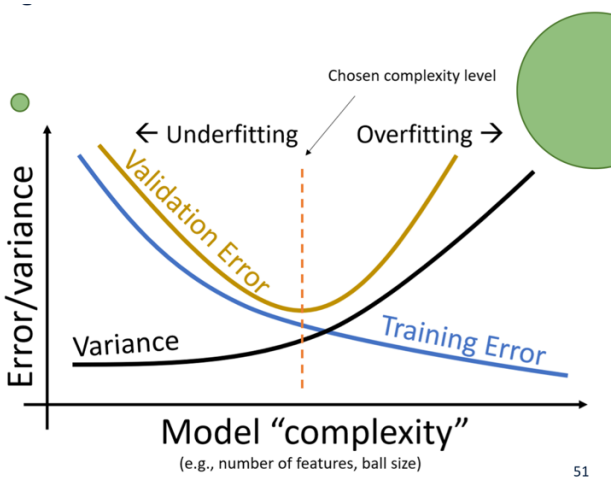
$$\text{mpg} = \beta_0 + \beta_1 \times \text{horsepower} + \beta_2 \times \text{horsepower}^2 + \cdots + \beta_m \times \text{horsepower}^d$$

It might seem that choosing d to be large would be a good thing. After all, a high degree polynomial is more flexible than a small degree polynomial.

- As the degree of a model increases, R^2 becomes larger (you can prove that this is always true if you add more predictors), but at some point, error will increase too.
  - Degree is not the same as the number of features/predictors, it refers to complexity.
- However, the p-value, as you increase the degree might become larger → showing that a strong relationship between variables doesn't exist.
- When in ties, choose simplest model (Occam's razor)

8. Cross-Validation

- Is a method that asses how the results of a predictive model will generalize to an independent testing set.
- Idea method
  - Split data into training (normally 70-90%) and testing (validation set) → Holdout Method
  - Train a variety of models on training.
  - Check the accuracy of each model on the testing dataset.
  - Determine which model is best.
- As we increase complexity of model → validation error (test error) decreases but then increases



→ Variance and Bias

- Bias: error caused by simplifying assumptions built into the method.
  How good prediction of model is depending on what I have?
  - Low bias: low prediction error, better accuracy.
- Variance: how much the model will change based on the sampled data.
  - Low variance: if data changes, the model doesn't.
  - High variance: not adapting well to unseen data. *(Overfitted model)*
- Cases:

- If we increase model complexity → variance increases, and bias decreases.
- If we add more samples, coming from the same stationary distribution as original data → variance decreases.

→ Generalizable models

- Model that does not only perform well on the training dataset but on similar data that the model hasn't seen yet (training).

→ Validation split

- Hyperparameter: value that controls the learning process (ex: degree of the model equation).
- To determine quality of a hyperparameter, we train on **training set** (one time per parameter), and then calculate model's error on **validation set** → check the **quality**.
- Because doing it with the same data for all parameters might be biased, we do **K-Fold Cross validation:**

- *K-fold cross validation*
  - We calculate the quality of 1 parameter on k datasets (subsets from original).
  - *Quality of hyperparameter = average of the k sets errors.*
  - Repeat process if you have other values for the hyperparameter.
  - Example on the right is for one hyperparameter, ex: alpha = 0.1

  - If k increases: each model is trained on a larger training set and tested on a smaller test fold. This leads to a lower prediction error as the model sees more of the available data.

9. Regularization: controlling overfitting

- We make something regular by adding information which creates a solution that prevents overfitting. The "something" we're making regular in our context is the "objective function", something we try to minimize during the optimization problem.
- Uses idea of -constraining our parameter search space- by adding a penalty term to the risk or error, so that a smaller subset of the entire data is chosen or given importance to.
  - Bigger ball: you will find optimal solution.
  - Smaller ball: closer/more similar values for the parameters

Let's return to our 9 feature model from before (d = 9). → which features to choose?

- If we pick a very small ball radius, what kind of model will we have?

$$\hat{y} = \theta_0 + \theta_1 x_1 + \ldots + \theta_9 x_9$$

a. A model that only returns zero.
b. A constant model.
c. Ordinary least squares.
d. Something else.

If the ball is very tiny, our gradient descent is stuck near the origin.

- If all parameters are zero including intercept, model always outputs zero.
- If all parameters are zero except intercept, model is a constant model (returns mean of the observations).

Let's return to our 9 feature model from before (d = 9).

- If we pick a very large ball radius, what kind of model will we have?

$$\hat{y} = \theta_0 + \theta_1 x_1 + \ldots + \theta_9 x_9$$

a. A model that only returns zero.
b. A constant model.
c. Ordinary least squares. → normal model
d. Something else.

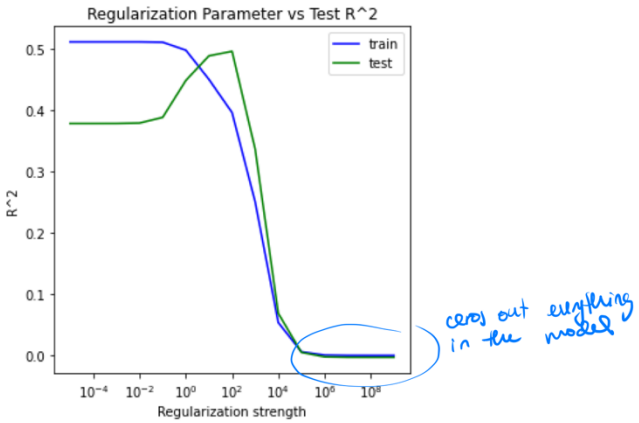
hp weight displacement hp^2 hp weight hp displacement weight^2 weight displacement displacement^2

→ When to use regularization?

1. When bias in model is too low: we use regularization because we want to add some bias into our model to prevent it overfitting to our training data.
2. Model variance is too high: means that model is not adapting well to unseen data (maybe for overfitting?)
3. If you have linearly dependent features
4. Weights are too large.
5. Training error is 0

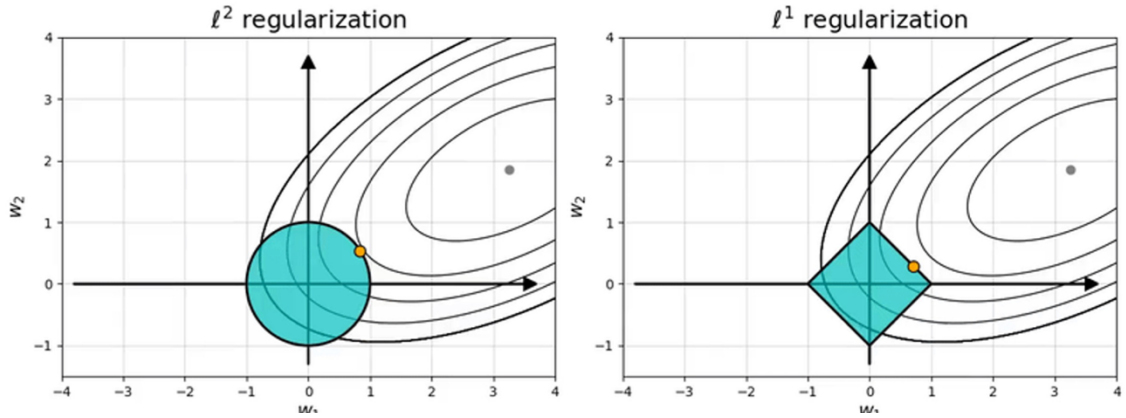
→ Influence of lambda

- Higher λ →
  - Increase regularization.
  - Likely decrease variance.
  - Likely increase bias.
  - Likely reduce error in test set
  - Likely increase error in train set
  - Reduce overfitting.
  - Reduce weights (L2) or reduce number of features (L1)



→ Type of regularization

Type	**L2 Regularization**	**L1 Regularization**
Also called	*Ridge Regression*	*Lasso*
Shape of constraint	Circle around origin.	Rombo around origin.
Find the thetas that minimize	$\frac{1}{n}\sum_{i=1}^{n}(y_i-(\theta_0+\theta_1 x_{i,1}+\ldots+\theta_p x_{i,p}))^2+\lambda\sum_{j=1}^{p}\theta_j^2$ (Regularization parameter) *First part of equation relates to error.*	$\frac{1}{n}\sum_{i=1}^{n}(y_i-(\theta_0+\theta_1 x_{i,1}+\ldots+\theta_p x_{i,p}))^2+\lambda\sum_{j=1}^{p}\lvert\theta_j\rvert$ *First part of equation relates to error.*
Intuition	Bigger lambda: smaller radius of ball, as it makes it more regularized.	For *very big* lambda → all weights go eventually to 0.
Outliers?	*It's not robust to outliers.* The squared terms will blow up the differences in the error of the outliers. The regularization would then attempt to fix this by penalizing the weights.	*Robust to outliers.*
Before using it	Standardize the predictors using the formula: $\tilde{x}_{ij}=\frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)^2}}$	
What does it do?	Combats overfitting by forcing weights to be small, but not making them exactly 0. Which means that less significant features still have some influence over the final model.	It's a form of feature selection, because when we assign a feature with a 0 weight, we're multiplying the feature values by 0 which returns 0, deleting the significance of that feature.
Sparse?	L2 regularization returns a non-sparse solution since the weights will be non-zero.	If the input features of our model have weights closer to 0, our L1 norm would be sparse (= many weights are 0).
Example	When predicting the value of a house, different input features won't have the same influence on the price. It's likely that the neighborhood or |rooms| have a higher influence on the price than the number of fireplaces. **L1 regularization** technique assigns the fireplaces feature with a zero weight, because it doesn't have a significant effect on the price. We can expect the neighborhood and the number rooms to be assigned non-zero weights.	



10.Summary of regression models

In all we are predicting real numbers Y

Standardization isn't needed in OLS, but it is in L2 and L1

Name	Model	Loss	Reg.	Objective Function to be minimized	Solution
OLS	$\hat{\mathbb{Y}} = \mathbb{X}\theta$	Squared loss	None	$\frac{1}{n}\|\|\mathbb{Y} - \mathbb{X}\theta\|\|_2^2$	$\hat{\theta}_{\text{OLS}} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{Y}$
Ridge Regression	$\hat{\mathbb{Y}} = \mathbb{X}\theta$	Squared loss	L2	$\frac{1}{n}\|\|\mathbb{Y} - \mathbb{X}\theta\|\|_2^2 + \lambda\|\|\theta^*\|\|_2^2$ ($\propto$)	$\hat{\theta}_{\text{Ridge}} = (\mathbb{X}^T\mathbb{X} + n\lambda I)^{-1}\mathbb{X}^T\mathbb{Y}$
LASSO	$\hat{\mathbb{Y}} = \mathbb{X}\theta$	Squared loss	L1	$\frac{1}{n}\|\|\mathbb{Y} - \mathbb{X}\theta\|\|_2^2 + \lambda\|\|\theta^*\|\|_1$ ($\propto$)	**No closed form**

Lecture 4: Classification (1)

1. Difference Regression and Classification

- Regression: Continuous output
  In the case when we want a binary classification, linear regression might do a good job, however, the y value might be out of label range (ex: 0.2). Also, it is more sensitive to outliers.
- Classification: categorical output

	Regression ($y \in \mathbb{R}$)	Classification ($y \in \{0, 1\}$)
1. Choose a model	Linear Regression $\hat{y} = f_\theta(x) = x^T\theta$	Logistic Regression $\hat{P}_\theta(Y = 1\|x) = \sigma(x^T\theta)$
2. Choose a loss function	Squared Loss or Absolute Loss	Cross-Entropy Loss
3. Fit the model	Regularization Sklearn/Gradient descent	Regularization Sklearn/Gradient descent
4. Evaluate model performance	R^2, MSE, Residuals, etc.	Numerical assessments: • **Accuracy, precision, recall/TPR, FPR.** • Area under curve (AUC), for ROC curves. Visualizations: • **Confusion matrices.** • Precision/recall curves. • ROC curves.

2. Classification

→ Types

1. Binary: y is either 0 or 1.
2. Multiclass: y can be many classes (ex: image labeling = cat, dog, car)

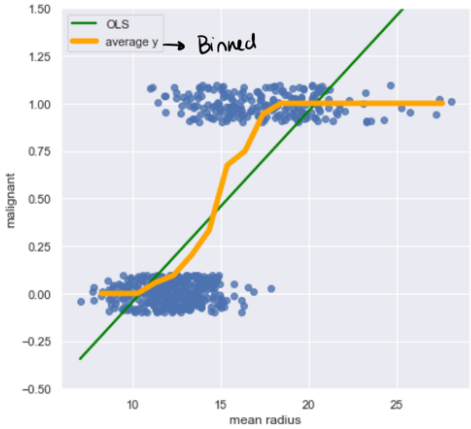
3. Logistic regression

→ What is it?

- The curve (orange) tells you the probability of the x_i being y_classification → classification.
- Is a generalized linear model.
- Is a binary classification method.

→ Model

- Aims to model the probability of an observation belonging to class 1, given some set of features.
- How to get there?
  - INITIALLY: A line will take many values between 0 and 1 and outside this range, whereas a logistic line (S shape) will tend to output more 0 and 1 values and predict much better.
  - To plot a logistic curve, we plot the average output for a bin's size (x) which corresponds to the probability of an observation belonging to class 1, given a feature:



$$P(Y=1|x) = \frac{\#\,(y == 1)\text{ in bin}}{\#\text{ datapoints in bin}}$$

- After this, the non-linear S curve is transformed into it looks linear → fit it as best possible.
  - This probability function needs to be transformed into a linear function to be able to use linear regression (maintaining its probability aspects: output between 0 and 1)
    - The logistic curve models $P(Y=1|\,x)$
    - Instead, we compute the odds of such event which is defined as the ratio of the probabilities of an event happening vs. not happening (Odds 2:1 means the event happens with probability 2/3)
    - Taking the log of the odds returns a linear combination of x and θ, which is the **assumption of Logistic Regression**

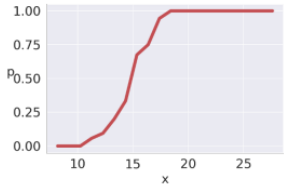
$$P(Y=1 \mid x)$$

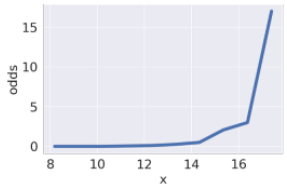
↓ odds

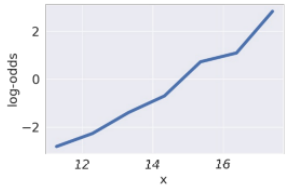
$$\frac{P(Y=1 \mid x)}{P(Y=0 \mid x)}$$

↓ Take log

$$\log\left(\frac{P(Y=1 \mid x)}{P(Y=0 \mid x)}\right)$$







$$log\left(\frac{p}{1-p}\right) = x^T\theta$$

$$p = \frac{1}{1+e^{-x^T\theta}}$$

(p ≥ 0.5 that output is 1; input feature)

Interpretation: We interpret $\frac{p(X)}{1-p(X)}$ as being the *odds* that $Y=1$. Note that $p(X)$ is a number between 0 and 1 so that the odds that $Y=1$ is a number between 0 and ∞. If $p(X)=0.5$, then odds = 1. We also have that $\text{logit}(p(X))$ is a number between $-\infty$ and ∞.

- Prediction:

$$\text{classify}(x) = \begin{cases} 1, & P(Y=1|x) \geq 0.5 \\ 0, & \text{otherwise} \end{cases}$$

→ Loss function

- We cannot use MSE loss, instead:
- Cross-Entropy Loss
  - If the true label is 1, we want the predicted probability to be as close to 1 as possible. → Penalize small predictions (around 0)
  - If the true label is 0, we want the predicted probability to be as close to 0 as possible. → Penalize large predictions (around 1)

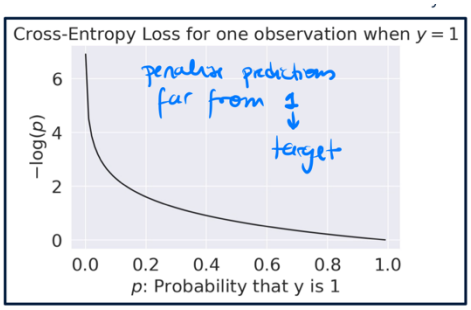
$$\text{Loss}(p) = \begin{cases} -\log(p) & \text{if } y=1 \\ -\log(1-p) & \text{if } y=0 \end{cases}$$

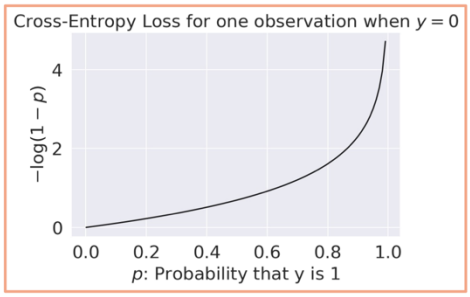
(if p ↓ then loss ↑; $0 \leq p \leq 1$)

  - For one data point, we write is like this:

$$-\left(y\log(p) + (1-y)\log(1-p)\right)$$

(− : makes loss positive; $y\log(p)$: for y = 1, only this term stays; $(1-y)\log(1-p)$: for y = 0, only this term stays)





For y = 1,
- p → 0: infinite loss
- p → 1: zero loss

For y = 0,
- p → 0: zero loss
- p → 1: infinite loss

- Average Cross-Entropy Loss/Empirical risk:

$$R(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\left(y_i \log(p_i) + (1-y_i)\log(1-p_i)\right)$$
$$= -\frac{1}{n}\sum_{i=1}^{n}\left(y_i \log(\sigma(X_i^T\theta) + (1-y_i)\log(1-\sigma(X_i^T\theta))\right)$$

(y_i is i-th response, and p_i is prob. that i-th response is 1) ($p_i = \sigma(X_i^T\theta)$ and X_i is i-th feature vector)

→ Minimizing $\hat{\theta}$

Putting it all together:

$$\underbrace{\hat{P}_\theta(Y=1|x)}_{\text{Estimated probability that given the features x, the response is 1}} = \underbrace{\frac{1}{1+e^{-x^T\theta}}}_{\text{Logistic function } \sigma(x^T\theta) \text{ at the value}}$$

at point x

The logistic regression model is most commonly written as follows:

$$\hat{P}_\theta(Y=1|x) = \sigma(x^T\theta)$$

Looks like linear regression. Now wrapped with $\sigma(\,)$!

- *You can use linear regression to find the coefficients.*
- Model: find the estimate $\hat{\theta}$ that minimizes the average cross-entropy loss:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} -\frac{1}{n}\sum_{i=1}^{n}\left(y_i \log(\sigma(X_i^T\theta) + (1-y_i)\log(1-\sigma(X_i^T\theta))\right)$$

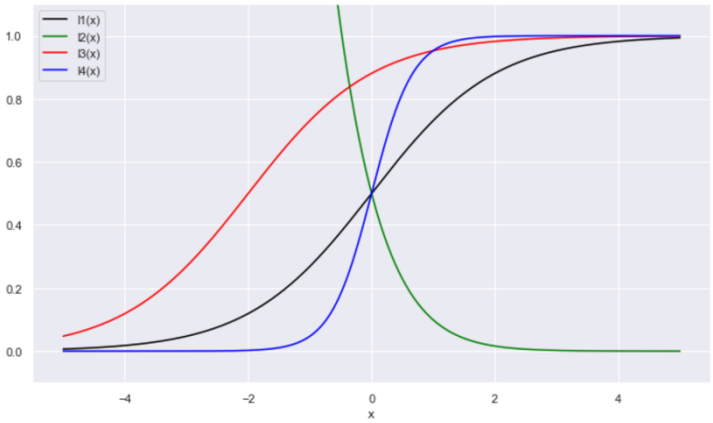
  - θ is a vector that can contain multiple θs. They represent the number of features:
    - 1 feature $\theta = 2$ (intercept + feature), 2 features $\theta = 3$.
  - θ_0: Controls where curve moves (left or right) // θ_1: Controls how steep curve

How do I interpret the coefficient?

- A one unit change in X is associated with a θ_1 change in the log-odds of $Y = 1$ or a one unit change in X is associated with an e^{θ_1} change in the odds that $Y = 1$.
- θ_0 shifts the curve right or left and θ_1 controls how steep the S-shaped curve is.
- If θ_1 is positive, then the predicted $P(Y = 1)$ goes from 0 for small values of X to 1 for large values of X and if θ_1 is negative, then $P(Y = 1)$ has the opposite association.

$\theta_0 + \theta_1 x$

```
l1 = lambda x: np.exp(0+x)/(1+np.exp(0+x))
l2 = lambda x: np.exp(0-x)/(1+np.exp(0+x))
l3 = lambda x: np.exp(2+x)/(1+np.exp(2+x))
l4 = lambda x: np.exp(0+3*x)/(1+np.exp(0+3*x))
plt.plot(x,l1(x),color='black',label='l1(x)')
plt.plot(x,l2(x),color='green',label='l2(x)')
plt.plot(x,l3(x),color='red',label='l3(x)')
plt.plot(x,l4(x),color='blue',label='l4(x)')
```



→ Regularization

- If we regularize θ_0, you need to consider that its role is to shifts.

Find θ that minimizes:

$$-\frac{1}{n}\sum_{i=1}^{n}\left(y_i \log\left(\sigma\left(X_i^T\theta\right)\right) + (1-y_i)\log\left(1-\sigma\left(X_i^T\theta\right)\right)\right) + \lambda\sum|\theta_j|$$ Lasso (L1)

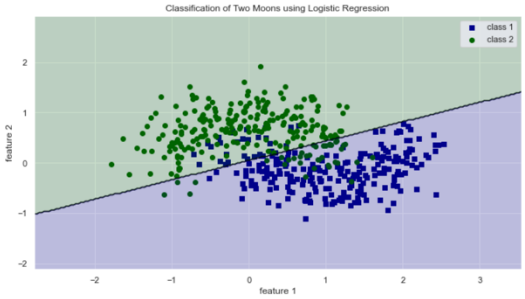
$$-\frac{1}{n}\sum_{i=1}^{n}\left(y_i \log(\sigma\left(X_i^T\theta\right)) + (1-y_i)\log(1-\sigma\left(X_i^T\theta\right))\right) + \lambda\sum\theta_j^2$$ Ridge (L2)

4. Logistic regression: for more than one variable and/or classes

- If you increase the number of samples for one of the classes, making the labels become unbalanced then you should ensure that the probability of the class (for the samples just added) is higher. This is the cause because the number of samples is way higher so you should ensure that most of the get correctly classified (reduces overall error).
- Multiple predictors:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \ldots + \theta_p X_p$$

$$p(X) = p(Y = 1|\ X) = \frac{e^{\theta_0+\theta_1 X_1+\theta_2 X_2+\ldots+\theta_p X_p}}{1 + e^{\theta_0+\theta_1 X_1+\theta_2 X_2+\ldots+\theta_p X_p}}$$



- Multiple classes:

$$P(Y = k|X) = \frac{e^{\theta_{0k}+\theta_{1k}X_1+\theta_{2k}X_2+...+\theta_{pk}X_p}}{\sum_{l=1}^{K} e^{\theta_{0l}+\theta_{1l}X_1+\theta_{2l}X_2+...+\theta_{pl}X_p}}$$

Each class gets one linear function (as above) and we weigh them against each other (that's what softmax does). Multiclass logistic regression is also referred to as multinomial regression.

When there are more than 2 categories in the response variable, then there is no guarantee that $P(Y = k) \geq 0.5$ for any one category. So any classifier based on logistic regression will instead have to select the group with the largest estimated probability. The classification boundaries are then much more difficult to determine.

5. Evaluation metrics for classifiers

Metric	Intuition	Formula	Comments
Accuracy	What proportion of points did our classifier classify correctly?	$\text{accuracy} = \frac{\#\text{ of points classified correctly}}{\#\text{ points total}}$ $\text{accuracy} = \frac{TP+TN}{n}$ $n = TP + TN + FP + FN$	-Is not meaningful when dealing with *class imbalance* in data
Precision	Of all observations that predicted to be 1, what proportion is actually 1?	$\text{precision} = \frac{TP}{TP+FP}$ penalize FP if FP↑ then precision ↓	+Good even with *class imbalance*
Recall TPR	Of all observations that were actually 1, what proportion did we predict as 1?	$\text{recall} = \frac{TP}{TP+FN}$ penalize FN $1 - FNR$	+Good even with *class imbalance*
FPR		$FPR = \frac{FP}{FP+TN}$ $FPR = 1 - recall$	

→ Class imbalance

- Occurs when there are many more instances of some classes than others.
- In such cases, standard classifiers tend to be overwhelmed by the large classes and ignore the small ones.
- Accuracy is bad for this example:

Emails = 100 / Spam = 5 / Ham = 95

Your friend ("Friend 1"):

Classify every email as **ham** (0). ← all 100 as ham

$$\text{accuracy}_1 = \frac{95}{100} = 0.95$$

High accuracy...
...but we detected none ⚠ of the spam!!!

Your other friend:

Classify every email as **spam** (1).

$$\text{accuracy}_2 = \frac{5}{100} = 0.05$$

Low ⚠ accuracy...
...but we detected all of the spam!!!

Never positive!

$$\text{precision}_1 = \frac{0}{0+0} = \text{undefined}$$

$$\text{recall}_1 = \frac{0}{0+5} = 0$$

↳ not getting any spam

$$\text{precision}_2 = \frac{5}{5+95} = 0.05$$ Many false positives!

$$\text{recall}_2 = \frac{5}{5+0} = 1.0$$ No false negatives!

→ Confusion Matrix

Actual y \ Prediction $\hat{y}$	0	1
0	True **negative** (TN)	False **positive** (FP)
1	False **negative** (FN)	True **positive** (TP)

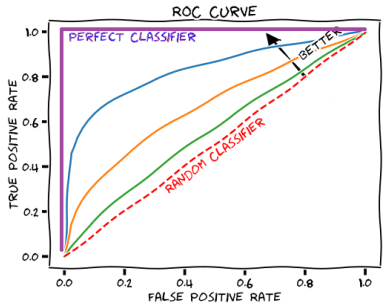
we want to maximize

False positives are "false alarms": we predicted 1, but the true class was 0.
False negatives are "failed detections": we predicted 0, but the true class was 1.

Predicted \ Actual	0	1
0	TN	FN
1	FP	TP

→ ROC Curve

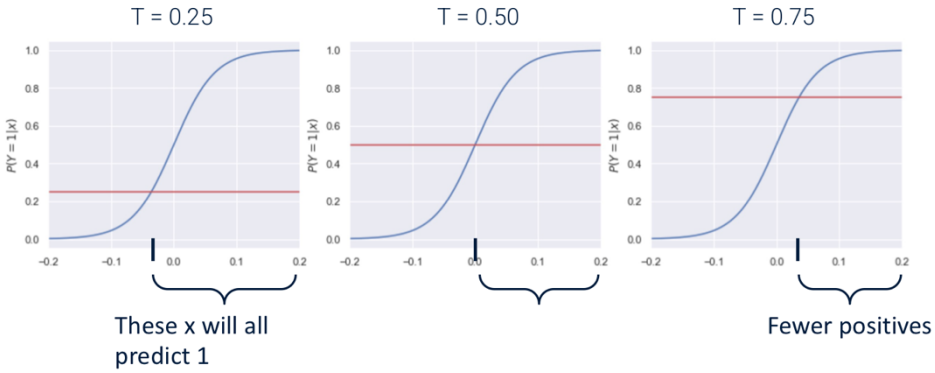
- Plots tradeoff between TPR and FPR
- We want area under curve (AUC) = 1, but is impossible.
- AUC = 0.5 is terrible → random
- Overall, the higher the better.



6. Classification threshold

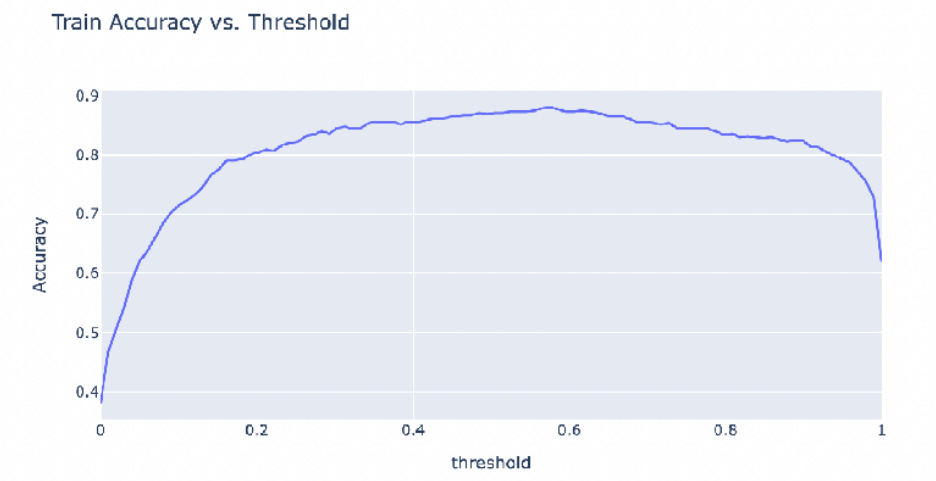
$$\hat{y} = \text{classify}(x) = \begin{cases} 1 & \hat{P}_\theta(Y=1|x) \geq T \\ 0 & \text{otherwise} \end{cases}$$

As we increase the threshold T, we "raise the standard" of how confident our classifier needs to be to predict 1 (i.e., "positive").



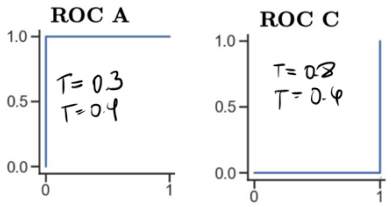
→ How to choose T?

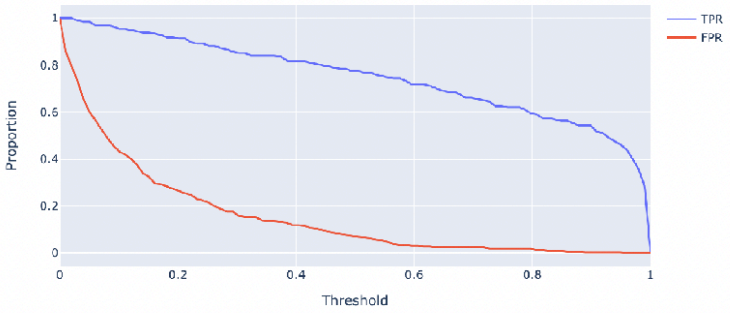
- To choose an adequate threshold you can do the model many times with different values of T, and calculate it's accuracy (or any other metric). Choose the point where accuracy is max. Ex: T= 0.57
- High T: most predictions are 0.
- Low T: most predictions are 1.



→ Relationship of TPR and FPR with T

- T = 0 , or close to 0 →everything is positive.
  - There are no TN (all of them are FP). → FPR = 1
  - We identified all true 1's as 1, so TPR = 1
- T = 1, or close to 1 → everything is negative.
  - There are no FP (all of them are TN). → FPR = 0
  - We identified zero true 1's as 1, so TPR = recall = 0
- A decreased TPR is bad.
- A decreased FPR is good.





7. KNN as classifier algorithm

- It helps us to have a curvy decision boundary.

→ Idea

Idea: To decide the class of a given point, find the k-nearest neighbors of that point, and let them "vote" on the class. That is, we assign the class to the sample that is most common among its k-nearest neighbors.

Considerations:

1. We must pick k, the number of voting neighbors (typically a small number, say k=10)
2. 'Nearest' means closest in distance, so there is some flexibility in defining the distance. Can you name me some of your favorite distance measures?
3. There are different ways to vote. For example, of the k nearest neighbors, I might give the closest ones more weight than farther ones.
4. We have to decide how to break ties in the vote.

- Tie breaker: decrease K by 1 until you have max of one class. Or take the majority class of entire dataset.

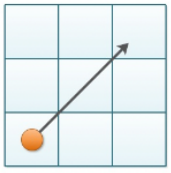
cosine_distance(u, v) = 1 - (u . v) / (||u|| ||v||)

where:

- u . v is the dot product of u and v
- ||u|| and ||v|| are the Euclidean lengths (or magnitudes) of u and v, respectively.

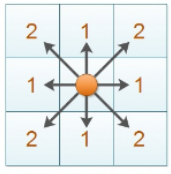
Note that the cosine distance ranges between 0 (for identical vectors) and 1 (for orthogonal vectors), with higher values indicating greater dissimilarity.

Euclidean Distance



$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

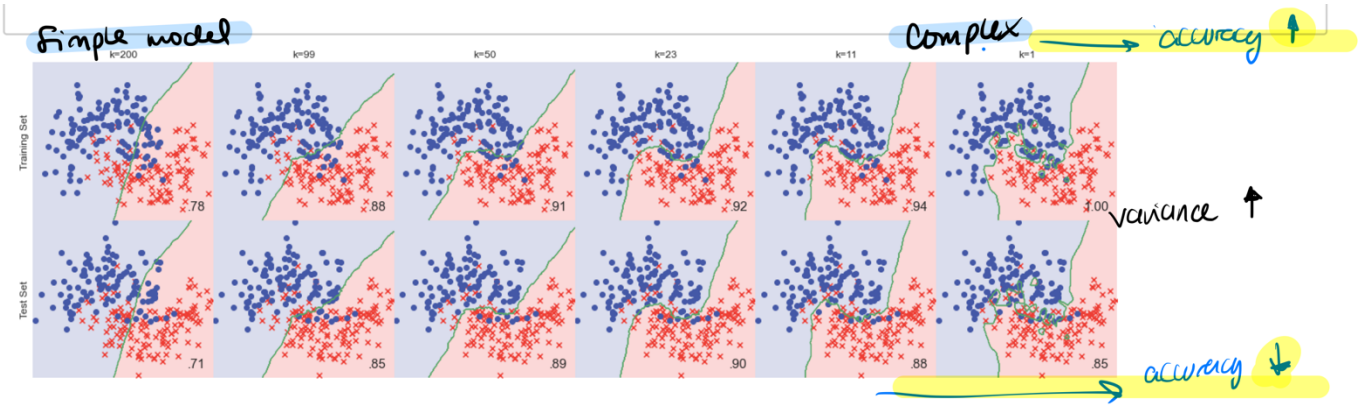
Manhattan Distance



$|x_1 - x_2| + |y_1 - y_2|$

→ Observations

K increases	K decreases
Decision boundary becomes **smoother**/straighter but curved	Decision boundary is **wiggly**
Model is **not complex**	Model is **complex**
Model has **low variance**: if the data changes, the model doesn't change much. Why? Because there are many neighbors.	Model has **high variance**
It **generalizes** to new data points: it performs well on similar data that it hasn't seen.	Since it depends on data, it **wouldn't generalize** to new data points well.
Model can **underfit** data	Model can **overfit** data



→ KNN vs Logistic Regression

- K-NN has better performance for good choices of K.
- Logistic regression is a linear classifier (decision boundary is straight) which can be bad.
- There exist extensions for logistic regression which allows it to curve.

Lecture 5: Classification (2)

1. Decision trees

→ How to build one?

- Start with an empty decision tree (undivided feature space)
- Choose the **best split** (choose the 'optimal' threshold value for splitting) for the **optimal predictor** (creates a partition of data where the **error** is minimized).
  - Choose decision that minimizes the error by the largest amount.
  - Calculate the mean error:

$$ME = \frac{1}{N}\sum_{i=1}^{N} \ell(\hat{Y}_i, Y_i),$$

($\hat{Y}_i$: predicted; Y_i: observed; $\ell(\hat{Y}_i, Y_i)$: error for sample i)

  - Error:
    - Regression: squared
    - Classification: cross-entropy or Gini impurity

- Recurse on each new node until stopping condition is met

→ Advantages

- Simple to explain and interpret (graphical display)
- Easy to handle categorical predictors (no dummy variables needed)

→ Disadvantages

- They don't have the predicative accuracy of other approaches as they tend to overfit.
  - Is easy to **overfit**.
  - How to avoid or improve performance?
    - Limit the depth of a tree.
    - Split only when we have more than N samples left → pruning by preventing splits that have less than x% of samples.
    - Use another technique that improves the performance, example: Random forests.
- Trees are non-robust (sensitive to small changes in the data)
  - They have high **variance**: not adapting well to unseen data.

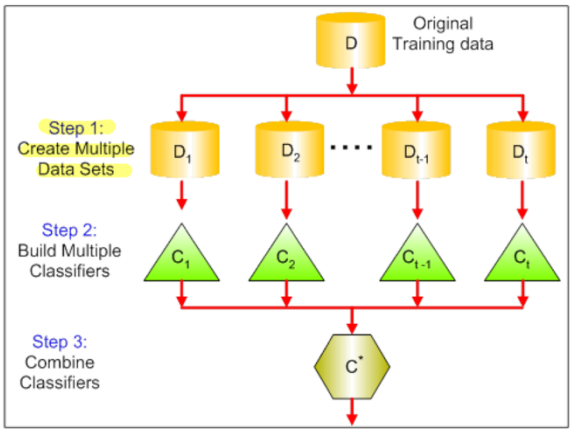
→ Ensemble Methods based on Decision Trees

- They have higher accuracy than trees but take longer to train.
- They use multiple algorithms at the same time and then come to a consensus to predict a label/output.

1. Bagging (Bootstrap Aggregating)

- Relies on randomness.
- Idea: generate several models (decision trees) based on subsets (sampling with replacement) of the data and take the average to get to a correct answer (ex: majority voting).
- Pros:
  - High expressiveness
  - Good to reduce variance: assuming that a sufficiently large number of trees were generated.
- Cons:
  - The model is not easy to interpretate.

- In practice, the trees tend to be highly correlated (ex: in titanic data, all trees will start with sex).



Training Data

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease
Yes	Yes	205	Yes
No	Yes	180	Yes
Yes	No	210	Yes
Yes	Yes	167	Yes
No	Yes	156	No
No	Yes	125	No
Yes	No	168	No
Yes	Yes	172	No

Do **N** times

1. Bootstrap the data (i.e. sample)
2. Train a new decision tree T_i

We have $\{T_1, T_2, T_3, \ldots, T_N\}$

sampling

Testing Data

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease
No	Yes	158	?

Take a majority vote from $\{T_1, T_2, T_3, \ldots, T_N\}$

2. Boosting

- This doesn't rely on randomness but rather it learns from its previous predictions/mistakes.
- Is a technique (like Regularization) that can be applied to many statistical learning methods.
- Data is not split.
- It uses **<u>weak models/learners</u>** (decision trees) to prevent overfitting since the complexity of the overall learner increases at each step and starting with weak learners implies that the final classifier will be less likely to ***overfit.***
- Parameters:
  - Number of trees: *q.*
  - Learning parameter: how much to consider at each new iteration. Typical values = 0.01 and 0.001.
  - Number of splits (d) in each tree
    - If d = 1 → Tree is a stump → Model: additive
- Idea:

We have $\{T_1, T_2, T_3, \ldots, T_N\}$ (weak tree)

$$T = \sum_h \lambda_h T_H$$

(λ_h: learning rate)

Each T_h is:

- a "weak"/simple decision tree (= "stump")
- built after the previous tree
- tries to learn the shortcomings (as given by errors or residuals) from the previous tree's predictions

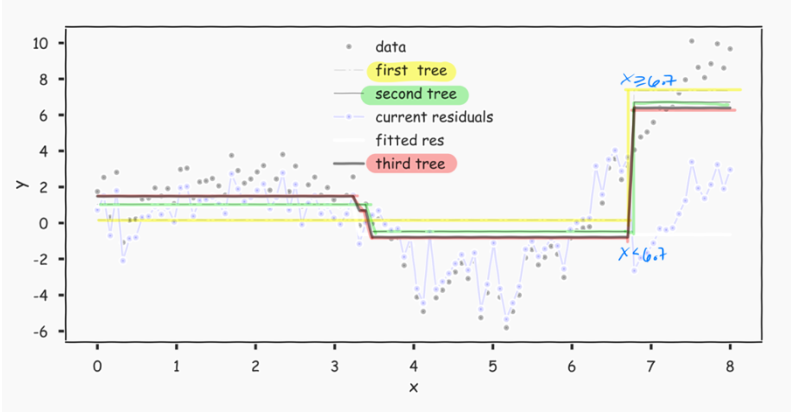
o increase weight of misclassified data points

11

If iteration 1 has a lower error by not applying any decision boundary, then don't create it and just classify the entire data set as +.

- Tasks:
  - Regression tasks: Gradient Boosting

- It models the residuals



- Classification: AdaBoost
  - It adds higher weights to previous misclassifications, so our dataset is constantly being changed.
  - To correctly classified instances, it also updates the weights, but all by the same value.

  - Example of weight update:
    - Since x_1 is correctly classified, the error rate = 1/3.
    - As the error rate < ½, the weights of correctly classified points will decrease and the weights of misclassified points will increase.
    - Hence x_2 is correctly classified.

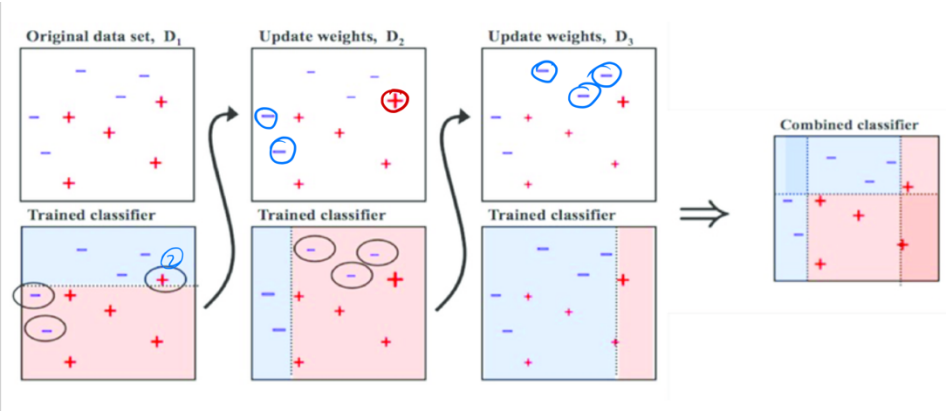
	True Label	Classifier Prediction	Initial Weight	Updated Weight
X_1	−1	−1	1/3	?
X_2	?	+1	1/3	$\sqrt{2}/3$
X_3	?	?	1/3	$\sqrt{2}/6$

  - Algorithm:

1. Choose an initial distribution over the training data, $w_n = 1/N$. (initial weights)
2. At the i^{th} step, fit a simple classifier $T^{(i)}$ on weighted training data $\{(x_1, w_1 y_1), \ldots, (x_N, w_N y_N)\}$
3. Update the weights so that misclassified points get higher values
4. Update $T: T \leftarrow T + \lambda^{(i)} T^{(i)}$ (every iteration is amended by some factor)

where λ is the learning rate.

14



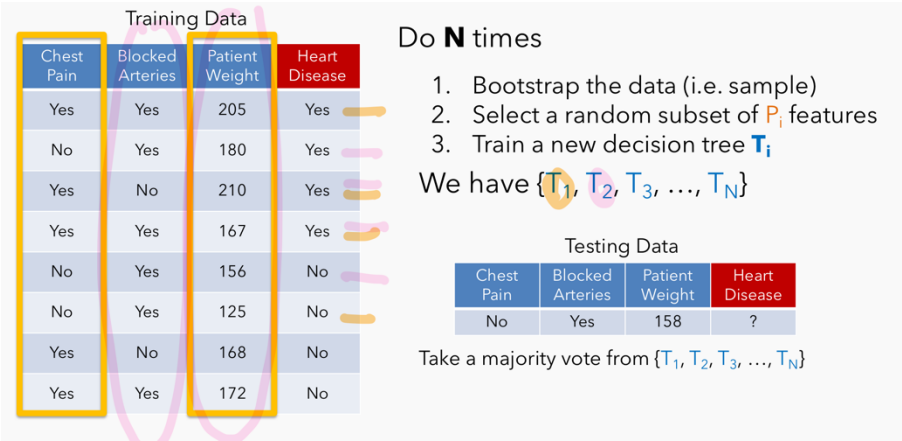
- Cons:
  - It can overfit if the number of trees is too large.

2. Random Forests (RF)

- Better than decision tree as it educed the variance of the model.

→ Idea

- Is a form of Bagging, with the difference that we build each tree by splitting on a random subset of predictors at each split. → Each tree is random.
- Process is done using $m = \sqrt{p}$ predictors, where p = total number of predictors.
- Then we combine the random trees together by averaging their predictions.
- Hyper-parameters to tune:
  - *m*
  - *total number of trees*
  - *Minimum leaf node size / Max depth / Minimum samples split*



→ Advantages

- Faster to train than decision trees because we are working only on a subset of features in this model.
- They provide the highest accuracy.
- We can obtain the feature importance.
- Increasing the number of trees, generally doesn't increase the risk of overfitting.
  - However, if the number of trees is too large, then the trees might become correlated and that increases the variance.

→ Disadvantages

- When the number of predictors is large, but the number of relevant predictors is small, RF perform poorly.
  - In each split the chances of selecting a relevant predictor are low.

Lecture 6: Timeseries

1. Introduction to temporal data analysis

- Timeseries: series of data points indexed by time.
- Temporal data analysis: study of timeseries.

→ Problem

- If correlation is present, then our approaches are not correct (they assume independence).

→ Solution

- We should check timeseries looking for:
  - Trends: increasing and decreasing value in the series
  - Seasonality: repeating short-term cycle in series ex: correlation on December. Sometimes include: time of day, daily, weekly, monthy, yearly.
- And remove them to generate **stationarity.**

$$x_t = m_t + s_t + r_t$$
$$= \text{trend} + \text{seasonal effect} + \text{stationary remainder}$$

→ Weak stationarity

- We want this on data.
- Timeseries is, if its statistical properties are all constant over time:
  - All datapoints have identical expected mean.
  - All datapoints have identical variance.
  - The autocovariance depends only on lag h.
- Stationary data have the following properties:
  - No trend
  - Variations around its mean have constant amplitude.
  - It wiggles in a consistent fashion.

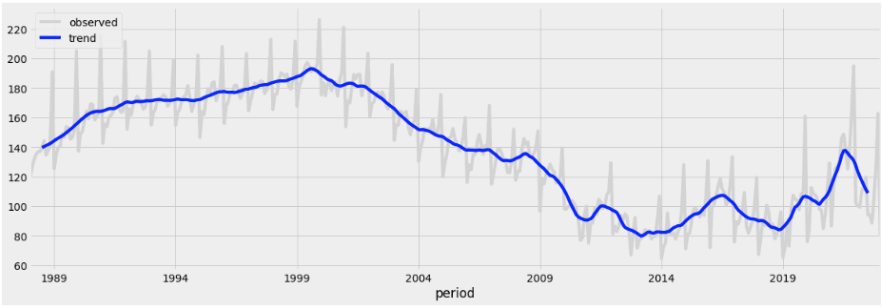
2. Visualizing time series

	period	value
0	1988-01-01	119.8
1	1988-02-01	125.3
2	1988-03-01	131.4
3	1988-04-01	134.9
4	1988-05-01	136.8
5	1988-06-01	136.9
6	1988-07-01	139.6
7	1988-08-01	144.3
8	1988-09-01	134.6
9	1988-10-01	137.8

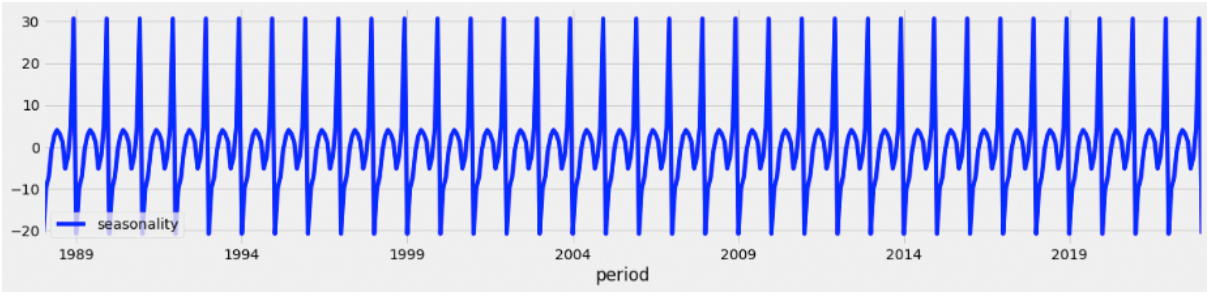
From the plot we can conclude that the time series has a **trend** and yearly **seasonality** components. We will try to identify them using models:

→ Computing trend

- Fit a linear model to the data.

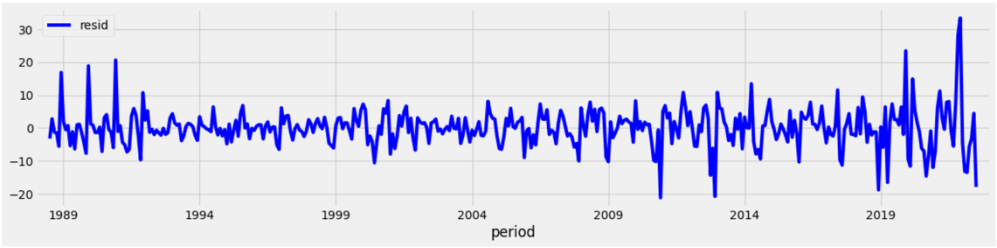


→ Computing seasonality



3. Stationarization

- Remove trend and seasonality.
- What is left: Residual, should meet the properties of stationarity.



→ they are randomly around 0

- Other techniques to consider:

A. Differencing

- You want to difference because it helps achieve stationarity, as it removes non-stationarity beyond just trend (ex: also non stationarity variance).
- How many times to apply it? Observe the series after each iteration and repeat until it appears stationary.
- Use if we know that there is a trend, but we cannot accurately estimate it, then one way to correct for the trend stationarity would be to try differencing (FD).

We assume a series with an additive trend, but no seasonal variation. We can then write as : $x_t = m_t + r_t$.
If we perfom differencing and assume a slowly-varying trend (such that $m_t \simeq m_{t+1}$) we obtain:

$$y_t = x_t - x_{t-1} \simeq r_t - r_{t-1}$$

and then y_t (aka the difference) is stationary.

- Seasonal growth can be corrected using log transformation or seasonal differences.

B. Box-Cox Transformation

- Helps suppress variance by transforming non-normal dependent variables into a normal shape.
- If after applying it, data is not stationary, apply C.

The Box-Cox transformation is a family of power transformations indexed by a parameter λ. Whenever you use it the parameter needs to be estimated from the data. The formal definition for a variable, say x_t is as follows:

For a time series x_t, the *q-day moving average at time t*, denoted MA_t^q, is the average of x_t over the past q days,

$$MA_t^q = \frac{1}{q}\sum_{i=0}^{q-1} x_{t-i}$$

shape of the time series. But for all other values of λ, the time series will change shape. See more on this

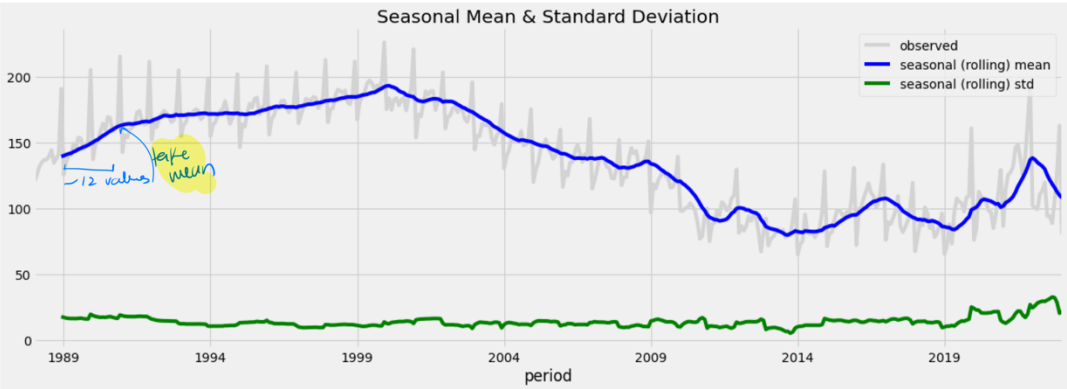
C. Smoothing / Moving averages

- Fast moving: smaller window: more variation + follows the time series
- Slow moving: bigger window: you see less details about the micro-changes in the time series.

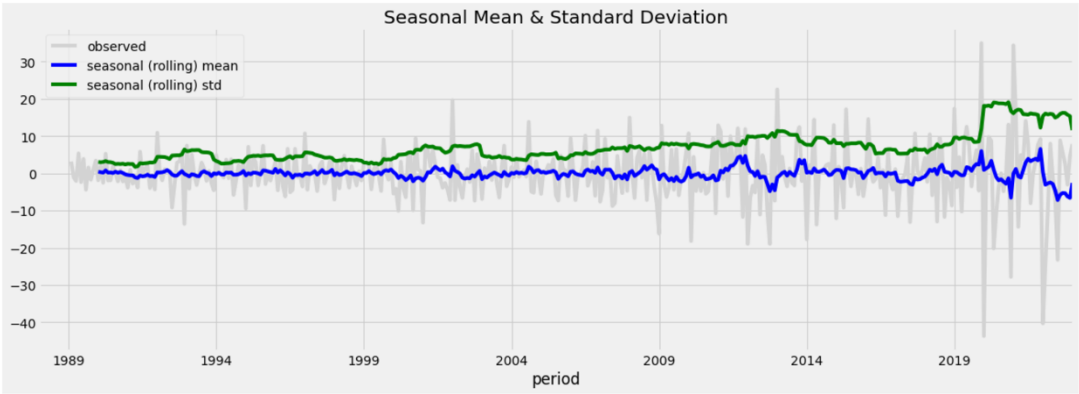
4. Stationarity of time series checks

A. Visual check

- You want to see that mean and std don't change.



No stationary



Stationary

B. Statistical tests

	Augmented Dickey-Fuller (ADF)	Kwiatkowsky-Phillips-Schmidt-Shin (KPSS)
Idea	If data has unit root (drastic change) then is not stationary. Test is used to test for a unit root.	Checks if timeseries is stationary around the mean (or the trend). If variance = 0 then random process is a constant so then is stationary.
H_0	There is a unit root	Time series is level stationary or trend stationary (stationary around a deterministic trend)
H_A	There is no unit root	There is a unit root
Results	ADF> critical value → reject H_0 So the unit root is not present and the time series is stationary.	KPSS > critical value → reject H_0 So the time series is non-stationary.
Disadvantage		*High rate of type 1 error: tends to reject H_0 too oftern.* *To deal with this, combine ADF and KPSS as they are complementary.*

C. Correlogram

- Correlogram is a plot where the autocorrelations (r_k) are plotted against k.
- $r_k = ACF$
  - $r_0 = 1$
  - Bands = 95% significance level. We want r_k to be within it.
    - When data is obtained from pure random, the probability of a r_k laying outside is 0.05.
    - Values outside bands: gives evidence against pure randomness.
    - The probability of getting at least one r_k outside the bands increases with the number of coefficients plotted.
  - ***1 or 2 r_k outside the bands can be ignored, but 3 or more mean no stationary.***
  - If you see that the ACF 'dies out' slowly, it means that there's a strong trend.

- Idea: for a purely random series, the given data cannot help predicting a new datapoint. Therefore, the correlation between x and x' is close to cero.

  The quantity r_1 is called the **Sample Autocorrelation Coefficient** of x series at lag one. We take lag one because this correlation is between x_t and x_{t+1}.

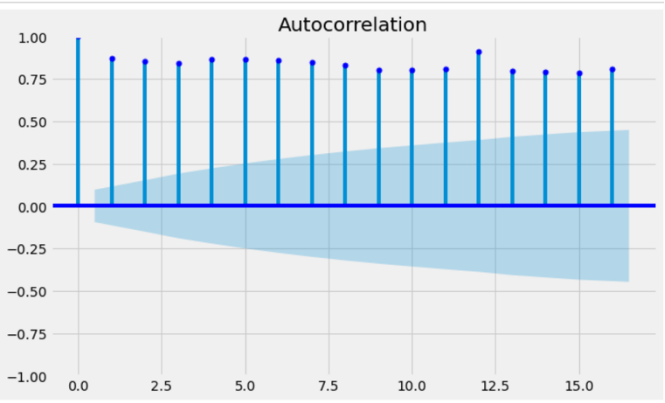
  When the series stems from a purely random process, r_1 is close to zero, particularly when n is large. One can similarly consider Sample Autocorrelations at other lags:

$$r_k = \frac{\sum_{t=1}^{n-k}(x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^{n}(x_t - \bar{x})^2}$$

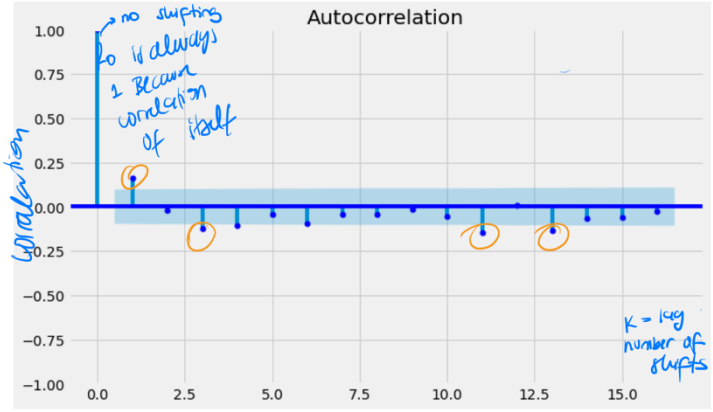
  When values $x_1, \ldots, x_n$ are obtained from a purely random process, then $r_1, r_2, \ldots$ are expected to be independently distributed according to $\mathcal{N}(0, 1/n)$.

- Sample ACF the empirical estimate of ACF.
- Plots:

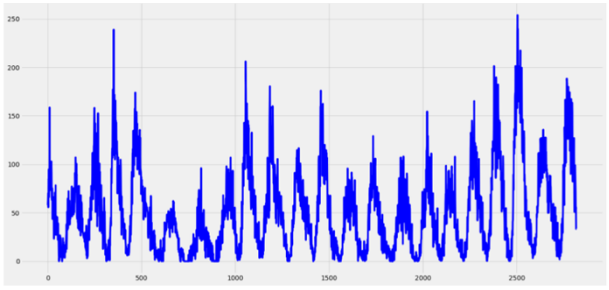
Original data:



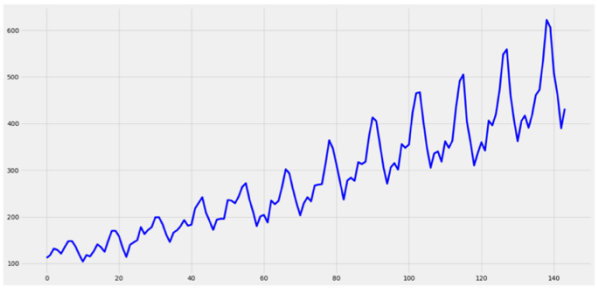
Residual data:



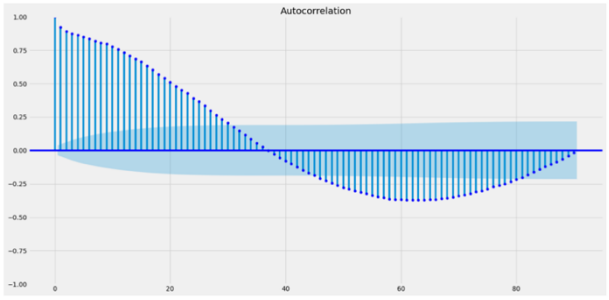
Non-stationary data:

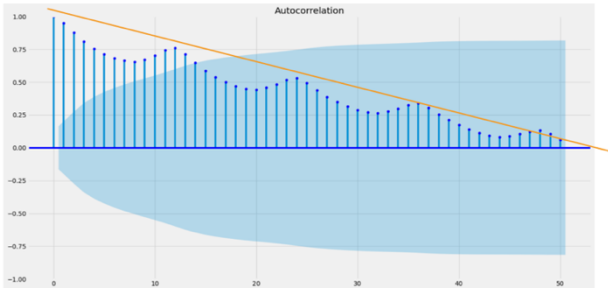


<Figure size 1440x720 with 0 Axes>



<Figure size 1440x720 with 0 Axes>



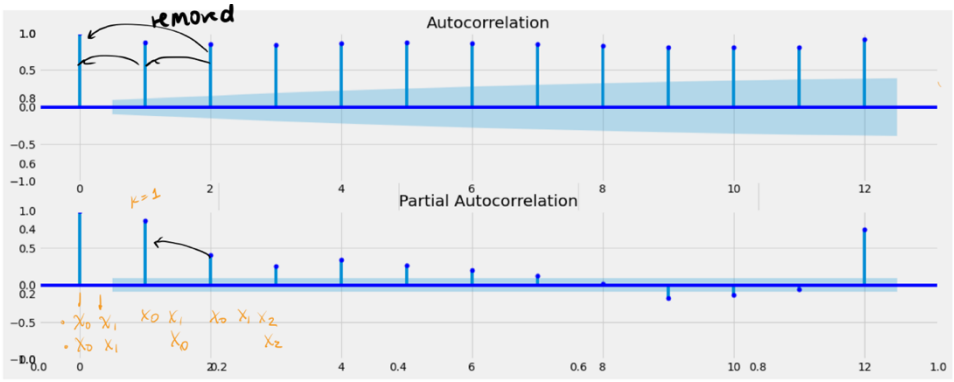


5. Autoregressive models as a baseline

- To predict future values, we look into building a regression model, but we should make sure that points in correlogram are not correlated, and if they are we should 'partial out' their effects on each other to get the unique contribution of each time-lag.

→ Partial Autocorrelation Plot (PACF) and Autocorrelation Plot (ACF)

- shows correlation between x and x' after removing the linear relationship of all operations between x and x'.
- Lag = 0 is always 1 for PACF and ACF because it's the series correlated with itself.
- Lag = 1 is the correlation when there is no partial effect yet, so ACF and PACF are the same.



ACF

PACF

We can fit an autoregression model and decide using a criterion (e.g. AIC) how many lagged values should be included in the model.

Practically, I fit a linear regression model where my independent variables are the time-lagged values of my timeseries.

6. Autoregressive Integrated Moving Average (ARIMA) Model

- d is chosen iteratively.

ARIMA models put together many of the concepts we already discussed (autoregression, differencing, smoothing) and are denoted with the notation $ARIMA(p, d, q)$. These parameters account for seasonality, trend, and noise in datasets:

- p - the number of lag observations to include in the model, or lag order. (AR)
- d - the number of times that the raw observations are differenced, or the degree of differencing. (I)
- q - the size of the moving average window, also called the order of moving average.(MA)

A linear regression model is constructed including the specified number and type of terms, and the data is prepared by a degree of differencing in order to make it stationary, i.e. to remove trend and seasonal structures that negatively affect the regression model. A value of 0 for a parameter indicates to not use that element of the model.

ARIMA is still a linear model!

$$y_t = c + \phi_1 y_{d,t-1} + \phi_p y_{d,t-p} + \ldots + \theta_1 \epsilon_{t-1} + \theta_q \epsilon_{t-q}$$

Some example models:

ARIMA(0,0,0) is simply predicting the mean of the overall time series, i.e., no structure.

ARIMA(0,1,0) works with differences, not raw values, and predicts the next value without any autoregression or smoothing.

ARIMA(1,0,0) and ARIMA(24,0,0) are AR models that take into account one previous terms and 24 previous terms respectively.

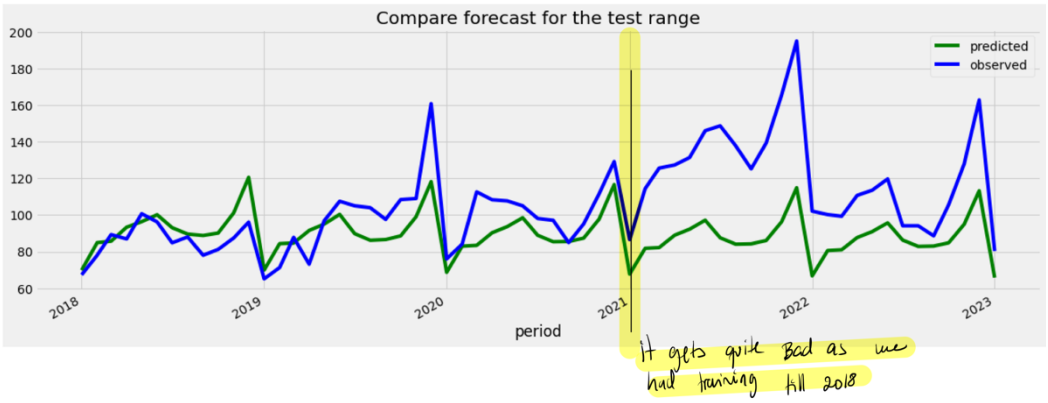
→ Determine AR/MA parameters

- After fitting ARIMA model, we need to determine whether **AR** (autoregressive) or **MA** (moving average) terms are needed to correct any autocorrelation that remains in the series.
- By looking into ACF and PACF we can choose the values:

- AR(p) model:
  ACF "dies off" gradually
  PACF "dies off" after p lags
- MA(q) model:
  ACF "dies off" after q lags
  PACF "dies off" gradually
- ARIMA(p,0,q) model:
  Both ACF/PACF "die off" gradually

To find p all the statements need to hold in AR model.

7. Results



Lecture 7: Storytelling & Effective communication

1. Fundamental questions

- What is the goal?
  - Predict future data?
  - Explain and understand a phenomenon?
  - Test a hypothesis?
  - Compare two groups?
  - Dimension reduction?
  - Build a good recommendation system?
  - Decide on a course of action or a policy?
- Who cares?

2. IMAC

- Helps us organize what we do.

I: inferential goal (scientific question of interest)
M: model (all models are wrong, some are useful)
A: algorithms
C: conclusions and checking

The C is crucial:
What did we learn? Was the model useful, and how well does it fit? How do we know whether the method is working?
Do we need to iterate and improve the model?
What are the limitations and future directions?

3. Ethical considerations

- How did you get the data (legally...)?
- Did you check the source?
- Are there considerations about the analysis you made?
- Did you have to make decisions that affect the analysis?
- What are the limitations of your analysis and your conclusions?
- Who are the stakeholders of your project?
- Who can benefit from this project, who can be harmed, who is excluded, etc.

↗ who should be involved & concerned about your analysis → who can we benefit + harm

Lecture 8: Dimensionality reduction

1. Dimensionality reduction

→ What is it?

- Process of reducing number of features in a dataset.
- Is part of exploratory analysis: we are not predicting anything but rather exploring data.
- Is unsupervised learning (goal: discover interesting things about data).
  - Supervised goal: learn how to predict response.
- **Goal**: *understand structure of data and reduce the number of features.*
- It could be applied as:
  - Feature selection: pick subset of available features. If some of your variables are one-hot-encoded, it would make sense to either turn them all on or off.
    - Forward Selection: start with one feature.
      - Find the best model with only one and then move to models with 2 (keeping the one best fixed), etc.
    - Backward Selection: start with a model with all, and reduce them one by one.
  - Feature extraction/transformation: data is transformed from a high dimensional space to lower. Most common method: PCA.

→ Why do it?

- Computational: compress data saves time/space.
- Remove redundancies: correlated features should be removed.
- Visualization: help understand structure of the data.
- Helps to detect anomalies/outliers.

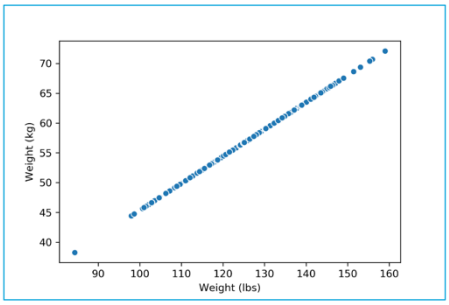
→ How does it work?

- Given a dataset, put it in matrix structure, where:
  - Rows = items = samples = **N**
  - Columns = features
  - **Rank = Dimensionality = d =** number of linearly independent columns.
    - One outlier is enough to change the rank of matrix, as you cannot obtain one column from other.
    - If columns are width, height, area, and perimeter (*2w+2h*). Rank = 3 because area is not linear combination.
    - *Dimensionality reduction can remove this outlier.*

- If one dimension/feature can be derived from other then we don't need it to describe the data, ex: Rank = 1

Weight (lbs)	Weight (kg)
113.0	51.3
136.5	61.9
153.0	69.4

one can be derived from other



2. Matrix Factorization or Matrix decomposition

- Is the opposite of matrix multiplication:

$$(A, B)\,(B, C) = (A, C)$$

- It decomposes a matrix into 2 separate ones.
  - There exist infinitely many such decompositions.
  - There exist many sizes of matrix that would result in same result (dimensions).

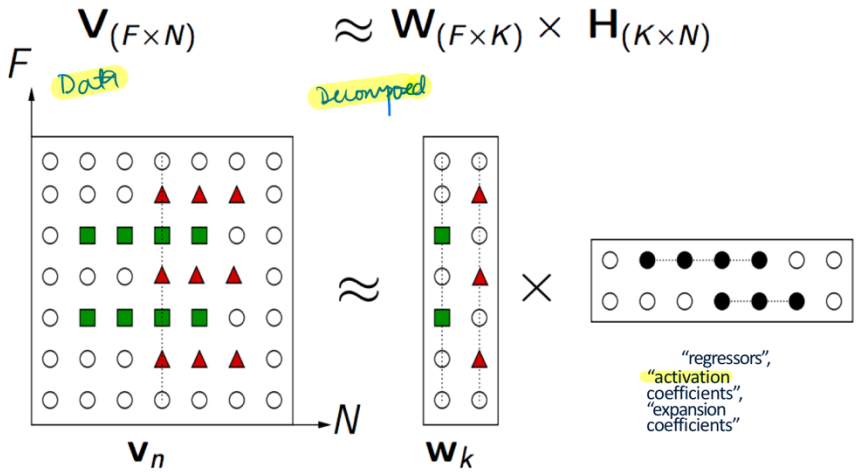
3. Singular Value Decomposition (SVD)

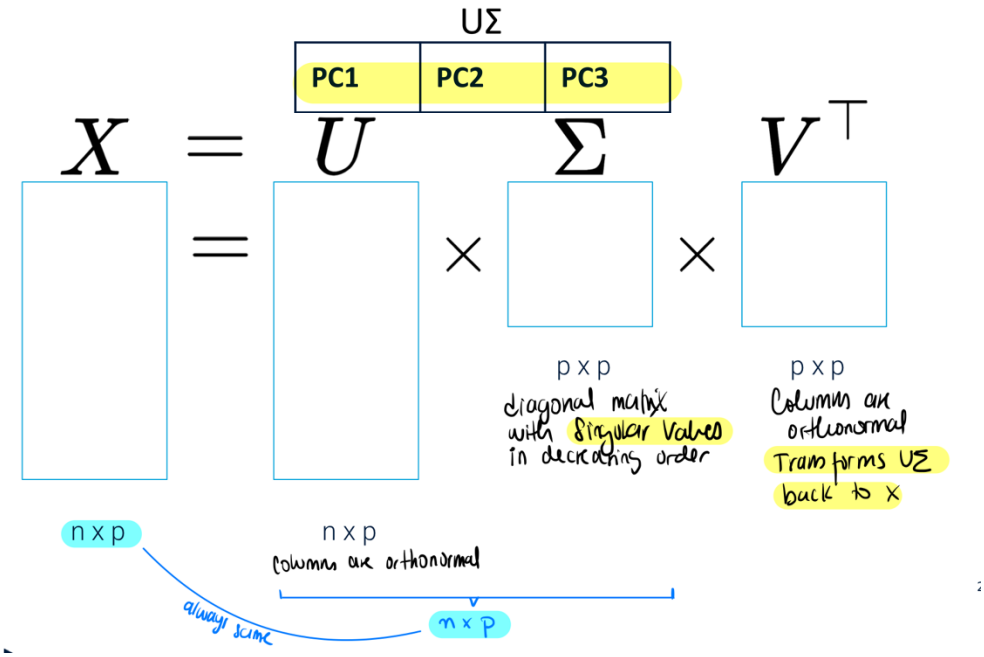
- Technique to automatically decompose a matrix into 2 matrices.
- It works by finding W or UΣ, such that it minimizes:

$$min_W \|H - WV\|^2$$

V^T, UΣ, original data = X

→ Components





- If column of UΣ has only 0s, you can drop it, and the row from V^T. This won't affect result X.
  - This is related to unnecessary columns because: rank < features.
  - To have a perfect reconstruction of X:

$$|\text{row}|_{\text{matrix 1}} = |\text{column}|_{\text{matrix 2}} \geq \text{rank}(X)$$

If this doesn't hold, you get approx. values.

- **Orthonormal set if:**
  - All of the vectors are unit vectors, i.e. have length 1.
  - All of the vectors are orthogonal → dot product of row with itself = 1, and with another = 0.

→ Singular values (Σ)

- Always non-negative.
- Beyond rank are set to 0.
- They represent how valuable the principal component of that column will be.
  - ith singular component: how much of the variance is captured by the ith principal component.

width	length	area	perimeter
2.97	1.35	24.78	8.64
-3.03	-0.65	-15.22	-7.36
-4.03	-1.65	-20.22	-11.36
...			
-3.03	1.35	-11.22	-3.36

calculate it as usual variance

Variance for each feature:

[7.7 5.3 338.7 50.8]

Total variance: 402.56

= Trace

how important is PC1 in reconstructing data

$$\begin{bmatrix} 197.39 & 0 & 0 & 0 \\ 0 & 27.43 & 0 & 0 \\ 0 & 0 & 23.26 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Singular value

= we don't need 4th dimension

35

N

Variance captured by 1st PC: $197.39^2/100 = 389.63$

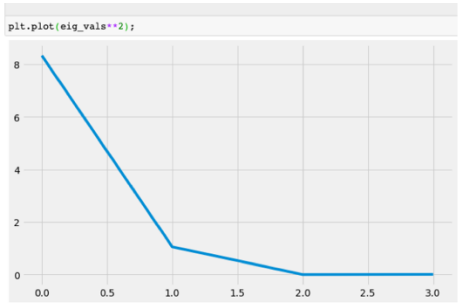
Variance captured by 2nd PC: $27.43^2/100 = 7.52$

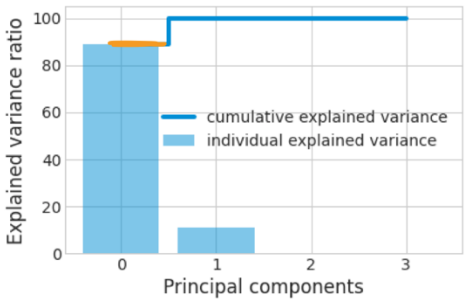
Variance captured by 3rd PC: $23.26^2 / 100 = 5.41$

$\Sigma = 402.56$

N: data points or samples

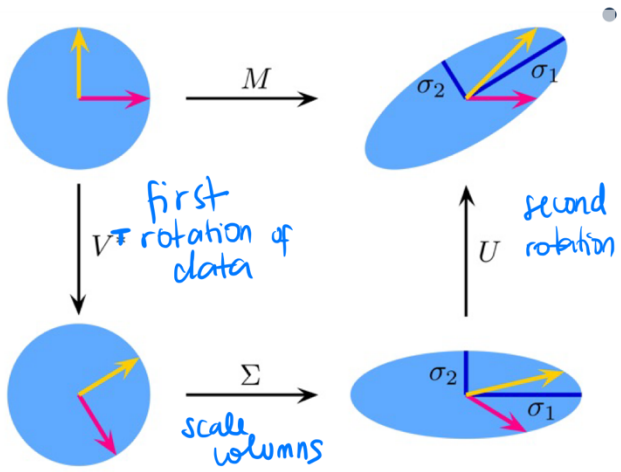
→ Scree plots and Variance Fraction Arrays





if I keep 1 PC I explain 80% of data

→ SVD: geometric interpretation



$$M = U \cdot \Sigma \cdot V^T$$

→ Making predictions from Decomposed Matrices

- Keep in mind: If we wanted to put this kind of system into production, we first would have to create a training and validation set and optimize the number of latent features (k) by minimizing e.g. the Root Mean Square Error. Intuitively, the Root Mean Square Error will decrease on the training set as increases (because we are approximating the original ratings matrix with a higher rank matrix).
- Matrix can also be negative but we prefer positive because then value is more interpretable.
- Example:

- Given a NXM matrix R with some entries unknown
  - N columns represent items
  - M rows represents users
  - Entry R_{ij} represents the i-th user's rating of j-th item
- We are interested in the unknown entries' possible values & the hidden dimensions
- Here I will attempt:

$$R_{N\times M} \approx W_{N\times 2} \times H_{2\times M}$$

entries

	Passengers	War of the Worlds	Bride Wars	The Matrix
John	2		3	
Arya		5		4
Sansa	4			5
Robb	4	5		

- ***L1 and L2 come from UΣ***
  - Problem is that there are many missing entries but NMF handles this problem!
    note: the default NMF from sklearn does not handle missing values

	L1	L2
John	0.2	0
Arya	0	0.5
Sansa	0.5	0
Robb	0	0.5

	Passengers	War of the Worlds	Bride Wars	The Matrix Reloaded
L1	6.3	0	1.1	8
L2	3.9	10.7	0	3.3

  - Estimate unknown ratings as inner products of latent factors
    - Rating of Arya for Passengers:
      6.3*0 + 3.9*0.5 = 1.95

John	2		3	
Arya		5		4
Sansa	4			5
Robb	4	5		

≈

	Passengers	War of the Worlds	Bride Wars	The Matr
John	1.3	0	0.2	1.6
Arya	2	5.4	0	1.7
Sansa	3.2	0	0.6	4
Robb	2	5.4	0	1.7

Maastricht University

4. Principal Component Analysis (PCA)

- Before looking into it, data must be centered (remove mean) and standardized (mean = 0, std = 1).
- Process of transforming data into new coordinate system such that the greatest variance occurs along the first dimension, second along second dimension, and so on.
- Dimensions are independent to each other (not correlated).
  - If data has only 2 dimensions/features → PC1 and PC2 will capture 100% variance.

→ Goal

- Identify patterns in data.

- Detect the correlation between variables: if a strong correlation exists, reducing the dimensionality makes sense. → Removes correlated dimensions.
- Finding the direction of maximum variance and projecting them onto a smaller dimensional subspace while retaining most of the information.

→ How's does it work?

1. Center (each column has mean 0) or Standardize (center and each column has std = 1) the data.
   - As PCA maximizes the variance along the axes, it makes sense to standardize the data, especially if it was measured on different scales.
2. $S = X'X$
   X': $Matrix_{variance}$
   X: $Matrix_{covariance\ =\ correlation}$

width	length	area	perimeter

```
[[  7.76676768  -0.17121212  35.27616162  15.19111111]
 [ -0.17121212   5.40151515  26.57272727  10.46060606]
 [ 35.27616162  26.57272727 342.15313131 123.69777778]
 [ 15.19111111  10.46060606 123.69777778  51.30343434]]
```

Variance-covariance Matrix
everything else

   - **Trace** of S = sum of the diagonals of the variance-covariance matrix
     Represents the total variance in the data.

3. Obtain the **eigenvectors** and **eigenvalues** *(not the same as singular value)* from S
   - Find **eigenvalues of S (***magnitude***)**: represent the variances of the coordinates on each principal component axis.
   - **Sum of all eigenvalues** = trace of S
   - Each **eigenvector (***direction***)** consists of values which represent contribution of each variable to the PC axis → 'loading'
     - Eigenvectors are independent (uncorrelated) between them.
     - Eigenvalues of eigenvectors show how much of the total variance, they explain.

Eigenvectors (**w**)

PC1

Width	[0.43759561	-0.65472309	0.5466161	0.28470791]
Height	[0.37560774	0.75521251	0.45584801	0.28421027]
Area	[0.57100602	0.02481468	0.	-0.82057075]
Perimeter	[0.58427822	-0.01938982	-0.70243394	0.40599208]
Eigenvalue	[2.9155136	1.03746732	-0.	0.08742311]

Area and perimeter contribute the most in PC1

```
[[ 1.01010101 -0.02670062  0.69121907  0.76870724]
 [-0.02670062  1.01010101  0.62435684  0.63473235]
 [ 0.69121907  0.62435684  1.01010101  0.94306847]
 [ 0.76870724  0.63473235  0.94306847  1.01010101]]
```

Trace = 4.04
(on standardized data)

not necessarily ordered

$\lambda_1 = 2.916$
$\lambda_2 = 1.037$
$\lambda_3 = 0$
$\lambda_4 = 0.087$

PC1 explain 2.91 of standardized data
$\frac{2.91}{4.04} \approx 80\%$

Note: $\Sigma\lambda_i = 4.04$ (!)

4. Sort eigenvalues in descending order and choose the k highest.
   - How to choose them? **Calculate Explained Variance**
     - Tells us, from the eigenvalues, how much information can be attributed to each principal component.
       - Eigenvalue^2 / N
5. Construct W: projection matrix.
   - This matrix transforms the data onto the new feature subspace.

6. Y = XW

   In this last step we will use the 4×2-dimensional projection matrix $\mathbf{W}$ to transform our samples onto the new subspace via the equation $\mathbf{Y} = \mathbf{X} \times \mathbf{W}$, where $\mathbf{Y}$ is a 150×2 matrix of our transformed samples. Recall that our original data is a 150×4-dimensional matrix.

7. To know how much information we will lose, we can calculate the "**Reconstruction error**":

   $$\mathbf{X}' = \mathbf{Y} \times \mathbf{W}^{\mathbf{T}}$$

   error = $\sum_i (X_i' - X_i)^2$, where i goes over all the elements of the matrix.

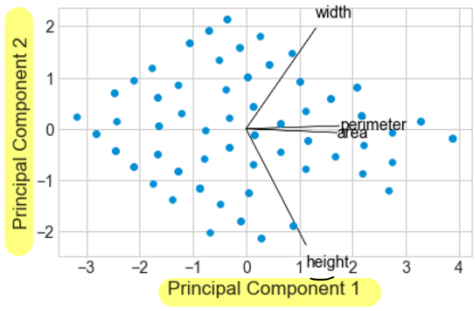
→ Order of steps

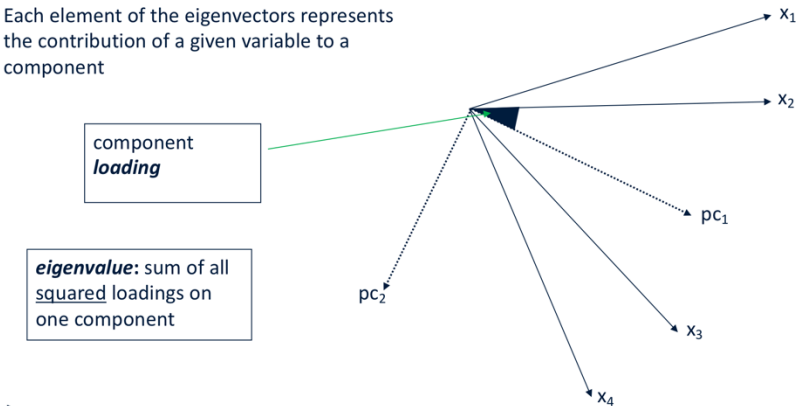
We can clearly see that all three approaches yield the same eigenvectors and eigenvalue pairs:

- Eigendecomposition of the covariance matrix after standardizing the data.
- Eigendecomposition of the correlation matrix.
- Eigendecomposition of the correlation matrix after standardizing the data.

→ Loadings and Eigenvectors

- Loadings: correlations between variables and the principal axes.
- Each element of the eigenvectors represents the contribution of a given variable to a component.





→ Usage of PCA

- Denoising images: by reconstructing image with PCAs.
- Face recognition: by extracting eigenfaces (representation of basis: compresses most important features of image) and then comparing them with new data.
    - If you try to project a new face (not from the training) then the model won't be able to recreate it.

Lecture 9: Interpretability

1. How do we gain trust?

	Interpretable models	Accuracy	A/B testing
Example	Linear/logistic regression, decision trees, EBMs		
Pro	easy to explain and people can see how decision is taken		Gold standard
Cons	often not accurate enough and with complex decisions the tree grows huge.	Unreliable: data leakage, changing environment, training data is not the same as real world	Expensive and tricky to interpret

- Models can be split into Interpretable or agnostic (SVM, Neural network).
- In interpretable models we are interested in inner working.
- In explainable we are interested into why decisions were made.

2. Interpretable models: *why are the following methods interpretable?*

→ Linear regression

- We use coefficients to find which features are the most important.

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \epsilon$$

importance of each coef. for each variable

We interpret β_j as the average effect on Y of a one unit increase in X_j, ***holding all other predictors fixed***.

→ Logistic regression

$$p(X) = p(Y = 1 | X) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_p X_p}}$$

If I increase a predictor value by one unit (and ***keep all others fixed***), then the probability of the "positive" class changes by the odds ratio. Let's see it through an example:

You can interpretate the importance of a predictor by calculating the $odds = \frac{prob\ of\ sucess}{probability\ of\ failure}$, and seeing how the value changes.

Remember that one unit increase in a variable j will have an effect of β_j on the log-odds ratio.

$$log(\text{odds ratio}) = \beta_j$$

or equivalenty (and with the approximation that $log \approx ln$ we have that:

$$\text{odds ratio} = e^{\beta_j}$$

Ex: if odds = 4 then it means that prob of having class 1 is more likely by 4

Out[12]:

	feature	meaning	coefficient	odds
x2	cp	chest pain type	0.775056	2.170713
x10	slope	the slope of the peak exercise ST segment	0.618767	1.856637
x6	restecg	resting electrocardiographic results	0.535698	1.708640
x7	thalach	maximum heart rate achieved	0.027081	1.027451
x0	age	age of patient	0.007980	1.008012
x4	chol	serum cholesterol in mg/dl	-0.001414	0.998587
x5	fbs	fasting blood sugar > 120 mg/dl (1 = true; 0 = false)	-0.001509	0.998492
x3	trestbps	resting blood pressure (in mm Hg on admission to the hospital)	-0.010965	0.989095
x9	oldpeak	ST depression induced by exercise relative to rest	-0.676346	0.508472
x11	ca	number of major vessels (0-3) colored by fluoroscopy	-0.751114	0.471841
x8	exang	exercise induced angina (1 = yes; 0 = no)	-0.829017	0.436478
x12	thal	3 = normal; 6 = fixed defect; 7 = reversable defect	-1.015574	0.362195
x1	sex	1 = male, 0 = female	-1.201090	0.300866

What can you say about this model?

- Chest pain type (`cp`) has a high impact, wow! When this value increases by one, the probability having a heart disease increases by 117.0%.
- If the `slope` variable increases by one, the probability increases by 85.6%.
- `age` and cholesterol (`chol`) have an odds ratio of 1, meaning that they have no effect (probability is 50-50)
- On the bottom we see the `sex` variable. This odds ratio is below 1, around 0.30. This means that having a heart disease is almost 3.3 (1/0.3) times more likely when you're female!

→ Decision trees

- We get feature importance based on the splits.
  - Gini: value calculated at each split; parent node has higher value.
- Feature importance can be calculated as: normalized total reduction of the criterion used for the split (Gini).
  - Keep in mind that feature importance is based on training set.

3. Model-agnostic methods: *these types of methods can be interpretated using.*

→ Permutation feature importance

- Feature importance is calculated: measuring the increase of the prediction error when you permute the feature's values.
- You calculate the prediction error twice: before and after the permutation.
- Important feature: higher difference between prediction errors, because the model depends on this feature.

→ Partial Dependence plots (PDP)

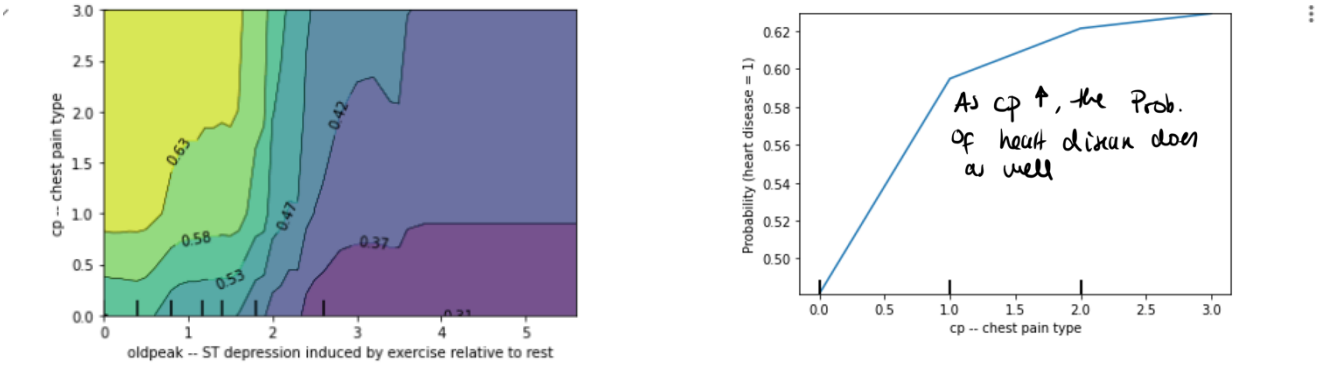
- Help see relationship between target and features.

The process to create the plots is as follows:

- Force all the instances to have the same feature value.
- Make predictions for these instances and average them.
- This gives the average prediction for this feature value.

This method gives you nice visualizations but most of the time only 1-2 features are investigated.

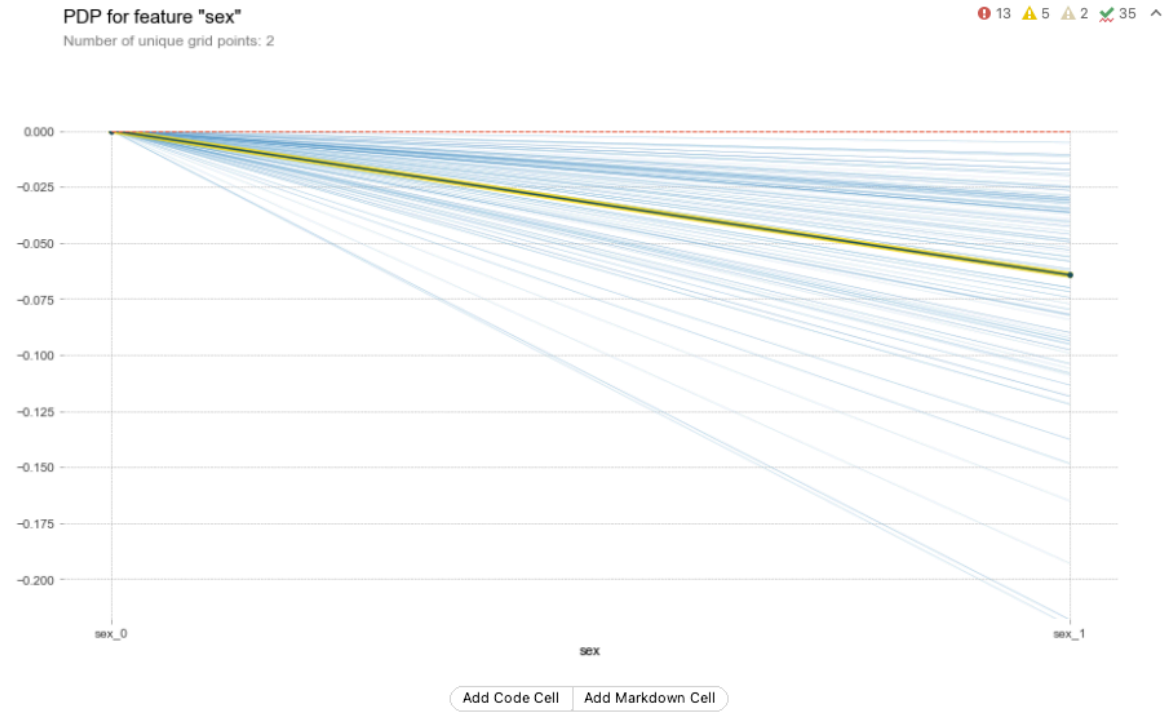
- Not so good because:
  - ○ You force all the instances to have the same feature value.
  - ○ The plots can be misleading if you only have a small amount of instances that have a certain feature value. It's better to include data distributions in your plot, so you can see if the data are equally distributed.
  - ○ It is assumed that features for which you compute the partial dependence are independent. So they shouldn't be correlated with other features. You can also easily miss complexity of the model, because the predictions are averaged.



When chest pain type (`cp`) is equal to 1, 2 or 3 and `oldpeak` has a low value, the probability having a heart disease is much higher (> 0.63) than when cp is equal to 0 and oldpeak has a high value (< 0.37).

→ Individual Conditional Expectation

- More detailed than PDP.
- Show how the prediction of each instance (blue line) changes when a feature value is changed.



Explanation:

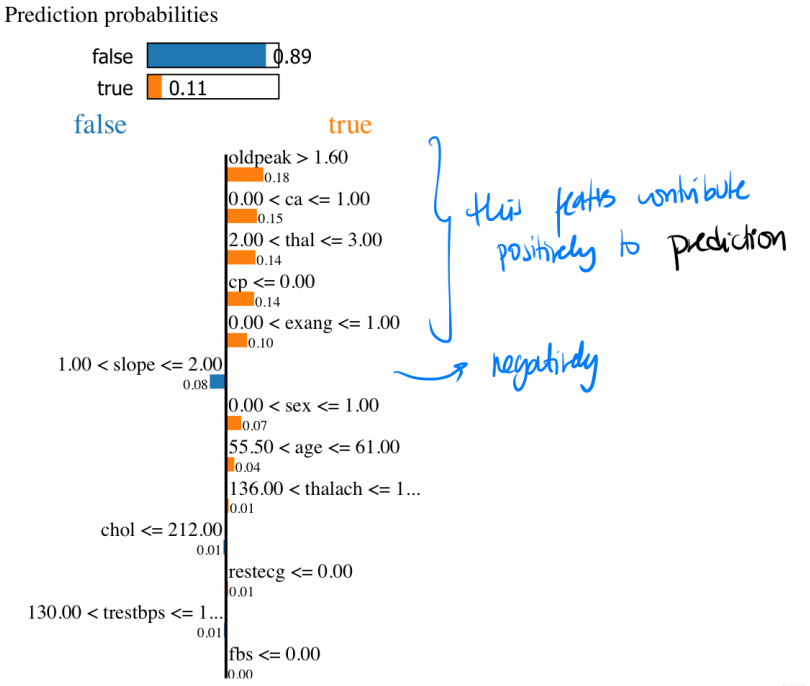
- Each blue line is an instance.
- You see that for some instances the prediction changes a lot when sex is changed to male, but for some instances the prediction almost stays the same, although it always has a negative effect to be female.

→ LIME (non examinable)

The steps are as follows:

1. Sample points around x_i (the point that I want to interpret)
2. I use my big/complex model to predict labels for each sample
3. I weigh samples according to the distance from x_i
4. Learn a new simple model (e.g. decision tree or linear regression) on weighted samples
5. Use simple model to explain the decision

- Cons:
  - If you explain the same record twice, the explanations can be different!
  - You can only explain one instance, so it's not possible to interpret the whole black box model.



→ SHAP

Shapley Values (Game Theory): Each feature value of the instance is “player” in a “game” where the prediction is the “payout”.

Each feature is assigned an importance value for a particular prediction. More specifically SHAP assigns each feature an importance value for a particular prediction to compute the explanation. This value is the unified measure of additive feature attributions and is called SHAP value, $\phi_i \in \mathbb{R}$. The formula for ϕ_i is:

$$\phi_i = \sum_{S \in F \setminus \{i\}} \frac{|S|!(M - |S| - 1)!}{M!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)]$$

where:

- F is the set of input features,
- S is a subset of input features,
- $M = |F|$ is the number of input features.

This formula computes the gravity of each feature by calculating its importance when it is present in the prediction and then subtracting it when it is not present. Moreover,

- $f_{S \cup \{i\}}(x_{S \cup \{i\}})$ is the output when the i-th feature is present
- $f_S(x_S)$ is the output when the i-th feature is withheld
- $\sum_{S \in F \setminus \{i\}} \frac{|S|!(M-|S|-1)!}{M!}$ is the weighted average of all possible subsets of S in F

(+)

- We can use it for local and global explanations.
- Has a strong theoretical foundation.
- Recent research has showed that they can work with dependent features.

(-)

- Computation speed can be very slowwwww (especially with many features)

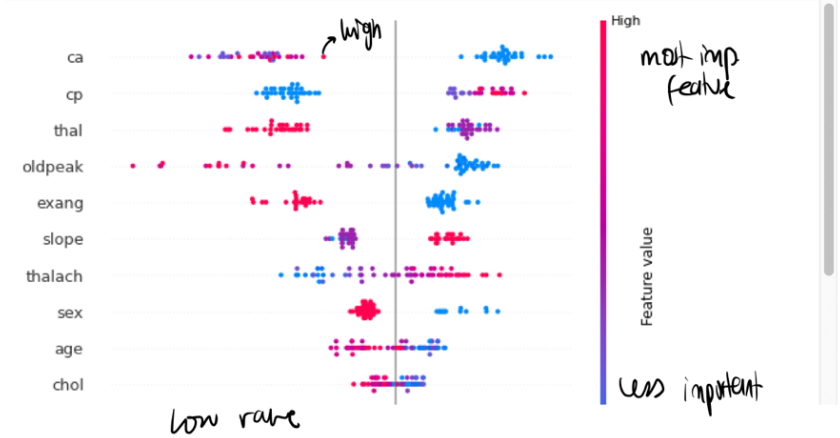
The summary plot shows the different SHAP values for high or low feature values.

For ca feature:

When this feature has a low value, the SHAP value is high, which means a higher probability having a heart disease. This plot also shows us the most important features to the least important ones (top to bottom).

In [38]:

```
shap.summary_plot(shap_values[1], x_test)
```



4. Explainable Boosting Machine (non-examinable)

→ What are they?

- Often as accurate as state-of-the-art blackbox models while remaining completely interpretable.
- Often slower to train than other modern algorithms, EBMs are extremely compact and fast at prediction time.
- The contribution of each feature to a final prediction can be visualized and understood by making a plot.

→ How does it work?

1. In first step of iteration use 1 feature and fit a single model tree.
2. Update the residual (error made in prediction) and move to next feature where you try to fix misclassifications.
3. After all iterations are completed, we calculate the influence of each feature in the final model, and we can show it: