Goals of this part of the course (about 2 lectures):
Provide a view of issues in Experimental Algorithmics

- Exploratory data analysis
- Presenting results in concise, meaningful graphs and tables
- Basics of inferential statistics
- Organization of Experiments (Experimental Design)
- Sequential statistical testing: a methodology for tuning

The goal of dealing with Experimental Algorithmics is not only producing a more correct analysis but adding an important tool to the development of a good, functioning solver for a given problem.

Experimental Algorithmics is an important part in the algorithm production cycle, which is referred to as Algorithm Engineering.
In empirical studies we consider simulation programs which are the implementation of a mathematical model (the algorithm).

Algorithmic models of programs can vary according to their level of instantiation:
- minimally instantiated (algorithmic framework), e.g., simulated annealing
- mildly instantiated: includes implementation strategies (data structures)
- highly instantiated: includes details specific to a particular programming language or computer architecture

**Definitions**

**Random Variables and Probability**

Statistics deals with random (or stochastic) variables.
A variable is called random, if, prior to observation, its outcome cannot be predicted with certainty.

The uncertainty is described by a *Probability Distribution*.

**Discrete variables**

- Probability distribution:
  \[
  p_i = P[X = y_i]
  \]

**Continuous variables**

- Probability density function (pdf):
  \[
  f(v) = \frac{df(v)}{dv}
  \]

- Cumulative Distribution Function (CDF):
  \[
  F(v) = P[X \leq v] = \sum_{i} p_i
  \]

- Mean: \[ \mu = E[X] = \sum_{i} y_i p_i \]

- Variance: \[ \sigma^2 = E[(X - \mu)^2] = \sum_{i} (y_i - \mu)^2 p_i \]

**Generalization**

On a specific instance, the random variable \( Y \) that defines the performance measure of an algorithm is described by its probability distribution/density function

\[ p(y|\pi) \]

It is often more interesting to generalize the performance on a class of instances \( C_\Pi \), that is,

\[ p(y, C_\Pi) = \sum_{\pi \in \Pi} p(y|\pi)p(\pi) \]

**Sampling**

In experiments,
- we sample the population of instances and
- we sample the performance of the algorithm on each sampled instance

If on an instance \( \pi \) we run the algorithm \( r \) times then we have \( r \) replicates of the performance measure \( Y \), denoted \( Y_1, \ldots, Y_r \), which are independent and identically distributed (i.i.d.), i.e.

\[ p(y_1, \ldots, y_r|\pi) = \prod_{i=1}^r p(y_i|\pi) \]

\[ p(y_1, \ldots, y_r) = \sum_{\pi \in C_\Pi} p(y_1, \ldots, y_r|\pi)p(\pi). \]
In real-life applications a simulation of \( p(\pi) \) can be obtained by historical data. In simulation studies instances may be:

- real world instances
- random variants of real world-instances
- online libraries
- randomly generated instances

They may be grouped in classes according to some features whose impact may be worth studying:

- type (for features that might impact performance)
- size (for scaling studies)
- hardness (focus on hard instances)
- application (e.g., CSP encodings of scheduling problems), ...

Within the class, instances are drawn with uniform probability \( p(\pi) = c \).

**Test Instance Selection**

**Statistical Methods**

The analysis of performance is based on finite-sized sampled data. Statistics provides the methods and the mathematical basis to:

- describe, summarizing the data (descriptive statistics)
- make inference on those data (inferential statistics)

Statistics helps to:

- guarantee reproducibility
- make results reliable (are the observed results enough to justify the claims?)
- help to extract relevant results from large amount of data

In the practical context of heuristic design and implementation (i.e., engineering), statistics helps to take sound decisions with the least amount of experimentation.

**Objectives of the Experiments**

**Comparison:**

- bigger/smaller, same/different
- Algorithm Configuration
- Component-Based Analysis

Standard statistical methods: experimental designs, test hypothesis and estimation

**Measures and Transformations**

**On a single instance**

Computational effort indicators

- number of elementary operations/algorithmic iterations (e.g., search steps, objective function evaluations, number of visited nodes in the search tree, consistency checks, etc.)
- CPU time (real time as measured by OS functions)

Solution quality indicators

- value returned by the cost function (or error from optimum/reference value)

**Measures and Transformations**

**On a class of instances**

Computational effort indicators

- no transformation if the interest is in studying scaling
- standardization if a fixed time limit is used
- otherwise, better to group homogeneously the instances

Solution quality indicators

Different instances implies different scales ⇒ need for an invariant measure
But also, see [?].

**Outline**

1. Algorithm Engineering and Experimental Analysis
   - Definitions
   - Performance Measures

2. Exploratory Data Analysis
   - Representation of Sampled Data
   - Regression Analysis
Summary Measures for Sampled Data

Measures to describe or characterize a population
- Measure of central tendency, location
- Measure of dispersion

One such a quantity is
- a parameter if it refers to the population (Greek letters)
- a statistics if it is an estimation of a population parameter from the sample (Latin letters)

Measure of dispersion
- Sample range
  \[ R = x_n - x_1 \]
- Sample variance
  \[ s^2 = \frac{1}{n-1} \sum (x_i - \bar{X})^2 \]
- Standard deviation
  \[ s = \sqrt{s} \]
- Inter-quartile range
  \[ IQR = Q_3 - Q_1 \]

Scenarios

- Single-pass heuristics
- Asymptotic heuristics:
  Two approaches:
  1. Univariate
     1.a Time as an external parameter decided a priori
     1.b Solution quality as an external parameter decided a priori
  2. Cost dependent on running time:

Definitions

Single-pass heuristics

Deterministic case: \( A^\perp \) on \( \pi \) returns a solution of cost \( x \) with computational effort \( t \) (e.g., running time).

The performance of \( A^\perp \) on \( \pi \) is the vector \( y = (x, t) \).

Randomized case: \( A^\perp \) on \( \pi \) returns a solution of cost \( X \) with computational effort \( T \), where \( X \) and \( T \) are random variables.

The performance of \( A^\perp \) on \( \pi \) is the bivariate \( Y = (X, T) \).

Single-pass Heuristics

Bivariate analysis: Example

Scenario:
- 3 heuristics \( A^\perp_1, A^\perp_2, A^\perp_3 \) on class \( C^\perp \).
- homogeneous instances or need for data transformation.
- 1 or \( r \) runs per instance
- Interest: inspecting solution cost and running time to observe and compare the level of approximation and the speed.

Tools:
- Scatter plots of solution-cost and run-time

Measures of central tendency
- Arithmetic Average (Sample mean)
  \[ \bar{X} = \frac{\sum x_i}{n} \]
- Quantile: value above or below which lie a fractional part of the data
  used in nonparametric statistics
  - Median
    \[ M = x_{(n+1)/2} \]
  - Quantile
    \[ Q_1 = x_{n+1/4} \quad Q_3 = x_{3(n+1)/4} \]
  - q-quantile
    \[ q \text{ of data lies below and } 1 - q \text{ lies above} \]
- Mode
  value of relatively great concentration of data
  \( \text{(Unimodal vs Multimodal distributions)} \)

R functions:

- \( \text{mean(x), median(x), quantile(x), quantile(x, 0.25)} \)
- \( \text{range(x), var(x), sd(x), IQR(x)} \)
- \( \text{fivenum(x)} \)

Summary Measures for Sampled Data

Measures to describe or characterize a population
- Measure of central tendency, location
- Measure of dispersion

One such a quantity is
- a parameter if it refers to the population (Greek letters)
- a statistics if it is an estimation of a population parameter from the sample (Latin letters)

Measure of dispersion
- Sample range
  \[ R = x_n - x_1 \]
- Sample variance
  \[ s^2 = \frac{1}{n-1} \sum (x_i - \bar{X})^2 \]
- Standard deviation
  \[ s = \sqrt{s} \]
- Inter-quartile range
  \[ IQR = Q_3 - Q_1 \]

Scenarios

- Single-pass heuristics
- Asymptotic heuristics:
  Two approaches:
  1. Univariate
     1.a Time as an external parameter decided a priori
     1.b Solution quality as an external parameter decided a priori
  2. Cost dependent on running time:

Definitions

Single-pass heuristics

Deterministic case: \( A^\perp \) on \( \pi \) returns a solution of cost \( x \) with computational effort \( t \) (e.g., running time).

The performance of \( A^\perp \) on \( \pi \) is the vector \( y = (x, t) \).

Randomized case: \( A^\perp \) on \( \pi \) returns a solution of cost \( X \) with computational effort \( T \), where \( X \) and \( T \) are random variables.

The performance of \( A^\perp \) on \( \pi \) is the bivariate \( Y = (X, T) \).

Single-pass Heuristics

Bivariate analysis: Example

Scenario:
- 3 heuristics \( A^\perp_1, A^\perp_2, A^\perp_3 \) on class \( C^\perp \).
- homogeneous instances or need for data transformation.
- 1 or \( r \) runs per instance
- Interest: inspecting solution cost and running time to observe and compare the level of approximation and the speed.

Tools:
- Scatter plots of solution-cost and run-time

R functions:

- \( \text{mean(x), median(x), quantile(x), quantile(x, 0.25)} \)
- \( \text{range(x), var(x), sd(x), IQR(x)} \)
- \( \text{fivenum(x)} \)
Asymptotic heuristics

There are two approaches:

1. Time as an external parameter decided \( a \ priori \).
   The algorithm is halted when time expires.

2. Randomized case: \( A^\infty \) on \( \pi \) returns a solution of cost \( x \).
   The performance of \( A^\infty \) on \( \pi \) is the
   univariate \( Y = x \).

Deterministic case: \( A^m \) on \( \pi \) returns a solution of cost \( x \).
The performance of \( A^m \) on \( \pi \) is the scalar \( y = x \).

Asymptotic Heuristics

Approach 1.a, Univariate analysis

Scenario:
- 3 heuristics \( A_1^\infty, A_2^\infty, A_3^\infty \) on class \( C_1 \).
  (Or 3 heuristics \( A_1^\infty, A_2^\infty, A_3^\infty \) on class \( C_2 \) without interest in computation time because negligible or comparable)
- homogeneous instances (no data transformation) or heterogeneous (data transformation)
- 1 or \( r \) runs per instance
- a \( \alpha \) priori time limit imposed

Interest: inspecting solution cost

Tools:
- Histograms (summary measures: mean or median or mode?)
- Boxplots
- Empirical cumulative distribution functions (ECDFs)

R functions:

```r
Y = \frac{\bar{x}}{\sigma}
```

Data representation

Asymptotic Heuristics

Approach 1.a, Univariate analysis: Example

R functions:

```r
G[1:5,]
V1 V2 V3 V4 V5 V6 V7 V8 V9
1 119 0.88012 0.0 0.188012
2 117 0.84012 0.0 0.188012
3 117 0.84012 0.0 0.188012
4 116 0.82012 0.0 0.182012
5 116 0.82012 0.0 0.182012
```

R functions:

```r
G$scale <- 0
```

Transform data in standard error

R functions:

```r
> gscale <- 0
> Gscale <- lapply(split(Gsol, Ginst), scale,center=TRUE,scale=gscale)
```

Transform the data in relative error

R functions:

```r
> G$err2 <- (G$sol-G$opt)/G$opt
```

Maintain the raw data

R functions:

```r
> #original data
> G$err2 <- (G$sol-G$opt)/G$opt
> boxplot(DSATUR,horizontal=TRUE,ylim=limits,xlab=expression("Relative error: \( \frac{x-x^{(opt)}}{x^{(opt)}} \))
```

If we want to make an aggregate analysis we have the following choices:
- maintain the raw data, transform data in standard error, transform the data in relative error, transform the data in an invariant error, or transform the data in ranks.

R functions:

```r
> #relative error
> G$err2 <- (G$sol-G$opt)/G$opt
```

```r
> boxplot(DSATUR,horizontal=TRUE,ylim=limits,xlab=expression("Relative error: \( \frac{x-x^{(opt)}}{x^{(opt)}} \))
```

We load the data and plot the comparative boxplot for each instance.

```r
> load("TS5.dataR")
```

```r
> T4 <- split(G$sol,list(G$inst))
```

```r
> T4 <- lapply(T3,scale,center=TRUE,scale=TRUE)
```

```r
> I <- as.data.frame(t(apply(H,1,rank)))
```

```r
> boxplot(V4~V1,horizontal=TRUE,notch=TRUE)
```

```r
> T3 <- lapply(T2,scale,center=TRUE,scale=TRUE)
```

```r
> T2 <- unsplit(T1,list(G$inst))
```

```r
> G$inst <- lapply(split(G$inst), scale,center=TRUE,scale=gscale)
```

```r
> boxplot(sol~alg,data=G,horizontal=TRUE,main=expression(paste("Randomized case: \( A^\infty \) on \( \pi \) returns a solution of cost \( x \).
   The performance of \( A^\infty \) on \( \pi \) is the
   univariate \( Y = x \).

The performance of \( A^m \) on \( \pi \) is the scalar \( y = x \).\n
Data representation

Asymptotic Heuristics

Approach 1.a, Univariate analysis

Scenario:
- 3 heuristics \( A_1^\infty, A_2^\infty, A_3^\infty \) on class \( C_1 \).
  (Or 3 heuristics \( A_1^\infty, A_2^\infty, A_3^\infty \) on class \( C_2 \) without interest in computation time because negligible or comparable)
- homogeneous instances (no data transformation) or heterogeneous (data transformation)
- 1 or \( r \) runs per instance
- a \( \alpha \) priori time limit imposed

Interest: inspecting solution cost

Tools:
- Histograms (summary measures: mean or median or mode?)
- Boxplots
- Empirical cumulative distribution functions (ECDFs)

R functions:

```r
> G$inst <- lapply(split(G$inst), scale,center=TRUE,scale=gscale)
```

Transform the data in relative error

R functions:

```r
> G$err2 <- (G$sol-G$opt)/G$opt
```

Maintain the raw data

R functions:

```r
> #original data
> boxplot(sol~alg,data=G,horizontal=TRUE,main="Original data")
```

Measures of central tendency (mean, median, mode)
and dispersion (variance, standard deviation, inter-quartile)

Transform data in standard error

R functions:

```r
> gscale <- 0
> split(Gscale, Ginst) <- lapply(split(Gsol, Ginst), scale,center=TRUE,scale=gscale)
```

If we want to make an aggregate analysis we have the following choices:
- maintain the raw data, transform data in standard error, transform the data in relative error, transform the data in an invariant error, or transform the data in ranks.

R functions:

```r
> gscale <- 0
> split(Gscale, Ginst) <- lapply(split(Gsol, Ginst), scale,center=TRUE,scale=gscale)
```

Maintain the raw data

R functions:

```r
> #original data
> boxplot(sol~alg,data=G,horizontal=TRUE,main="Original data")
```
Transform the data in an invariant error
We use as surrogate of $x_{worst}$ the median solution returned by the simplest algorithm for the graph coloring, that is, the ROS heuristic.

> `error 3`
> load("ROS.class-G.data")
> F1 <- aggregate(F[1],list Inst=F[1]$inst),median)
> F2 <- split(F,Inst= instantiation)
> G$err3 <- (G$sol-G$opt)/(G$ref-G$opt)
 hustle <- snapshot(G$inst,paste("

Asymptotic Heuristics
Approach 1.a, Univariate analysis: Example

On a class of instances

Stochastic Dominance

Definition: Algorithm $A_1$ probabilistically dominates algorithm $A_2$ on a problem instance, iff its CDF is always “below” that of $A_2$, i.e.: $F_1(x) \leq F_2(x), \forall x \in X$

Dealing with Censored Data
Asymptotic heuristics, Approach 1.b

Dealing with Censored Data
Asymptotic heuristics, Approach 1.b: Example

Definitions

Asymptotic heuristics
There are two approaches:

1.b. Solution quality as an external parameter decided a priori. The algorithm is halted when quality is reached.

Deterministic case: $A^\infty$ on $\pi$ finds a solution in running time $t$. The performance of $A^\infty$ on $\pi$ is the scalar $y = t$.

Randomized case: $A^\infty$ on $\pi$ finds a solution in running time $T$, where $T$ is a random variable.

The performance of $A^\infty$ on $\pi$ is the univariate $Y = T$.

An exact vs an heuristic algorithm for the 2-Edge-connectivity augmentation problem.

Interest: time to find the optimum on different instances.

Uncensored:

$$ F(t) = \frac{\# \text{ runs } < t}{\# \text{ runs } } $$

Censored:

$$ F(t) = \frac{\# \text{ runs } < t}{\# \text{ runs } } $$
Asymptotic heuristics
There are two approaches:

1. Cost dependent on running time:
   Deterministic case: $A^\infty$ on $\pi$ returns a current best solution $x$ at each observation in $t_1, \ldots, t_k$.
   The performance of $A^\infty$ on $\pi$ is the profile indicated by the vector $\bar{y} = (\bar{x}(t_1), \ldots, \bar{x}(t_k))$.

2. Cost dependent on running time:
   Randomized case: $A^\infty$ on $\pi$ produces a monotone stochastic process in solution cost $X(t)$ with any element dependent on the predecessors.
   The performance of $A^\infty$ on $\pi$ is the multivariate $\bar{Y} = (X(t_1), X(t_2), \ldots, X(t_k))$.

Asymptotic Heuristics
Approach 2, Multivariate analysis: Example

The performance is described by multivariate random variables of the kind $\bar{Y} = (Y(t_1), Y(t_2), \ldots, Y(t_k))$.

Sampled data are of the form $\bar{Y}_i = (Y(t_1), Y(t_2), \ldots, Y(t_k))$, $i = 1, \ldots, 10$ (10 runs per algorithm on one instance).

Correlation Analysis

Scenario:
- heterogeneous instances, hence data transformation
- or $\tau$ run per instance
- consider time to goal or solution quality
- Interest: inspecting whether instances are all equally hard to solve for different algorithms or whether some features make the instances harder

Tools: correlation plots (each point represents an instance), correlation coefficient:

\begin{align*}
\rho_{XY} &= \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \\
\tau_{XY} &= \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{(n - 1)\sigma_X \sigma_Y}
\end{align*}

Scaling Analysis

Scenario:
- one heuristic $A^+$ or $A^\infty$ with a priori quality goal
- data collected on instances of different size or features
- Interest: characterizing the growth of computational effort.

Given a set of data points $(N_i, Y_i)$ obtained from an experiment in which $Y_i = f(N_i)$, for some unknown function $f(\pi)$, Find growth function class $O(g_a(\pi))$ and/or $\Omega(g_b(\pi))$ to which $f(\pi)$ belongs.

Mix of interpolation of the data trend and extrapolation beyond the range of experimentation.
Making plots

- Linking points by straight line in a plot. Should not be viewed as an interpolation but rather as a visual aid to find points that belong together.
- Plotting measuring points plus curves stemming from analytical model.
- If very dense measurements are done and they form a smooth line then it is possible to use interpolation:
  - Log-log transformation and linear regression
  - Smoothing techniques
Such curves should only be used if we actually conjecture that the interpolation used is close to the truth.

Linear Regression

Simple Linear Regression: dependent variable + independent variable

\[ Y_i = \beta_0 + \beta X_i + \epsilon_i \]

Uses the Least Squares Method:

\[ \min \sum e_i^2 = \sum (Y_i - \beta_0 - \beta X_i) \]

The indicator of the quality of fitness is the coefficient of determination \( R^2 \) (but use with caution)

Multiple Linear regression considers multiple predictors

\[ Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \epsilon_i \]

The indicator of the quality of fitness is the adjusted \( R^2 \) statistic

Making Plots

- Should the experimental setup from the exploratory phase be redesigned to increase conciseness or accuracy?
- What parameters should be varied? What variables should be measured?
- How are parameters chosen that cannot be varied?
- Can tables be converted into curves, bar charts, scatter plots or any other useful graphics?
- Should tables be added in an appendix?
- Should a 3D-plot be replaced by collections of 2D-curves?
- Can we reduce the number of curves to be displayed?
- How many figures are needed?
- Scale the x-axis to make y-values independent of some parameters?
- Should the x-axis have a logarithmic scale? If so, do the x-values used for measuring have the same basis as the tick marks?
- Should the x-axis be transformed to magnify interesting subranges?

- Connect points belonging to the same curve.
- Only use splines for connecting points if interpolation is sensible.
- Do not connect points belonging to unrelated problem instances.
- Use different point and line styles for different curves.
- Use the same styles for corresponding curves in different graphs.
- Place labels defining point and line styles in the right order and without concealing the curves.
- Captions should make figures self contained.
- Give enough information to make experiments reproducible.

Scaling Analysis

Example

Running time of RLF for Graph Coloring: with two regressors: # vertices and edge density

Uniform random graphs

- Is the range of x-values adequate?
- Do we have measurements for the right x-values, i.e., nowhere too dense or too sparse?
- Should the y-axis be transformed to make the interesting part of the data more visible?
- Should the y-axis have a logarithmic scale?
- Is it misleading to start the y-range at the smallest measured value?
- Clip the range of y-values to exclude useless parts of curves?
- Can we use banking to 45?
- Are all curves sufficiently well separated?
- Can noise be reduced using more accurate measurements?
- Are error bars needed? If so, what should they indicate? Remember that measurement errors are usually not random variables.