SURVEY ANALYSIS

SURVEY ANALYSIS

- 16 respondents.
- 62.50% have never written any C code before starting the CSLP.
- 31.25% have only written simple C code.
- 43.75% still have difficulties working across different functions with variables allocated on the heap, BUT the majority think they can learn memory management without guidance.
  - If you think this is crucial to your project but cannot figure how out it works, please come and meet me.

SURVEY ANALYSIS (CONT'D)

- Most of you have done janitorial work, but 3/4 do not write reusable code.
- 94% found the lecture on code structuring and coding strategy useful.
- None of you have performed *both* code and compilation optimisations.
- Graph theory expertise: 2 camps (the expert crowd smaller though).
- CSLP difficulty: 75% know what they have to do, but think a lot of work is required.
  - This is where time management becomes particularly important.

SURVEY ANALYSIS (CONT'D)

- 15 out 16 received satisfactory answers from me. If you still have doubts, please get in touch!
- A couple of you noted I haven't used the mic it the AT LT. I apologise for that and am correcting it.
- "It would have been more beneficial if the deadline for the first part was muc later"
  - Finding the right trade off is not straightforward and now it is a bit late, but will improve next year.

QUESTIONS & REQUESTS

Experiment vs simulation

- With simulation your are building an abstract model of a real system which you then analyse.
- Experiment may refer to:
  - Taking a prototype to a test bed, lab, or operational system, to validate it or make new discoveries.
  - Take a course of action when you do not know the outcome, but want to determine it – this is what you do with your simulator.

QUESTIONS & REQUESTS

- More resources about memory management in C
  - Please check the 'Clarifications' tab on the course **web page**.
  - A lot of you know these things, but those who want to acquire more in depth knowledge can also check **this page** on MIT OpenCourseware.

QUESTIONS & REQUESTS

"I really disagree with using C"

1. Not using a OOP language is making everything unnecessarily hard.
2. Relevance of C is questionable (TIOBE index based on search results).
3. Missing abstract types, could just use Java or Python instead.

WHY C?

1. C compiles down to machine code, i.e. you can run code before even having a file system or processes.
2. It's ultra fast and portable.
3. This is a Computer Science project – I could have asked you to code a device driver, but that would have been extremely tough for people with no C experience (quite an important number).
4. TIOBE does not say what is the **best** programming language – you can check yourselves if C developers are in demand and how they are remunerated ;-)

QUESTIONS & REQUESTS

- Input parsing/handling in C – quite specific, but will give some tips and you can come and talk to me.
- Overview of architectural patterns for the simulator
  – I will cover some aspects on that today.
- A number of you are interested in more techniques for manipulating graphs/perform routing – will cover that during the next lecture without suggesting a particular solution.

INPUT PARSING

TIPS ON INPUT PARSING

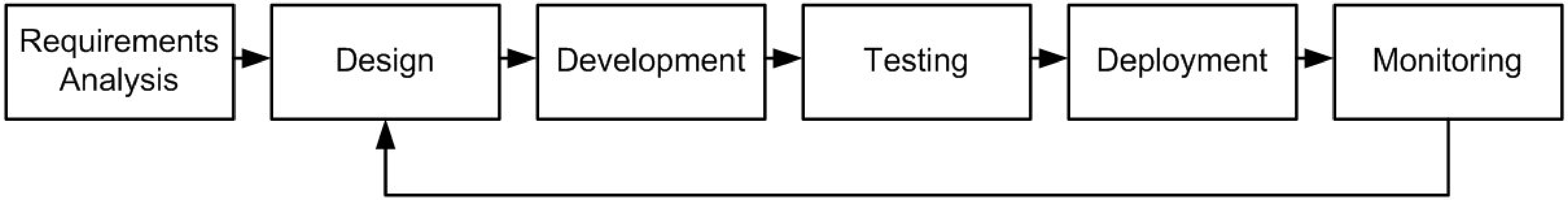
- You are given the input parameters in a **plain text** file.
- There a multiple ways to parse the contents of such files. I will not go into details, but give some tips.
- One approach:
  1. Read the file line by line with `fgets`.
  2. Split the line into tokens.
  3. Use `strcmp, strtol` and `strtod` to convert tokens into different data types.

TIPS ON INPUT PARSING

- Another approach:
  1. Work with `fscanf` and `sscanf` to read.
  2. Advance the file position indicator with `fseek`.
  3. May prove trickier when parameters are not encountered in the expected order.
- Employing a `switch`approach with multiple cases is likely quite appropriate

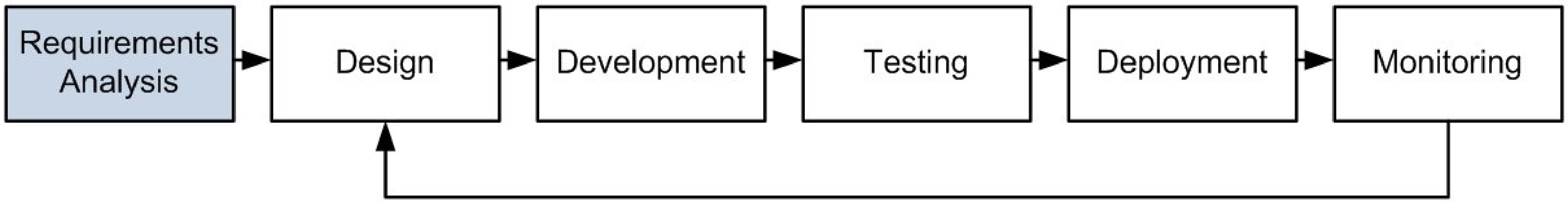
DESIGN ASPECTS

SYSTEM/PROCESS IMPLEMENTATION



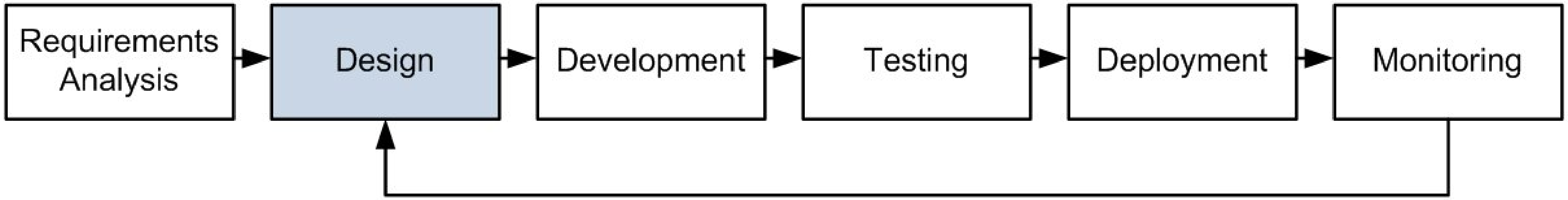
- Designing and implementing logistics operations, complex processes, and systems involves several steps.
- There is often a feedback loop involved, which allows to refine/improve/extend the system.

REQUIREMENTS ANALYSIS



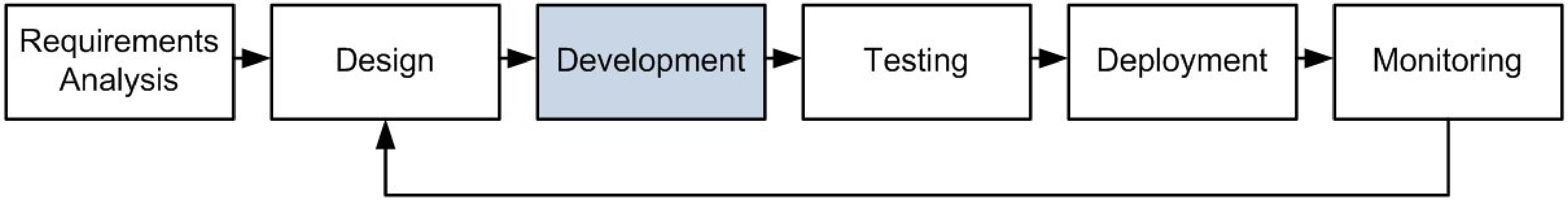
- Understand the problem domain and specifications, and identify the key entities involved.
- Build an abstract representation of the system to be able to handle various input scenarios.

SYSTEM DESIGN



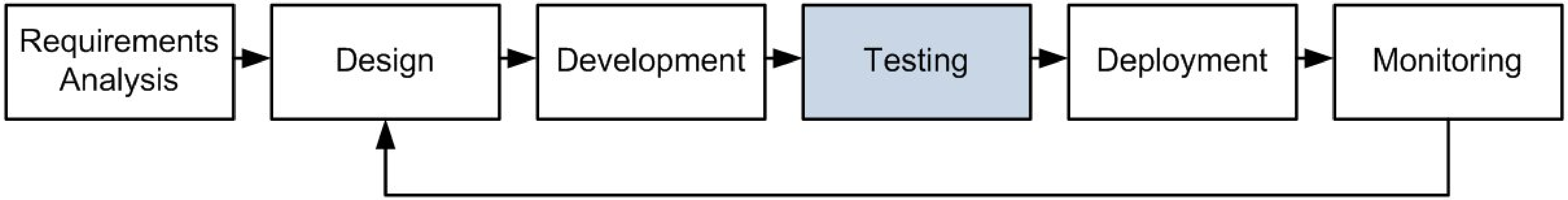
- Dividing the system into components; choosing suitable methodologies for implementation each component.
- Defining appropriate data structures, input/output formats, and so on.

DEVELOPMENT



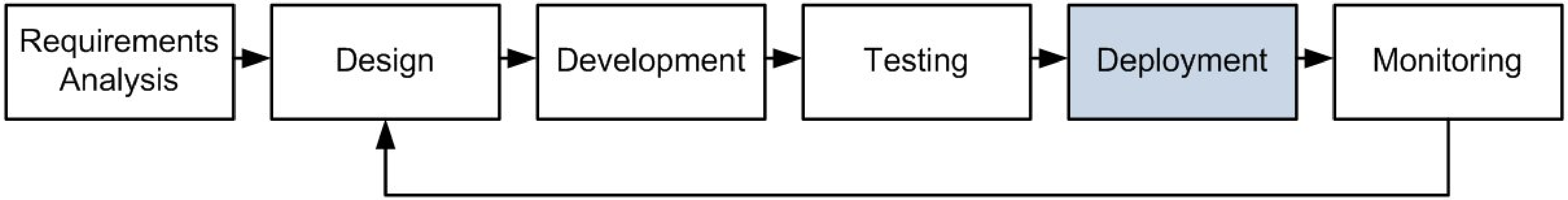
- This is the actual implementation work and is typically coupled with some preliminary testing.
- For source code, janitorial work, refactoring and some optimisations are also performed at this stage.

TESTING



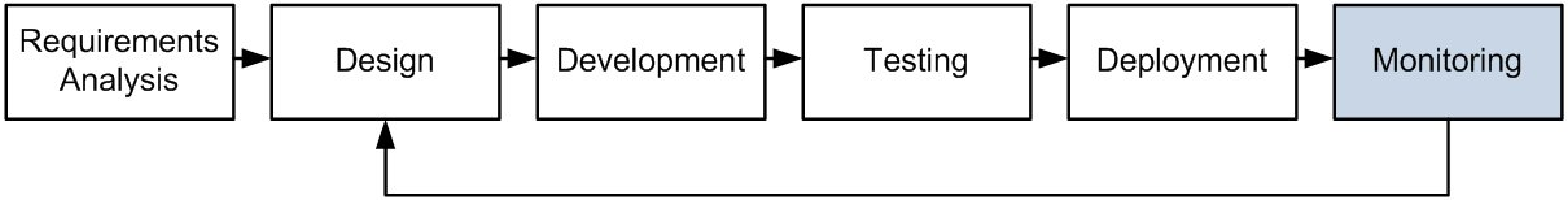
- Validation is performed once the system is partially/ entirely developed; also benchmarking and profiling.
- A system's *performance evaluation* is undertaken (experimentation with different inputs, distributions).

DEPLOYMENT



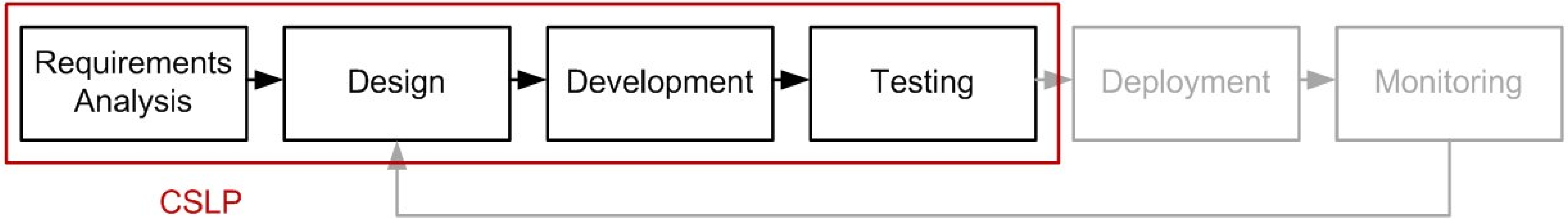
- Once the tool (planner, simulator, etc.) has been thoroughly tested it can be deployed in a real setting.
- The input will be based on actual data and inputs may change over time (e.g. based on certain events).

MONITORING



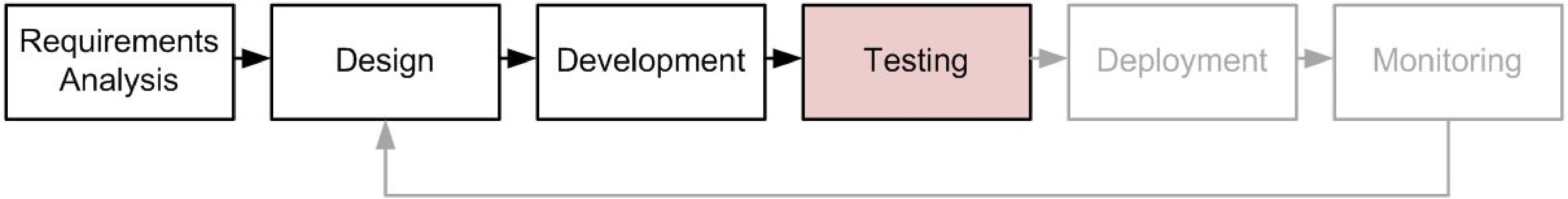
- Once the system is operational, it is possible to gather real measurements an[d] use those to refine the design.
- If new requirements are identified during operation, the system can be furthe[r] extended.

THE BIN SERVICE PROCESS



- Your simulator will be implementing a good bit of what could become a real logistics system.
- Unfortunately you will not have the opportunity to experiment with real data but (time permitting) you have the flexibility to develop additional features.

PERFORMANCE EVALUATION



- We have discussed the requirements, as well as different design and development aspects for your simulator.
- We will now look into performance evaluation issues. Some of the things I wi present may not be needed for this assignment, but will likely prove useful later.

PERFORMANCE EVALUATION

- Generally speaking, this is about quantifying the performance of a system.
- The first step is to identify the relevant *metrics*, i.e. measurable quantities tha capture properties of interest.
  - This could the throughput of a network link, the power consumption of a mobile device, the memory used by a software application, etc.
  - For CSLP we are interested in the average trip duration, trip efficiency, percentage of missed requests, average passenger waiting time, average trip deviation.

METRICS

- It is essential to understand the performance evaluation goals, i.e. whether a metric should be small or large.
- It is also important to be aware of the goals of the evaluation:
  - Improve the dimensioning/parametrisation of a system or process.
  - Compare how different designs perform under different inputs and chose the best one.

METHODOLOGIES

When designing a system, performance evaluation can be conducted through one or more of the following methodologies:

1. **Numerical analysis** - plugging some numerical values into a mathematical model of the system and computing the metrics of interest.
2. **Simulation** - constructing a simplified model of a more complex real system and simulating its behaviour; typically fast, but neglecting certain practical aspects.
3. **Experimentation** - Analysing the performance of a system via measurements. Assessing performance under exceptional circumstances may be infeasible.

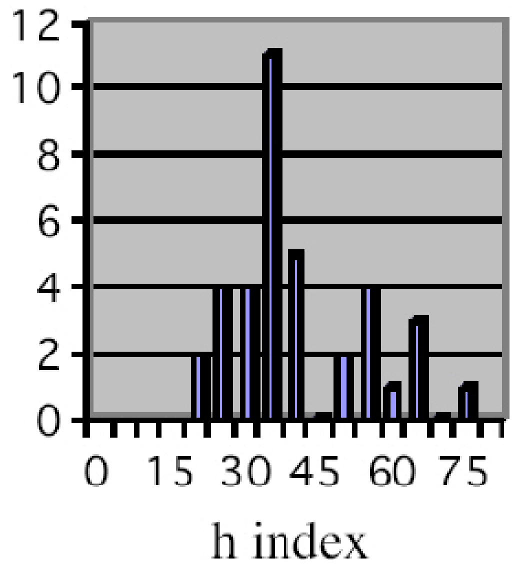
ACCURACY

- It is advisable that the assumptions made for the evaluation campaign are we documented, to ensure the tests performed are *reproducible*.
- You are working with a stochastic simulator and thus there will be some variability in the results of different tests with the same input.
- For this practical you have been asked to give average values of a set of metrics.
- In rigorous studies, it is necessary to also provide some confidence intervals for the results.

SUMMARY STATISTICS

Histograms are graphical representations of the distribution of a set of measurements.

Example: distribution of the *h*-index of Nobel-prize recipients in Physics between 1985-2005.



Source: J.E. Hirsch, "An index to quantify an individual's scientific research output", Proc. NAS, 2005.

h-index: number of papers with *h* or more citations.

HISTOGRAMS

- In mathematical terms, the histogram is a function that counts the number o observations in different categories (bins)
- The number of bins is typically computed as

$$k = \frac{\max(x) - \min(x)}{\sqrt{n}}$$

where n is the number of samples in the data set x.

MEAN AND STANDARD DEVIATION

Computing the *mean* (average) of a set of measurements is straightforward:

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$$

The *standard deviation* gives a measure of the variation of the measurements from the mean:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2}$$

CONFIDENCE INTERVALS

- These can be used to quantify the uncertainty about the average of a set of measurements subject to randomness.
- When computing averages across multiple simulations, you are gathering samples to estimate an unknown population mean.
- You choose the significance level that will reflect how confident you can be that the true value lies within that interval,
- E.g. for a significance level of 0.05, you will obtain a 95% confidence interval (typically used in practice).

CONFIDENCE INTERVALS

- The width of the confidence is affected by:
  - sample size,
  - population variability (standard deviation),
  - confidence level chosen.
- Central Limit Theorem: For a large sample size, the sampling distribution of the mean will approach a *normal distribution*.
- The sample mean and the mean of the population are identical.

CONFIDENCE INTERVALS

A quick method to compute a CI is:

$$\mu \pm z_{\alpha/2} \frac{\sigma}{\sqrt{n}}$$

where $z_{\alpha/2}$ is the critical coefficient corresponding to a confidence level α and i obtained from z-score tables.

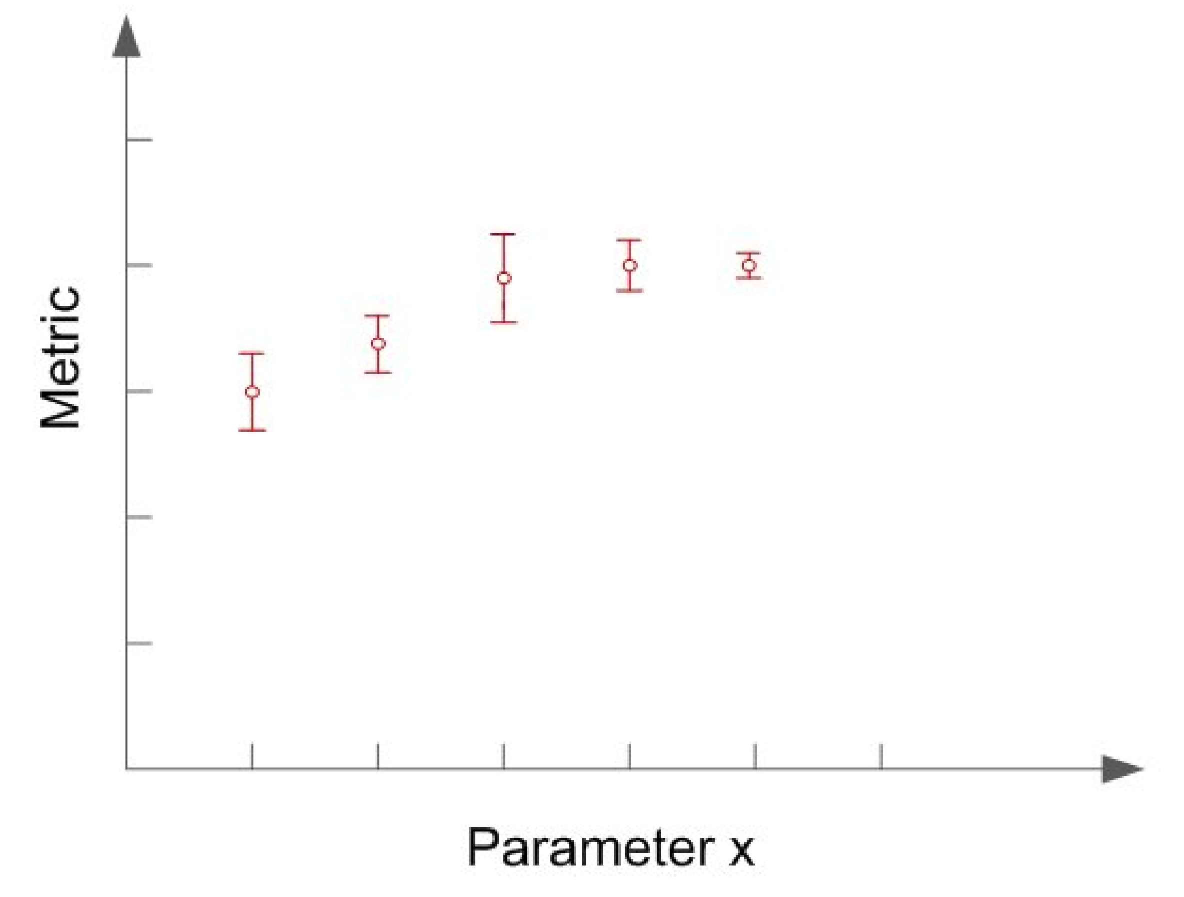
Example: Sample size 20, mean 10, standard deviation 1.45, 95% confidence level, i.e. a critical coefficient corresponding to a z-score of 0.475, which is 1.96.

CI is 20±0.02

i.e. [19.8, 20.2]

CONFIDENCE INTERVALS

Plotting CIs



QUESTIONS?

PLANNING FOR THE REMAINING DURATION OF THE SEMESTER

- 13 Nov – No lecture (next week).
- 20 Nov – Manipulating graphs and final points.
- 27 Nov – **Guest lecture:** Dr George Hazel, Scottish Enterprise (will speak about efforts to implement on demand mobility schemes in Scotland.)
- Reminder: Deadline for submitting Part 2 is the **21st December at 16:00**.
- Feedback by 21 January.